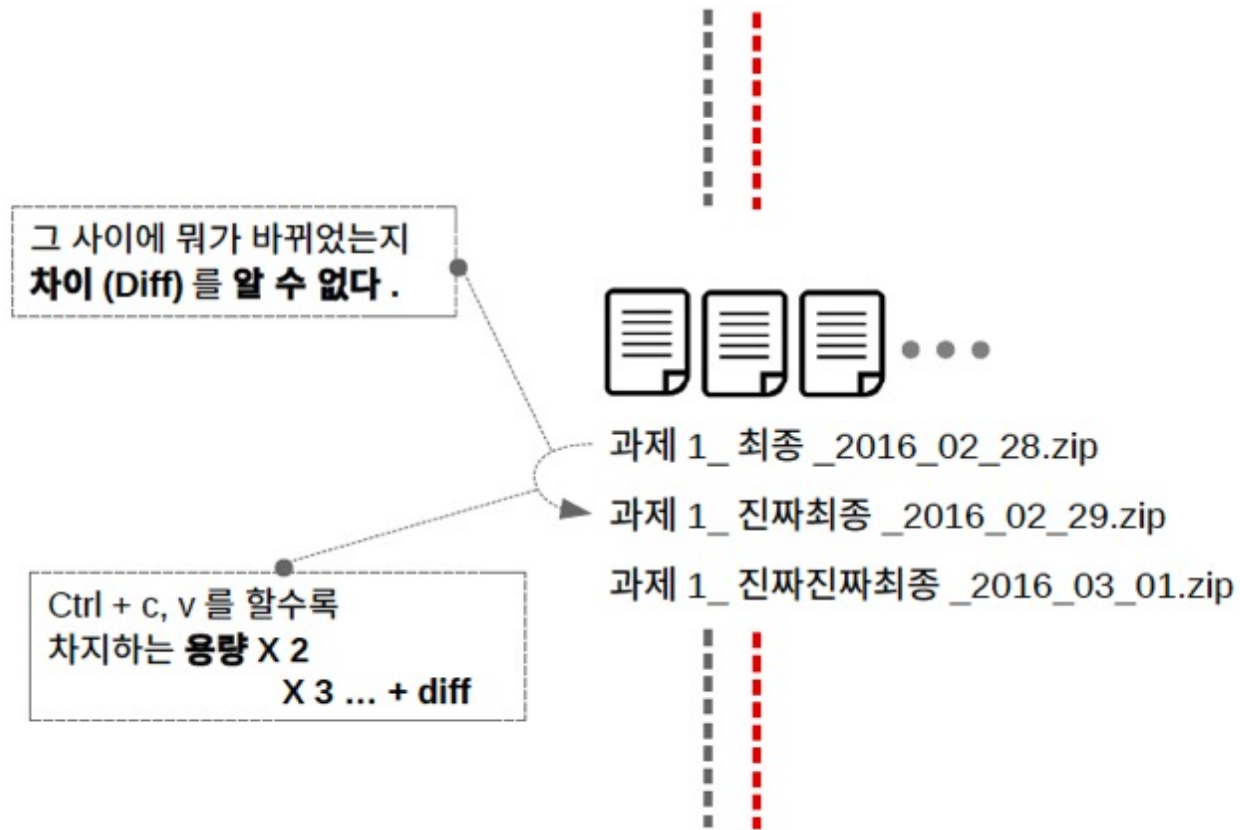


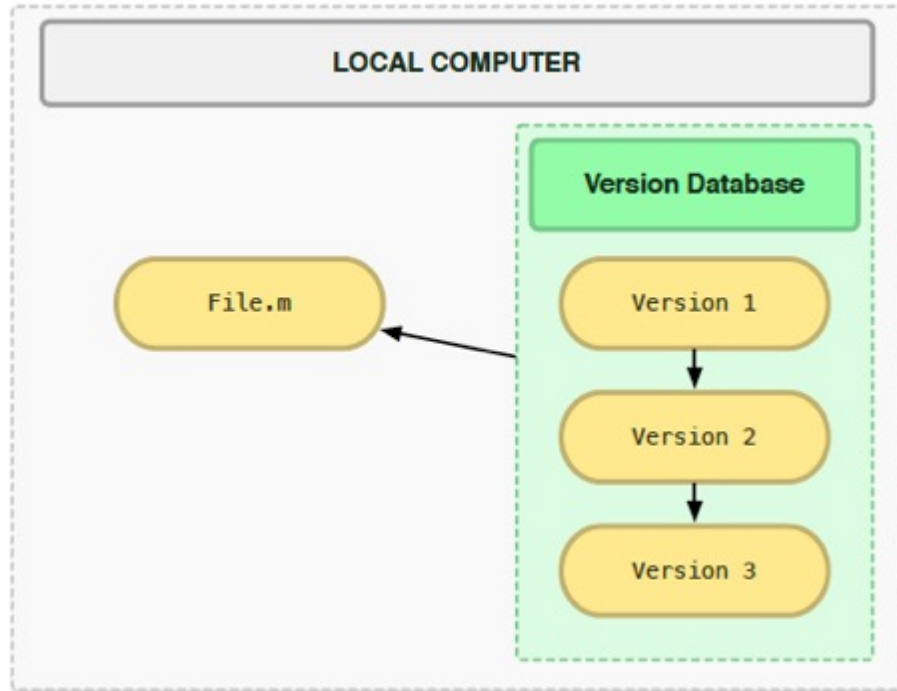
Advanced Git

Hyunsang Choi

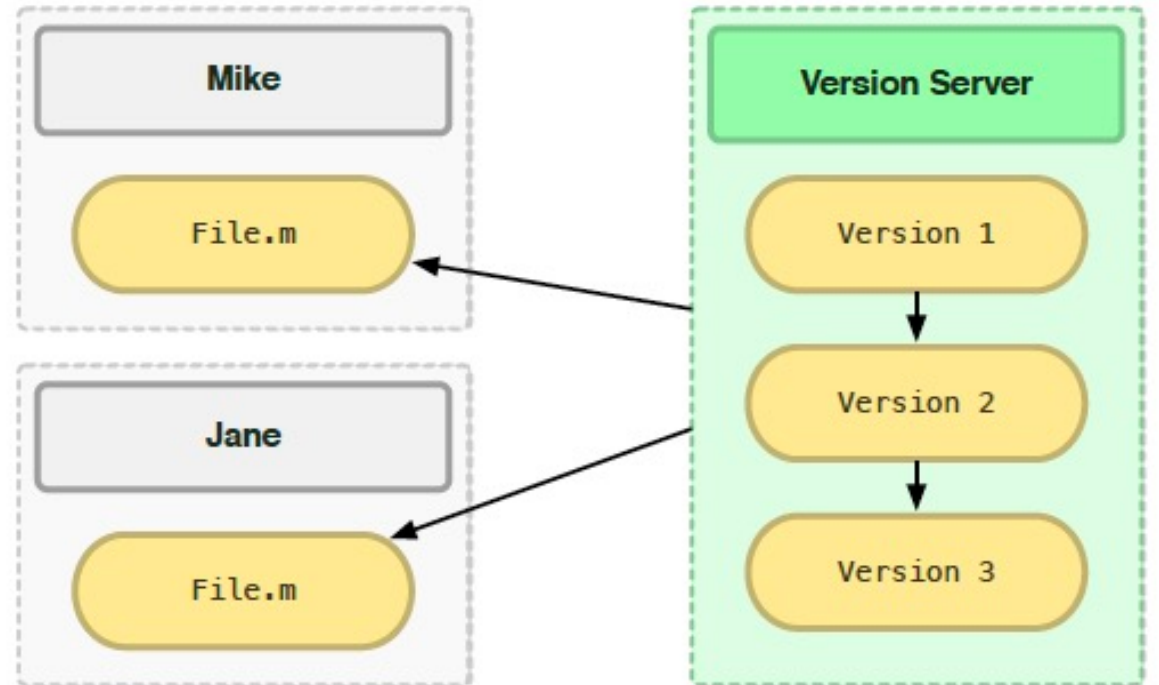
File Backup



Version Control Systems

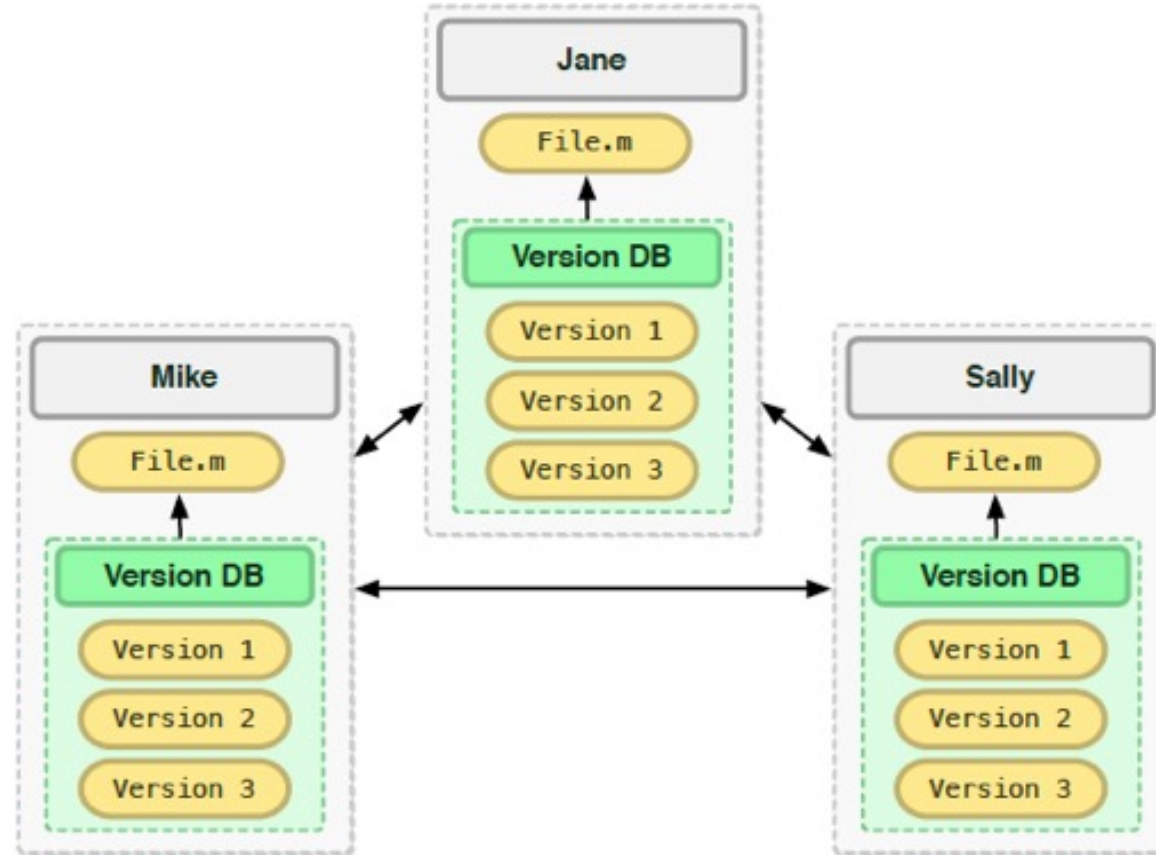


LVCS

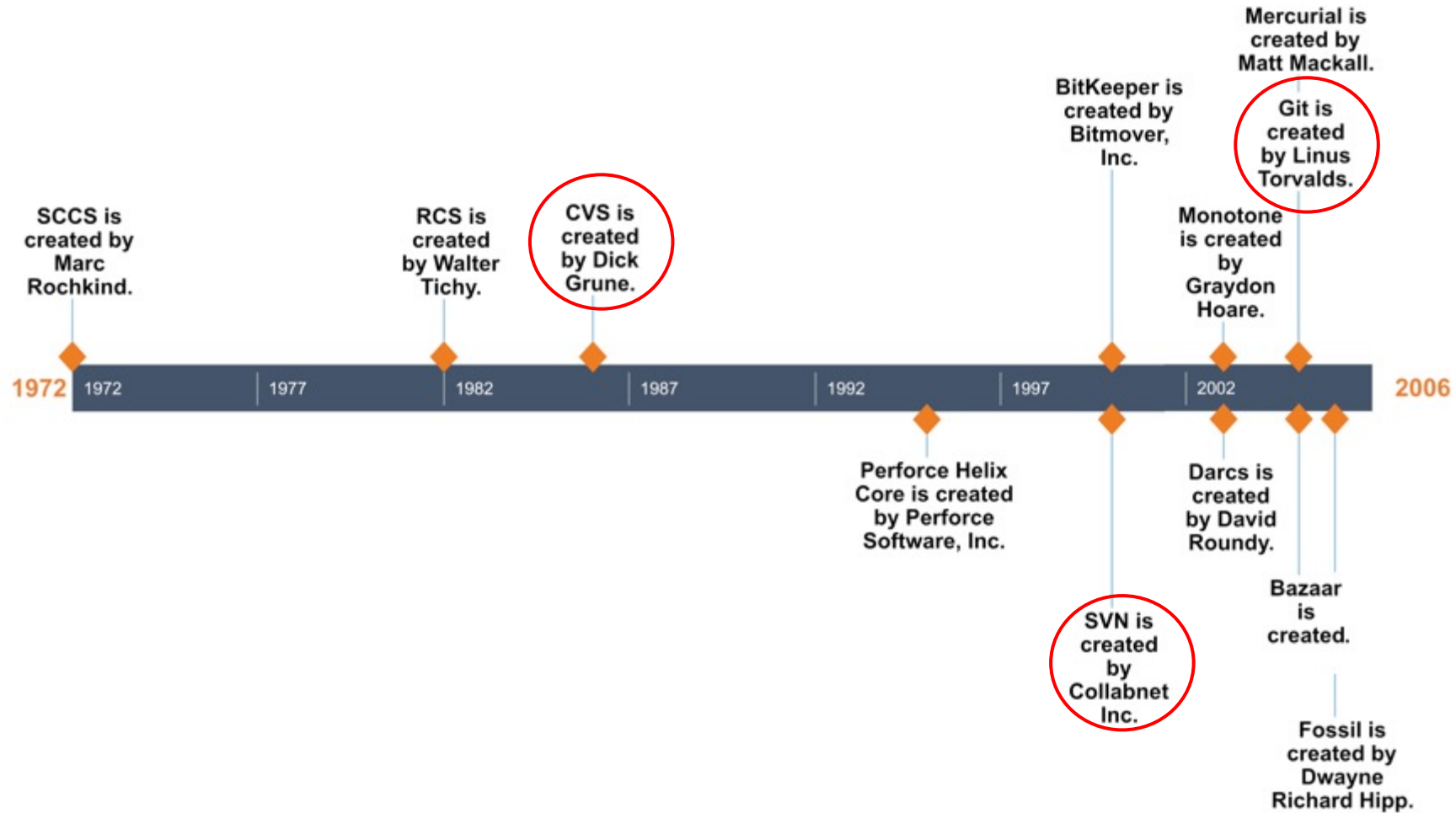


CVS, SVN

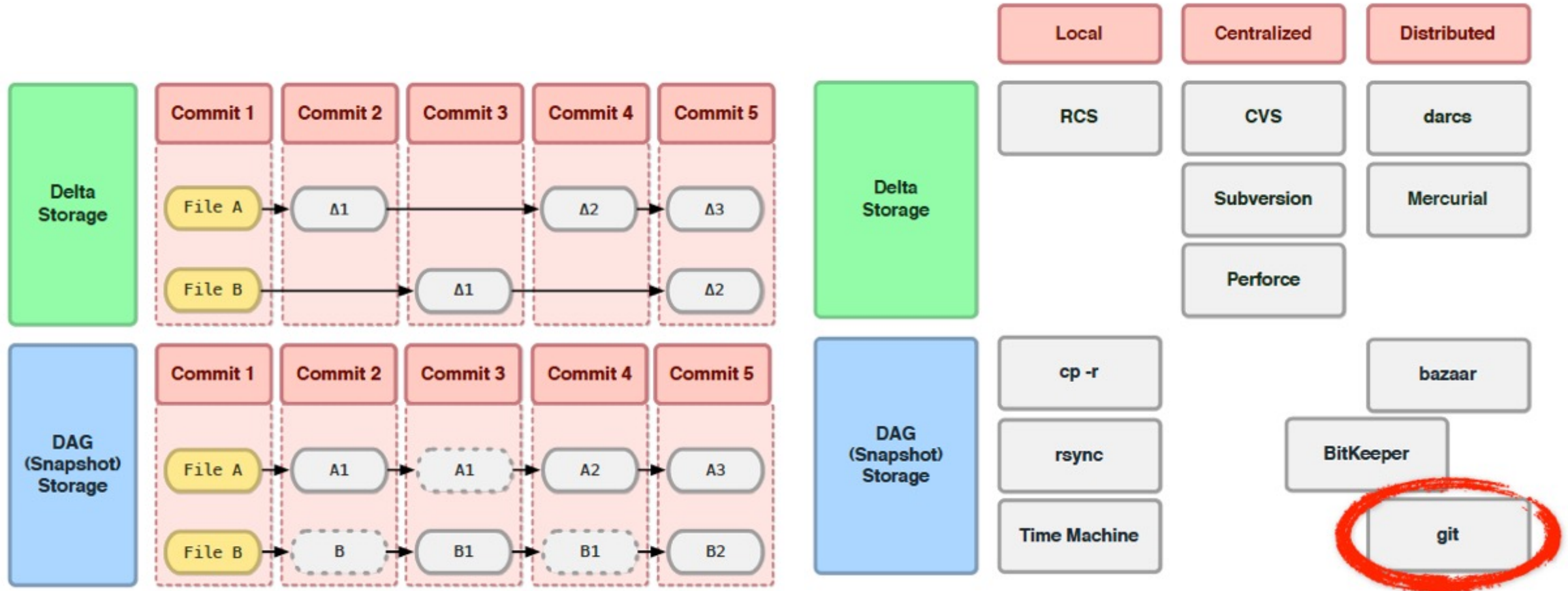
Distributed VCS



VCS History

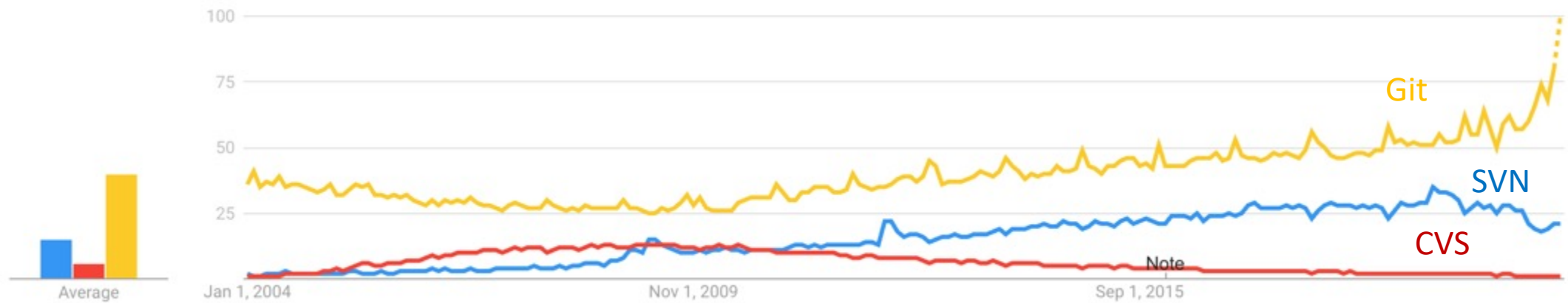


Delta vs Snapshot



Interest over Time

