Using GIT

sachin*

October 26, 2016

1 Theory

- Git uses SHA-1 hash(of a content itself)
  - 40 hex characters
  - 20 bytes

2 Install

2.1 using package manager

on Ubuntu or debian based system

```
sudo apt-get install/update git
```

on RHEL or Fedora based systems

```
sudo yum install -y git-core
```

2.2 Manual install

- Dependencies:

  ```
  yum install libcurl-devel perl-Tk-devel
  ```

3 Usage

3.1 Elementary usage

- Create git enables repository

  ```
  git init MyRepo
  ```

- Check the status of project

  ```
  git status
  ```

- Add file(s) to `git`
  Add single file.

  ```
  git add <FILENAME>
  ```

  Add everything

  ```
  git add .
  ```

  Add all files from current directory

  ```
  git add *
  ```

*iclcoolster@gmail.com

Add notes

```
git notes add
```

Remove(Unstage file). This is used when you add file(s) using `git add` and now you want to undo your action. This is removing the file from git's staging area.

```
git reset HEAD <FILENAME>
```

- Make `commit`

  With one line message

  ```
  git commit -m "MESSAGE"
  ```

  Open default text editor to write descriptive message. You can set the default editor using =git config –global core.editor "vim"=

  ```
  git commit
  ```

  Add and commit at the same time

  ```
  git commit -am "MESSAGE"
  ```

- View logs(commits)

  ```
  git log
  ```

- View `diff` between unstage file(s)

  Entire `diff`

  ```
  git diff
  ```

  File specific `diff`

  ```
  git diff <FILENAME>
  ```

  Show only file names in git diff

  ```
  git diff --name-only HASH1..HASH2
  ```

  Compare file between two commits

  ```
  git diff HASH1..HASH2 -- </PATH/TO/FILE>
  ```

- Show commit hash specific state of a file.

  ```
  git show <COMMIT HASH>:</PATH/TO/FILE>
  ```

- Reset to previous commit

  ```
  git reset --hard <COMMIT HASH>
  ```

3.2 Intermediate usage

- Branching

  Create branch

  ```
  git branch <BRANCH NAME>
  ```

  Delete branch

  ```
  git branch -d <BRANCH NAME>
  ```

  Change branch

  ```
  git checkout <BRANCH NAME>
  ```

  Merge branch

  ```
  git merge <BRANCH NAME>
  ```

3.3 Remotes

- Manage git remotes

  View remote host

  ```
  git remote -v
  ```

  Add remote

  ```
  git remote add <REMOTE NAME> <REMOTE URL>
  ```

  Example:

  ```
  git remote add origin https://github.com/iitbaakash/project1.git
  ```

  Remove remote

  ```
  git remote remove <REMOTE NAME>
  ```

  Example

  ```
  git remote remove origin
  ```

  Create local branch from remote

  ```
  git clone -b ics https://github.com/androportal/abt.git
  git checkout -b holo_theme remote/origin/holo_theme
  ```

3.4 Push/Pull

- Manage push/pull

  Push source for the first time

  ```
  git push -u <REMOTE NAME> <BRANCH NAME>
  ```

  and then

  ```
  git push
  ```

  Pull source from remote(assuming that the remote is already been added)

  ```
  git pull
  ```

- Push Notes:

  ```
  git push <remote> refs/notes/*
  ```

3.5 etal

- tags

  ```
  # Annotate with message
  git tag -a v1.1 -m "tag message"
  ```

  or

  ```
  # Will pop text editor to write message
  git tag -a v1.1
  ```

- show tags

  ```
  git tags
  ```

- Push tags

  ```
  git push origin --tags
  ```

- Fetch and merge

```
git fetch origin master
git log ..origin/master
git merge <commit hash of fetched commit>
```

- Apply a patch

```
patch -p1 < ~/experiment.patch
```

3.6 Hosting

- webserver

```
cd project
git instaweb --httpd=apache2
git instaweb --stop
```

- generate a patch between two branch

```
git diff master..experiment > experiment.patch
```

- Host git repos on (local)different machine
  - On machine A(192.168.1.11):

