

Git : model and workflows

Julien Durillon

Clever Cloud

March 16, 2011

Table of Contents

1 Introduction

2 Objects

3 Remotes

4 Workflows

Questions?

What now?

1 Introduction

2 Objects

3 Remotes

4 Workflows

Introduction

- Hard to manage all the commands?
- What does checkout do ?

Introduction

- Hard to manage all the commands?
- What does checkout do ?
- to revert unstaged changes

Introduction

- Hard to manage all the commands?
- What does checkout do ?
- to revert unstaged changes
- Set a file in an old state. . .

Introduction

- Hard to manage all the commands?
- What does checkout do ?
- to revert unstaged changes
- Set a file in an old state. . .
- of a commit, of a tag, of a branch

Questions?

What now?

1 Introduction

2 Objects

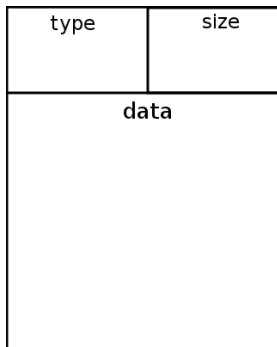
- User objects
- Commits
- Storage
- Examples
- Branches, Tags, all that stuff
- Back to checkout

3 Remotes

4 Workflows

Git objects

What do they all look like?



Addressed by a SHA1 identifier calculated from the whole content of the object.

Blobs and Trees

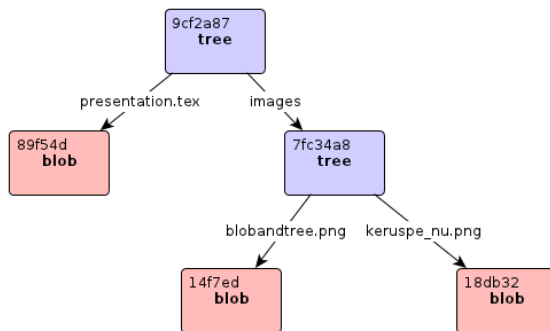
Basically. . .

- A blob is a file
- A tree is a folder

Blobs and Trees

Basically. . .

- A blob is a file
- A tree is a folder



Commits

Contain

- Author
- Committer
- Parent
- Tree
- Message

Commits

Contain

- Author – The one who wrote the code
- Commiter
- Parent
- Tree
- Message

Commits

Contain

- Author – The one who wrote the code
- Commiter – The one who created the commit object
- Parent
- Tree
- Message

Commits

Contain

- Author – The one who wrote the code
- Commiter – The one who created the commit object
- Parent – The commit(s) before this one
- Tree
- Message

Commits

Contain

- Author – The one who wrote the code
- Committer – The one who created the commit object
- Parent – The commit(s) before this one
- Tree – The top tree object of the committed state
- Message

Commits

Contain

- Author – The one who wrote the code
- Commiter – The one who created the commit object
- Parent – The commit(s) before this one
- Tree – The top tree object of the committed state
- Message – Why the commit was done

Storage

What does .git look like?

```
.git
|-- config
|-- description
|-- HEAD
|-- hooks
|-- info
|-- objects
`-- refs
```

Storage

"We need to go deeper"

```
.git/objects
|-- 10
|   |-- 8c8bff3b82234d1726531c6643cbb541d4bc4e
|-- 42
|   |-- 0f12c08fcb785df9d8c1887e2545882d6168ea
|-- 5c
|   |-- 4b717a0ea3b0c100c5e4141da431014253d44b
|-- info
|-- pack
```

Storage

"We need to go deeper"

```
.git/objects
|-- 10
|   `-- 8c8bff3b82234d1726531c6643cbb541d4bc4e
|-- 42
|   `-- 0f12c08fcb785df9d8c1887e2545882d6168ea
|-- 5c
|   `-- 4b717a0ea3b0c100c5e4141da431014253d44b
|-- info
`-- pack

.git/refs
|-- heads
|   `-- master
`-- tags
```

Example

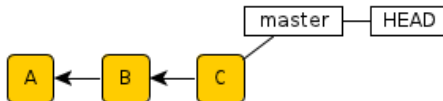
Plumbing tools

- hash-object
- cat-file

Only “labels”

No working tree copy

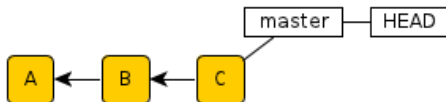
A simple project:



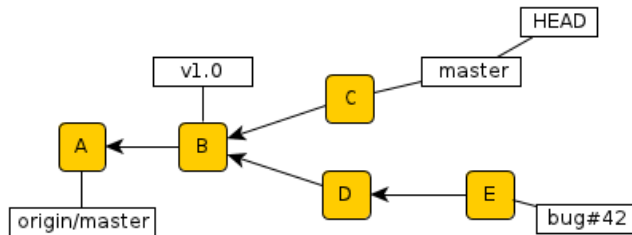
Only “labels”

No working tree copy

A simple project:



A less simple project:



Back to checkout

Does one thing: get a file/tree in the state of a commit.

Questions?

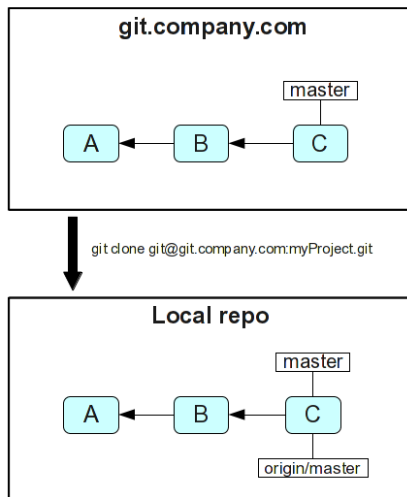
What now?