Interest over time ?



Git

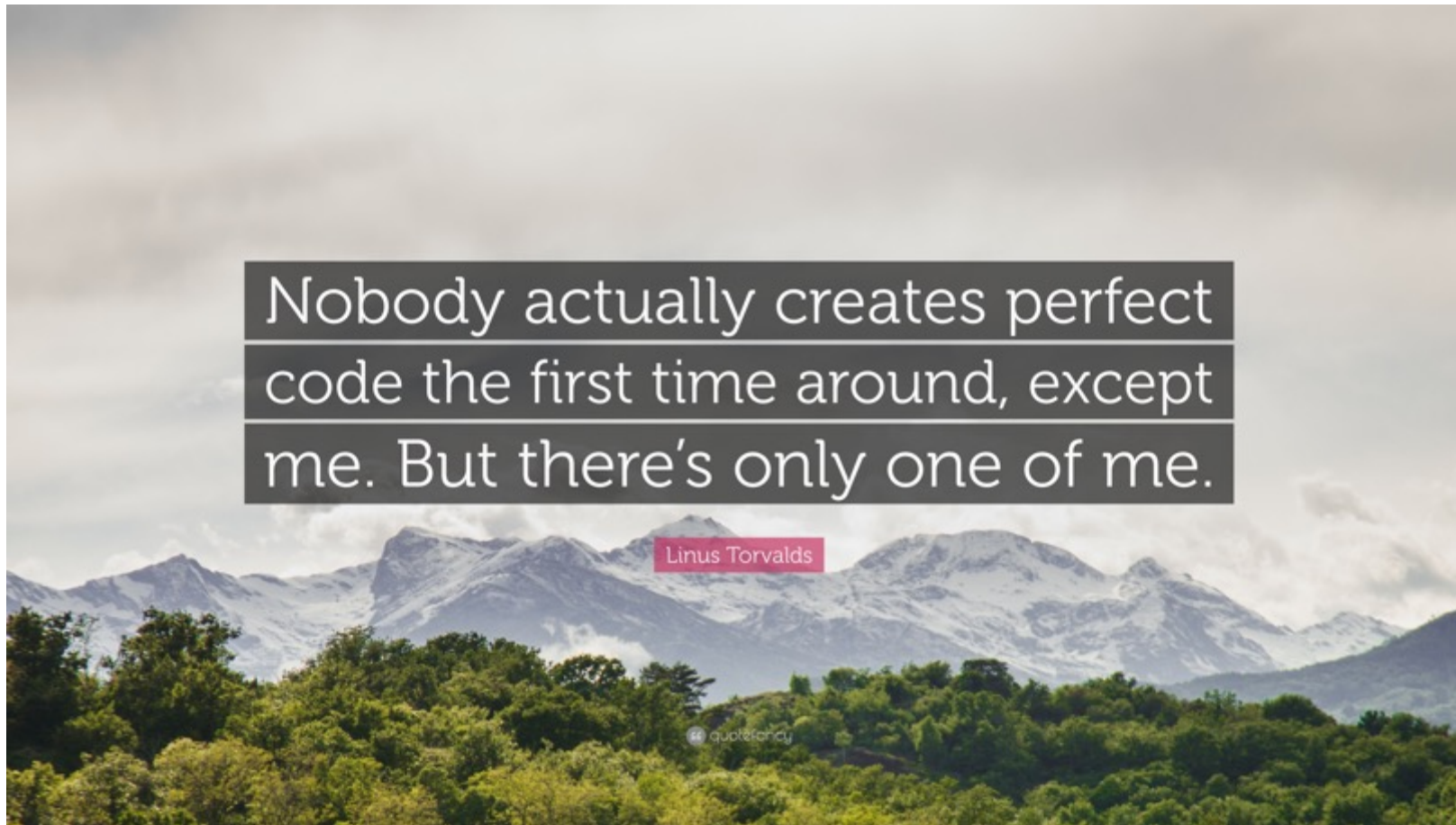
- *git* (which means "unpleasant person" in [British English](#) slang)
- Since 2005 (project's core maintainer: Junio Hamano)



Version	Original release date	Latest (patch) version	Release date (of the patch)
0.99	2005-07-11	0.99.9n	2005-12-15
1.0	2005-12-21	1.0.13	2006-01-27
2.17	2018-04-02	2.17.6	2021-03-09
2.18	2018-06-21	2.18.5	2021-03-09
2.30	2020-12-27	2.30.2	2021-03-09



