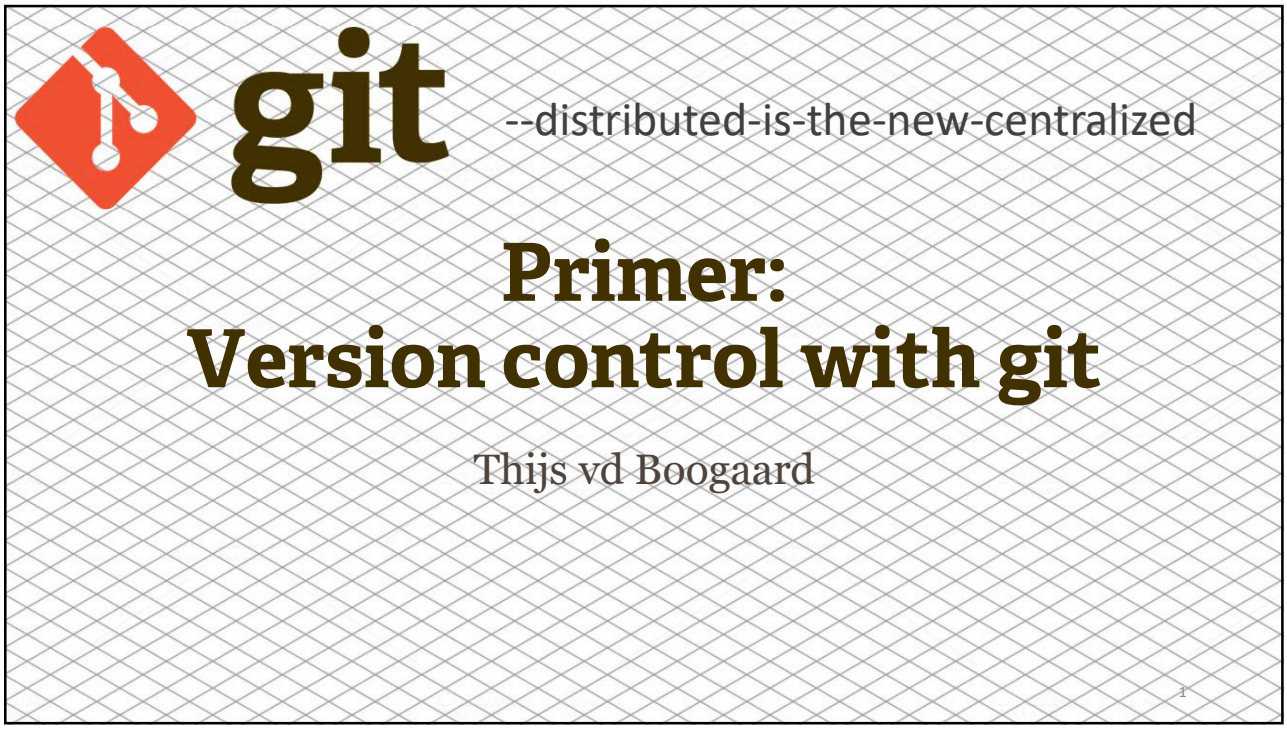

git
--distributed-is-the-new-centralized
Primer:
Version control with git
Thijs vd Boogaard
1


What the hell is git?
git
• Versioning and history
• Since 2005
• Developed by Linus Torvalds & co
• Used by multiple big companies
Google
facebook
Microsoft
twitter
LinkedIn
NETFLIX
PostgreSQL
android
RAILS
Qt
GNOME
eclipse
2

Why git?

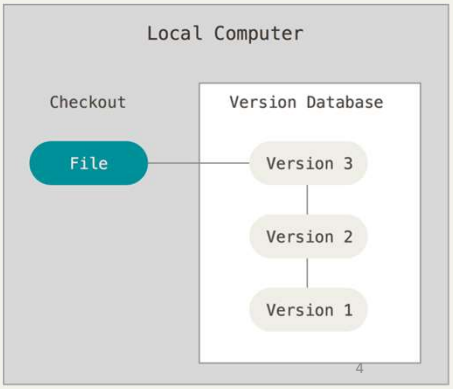
- Cooperation without the hassle
- Failsafe mechanisms
- Almost everything is local
- .....

- **Not only useful for code!**

3

Local version control

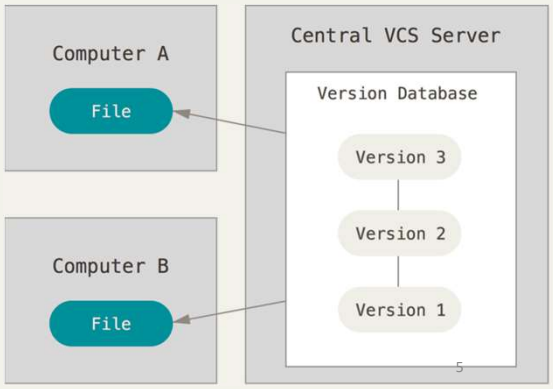
- Changes versioned only on 1 machine
- How do people cooperate in the same files?



