Distributed Source Version Control System

Apr 2013

Who am i

Lâm Phương Duy

Software Architect

duylam@kms-technology.com

- Use Git from 2009
- http://vn.linkedin.com/in/duylamphuong

Objectives

- To use Git in software project as doing with SVN or TFS
- To self study Git for advanced needs

Syllabus

- Git theory
- Daily development workflow
- Daily collaboration workflow
- More on Git

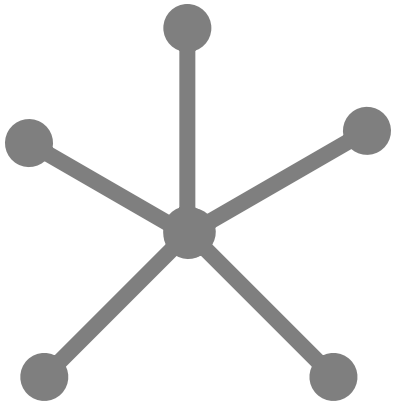
Git’s history

Source Version Control in Linux kernel project

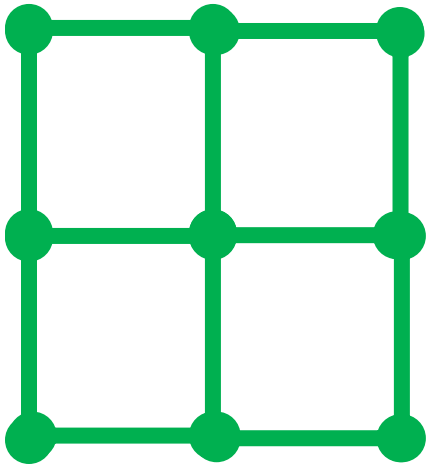
Syllabus

- Git theory
- Daily development workflow
- Daily collaboration workflow
- More on Git

