[How to tell git which private key to use?](https://superuser.com/questions/232373/how-to-tell-git-which-private-key-to-use)

[Ask Question](https://superuser.com/questions/ask)

582

236

ssh has the -i option to tell which private key file to use when authenticating:

-i *identity\_file*

Selects a file from which the identity (private key) for RSA or DSA authentication is read.  The default is ~/.ssh/identity for protocol version 1, and ~/.ssh/id\_rsa and ~/.ssh/id\_dsafor protocol version 2.  Identity files may also be specified on a per-host basis in the configuration file.  It is possible to have multiple **-i** options (and multiple identities specified in configuration files).

Is there a similar way to tell git which private key file to use on a system with multiple private keys in the ~/.ssh directory?

[ssh](https://superuser.com/questions/tagged/ssh) [git](https://superuser.com/questions/tagged/git) [authentication](https://superuser.com/questions/tagged/authentication) [private-key](https://superuser.com/questions/tagged/private-key)

[share](https://superuser.com/q/232373)[improve this question](https://superuser.com/posts/232373/edit)

[edited Jun 16 '15 at 5:31](https://superuser.com/posts/232373/revisions)

[G-Man](https://superuser.com/users/354511/g-man)

**6,142**112563

asked Jan 12 '11 at 18:20

[[https://www.gravatar.com/avatar/2d085a853e59e7fe49339202c4952331?s=32&d=identicon&r=PG](https://superuser.com/users/53622/jrdioko)](https://superuser.com/users/53622/jrdioko)

[jrdioko](https://superuser.com/users/53622/jrdioko)

**3,550**52126

* 3

See [this question in StackOverflow](http://stackoverflow.com/q/4565700/247696) as well. – [Flimm](https://superuser.com/users/90668/flimm" \o "4,972 reputation) May 8 '15 at 9:46

* 2

Also related [serverfault.com/questions/194567/…](http://serverfault.com/questions/194567/how-do-i-tell-git-for-windows-where-to-find-my-private-rsa-key) – [Machavity](https://superuser.com/users/450209/machavity" \o "362 reputation) Jun 28 '16 at 14:51

add a comment

17 Answers

[active](https://superuser.com/questions/232373/how-to-tell-git-which-private-key-to-use?answertab=active#tab-top)[oldest](https://superuser.com/questions/232373/how-to-tell-git-which-private-key-to-use?answertab=oldest#tab-top)[votes](https://superuser.com/questions/232373/how-to-tell-git-which-private-key-to-use?answertab=votes#tab-top)

623

In ~/.ssh/config, add:

host github.com

HostName github.com

IdentityFile ~/.ssh/id\_rsa\_github

User git

Now you can do git clone git@github.com:username/repo.git.

NOTE: Verify that the permissions on IdentityFile are 400.SSH will reject, in a not clearly explicit manner, SSH keys that are too readable. It will just look like a credential rejection. The solution, in this case, is:

chmod 400 ~/.ssh/id\_rsa\_github

[share](https://superuser.com/a/232406)[improve this answer](https://superuser.com/posts/232406/edit)

[edited Oct 7 '15 at 8:31](https://superuser.com/posts/232406/revisions)

[[https://www.gravatar.com/avatar/c347f0b903a10ea49a5fce1e443209b5?s=32&d=identicon&r=PG](https://superuser.com/users/20671/silviot)](https://superuser.com/users/20671/silviot)

[silviot](https://superuser.com/users/20671/silviot)

**293**49

answered Jan 12 '11 at 19:36

[[https://www.gravatar.com/avatar/ad491004f55ce64ba22e66be2dbf5ac0?s=32&d=identicon&r=PG](https://superuser.com/users/56383/shellholic)](https://superuser.com/users/56383/shellholic)

[shellholic](https://superuser.com/users/56383/shellholic)

**7,107**1128

* 95

What if you need to connect to the same host with different keys? – [Valentin Klinghammer](https://superuser.com/users/72722/valentin-klinghammer) Nov 30 '12 at 11:24

* 6

@Quelltextfabrik - you can add another section with a different Host: [nerderati.com/2011/03/…](http://nerderati.com/2011/03/simplify-your-life-with-an-ssh-config-file/) – [Ben Challenor](https://superuser.com/users/84756/ben-challenor) Dec 4 '12 at 14:17

* 1

@Cliff Nop, in my manpage: "HostName: Specifies the real host name to log into. This can be used to specify nicknames or abbreviations for hosts." My ssh version is openssh-6.7p1. – [Grissiom](https://superuser.com/users/33849/grissiom" \o "111 reputation) Jan 7 '15 at 2:17

