

CSCI4140

Open-Source Software Project Development

Tutorial 1

Git

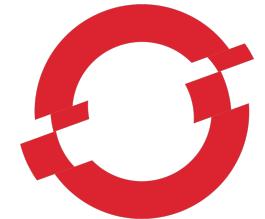
OpenShift

Assignment 1 Preview

15 Jan: Some minor amendment after 11:30 session

Outline

- Git
- OpenShift
- Assignment 1 Preview



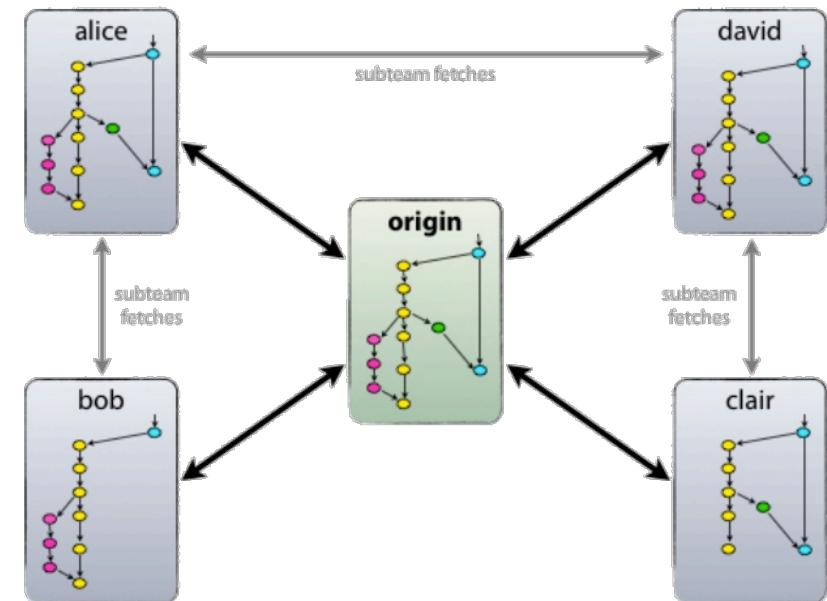
Version Control

- Managing your code
 - Clear picture of development
 - No need to worry about modifying wrong code
 - History of development
 - Compare with / restore old code
 - Coordinate with others



Git

- *Distributed Version Control System*
 - Everyone having a copy of repository in local
 - Communicate (push / pull) with central repository (remote)
 - Flexible for project with multiple developers

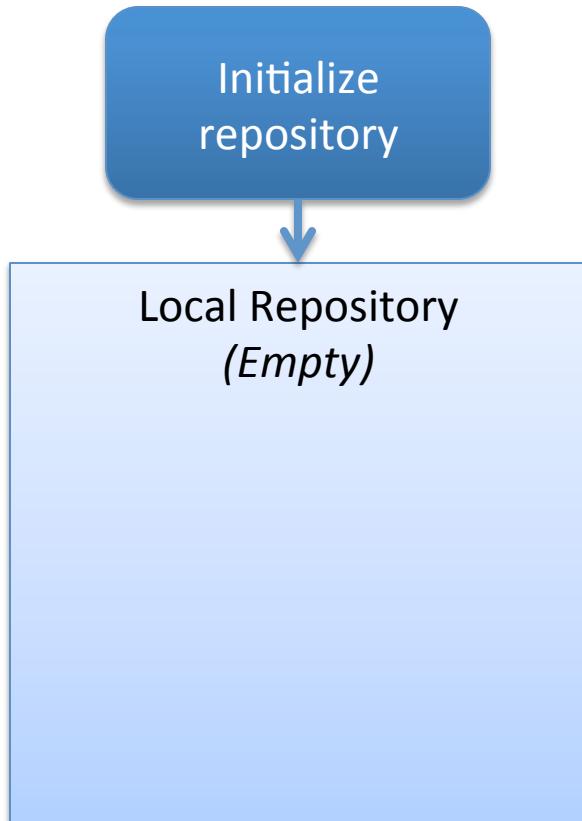




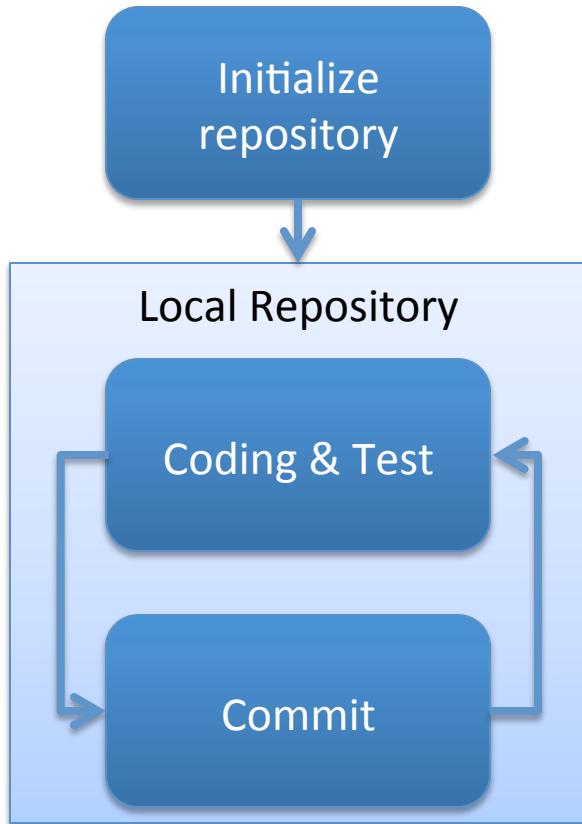
Installing Git

- Mac
 - Comes with command line tools
 - Prompt pop up by typing git in terminal
 - Or install with Homebrew ([brew.sh](#))
`brew install git`
- Ubuntu
`sudo apt-get install git-core`
- Windows
 - Cygwin
 - Refer to supplementary notes