Version Control Systems



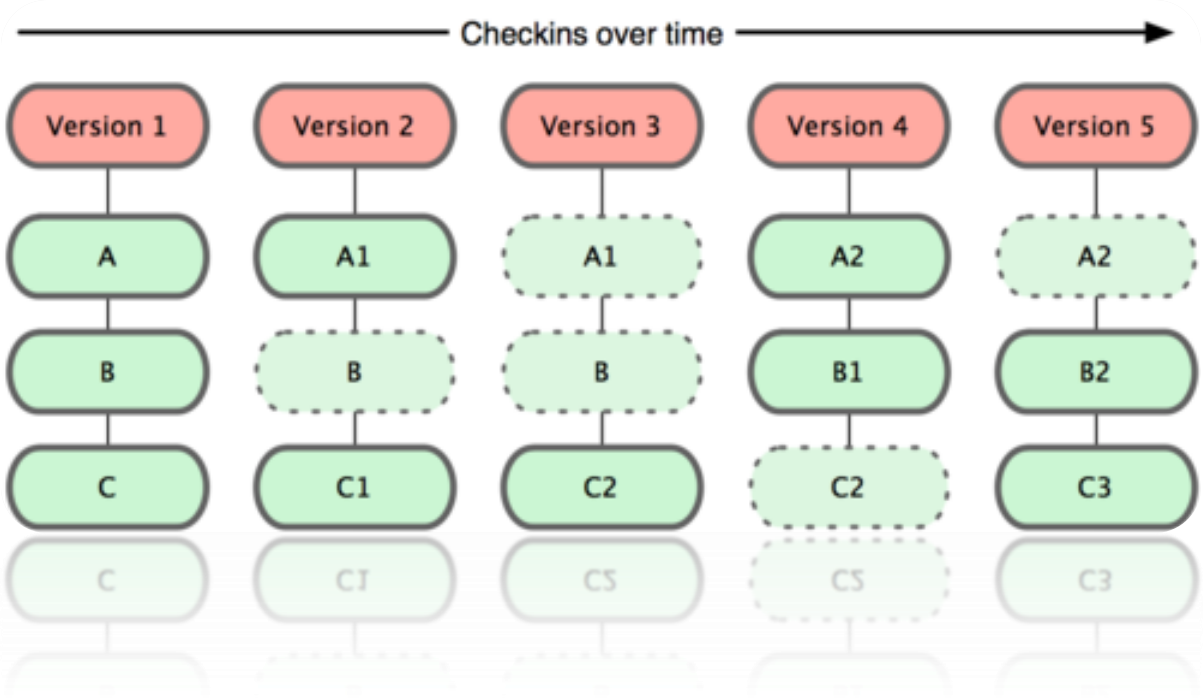
Centralized Version Control Systems



Distributed Version Control Systems

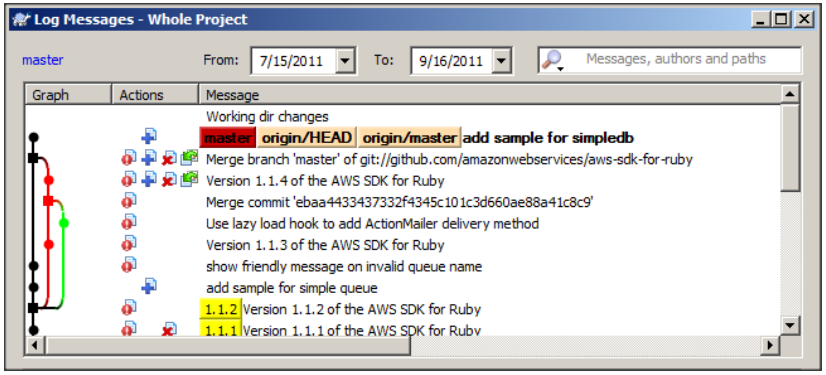
Git theory

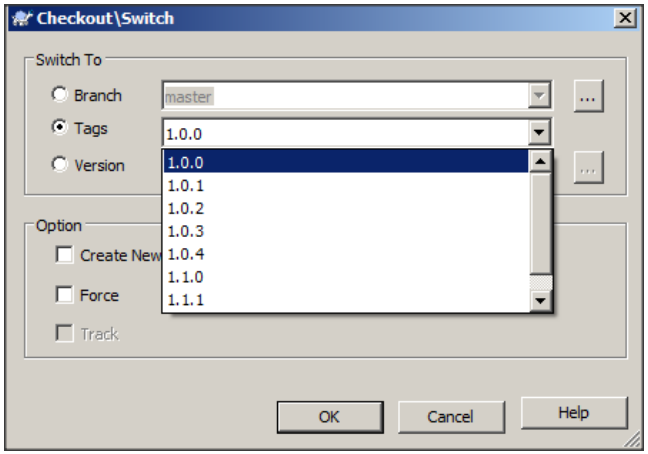
- Data = Snapshot
- No network
- Three states

