About remote repositories

In this article

GitHub's collaborative approach to development depends on publishing commits from your local repository to GitHub Enterprise Cloud for other people to view, fetch, and update.

About remote repositories 🔗

A remote URL is Git's fancy way of saying "the place where your code is stored." That URL could be your repository on GitHub, or another user's fork, or even on a completely different server.

You can only push to two types of URL addresses:

- An HTTPS URL like `https://github.com/user/repo.git`
- An SSH URL, like `git@github.com:user/repo.git`

Git associates a remote URL with a name, and your default remote is usually called `origin`.

Creating remote repositories 🔗

You can use the `git remote add` command to match a remote URL with a name. For example, you'd type the following in the command line:

```
git remote add origin <REMOTE_URL>
```

This associates the name `origin` with the `REMOTE_URL`.

You can use the command `git remote set-url` to [change a remote's URL](#).

Choosing a URL for your remote repository 🔗

There are several ways to clone repositories available on GitHub.com.

When you view a repository while signed in to your account, the URLs you can use to clone the project onto your computer are available below the repository details.

For information on setting or changing your remote URL, see "Managing remote repositories."

Cloning with HTTPS URLs 🔗

The `https://` clone URLs are available on all repositories, regardless of visibility. `https://` clone URLs work even if you are behind a firewall or proxy.

When you `git clone`, `git fetch`, `git pull`, or `git push` to a remote repository using HTTPS URLs on the command line, Git will ask for your GitHub Enterprise Cloud username and password. When Git prompts you for your password, enter your personal access token. Alternatively, you can use a credential helper like Git Credential Manager. Password-based authentication for Git has been removed in favor of more secure authentication methods. For more information, see "Managing your personal access tokens."

If you are accessing an organization that uses SAML SSO and you are using a personal access token (classic), you must also authorize your personal access token to access the organization before you authenticate. For more information, see "About authentication with SAML single sign-on" and "Authorizing a personal access token for use with SAML single sign-on."

> **Tips**:
>
> - You can use a credential helper so Git will remember your GitHub credentials every time it talks to GitHub. For more information, see "Caching your GitHub credentials in Git."
> - To clone a repository without authenticating to GitHub Enterprise Cloud on the command line, you can use GitHub Desktop to clone instead. For more information, see "Cloning a repository from GitHub to GitHub Desktop."

If you'd rather use SSH but cannot connect over port 22, you might be able to use SSH over the HTTPS port. For more information, see "Using SSH over the HTTPS port."

Cloning with SSH URLs 🔗

SSH URLs provide access to a Git repository via SSH, a secure protocol. To use these URLs, you must generate an SSH keypair on your computer and add the **public** key to your account on GitHub.com. For more information, see "Connecting to GitHub with SSH."

When you `git clone`, `git fetch`, `git pull`, or `git push` to a remote repository using SSH URLs, you'll be prompted for a password and must provide your SSH key passphrase. For more information, see "Working with SSH key passphrases."

If you are accessing an organization that uses SAML single sign-on (SSO), you must authorize your SSH key to access the organization before you authenticate. For more information, see "About authentication with SAML single sign-on" and "Authorizing an SSH key for use with SAML single sign-on."

> **Tip**: You can use an SSH URL to clone a repository to your computer, or as a secure way of deploying your code to production servers. You can also use SSH agent forwarding with your deploy script to avoid managing keys on the server. For more information, see "Using SSH agent forwarding."

Cloning with GitHub CLI 🔗

You can also install GitHub CLI to use GitHub Enterprise Cloud workflows in your

terminal. For more information, see "About GitHub CLI."

Cloning with Subversion 🔗

> **Note**: Subversion support will be removed from GitHub on January 8, 2024. A future release of GitHub Enterprise Server after January 8, 2024 will also remove Subversion support. To read more about this, see the GitHub blog.

You can also use a Subversion client to access any repository on GitHub. Subversion offers a different feature set than Git. For more information, see "What are the differences between Subversion and Git?"

You can also access repositories on GitHub from Subversion clients. For more information, see "Support for Subversion clients."

Legal

Terms   Privacy   Status   Pricing   Expert services   Blog