

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

[Sign up](#)

Join the Stack Overflow community to:



Ask
programming
questions



Answer and
help your peers



Get recognized for your
expertise

Multiple heroku accounts

I'm having trouble when working with multiple Heroku accounts on my computer.

The thing is that it's worked great before by switching accounts with the heroku gem. But now (even though I've generated new SSH keys) it won't work.

When I do a `git push heroku master` it just says:

```
my@email.com not authorized to access my_app_name
```

Does anyone have any advice on how I could solve this?

[git](#) [heroku](#)

edited Nov 14 '14 at 11:46



[Jawa](#)
2,080 ● 5 ● 23 ● 32

asked Jan 11 '11 at 21:58



[Erik](#)
426 ● 1 ● 5 ● 3

what does `ssh yourid@server -v` give? – [Adam Dymitruk](#) Jan 11 '11 at 21:59

"Permission denied (publickey)."? – [Erik](#) Jan 11 '11 at 22:03

Oh, one note... The e-mail "my@email.com" is not the same e-mail as the Heroku account e-mail... so there seems to be some kind of mix up that's causing the problem. – [Erik](#) Jan 11 '11 at 22:05

copy both private and public keys under your `git/.ssh/` directory. that works for me! – [Dmitri Algazin](#) Feb 20 '14 at 0:01

12 Answers

You need to ensure that you are using the correct ssh key.

The way to do this (and the way the heroku accounts plugin prompts you) is to add a section to your ssh config file in `~/.ssh/config`. For instance, for my work heroku account I have this in my ssh config:

```
Host heroku.work
  HostName heroku.com
  IdentityFile ~/.ssh/id_heroku_work_rsa
  IdentitiesOnly yes
```

Now, and this is crucial, you need to make sure that your git remote is set up to use **that same named host**. In this case it is **heroku.work**. Normally it would be heroku.com if you were using heroku with only a single account.

So you'll need to edit the git remote (you can do this in the `.git/config` file of your repo on your machine). Change the file to look like:

```
[remote "heroku"]
  url = git@heroku.work:<appname>.git
```

Note the heroku.**work**, not heroku.**com**, and replace `<appname>` with the name of your app (aka your repo name) on heroku.

edited Nov 10 '15 at 23:38




[Tony](#)
6,607 ● 1 ● 20 ● 47

answered Jan 13 '12 at 19:40



[bantic](#)
2,982 ● 2 ● 17 ● 30

-
- 4 yeah! editing .git/config did the trick for me - [rodowi](#) Jun 8 '12 at 21:07
-
- 1 same here! works well :) - [jones](#) Aug 2 '12 at 19:32
-
- 4 thank you thank you thank you thank you - [Tom Rose](#) Aug 20 '12 at 22:24
-
- 1 Thx, that was what I needed when I created my prod app from heroku's dashboard. - [le_Daf](#) May 20 '13 at 11:54
-
- 1 git@heroku.work... doesn't exist as an endpoint, though. How do I make that the endpoint to push to? -- Ah, you need the heroku-accounts plugin. - [jmstone](#) Aug 20 '13 at 18:07 
-

I solved this by using the heroku-accounts plugin

see here for instructions

<http://martyhaught.com/articles/2010/12/14/managing-multiple-heroku-accounts/>

answered Mar 1 '11 at 16:30

 [Louis Gillies](#)
376 ● 2 ● 3

The biggest problem with this tool is that it generates unencrypted SSH keys. - [Wolfram Arnold](#)
Nov 7 '12 at 2:54

create the keys yourself manually and that problem goes away. this plug is perfect for this! - [chrishough](#) Jun 6 '13 at 9:37


You've got to be accepted as a collaborator on the project you don't own.