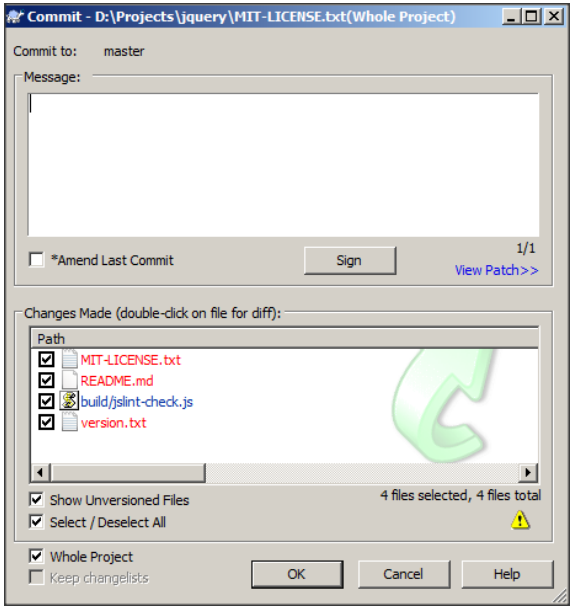


Git theory

- Data = Snapshot
- No network
- Three states

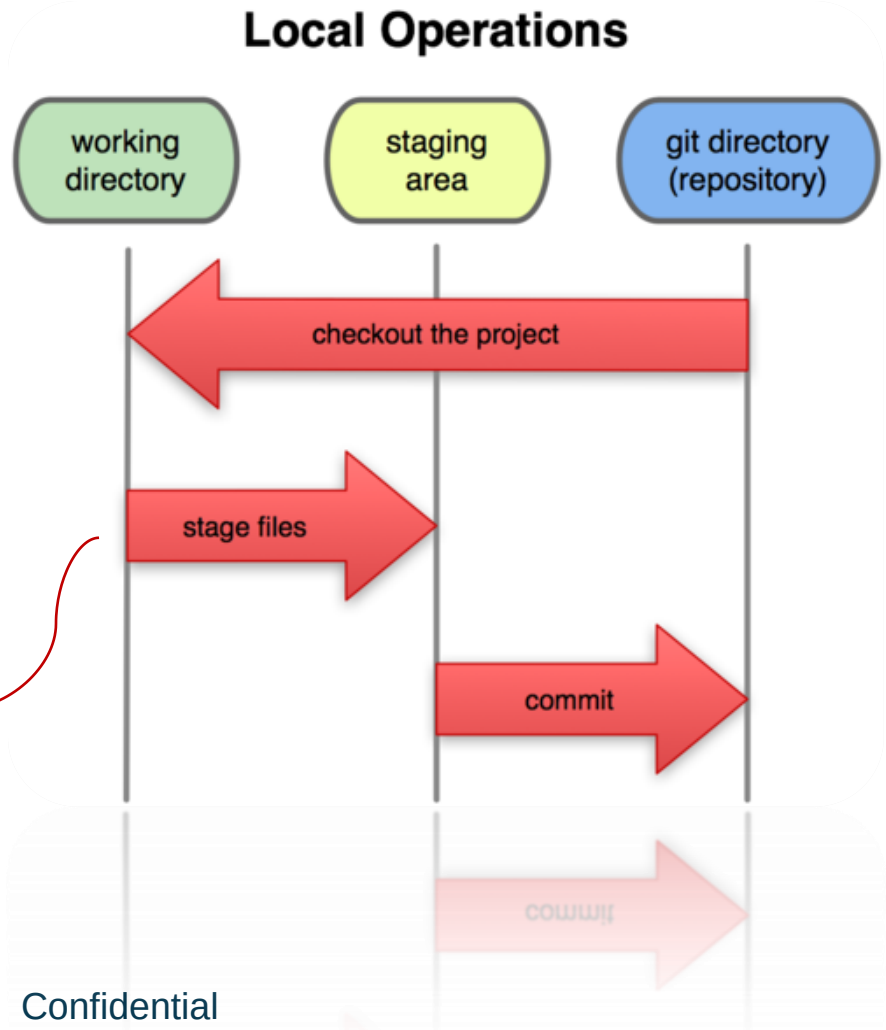






Git theory

- Data = Snapshot
- No network
- Three states

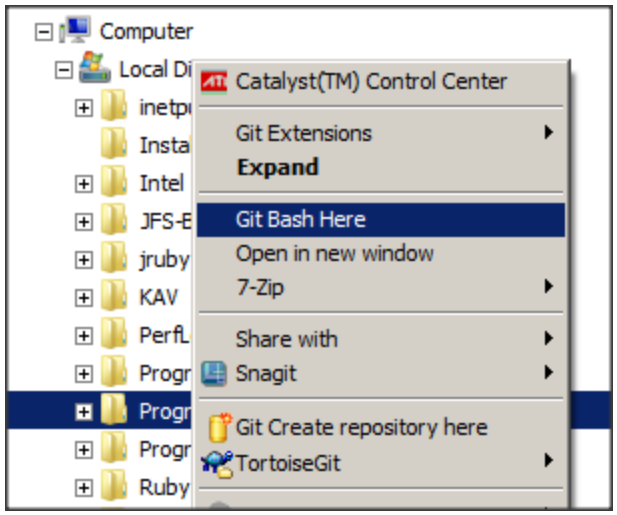


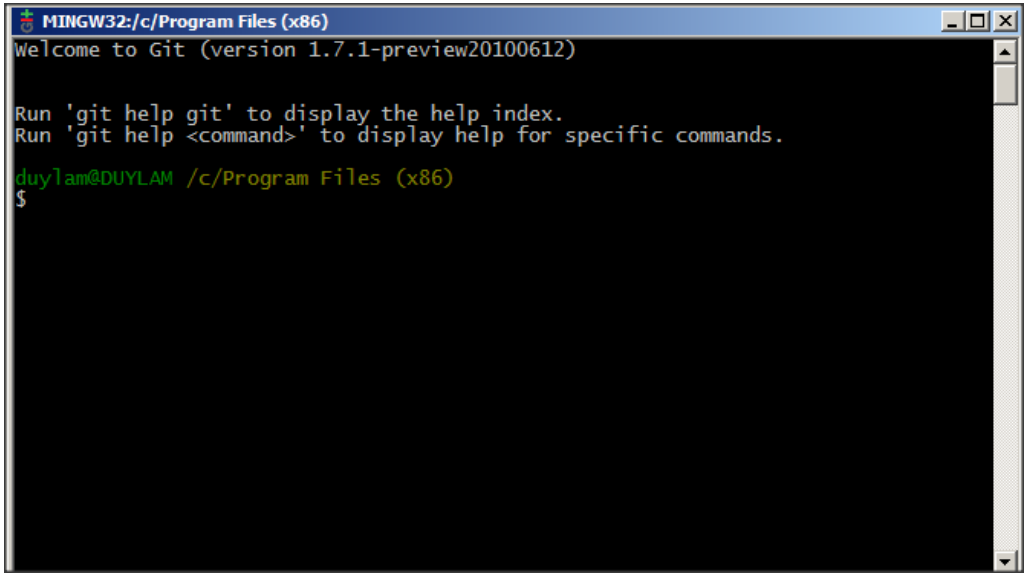

```
git add
git rm
git status
```

Syllabus

- Git theory
- Daily development workflow
- Daily collaboration workflow
- More on Git

Launch Git shell





Git in daily development

```
git init
git clone
```

Setup repository

```
git commit
```

Change repository

```
git log
git status
git diff
```

Check repository

```
git checkout
git reset
```

Undoing

```
git add    git checkout
git rm     git mv
```

Update staging area

Git remote url protocols

Local	/data/git/project.git
SSH	user@server:project.git
Git	git://server/project.git
HTTP	http://server/project.git

Remote repository

```
git remote add <url>
git clone
```

Get remote repository

```
git fetch
git pull
git push
```

Get updates

Authenticate with remote repository

1) Generate key files with `ssh-keygen`
2) Upload `%USER_HOME%/.ssh/id_rsa.pub` to remote repository hosting

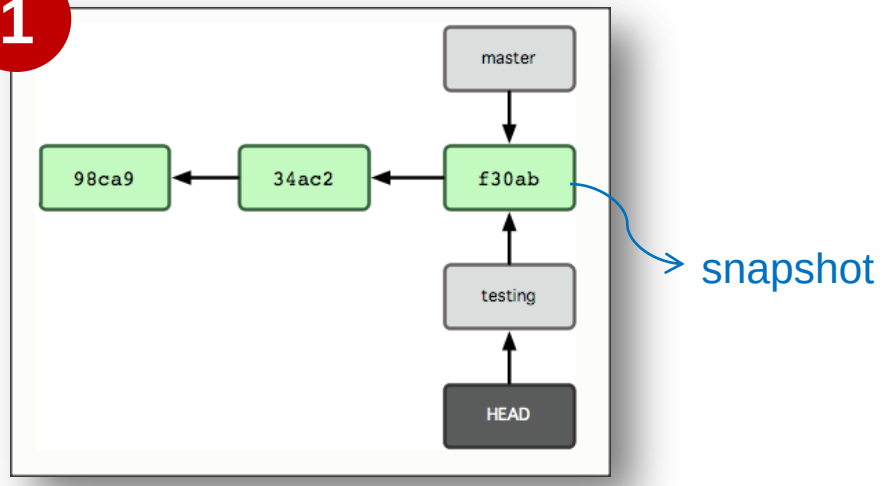
Read more: https://help.github.com/articles/generating-ssh-keys

Syllabus

- Git theory
- Daily development workflow
- Daily collaboration workflow
- More on Git

