



Git bash for windows

**Greg Foster**

Graphite software engineer

Note

This guide explains this concept in vanilla Git. For Graphite documentation, see our [CLI docs](#).

[Git Bash](#) is a command-line interface for using [Git for Windows](#), providing users with a [Bash](#) emulation to run Git commands with more flexibility and efficiency. This guide will cover the essentials of Git Bash, from downloading and installing, to package management and executing your first commands.



Become a Git expert in 5 min.
with our free cheat-sheet

Your email

Download for free

Join 20,000+ developers at

Stop wrestling with Git commands

The Graphite CLI takes all the pain out of Git, allowing you to ship faster and stop googling Git commands.

[Learn more](#)[Request a Demo](#)

Downloading and installing Git Bash on Windows

1. **Git download for Windows:** To get started with Git Bash, download the latest version of Git for Windows from the official website git-scm.com. It is available for various Windows versions including Windows 10, Windows 11, and older versions like Windows 7 and Windows 8.
2. **Git install Windows:** Run the downloaded `.exe` file to start the installation process. During installation, you will encounter various setup options. Default settings are typically sufficient for most users, but you can customize components and the default editor used for Git commands if necessary.
3. **Git Bash setup:** Among the options during installation, you'll choose how Git is run in the command prompt. We recommend selecting the "Use Git from the Windows Command Prompt" option, unless you have specific additional Unix tools you want to install alongside Git Bash. You can always change this setting later.

Basic Git bash commands

After installation, you can access Git Bash through the [Start menu](#) or by clicking on the shortcut you created during installation.

Git Bash operates on a [unix](#) would on a [unix system](#).

Become a Git expert in 5 min.
with our free cheat-sheet

- **Navigating directories**
- **Creating directories:** `mkdir <dir-name>`
- **Creating files:** `touch <file-name>` Creates a new file.

- **Checking Git version:** `git --version` shows the installed Git version, helpful for ensuring you have the latest features and security updates.

Git configuration

Before you start using Git, configure your user name and email address:

TERMINAL



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

These details are important because every Git commit includes this information in its metadata.


TIP: If you want to change either field temporarily, you can run this command later without the `--global`, which will set email and name in the scope of the current git project.

Your first git repository

1. **Create a workspace directory:** It's good practice to store all of your git repos in the same top level directory. This keeps all of your repositories organized and easy to access. Create a directory named `workspace` in your home directory.
2. **Creating a repository (repo):** Next navigate create a new folder inside of your workspace directory and run `git init`. This [command](#) creates a new Git repository and adds a few Git [metadata files](#).
3. **Create a README:** Create a `README.md` file in your repository. You can do this with the command `echo "# My Project" > README.md`.
4. **Staging changes:** Use `git add .` to stage all files in the entire directory, in the next step we will use `git commit -m "Initial commit"` to commit this [guide on how to use git](#).

Become a Git expert in 5 min.
with our free cheat-sheet

5. **Committing changes:** Commit your staged changes with `git commit -m "Your commit message"`. Commit messages should be meaningful, describing what changes were made and why.



main

Join 20,000+ developers at top companies

The best engineers use Graphite to simplify Git

Engineers at Vercel, Snowflake & The Browser Company are shipping faster and staying unblocked with Graphite.

Git started for free

Request a Demo

Working with remote repositories

- **Git clone:** To copy an existing Git repository, use `git clone <repository-url>`. This is often the first step when contributing to a project or starting a new feature based on an existing codebase. For further information, see [this guide on cloning a repository](#).
- **Git push:** After committing your changes, use `git push <remote-name>` to upload them to the remote repository.
- **Git pull:** To update your local repository with changes from the remote, use `git pull <remote-name> <branch-name>`.

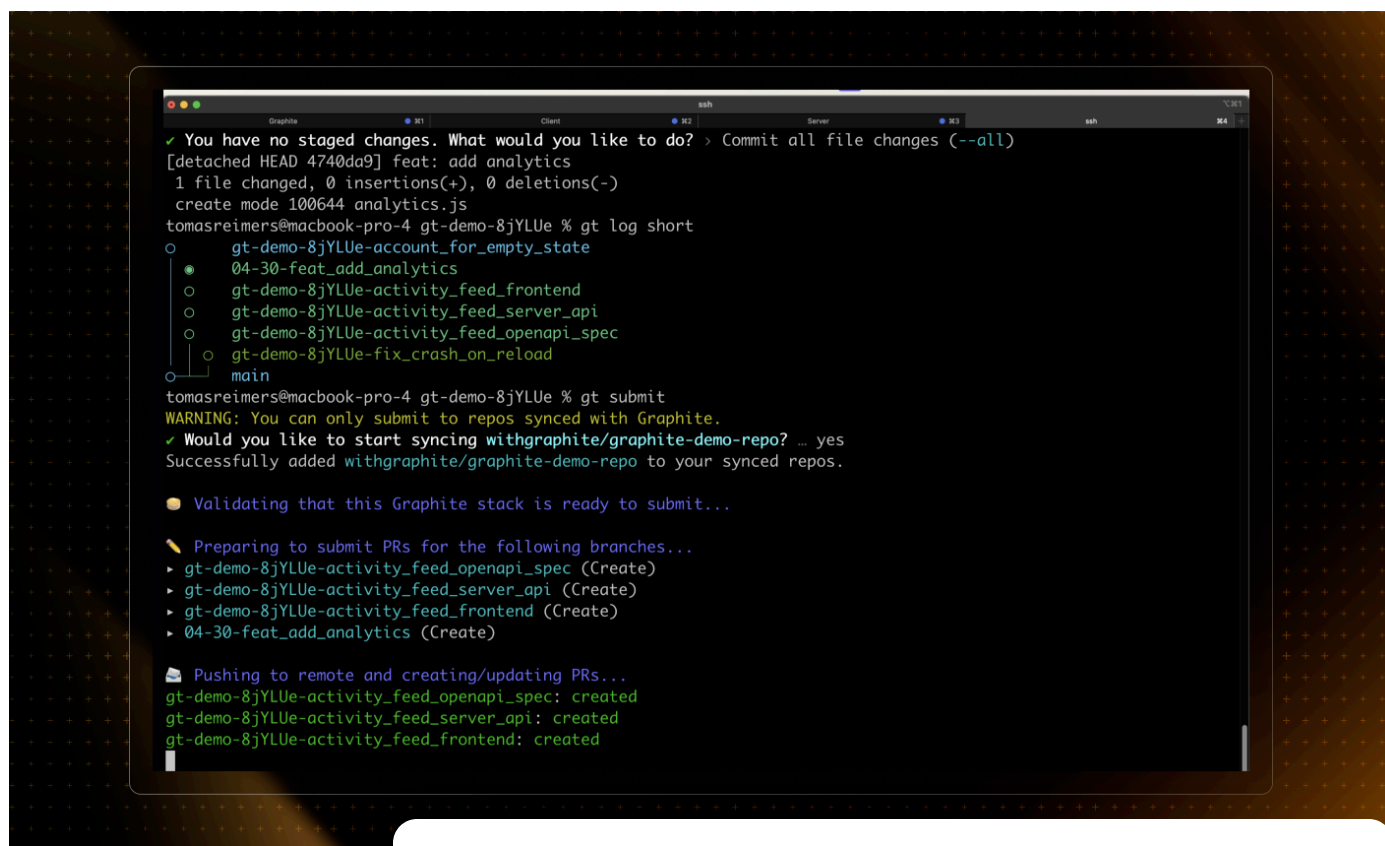
Become a Git expert in 5 min.
with our free cheat-sheet