Assume that directory 'github' contain all your git repositories:

```
cd github
git daemon --reuseaddr \
        --base-path=. \
        --export-all \
        --verbose
```

If you want to host a single repo:

```
cd github/insert_shebang
git daemon --reuseaddr \
        --base-path=. \
        --export-all \
        --verbose
```

- On machine B(192.168.1.12):

```
git clone git://192.168.1.11/insert_shebang

# WORKING EXAMPLE
git clone sachin@10.118.248.42:/home/sachin/github/apc-rock-II-kernel

# for single repo hosting
git clone git://192.168.1.11/ insert_shebang
# Note the SPACE between the address and repository name)
```

3.7 Test

- git blame

```
git blame -f <branch> <path/to/the/file>

# (scan lines from 1 upto 20)
git blame -f <branch> -L 1,20 <path/to/the/file>

# -C: Code copied or moved to another files
git blame -C -f <branch> <path/to/the/file>
```

- TODO: git bisect

3.8 format-patch

- Create patch for last three commits in directory 'patches/'

  ```
  git format-patch -o patches/ -3 HEAD
  ```

- Create patch for last three commits starting form specific commit hash(say c0101f0c)

  ```
  git format-patch -o patches/ -3 c0101f0c
  ```

- Alter subject prefix line(By default it is [PATCH n/m])

  ```
  git format-patch --subject-prefix="RFC" -o patches/ -3
  ```

3.9 git send-email

- Install Net::SMTP::SSL

  ```
  sudo -H cpan Net::SMTP::SSL
  sudo -H cpan MIME::Base64
  sudo -H cpan Authen::SASL
  ```

- send email to

  ```
  git send-email --to <EMAIL_ID> patches/*.patch
  ```

- Do not create patches but send directly

  ```
  git send-email -3 --to=<EMAIL_ID>
  ```

3.10 git-shell

- Create user

  ```
  adduser --create-home \
                  --skel /dev/null \
                  --home-dir /home/git \
                  --shell /usr/bin/git-shell git
  ```

  –skel: Directory having skeleton files like .bashrc, .profile etc.(refer `man useradd`)

  Set password

  ```
  passwd git
  ```

- Try to login via as user git

  ```
  su - git
  ```

  Should get an error:

  ```
  fatal: Interactive git shell is not enabled.
  hint: ~/git-shell-commands should exist and have read and execute access.
  ```

  for more info refer: GIT_{SRC}/contrib/git-shell-commands/README and http://planzero.org/blog/2012/10/24/hosting_an_admin-friendly_git_server_with_git-shell

4 Advance

4.1 *Commitish, Hashish*

```
# TODO: Heavy commenting needed
git init newproject
cd newproject/
tree .git
git status
echo "Hello World" > hello.txt
git add hello.txt
```

```
tree .git
git commit -m "First hello"
tree .git
printf "blob 12\000Hello World\n" | shasum
echo "Hello World" | git hash-object -w --stdin
alias deflate="perl -MCompress::Zlib -e 'undef $/; print uncompress(<>)'"
deflate .git/objects/55/7db03de997c86a4a028e1ebd3a1ceb225be238
tree .git
git update-index --add \
        --cacheinfo 100644 \
        557db03de997c86a4a028e1ebd3a1ceb225be238 \
        hello.txt
tree .git
git log -1
git cat-file -t 5fc7c3f9
git cat-file -p 5fc7c3f9
git cat-file -t 97b4
git cat-file -p 97b4
cd ..
cp -r newproject newprojectdamaged
cd newprojectdamaged/
# open .git/objects/5f/c7c3f9d5ba98390244cec5ff0675672867f4a2
# and add '1' at the very begining
git status
git fsck
cd ../newproject
git br feature1
git br
~/bin/generaterandomchanges 2 master txt
git co feature1
~/bin/generaterandomchanges 2 feature1 txt
git merge-base master feature1
git co master
git merge feature1
git rev-parse 5fc7c3
# git full hash. This is run at the begining of every command with
# hash
git tag IJUSTLEFTMYCOMPUTERFORCOFFEE21
~/bin/generaterandomchanges 2 sample2 txt
git describe HEAD
git tag -a IJUSTLEFTMYCOMPUTERFORCOFFEE22
git gc
# garbage collect
tree .git/objects/
git tag NORMALTAG
tree .git/objects/
# no change
git tag -a SAMPLEWITHANNOTATED
tree .git/objects/
git cat-file -t 891c
# a tag
git cat-file -p 891c
git rev-parse SAMPLEWITHANNOTATED
git rev-parse IJUSTLEFTMYCOMPUTERFORCOFFEE21
git cat-file -p 3f244
git cat-file -t 3f244
git cat-file -t 891c
git rev-parse 'master^{tree}'
git rev-parse HEAD
git rev-parse HEAD^
```

```
git rev-parse master^
git log
git rev-parse 'master^{tree}'
git cat-file -t fe31
git cat-file -p fe31
git log
git cat-file -p 3ce0
git describe HEAD
```