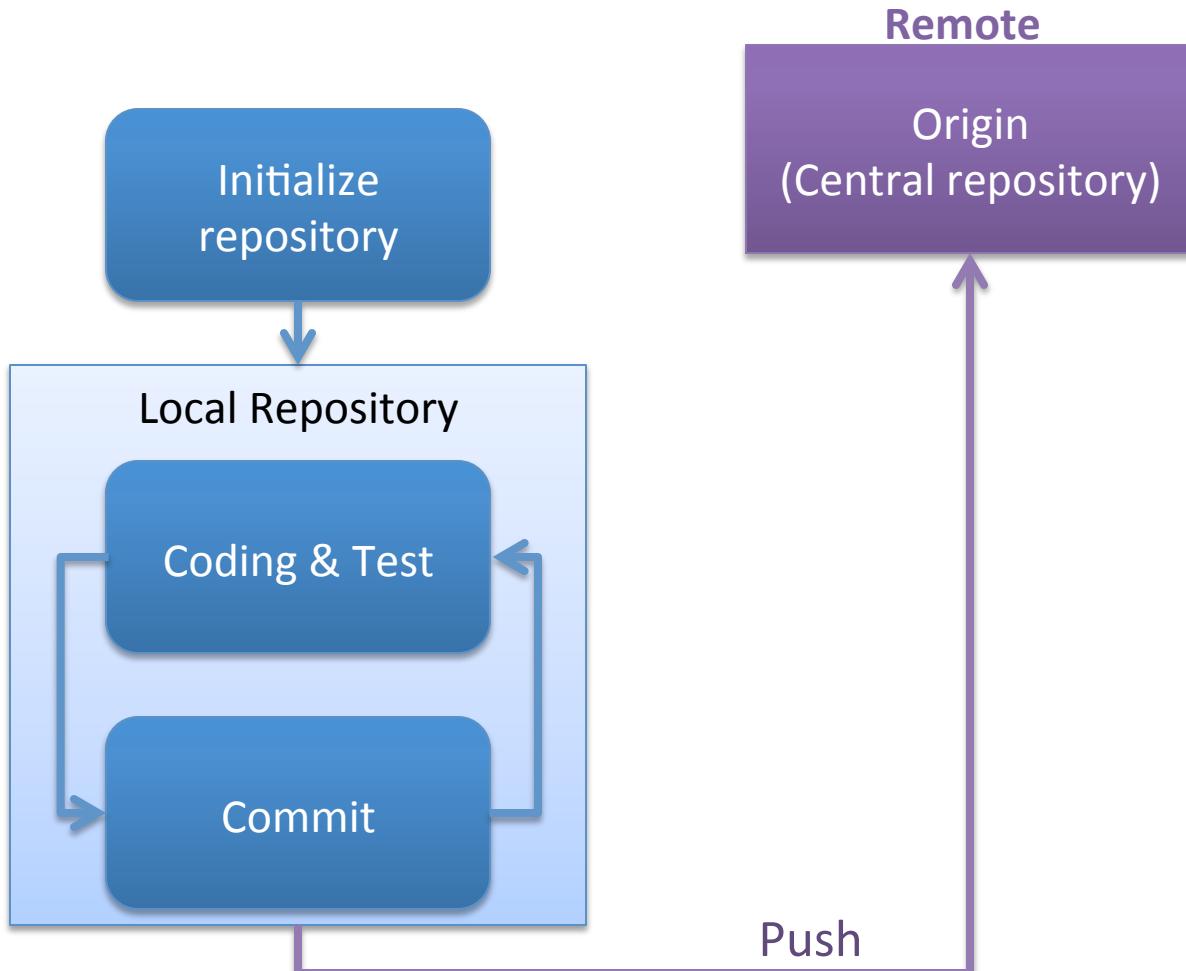
Workflow



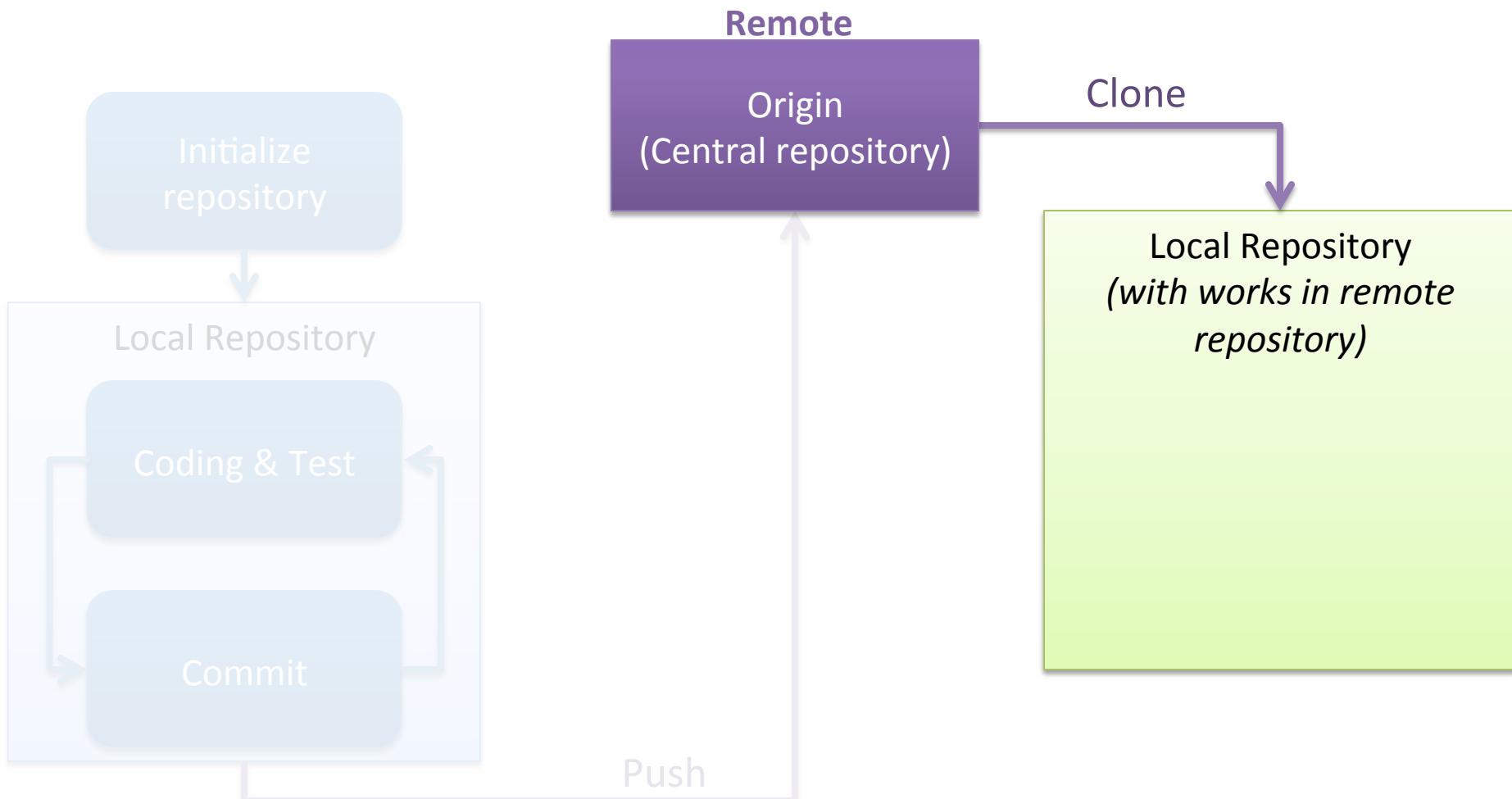
Workflow



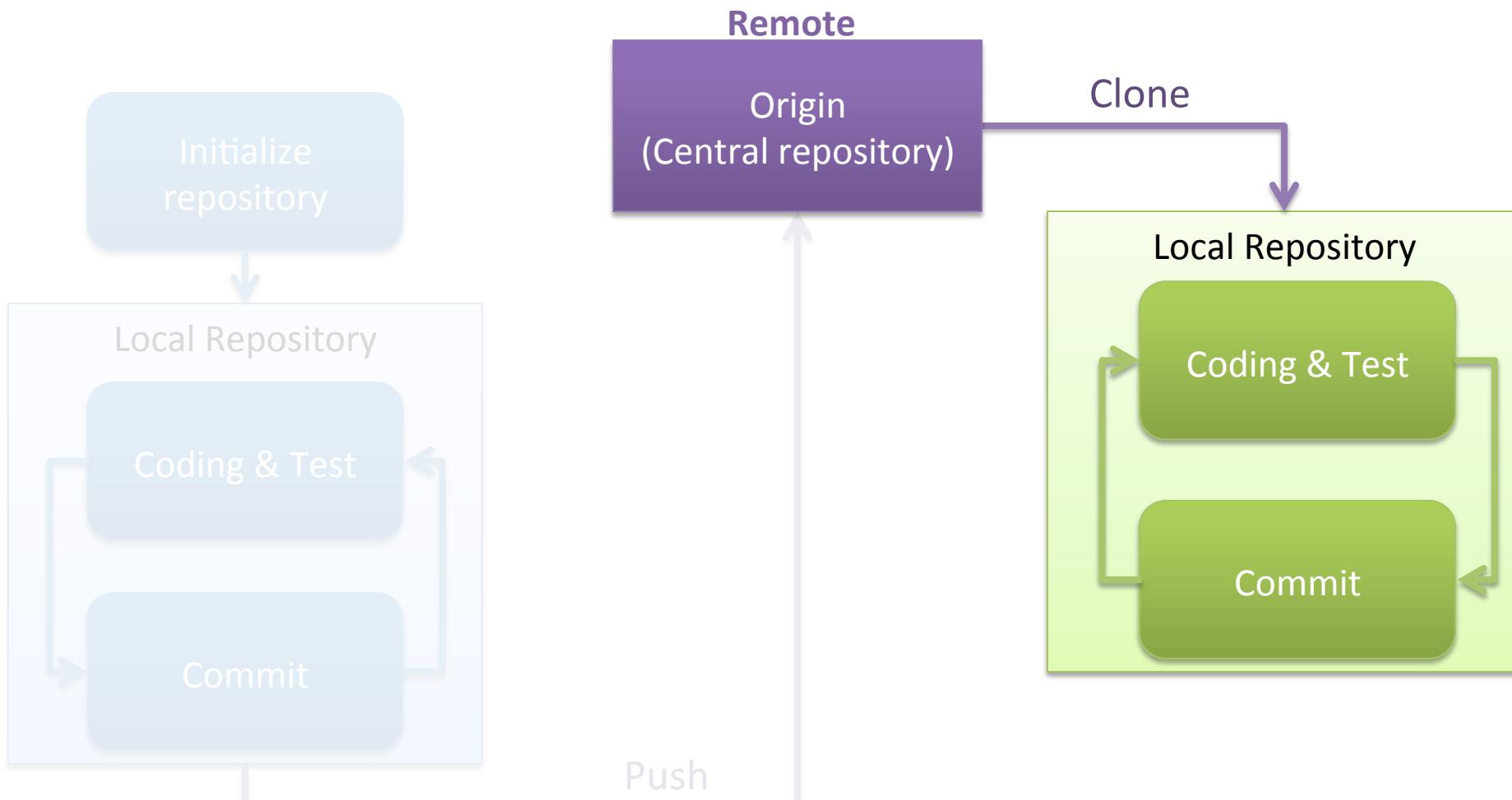
Workflow



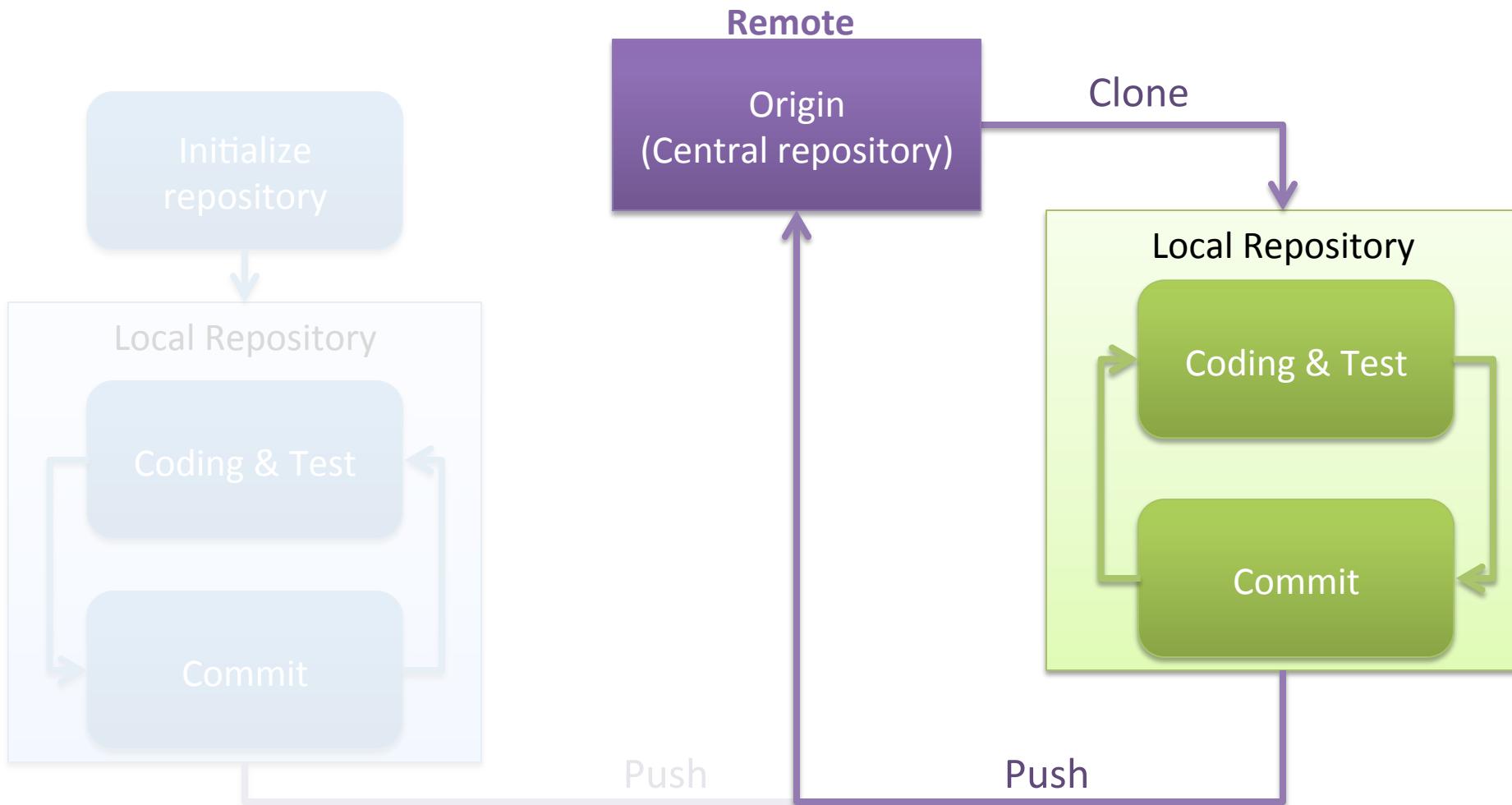
Workflow



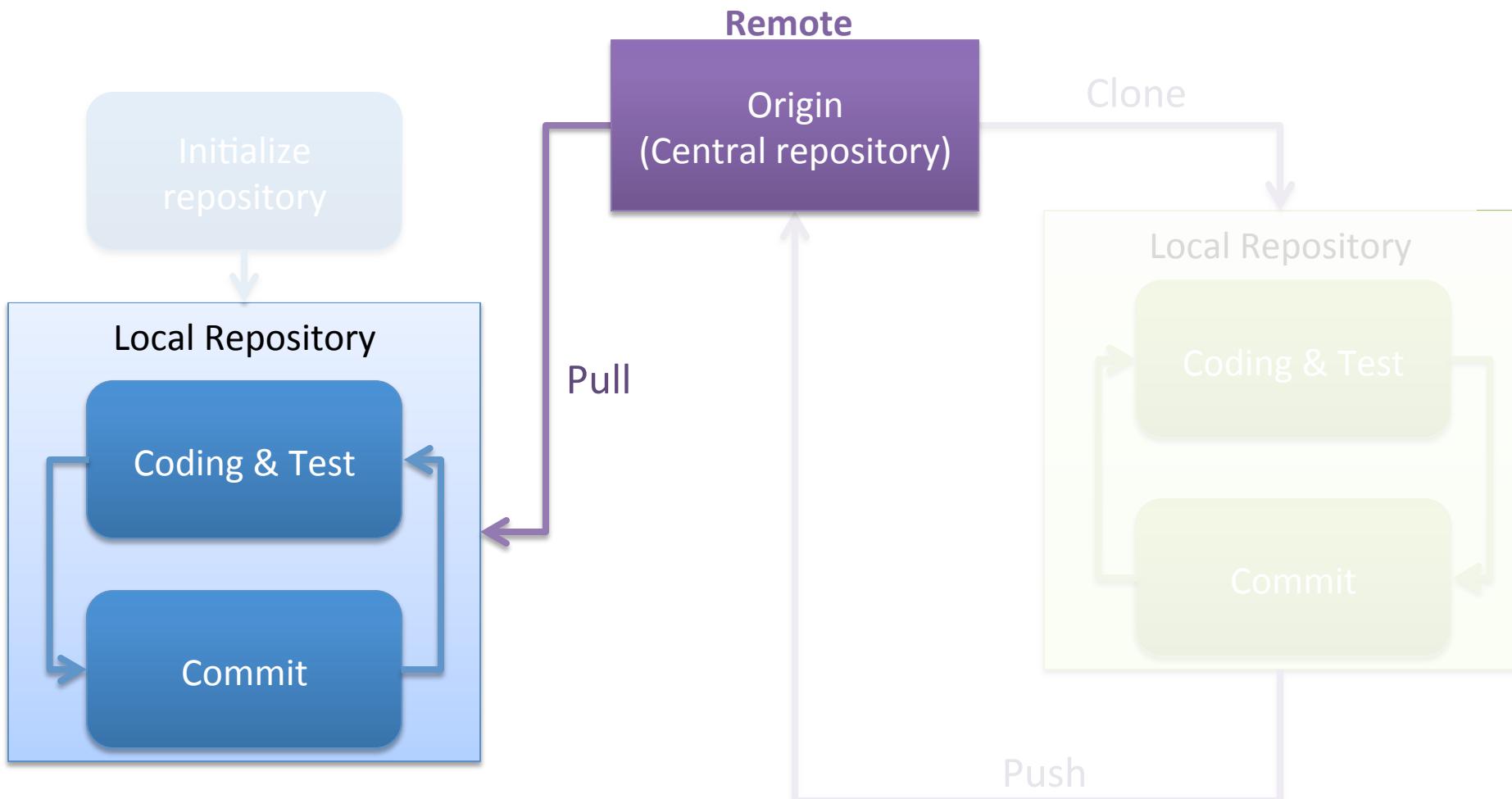
Workflow



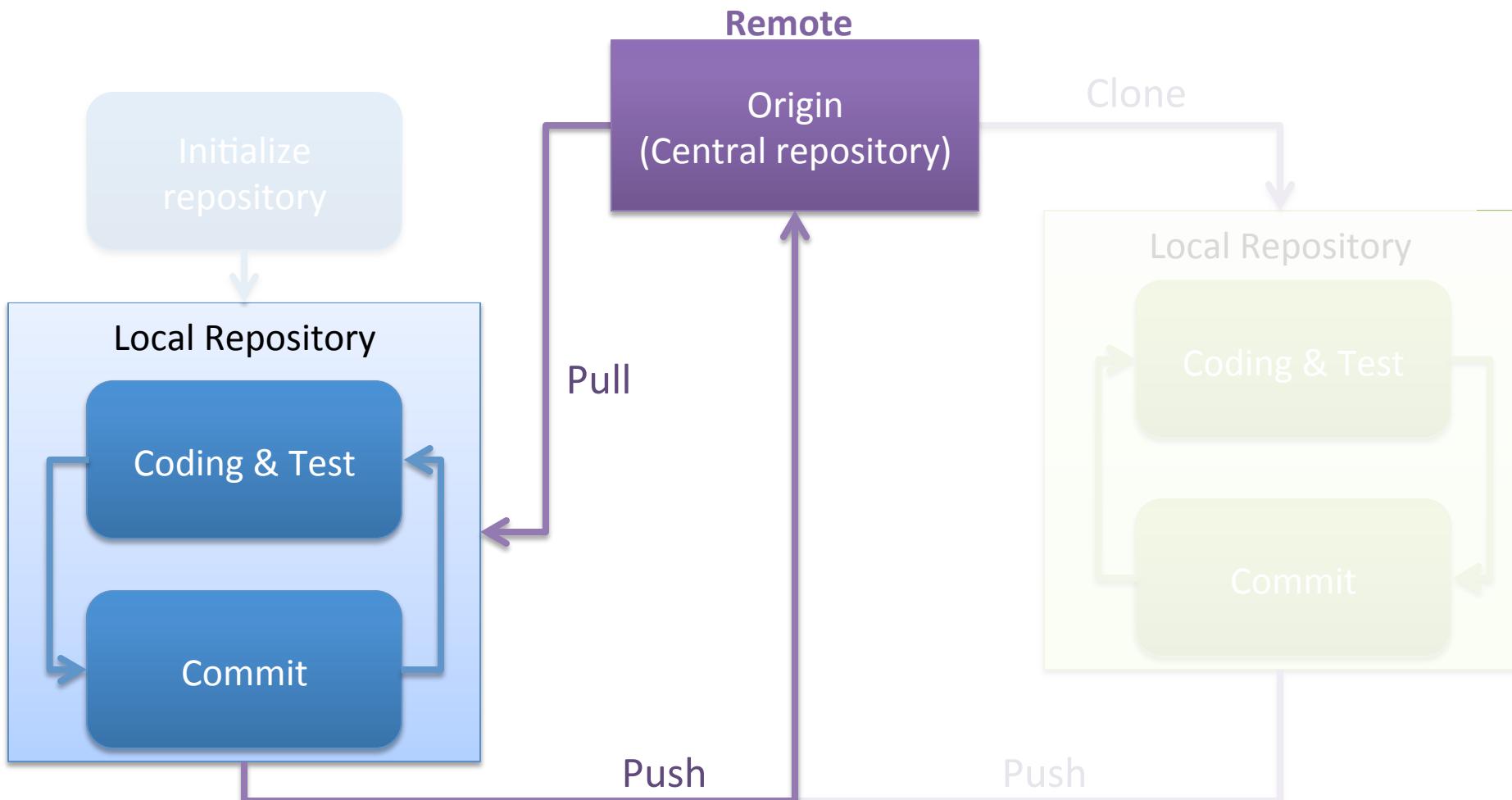
Workflow



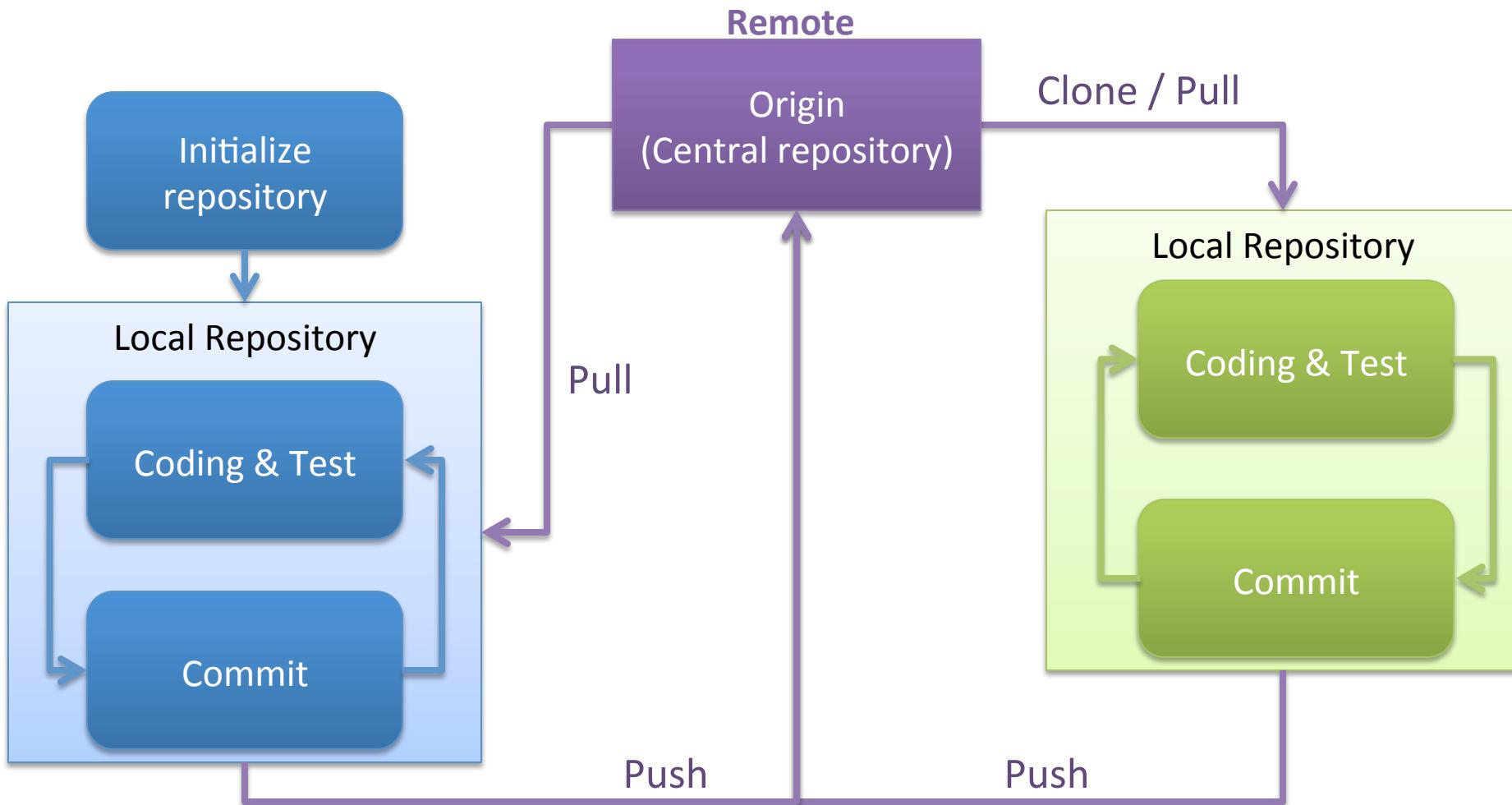
Workflow



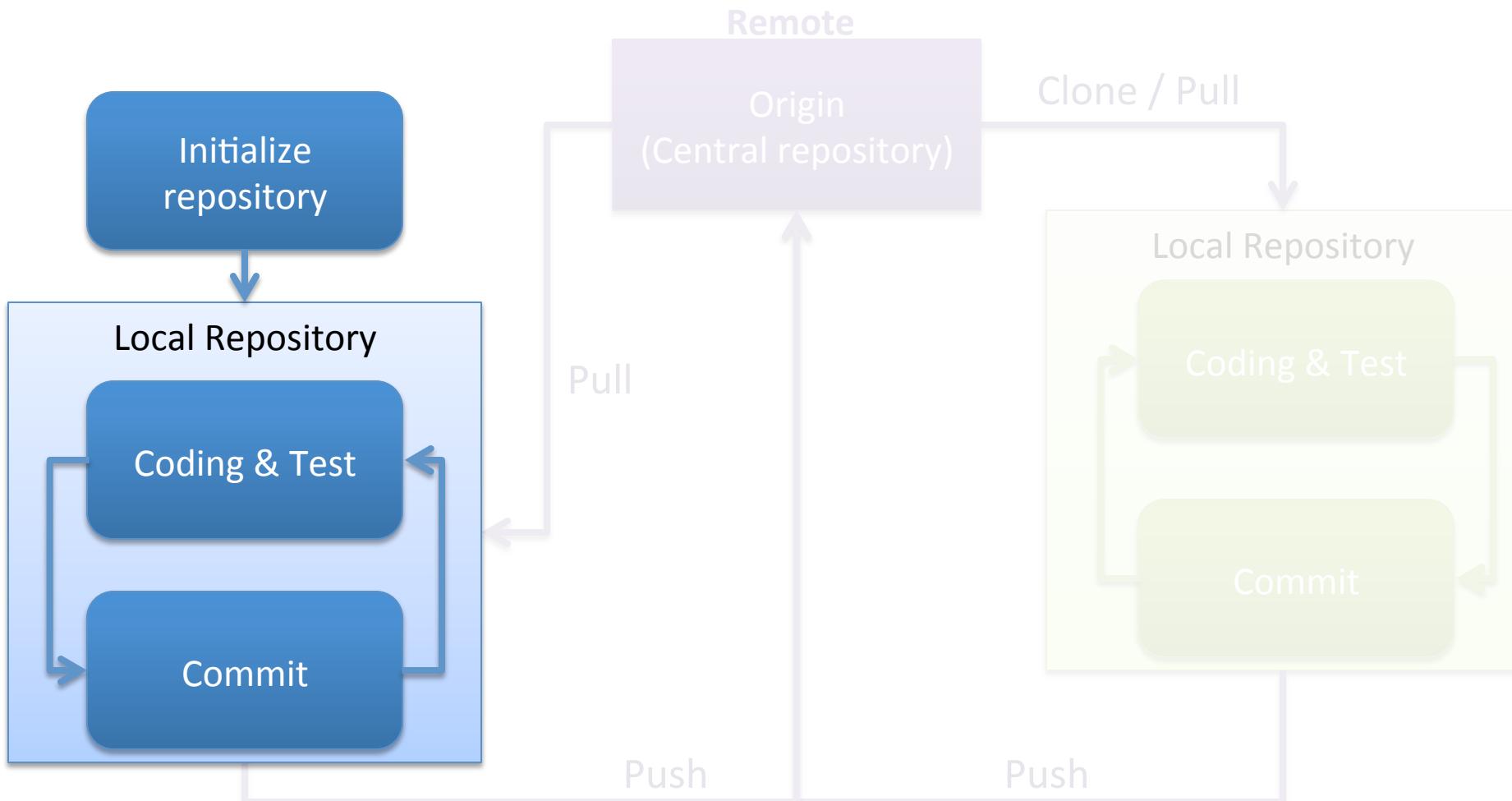
Workflow



Workflow



Working on Local Repository



Init

git init

- Initial git repository
 - At current working directory

```
✓ 17:04:04 jimmy@JimmyMBA ~/git/tmp git init
Initialized empty Git repository in /Users/jimmy/git/tmp/.git/
✓ 17:04:07 jimmy@JimmyMBA ~/git/tmp ¶ master
```

- Or at a new folder

```
✓ 22:19:49 jimmy@JimmyHMac ~/git git init demo
Initialized empty Git repository in /Users/jimmy/git/demo/.git/
```

For assignment: OpenShift initialized for you

Status

git status

- Check status of git repository
 - Just after init, nothing in repository

```
✓ 22:20:11 jimmy@JimmyHMac ~/git > cd demo
✓ 22:20:49 jimmy@JimmyHMac ~/git/demo > master > git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

- Even there are files inside, they are not *tracked* by git

```
✓ 22:21:26 jimmy@JimmyHMac ~/git/demo > master > git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    sosad.c

nothing added to commit but untracked files present (use "git add" to track)
```

Add

git add <file>

- Add file to git repository (index)
 - New file to tracked by git repository