Tech Talk: Linus Torvalds on Git at Google



Advantages

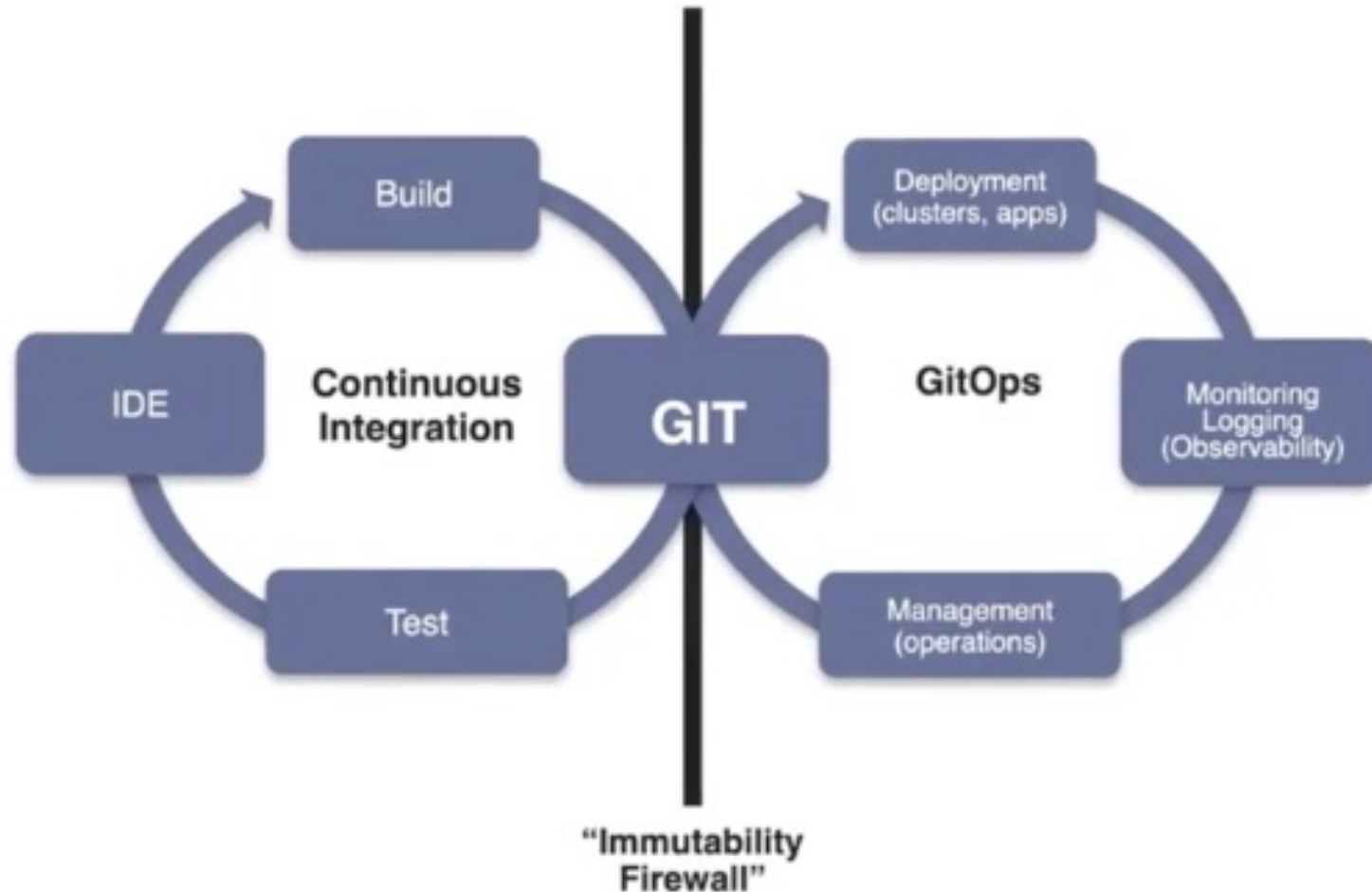
Everything is Fast

Everything is Transparent

Every Clone is a Backup

You Can Work Offline

GitOps: An Operating Model for Cloud Native



**Unifying Deployment,
Monitoring and Management.**

Git as the single source of truth
of a system's desired state

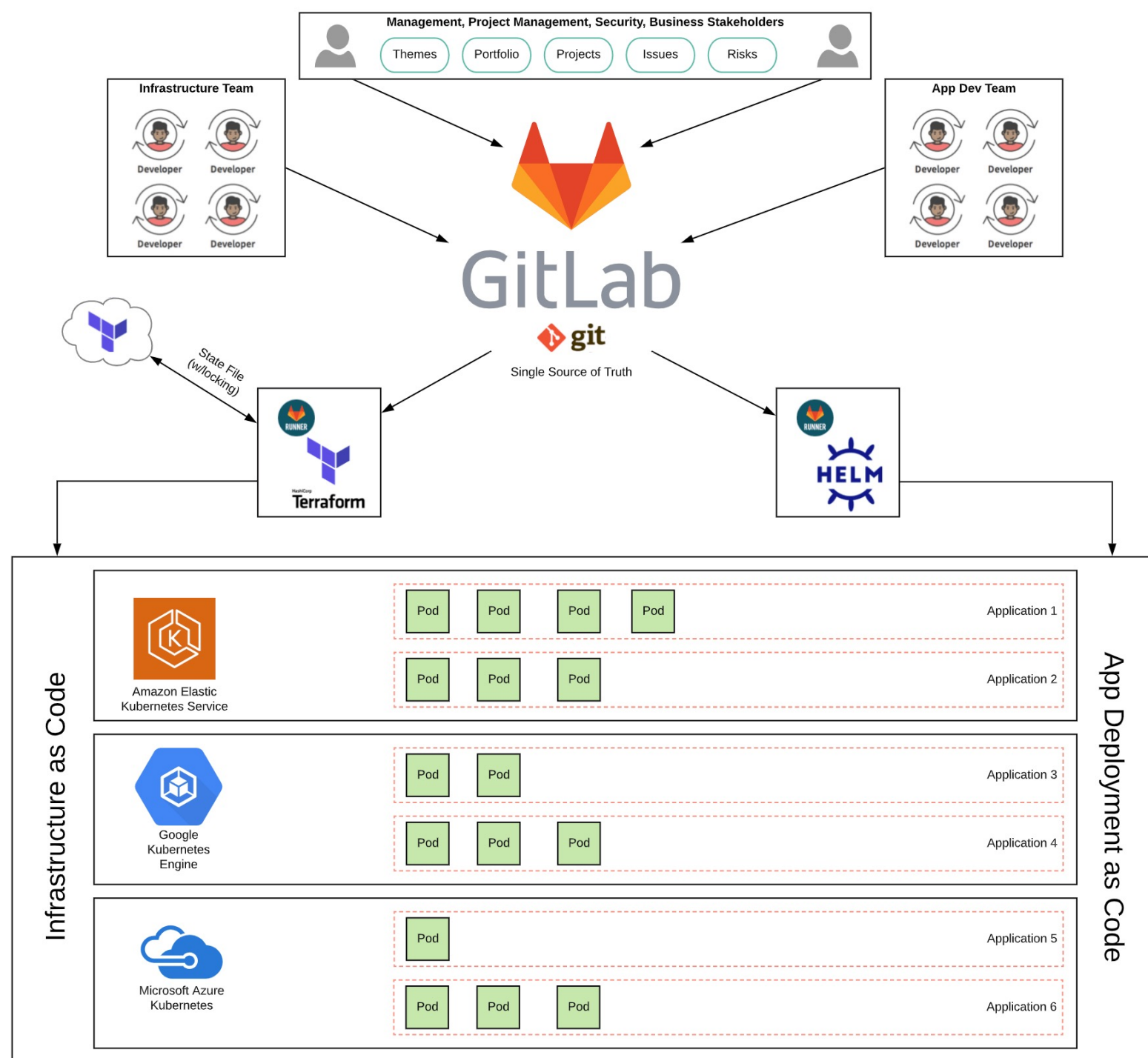
ALL intended operations are
committed by pull request

ALL diffs between intended and
observed state with automatic
convergence

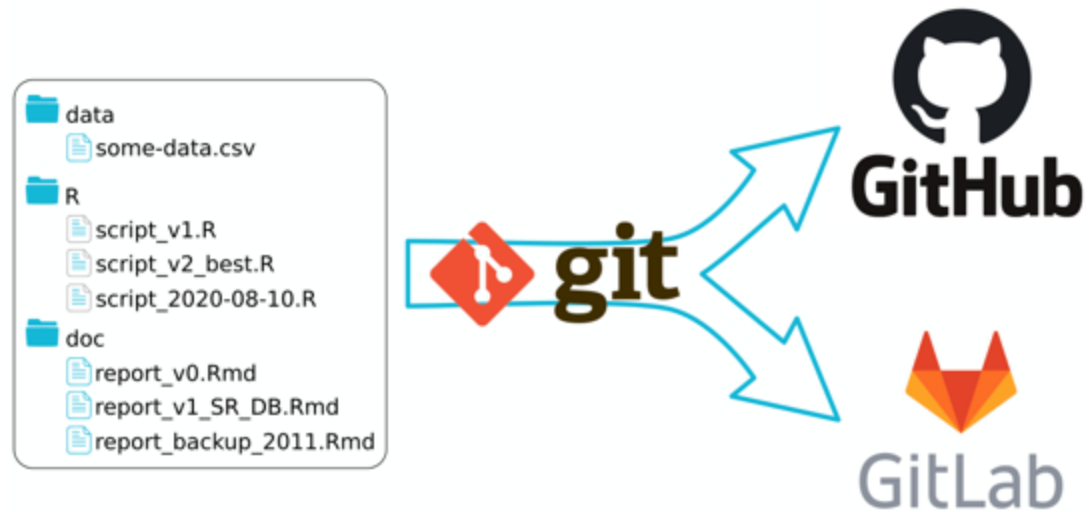
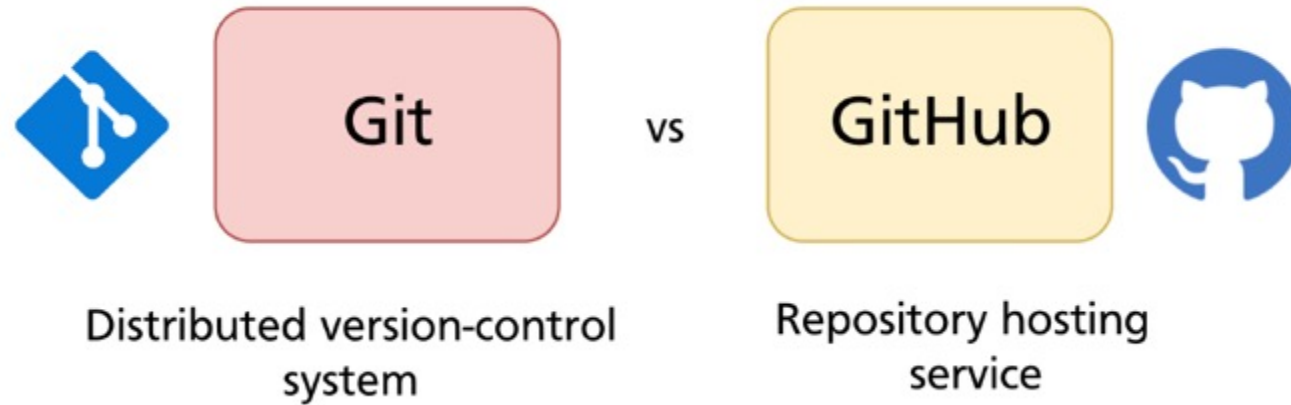
ALL changes are observable,
verifiable and auditable



GitOps Example



Git & Github



Github

- Since 2008: Developed by [Chris Wanstrath](#), [P. J. Hyett](#), [Tom Preston-Werner](#) and Scott Chacon
- [Ruby on Rails](#)
- Located in San Francisco
- June 4, 2018: Microsoft acquired
- 56 million users (Sep. 2020)



Github Numbers

Let's look back at the code and
communities built on GitHub this year...

Based on the data collection range of October 2019 – September 2020.