It works like this (using git) :

1. The owner of the project goes to heroku's console for the project
2. The owner adds the email address of the collaborator (the one that was used to register the collaborator's)
3. The collaborator sets up his public security key in his account (if he didn't already)
4. The collaborator adds and sets up the other project as another remote

And that should be all folks ; you just have to push on the other remote when deploying on the other app.

answered May 10 '11 at 9:40

 [Bastes](#)
373 ● 3 ● 13

Thank you!!! I spent hours trying to figure this out. Even trying all the different approaches on SO and other sites. Adding the same email to both projects on Heroku is definitely the easiest solution. - [B Seven](#) Jul 18 '12 at 4:41

I spent ages setting up an identify file and doing all sorts of tricks to make multiple keys work but this answer is by far the easiest way to make this work. Add your original account as a collaborator and use the same key. Easy. - [Peter Nixey](#) Nov 29 '13 at 12:33

I'd add a couple of things re the [heroku-accounts plugin](#)... this example assumes your pre-existing heroku account will be heroku.work and you are adding a new account heroku.personal.

It also describes how to move an app from one to the other, for example, if you have hit your 25-app max and are creating a new account and want to move some of your side projects and experimental apps out of your main account.

1. You probably need to use the plugin to create BOTH your old and your new accounts, eg, if you had one to start with you can't just add a new heroku.personal, you need to create heroku.work to replace your old one
2. You'll need to reassign your old apps to your heroku.work
3. if you are *moving* an app, don't assign the app to heroku.personal (using `heroku accounts:set personal`) until *after* you transfer the app using the heroku `sharing` command as described
<https://devcenter.heroku.com/articles/transferring-apps>

So the steps you probably need are:

1. Install the heroku-account plugin
2. Go to heroku.com and create your new account (if you don't have a second account already)
3. run `heroku accounts:add work --auto` to create a new heroku.work account to replace your old default account (so use the same credentials you've been using all along). You ALSO need to upload your new key to heroku: `heroku keys:add ~/.ssh/identity.heroku.work.pub`
4. run `heroku accounts:add personal --auto` to create a new heroku.personal account (eg use your new credentials for your new heroku account). You ALSO need to upload your new key to heroku: `heroku keys:add ~/.ssh/identity.heroku.personal.pub`
5. For each app, cd to the project root then run `heroku accounts:set work` (all that does it setup git to use your heroku.work, which is (unin this example) the same heroku account you've been using all along, only now it's called heroku.work.
6. If you want to move an app from heroku.work to heroku.personal, (a) add the new heroku account as a collaborator with `heroku sharing:add <new owner email>` (b) in your local directory, switch the app to heroku.personal using `heroku accounts:set personal` (c) transfer** the app from the old account to the new account using `heroku sharing:transfer <new owner email>` then (d) remove the old email address collaborator (if you want to) using `heroku sharing:remove <old owner email>`

** Note the heroku link I provided says there are special requirements to move an app with any paid resources. Generally, remove the paid resoures, then add them back later, or have support do it.

edited Nov 7 '12 at 2:16



Wolfram Arnold
4,615 • 31 • 57

answered Oct 16 '12 at 2:01



jpwynn
7,110 • 14 • 70 • 137

I started using the heroku-accounts plugin but then realized my buddy could just make me a collaborator on his heroku account and that was all that was needed.

answered Nov 25 '12 at 2:12



linojon
749 • 5 • 16

upvoted this one, especially if you have the password for both accounts – [portforwardpodcast](#)
Feb 11 '14 at 10:46

The Heroku plugin <https://github.com/ddollar/heroku-accounts> has been deprecated. Heroku now maintains its own at <https://github.com/heroku/heroku-accounts>. Unfortunately it doesn't seem to store project-by-project which account it should be using so you need to run

```
$ heroku accounts:set account_name
```

each time you want to use the Heroku command-line tool.