```
✓ 22:21:28 jimmy@JimmyHMac ~/git/demo ↵ master git add sosad.c
✓ 22:22:43 jimmy@JimmyHMac ~/git/demo ↵ master git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   sosad.c
```

- Make modified file become *staged* (ready to commit)

Commit

git commit

- Make *staged* change persistent (as commit)
 - Confirm a change / add of a file
- Commit message
 - Type into text editor get into after `git commit`
 - Or specified by `-m` flag

```
✓ 22:25:18 jimmy@JimmyHMac ~/git/demo ↵ master ➤ git commit -m "sosad"
[master (root-commit) d36ab95] sosad
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 sosad.c
 ✓ 22:25:30 jimmy@JimmyHMac ~/git/demo ↵ master ➤ git status
On branch master
nothing to commit, working directory clean
```

Log

git log

- Log of all previous commit
 - With commit hash shown

```
commit d36ab955db40e6e91b57a6c4bcc2c2ff57bb982a
Author: Jimmy Sinn <jimmysinn91@gmail.com>
Date:   Wed Jan 7 22:25:30 2015 +0800
```

sosad

Commit message

Commit hash key

Author info.
(Change by `git config`)

- Usually only need first 4 characters from commit hash

File Status (1)

- After you work a while ...

Demo:

- Modified sosad.c
- Add Makefile
- Compiled sosad.c to sosad
- Add Makefile to git (`git add`)

- You found more file status from
`git status`

```
✓ 22:29:45 jimmy@JimmyHMac ~/git/demo ✭ master •+ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:  Makefile          Staged

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   sosad.c          Unstaged

