

Authentication

Authenticating with Git

There are many ways to authenticate with Git when pushing to remote repositories. See some of the below for examples.

SSH Keys

You can use `ssh-keygen` to generate a key and then manually add the key to your GitHub profile. To do this, run `ssh-keygen -t ed25519` and follow the prompts. Be sure to include the file path when naming your key, or the key will be output into your working directory. Once generated, simply `cat user_ed25519.pub` and copy / paste your public key into the field within your settings on GitHub. To do this, you'll need to run `git remote set-url origin git@github.com:User/UserRepo.git` in order to configure your local repository to use SSH private keys when connecting to git. For more on ssh-keygen, check out [Creating SSH Login Keys](#).

Multiple Accounts

If you have multiple accounts on GitHub or GitLab, you can use the `~/.ssh/config` file to specify which users to relate to which key.

To get this working, you'll first need to generate your SSH keys and register them with the Git accounts you want to use them for. Then, you'll need to run `ssh-add /path/to/key_ed25519` and restart your terminal session for the changes to be applied.

After following the steps above, you can modify your `~/.ssh/config` to use the below settings for your own usernames and SSH keys.

```
# Run `ssh-add /path/to/user_ed25519` to register keys first
```

```
# GitHub
```

```
# You can set a default key for a domain
```

```
Host github.com
```

```
    HostName github.com
```

```
    Port 22
```

```
    IdentityFile /home/kapper/.ssh/fake_ed25519
```

```
# Or set a key for a fake.github.com domain
# + Then clone with `git clone git@fake.github.com:/user/repo.git`
Host fake.github.com
  HostName github.com
  User fake
  Port 22
  IdentityFile /home/kapper/.ssh/fake_ed25519

# GitLab

Host fake.GitLab
  HostName gitlab.com
  User fake
  Port 22
  IdentityFile /home/kapper/.ssh/fake_ed25519

Host diffake.gitlab.com
  HostName gitlab.com
  User different_fake
  Port 22
  IdentityFile /home/kapper/.ssh/different_fake_ed25519

# VPS Configurations

Host knoats.com
  HostName knoats.com
  User fake
  Port 22
  IdentityFile /home/kapper/.ssh/fake_ed25519

Host dev-box
  HostName 999.999.999.999
  User different_fake
  Port 22
  IdentityFile /home/kapper/.ssh/different_fake_ed25519
```

Personal Access Tokens

Alternatively, you could generate a static Personal Access Token - a token that once generated can be paired with a YubiKey or similar product. This allows you to clone / work from anywhere without having to provision or SSH keys or manage long passwords / 2FA methods. Plug your key into USB and tap your desired configuration and the static access key will be input, allowing immediate access *for this one time*. So, every time you push, unless you configure otherwise, you will have to enter this token by tapping the YubiKey.

Credential Caching

“ If you don’t want to authenticate every time you push, you can set up a “credential cache”. The simplest is just to keep it in memory for a few minutes, which you can easily set up by running `git config --global credential.helper cache`.

[https://git-scm.com/book/en/v2/Git-Tools-Credential-](https://git-scm.com/book/en/v2/Git-Tools-Credential-Storage#_credential_caching)

[Storage#_credential_caching](https://git-scm.com/book/en/v2/Git-Tools-Credential-Storage#_credential_caching)

Revision #3

Created 28 June 2020 14:21:56 by Shaun Reed

Updated 19 December 2021 02:46:07 by Shaun Reed