4

Centralized version control

- 1 server which records all history (SVN, Perforce, …)
- What happens when server goes down, or even dies?

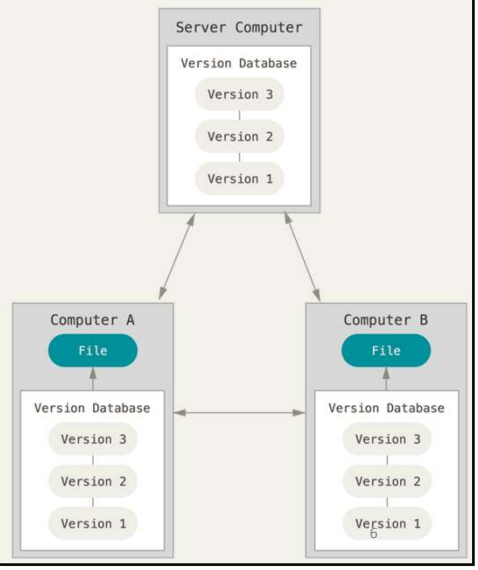


5

Distributed version control

- All users (clients) have a complete copy
- Any change is replicated
- If any server dies, every client can restore

- Each clone is essentially full backup!

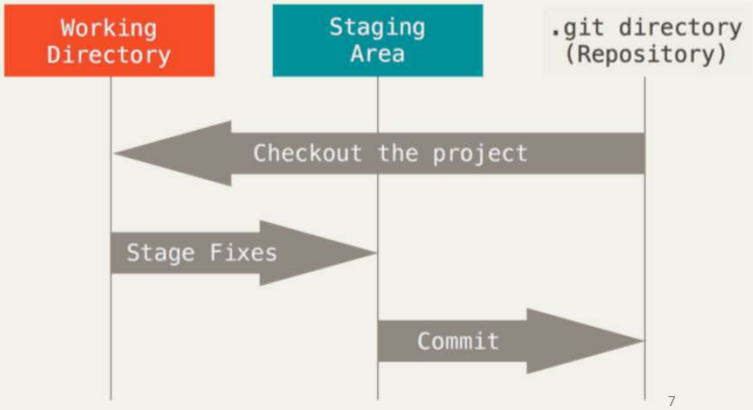


6

Basics

- Versioned files are contained in “repository”

1. Modify
2. Stage
3. Commit
4. Push



7

Setting up

- Open git bash (or Terminal for OS X)

```
thijs@NormandySR2 MINGW64 ~
$ git config --global user.email “thijs.van.den.boogaard2@hva.nl”

thijs@NormandySR2 MINGW64 ~
$ git config –global user.name “Toesoe”
```

8

Making a remote

```
thijs@NormandySR2 MINGW64 ~
$ mkdir ~/git_remote

thijs@NormandySR2 MINGW64 ~
$ cd ~/git_remote

thijs@NormandySR2 MINGW64 ~/git_remote
$ git init --bare
Initialized empty git repository in C:/Users/thijs/git_remote/

thijs@NormandySR2 MINGW64 ~/git_remote (BARE:master)
$
```

9

Getting started

```
thijs@NormandySR2 MINGW64 ~/git_remote (BARE:master)
$ mkdir ~/git_local && cd ~/git_local

thijs@NormandySR2 MINGW64 ~/git_local
$ git init
Initialized empty git repository in C:/Users/thijs/git_local/.git/

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git status
On branch master

No commits yet

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

10

Editing repo contents (modify)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ echo “Test!” >> testfile.txt

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git status
On branch master

No commits yet

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

        testfile.txt

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

11

Tracking files (stage)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git add *

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git status
On branch master

No commits yet

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

        new file:   testfile.txt
```

12

Storing changes (commit)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git commit -m "Initial commit"
[master (root-commit) dec2917] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 testfile.txt

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git status
On branch master
nothing to commit, working tree clean
```

13

Updating remote (push)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git remote add origin ~/git_remote

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 235 bytes | 235.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To C:/Users/thijs/git_remote
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

14

- Pretty easy huh?
- Any changes made can be recorded using same process

15

Conflicts

- Time to make shit hit the fan!

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ mkdir ~/git_newtree && cd ~git_newtree

thijs@NormandySR2 MINGW64 ~/git_newtree
$ git clone ~/git_remote
Cloning into 'git_remote'...
done.

thijs@NormandySR2 MINGW64 ~/git_newtree
$ cd git_remote

thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ ls
testfile.txt
```

16

Conflicts (2)

```
thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ echo “Working tree 2!” >> testfile.txt

thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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:   testfile.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

17

Conflicts (3)

```
thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ cat testfile.txt
Test!
Working tree 2!

thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ git add .

thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ git commit -m "Changes from working tree 2"
[master a98516c] Changes from working tree 2
 1 file changed, 1 insertion(+)

thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ git push
Counting objects: 3, done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To C:/Users/thijs/git_remote
   dec2917..a98516c  master -> master
```

18

Conflicts (4)

```
thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ cd ~/git_local

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ cat testfile.txt
Test!