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    sosad                         Untracked
```

File Status (2)

	Change not tracked by git	Change tracked by git
Not changed	Untracked	Tracked
Changed from prev. commit	Unstaged	Staged

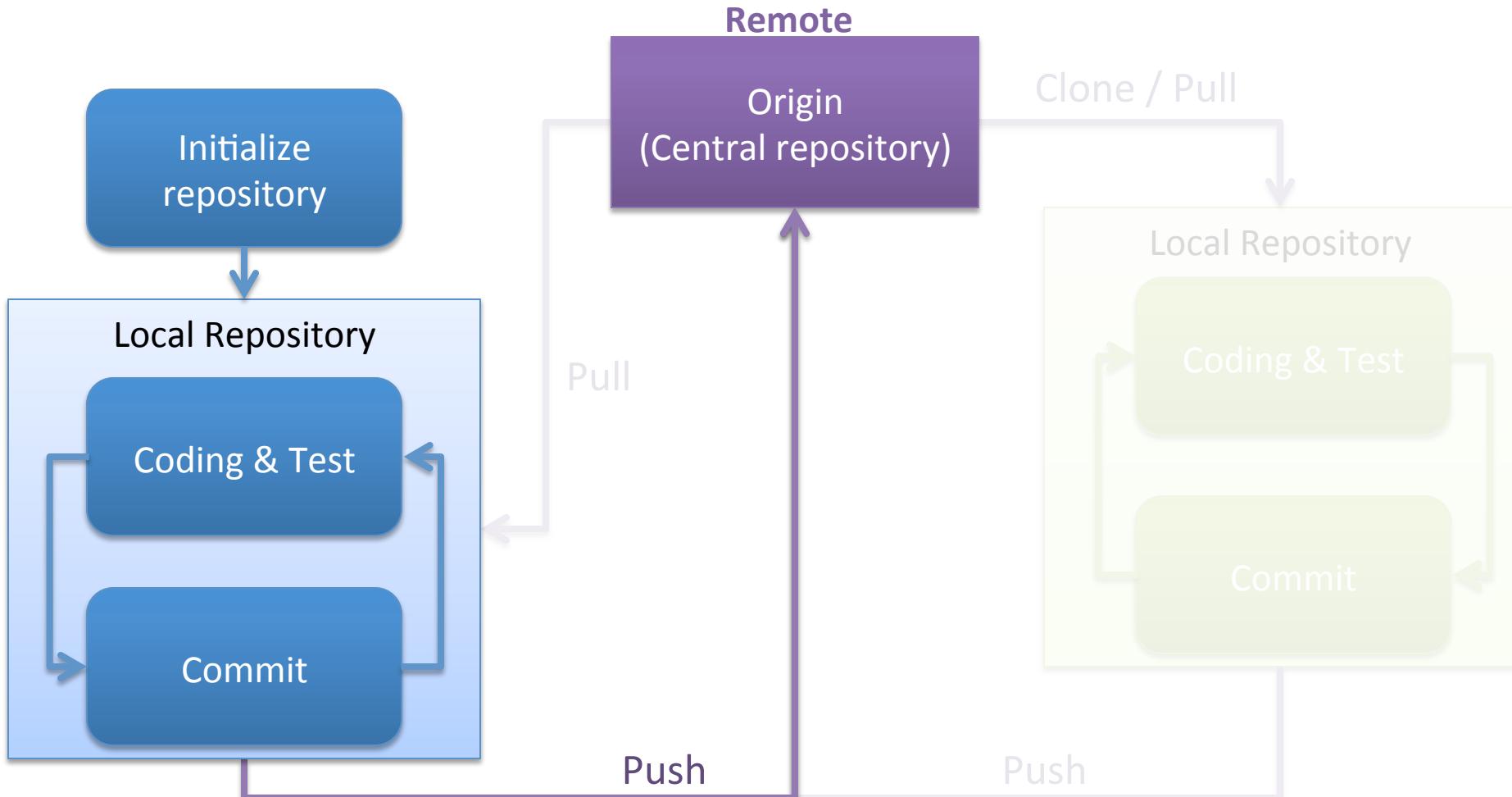
- Only staged file will be committed
 - Or `git commit -a` to commit all files in index (known by git)
- Some file do not need to be tracked by git
 - Add filename into `.gitignore`

```

22:43:35 jimmy@JimmyHMac ~/git/demo ↵ master + cat > .gitignore
sosad
22:44:46 jimmy@JimmyHMac ~/git/demo ↵ master + git add .gitignore

```

Ready for Working with Others



Initialize a Remote Repository

- First, initialize a git repository on remote machine (server)

```
(0) [02:47:40] ltsinn@linux3 ~$ git init --bare demo.git  
Initialized empty Git repository in /net/fokker/f0/data/gds/lts  
inn/git/demo.git/
```

- --bare for bare repository (no working copy)

```
(0) [02:43:20] ltsinn@linux3 ~/git/demo.git (O) $ ls  
HEAD branches config description hooks info objects refs
```

- This is not compulsory

First Push to Remote

- Add remote to local repository

`git remote add <name> <url>`

- URL: Depends on how to connect to remote

- E.g. connect via SSH

```
✓ 02:44:11 jimmy@JimmyHMac ~/git/demo
└ master git remote add linux ssh://linux3/
~/git/demo.git
```

- URL will be provided by service provider

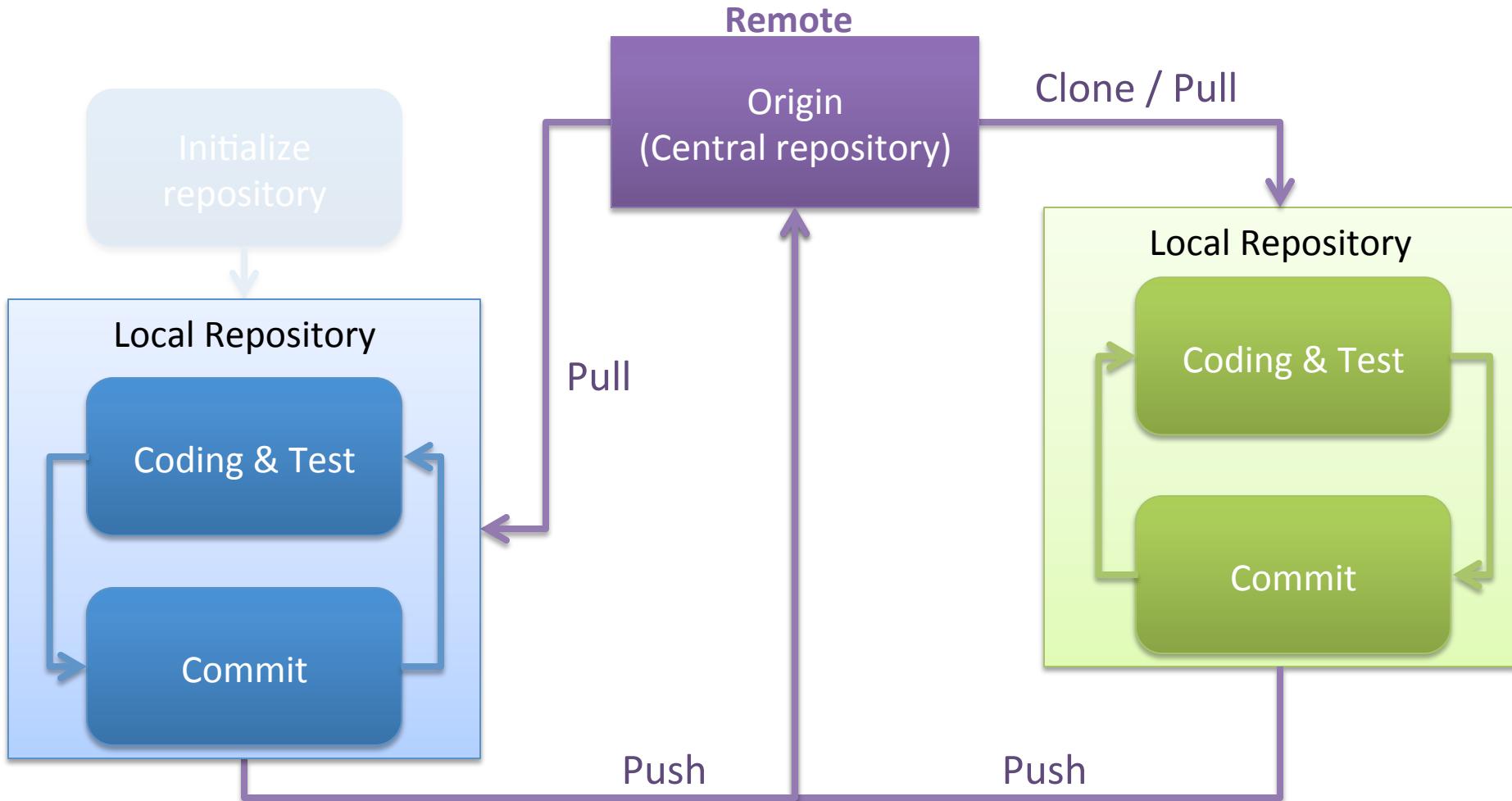
- First push

`git push <remote-name> <branch>`

- Branch name: master (default)

