

oanhnn / using-multiple-github-accounts-with-ssh-keys.md

Last active yesterday • Report abuse



Star

<> Code Revisions 9 Stars 1,142 Forks 363

Using multiple github accounts with ssh keys

using-multiple-github-accounts-with-ssh-keys.md

Problem

I have two Github accounts: *oanhnn* (personal) and *superman* (for work). I want to use both accounts on same computer (without typing password everytime, when doing git push or pull).

Solution

Use ssh keys and define host aliases in ssh config file (each alias for an account).

How to?

1. Generate ssh key pairs for accounts and add them to GitHub accounts.
2. Edit/Create ssh config file (`~/.ssh/config`):

```
# Default github account: oanhnn
Host github.com
  HostName github.com
  IdentityFile ~/.ssh/oanhnn_private_key
  IdentitiesOnly yes

# Other github account: superman
Host github-superman
  HostName github.com
  IdentityFile ~/.ssh/superman_private_key
  IdentitiesOnly yes
```

NOTE: If you use any account frequently, you should use the default hostname (`github.com`).

3. Add ssh private keys to your agent:

```
$ ssh-add ~/.ssh/oanhnn_private_key
$ ssh-add ~/.ssh/superman_private_key
```

4. Test your connection

```
$ ssh-keyscan github.com >> ~/.ssh/known_hosts
$ ssh -T git@github.com
$ ssh -T git@github-superman
```

If everything is OK, you will see these messages:

```
Hi oanhnn! You've successfully authenticated, but GitHub does not provide
```



```
Hi superman! You've successfully authenticated, but GitHub does not provi
```



5. Now all are set, you need remeber

```
git@github-superman:org/project.git => user is superman
git@github.com:org/project.git.      => user is oanhnn
```

- If you need clone a repository, just do:

```
$ git clone git@github-superman:org1/project1.git /path/to/project1
$ cd /path/to/project1
$ git config user.email "superman@example.com"
$ git config user.name "Super Man"
```

- If you already have the repo set up, after the ssh config instructions, you need change the URL of origin , just do:

```
$ cd /path/to/project2
$ git remote set-url origin git@github-superman:org2/project2.git
$ git config user.email "superman@example.com"
$ git config user.name "Super Man"
```

- If you are creating a new repository on local:

```
$ cd /path/to/project3
$ git init
$ git remote add origin git@github-superman:org3/project3.git
$ git config user.email "superman@example.com"
```

```
$ git config user.name "Super Man"
$ git add .
$ git commit -m "Initial commit"
$ git push -u origin master
```

Done! Goodluck!

Addon:

The bash script that prompts for your git account. Thank @davorpa

```
#!/bin/bash

# silent prompt
read -p 'GIT profile: ' profile

# switch
case $profile in
    superman)
        git config user.email "superman@example.com"
        git config user.name "superman"
        git config user.signingKey "superman_gpg_public_key"
        ;;
    oanhnn)
        git config user.email "oanhnn@example.com"
        git config user.name "oanhnn"
        git config user.signingKey "oanhnn_gpg_public_key"
        ;;
    # default case: raise error
    *)
        >&2 echo "ERR: Unknown profile: $profile"
        exit 1
esac
```

[Load earlier comments...](#)

chenxiao218 commented on Mar 12 • edited ▾

```
git remote set-url origin git@github-personal:user/repo-name.git
```

This works to me, thanks.

Pfuufy commented on Apr 7

Very helpful. Thanks!

webThreeBuilder commented on Apr 19

If you already have the repo set up, after the ssh config instructions, just do:

- Remote is called probably `origin`
- I have work and personal accounts, for this I used `git@github-personal` as host from your SSH config
- `user/repo-name.git` for the Github repository

```
$ git remote set-url origin git@github-personal:user/repo-name.git
```

Hi, thanks for your good tips 👍, its worked for my case by your great comment, after following the steps above, we also have to reset the remote origin url with our custom "git-account-host" in the config

abdeljabarTaoufikallahPro commented on Apr 21

Thanks man

akhill4054 commented on Apr 30

Thank you man! You are a life saver!