* 1

@Grissiom That's exactly what it says. But you seem to understand the meaning backwards. Host (or Match) is required. To create a host nickname you place the nickname in the Host line and the real hostname in the HostName line. Examples: [saltycrane.com/blog/2008/11/…](http://www.saltycrane.com/blog/2008/11/creating-remote-server-nicknames-sshconfig/) – [Cliff](https://superuser.com/users/64851/cliff) Jan 8 '15 at 3:55

* 10

If the config file is new, don't forget to do chmod 600 ~/.ssh/config – [elysch](https://superuser.com/users/333392/elysch" \o "108 reputation) Mar 15 '16 at 23:02

[show **9** more comments](https://superuser.com/questions/232373/how-to-tell-git-which-private-key-to-use/912281)

293

Environment variable GIT\_SSH\_COMMAND:

From Git version 2.3.0, you can use the environment variable GIT\_SSH\_COMMAND like this:

GIT\_SSH\_COMMAND="ssh -i ~/.ssh/id\_rsa\_example" git clone example

Note that -i can sometimes be overridden by your config file, in which case, you should give SSH an empty config file, like this:

GIT\_SSH\_COMMAND="ssh -i ~/.ssh/id\_rsa\_example -F /dev/null" git clone example

Configuration core.sshCommand:

From Git version 2.10.0, you can configure this per repo or globally, so you don't have to set the environment variable any more!

git config core.sshCommand "ssh -i ~/.ssh/id\_rsa\_example -F /dev/null"

git pull

git push

[share](https://superuser.com/a/912281)[improve this answer](https://superuser.com/posts/912281/edit)

[edited Oct 28 '16 at 11:30](https://superuser.com/posts/912281/revisions)

answered May 8 '15 at 9:43

[[https://www.gravatar.com/avatar/936383b387d6cc6d51279e429e5d7cd9?s=32&d=identicon&r=PG](https://superuser.com/users/90668/flimm)](https://superuser.com/users/90668/flimm)

[Flimm](https://superuser.com/users/90668/flimm)

**4,972**32132

* 1

I had to export the *shell* variable to an environment variable to make this work, i.e. export GIT\_SSH\_COMMAND="ssh -i ~/.ssh/id\_rsa\_example", then git clone example – [Abdull](https://superuser.com/users/137881/abdull" \o "1,301 reputation) Dec 1 '15 at 13:46

* 6

@Abdull In Bash, doing the assignment on the same line as the command exports the environment variable for just that command. Try it: example=hello /usr/bin/env | grep example. – [Flimm](https://superuser.com/users/90668/flimm" \o "4,972 reputation) Jan 8 '16 at 9:50

* 3

things have become even better: as of Git 2.10, you can store the command in your Git configuration: [stackoverflow.com/a/38474220/520162](http://stackoverflow.com/a/38474220/520162) – [eckes](https://superuser.com/users/62229/eckes" \o "278 reputation) Oct 21 '16 at 7:57

* 1

@Noitidart /dev/null is only a valid filename in UNIX-like operating systems, it doesn't work on Windows. – [Flimm](https://superuser.com/users/90668/flimm" \o "4,972 reputation) Mar 7 '17 at 8:19

* 1

If you need multiple keys, the -i parameter can be repeated, and ssh will try each key in turn. git config core.sshcommand "ssh -i /path/to/keyA -i /path/to/keyB". This lets git use different keys with different remote hosts. – [Mark](https://superuser.com/users/75124/mark) Jun 23 '17 at 3:44

[show **6** more comments](https://superuser.com/questions/232373/how-to-tell-git-which-private-key-to-use/912281)

92

There is **no *direct* way** to tell git which private key to use, because it relies on ssh for repository authentication. However, there are still a few ways to achieve your goal:

Option 1: ssh-agent

You can use ssh-agent to temporarily authorize your private key.

For example:

$ ssh-agent sh -c 'ssh-add ~/.ssh/id\_rsa; git fetch user@host'

Option 2: GIT\_SSH\_COMMAND

Pass the ssh arguments by using the GIT\_SSH\_COMMAND environment variable (Git 2.3.0+).

For example:

$ GIT\_SSH\_COMMAND='ssh -i ~/.ssh/id\_rsa -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no' \

git clone user@host

You can type this all on one line — ignore$and leave out the\.