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ echo “Working tree 1!” >> testfile.txt

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git add .

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git commit –m “Changes from working tree 1”
[master a08a31d] Changes from working tree 1
 1 file changed, 1 insertion(+)
```

19

Conflicts (5)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git push
To C:/Users/thijs/git_remote
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'C:/Users/thijs/git_remote'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

20

Conflicts (6)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git pull
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From C:/Users/thijs/git_remote
   dec2917..a98516c  master     -> origin/master
Auto-merging testfile.txt
CONFLICT (content): Merge conflict in testfile.txt
Automatic merge failed; fix conflicts and then commit the result.
```

21

Conflicts (7)

```
thijs@NormandySR2 MINGW64 ~/git_local (master|MERGING)
$ git diff
diff --cc testfile.txt
index 4acd45d,ec4b305..0000000
--- a/testfile.txt
+++ b/testfile.txt
@@@ -1,2 -1,2 +1,6 @@@
  Test!
++<<<<<<< HEAD
 +Working tree 1!
++=======
+ Working tree 2!
++>>>>>>> a98516cdc9c138416323da1de1a465f57e8aee0d

thijs@NormandySR2 MINGW64 ~/git_local (master|MERGING)
$ nano testfile.txt
```

22

Conflicts (8)

```
GNU nano 2.9.3                                                testfile.txt

Test!
<<<<<<< HEAD
Working tree 1!
=======
Working tree 2!
>>>>>>> a98516cdc9c138416323da1de1a465f57e8aee0d
```

23

Conflicts (9)

```
GNU nano 2.9.3                                                testfile.txt

Test!
Working tree 1 and 2!
```

24

Conflicts (10)

```
thijs@NormandySR2 MINGW64 ~/git_local (master|MERGING)
$ git add .

thijs@NormandySR2 MINGW64 ~/git_local (master|MERGING)
$ git commit –m “Resolved merge conflict”
[master 4235e5b] Resolved merge conflict

thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git push
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 642 bytes | 642.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To C:/Users/thijs/git_remote
   a98516c..4235e5b  master -> master
```

25

Conflicts (11)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ git log
commit 4235e5b460b0a27978675451aa245e8ad6934b4d (HEAD -> master, origin/master)
Merge: a08a31d a98516c
Author: Thijs van den Boogaard <thijs.van.den.boogaard2@hva.nl>
Date:   Sat May 12 15:32:06 2018 +0200

    Resolved merge conflict

commit a08a31d5d17fc15b58cd27aaae03ea192359beee
Author: Thijs van den Boogaard <thijs.van.den.boogaard2@hva.nl>
Date:   Sat May 12 15:15:00 2018 +0200

    Changes from working tree 1
```

26

Conflicts (12)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ cd ~/git_newtree/git_remote

thijs@NormandySR2 MINGW64 ~/git_newtree/git_remote (master)
$ git pull
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From C:/Users/thijs/git_remote
   76ad91b..b86844d  master     -> origin/master
Updating 76ad91b..b86844d
Fast-forward
 testfile.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

27

git and Unity

- git and Unity are not friends
- HvA gitlab (gitlab.fdmci.hva.nl)
- .gitignore
- LFS

28

gitlab setup

- First, login to gitlab and go to Settings > SSH Keys
- Generate a new SSH key:

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ ssh-keygen -t rsa -C "thijs.van.den.boogaard2@hva.nl" -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/thijs/.ssh/id_rsa): <enter>
Created directory '/c/Users/thijs/.ssh'.
Enter passphrase (empty for no passphrase): <enter>
Enter same passphrase again: <enter>
Your identification has been saved in /c/Users/thijs/.ssh/id_rsa.
Your public key has been saved in /c/Users/thijs/.ssh/id_rsa.pub.
```

29

gitlab setup (2)

```
thijs@NormandySR2 MINGW64 ~/git_local (master)
$ cat ~/.ssh/id_rsa.pub

<a lot of text which is your public key>
```

30

gitlab setup (3)

- https://pastebin.com/PAAkp5YB
- https://pastebin.com/hxKpHbGv

31

gitlab setup (4)

```
cd ~/Documents/”Unity Project” #C:\Users\Thijs\Documents\Unity Project

git init

git remote add origin git@gitlab.fdmci.hva.nl:boogaam004/UnityFab.git

git add .

git commit -m "Initial commit“

git lfs install

git push -u origin master
```

32

gitlab setup (5)

```
thijs@NormandySR2 MINGW64 ~/Documents/Unity Project (master)
$ echo “My project is now version controlled!”
My project is now version controlled!
```

33

gitlab setup

- Don’t forget, after making changes:

1. Add
2. Commit
3. Push

34

Thanks for listening!

35