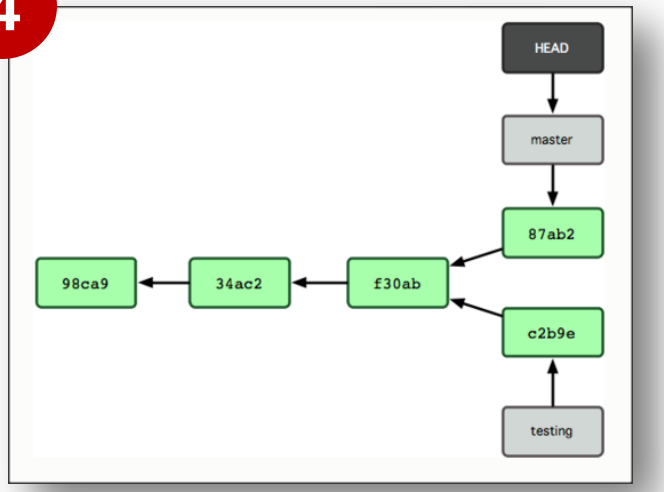
Branch (`git branch`)

1

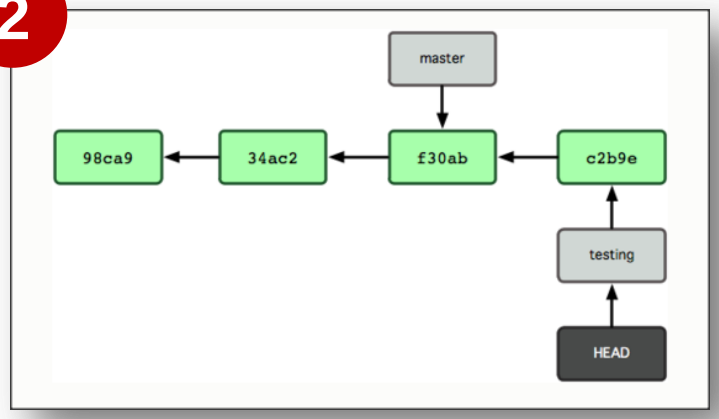


snapshot

4

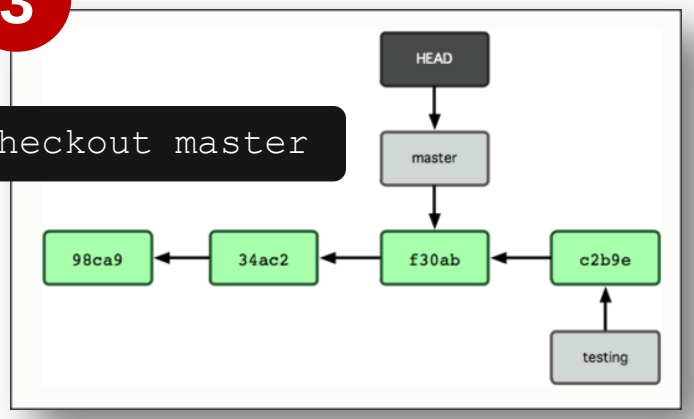


2

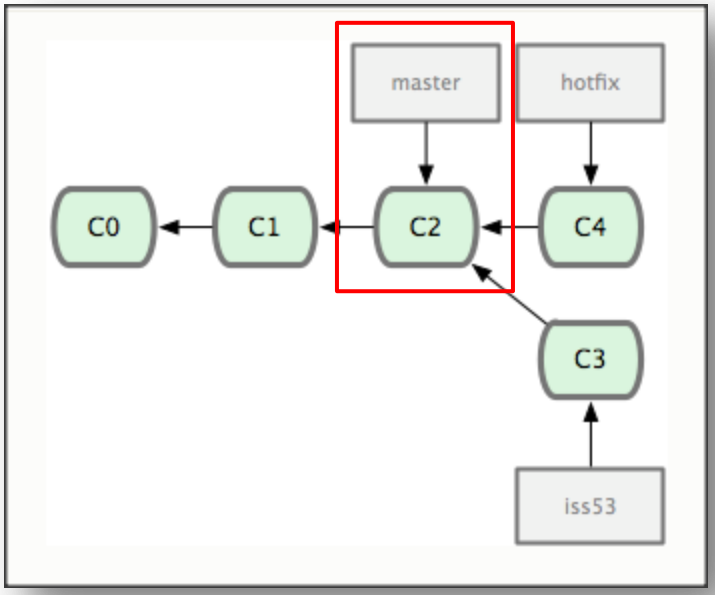


3

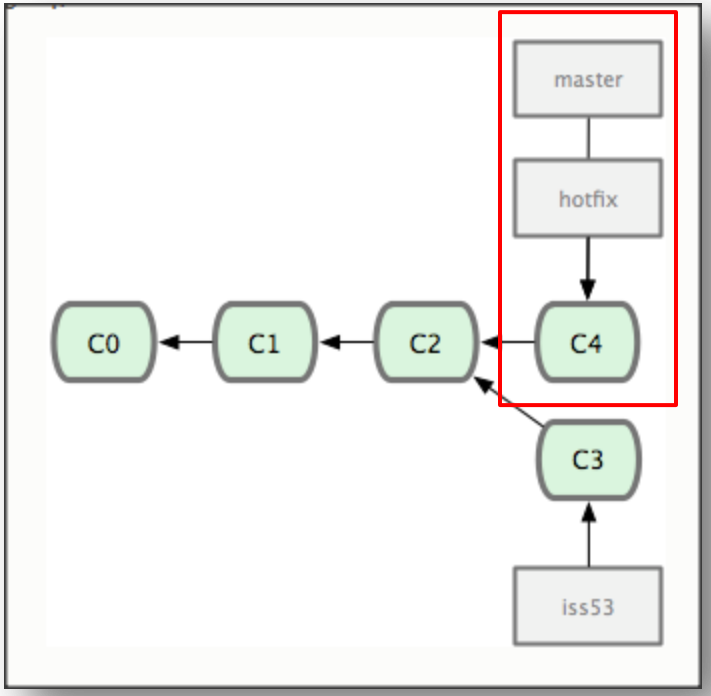
```
git checkout master