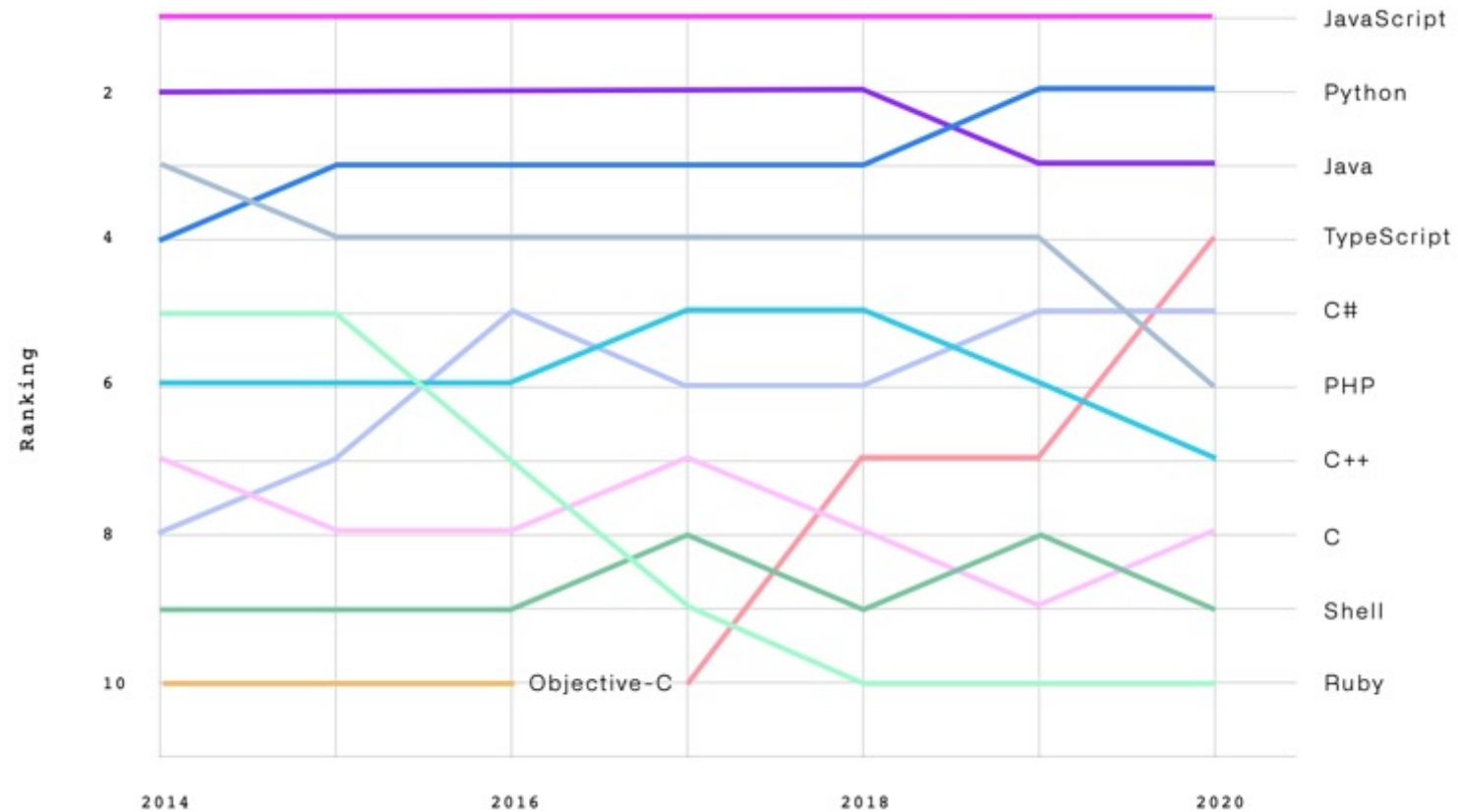
56 M+
total developers on
GitHub

60 M+
new repositories created
in the last year

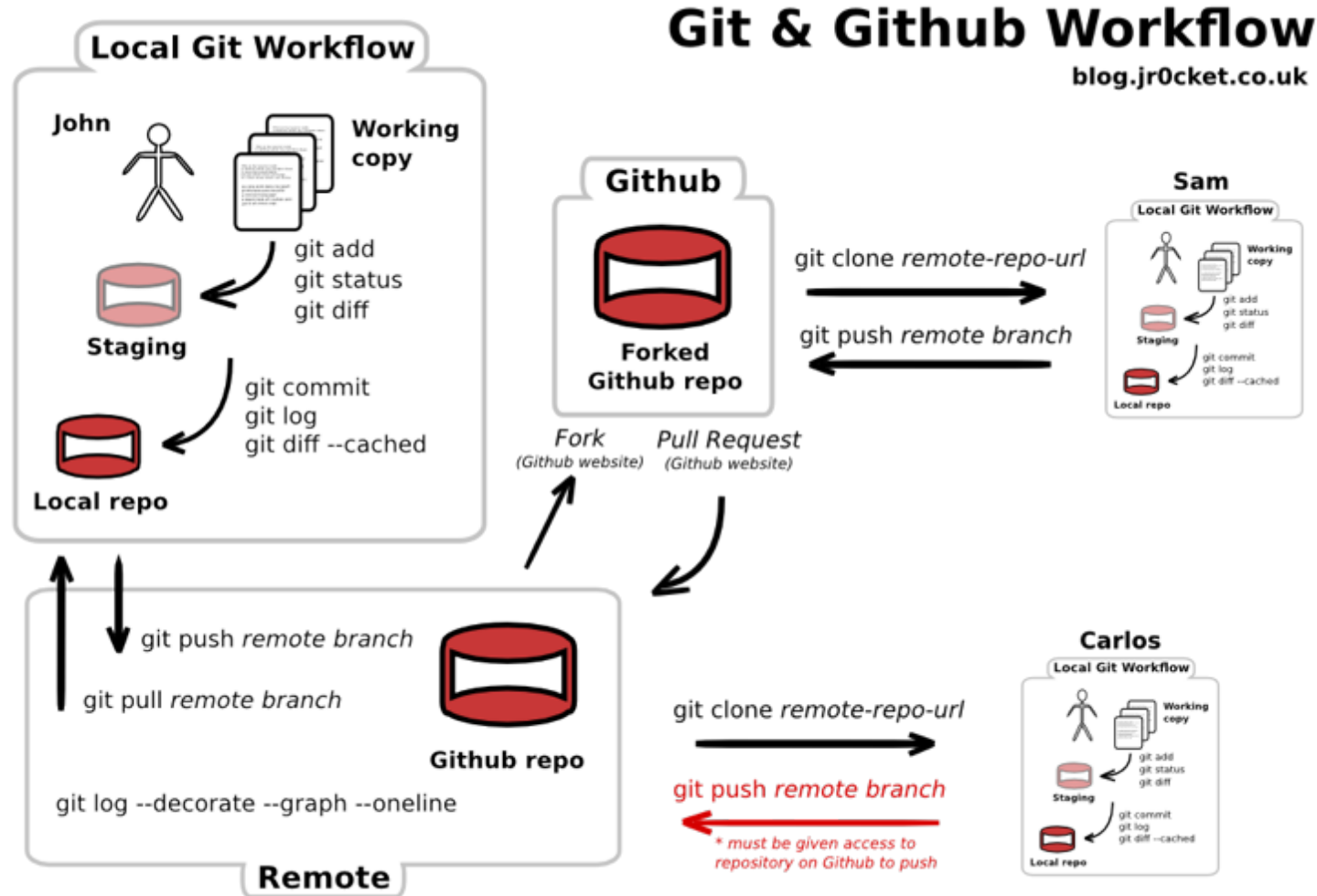
72 %
of Fortune 50 companies
use GitHub Enterprise

1.9 B+
contributions added
in the last year

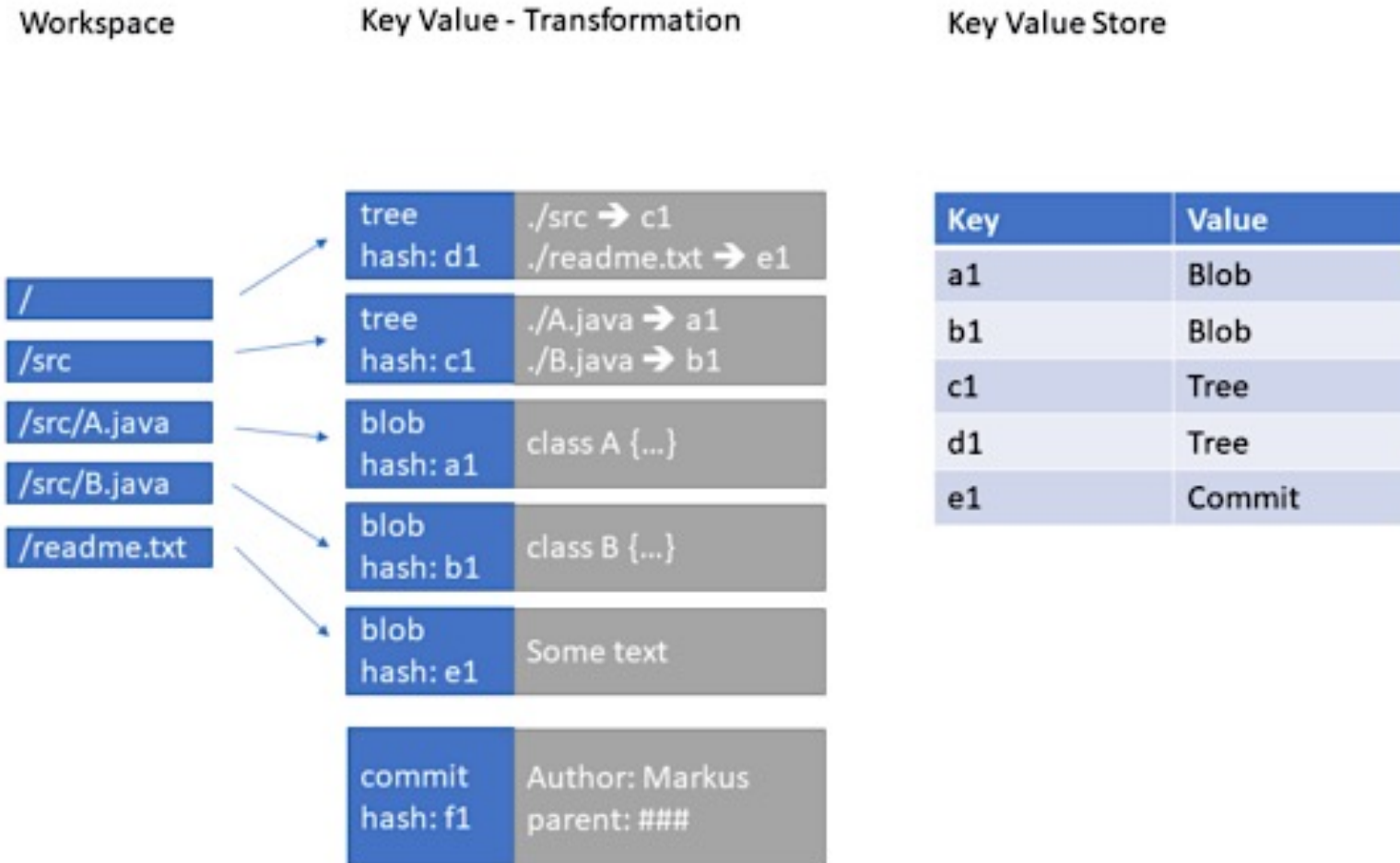
Language Ranking on Github



Typical Workflow



Git is a “Key-Value Store”





```
$ ls -lA
-rw-r--r--@ 1 pbhogan  staff  21508 Jul  3 15:21 .DS_Store
drwxr-xr-x 14 pbhogan  staff   476 Jul  3 14:06 .git
-rw-r--r--@ 1 pbhogan  staff   115 Aug 11 2010 .gitignore
-rw-r--r--@ 1 pbhogan  staff   439 Dec 27 2010 Info.plist
drwxr-xr-x 17 pbhogan  staff   578 Feb  6 10:54 Resources
drwxr-xr-x  7 pbhogan  staff   238 Jul 18 2010 Source
...
```

```
$ ls .git
branches/  description  hooks/  info/  objects/  refs/
config     FETCH_HEAD  HEAD    index  logs/     ORIG_HEAD
```

Configuration files

config	main config file
description	description of the repository
hooks/	hook scripts

