FEATURED ON Product Hunt ▲ 532



# 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.


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


Git Bash operates on a unix
would on a unix system.

- **Navigating directories**
- **Creating directories**: `mk`
- **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.

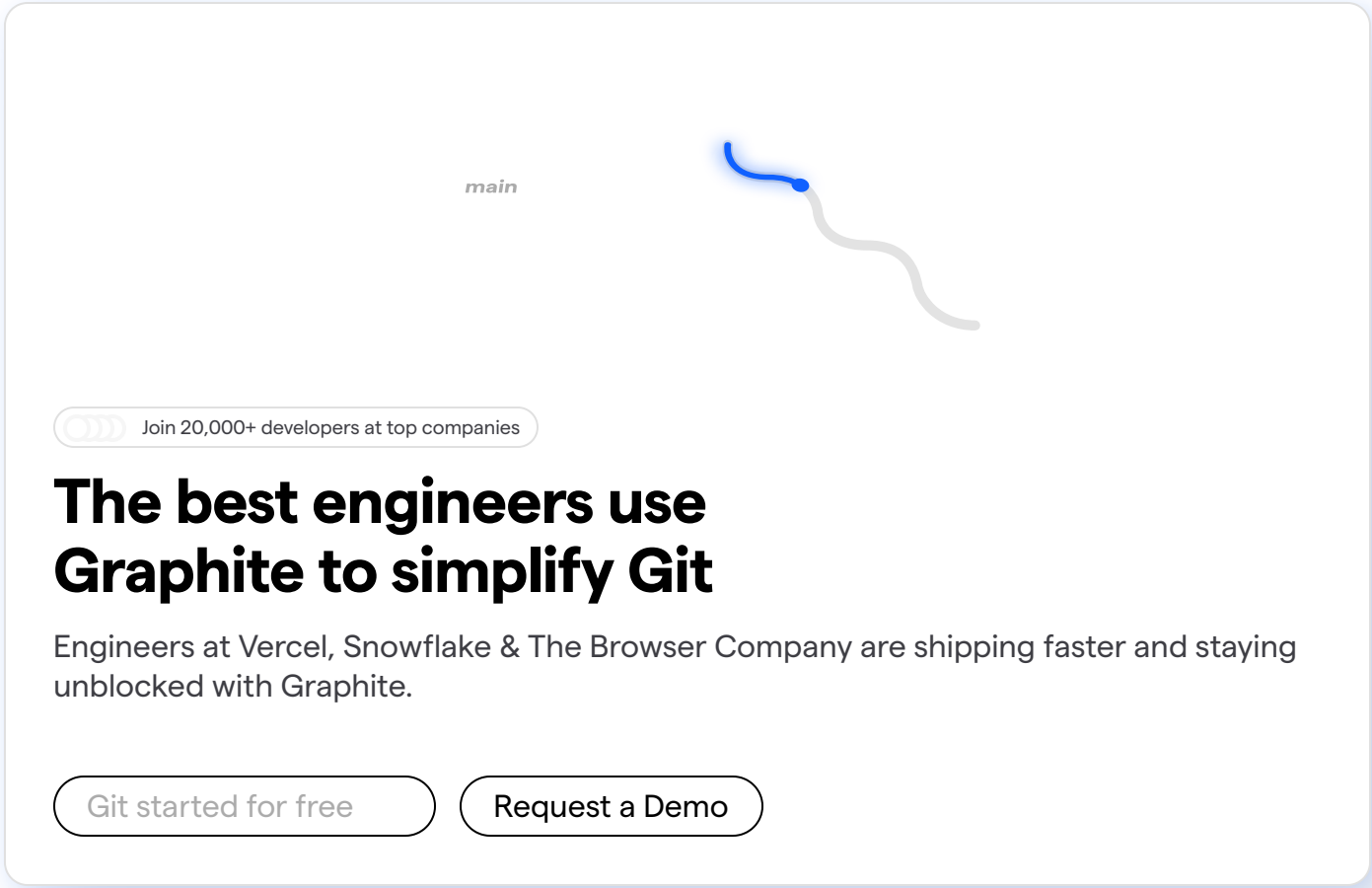
TIP: If you want to change either field temporarily, you can run this command later without the `--global` , which will set email and name in the scope of the current git project.

## 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:** Crea
   can do this with the com
4. **Staging changes**: Use `g`
   the entire directory, in th
   this guide on how to us

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.