- 1 Introduction
- 2 Objects
- 3 Remotes**
 - Storing the remotes
 - Understanding the remotes
- 4 Workflows

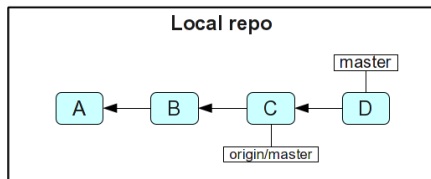
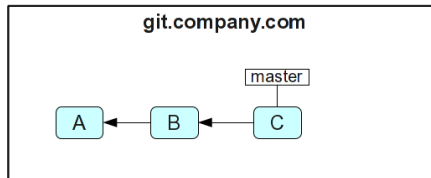
Storing the remotes

- In `.git/refs/remotes`
- points to a commit like a branch or a tag
- can't and shouldn't be changed by other than fetch

Understanding remotes



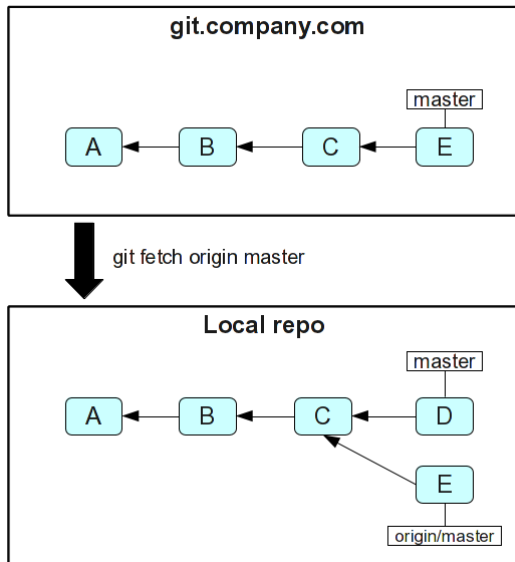
Understanding remotes



On branch master
Your branch is ahead of
'origin/master' by 1 commit.

nothing to commit (working directory clean)

Understanding remotes



Questions?

What now?

1 Introduction

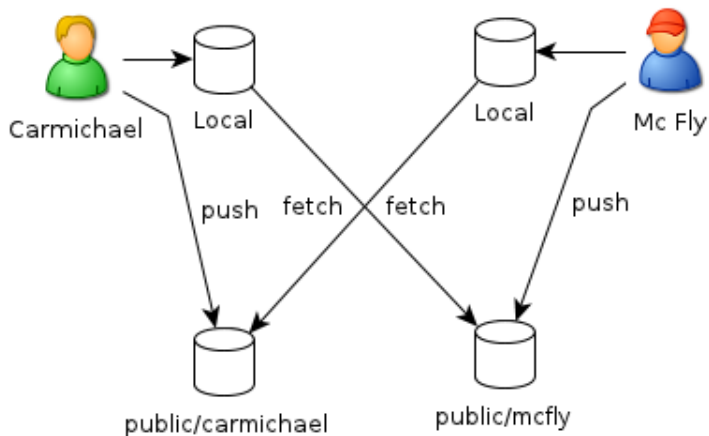
2 Objects

3 Remotes

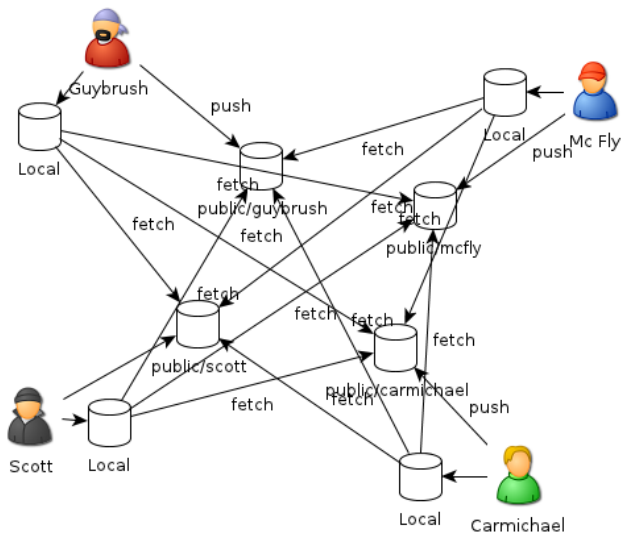
4 Workflows

- Basic "recommended" usage
- Centralized: à la Subversion
- Big project: The benevolent dictator

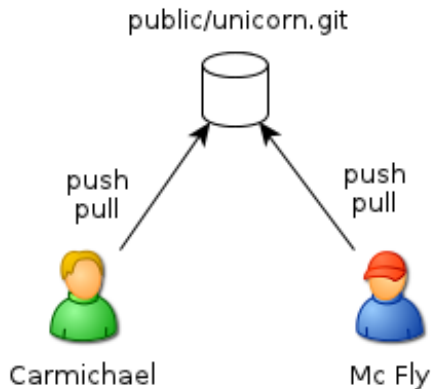
Fully decentralized: the official way



Fully decentralized: limitations



Centralized: à la Subversion



The benevolent dictator