Database (immutable objects)

objects/	commits, trees, blobs, tags
----------	-----------------------------

State files (mutable objects)

References (branches & tags)

refs/	raw store (one file per ref)
packed-refs	packed store (all refs in the same file)
branches/	remote references (deprecated ¹)

Special references

HEAD FETCH_HEAD ORIG_HEAD MERGE_HEAD CHERRY_PICK_HEAD

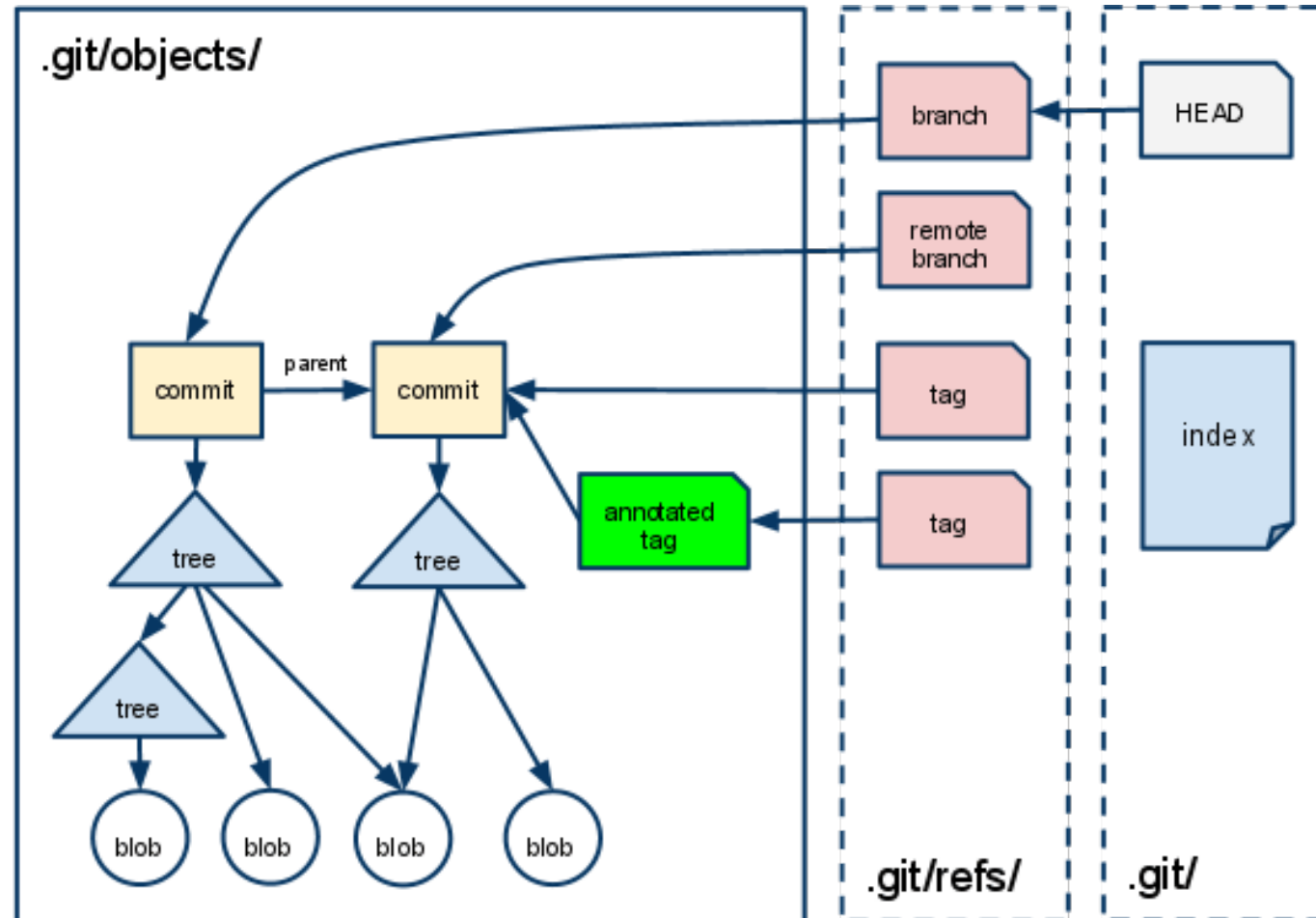
Index

index	state of the index
-------	--------------------

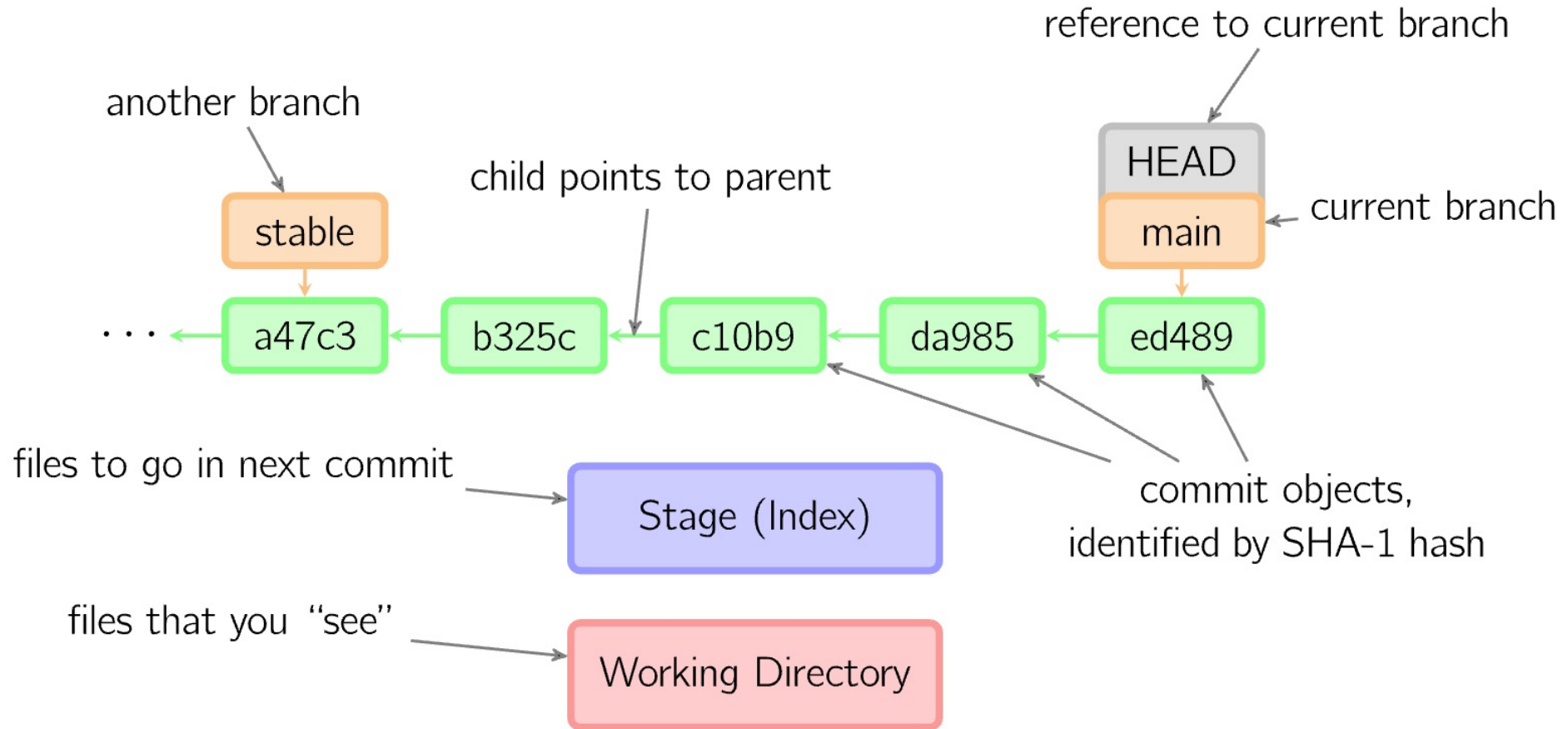
Logs	logs/	history of branches
-------------	-------	---------------------

Misc	info/	config files (exclude attributes) and cached state (refs)
-------------	-------	---