Option 3: GIT\_SSH

Pass the ssh arguments by using the GIT\_SSH environment variable to specify alternate ssh binary.

For example:

$ echo 'ssh -i ~/.ssh/id\_rsa -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $\*' > ssh

$ chmod +x ssh

$ GIT\_TRACE=1 GIT\_SSH='./ssh' git clone user@host

Note: The above lines are shell (terminal) command lines which you should paste into your terminal. They will create a file named*ssh*, make it executable, and (indirectly) execute it.

Note:[GIT\_SSH is available since v0.99.4](https://github.com/git/git/commit/4852f7232b7a83fbd1b745520181bd67bf95911b)(2005).

Option 4: ~/.ssh/config

Use the ~/.ssh/config file as suggested in other answers in order to specify the location of your private key, e.g.

Host github.com

User git

Hostname github.com

IdentityFile ~/.ssh/id\_rsa

[share](https://superuser.com/a/868699)[improve this answer](https://superuser.com/posts/868699/edit)

[edited Nov 17 '17 at 14:13](https://superuser.com/posts/868699/revisions)

answered Jan 23 '15 at 22:08

[[https://www.gravatar.com/avatar/8bbc94977312d285045b2412257b6cb8?s=32&d=identicon&r=PG](https://superuser.com/users/87805/kenorb)](https://superuser.com/users/87805/kenorb)

[kenorb](https://superuser.com/users/87805/kenorb)

**12.1k**1783123

* 1

// , What if your identity in ssh-agent is forwarded, though, as in this question? [superuser.com/questions/971732/…](http://superuser.com/questions/971732/how-do-i-undo-an-ssh-add-on-a-forwarded-identity-to-access-github) – [Nathan Basanese](https://superuser.com/users/383483/nathan-basanese) Sep 11 '15 at 18:06

* 1

I've allowed me to reformat this post: IMO this is by far the most comprehensive answer. In its original design, a quick scan suggested the post where describing a single complicated solution to the problem, so I missed it. – [Alberto](https://superuser.com/users/77252/alberto) Jan 21 '16 at 10:01

* 2

$ ssh-agent sh -c 'ssh-add ~/.ssh/id\_rsa; git fetch user@host' worked for me when nothing else would. Kudos. – [Daniel Dewhurst](https://superuser.com/users/519714/daniel-dewhurst) Sep 11 '17 at 15:24

* I had to use ~/.ssh/config method, env vars didn't work for me... – [Greg Dubicki](https://superuser.com/users/254433/greg-dubicki) Sep 22 '17 at 7:53
* 1

GIT\_SSH is available since [v0.99.4 (August 2005)](https://github.com/git/git/commit/4852f7232b7a83fbd1b745520181bd67bf95911b), so basically since Git exists (April 2005). – [Dominik](https://superuser.com/users/126804/dominik) Nov 17 '17 at 8:10

add a comment

32

Write a script that calls ssh with the arguments you want, and put the filename of the script in $GIT\_SSH. Or just put your configuration in ~/.ssh/config.

[share](https://superuser.com/a/232375)[improve this answer](https://superuser.com/posts/232375/edit)

answered Jan 12 '11 at 18:25

[[https://www.gravatar.com/avatar/2343ae368d3241278581ce6c87f62a25?s=32&d=identicon&r=PG](https://superuser.com/users/26316/ignacio-vazquez-abrams)](https://superuser.com/users/26316/ignacio-vazquez-abrams)

[Ignacio Vazquez-Abrams](https://superuser.com/users/26316/ignacio-vazquez-abrams)

**97.5k**7156213

* 2

[Another explanation](http://superuser.com/a/355696/235231) of how to do this. – [Sithsu](https://superuser.com/users/235231/sithsu" \o "185 reputation) May 12 '14 at 19:44

* 1

~/.ssh/config Is *the* way to go. – [hek2mgl](https://superuser.com/users/209716/hek2mgl) May 8 '15 at 13:56

* I work on a machine (A) from which I git push to a server (B) that only accepts ssh key authentication. While my ~/.ssh/config setup on (A) works perfectly fine when I work directly *on* that machine, it does not when I login from some other location (C). Using $GIT\_SSH and a script solved this problem. Thanks! – [bsumirak](https://superuser.com/users/529341/bsumirak" \o "101 reputation) Dec 3 '15 at 17:28

add a comment

17

If you do not want to have to specify environment variables every time you run git, do not want another wrapper script, do not/can not run ssh-agent(1), nor want to download another package just for this, use the git-remote-ext(1) external transport:

$ git clone 'ext::ssh -i $HOME/.ssh/alternate\_id git.example.com %S /path/to/repository.git'

Cloning into 'repository'

(...)

