clearvision

Module 8

Tagging

git

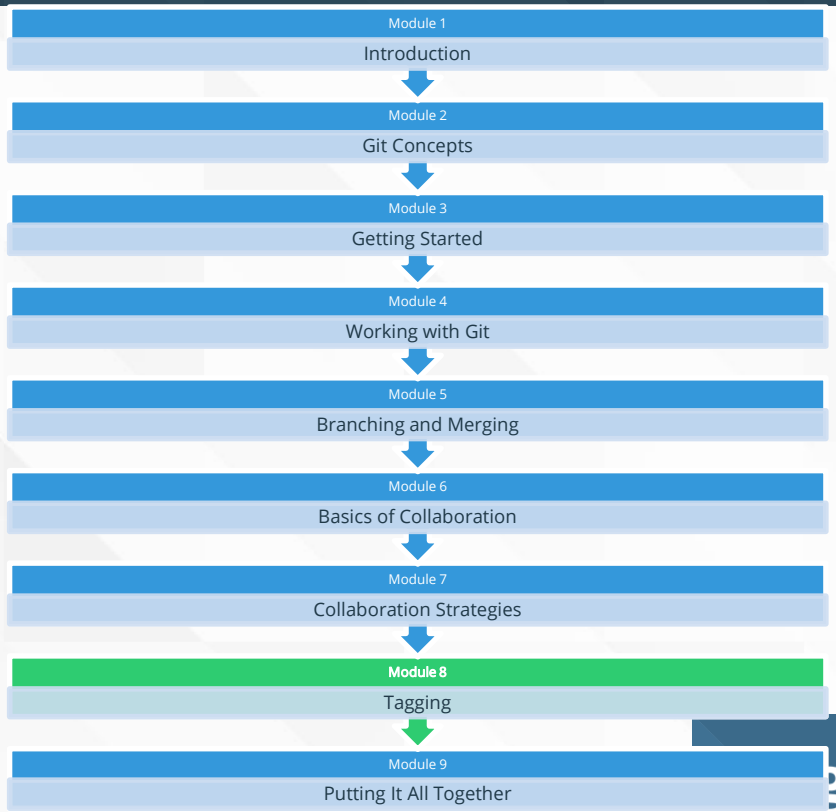
 |

Module Contents

clearvision

Module Contents

- What Is A Tag
- Viewing And Creating Tags
- Signing And Verifying Tags
- Tagging Later
- Sharing Tags



Tags

clearvision

Whether its called a Label a Baseline or a Tag, this underlying concept is common to most version control systems:

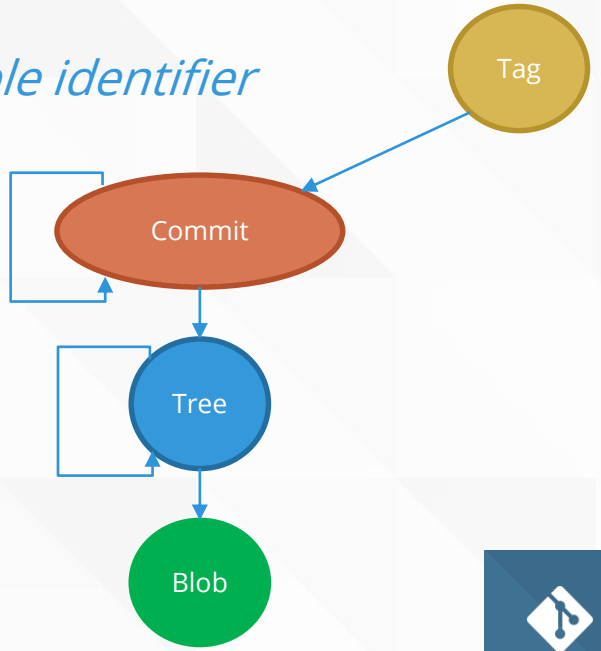
To flag a specific set of files as important with a simple identifier

Git uses tag objects to achieve this.

A tag can provide that identifier:

- Release Version: Rel_1.2, Rel_1.3
- Status Indicator: Tested, Draft, Signed Off

The tag can be used to refer to the commit instead of the SHA-1 value



git

Viewing Tags

clearvision

To view all existing git tags you can use the *git tag* command.

- Use the *git tag -l* command to filter the results.

- To see details of an individual tags use *git show <tag>*

- A file is created for each tag and it stores the SHA-1 value of the commit it is labelling.

git

Creating Tags

clearvision

There are two main types of tags that you can create in git, 'annotated' and 'lightweight'.