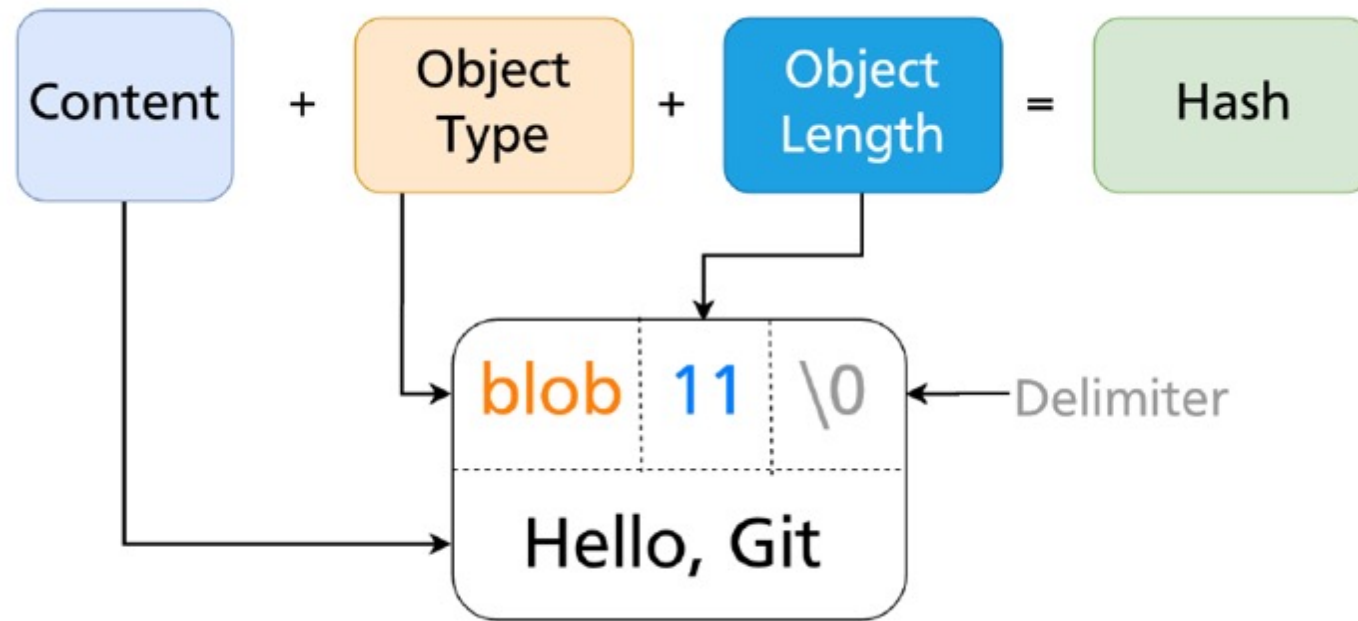
Git Architecture



Conventions

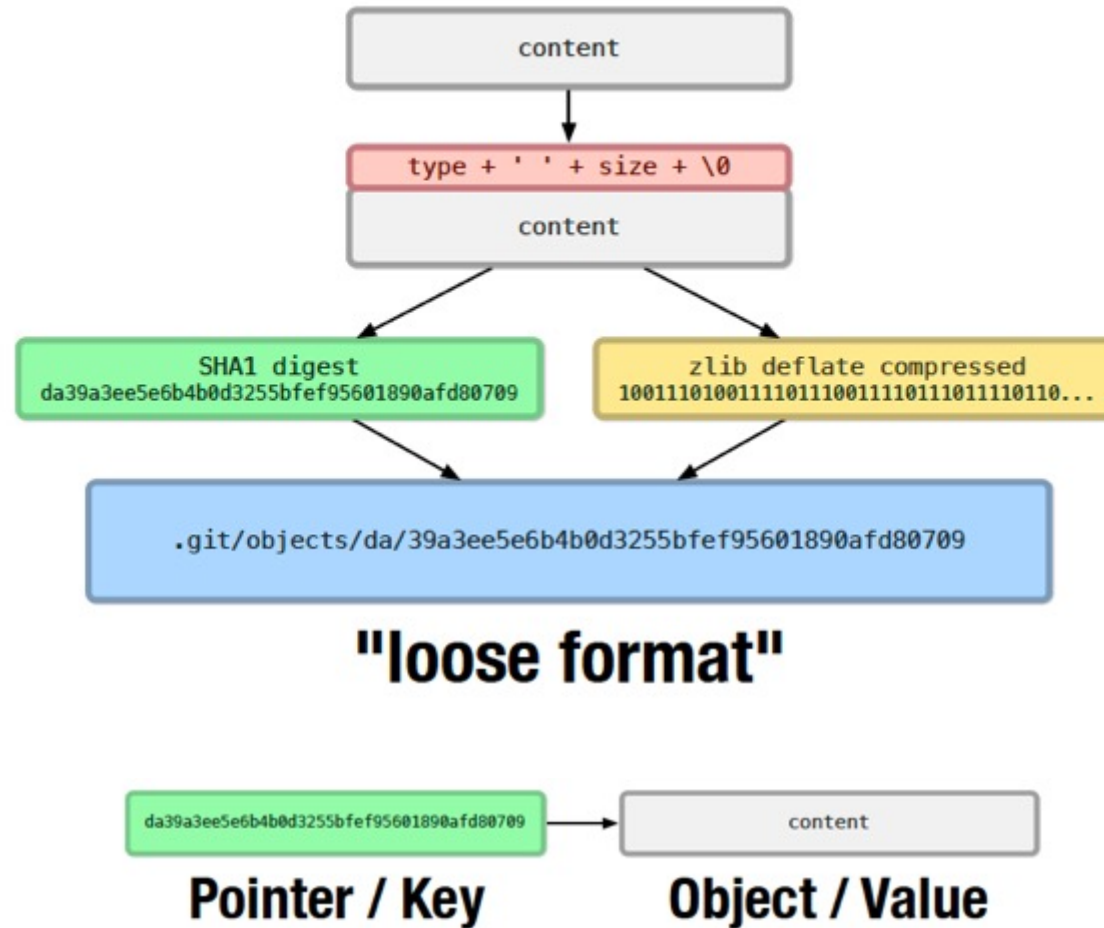


Git Object





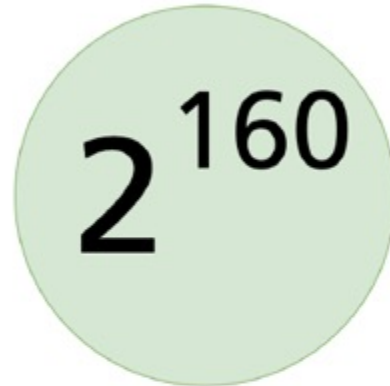
Object Database



Quiz

- How many files git can store in the same repository?

SHA1 combinations


$$2^{160}$$

1461501637330902918203684832716283019655932542976

Unique Hash

da39a3ee5e6b4b0d3255bfef95601890afd80709

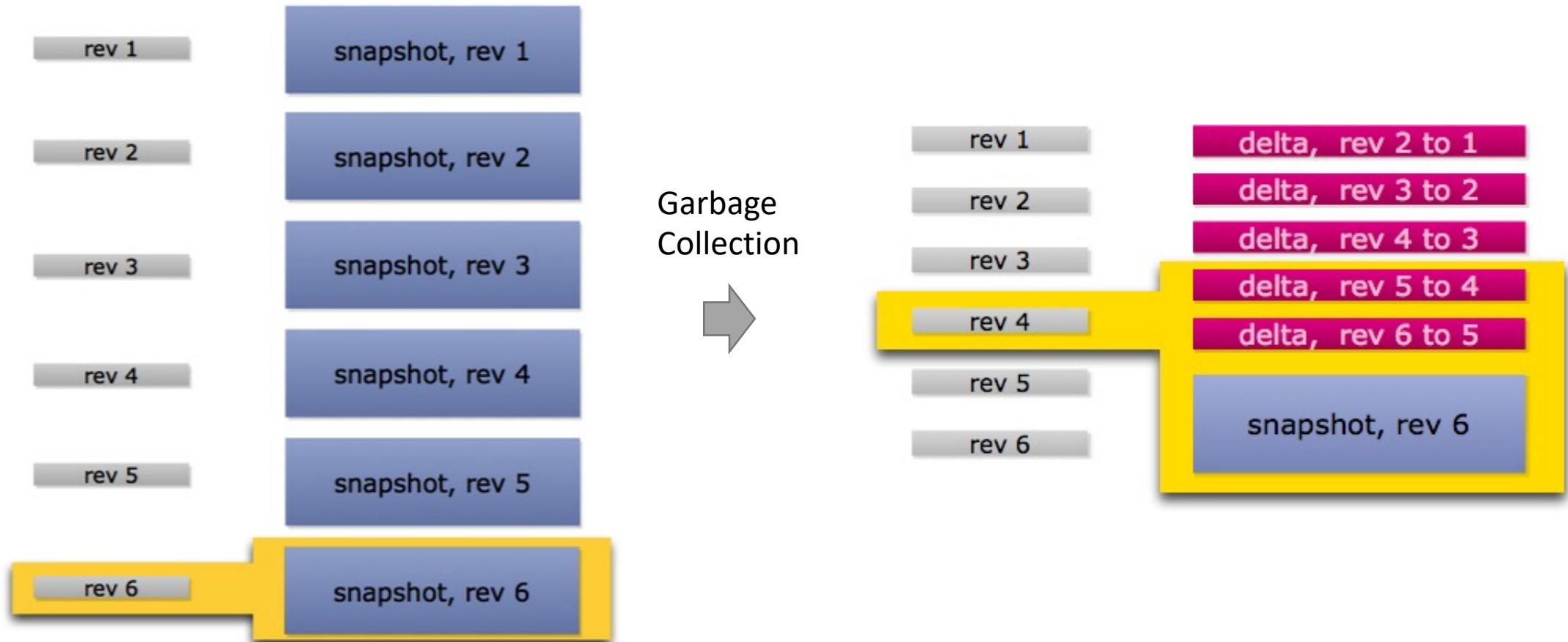
da39a3ee5e6b4b0d3255...

da39a3ee5e6...

da39a...

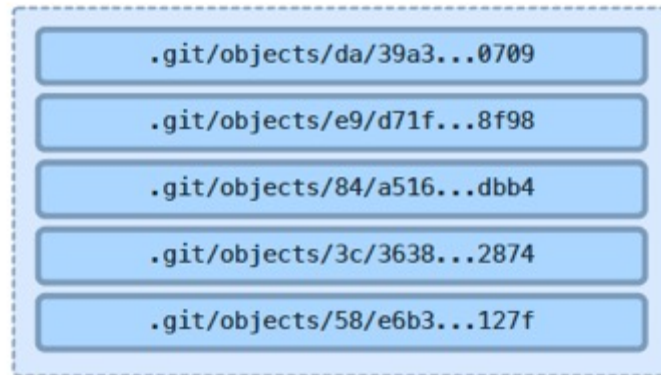
**Equivalent if common
prefix is unique!**

Snapshot & Delta

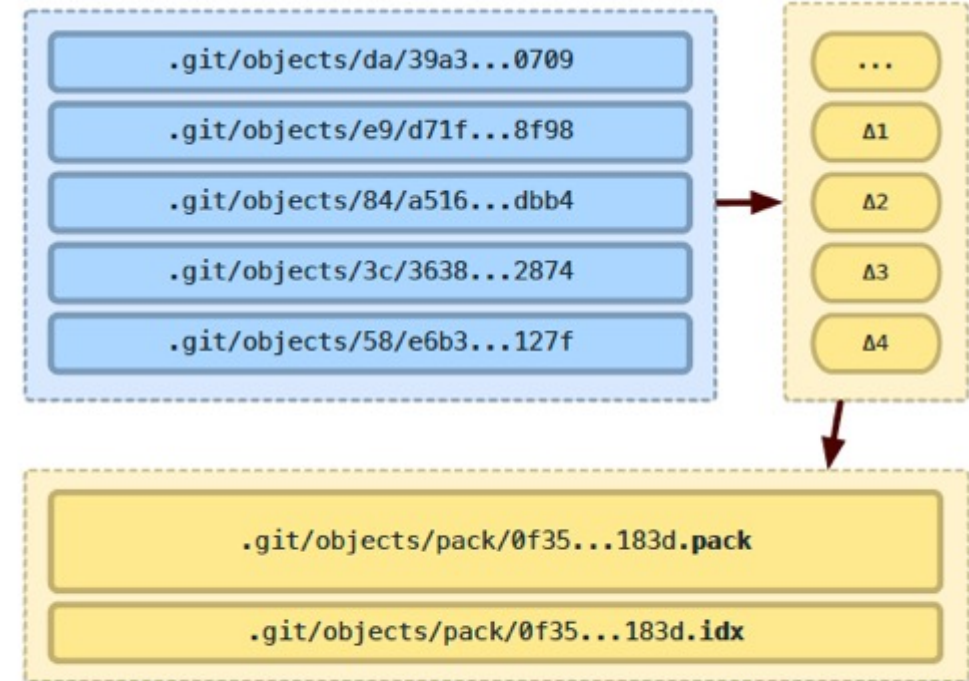


Garbage Collection

Similar Objects



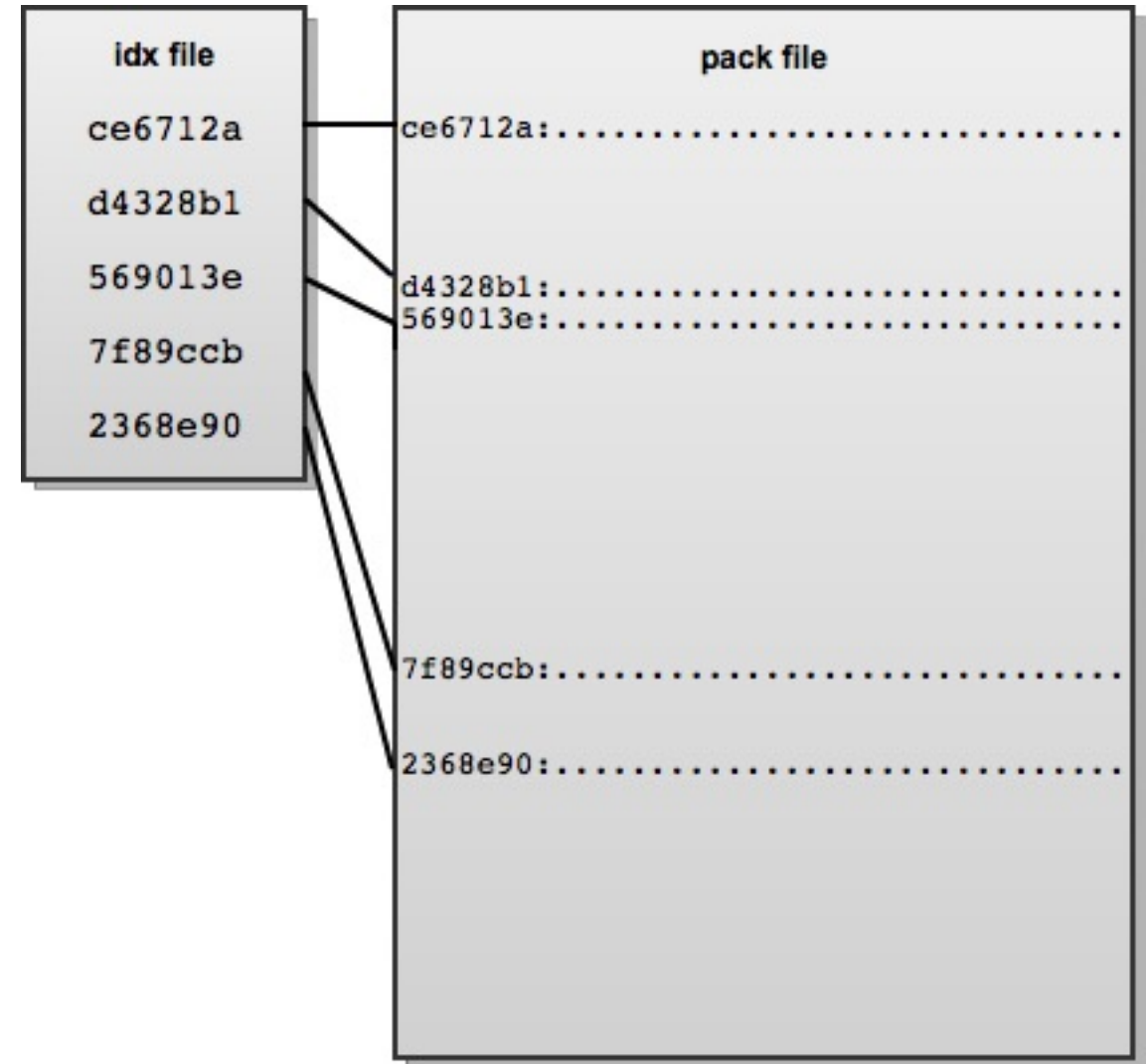
git gc



"packed format"

