GIT: From Beginner to Fearless

Activity: Setting credentials

Brian Gorman, Author/Instructor/Trainer

Introduction

Setting your credentials is much more trivial now than it used to be. We are lucky that we have credential managers. Credential managers allow us to store our username and password combinations in one place and thereby forego entering our username/password combination on every critical action going forward.

In this activity, we'll take a look at some different ways that we can setup our credentials going forward. For the most part, however, we'll be relying on the credential managers. Additionally, not doing this activity just means you would need to enter your information more frequently – but that's only if you also don't have a credential manager in place by default.

Remember that there are three levels of config {System, Global, Local}. Since we are an individual user, the best place to store the credentials would be in the Global level [for the user account].

Let's get started!

Step 1: Determine your credentials:

a) *First, see if you have any credentials set*
[**`git config --list`**]

b) *If credentials are set, make a note of what they currently are.*

Step 2: See what it's like with no credentials:

This step shows us what it's like if credentials are not stored. We'll have to enter our credentials every time we want to do something that requires permission, even if the activity is not destructive.

a) *Remove your credentials*
[**`git config --global --unset-all credential.helper`**]
or [**`git config --unset-all credential.helper`**]

b) *Clone a private repo*
[**`git clone <url> <newFolderRepoName>`**]
requires permissions. If you don't have a private repo, create one at GitHub or BitBucket

c) *Enter your credentials*

d) Remove the repo by deleting the files and folders
[**`rm -rf <root_folder_name>`**]

e) *Repeat steps b and c.*
You should be required to enter your credentials again.

Step 3: Set up credentials with a cache:

This step takes us through what it's like to work with the cache to store credentials. Essentially, here we can setup our credentials to be temporarily stored in cache.

Notes

a) *Set credentials to timeout in 2 minutes [120 seconds]*

[**git config --global credential.helper 'cache --timeout=120'**]

b) *Clone the repo And enter your credentials*

[**git clone <url> <folder>**]

c) *Remove the repo from your file system*

[**rm –rf <foldername>**]

d) *Clone the repo – this time you should not have to enter credentials*

[**git clone <url> <folder>**]

e) *Remove the repo from your file system*

[**rm –rf <foldername>**]

f) *Play solitaire, look at facebook, etc, for 2 minutes*

g) *Clone the repo – expect to reenter your credentials*

[**git clone <url> <folder>**]

Step 4: Unset credentials:

a) *Unset existing credentials*

[**git config --global --unset-all credential.helper**]

b) *Remove the credential section from your config*

[**git config --global --remove-section credential**]

Step 5: Set credentials to permanent store:

a) *Set credentials to store*

[**git config --global credential.helper store**]

b) *Clone the repo – enter your credentials*

[**git clone <url> <folder>**]

c) *Remove the repo from your file system*

[**rm –rf <foldername>**]

d) *Play solitaire, look at facebook, etc, for 2 minutes*

e) *Clone the repo – no timeout so credentials should not need to be re-entered*

[**git clone <url> <folder>**]

Step 6: Resetting your credentials:

a) *Reset your credentials to use the credential manager [Windows]*
   [**`git config --global credential.helper manager`**]

b) *See additional resources to set up OS X Keychain [MAC]*

c) *See additional resources to use a keystore in Ubuntu [Linux]*

Step 7: View your credentials:

a) *List your current credentials*
   [**`git config --global --get credential.helper`**]

Closing Thoughts and additional resources

In this activity, we've seen how we can use various methods to store our credentials. Since there are various security implications for some of the storage mechanisms, the safest and best way is using a credential manager or keystore. If that is not possible, however, we now know that we can easily set our credentials for at least a period of time to avoid having to re-enter them each time a critical command is issued.

Additional Resources:

MAC:
http://tech.lds.org/wiki/Git_Credential_Caching_on_Mac_OS_X

Linux [Ubuntu]
https://askubuntu.com/questions/740183/store-git-credentials-permanently-and-encrypted-using-a-keystore-in-ubuntu

Notes