## 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 inf
- **Git push**: After committ
`name>` to upload them t
- **Git pull**: To update your
`<branch-name>`.

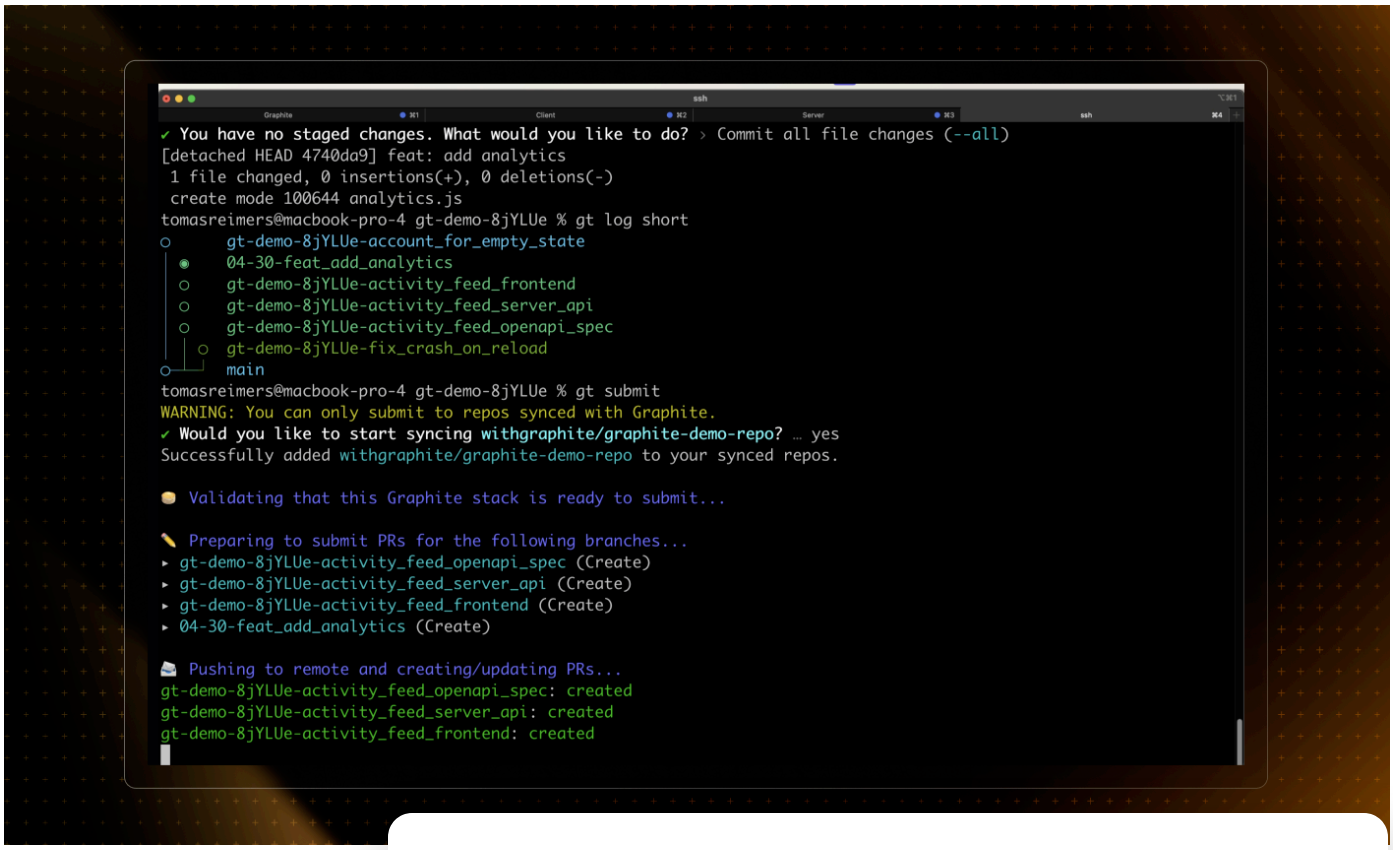
Become a Git expert in 5 min.
with our free cheat-sheet

# 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.**




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

## Git Bash Customiz

Git Bash can be customized
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**



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 easier to read, easier to write, and easier to manage.
Teams that stack ship

Learn more

Get started for free

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

Or install our **CLI**.

Graphite © 2024

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