```



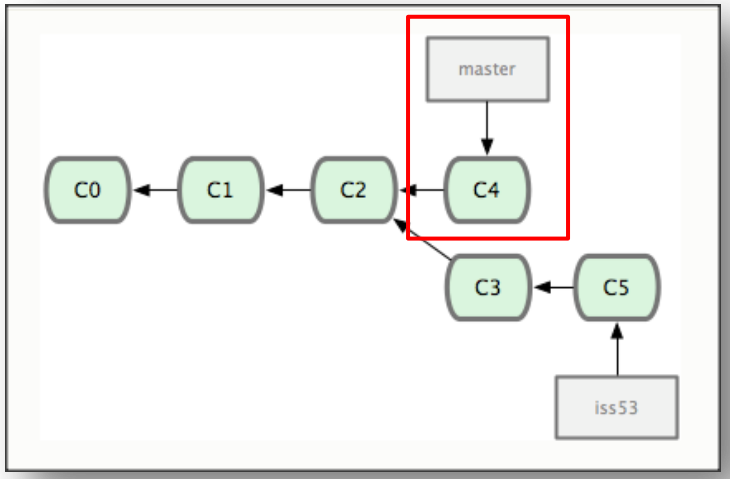
Merging - Fast forward




```
git merge hotfix
```

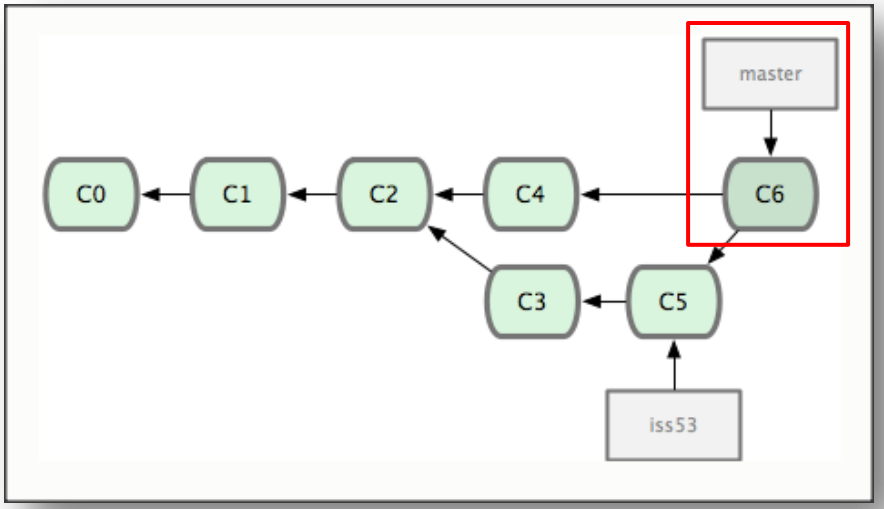


Merging - Non fast forward




```
git merge iss53
```

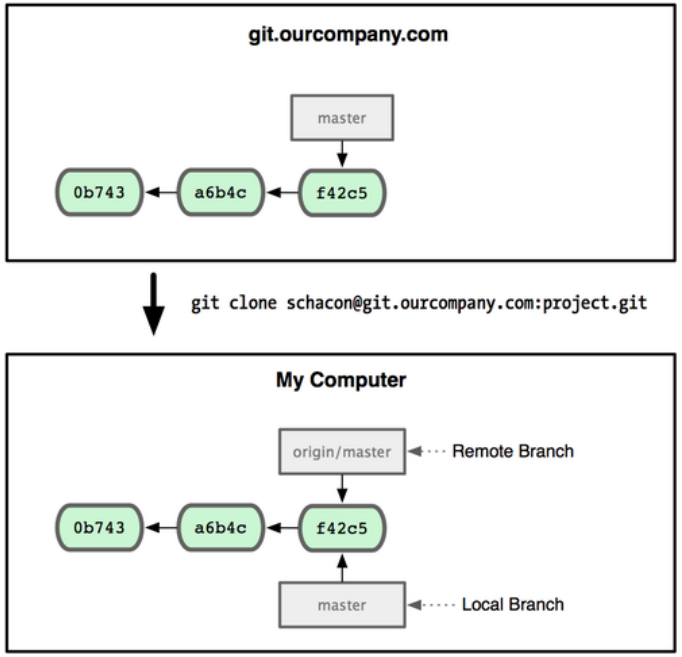
(resolve conflicts if any)