Packfile

- Packfile?
 - Git packs up several of these objects into a single binary file called a “packfile” in order to save space and be more efficient
- When?
 - Too many loose objects around
 - git gc command manually
 - Push to a remote server



git cat-file options

git cat-file options

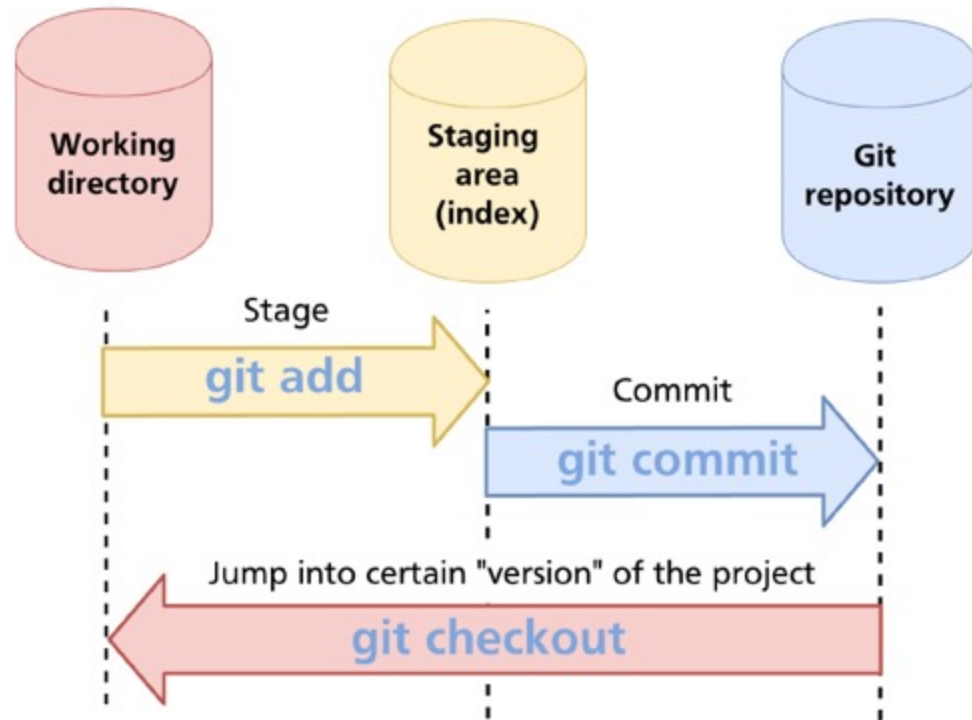
`git cat-file -p <hash>` Contents of the object

`git cat-file -s <hash>` Size of the object

`git cat-file -t <hash>` Type of the object

git ls-files

git ls-files -s



040000	Directory
100644	Regular non-executable file
100664	Regular non-executable group-writeable file
100755	Regular executable file
120000	Symbolic link
160000	Gitlink

Low-Level Commands (Plumbing Commands)

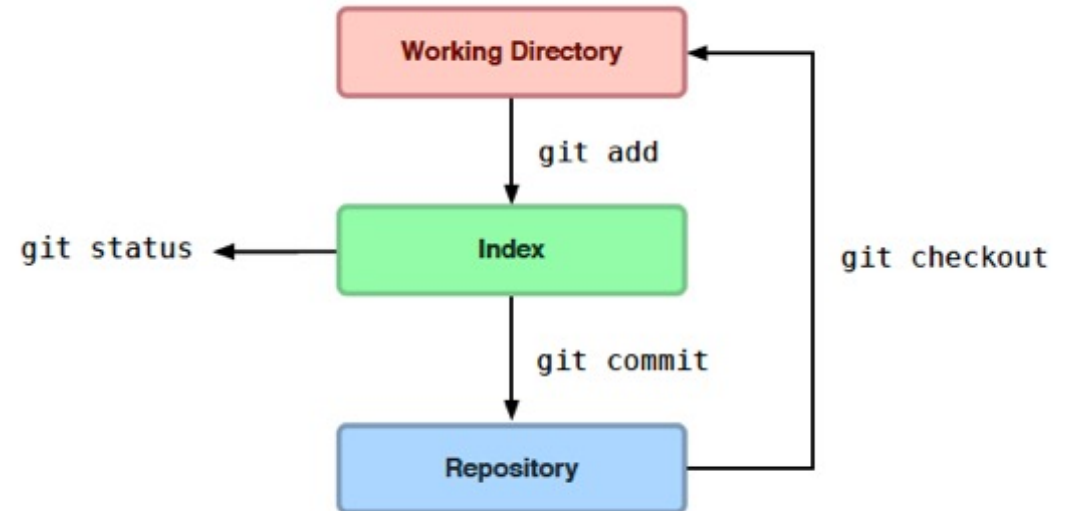
<code>git cat-file type object</code>	read a DB entry
<code>git hash-object -w -t type file</code>	write a DB entry
<code>git ls-tree tree</code>	read a tree object (txt format)
<code>git mk-tree</code>	make a tree object (from txt format)
<code>git diff-tree tree1 tree2</code>	show the differences between two trees
<code>git commit-tree -p parent -m message tree</code>	write a new commit object
<code>git merge-tree base-tree left-tree right-tree</code>	prepare a 3-way merge

Index

== Staging Area

Why staging?

No Need To Commit All At Once
Pick (Stage) Logical Units to Commit
Helps You Review Your Changes
Lets You Write Your History Cleanly





Index File



Object Types



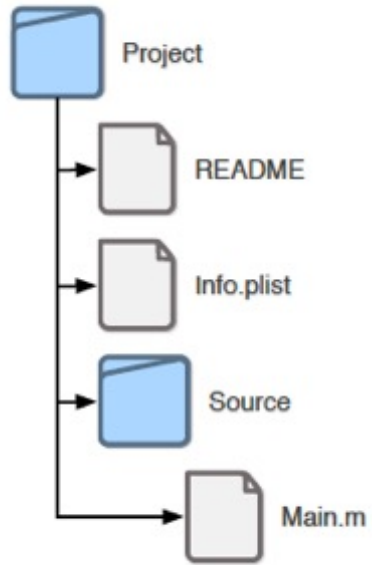
blob

```
$ echo 'Hello World!' | git hash-object --stdin -t blob -w  
980a0d5f19a64b4b30a87d4206aade58726b60e3
```

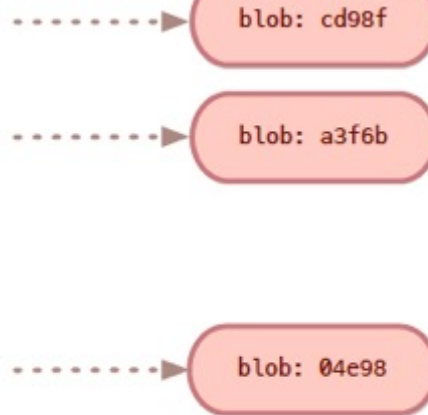
```
$ git cat-file blob 980a0d5f19a64b4b30a87d4206aade58726b60e3  
Hello World!
```

blob

Working Directory



Git Directory



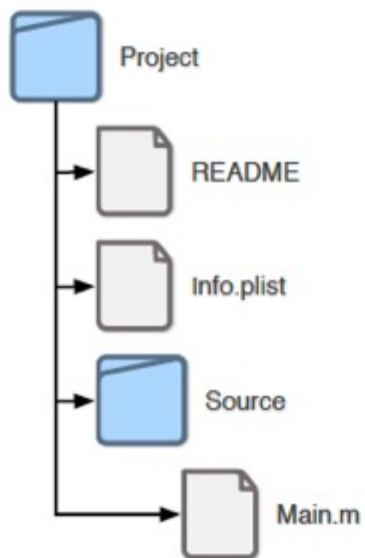
blob 109\0

```
#import <Cocoa/Cocoa.h>  
  
int main(int argc, const char *argv[])  
{  
    return NSApplicationMain(argc, argv);  
}
```

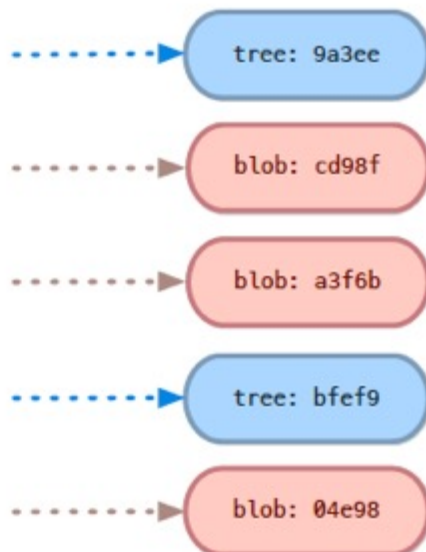
tree

tree

Working Directory



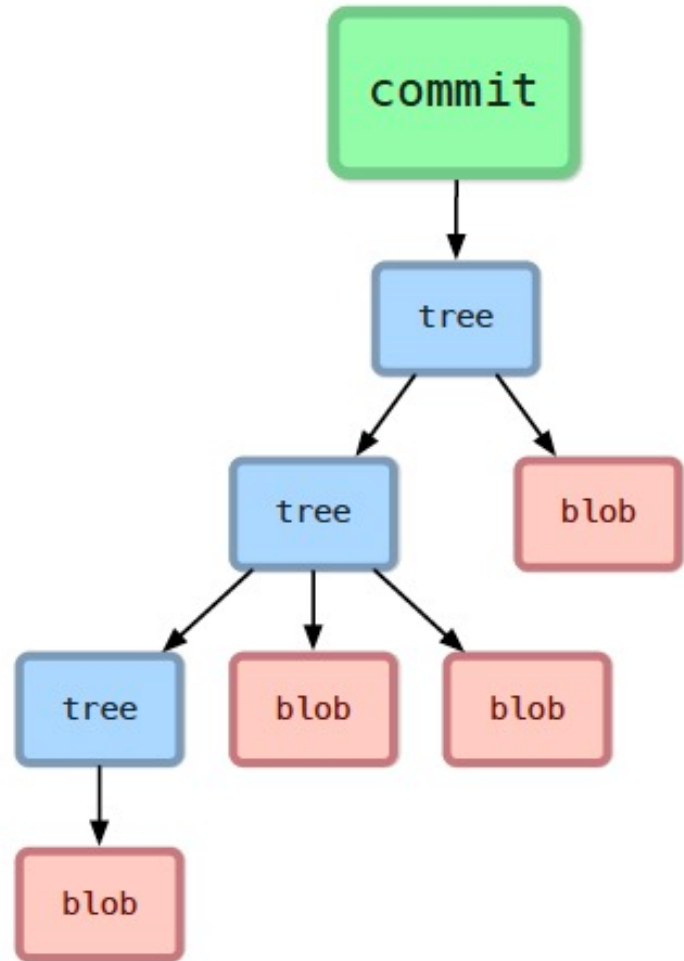
Git Directory



tree 84\0

```
100644 blob cd98f  README
100644 blob a3f6b  Info.plist
040000 tree bfef9  Source
```

commit



```
commit 155\0
```

```
tree 9a3ee
parent fb39e
author Patrick Hogan <pbhogan@gmail.com> 1311810904
committer Patrick Hogan <pbhogan@gmail.com> 1311810904
```

```
Fixed a typo in README.
```


tag

tag

tag 121\0

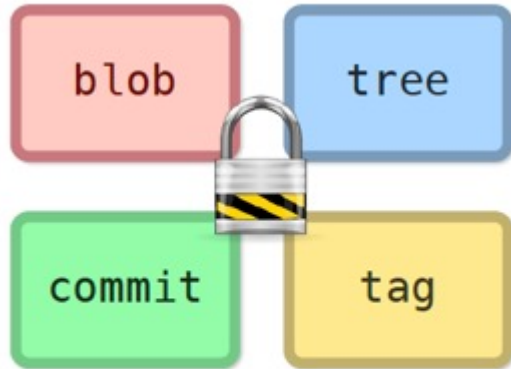
```
object e4d23e
type commit
tag v1.2.0
tagger Patrick Hogan <pbhogan@gmail.com> 1311810904

Version 1.2 release -- FINALLY!
```

.git/objects/20/c71174453dc760692cd1461275bf0cffeb772f

.git/refs/tags/v1.2.0

Immutable



Immutable!

