Generating SSH keys

MAC | WINDOWS | LINUX | ALL

SSH keys are a way to identify trusted computers, without involving passwords. The steps below will walk you through generating an SSH key and adding the public key to your GitHub account.

We recommend that you regularly review your SSH keys list and revoke any that haven't been used in a while.

Tip: If you have GitHub for Windows installed, you can use it to clone repositories and not deal with SSH keys. It also comes with the Git Bash tool, which is the preferred way of running `git` commands on Windows.

Step 1: Check for SSH keys

First, we need to check for existing SSH keys on your computer. Open Git Bash and enter:

```
$ ls -al ~/.ssh
# Lists the files in your .ssh directory, if they exist
```

Check the directory listing to see if you already have a public SSH key. By default, the filenames of the public keys are one of the following:

- *id_dsa.pub*
- *id_ecdsa.pub*
- *id_ed25519.pub*
- *id_rsa.pub*

If you see an existing public and private key pair listed (for example *id_rsa.pub* and *id_rsa*) that you would like to use to connect to GitHub, you can skip **Step 2** and go straight to **Step 3**.

Tip: If you receive an error that *~/.ssh* doesn't exist, don't worry! We'll create it in **Step 2**.

Step 2: Generate a new SSH key

1. With Git Bash still open, copy and paste the text below. Make sure you substitute in your GitHub email address.

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
# Creates a new ssh key, using the provided email as a label
Generating public/private rsa key pair.
```

2. We strongly suggest keeping the default settings as they are, so when you're prompted to "Enter a file in which to save the key", just press **Enter** to continue.

```
Enter file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]
```

Article versions

GitHub.com
GitHub Enterprise 2.3
GitHub Enterprise 2.2
GitHub Enterprise 2.1
GitHub Enterprise 2.0

3 You'll be asked to enter a passphrase.

```
Enter passphrase (empty for no passphrase): [Type a passphrase]
Enter same passphrase again: [Type passphrase again]
```

Tip: We strongly recommend a very good, secure passphrase. For more information, see "Working with SSH key passphrases".

4 After you enter a passphrase, you'll be given the fingerprint, or *id*, of your SSH key. It will look something like this:

```
Your identification has been saved in /Users/you/.ssh/id_rsa.
Your public key has been saved in /Users/you/.ssh/id_rsa.pub.
The key fingerprint is:
01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db your_email@example.com
```

Step 3: Add your key to the ssh-agent

To configure the ssh-agent program to use your SSH key:

If you have GitHub for Windows installed, you can use it to clone repositories and not deal with SSH keys. It also comes with the Git Bash tool, which is the preferred way of running `git` commands on Windows.

1 Ensure ssh-agent is enabled:

- **If you are using Git Bash**, turn on ssh-agent:

```
# start the ssh-agent in the background
$ ssh-agent -s
Agent pid 59566
```

- **If you are using another terminal prompt**, such as msysgit, turn on ssh-agent:

```
# start the ssh-agent in the background
$ eval $(ssh-agent -s)
Agent pid 59566
```

2 Add your SSH key to the ssh-agent:

```
$ ssh-add ~/.ssh/id_rsa
```

Tip: If you didn't generate a new SSH key in **Step 2**, and used an existing SSH key instead, you will need to replace *id_rsa* in the above command with the name of your existing private key file.

Step 4: Add your SSH key to your account

To configure your GitHub account to use your SSH key:

Copy the SSH key to your clipboard. If your key is named `id_dsa.pub`, `id_ecdsa.pub` or

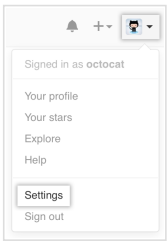
`id_ed25519.pub`, then change the filename below from *id_rsa.pub* to the one that matches your key:

```
$ clip < ~/.ssh/id_rsa.pub
# Copies the contents of the id_rsa.pub file to your clipboard
```

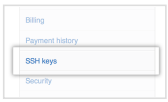
Warning: It's important to copy the key exactly without adding newlines or whitespace.

Add the copied key to GitHub:

1. In the top right corner of any page, click your profile photo, then click **Settings**.

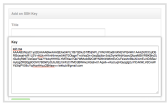


2. In the user settings sidebar, click **SSH keys**.

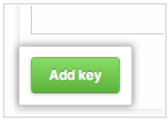


3. Click **Add SSH key**.

4. In the Title field, add a descriptive label for the new key. For example, if you're using a personal Mac, you might call this key "Personal MacBook Air".
5. Paste your key into the "Key" field.



6. Click **Add key**.



7. Confirm the action by entering your GitHub password.

Step 5: Test the connection

To make sure everything is working, you'll now try to SSH into GitHub. When you do this, you will be asked to authenticate this action using your password, which is the SSH key passphrase you created earlier.

1. Open Git Bash and enter:

```
$ ssh -T git@github.com
# Attempts to ssh to GitHub
```

2. You may see this warning:

```
The authenticity of host 'github.com (207.97.227.239)' can't be established.
RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)?
```

Verify the fingerprint in the message you see matches the following message, then type `yes`:

```
Hi username! You've successfully authenticated, but GitHub does not
```

```
provide shell access.
```

3 If the username in the message is yours, you've successfully set up your SSH key!

If you receive a message about "access denied," you can read these instructions for diagnosing the issue.

If you're switching from HTTPS to SSH, you'll now need to update your remote repository URLs. For more information, see Changing a remote's URL.

Contact a human