```
x 02:44:41 jimmy@JimmyHMac ~/git/demo
└ master git push linux master
bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
Counting objects: 8, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (8/8), 511 bytes | 0 bytes/s, done.
Total 8 (delta 0), reused 0 (delta 0)
Killed by signal 1.
To ssh://linux3/~/git/demo.git
 * [new branch]      master -> master
```

Synchronize with Remote Repository



Remote Repository (1)

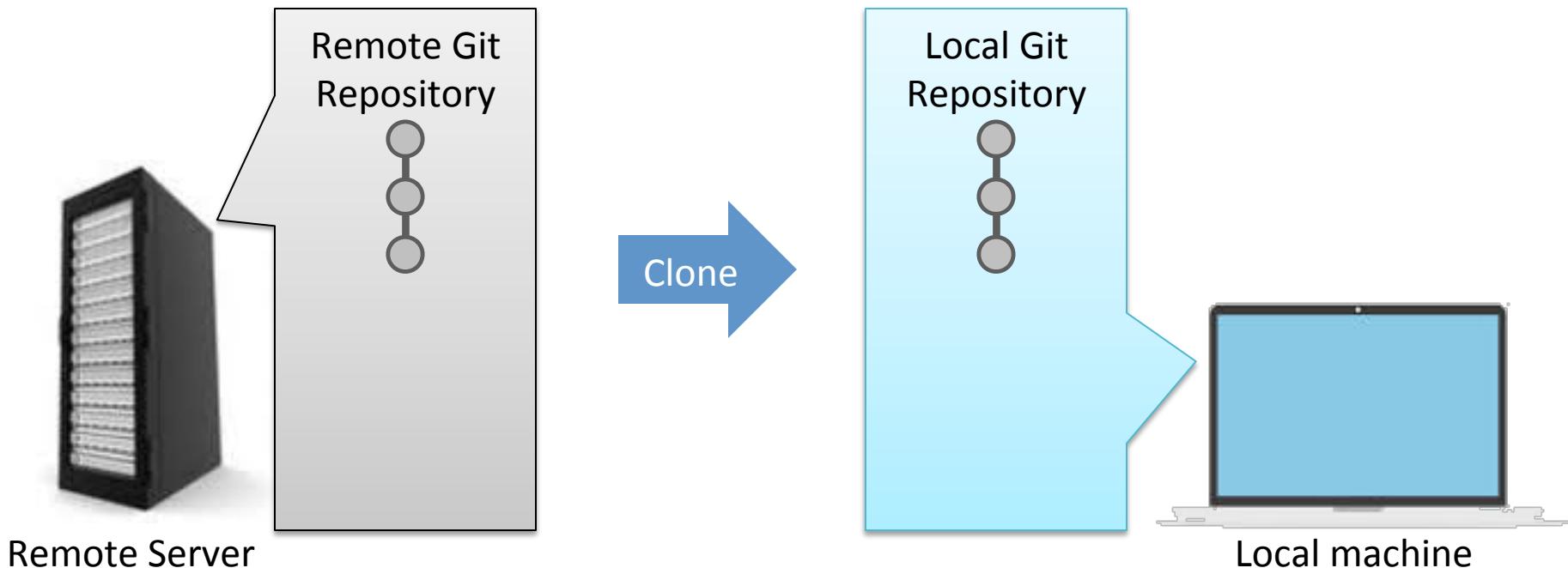
- Synchronize with remote repository
 - Remote repository hosted somewhere else
- To work in local machine, we first need a repository in local machine



Remote Repository (2)

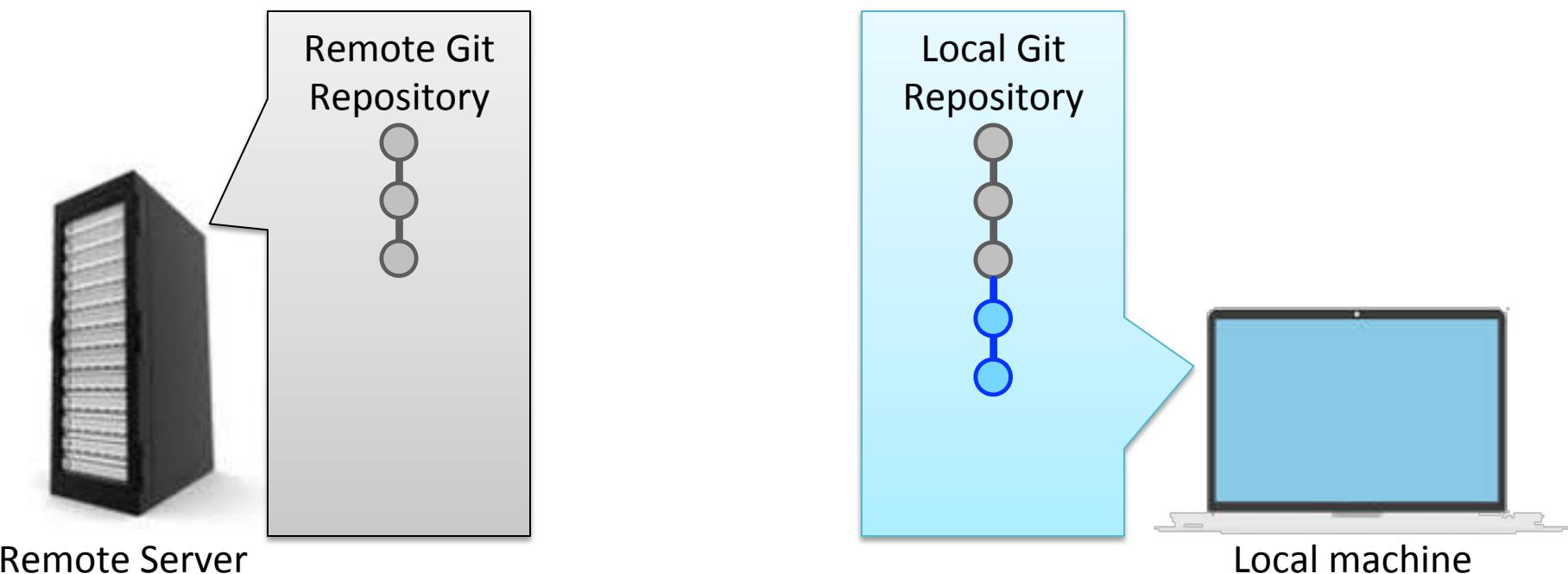
- Synchronize with remote repository
- Clone the remote repository to local
 - Instead of `git init`

`git clone <URL>`



Remote Repository (3)

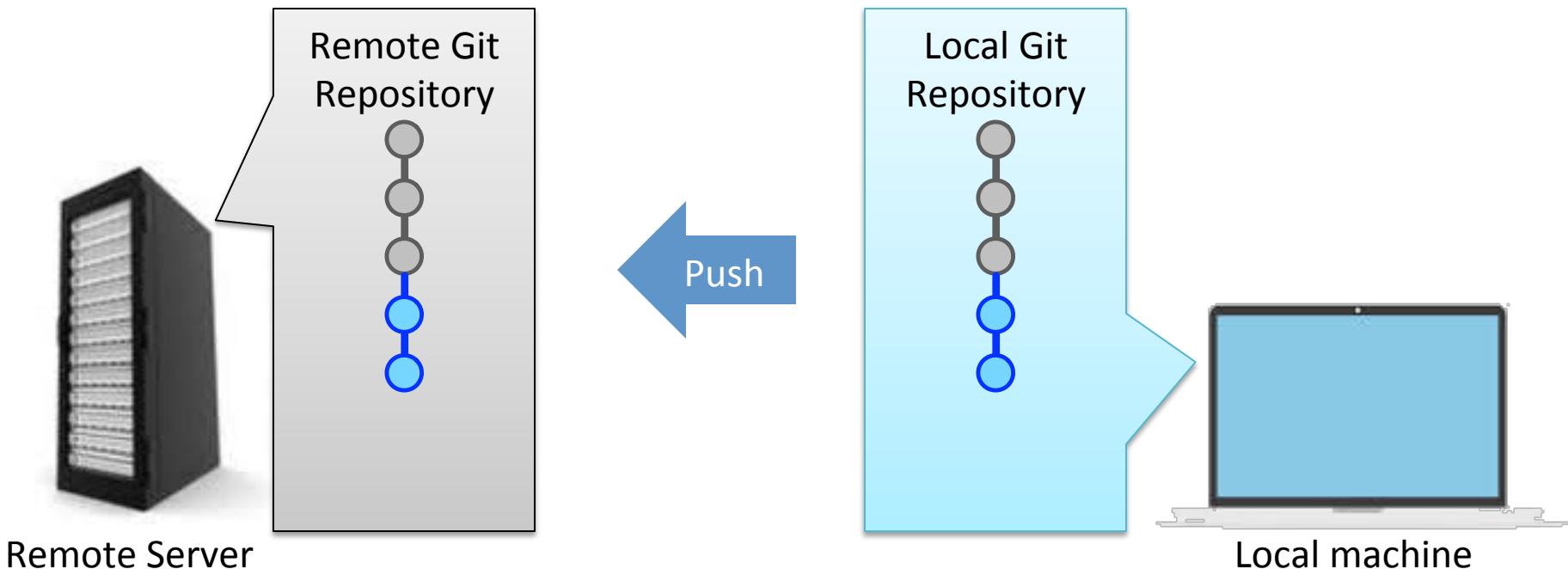
- Synchronize with remote repository
- Work as usual
 - `git commit`



Remote Repository (4)

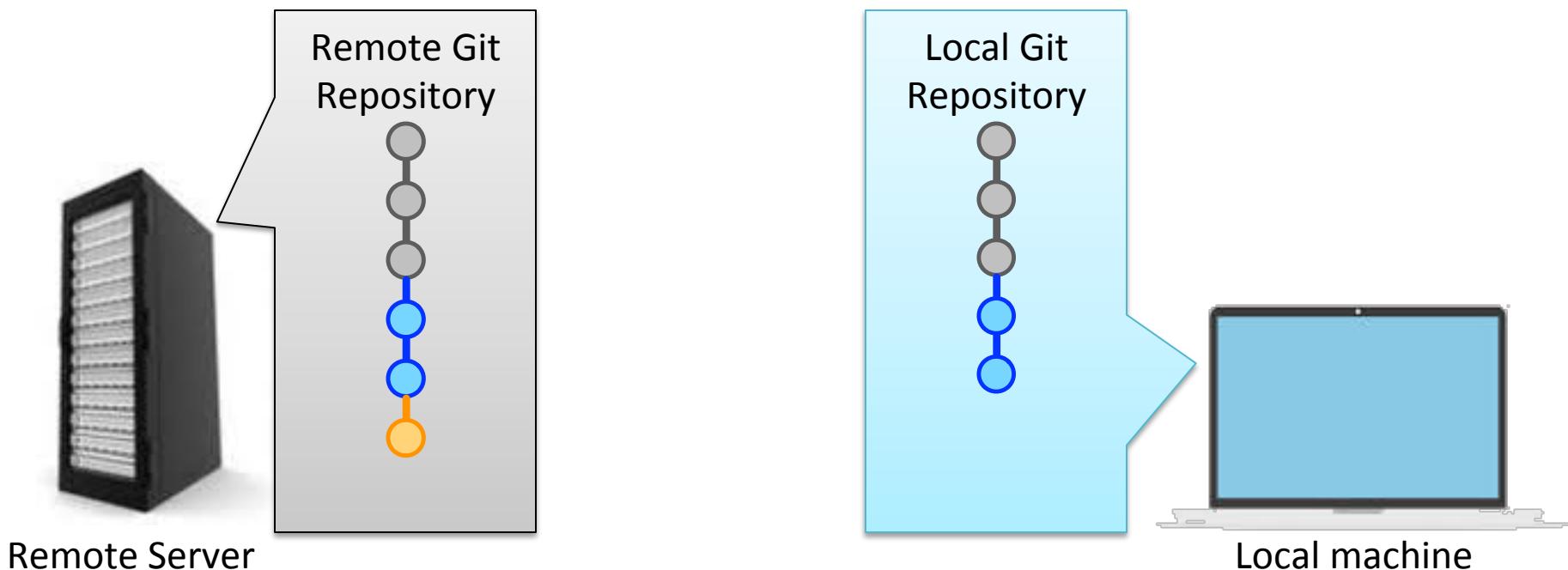
- Synchronize with remote repository
- After local edit and commit, push change to remote server

git push



Remote Repository (5)

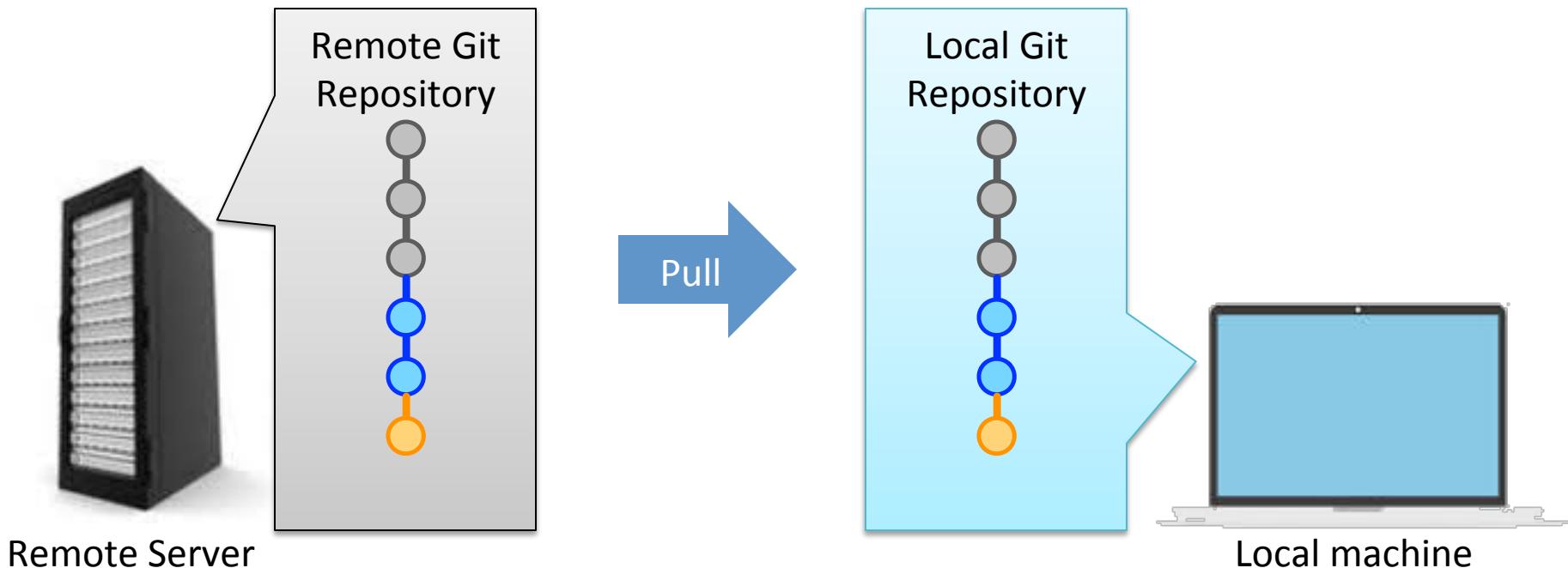
- Synchronize with remote repository
- Afterward, remote repository may be changed
 - By other teammate
 - Or from different machine



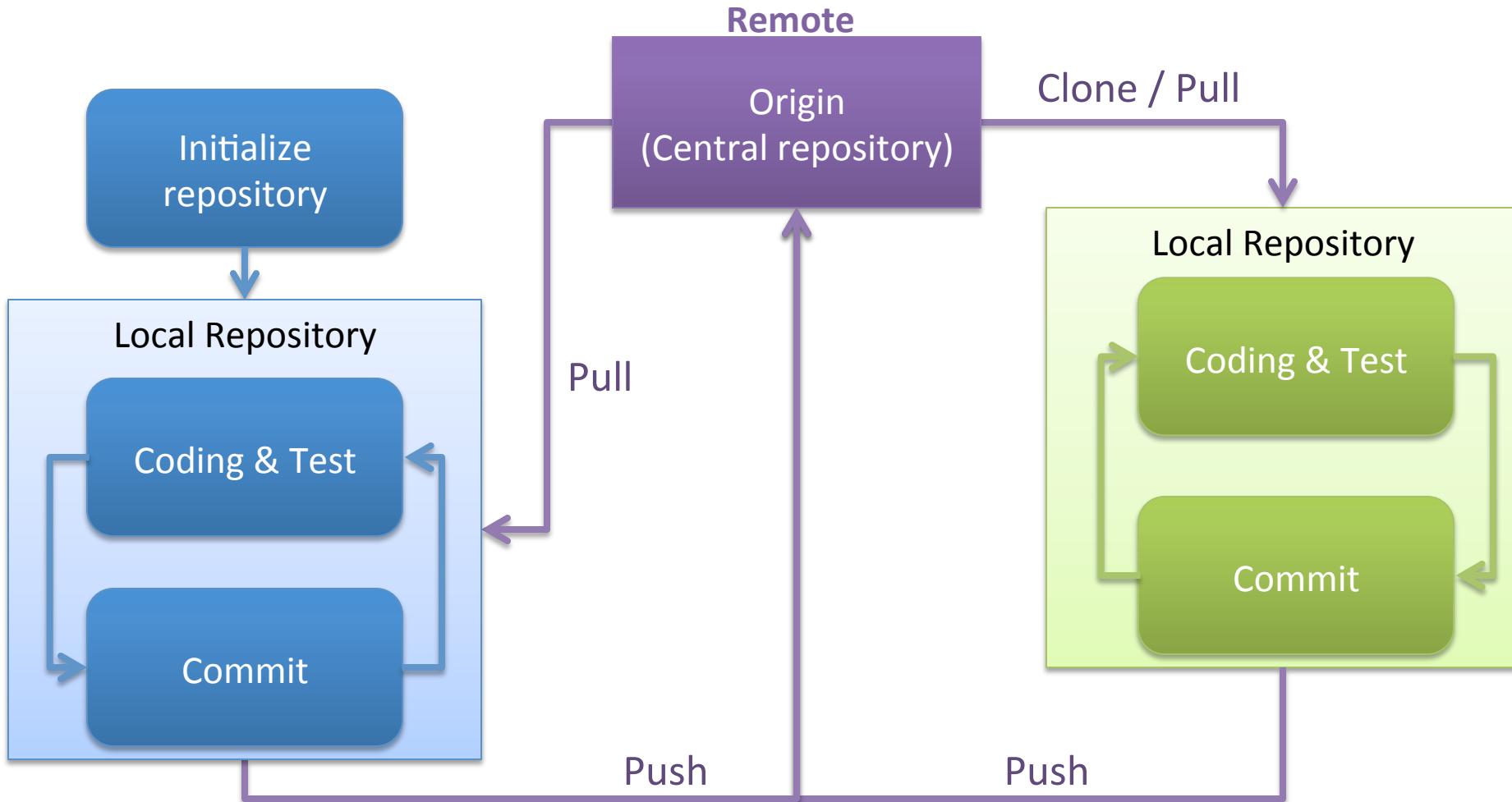
Remote Repository (6)

- Synchronize with remote repository
- Pull the commit from remote to local machine

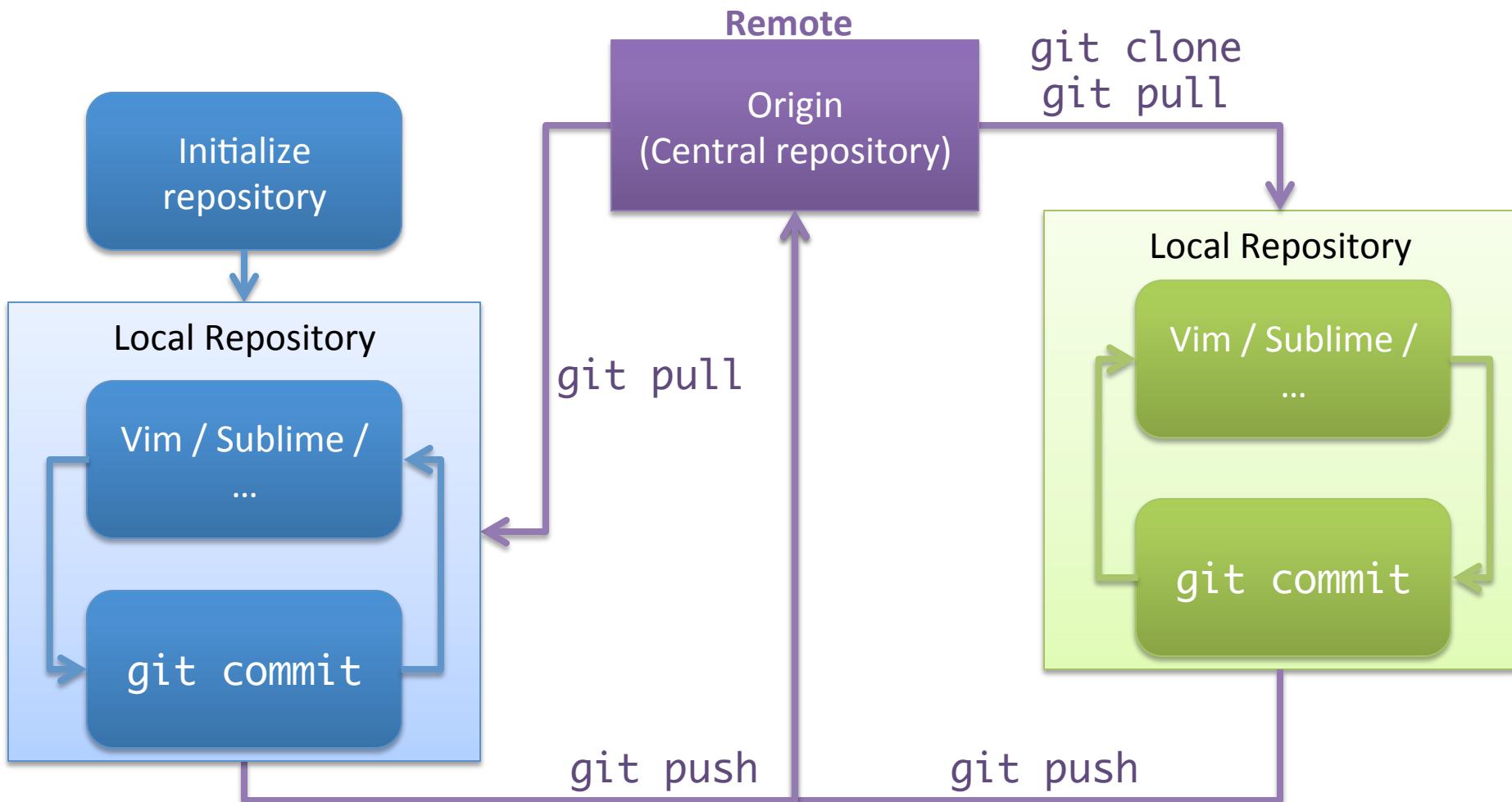
git pull



Workflow



Workflow (Commands)

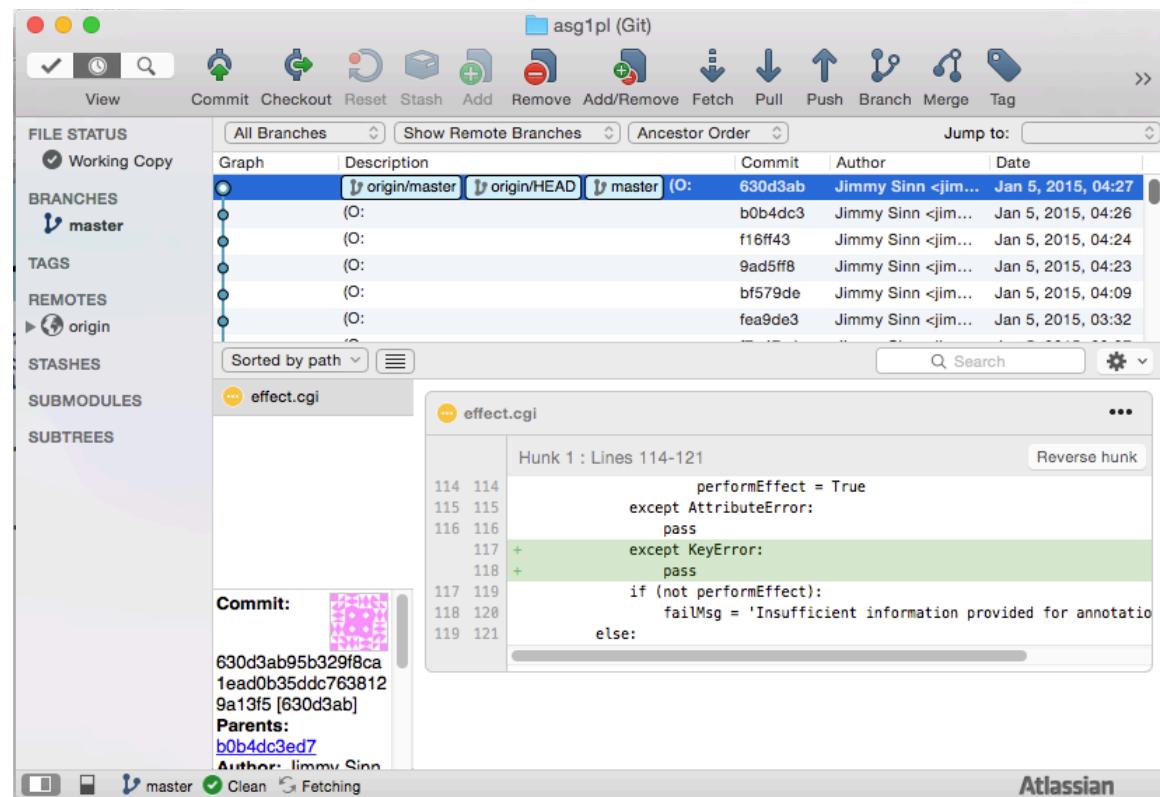


Many more operations ... → Appendix

GUI Clients

- **SourceTree**

- <https://www.atlassian.com/software/sourcetree/>
- For Mac and Windows



Github / BitBucket

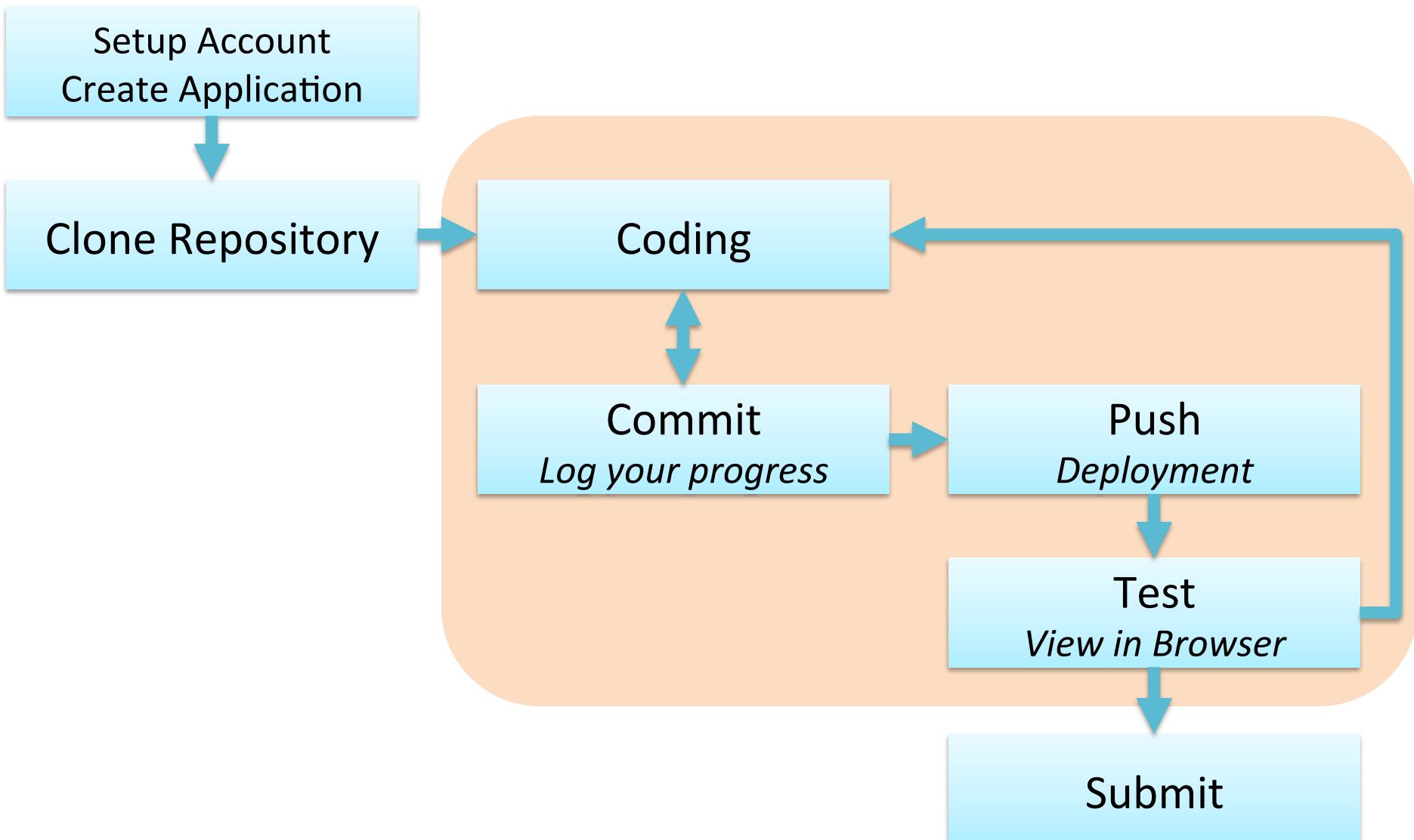
- Repository hosting service
 - Host your repository
 - Manage team works
- **GitHub** (<https://github.com>)
 - Free user: All repositories are public
 - Encourage social collaboration
 - Education plan available: <https://education.github.com/>
-  **Bitbucket** (<https://bitbucket.org/>)
 - Free user: 5 private repositories
 - Academic plan: Unlimited private repositories

OPENSHIFT

Redhat OpenShift

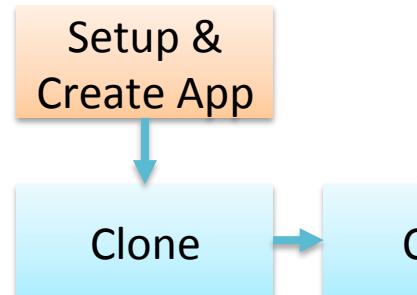
- RedHat Cloud Service
 - <https://www.openshift.com/>
 - Platform as a Service (PaaS)
 - You don't need to take care about hardware / OS / server implementation
- Free
 - Up to 3 applications
- Setup web ‘server’ easily
 - Support perl, python, PHP, node.js, RoR etc.
 - One click to setup database

Develop using OpenShift



Setup

- Register account
- Add SSH *public* key to OpenShift profile
 - Generate SSH key pair: ssh-keygen or PuTTYgen
 - *Public* key default location: `~/.ssh/id_rsa.pub`
 - *Private* key: Your identity
- Create Perl application
 - Don't use python application
- Add Cartridges
 - MySQL
 - phpMyAdmin (recommended)



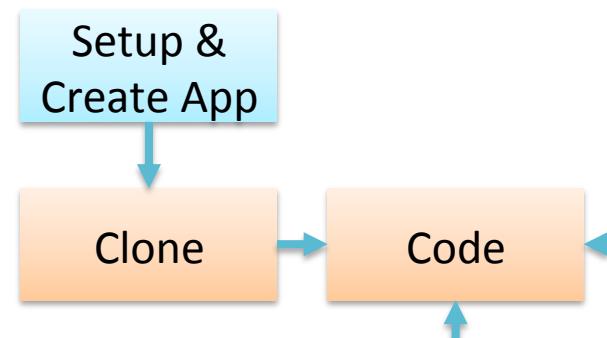
Start working

- Remote repository is created on OpenShift server
 - Clone to local machine first: `git clone`
- To run Python script in Perl application
 - File extension should be `.cgi`
 - First line of script should use proper shebang (`#!`)