$ cd repository

$ git remote -v

origin ext::ssh -i $HOME/.ssh/alternate\_id git.example.com %S /path/to/repository.git (fetch)

origin ext::ssh -i $HOME/.ssh/alternate\_id git.example.com %S /path/to/repository.git (push)

I consider this solution superior because:

* It is repository/remote specific
* Avoid wrapper script bloat
* Do not need the SSH agent -- useful if you want unattended clones/push/pulls (e.g. in cron)
* Definitely, no external tool needed

[share](https://superuser.com/a/943533)[improve this answer](https://superuser.com/posts/943533/edit)

answered Jul 21 '15 at 19:44

[[https://www.gravatar.com/avatar/3df252678554c7456193655d56568f98?s=32&d=identicon&r=PG&f=1](https://superuser.com/users/298109/flaviovs)](https://superuser.com/users/298109/flaviovs)

[flaviovs](https://superuser.com/users/298109/flaviovs)

**189**15

* 2

This answer was exactly what I needed to force Chef's git resource to use repository-specific deployment keys to clone/fetch from private Github repositories. The additional advantage of this method over the environment/script based ones is that since the key-path is encoded in the working-repo's config, it will use the same key on both initial clone and subsequent fetches/pushes. – [Adam Franco](https://superuser.com/users/9110/adam-franco) Nov 19 '15 at 16:20

* 2

WOW! This is just great, didn't know about this. Thanks for the answer, quite helpful as well in puppet environments, to prevent the extra hassle to manage .ssh/config etc. +1! – [gf\_](https://superuser.com/users/421253/gf) Jun 10 '16 at 18:36

* 1

This solution does not work together with the --recursive flag. The submodules are not fetched using the specified key and therefor fails if they require auth. – [Daniele Testa](https://superuser.com/users/381706/daniele-testa) Dec 21 '16 at 16:05

* 1

Each submodule is an entirely different repository, with their own set of remotes. They're glued together by Git for your convenience, but in no way remotes for a submodule are tied to the ones in the parent repository. I'm afraid that you must set the remote using the ext transport in *each* submodule for recursion in the parent to work. – [flaviovs](https://superuser.com/users/298109/flaviovs" \o "189 reputation) Dec 22 '16 at 19:10

* 2

If you encounter the following error fatal: transport 'ext' not allowed, you have to whitelist the ext protocol via the export GIT\_ALLOW\_PROTOCOL=ext. Basically, the git-remote-ext remote helper (which supports "ext::ssh example.com %S foo/repo" URLs) allows arbitrary command execution. This normally isn't ever a concern because user always sees and trusts the URL they pass to git. However git submodules, through the .gitmodules file, allow an attacker to request the client to fetch arbitrary git URLs. [hackerone.com/reports/104465](https://hackerone.com/reports/104465) – [Gomino](https://superuser.com/users/258763/gomino" \o "101 reputation) May 13 '18 at 17:21

[show **9** more comments](https://superuser.com/questions/232373/how-to-tell-git-which-private-key-to-use/912281)

15

After my struggle with $GIT\_SSH I would like to share what worked for me.

Through my examples I will assume you have your private key located at/home/user/.ssh/jenkins

Error to avoid: GIT\_SSH value includes options

$ export GIT\_SSH="ssh -i /home/user/.ssh/jenkins"

or whatever similar will fails, as **git will try to execute the value as a file**. For that reason, you have to create a script.

Working example of $GIT\_SSH script /home/user/gssh.sh

The script will be invoked as follows:

$ $GIT\_SSH [username@]host [-p <port>] <command>

Sample script working could look like:

#!/bin/sh

ssh -i /home/user/.ssh/jenkins $\*

Note the $\* at the end, it is important part of it.

Even safer alternative, which would prevent any possible conflict with anything in your default config file (plus explicitly mentioning the port to use) would be:

#!/bin/sh

ssh -i /home/user/.ssh/jenkins -F /dev/null -p 22 $\*

Assuming the script is in /home/user/gssh.sh, you shall then:

$ export GIT\_SSH=/home/user/gssh.sh

and all shall work.