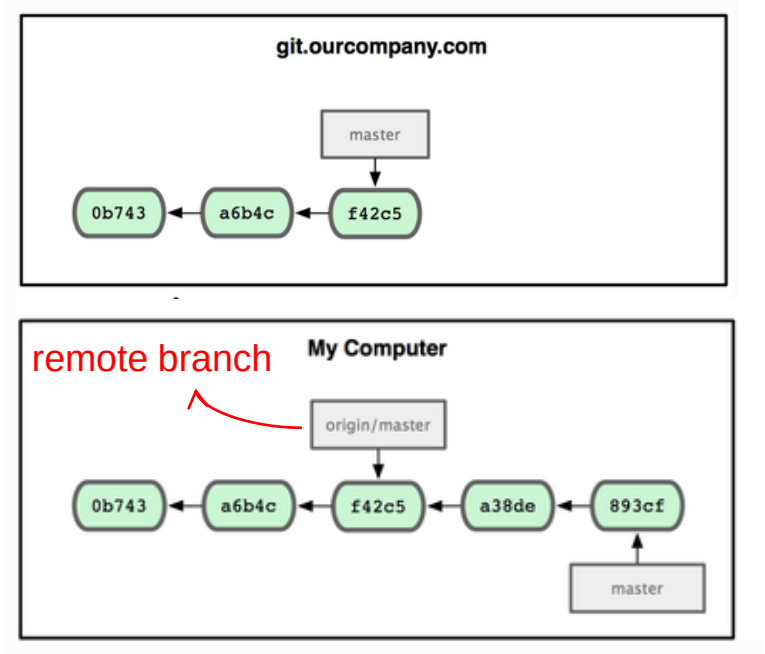


Remote branch workflows

- Create default remote branch
- Get changes from remote repository
- Merge changes from remote branch in local (two methods)
- Update changes to remote repository

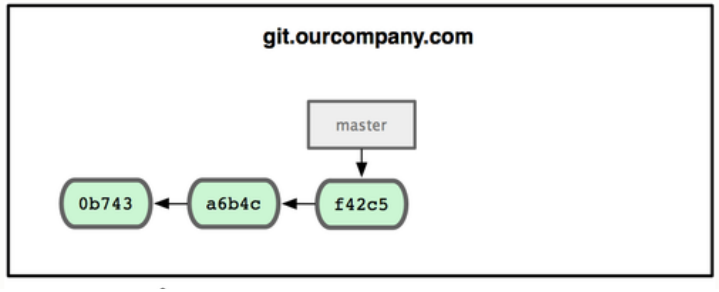
Create default remote branch

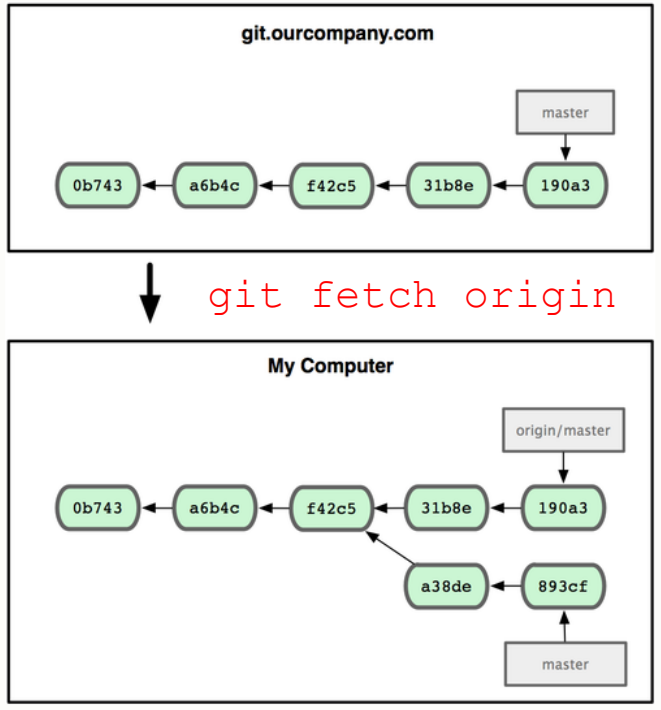




Time

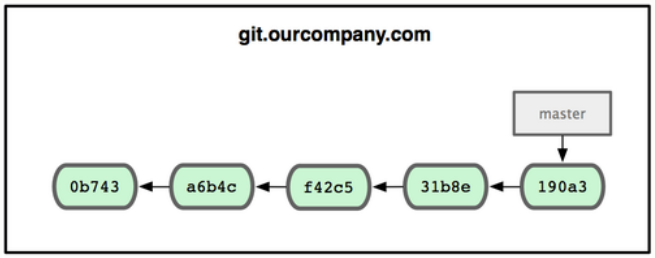
Get changes from remote repository

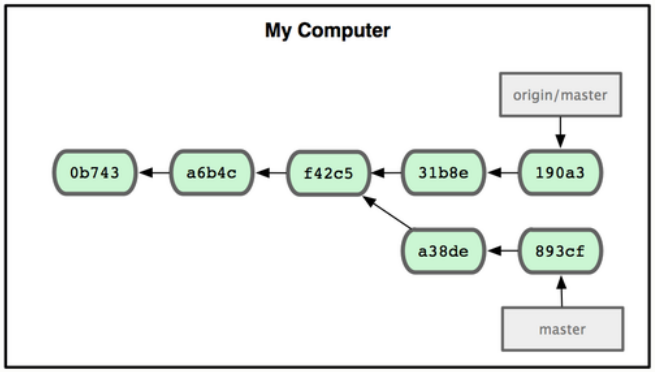


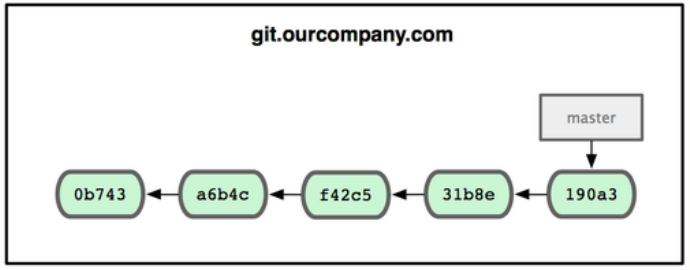


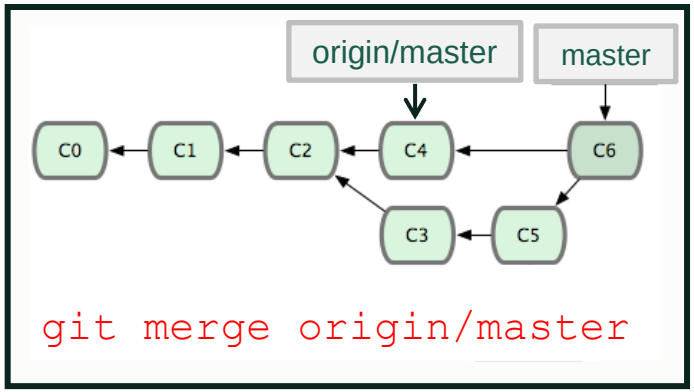
Time

Merge changes from remote branch in local (v.1)



