Caching your GitHub credentials in Git

In this article

If you're cloning GitHub Enterprise Server repositories using HTTPS, we recommend you use GitHub CLI or Git Credential Manager (GCM) to remember your credentials.

Mac Windows Linux

Tip: If you clone GitHub Enterprise Server repositories using SSH, then you can authenticate using an SSH key instead of using other credentials. For information about setting up an SSH connection, see "Connecting to GitHub with SSH."

GitHub CLI

GitHub CLI will automatically store your Git credentials for you when you choose `HTTPS` as your preferred protocol for Git operations and answer "yes" to the prompt asking if you would like to authenticate to Git with your GitHub Enterprise Server credentials.

1. Install GitHub CLI on macOS, Windows, or Linux.
2. In the command line, enter `gh auth login`, then follow the prompts.
   - When prompted for your preferred protocol for Git operations, select `HTTPS`.
   - When asked if you would like to authenticate to Git with your GitHub Enterprise Server credentials, enter `Y`.

For more information about authenticating with GitHub CLI, see `gh auth login`.

Git Credential Manager

Git Credential Manager (GCM) is another way to store your credentials securely and connect to GitHub over HTTPS. With GCM, you don't have to manually create and store a personal access token, as GCM manages authentication on your behalf, including 2FA (two-factor authentication).

1. Install Git using Homebrew:

```
brew install git
```

2. Install GCM using Homebrew:

```
brew install --cask git-credential-manager
```

For MacOS, you don't need to run `git config` because GCM automatically configures Git for you.

The next time you clone an HTTPS URL that requires authentication, Git will prompt you to log in using a browser window. You may first be asked to authorize an OAuth app. If your account or organization requires two-factor auth, you'll also need to complete the 2FA challenge.

Once you've authenticated successfully, your credentials are stored in the macOS keychain and will be used every time you clone an HTTPS URL. Git will not require you to type your credentials in the command line again unless you change your credentials.

1. Install Git for Windows, which includes GCM. For more information, see "Git for Windows releases" from its releases page.

We recommend always installing the latest version. At a minimum, install version 2.29 or higher, which is the first version offering OAuth support for GitHub.

The next time you clone an HTTPS URL that requires authentication, Git will prompt you to log in using a browser window. You may first be asked to authorize an OAuth app. If your account or organization requires two-factor auth, you'll also need to complete the 2FA challenge.

Once you've authenticated successfully, your credentials are stored in the Windows credential manager and will be used every time you clone an HTTPS URL. Git will not require you to type your credentials in the command line again unless you change your credentials.

Warning: Older versions of Git for Windows came with Git Credential Manager for Windows. This older product is no longer supported and cannot connect to GitHub via OAuth. We recommend you upgrade to the latest version of Git for Windows.

Warning: If you cached incorrect or outdated credentials in Credential Manager for Windows, Git will fail to access GitHub Enterprise Server. To reset your cached credentials so that Git prompts you to enter your credentials, access the Credential Manager in the Windows Control Panel under User Accounts > Credential Manager. Look for the GitHub Enterprise Server entry and delete it.

For Linux, install Git and GCM, then configure Git to use GCM.

1. Install Git from your distro's packaging system. Instructions will vary depending on the flavor of Linux you run.
2. Install GCM. See the instructions in the GCM repo, as they'll vary depending on the flavor of Linux you run.
3. Configure Git to use GCM. There are several backing stores that you may choose from, so see the GCM docs to complete your setup. For more information, see "GCM Linux."

The next time you clone an HTTPS URL that requires authentication, Git will prompt you to log in using a browser window. You may first be asked to authorize an OAuth app. If your account or organization requires two-factor auth, you'll also need to complete the 2FA challenge.

Once you've authenticated successfully, your credentials are stored on your system and will be used every time you clone an HTTPS URL. Git will not require you to type your credentials in the command line again unless you change your credentials.

For more options for storing your credentials on Linux, see Credential Storage in Pro Git.

For more information or to report issues with GCM, see the official GCM docs at "Git Credential Manager."