

# SSH Keys for GitHub

## Objectives

- Explain what an SSH key is
- Generate your own SSH key pair
- Add your SSH key to your GitHub account
- Learn how to use your SSH key in your GitHub workflow

## Why Use an SSH Key?

When working with a GitHub repository, you'll often need to identify yourself to GitHub using your username and password. An SSH key is an alternate way to identify yourself that doesn't require you to enter your username and password every time.

SSH keys come in pairs, a public key that gets shared with services like GitHub, and a private key that is stored only on your computer. If the keys match, you're granted access.

The cryptography behind SSH keys ensures that no one can reverse engineer your private key from the public one.

## Generating an SSH key pair

The first step in using SSH authorization with GitHub is to generate your own key pair.

You might already have an SSH key pair on your machine. You can check to see if one exists by moving to your `.ssh` directory and listing the contents.

```
$ cd ~/.ssh
$ ls
```

If you see `id_rsa.pub`, you already have a key pair and don't need to create a new one.

If you don't see `id_rsa.pub`, use the following command to generate a new key pair. Make sure to replace `your@email.com` with your own email address.

```
$ ssh-keygen -t rsa -C "your@email.com"
```

When asked where to save the new key, hit enter to accept the default location.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/username/.ssh/id_rsa):
```

You will then be asked to provide an optional passphrase. This can be used to make your key even more secure, but for this lesson you can skip it by hitting enter twice.

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

When the key generation is complete, you should see the following confirmation:

```
Your identification has been saved in /Users/username/.ssh/id_rsa.
Your public key has been saved in /Users/username/.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.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|      . E +
|      . o = .
|      . S = o
|      o.o . o
|      o .+ .
|      . o+..
|      .+=o
+-----+
|
```

The random art image is an alternate way to match keys but we won't be needing this.

## Add your public key to GitHub

We now need to tell GitHub about your public key. Display the contents of your new public key file with `cat`:

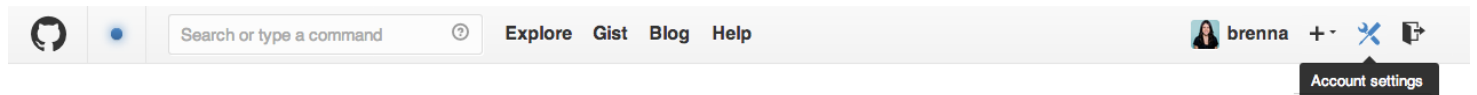
```
$ cat ~/.ssh/id_rsa.pub
```

The output should look something like this:


```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA879BjG1PTLIuc9/R5MYiN4yc/YiClcdBpSdzgK9Dt0Bkfe3rSz5cPm4wmehdE7GkVFXrBJ2YHqPLuM1yx1AUxIebpwLI19f/aUHOts9eVnVh4NztPy0iSU/Sv0b20DQ
Qvcy2vYujlorscl8JjAgfWs03W4iGEe6QwBpVomcME8IU35v5VbylM90RQa6wvZMvrPECBvwItTY8cPWH3MGZiK/74eHbSLKA4PY3gM4GHI450Nie16yggEg2aTQfWA1rry9JYWEoHS9pJ1dnLqZU3k/80WgqJrilwS
oC5rGjgp93iu0H8T6+mEHGRQe84Nk1y5LESSWIbn6P636B13uQ== your@email.com
```

Copy the contents of the output to your clipboard.

Login to [github.com](https://github.com) and bring up your account settings by clicking the tools icon.



Select **SSH Keys** from the side menu, then click the **Add SSH key** button.

 **brenna**

[Profile](#)

[Account settings](#)

[Emails](#)

[Notification center](#)

[Billing](#)

[Payment history](#)

**SSH keys**

[Security](#)

[Applications](#)


[Repositories](#)

[Organizations](#)

Need help? Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#)


SSH Keys Add SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.




**info@brennaobrien.com**  
a0:56:bc:a0:ef:06:39:d6:03:cc:9a:7a:0e:d8:92:5c  
Added 2 years ago — Last used on April 28, 2014

Delete



**test server**  
4a:dd:4b:0a:c5:20:55:8c:1a:3c:ac:14:c5:b7:63:04  
Added a year ago — Last used on March 20, 2014

Delete



**hackeryou SSH**  
93:89:a8:46:ef:46:1b:99:7a:fe:66:c0:ba:28:1c:c9  
Added 6 months ago — Last used on April 07, 2014

Delete

Name your key something whatever you like, and paste the contents of your clipboard into the **Key** text box.

Add an SSH Key

Title

home laptop

Key

ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQCsYZ8xOg7TbBI7b1GucA/hjiV3wtNGshmq0Rcy6oLFNzwyP6mdUPAit  
DrniBk4HtCWakC3OfMkwfwGoGilluzVaUE6x9wFkWJ/XnYFXgz13R2gqeP3z9D8GFidOfyC3sT8IMQY2TErA+ptC8ou  
1uMKadhAD9e950ypl1e5YpL6icalhwEWMm2vdoQW64WAI1sle3so09eYO1NYZPD8QrxW9MeMX07L/TSFVzglvw9j  
B9Gs4qkhgWNArUrAnzloCktwTT042a/rSnMi4i4Yir6z1K9ZfpPvrmNtY8yuMZyPx6t6UUXbPzCyTX+EpsmlLUZIZgQf8  
4inqpam79JovjA/3 your@email.com

Add key

Finally, hit **Add key** to save. Enter your github password if prompted.

### Using Your SSH Key


Going forward, you can use the SSH clone URL when copying a repo to your local machine.

#### SSH clone URL

git@github.com:swca



You can clone with [HTTPS](#), [SSH](#),  
or [Subversion](#). ⓘ

 Clone in Desktop

 Download ZIP

This will allow you to bypass entering your username and password for future GitHub commands.

### Key Points

- SSH is a secure alternative to username/password authorization
- SSH keys are generated in public / private pairs. Your public key can be shared with others. The private keys stays on your machine only.
- You can authorize with GitHub through SSH by sharing your public key with GitHub.