[share](https://superuser.com/a/920849)[improve this answer](https://superuser.com/posts/920849/edit)

answered May 28 '15 at 17:09

[[https://www.gravatar.com/avatar/b0a60a250d33ebf1c57ca7e603e21311?s=32&d=identicon&r=PG](https://superuser.com/users/153413/jan-vlcinsky)](https://superuser.com/users/153413/jan-vlcinsky)

[Jan Vlcinsky](https://superuser.com/users/153413/jan-vlcinsky)

**346**29

* Thanks. Just note: use "$@" instead of $\* for pass-thru arguments, as the former behaves correctly when arguments contain whitespace. – [Piotr Findeisen](https://superuser.com/users/29509/piotr-findeisen) Mar 31 '16 at 7:39
* @PiotrFindeisen Thanks for your note. However, I do not understand it completely - in zsh it helps me to keep strings with space in one piece, but in bash not. Can you tell me more or point to some explanation? I do not want to add some modification blindly. – [Jan Vlcinsky](https://superuser.com/users/153413/jan-vlcinsky) Mar 31 '16 at 10:33
* You should remove the first half of your answer. No one's interested in a solution that doesn't work, and it's wasted reading that obfuscates the correct answer at the bottom, which works wonderfully. – [Cerin](https://superuser.com/users/35806/cerin" \o "2,822 reputation) Aug 3 '18 at 16:47
* @Cerin If you mean removing the "Error to avoid" I am going to keep it there. It shares common pitfall to avoid and it is very short. I am sure, someone would try optimizing the solution by providing all the things into variable (this happened to me), so I tried to shorten the path to success. – [Jan Vlcinsky](https://superuser.com/users/153413/jan-vlcinsky) Aug 4 '18 at 20:14

add a comment

13

Use custom host config in ~/.ssh/config, like this:

Host gitlab-as-thuc

HostName github.com

User git

IdentityFile ~/.ssh/id\_rsa.thuc

IdentitiesOnly yes

then use your custom hostname like this:

git remote add thuc git@gitlab-as-thuc:your-repo.git

[share](https://superuser.com/a/1077869)[improve this answer](https://superuser.com/posts/1077869/edit)

[edited Apr 18 '18 at 4:53](https://superuser.com/posts/1077869/revisions)

answered May 17 '16 at 15:03

[[https://www.gravatar.com/avatar/6bf9cd2937957ea226eee4421bf167a9?s=32&d=identicon&r=PG](https://superuser.com/users/364362/thucnguyen)](https://superuser.com/users/364362/thucnguyen)

[thucnguyen](https://superuser.com/users/364362/thucnguyen)

**239**24

* 1

This is the answer I was looking for, as I have separate GitHub accounts for home and work. I just had to set Host work.github.com HostName github.com IdentityFile ~/.ssh/work, and then replace "github.com" by "work.github.com" whenever I clone a work repository. It still connects to "github.com", but using a non-default key pair. – [Mikkel](https://superuser.com/users/141380/mikkel" \o "395 reputation) May 25 '16 at 17:06

* 1

The URL for details ("[itblog.study.land/..."](http://itblog.study.land/...%22)) doesn't work any more :( – [Carl Smotricz](https://superuser.com/users/18048/carl-smotricz) Sep 15 '17 at 8:10

* @CarlSmotricz the original one was moved here: [medium.com/@thucnc/…](https://medium.com/@thucnc/how-to-specify-different-ssh-keys-for-git-push-for-a-given-domain-bef56639dc02) – [thucnguyen](https://superuser.com/users/364362/thucnguyen) Nov 27 '18 at 7:13
* 3

FINALLY!!! This answer actually shows how you can utilize what you put in the ~/.ssh/config file. Every other answer misses how you can set the host when you add the origin, which automatically allows git to use the correct key file. THANK YOU!! – [BrianVPS](https://superuser.com/users/732122/brianvps" \o "101 reputation) Dec 4 '18 at 15:17

* Nice, that was what I was looking :) – [Lucas D'Avila](https://superuser.com/users/1017047/lucas-davila) Apr 4 at 18:19

add a comment

5

You can just use ssh-ident instead of creating your own wrapper.

You can read more at: <https://github.com/ccontavalli/ssh-ident>

It loads ssh keys on demand when first needed, once, even with multiple login sessions, xterms or NFS shared homes.

With a tiny config file, it can automatically load different keys and keep them separated in different agents (for agent forwarding) depending on what you need to do.