datatravelandexperiments commented on May 3

Thank you! Note that if you have ssh multiplexing set up globally, especially with persistence, you may want to disable it for github, since multiplexing works per-machine and not per `Host` entry. Add `ControlMaster no` to each `Host` configuration.

silicakes commented on May 9

This is golden, thanks!

aicals commented on Jun 2

Pure gold! Thanks!

mahe113vsp commented on Jun 9

thank you!

cognivator commented on Jun 13

@jdvivar Many thanks. This host customization works great for tools like SourceTree in which you don't enter the ssh host alias directly.

kqfu commented on Jun 17

Great post. This is by far the cleanest solution I've found.

aacassandra commented on Aug 12

great! thankyou sir

rohanrmallya commented on Aug 12

Neat! 🔥 Thanks for this. :)

odooerpdevelopers commented on Aug 16 • edited ▼

| \$

Thanks bro, I had problems with this configuration since I work with docker and docker recommends these settings to use remote connections, but in github I had to uncomment those lines and it works fine :))
(~/.ssh/config)
#ControlMaster auto
#ControlPath ~/.ssh/control-%C
#ControlPersist yes

ImadMAKS commented on Aug 17

Thank you for this post; very clear and helpful!
I have a question regarding the host, though. When I change the origin host on my local repo, shouldn't I change it remotely in my GitHub account? I can't figure out how, though.
Maybe I am missing something, but I tried changing it only locally, and I tried to fetch I get from the same repo and I still get:

```
ERROR: Repository not found.  
fatal: Could not read from remote repository.  
Please make sure you have the correct access rights, and the repository exists.
```

I'm wondering what I'm missing.

My config file contains the following:

```
Host git@github.com  
    HostName github.com  
    User git
```

```
IdentityFile ~/.ssh/X.pub
IdentitiesOnly yes
Host git@github.com-work
  HostName github.com
  User git
  IdentityFile ~/.ssh/Y.pub
  IdentitiesOnly yes
```

I changed my local git repo's URL to `git@github.com-work:username/an-example.git`

oanhnn commented on Aug 18

Author

@ImadMAKS

Please using Host github.com and Host github.com-work instead of Host git@github.com and Host git@github.com-work in your config.

After that, you need change remote URL of all repository for work.

You can use below command:

```
$ git remote set-url origin git@github.com-work:username/an-example.git
```

thanhan2101 commented on Aug 25

Thank for your guide, it works with me.

arpit-turing commented on Aug 25

Thank you mate for this.

ImadMAKS commented on Aug 25 • edited ▼

Please using Host github.com and Host github.com-work instead of Host [git@github.com](#) and Host [git@github.com-work](#) in your config

This worked! Thank you.

jakubkalicki commented on Aug 28

Great guide, it helps me a lot! And I can expand this guide with another trick.

In your `~/.gitconfig`

```
[url "git@github.com-company:company_github_account/"]
  insteadOf = git@github.com:company_github_account/
```

After this setting, you don't have to change URL manually every time when you want to clone repos.

Works flawlessly!

@oanhnn, would you mind including this into your gist?

oanhnn commented on Aug 30

Author

@jakubkalicki

I didn't use that way.

What will happen if I clone `git@github.com-company:company_github_account/project1.git` and `github.com-company:other-org/project2.git` after setting up in `~/.gitconfig` ?

jakubkalicki commented on Aug 30 • edited ▼

It should work as usual. What is cool about the `.gitconfig` thingy is that it allows for using standard url that you copy from Github repository page. It saves you from changing `github.com` to the custom Host name, because it will happen automatically. One less thing to remember about.

oanhnn commented on Aug 31 • edited ▼

Author

@jakubkalicki

I'm working fine with my configuration.

I'm not sure your configuration is correct in all cases, mine works fine (because it affects to SSH). I also don't have much time to compare and experiment with your configuration.

Sorry for that.

If possible, write a guide for newbies, I can link to it here.

manzloros commented on Sep 3 • edited ▼

These instructions didn't work for me.

```
# Other github account: my personal account
Host <my personal alias>
  HostName github.com
  IdentityFile ~/.ssh/<my private key generated separately from my work private key>
  IdentitiesOnly yes
```

```
ssh -T git@<my personal alias>
```

Hi <my WORK user name>! You've successfully authenticated, but GitHub does not provide shell access.