```
✓ 02:26:04 jimmy@JimmyHMac ~/openshift/asg1pl ↵ master head -n1 index.cgi
#!/usr/bin/python
```

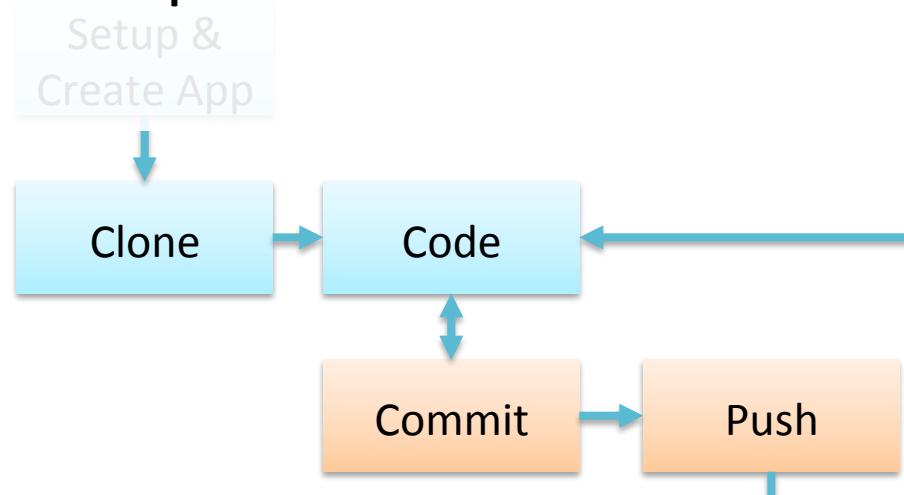
- Remember set executable for script

