https://help.github.com/articles/generating-ssh-keys

**Generating SSH Keys**

* [mac](https://help.github.com/articles/generating-ssh-keys#platform-mac)  [windows](https://help.github.com/articles/generating-ssh-keys#platform-windows)  [linux](https://help.github.com/articles/generating-ssh-keys#platform-linux)  [all](https://help.github.com/articles/generating-ssh-keys#platform-all)

Skip this guide. Download our native app instead.

If you have decided not to use the [recommended HTTPS method](https://help.github.com/articles/set-up-git), we can use SSH keys to establish a secure connection between your computer and GitHub. The steps below will walk you through generating an SSH key and then adding the public key to your GitHub account.

**Step 1: Check for SSH keys**

First, we need to check for existing ssh keys on your computer. Open up Git Bash and run:

cd ~/.ssh

ls

# Lists the files in your .ssh directory

Check the directory listing to see if you have a file named either id\_rsa.pub or id\_dsa.pub. If you don't have either of those files go to **step 2**. Otherwise, you already have an existing keypair, and you can skip to **step 3**.

**Step 2: Generate a new SSH key**

To generate a new SSH key, enter the code below. We want the default settings so when asked to enter a file in which to save the key, just press enter.

ssh-keygen -t rsa -C "*your\_email@example.com*"

# Creates a new ssh key, using the provided email as a label

# Generating public/private rsa key pair.

# Enter file in which to save the key (/c/Users/*you*/.ssh/id\_rsa): *[Press enter]*

ssh-add id\_rsa

Now you need to enter a passphrase.

#### [Why do passphrases matter?](https://help.github.com/articles/generating-ssh-keys)

Passwords aren't very secure, you already know this. If you use one that's easy to remember, it's easier to guess or brute-force (try many options until one works). If you use one that's random it's hard to remember, and thus you're more inclined to write the password down. Both of these are Very Bad Things™. This is why you're using ssh keys.

But using a key without a passphrase is basically the same as writing down that random password in a file on your computer. Anyone who gains access to your drive has gained access to every system you use that key with. This is also a Very Bad Thing™. The solution is obvious: add a passphrase.

But I don't want to enter a long passphrase every time I use the key!

Neither do we! Thankfully, there's a nifty little tool called ssh-agent that can save your passphrase securely so you don't have to re-enter it. Depending on your OS, ssh-agent may be automatically run for you when you log in.

For more information about SSH key passphrases, check out our [help guide](https://help.github.com/articles/working-with-ssh-key-passphrases).

# Enter passphrase (empty for no passphrase): [Type a passphrase]

# Enter same passphrase again: [Type passphrase again]

Which should give you something like this:

# Your identification has been saved in /c/Users/*you*/.ssh/id\_rsa.

# Your public key has been saved in /c/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 SSH key to GitHub**

Run the following code to copy the key to your clipboard.

**Windows:**

$ clip < ~/.ssh/id\_rsa.pub

# Copies the contents of the id\_rsa.pub file to your clipboard

**Linux:**

$ sudo apt-get install xclip

# Downloads and installs xclip. If you don't have `apt-get`, you might need to use another installer (like `yum`)

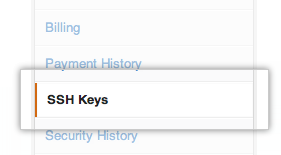
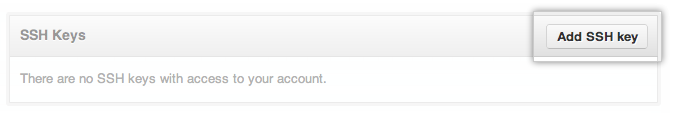
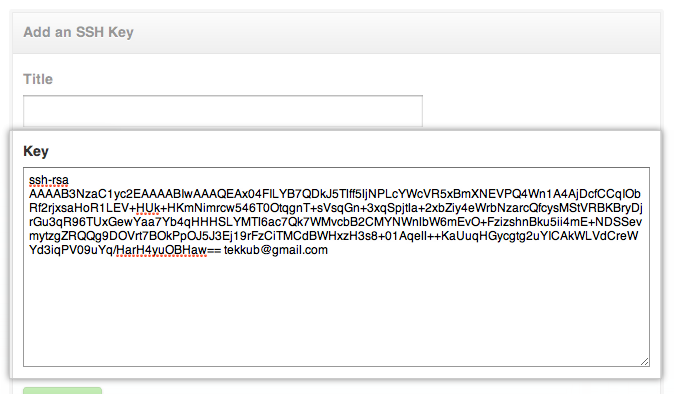
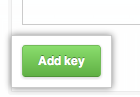
$ xclip -sel clip < ~/.ssh/id\_rsa.pub

# Copies the contents of the id\_rsa.pub file to your clipboard

**Mac:**

$ pbcopy < ~/.ssh/id\_rsa.pub

# Copies the contents of the id\_rsa.pub file to your clipboard

1. Account settings buttonGo to your [Account Settings](https://github.com/settings)
2. Click ["SSH Keys"](https://github.com/settings/ssh) in the left sidebar
3. Click "Add SSH key"
4. Paste your key into the "Key" field
5. Click "Add key"
6. Confirm the action by entering your GitHub password

**Step 4: Test everything out**

To make sure everything is working you'll now SSH to GitHub. When you do this, you will be asked to authenticate this action using your password, which for this purpose is the passphrase you created earlier. Don't change the git@github.com part. That's supposed to be there.

ssh -T git@github.com

# Attempts to ssh to github

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)?

Don't worry, this is supposed to happen. Verify that the fingerprint matches the one here and type "yes".

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

# provide shell access.

If that username is correct, you've successfully set up your SSH key. Don't worry about the shell access thing, you don't want that anyway.

If you see "access denied" please consider using [HTTPS](https://help.github.com/articles/set-up-git) instead of SSH. If you need SSH start at [these instructions](https://help.github.com/articles/error-permission-denied-publickey) for diagnosing the issue.