So even though I have the personal alias defined in the `~/.ssh/config` , `ssh -T` returns a response from GitHub that's using my work user name.

oanhnn commented on Sep 5

Author

@manzaloros Can you re-run `ssh -vT git@<my personal alias>` and see why it is not correct?

manzaloros commented on Sep 5

Thanks @oanhnn . What I don't understand is why once ssh recognizes I have that personal alias it continues to try my work account credentials:

```
debug1: /Users/<me>/.ssh/config line 9: include ~/.ssh/config.v1.1.67 matched no files
debug1: Reading configuration data /Users/<me>/.ssh/config.custom
debug1: /Users/<me>/.ssh/config.custom line 3: Applying options for <my personal alias>

...

debug1: identity file /Users/<me>/.ssh/<my personal alias>private_key type 3
debug1: identity file /Users/<me>/.ssh/<my personal alias>private_key-cert type -1
debug1: identity file /Users/<me>/.ssh/<my work account>ssh_key type 0
debug1: identity file /Users/<me>/.ssh/<my work account>_ssh_key-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_9.0
debug1: Remote protocol version 2.0, remote software version babeld-7e018303
debug1: compat_banner: no match: babeld-7e018303
debug1: Authenticating to github.com:22 as 'git'
debug1: load_hostkeys: fopen /Users/<me>/.ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or directory

...

debug1: Will attempt key: /Users/<me>/.ssh/<my work account>_ssh_key <RSA key> explicit
agent
debug1: Will attempt key: /Users/<me>/.ssh/<my personal account>private_key <ED key>
explicit

...

debug1: Offering public key: /Users/<me>/.ssh/<my work account>_ssh_key <RSA key>
explicit agent
debug1: Server accepts key: /Users/<me>/.ssh/<my work account>_ssh_key <RSA key> explicit
agent

debug1: client_input_hostkeys: searching /Users/<me>/.ssh/known_hosts for github.com /
(none)
debug1: client_input_hostkeys: searching /Users/<me>/.ssh/known_hosts2 for github.com /
(none)
debug1: client_input_hostkeys: hostkeys file /Users/<me>/.ssh/known_hosts2 does not exist
debug1: client_input_hostkeys: no new or deprecated keys from server
```

oanhnn commented on Sep 6 • edited ▼

Author

@manzaloros

I see SSH was attempted with two keys

- K1: /Users/<me>/ .ssh/<my work account>_ssh_key <RSA key>
- K2 /Users/<me>/ .ssh/<my personal account>private_key <ED key>

For some reason, both K1 and K2 match with host, but K1 was given priority over K2.

I'm not sure but it could be due to the key type and algorithms.

All algorithms involved in SSH connection, including the same process, select the host key type:

- First the server and the client exchange lists of algorithms they support
- Then one party (in this case the client) picks the one it prefers out of commonly supported algorithms.

You should:

- Check all `ssh_config` files and included files (include `~/ .ssh/config.v1.1.67` is missing file)
- Check all `ssh_known_hosts` for Github host and your alias host
- Check `HostKeyAlgorithms` config