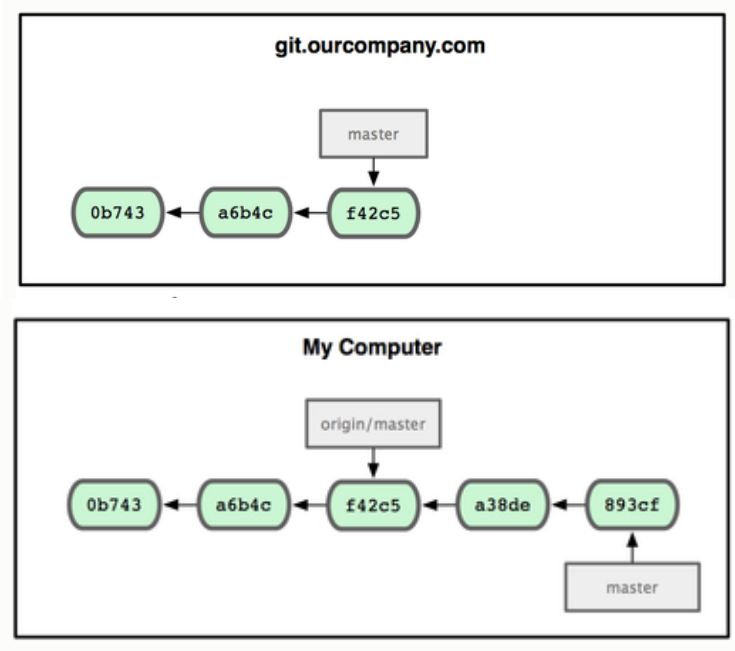




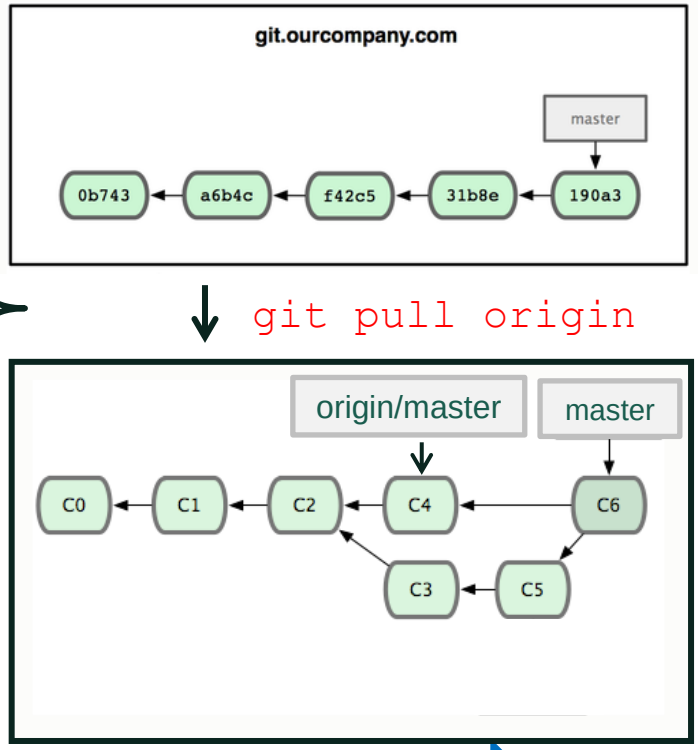


Time

Merge changes from remote branch in local (v.2)