```
✓ 02:26:16 jimmy@JimmyHMac ~/openshift/asg1pl ↵ master chmod +x index.cgi
```



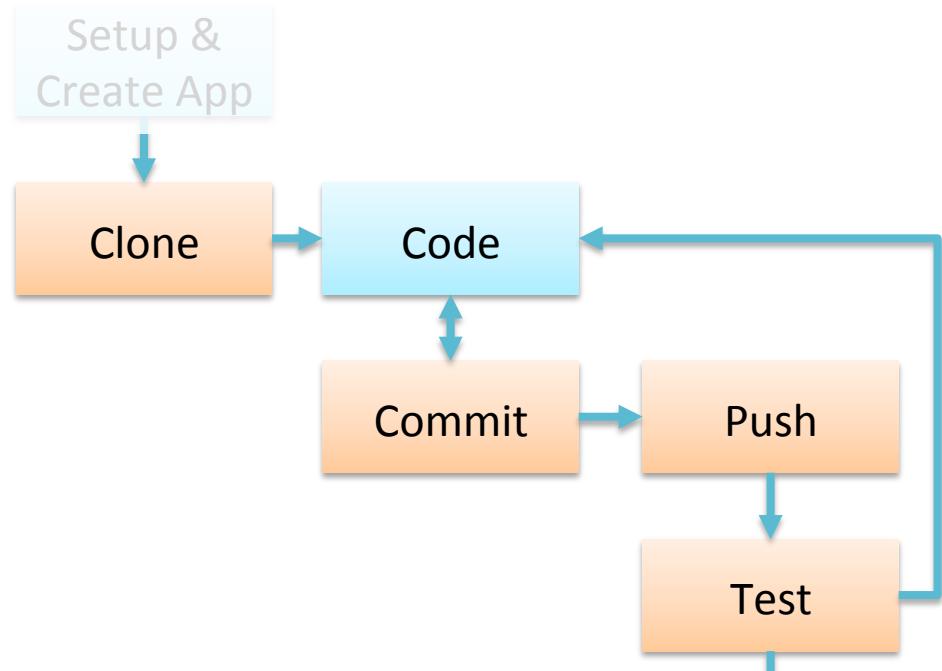
Push to OpenShift Server

- Synchronize repository by `git pull` and `git push`
- Scripts will run on server when you **push** your code
 - Restart the apache process
 - **Flush** all your local changes on OpenShift



Demo

- Creating SSH key at default location
- Get repository from OpenShift using `git clone`
- Configure application to execute python script
- Push and Deploy code



Repository

- Structure of repository

- Action hooks

- We will need it later

- Markers

- Some flag
 - Add by touch <marker>

- Put your python script at root directory

- Just remove index.pl by `git rm index.pl`

See: http://openshift.github.io/documentation/oo_cartridge_guide.html#perl

ASSIGNMENT 1 PREVIEW

Web Instagram

- Upload an image
- Perform operations on uploaded image
- Simplified photo gallery
- Specification will be released soon
- Coming tutorials:
 - HTML / Python
 - File upload
 - ImageMagick
 - Connect to database
 - Cookies / Sessions
 - Debugging skills
 - Submission Guideline

Before the end ...

- Demo walkthrough available
 - Follow the step at home
- Try to deploy simply python script to OpenShift
- Join Facebook Group
 - Announcement there
 - Feel free to ask questions

END

Contact: Jimmy, Sinn Lok Tsun (Office: SHB115 / SHB1026)

Facebook Group: <http://goo.gl/JknhKr>

→ Appendix: Advanced (?) topics on git & Techniques in SSH

APPENDIX

*Advanced (?) Topics on Git
Techniques in SSH*

Config

- Configure your name, email

```
git config --global user.name <name>  
git config --global user.email <email>
```

- Enable color

```
git config --global color.ui auto
```

Move / Remove Tracked File

- Once a file is tracked by git, we need to use git to operate the file

- To move (or rename) a file

```
git mv <old_file> <new_file>
```

- To remove a file

```
git rm <file>
```

- -f if the file is modified

- --cached if you want to keep the file but remove from git

Diff

`git diff <file>`

- Produce difference of the file in working directory and in latest commit
- You can specify the commit to diff

`git diff <commit-hash> <file>`

Revert Change

- If you have not yet commit
 - For file changes NOT staged

```
git checkout -- <filename>
```

- Unstage first if staged

```
git reset HEAD <filename>
```

```
git checkout -- <filename>
```

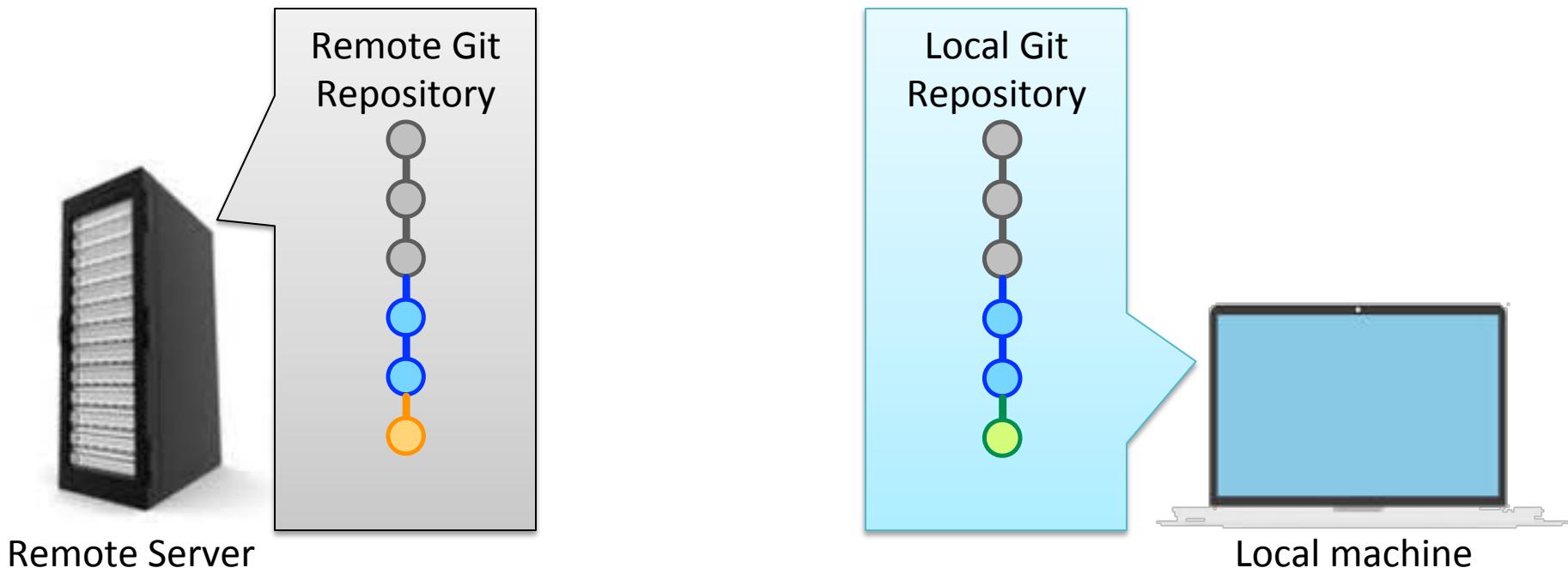
- If you want to rollback to previous commit

