git @ietf

...

Agenda

What is git?

Github and the github workflow

Using github for IETF work

Tools

Hands on

caveat lector

This presentation is ~~all lies~~ simplified

The information presented is subtly, dangerously incorrect

The instructions are flat-out wrong

... expect to violate every rule once you understand better

... expect to think that the presenters are idiots once you understand better

... if you don't already, that is

The truth is out there ... in the linked instructions

setup

Required: git, make, python -> xml2rfc

Recommended: ruby -> kramdown-rfc2629

Windows: cygwin for all of the above

See the instructions

what is git?

git

I'm an egotistical bastard, and I name all my projects after myself. First Linux, now git.

-- Linus Torvalds

git howto


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

git howto

“If that doesn't fix it, git.txt contains the phone number of a friend of mine who understands git. Just wait through a few minutes of 'It's really pretty simple, just think of branches as...' and eventually you'll learn the commands that will fix everything.”

--http://xkcd.com/1597/

git == distributed version control

Every repository has a bunch of versions (or commits)

A commit describes the entire state of the repository (the files and their contents)

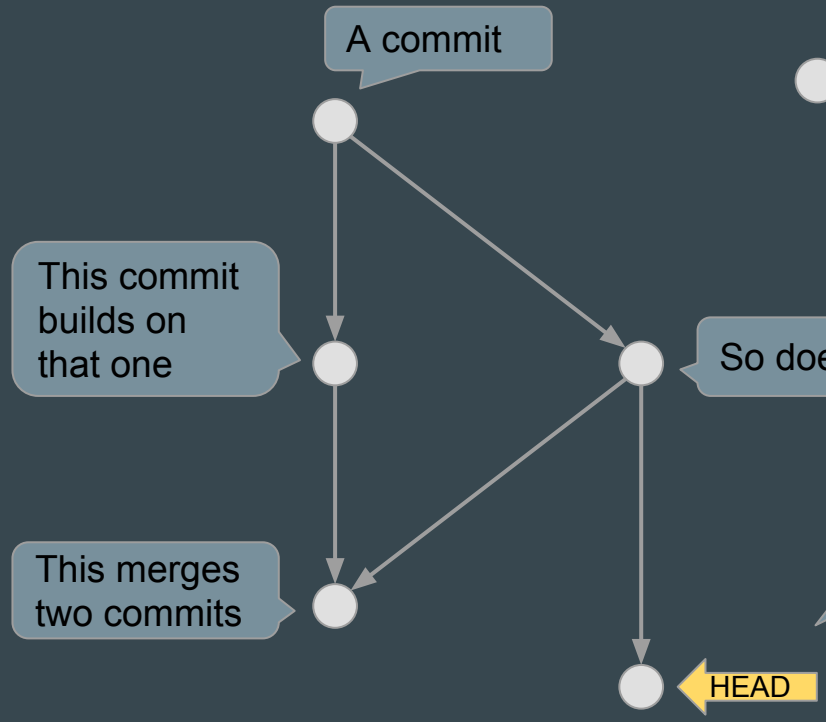
Every commit builds on one or more other commits