[share](https://superuser.com/a/570191)[improve this answer](https://superuser.com/posts/570191/edit)

[edited Jul 15 '14 at 18:44](https://superuser.com/posts/570191/revisions)

[[https://www.gravatar.com/avatar/a007be5a61f6aa8f3e85ae2fc18dd66e?s=32&d=identicon&r=PG](https://superuser.com/users/-1/community)](https://superuser.com/users/-1/community)

[Community](https://superuser.com/users/-1/community)**♦**

**1**

answered Mar 23 '13 at 1:35

[[https://www.gravatar.com/avatar/e3cd664ef327c2b4543e7b4e16fc9a2a?s=32&d=identicon&r=PG](https://superuser.com/users/209591/rabexc)](https://superuser.com/users/209591/rabexc)

[rabexc](https://superuser.com/users/209591/rabexc)

**161**12

add a comment

4

I had a client that needed a separate github account. So I needed to use a separate key just for this one project.

My solution was to add this to my .zshrc / .bashrc:

alias infogit="GIT\_SSH\_COMMAND=\"ssh -i ~/.ssh/id\_specialkey\" git $@"

Whenever I want to use git for that project I replace "infogit" with git:

infogit commit -am "Some message" && infogit push

For me, it's easier to remember.

[share](https://superuser.com/a/1308163)[improve this answer](https://superuser.com/posts/1308163/edit)

answered Mar 26 '18 at 19:26

[Michael Cole](https://superuser.com/users/333426/michael-cole)

**163**4

add a comment

3

So I set the GIT\_SSH env variable to $HOME/bin/git-ssh.

In order to support having my repo configuration dictate which ssh identity to use, my ~/bin/git-sshfile is this:

#!/bin/sh

ssh -i $(git config --get ssh.identity) -F /dev/null -p 22 $\*

Then I have a global git config setting:

$ git config --global ssh.identity ~/.ssh/default\_id\_rsa

And within any git repository I can just set a local ssh.identity git config value:

$ git config --local ssh.identity ~/.ssh/any\_other\_id\_rsa

Voila!

If you can have a different email address for each identity, it gets even simpler, because you can just name your keys after your email addresses and then have the git config's user.email drive the key selection in a ~/bin/git-ssh like this:

#!/bin/sh

ssh -i $HOME/.ssh/$(git config --get user.email) -F /dev/null -p 22 $\*

[share](https://superuser.com/a/1380854)[improve this answer](https://superuser.com/posts/1380854/edit)

[edited Dec 5 '18 at 23:27](https://superuser.com/posts/1380854/revisions)

answered Dec 4 '18 at 22:21

[[https://www.gravatar.com/avatar/944973814a5fb4df6912106ff16074a0?s=32&d=identicon&r=PG](https://superuser.com/users/969977/brendan-baldwin)](https://superuser.com/users/969977/brendan-baldwin)

[Brendan Baldwin](https://superuser.com/users/969977/brendan-baldwin)

**31**3

add a comment

2

My solution was this:

create a script:

#!/bin/bash

KEY=dafault\_key\_to\_be\_used

PORT=10022 #default port...

for i in $@;do

case $i in

--port=\*)

PORT="${i:7}";;

--key=\*)KEY="${i:6}";;

esac

done

export GIT\_SSH\_COMMAND="ssh -i $HOME/.ssh/${KEY} -p ${PORT}"

echo Command: $GIT\_SSH\_COMMAND

then when you have to change the var run:

. ./thescript.sh [--port=] [--key=]

Don't forget the extra dot!! this makes the script set the environments vars!! --key and --port are optional.

[share](https://superuser.com/a/1009458)[improve this answer](https://superuser.com/posts/1009458/edit)

answered Dec 5 '15 at 12:22

[[https://www.gravatar.com/avatar/2c8e23258738dfd044898d6816fe74a0?s=32&d=identicon&r=PG](https://superuser.com/users/530046/salsicha)](https://superuser.com/users/530046/salsicha)

[Salsicha](https://superuser.com/users/530046/salsicha)

**21**1

add a comment

2

I build on @shellholic and [this SO thread](https://stackoverflow.com/questions/4565700/specify-private-ssh-key-to-use-when-executing-shell-command) with a few teaks. I use GitHub as an example and assume that you have a private key in ~/.ssh/github (otherwise, see [this SO thread](http://stackoverflow.com/questions/50096/how-to-pass-password-to-scp)) and that you added the public key to your GitHub profile (otherwise see [GitHub's help](https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/)).