Quick solution for people with access to the bash shell: make a local bin directory (it's already there in Rails applications) and create bin/heroku there. Contents:

```
#!/bin/bash
/usr/bin/heroku accounts:set account_name
/usr/bin/heroku $*
```

(replace "project_name" as appropriate) Then run

```
$ chmod +x bin/heroku
```

You can then run

```
$ bin/heroku run console
```

and it will automatically use the right Heroku account for you. Better still is to add `./bin` to your path so you're back to just running:

```
$ heroku run console
```

answered Sep 24 '15 at 15:58



James Roscoe
535 • 3 • 10

To get this to work for me with both Git and the Heroku console, I had to add create the ssh config as instructed by bantic, but I also had to create two remotes to my .git config:

```
[remote "origin"]
  url = git@heroku.work:pacific-rock-4904.git
  fetch = +refs/heads/*:refs/remotes/heroku/*
[remote "heroku"]
  url = git@heroku.com:pacific-rock-4904.git
  fetch = +refs/heads/*:refs/remotes/heroku/*
```

answered May 26 '13 at 14:09



jamespaden

131 ● 4

And then push to what? heroku or origin? – [jmstone](#) Aug 20 '13 at 19:04

This has been bugging me for a while, and I never found a solution that I liked. Finally found one.

(OS X specific answer.)

You can just create new users. Go to System Preferences -> Users & Groups -> Add user or group button. You can create a few users for different heroku accounts, I just named mine user1, user2, etc.

Then go to System Preferences -> Sharing -> Remote Login -> Allow access for: All Users

At this point, you can go to a terminal and do the following:

```
$ ssh user1@localhost
```

Then you can do things like this:

```
$ heroku login
$ heroku keys:add
```

answered Oct 1 '14 at 21:29



funroll

13.4k ● 5 ● 33 ● 37

This seems perfect, but is it possible to give different ssh users access to the same folders and files? – [Numbers](#) Oct 10 '14 at 15:43

I have the exact same problem.

This is not the answer, but a temp solution is to add your "standard" heroku email as a collaborator to the app you can't access.

answered Jan 13 '11 at 19:35



kbjerring

415 ● 4 ● 16

Sorry I forgot to follow up on this, but restarting my computer solved the problem. May not solve yours. – [Erik](#) Jan 17 '11 at 17:15

Yeah, I read about that some place too. It didn't work for me though. – [kbjerring](#) Jan 17 '11 at 21:42

i am using **win7**

1) create another user account in win7 and log into it

2) open git bash and log into your new heroku account

3) create and upload new RSA key (instruction here : <https://devcenter.heroku.com/articles/keys>)

4) then do what ever you want (create new app, push file to it)

5) every time you need to work with your original heroku account just log into your original win user account and work with heroku

answered Jan 23 '13 at 9:24



mehdok

I wouldn't recommend this - having to switch local PC accounts is overkill when there are better solutions below. - [Rafi Jacoby](#) Feb 13 '13 at 16:51

this is an option anyway - [mehdok](#) Feb 22 '13 at 14:39

create new public/private key for new email as follows

- 1) ssh-keygen -t rsa -C "mvjohn@klairvoynat.com"
- 2) provide new file name in which to save the key (/home/.ssh/seckey)
- 3) Then add your new key to the ssh-agent:- ssh-add ~/.ssh/seckey
- 4) Then add your new key to heroku :- heroku keys:add ~/.ssh/seckey
- 5) add SSH Config File as ~/.ssh/config and content as follows

```
Host heroku.com Hostname heroku.com Port 22 IdentitiesOnly yes IdentityFile  
~/.ssh/seckey User mvjohn@klairvoynat.com
```

```
Host heroku.com Hostname heroku.com Port 22 IdentitiesOnly yes IdentityFile  
~/.ssh/id_rsa User mvjohn@hotmail.com
```

6) clone the code using either

a) git clone git@heroku.com:your-application.git b) heroku git:clone -a your-application

answered May 26 '14 at 17:44



[John](#)
101 ● 3

Edit your config for that project, setting the correct email. From the root of that repository, `git config user.email <heroku email>`

answered Jan 11 '11 at 22:13



[catsby](#)
8,546 ● 2 ● 22 ● 29

Seems to have no effect I'm afraid... - [Erik](#) Jan 11 '11 at 22:18

What about when you add the "--global" flag to that command? Sometimes Heroku gem was reading that instead of the project's repo email for me. - [danneu](#) Jan 13 '11 at 19:55