5 Used cases

5.1 A word about –amend

```
git commit -m "Fix #1: grep & awk to rescue"
git commit --amend -m "Fix #2: grep & awk to rescue"

git push origin master

Username for 'https://github.com': psachin
Password for 'https://psachin@github.com':
To https://github.com/psachin/bash_scripts.git
 ! [rejected]                master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/psachin/bash_scripts.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Behind the scenes

```
git log -2

commit db87208c1cd6ca49af0dbed028573a71045b9780
Author: sachin <iclcoolster@gmail.com>
Date:   24 minutes ago

        Fix #2: grep & awk to rescue

commit 232b805b8812412156da655d240dd952d5c8375e
Author: sachin <iclcoolster@gmail.com>
Date:   6 days ago

        Better to have a function: mplayer-eq

git log -2 origin/master


commit ec692611b83489f4393491eadc3e05821afe1be9
Author: sachin <iclcoolster@gmail.com>
Date:   25 minutes ago

        Fix #1: grep & awk to rescue

commit 232b805b8812412156da655d240dd952d5c8375e
Author: sachin <iclcoolster@gmail.com>
Date:   6 days ago

        Better to have a function: mplayer-eq

git pull origin master
```

```
From https://github.com/psachin/bash_scripts
* branch                      master      -> FETCH_HEAD
Merge made by the 'recursive' strategy.
```

```
git log -2
```

```
commit bc7bbb3068f907cca1460a7c4289038d87d0ec36
Merge: db87208c1cd6 ec692611b834
Author: sachin <iclcoolster@gmail.com>
Date:   54 seconds ago

        Merge branch 'master' of https://github.com/psachin/bash_scripts

commit db87208c1cd6ca49af0dbed028573a71045b9780
Author: sachin <iclcoolster@gmail.com>
Date:   3 hours ago

        Fix #2: grep & awk to rescue

commit ec692611b83489f4393491eadc3e05821afe1be9
Author: sachin <iclcoolster@gmail.com>
Date:   3 hours ago

        Fix #1: grep & awk to rescue

commit 232b805b8812412156da655d240dd952d5c8375e
Author: sachin <iclcoolster@gmail.com>
Date:   6 days ago

        Better to have a function: mplayer-eq
```

```
git log -4 origin/master(OR origin/HEAD
```

```
commit bc7bbb3068f907cca1460a7c4289038d87d0ec36
Merge: db87208c1cd6 ec692611b834
Author: sachin <iclcoolster@gmail.com>
Date:   7 minutes ago

        Merge branch 'master' of https://github.com/psachin/bash_scripts

commit db87208c1cd6ca49af0dbed028573a71045b9780
Author: sachin <iclcoolster@gmail.com>
Date:   3 hours ago

        Fix #2: grep & awk to rescue

commit ec692611b83489f4393491eadc3e05821afe1be9
Author: sachin <iclcoolster@gmail.com>
Date:   3 hours ago

        Fix #1: grep & awk to rescue

commit 232b805b8812412156da655d240dd952d5c8375e
Author: sachin <iclcoolster@gmail.com>
Date:   6 days ago

        Better to have a function: mplayer-eq
```

```
git push origin master
```