

Author: Eric Gibbons

Subject: Using GitHub and SSH Keys

Date: September 3, 2025



How to Use GitHub SSH Keys: Making GitHub Authentication Easy

Using an SSH key is a secure and convenient way to interact with GitHub without having to enter your username and personal access token for every push or pull. This document outlines the procedure for setting up SSH keys on Unix-based systems (macOS and Linux) and on Windows.

Setting Up SSH Keys on Unix (macOS and Linux)

The process on macOS and Linux is identical, as both are Unix-based operating systems. You will use the terminal for all steps.

Step 1: Check for Existing SSH Keys

First, check if you already have an SSH key pair on your machine.

1. Open your terminal.
2. List the files in your `~/.ssh` directory by running:

```
ls -al ~/.ssh
```

3. Look for files named `id_ed25519.pub`, `id_rsa.pub`, or similar. If you see a file pair (e.g., `id_ed25519` and `id_ed25519.pub`), you already have a key and can skip to Step 3.

Step 2: Generate a New SSH Key

If you don't have a key, you can generate a new one. The recommended algorithm is Ed25519.

1. In the terminal, run the following command, replacing the email with your own GitHub email address.

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

2. When prompted to "Enter a file in which to save the key," you can press **Enter** to accept the default location (`~/.ssh/id_ed25519`).
3. You will be asked to enter a passphrase. This is an optional but highly recommended layer of security. Type a strong passphrase and press **Enter**. You will need to enter it again to confirm.

Step 3: Add Your SSH Key to the ssh-agent

The ssh-agent is a background program that handles your keys and passphrases.

1. Start the agent in the background:

```
eval "$(ssh-agent -s)"
```

2. Add your new private key to the agent. If you used a different name for your key, replace `id_ed25519` with your key's filename.

```
ssh-add ~/.ssh/id_ed25519
```

Step 4: Add the Public Key to Your GitHub Account

The final step is to tell GitHub about your public key.

1. Display the public key's content on the screen.

```
cat ~/.ssh/id_ed25519.pub
```

2. Select and copy the *entire output* to your clipboard. It starts with `ssh-ed25519` and ends with your email address.
3. Open GitHub in your browser and navigate to **Settings** by clicking your profile picture in the top-right corner.
4. In the user settings sidebar, click **SSH and GPG keys**.
5. Click **New SSH key** or **Add SSH key**.
6. In the "Title" field, add a descriptive label for the new key (e.g., "My MacBook Pro").
7. Paste your copied key into the "Key" field.
8. Click **Add SSH key**. You may be asked to confirm your GitHub password.

You can test your connection by running `ssh -T git@github.com`. You should see a welcome message from GitHub.

Setting Up SSH Keys on Windows

For Windows users, the easiest method is to use Git Bash, which is included with the standard installation of Git for Windows. The process is nearly identical to the Unix procedure.

Step 1: Open Git Bash and Check for Keys

1. Open the **Git Bash** application.
2. Check for existing keys using the same command as on Unix:

```
ls -al ~/.ssh
```

Step 2: Generate a New Key

If no key exists, generate one.

1. Use the same key generation command, replacing the email with your own:

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

2. Follow the prompts, accepting the default file location and setting a secure passphrase.

Step 3: Add the Key to the ssh-agent

1. Start the agent in the background:

```
eval "$(ssh-agent -s)"
```

2. Add your private key to the agent:

```
ssh-add ~/.ssh/id_ed25519
```

Step 4: Add the Public Key to GitHub

1. Copy the public key to your clipboard. You can display it with `cat` as on Linux, or use the `clip` utility for convenience.

```
clip < ~/.ssh/id_ed25519.pub
```

2. Follow the same steps as described in the Unix section to navigate to your GitHub SSH settings.
3. Create a **New SSH key**, give it a title, and paste the key from your clipboard into the "Key" text box.
4. Click **Add SSH key** to finish.

Your setup is complete. You can now clone, pull, and push to your GitHub repositories from Git Bash without needing to authenticate each time.