Everyone has their own repository

People can share commits

... which leads to some repositories having the same commits in them

git - commits



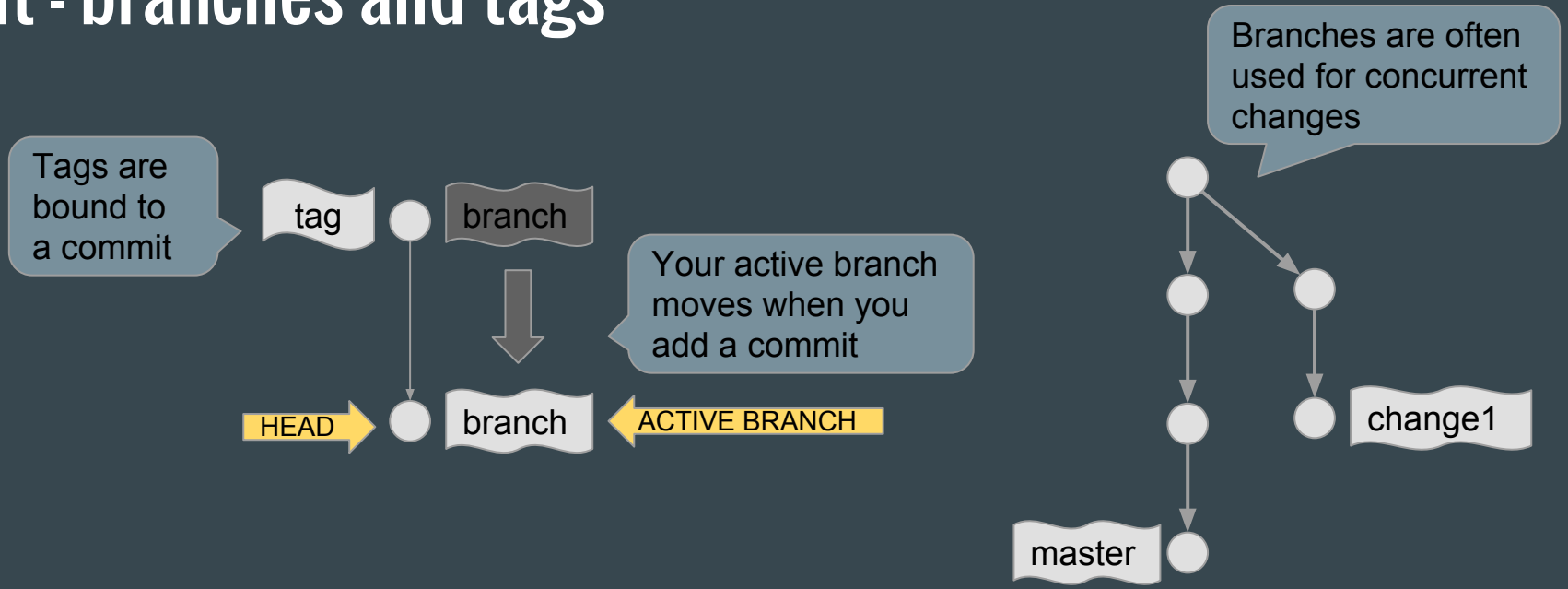
Your working directory is a copy of a commit, plus any local changes.

To create a new commit:

1. stage changes with **git add**
2. create a commit with **git commit**

To move between commits, use **git checkout**

git - branches and tags



You can check out a tag or branch with **git checkout**.

Always check out branches.