Using the Graphite CLI in Git Bash on Windows

While Git is an incredibly useful tool, it has many shortcomings, particularly with rebasing, and managing [stacked pull requests](#).

Compatible with Git Bash for Windows, the Graphite CLI simplifies `git`, handles rebasing automatically, and allows you to create, submit, and stack pull requests right from the command line.

Under the hood, the CLI runs Git to create branches, commits, and metadata, which means you can still use Git in your scripts, tooling, or whenever you feel like it. **Read more about installing the Graphite CLI in our [docs](#).**

A screenshot of a terminal window with a dark background and light-colored text. The terminal shows the Graphite CLI interface. At the top, it says 'You have no staged changes. What would you like to do? > Commit all file changes (--all)'. Below that, it shows the commit message '[detached HEAD 4740da9] feat: add analytics' and the file changes '1 file changed, 0 insertions(+), 0 deletions(-)' and 'create mode 100644 analytics.js'. Then, it shows the command 'tomasreimers@macbook-pro-4 gt-demo-8jYLUe % gt log short' and the output of the log command, which lists several branches and the main branch. Next, it shows the command 'tomasreimers@macbook-pro-4 gt-demo-8jYLUe % gt submit' and the output, which includes a warning 'WARNING: You can only submit to repos synced with Graphite.', a confirmation prompt 'Would you like to start syncing withgraphite/graphite-demo-repo? ... yes', and a message 'Successfully added withgraphite/graphite-demo-repo to your synced repos.'. Finally, it shows the command 'Validating that this Graphite stack is ready to submit...' and the output, which includes a message 'Preparing to submit PRs for the following branches...' and a list of branches to be submitted: 'gt-demo-8jYLUe-activity_feed_openapi_spec (Create)', 'gt-demo-8jYLUe-activity_feed_server_api (Create)', 'gt-demo-8jYLUe-activity_feed_frontend (Create)', and '04-30-feat_add_analytics (Create)'. The output also shows 'Pushing to remote and creating/updating PRs...' and the status of the PRs: 'gt-demo-8jYLUe-activity_feed_openapi_spec: created', 'gt-demo-8jYLUe-activity_feed_server_api: created', and 'gt-demo-8jYLUe-activity_feed_frontend: created'.

Become a Git expert in 5 min.
with our free cheat-sheet

Git Bash Customiz

Git Bash can be customized with [custom prompts](#). For Windows pack included package managen

Troubleshooting and Further Learning

Git Bash documentation and help: For more detailed information and troubleshooting, refer to the [official Git For Windows documentation](#) or use `git help <command>` to get help for specific Git commands.

Related guides

Code review comment types

Graphite's guide to code review comments lays the foundation for a constructive, collaborative, and ...

Gerrit vs. Phabricator: Key Differences

Gerrit and Phabricator cater to different needs within the software development lifecycle. Gerrit is...

Sapling

Sapling: A Source Control System developed at Meta, designed to emphasize usability and scalability....

On this page

Downloading and installing Git Bash on Windows

Basic Git bash commands

[Git configuration](#)

Your first git repository

Working with remote repositories

Using the Graphite CLI

Git Bash Customization

Troubleshooting and Further Learning

Become a Git expert in 5 min.
with our free cheat-sheet

Git gud

"It's the first Git workflow I've used that actually feels good."

—@robboclancy 

Learn more →

Graphite Git stacked on GitHub

Stacked pull requests are essential for read, write, and delete access to manage.

Teams that stack shi

Become a Git expert in 5 min.
with our free cheat-sheet

Learn more

Get started for free

Or install our **CLI**.



Product

Features

Pricing

Docs

Customers

Company

Blog

Careers

Contact us

Resources

Community

Privacy policy

Terms of service

Stacking workflow

Developers

Status

GitHub

Become a Git expert in 5 min.
with our free cheat-sheet