If needed, create a new SSH config file at ~/.ssh/config and change permissions to 400

touch ~/.ssh/config

chmod 600 ~/.ssh/config

Add this to the ~/.ssh/config file:

Host github.com

IdentityFile ~/.ssh/github

IdentitiesOnly yes

If you already have a remote set up, you may want to delete it, otherwise you may still be prompted for username and password:

git remote rm origin

Then add a remote to the git repository, and notice the colon before the user name:

git remote add origin git@github.com:user\_name/repo\_name.git

And then git commands work normally, e.g.:

git push origin master

git pull origin

@HeyWatchThis on [this SO thread](https://stackoverflow.com/questions/4565700/specify-private-ssh-key-to-use-when-executing-shell-command) suggested adding IdentitiesOnly yes to prevent the SSH default behavior of sending the identity file matching the default filename for each protocol. See that thread for more information and references.

[share](https://superuser.com/a/1319381)[improve this answer](https://superuser.com/posts/1319381/edit)

answered May 3 '18 at 10:17

[[https://www.gravatar.com/avatar/bc11aefb52952cbf4956c0bb13920d40?s=32&d=identicon&r=PG&f=1](https://superuser.com/users/868320/mmorin)](https://superuser.com/users/868320/mmorin)

[mmorin](https://superuser.com/users/868320/mmorin)

**305**313

* This was my mistake: "If you already have a remote set up...". Thanks a lot!!! – [Allan Andrade](https://superuser.com/users/607812/allan-andrade) Oct 4 '18 at 17:43
* common mistake---this is the answer – [Goddard](https://superuser.com/users/276523/goddard) May 16 at 21:07

add a comment

1

Generally, you want to use ~/.ssh/config for this. Simply pair server addresses with the keys you want to use for them as follows:

Host github.com

IdentityFile ~/.ssh/id\_rsa.github

Host heroku.com

IdentityFile ~/.ssh/id\_rsa.heroku

Host \*

IdentityFile ~/.ssh/id\_rsa

Host \* denotes any server, so I use it to set ~/.ssh/id\_rsa as the default key to use.

[share](https://superuser.com/a/1038762)[improve this answer](https://superuser.com/posts/1038762/edit)

answered Feb 11 '16 at 22:44

[[https://www.gravatar.com/avatar/7bc4d5f38183597c6e47f3f6f3be2ff6?s=32&d=identicon&r=PG](https://superuser.com/users/44499/zaz)](https://superuser.com/users/44499/zaz)

[Zaz](https://superuser.com/users/44499/zaz)

**1,408**21530

add a comment

1

Just use ssh-agent and ssh-add commands.

# create an agent

ssh-agent

# add your default key

ssh-add ~/.ssh/id\_rsa

# add your second key

ssh-add ~/.ssh/<your key name>

After executing the above commands, you can use both keys as same time. Just type

git clone git@github.com:<yourname>/<your-repo>.git

to clone your repository.

You need to execute the above command after you reboot your machine.

[share](https://superuser.com/a/1336671)[improve this answer](https://superuser.com/posts/1336671/edit)

[edited Mar 17 at 22:52](https://superuser.com/posts/1336671/revisions)

[Run5k](https://superuser.com/users/650163/run5k)

**11.9k**73555

answered Jul 5 '18 at 9:53

[Jinmiao Luo](https://superuser.com/users/767644/jinmiao-luo)

**11**3

* Please explain the process including How can I create an agent – [Srikrushna Pal](https://superuser.com/users/1017985/srikrushna-pal" \o "101 reputation) Apr 6 at 20:20

add a comment

0

I'm using git version 2.16 and I don't need a single piece of script not even a config or modified commands.

* Just copied my private key to .ssh/id\_rsa
* set permissions to 600

And git reads to key automatically. I doesn't ask anything and it doesn't throw an error. Just works fine.

[share](https://superuser.com/a/1328970)[improve this answer](https://superuser.com/posts/1328970/edit)

answered Jun 6 '18 at 4:37

[akprajapati](https://superuser.com/users/754243/akprajapati)

**21**3

* Did you notice that the question is about “a system with multiple private keys in the ~/.ssh directory”? – [Scott](https://superuser.com/users/150988/scott" \o "16,844 reputation)Jun 6 '18 at 5:16

add a comment

0

While the question doesn't request it, I am including this answer for anyone else looking to solve the same problem just specifically for [gitlab](https://superuser.com/questions/tagged/gitlab" \o "show questions tagged 'gitlab').