git - push and pull

Push copies commits from a local repository to a remote one

Pull copies commits from a remote repository to a local one

Pull and push are usually used to update branches

github

github is the bazaar

Projects often use github to host a central repository

github provides a bunch of tools for coordinating changes

- tracking issues
- proposing changes (“pull requests”)
- reviewing documents

github is where you will find a **very large** community of contributors

demo

github sign up

demo

create new repo

pull request

= “send text” for the modern era

github flow

create a new branch

work; add commits to that branch

push branch to github

create pull request from branch

fix; add more commits to the branch and push

merge pull request

github flow

create a new branch

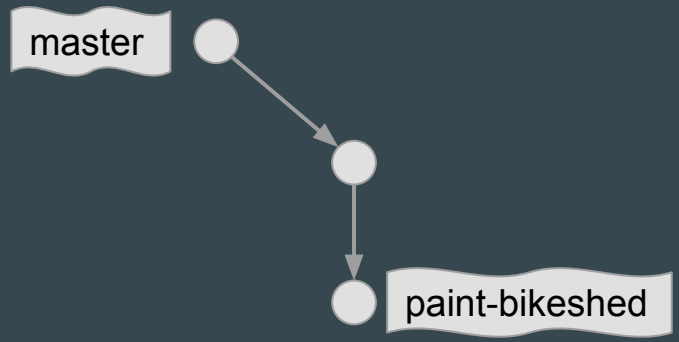
work; add commits to that branch

push branch to github

create pull request from branch

fix; add more commits to the branch and push

merge pull request



github flow

create a new branch

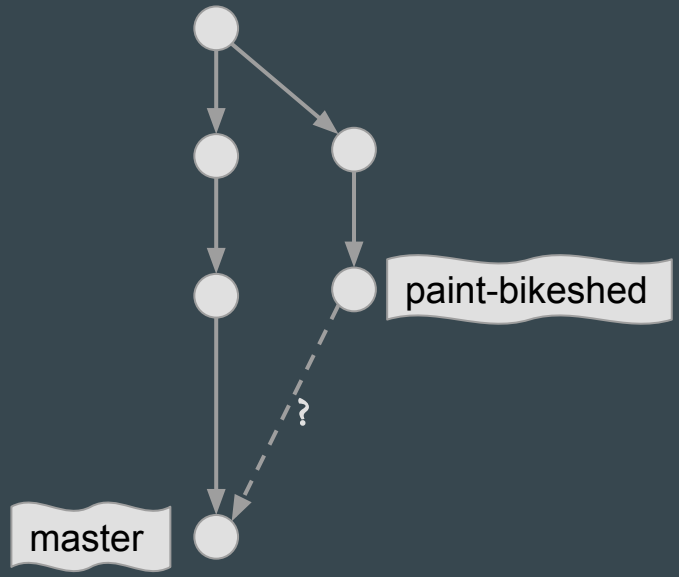
work; add commits to that branch

push branch to github

create pull request from branch

fix; add more commits to the branch and push

merge pull request



github flow

create a new branch

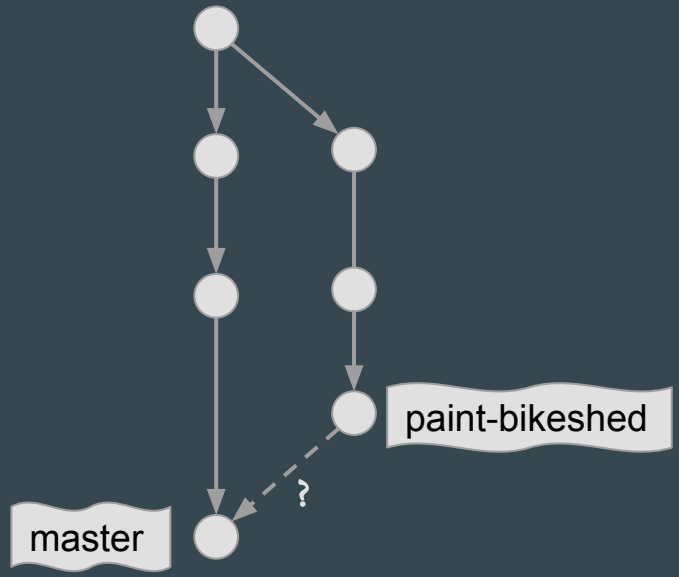
work; add commits to that branch

push branch to github

create pull request from branch

fix; add more commits to the branch and push

merge pull request



github flow

create a new branch

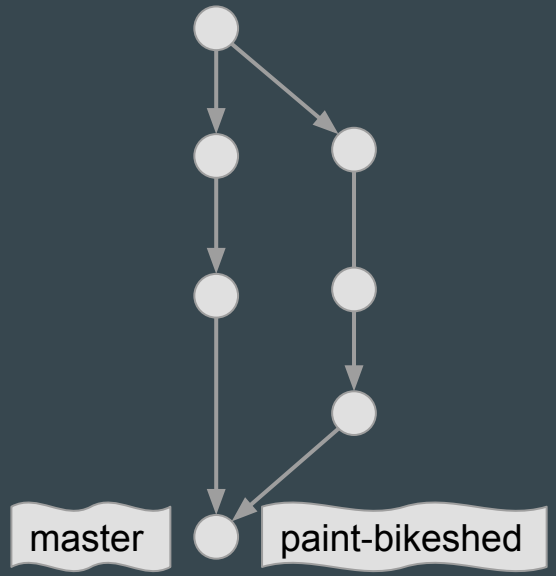
work; add commits to that branch

push branch to github

create pull request from branch

fix; add more commits to the branch and push

merge pull request



github flow

create a new branch

work; add commits to that branch

push branch to github

create pull request from branch

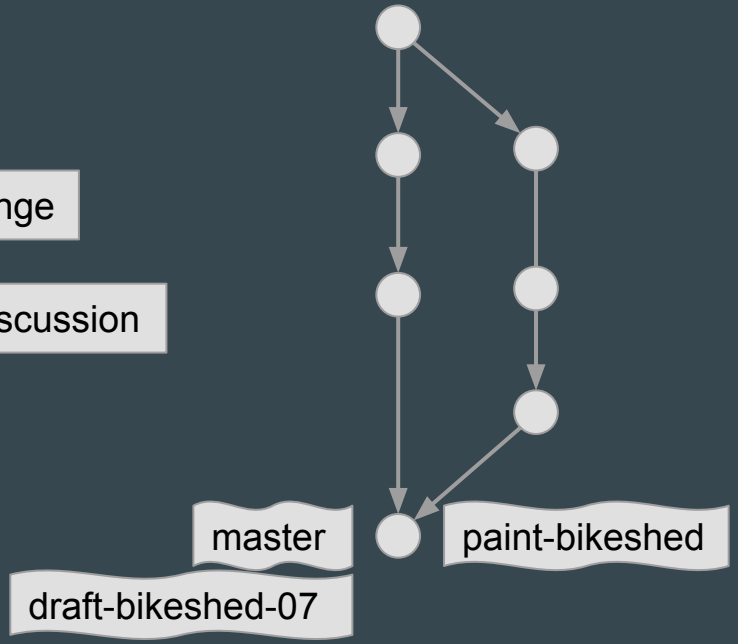
fix; add more commits to the branch and push

merge pull request

Propose change

WG discussion

Editor makes the change



demo

github flow

demo

quick PR using github web interface

using github for ietf work

github @ietf on one slide

Draft source goes into a repository

Editors get commit access, can make changes, accept pull requests

Working group can open issues and pull requests

An updated draft is occasionally tagged, built and submitted to the datatracker

guidelines

Each working group should create a new github organization

... e.g., rtcweb-wg, ietf-wg-acme, tlswg, httpwg, webpush-wg

Chairs should own the organization

Teams can be created for editors

