Generating SSH keys (Mac OS X)

This will step you through the process of generating a SSH keypair on Mac OS X.

Begin by opening your Terminal, generally found in the "Utilities" subdirectory of your "Applications" directory.

Generating a keypair

Before you generate your keypair, come up with a passphrase. The rules for good passwords also apply here: mix of upper and lower case, numbers, spaces and punctuation. Limit it to less than 31 characters.

Now, generate your keypair! Enter the following:

```
ssh-keygen -t rsa -C "yourname@yourdomain.ext"
```

Note: Do not type the dollar sign above; it is an example of the default command prompt shown by Mac OS X. Your actual prompt may be different. In the example above and below, the actual part you should type is the part that follows the dollar sign.

Your terminal should respond:

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

Press Return to accept the default value. Your terminal should respond:

```
Enter passphrase (empty for no passphrase):
Enter the passphrase you decided on above. The response will be:
Enter same passphrase again:
```

Enter the passphrase again and press Return. The program will think a bit, and respond with something like this.

Note that many of the details in the example below are just for example purposes; much of the actual output you see will differ from the below.

```
Your identification has been saved in id_rsa.
Your public key has been saved in id_rsa.pub.
The key fingerprint is:
3c:fb:bf:4b:71:13:dd:d5:36:0d:94:6a:c7:23:97:75 #yourusername#@#yourmacname.local
```

How do I copy my public key into my Mac's clipboard?

You can use the pbcopy utility to easily insert your public key (or other text files) into your Mac's clipboard so that you can add it to your Drupal.org profile, GitHub, or other places. The filename should be yourfilename.pub - with yourfilename being the filename you entered when you first created this file. If you just hit enter, the default is id_rsa.pub.

```
$ pbcopy < ~/.ssh/id_rsa.pub
```

You won't see any output in the terminal, but the contents of your public key will now be in your clipboard and can be easily pasted anywhere where you can normally paste text.
In case you're curious, the pbpaste utility works the other way, allowing you to easily grab the contents of the clipboard for use in the terminal. For example, the following command will write the contents of the clipboard to a file:

```
$ pbpaste > ~/clipboard.text
```

SSH Agent

When generating an SSH key, you'll need to add your newly created (or existing) SSH key to the ssh-agent.

Tip: If you used an existing SSH key rather than generating a new SSH key, you'll need to replace id_rsa in the above command with the name of your existing private key file.
1
Ensure ssh-agent is enabled:
start the ssh-agent in the background

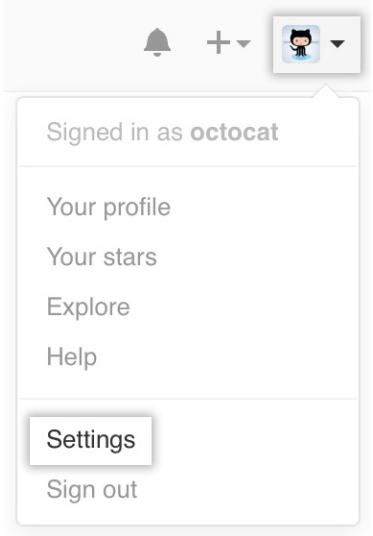
```
$ eval "$(ssh-agent -s)"
Agent pid 59566
```

2
Add your SSH key to the ssh-agent:

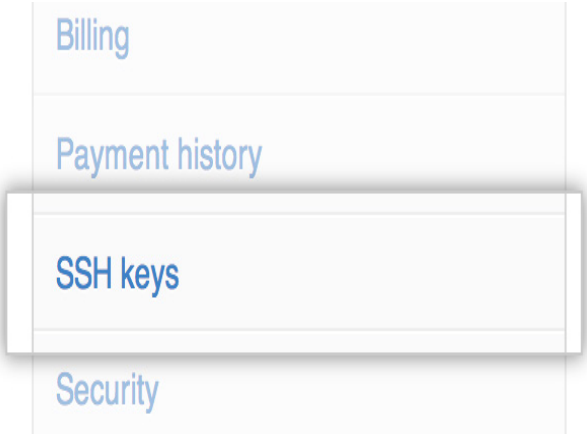
```
$ ssh-add ~/.ssh/id_rsa
```

To configure your GitHub account to use your new (or existing) SSH key, you'll also need to add it to your GitHub account.

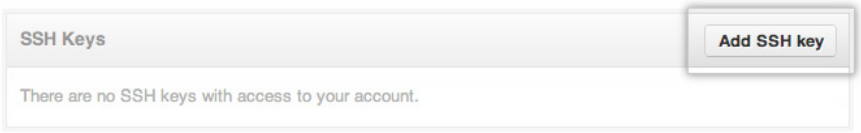
In the top right corner of any page, click your profile photo, then click Settings.



In the user settings sidebar, click SSH keys.

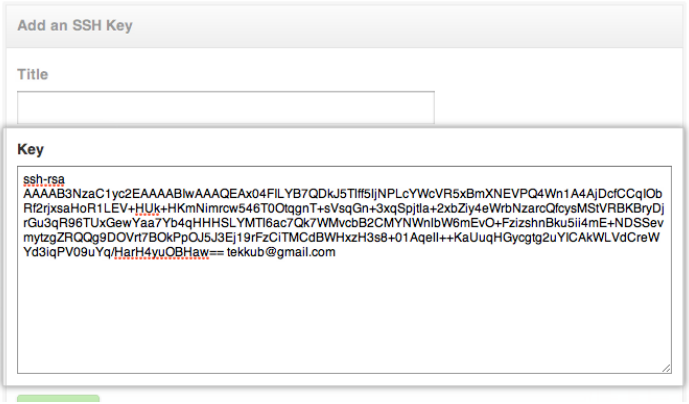


Click Add SSH key.



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".

Paste your key into the “Key” field.



Click Add key.

Confirm the action by entering your GitHub password.