```
git checkout <commit-hash> <filename>
```

- Sometime first 4 characters of commit hash is enough

Fetch, Merge and Pull (1)

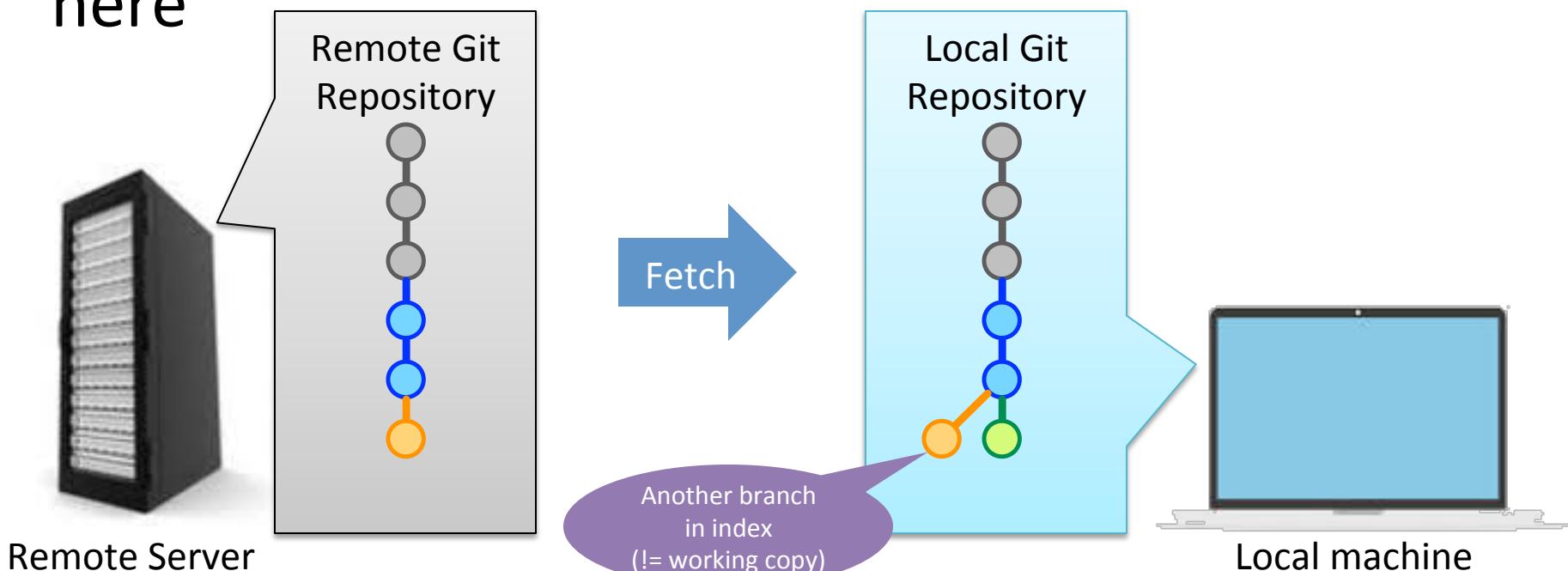
- What if the remote repository is departed from local repository
 - Maybe teammate make commit and I forgot to pull
 - Or some changes is based on previous commit



Fetch, Merge and Pull (2)

- If you just want to get latest commit from remote but NOT apply to your code
- Fetch will leads to two branches in local repository here

git fetch

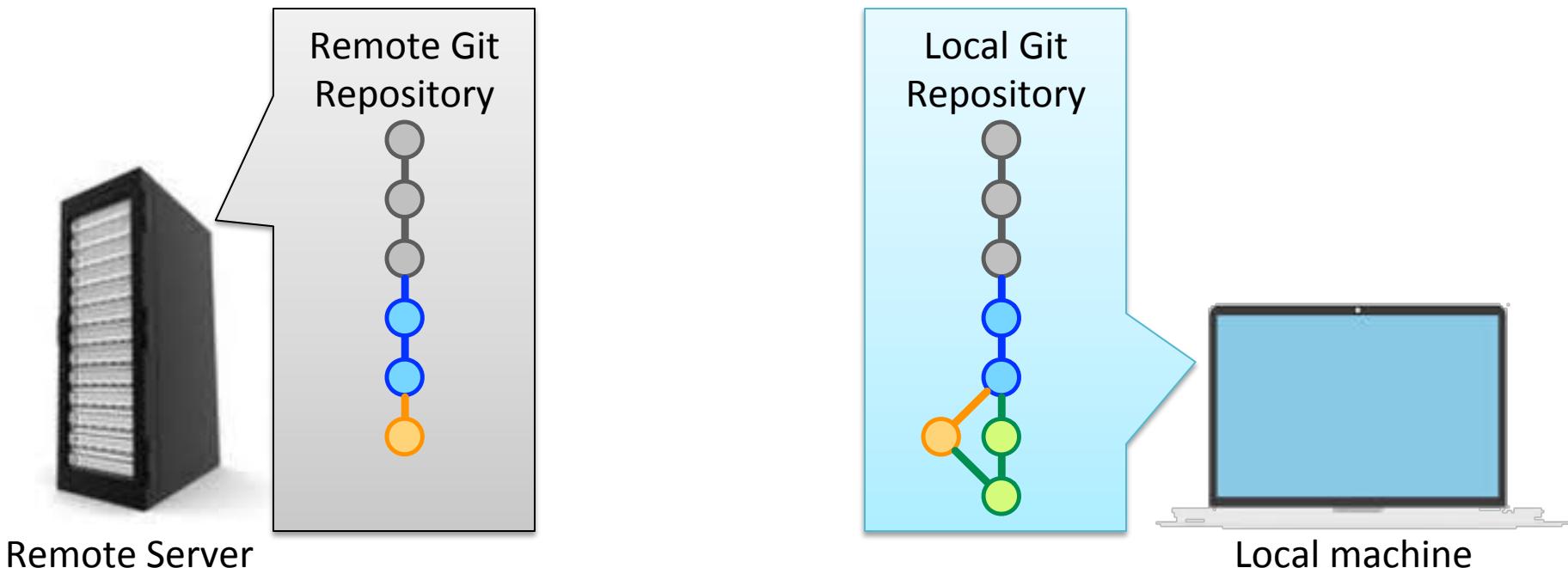


Fetch, Merge and Pull (3)

- We then merge the two branch

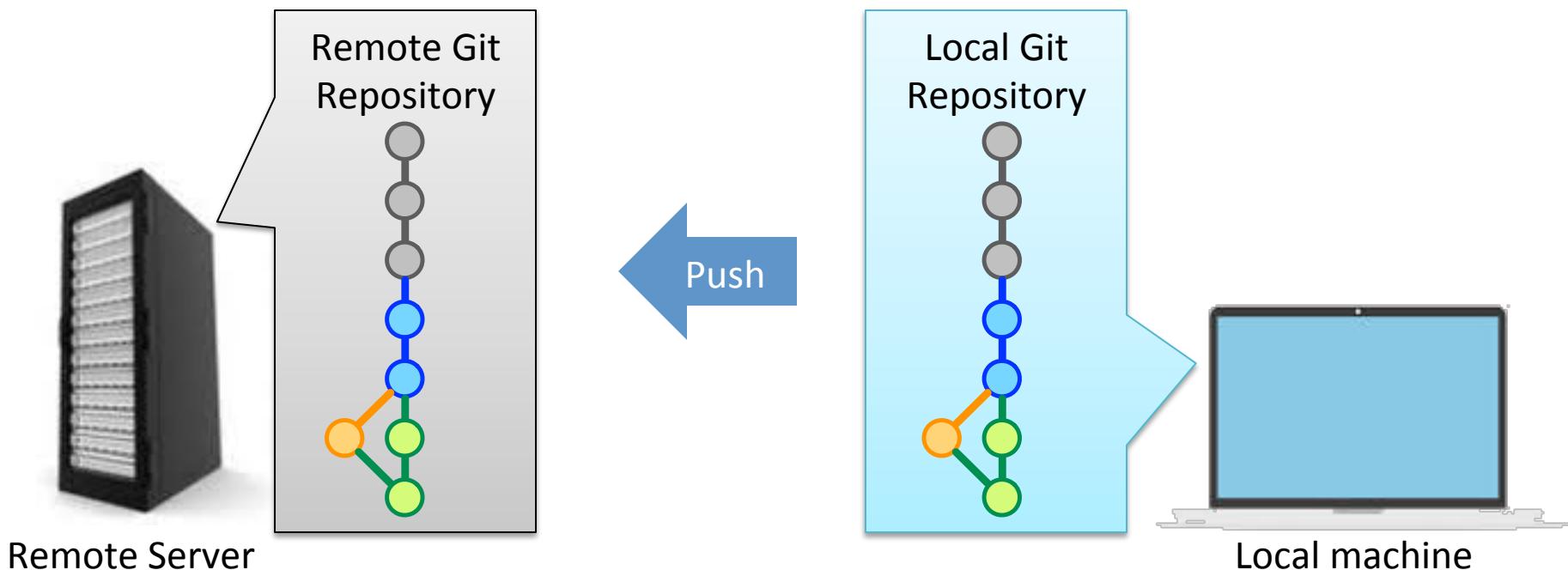
`git merge <branch>`

- If no conflict, git will automatically merge for you
- If conflict solve the conflict by yourself



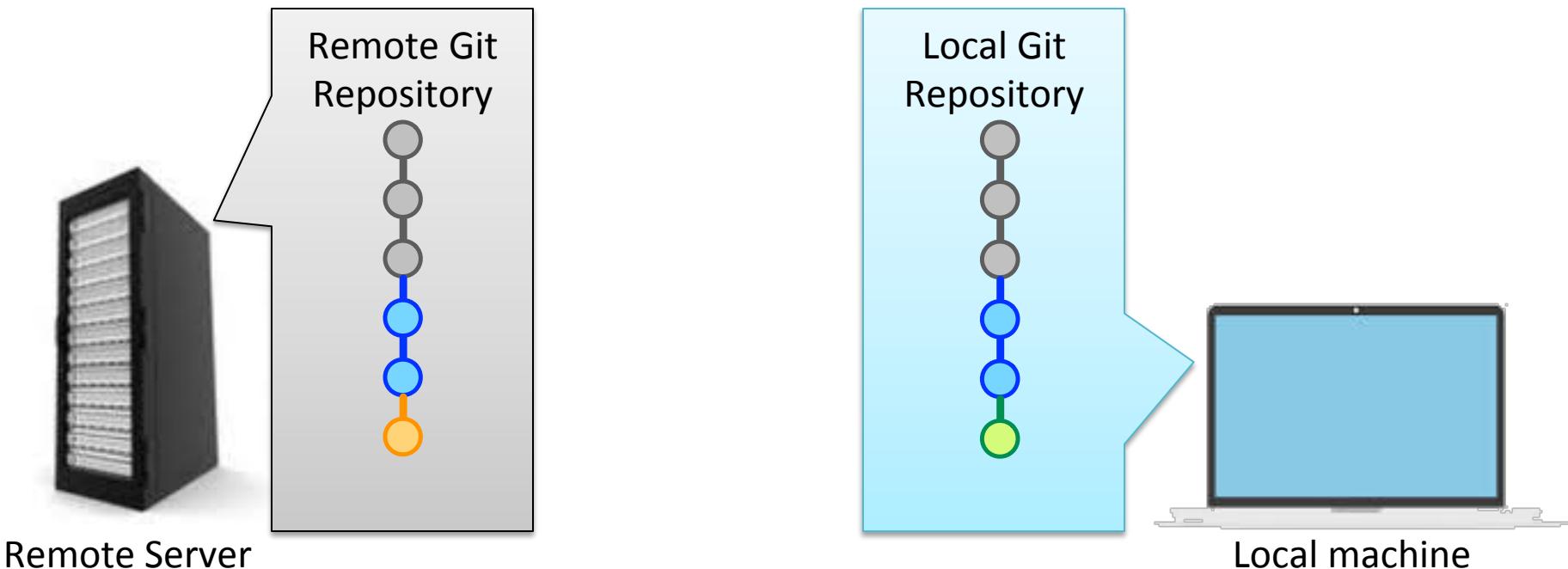
Fetch, Merge and Pull (4)

- After merge, the merge result will be committed
- Then push the merge result to remote repository by **git push**



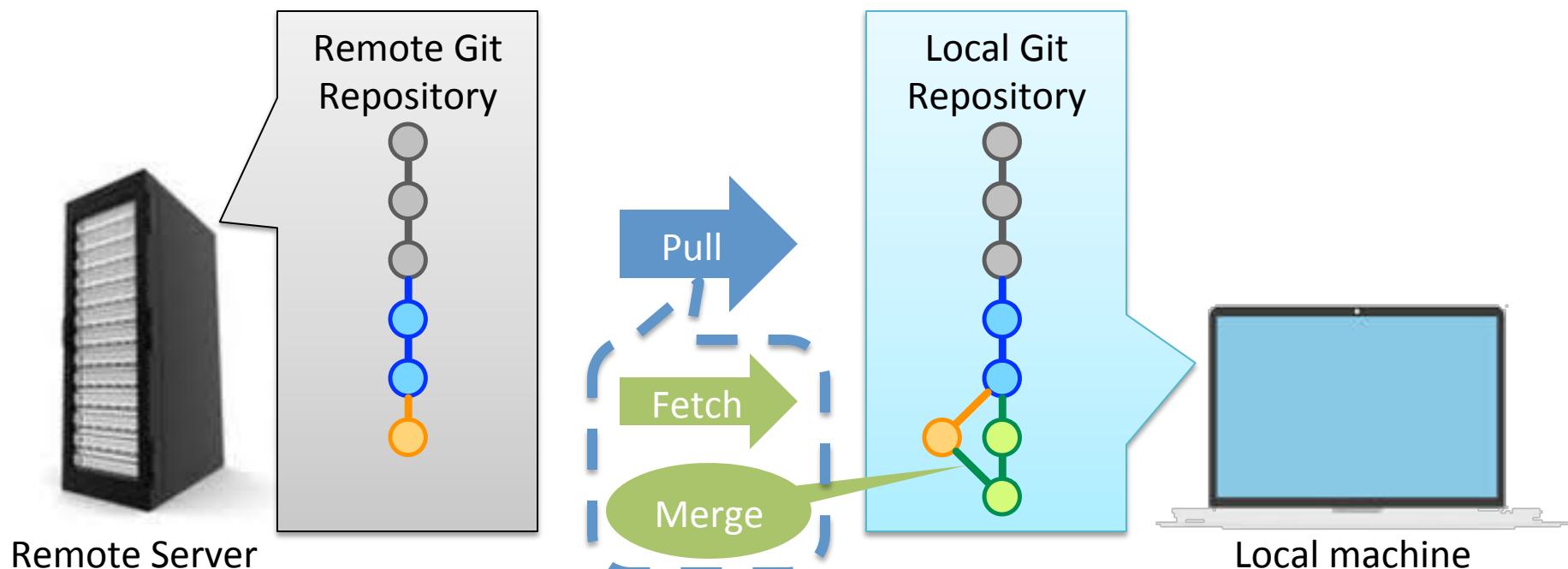
Fetch, Merge and Pull (5)

- So ... what is `git pull`?



Fetch, Merge and Pull (5)

- So ... what is `git pull`?
 - It is just fetch then merge



Accessing OpenShift from Department Machine

- You cannot directly ssh / git pull from OpenShift in department machine
- Add socksify (for linux*) or runsocks (for sparc*) to beginning of command that need to connect to internet

Using appsrv for Trying

- For quick testing and modifications only
 - You still need to test your assignment on OpenShift
- Place your files under ~/www/
 - Put scripts into ~/www/cgi-bin/
 - Change permission of files and directories (chmod)
 - **Files:** 644
 - **Directories:** 711
 - Remember the directory www as well
 - **Scripts:** 700
 - Remember use proper shebang (#!)
- Access by following link (e.g. test.cgi)
 - <http://appsrv.cse.cuhk.edu.hk/~tmchan2/cgi-bin/test.cgi>

More detail: <https://wiki.cse.cuhk.edu.hk/tech/userguide/web/cgi>

SSH Key (1)

- Public key cryptography / Asymmetric cryptography
 - Public key: Can be disclosed
 - Give it to server
 - Private key: Secret
 - Keep it by yourself
- To authenticate
 - Server use public key to encrypt text
 - You can use private key to decrypt the text
 - Server then know you are the one holding private key

SSH Key (2)

- Default location of SSH key pair
 - Private key: \${HOME}/.ssh/id_rsa
 - Public key: \${HOME}/.ssh/id_rsa.pub
- To use key from other location,
`ssh -i <private_key> <host>`
- To setup using specific key to specific host, use ssh config

SSH Config (1)

- Alias to servers / hosts
 - Configure setting every connect to the host
 - Using SSH key **other than** at default location
- Edit the file `~/.ssh/config`

```
git ssh://52c84c3f4382ec4f030002b0@  
asg1-csci4140ltsinn.rhcloud.com/  
~/git/asg1.git/
```

```
Host openshift-demo1  
hostname asg1-csci4140ltsinn.rhcloud.com  
user 52c7fe41500446534c0000ac  
identityfile ~/csci4140/openshift
```

Private key file

- Access by ssh `<host>`
 - Copy file: `scp <host>:path .`

SSH Config (2)

- Verify your setting
 - ssh -v <host>

```
✓ 03:18:35 jimmy@JimmyHMac ~ ssh -v linux3
OpenSSH_6.2p2, OSSLShim 0.9.8r 8 Dec 2011
debug1: Reading configuration data /Users/jimmy/.ssh/config
debug1: /Users/jimmy/.ssh/config line 6: Applying options for linux*
debug1: Reading configuration data /etc/ssh_config
debug1: /etc/ssh_config line 20: Applying options for *
debug1: Executing proxy command: exec ssh csegw -W linux3.cse.cuhk.edu.hk:22
debug1: permanently_drop_suid: 501
```