demo

creating organizations and teams

pitfalls

Contributors need to be aware of IPR rules

Discussion venue gets confused: mailing list, github issues, even review comments

Editors make changes without consensus (not a new problem)

ipr notices

Prominent notices for IPR rules

... a link to CONTRIBUTING.md is shown for all new contributions

... README.md is shown to new visitors

Both contain a copy of the Note Well statement

discussion on the mailing list

Substantive, technical discussion on mailing list only

Issue tracker records existence of issues, and any decisions made

Editorial discussion can continue on github

maintaining transparency

github is a strict improvement in transparency

Some additional steps can improve things further

Labels for issues: editorial, design, editor-ready

Substantial changes are created as pull requests

... these can be reviewed and discussed before merging

tools

the well-trodden path

https://github.com/martinthomson/i-d-template

Consists of a few makefiles that

... compile from .xml, .md, or .org

... handles multiple drafts in the same repo

... produce draft diffs, check nits

... includes the standard notices

... integrate with continuous integration services (travis and circle currently)

draft versioning assumptions

Drafts are named in the form **draft-<name>-<wg>-<blah>.{xml,md,org}**

The version corresponding to a submitted version is tagged with the submitted name

... draft-author-wg-ponies-00

The draft contains a version number of **-latest**

... draft-author-wg-ponies-latest

make submit creates files for the next version based on the tags

demo

using the template

the gist

Github provides tools for more transparency and more openness

... and they're tools that a lot of contributors are already using

... and they've been used in several WGs already (HTTPBIS, TLS, ACME, ...)

These tools are not a replacement for WG process; they're just tools

... and it's still the chair's job to run the process

questions

tips

Pull from main repo, push to your private fork

```
git clone https://github.com/main/repo
cd repo
git remote set-url --push origin https://github.com/personal/fork
```

Windows users, don't mark everything as executable by accident

```
git config --global core.fileMode false
```

git help

https://help.github.com/

https://guides.github.com/introduction/flow/

https://help.github.com/articles/good-resources-for-learning-git-and-github/

git help <command> is useful once you understand git, but not before