**The gitlab solution**

I tried using the [environment-variables](https://superuser.com/questions/tagged/environment-variables) approach, but even [the git documentation](https://git-scm.com/book/en/v2/Git-Internals-Environment-Variables) recommends using ~/.ssh/config for anything more than the simple case. In my case I am pushing to a [gitlab](https://superuser.com/questions/tagged/gitlab" \o "show questions tagged 'gitlab') server - and I wanted to do so as a specific user - which is of course defined by the [private-key](https://superuser.com/questions/tagged/private-key) during [authentication](https://superuser.com/questions/tagged/authentication) and not the username git. Once implemented I simply perform the following:

~/myrepo> git mycommit -m "Important Stuff"

~/myrepo> git mypush

[proceed to enter passphrase for private key...]

**Setup**

Recall the location of your [private-key](https://superuser.com/questions/tagged/private-key) /myfolder/.ssh/my\_gitlab\_id\_rsa in my case.

Add an entry in ~/.ssh/config:

Host gitlab-delegate

HostName gitlab.mydomain.com

User git

IdentityFile /myfolder/.ssh/my\_gitlab\_id\_rsa

IdentitiesOnly yes

Add the [git-alias](https://superuser.com/questions/tagged/git-alias" \o "show questions tagged 'git-alias') in ~/.gitconfig:

mypush = "!f() { \

path=$(git config --get remote.origin.url | cut -d':' -f2); \

branch=$(git rev-parse --abbrev-ref HEAD); \

git remote add gitlab\_as\_me git@gitlab-delegate:$path && \

git push gitlab\_as\_me $branch && \

git pull origin $branch; \

git remote remove gitlab\_as\_me; \

}; f"

As a bonus, I [perform my commits on this same host as a specific user](https://stackoverflow.com/a/19841156/1808261) with this [git-alias](https://superuser.com/questions/tagged/git-alias" \o "show questions tagged 'git-alias'):

mycommit = "!f() { \

git -c "user.name=myname" -c "user.email=myname@mysite.com" commit \"$@\"; \

}; f"

**Explanation**

All of the above assumes the relevant remote is origin and the relevant branch is currently checked out. For reference I ran into several items that needed to be addressed:

* The solution requires creating a new remote gitlab\_as\_me, and I didn't like seeing the extra remote [hanging around in my log tree](https://www.merriam-webster.com/dictionary/pun) so I remove it when finished
* In order to create the remote, there is a need to generate the remote's url on the fly - in the case of gitlab this was achieved with a simple bash [cut](https://superuser.com/questions/tagged/cut)
* When performing a push to gitlab\_as\_me you need to be specific about what branch you are pushing
* After performing the push your local origin pointer needs to be "updated" in order to match gitlab\_as\_me (the git pull origin $branch does this)

[share](https://superuser.com/a/1391593)[improve this answer](https://superuser.com/posts/1391593/edit)

answered Jan 7 at 18:26

[dtmland](https://superuser.com/users/201302/dtmland)

**1,886**11427

add a comment

0

When you have multiple git account and you want different ssh key

You have to follow same step for generating the ssh key, but be sure you

ssh-keygen -t ed25519 -C "your-email-id@gmail.com"

Enter the path you want to save(Ex: my-pc/Desktop/.ssh/ed25519)

Add the public key to your gitlab ([How to adding ssh key to gitlab](https://gitlab.com/help/ssh/README#generating-a-new-ssh-key-pair))

You have to new ssh identity using the below comand

ssh-add ~/my-pc/Desktop/.ssh/ed25519

[share](https://superuser.com/a/1422328)[improve this answer](https://superuser.com/posts/1422328/edit)

answered Apr 6 at 20:42

[Srikrushna Pal](https://superuser.com/users/1017985/srikrushna-pal)

**101**1

* (1) Are you quoting somebody or something? If so, please identify the source. If not, please don’t use quote formatting.  (2) What is “ed25519”? … … … … … … … … … … … Please do not respond in comments; [edit](https://superuser.com/posts/1422328/edit) your answer to make it clearer and more complete. – [Scott](https://superuser.com/users/150988/scott) Apr 7 at 0:35

add a comment

Not the answer you're looking for? Browse other questions tagged [ssh](https://superuser.com/questions/tagged/ssh" \o "show questions tagged 'ssh') [git](https://superuser.com/questions/tagged/git" \o "show questions tagged 'git') [authentication](https://superuser.com/questions/tagged/authentication) [private-key](https://superuser.com/questions/tagged/private-key)or [ask your own question](https://superuser.com/questions/ask).