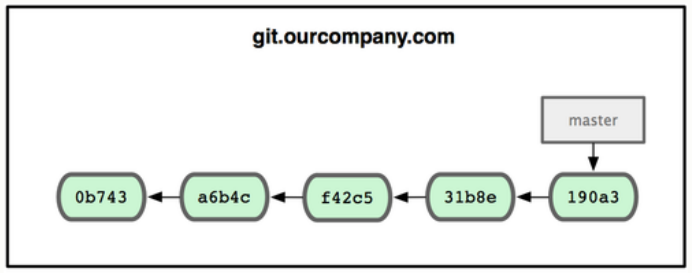


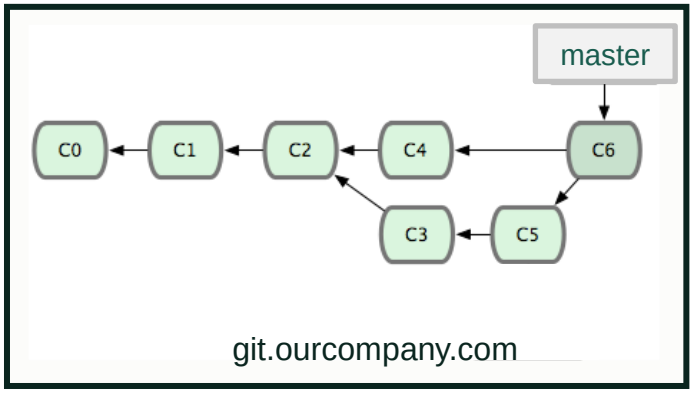

```
git fetch
    +
git merge
```

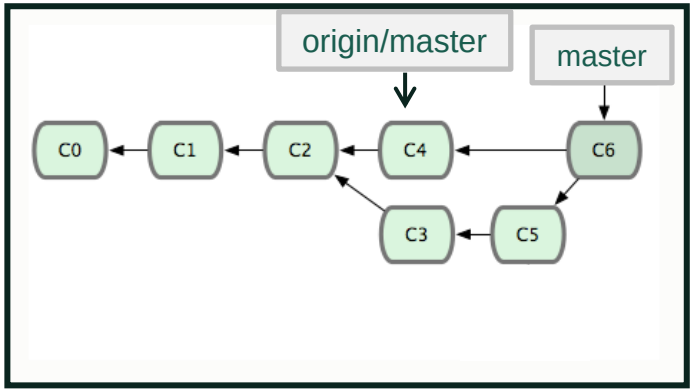


Time

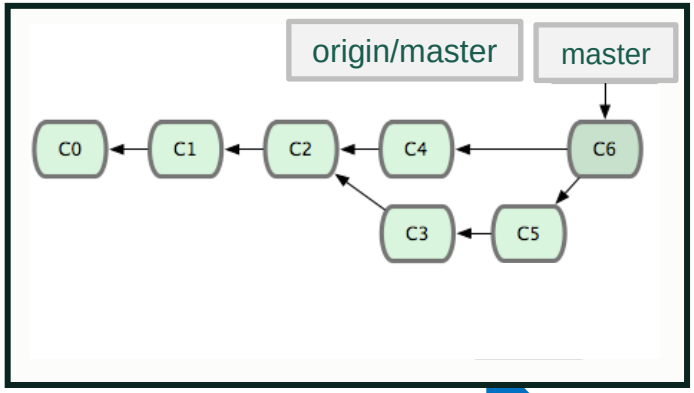
Update changes to remote repository







git push origin/master



Time

Remote tracking branch

- Use tracking branch to let Git know which server and branch to push / pull
- Create remote tracking branch:

```
> git checkout -b [branch] [remote name]/[branch]
```

Syllabus

- Git theory
- Daily development workflow
- Daily collaboration workflow
- More on Git

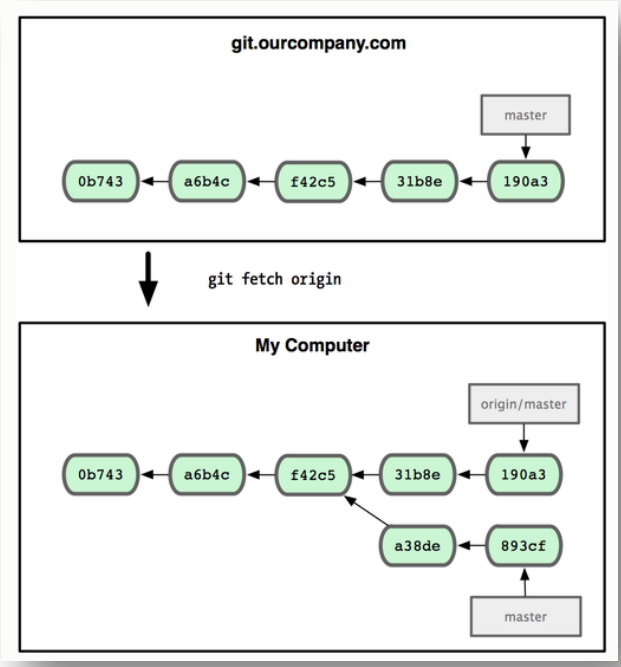
Useful features

```
> git tag

> git stash

> git submodule
```

Common problems

```
> git push
! [rejected]        master -> master (non-fast forward)
Error: failed to push some refs to ‘git@gitproxy:rip747/cfwheels.git’
```

```
> git pull
Merge made by recursive
> git push
To git@gitproxy:rip747/cfwheels.git
1717535..1406e8c  master -> master
```

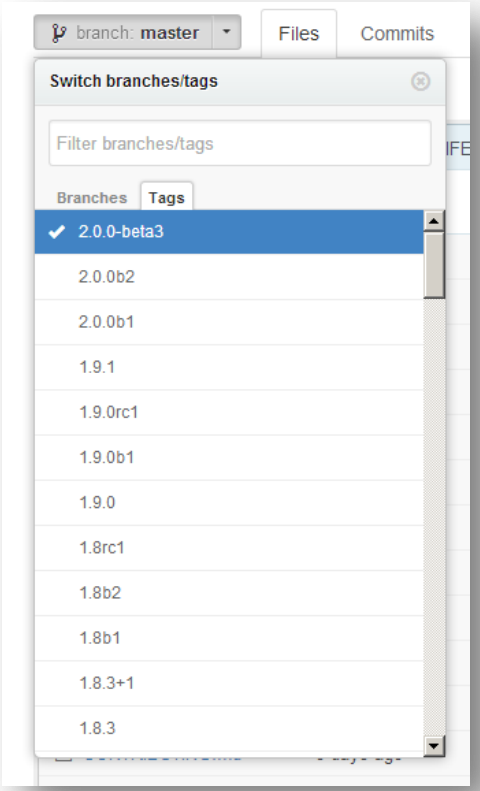


Common problems (cont.)

- To remove remote branch e.g *origin/iss105*

```
> git push origin :iss105
```

- Use `git tag` to mark releases



Reference

For everything you want to read more about Git

http://git-scm.com

THANK YOU