

GitHub/GitLab: одновременная работа с несколькими репозиториями

Введение

По долгу службы приходится много работать с git, и чаще всего это клиент командной строки. Я не использую различные gui клиенты, а их на самом деле уже достаточно много:

- [source tree](#)
- [tortoise git](#)
- [GitHub Desktop](#)
- [giteye](#)
- [git for windows](#)
- [Git Extensions](#)

Возможно дело привычки, а возможно, мне просто не нужны все возможности таких клиентов. Как правило мне хватает git клиента из состава [cmdr](#), без которого работа в консоли windows это просто сущий ад.

Так как большая часть репозитариев использует аутентификацию по ключам, то для быстрого доступа очень удобно использовать настройки ssh клиента, которые описываются в файле `~/.ssh/config` и имеют вид

```
Host gitlab.example.com
  HostName gitlab.example.com
  User git
  IdentityFile ~/.ssh/gitlab.example.com.key
  IdentitiesOnly yes
  PreferredAuthentications publickey
```

В случае windows файл будет располагаться в `C:\Users\alex\.ssh\config`. Но в определенный момент возникает проблема, как производить аутентификацию, когда url репозитария один и тот же, а меняется лишь имя репозитария, ну например,

```
ssh://git@gitlab.example.com:2222/GR0UP1/project11.git
ssh://git@gitlab.example.com:2222/GR0UP2/project21.git
ssh://git@gitlab.example.com:2222/GR0UP3/project31.git
```

Есть несколько путей решения данной проблемы. Ниже рассмотрим их.

Настройка ssh клиента

Первый способ заключается в использовании т.н. алиасов, т.е. когда в Host мы указываем

уникальный идентификатор для связки host + repository. Ну например

```
Host group1-project1
  HostName gitlab.example.com
  User git
  IdentityFile ~/.ssh/group1_project1.key
  IdentitiesOnly yes
  PreferredAuthentications publickey
```

И затем при работе с этим репозитарием использовать

```
λ git clone ssh://group1-project1:2222/GR0UP1/project11.git
```

Итак, мой тестовый config файл имеет вид

```
λ cat ~/.ssh/config
Host github-casperjs
  HostName gitlab.example.net
  User git
  IdentityFile ~/.ssh/github_casperjs.key
  IdentitiesOnly yes
  PreferredAuthentications publickey

Host github-phantomjs
  HostName gitlab.example.net
  User git
  IdentityFile ~/.ssh/github_phantomjs.key
  IdentitiesOnly yes
  PreferredAuthentications publickey
```

А теперь пробуем клонировать проекты

```
λ git clone ssh://github-casperjs:2222/github/casperjs.git
Cloning into 'casperjs'...
remote: Counting objects: 14841, done.
remote: Compressing objects: 100% (5441/5441), done.
remote: Total 14841 (delta 9312), reused 14841 (delta 9312)
Receiving objects: 100% (14841/14841), 8.50 MiB | 1.53 MiB/s, done.
Resolving deltas: 100% (9312/9312), done.
Checking connectivity... done.
Checking out files: 100% (285/285), done.

λ git clone ssh://github-phantomjs:2222/github/phantomjs.git
Cloning into 'phantomjs'...
remote: Counting objects: 63957, done.
remote: Compressing objects: 100% (30569/30569), done.
remote: Total 63957 (delta 31979), reused 63574 (delta 31601)
Receiving objects: 100% (63957/63957), 127.61 MiB | 1.28 MiB/s, done.
Resolving deltas: 100% (31979/31979), done.
Checking connectivity... done.
```

```
Checking out files: 100% (638/638), done.
```

Как видим, все работает.

Использование GIT_SSH_COMMAND

Альтернативный вариант заключается в использовании переменной окружения **GIT_SSH_COMMAND**. Но данный вариант будет работать только на linux, ну либо при использовании git bash

```
$ GIT_SSH_COMMAND="ssh -i ~/.ssh/github_casperjs.key" git clone
git@gitlab.example.net:2222/github/casperjs.git
Cloning into 'casperjs'...
remote: Counting objects: 14841, done.
remote: Compressing objects: 100% (5441/5441), done.
remote: Total 14841 (delta 9312), reused 14841 (delta 9312)
Receiving objects: 100% (14841/14841), 8.50 MiB | 1.53 MiB/s, done.
Resolving deltas: 100% (9312/9312), done.
Checking connectivity... done.
Checking out files: 100% (285/285), done.
```

From:

<http://www.sys-adm.org.ua/> - **wiki.sys-adm.org.ua**

Permanent link:

<http://www.sys-adm.org.ua/vcs/multiple-gitlab-accounts-ssh-config>

Last update: **2016/05/30 21:58**