- Lightweight

  A light weight tag stores the commit SHA-1 of the current HEAD, in a file with no other information - like a branch that doesn't move.

git tag Rel_1.0

git

Creating Tags

clearvision

- Annotated

An annotated tag is a full git object stored in the object database. It too references the SHA-1 of the current HEAD but stores additional information: tagger name, email, date and a tag message and has its own SHA-1 value.

git tag -a Rel_1.0 -m "First release“

- When you create an annotated tag, you also need to supply a tag message (just like a commit message)

git

Signed Tags

clearvision

A more formal version of the annotated tag is a signed tag.

This uses a GPG (Gnu Privacy Guard) signature to sign the tag, to enable verification of the commit by others.

To sign a tag use the command *git tag -s <tagname>*

To see the details of a signed tag use the command *git show <tagname>*

Verifying Signed Tags

clearvision

In order to verify a signed tag you'll need that users public key in your keyring. (so they'll need to publish it)

You can then use the command *git tag –v <tagname>.*

git

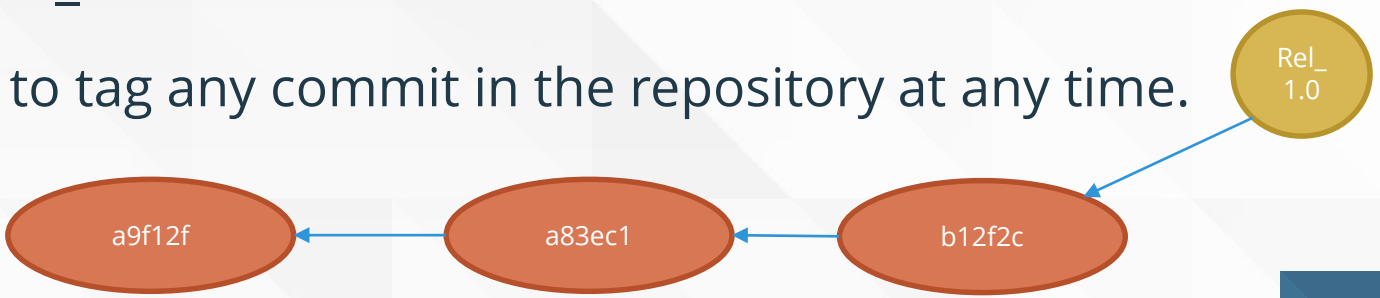
Advanced Tagging

clearvision

So far we have seen that the tag command just tags the current HEAD commit.
You can tag a commit you have moved past by adding the SHA-1 of the commit object at the end of the tag command.

- git tag -a Rel_1.0 -m "First release" a9f12f

This allows you to tag any commit in the repository at any time.



git

Sharing Tags

clearvision

By default git does not share tags when you move commits between repositories.

You must explicitly tell Git to share tags when you move commits, by using the *--tags* option or the tag name with the git push command.

git push origin <tagname>
git push --tags
git pull --tags

They are automatically copied from a remote when you clone

git

Tagging

clearvision

Tags are immutable objects, meaning their content cannot be changed once created.

You can however rename or remove a tag :

Git tag -f <old_tag_name> <new_tag_name>
Git tag -d <tag_name>

git

clearvision

Want to know more about Git?

Additional Reading

git

 |

More about Git

clearvision

- Want to know more about Git?
  - WikiPedia has a good article
    - http://en.wikipedia.org/wiki/Git_%28software%29
  - The Official Git Web Site and documrntaiong
    - http://git-scm.com/
  - Books/References
    - Pro Git
      - By Scott Chacon
    - Ry's Friendly Guide to Git
      - by Ryan Hodson (24 Jan 2012) [Kindle]
    - Pragmatic Version Control Using Git: 1 (Pragmatic Starter Kit)
      - by Travis Swicegood (4 Jan 2009)

git

Online Tutorials

clearvision

- Free Tutorials :
  - git-scm.com - http://git-scm.com/docs/gittutorial
  - Atlassian - https://www.atlassian.com/git/tutorial
  - Codeschool - https://try.github.io/
  - Vogella - http://www.vogella.com/tutorials/Git/article.html
- Additional Git training from Clearvision :
  - Git advanced
  - Atlassian Stash
  - Egit for Git
  - Smartgit for Git

git

clearvision

Lab Exercise

git

 |

clearvision

GIT Fundamentals
The End!

Please email any feedback, comments or questions to sales@clearvision-cm.com

Or use the following link to access our 5 minute online survey :

https://www.surveymonkey.com/s/training_clearvision

git