My debug

```
$ ssh -vT <my host alias>
OpenSSH_9.0p1, LibreSSL 3.3.6
debug1: Reading configuration data /Users/<me>/ .ssh/config
debug1: Reading configuration data /Users/<me>/ .colima/ssh_config
debug1: /Users/<me>/ .ssh/config line 167: Applying options for <my host alias>
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 21: include /etc/ssh/ssh_config.d/* matched no files
debug1: /etc/ssh/ssh_config line 54: Applying options for *
debug1: Connecting to github.com port 22.
debug1: Connection established.
debug1: identity file /Users/<me>/ .ssh/id_rsa type 0
debug1: identity file /Users/<me>/ .ssh/id_rsa-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_9.0
debug1: Remote protocol version 2.0, remote software version babeld-7e018303
debug1: compat_banner: no match: babeld-7e018303
debug1: Authenticating to github.com:22 as 'git'
debug1: load_hostkeys: fopen /Users/<me>/ .ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or directory
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: algorithm: curve25519-sha256
debug1: kex: host key algorithm: ssh-ed25519
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit>
compression: none
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: SSH2_MSG_KEX_ECDH_REPLY received
debug1: Server host key: ssh-ed25519 SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU
debug1: load_hostkeys: fopen /Users/<me>/ .ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or directory
debug1: Host 'github.com' is known and matches the ED25519 host key.
debug1: Found key in /Users/<me>/ .ssh/known_hosts:6
debug1: rekey out after 134217728 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: rekey in after 134217728 blocks
```

```

debug1: get_agent_identities: bound agent to hostkey
debug1: get_agent_identities: agent returned 3 keys
debug1: Will attempt key: /Users/<me>/.ssh/id_rsa <RSA key> explicit agent
debug1: SSH2_MSG_EXT_INFO received
debug1: kex_input_ext_info: server-sig-algs=<ssh-ed25519-cert-v01@openssh.com,ecdsa-sha2-
nistp521-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-
nistp256-cert-v01@openssh.com,sk-ssh-ed25519-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-
cert-v01@openssh.com,rsa-sha2-512-cert-v01@openssh.com,rsa-sha2-256-cert-
v01@openssh.com,ssh-rsa-cert-v01@openssh.com,sk-ssh-ed25519@openssh.com,sk-ecdsa-sha2-
nistp256@openssh.com,ssh-ed25519,ecdsa-sha2-nistp521,ecdsa-sha2-nistp384,ecdsa-sha2-
nistp256,rsa-sha2-512,rsa-sha2-256,ssh-rsa>
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey
debug1: Next authentication method: publickey
debug1: Offering public key: /Users/<me>/.ssh/id_rsa <RSA key> explicit agent
debug1: Server accepts key: /Users/<me>/.ssh/id_rsa <RSA key> explicit agent
Authenticated to github.com ([xx.xx.xx.xx]:22) using "publickey".
debug1: channel 0: new [client-session]
debug1: Entering interactive session.
debug1: pledge: filesystem
debug1: client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
debug1: client_input_hostkeys: searching /Users/<me>/.ssh/known_hosts for github.com /
(none)
debug1: client_input_hostkeys: searching /Users/<me>/.ssh/known_hosts2 for github.com /
(none)
debug1: client_input_hostkeys: hostkeys file /Users/<me>/.ssh/known_hosts2 does not exist
debug1: client_input_hostkeys: no new or deprecated keys from server
debug1: Sending environment.
debug1: channel 0: setting env LC_TERMINAL_VERSION = "3.4.20"
debug1: channel 0: setting env LC_CTYPE = "UTF-8"
debug1: channel 0: setting env LC_TERMINAL = "iTerm2"
debug1: client_input_channel_req: channel 0 rtype exit-status reply 0
Hi <me>! You've successfully authenticated, but GitHub does not provide shell access.
debug1: channel 0: free: client-session, nchannels 1
Transferred: sent 3532, received 3056 bytes, in 0.9 seconds
Bytes per second: sent 4068.4, received 3520.1
debug1: Exit status 1

```

manzaloros commented on Sep 10 • edited ▼

Thanks @oanhnn .

One note is that I didn't add an alias for my work account — just one for my personal account.

[According to this discussion](#) it looks like the search for the host is recursive and once SSH finds my hostname is github.com, it just defaults to my work settings.

Also worth noting that adding

```
sshCommand = ssh -i ~/.ssh/my-personal-key -F /dev/null
```

to the repo's git config changes the error from

```
ERROR: Permission to <personal repo>.git denied to <my work account>.
```

to:

```
ssh: Could not resolve hostname <my personal alias>: nodename nor servname provided, or not known
```

ploissken commented last month

Finally something that works! Thank you so much!

fons-digitect commented 3 weeks ago • edited ▼

@oanhnn It is worth specifically mentioning that with step 4, the file should **literally** be called "config" without any extension. Otherwise testing the connection will fail, because your config file isn't recognized by ssh. Check out the green answer at: <https://github.com/orgs/community/discussions/22589>

For the rest this article helped, thanks!