Never Removes Data
(Almost)

```
$ git commit -m "some changes"  
.. oops! one file is missing ..  
$ git add new_file  
$ git commit --amend
```

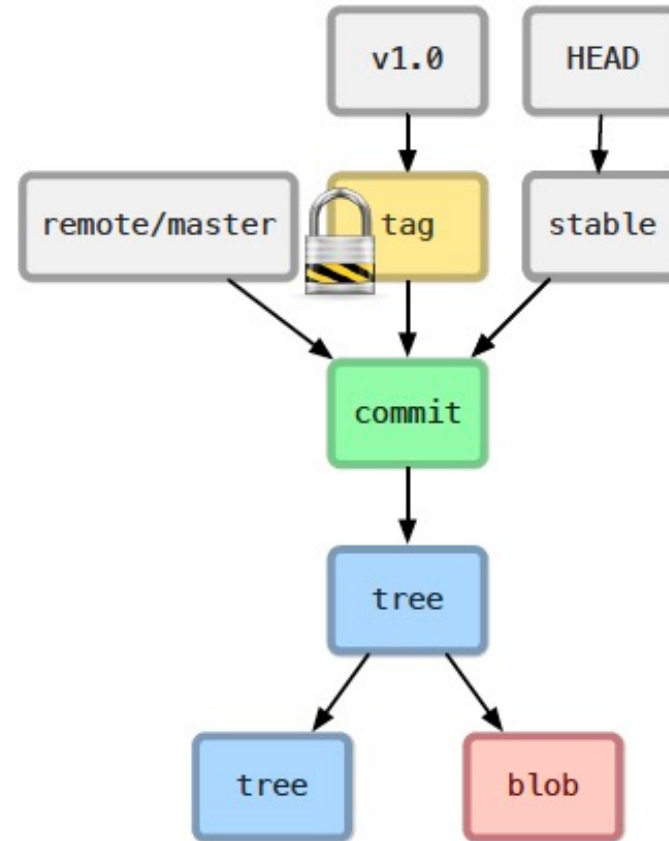
`git reset --hard HEAD~1`

References

**Lightweight, Movable
Pointers to Commits**
(and other things)

A branch is a named reference to a commit

HEAD points to the branch we are currently on

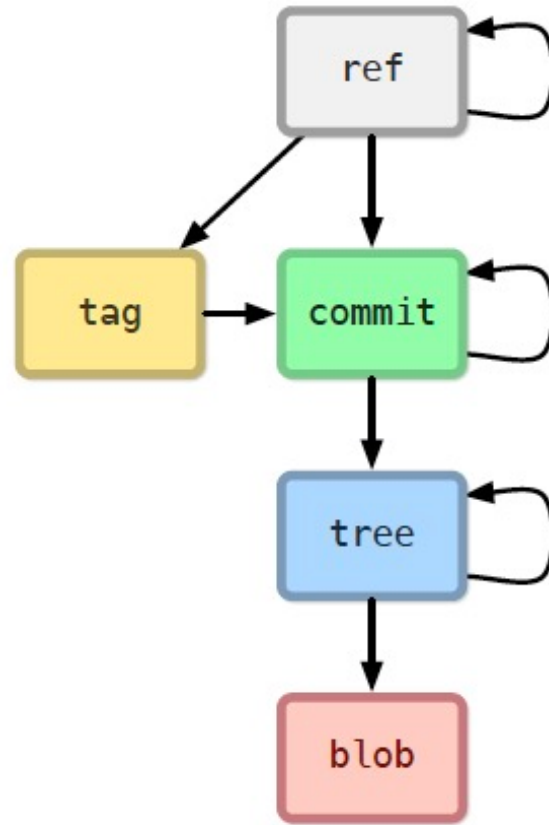


References

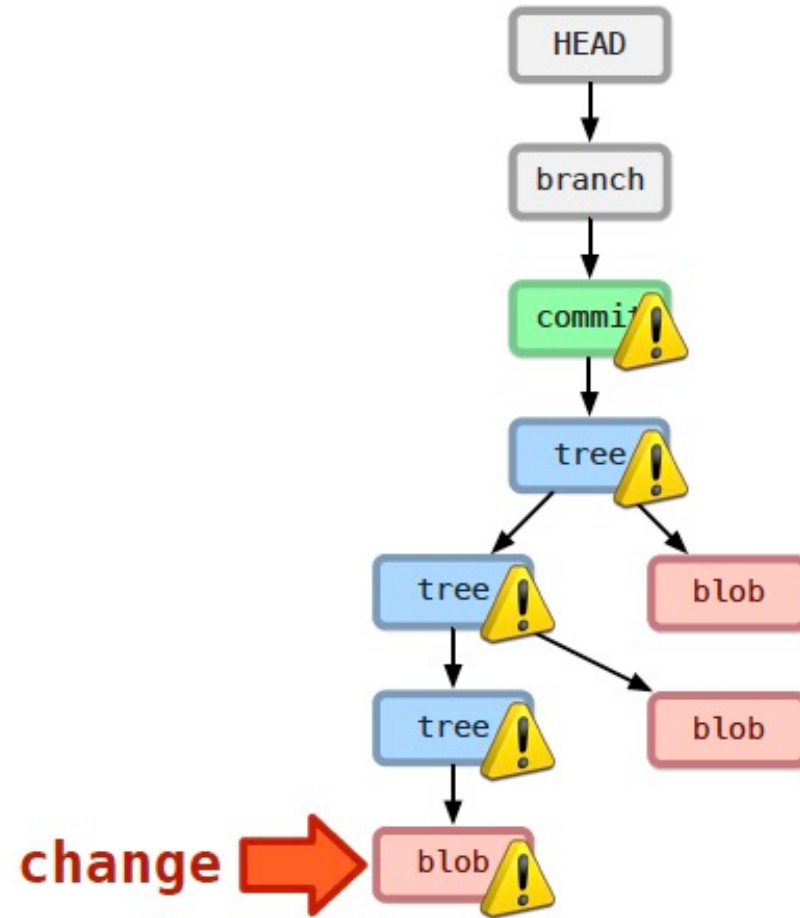
1. special ref (.git/*)
HEAD
2. raw reference (.git/refs/*)
heads/master
tags/v1.0
remotes/origin/master
remotes/origin/HEAD
3. tag (.git/refs/tags/*)
v1.0
4. local branch (.git/refs/heads/*)
master
5. remote branch (.git/refs/remotes/*)
origin/master
6. HEAD of a remote repository (.git/refs/remotes/*/HEAD)
origin

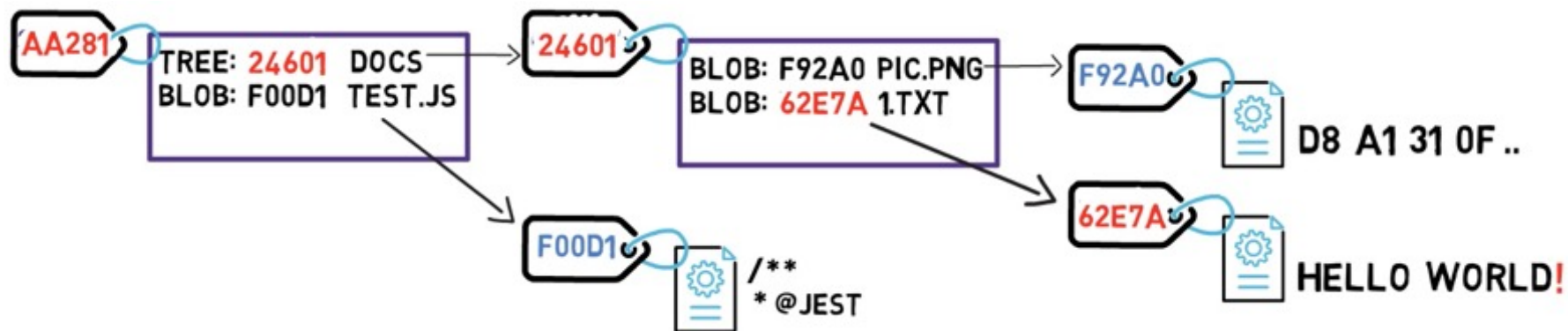
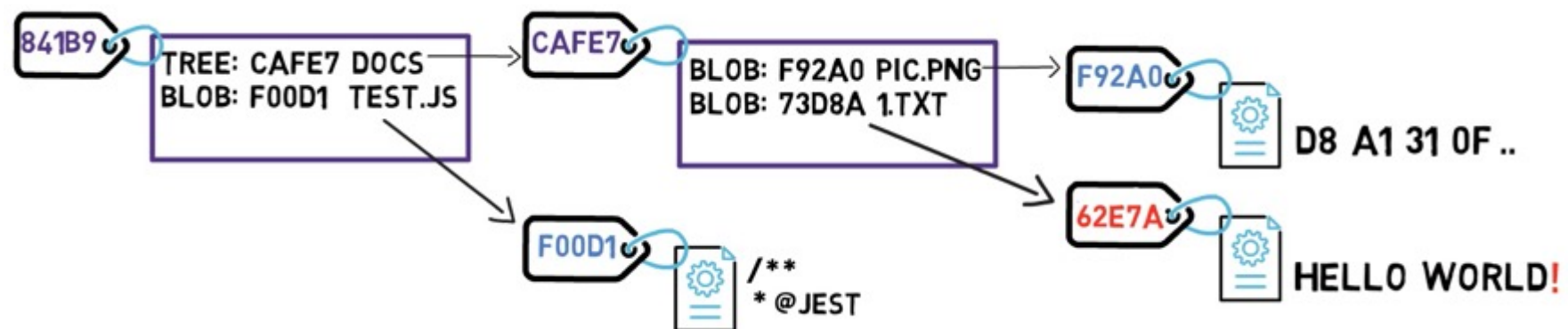
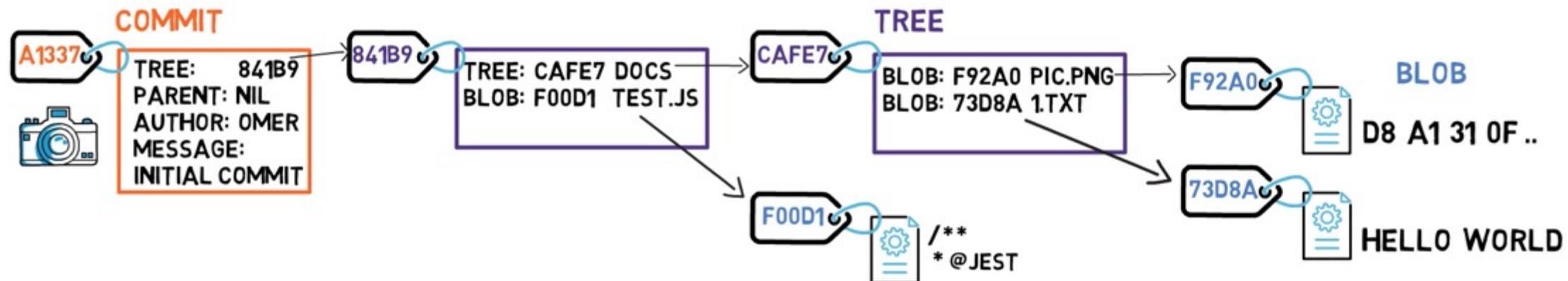


Git Object Model (Pointing)

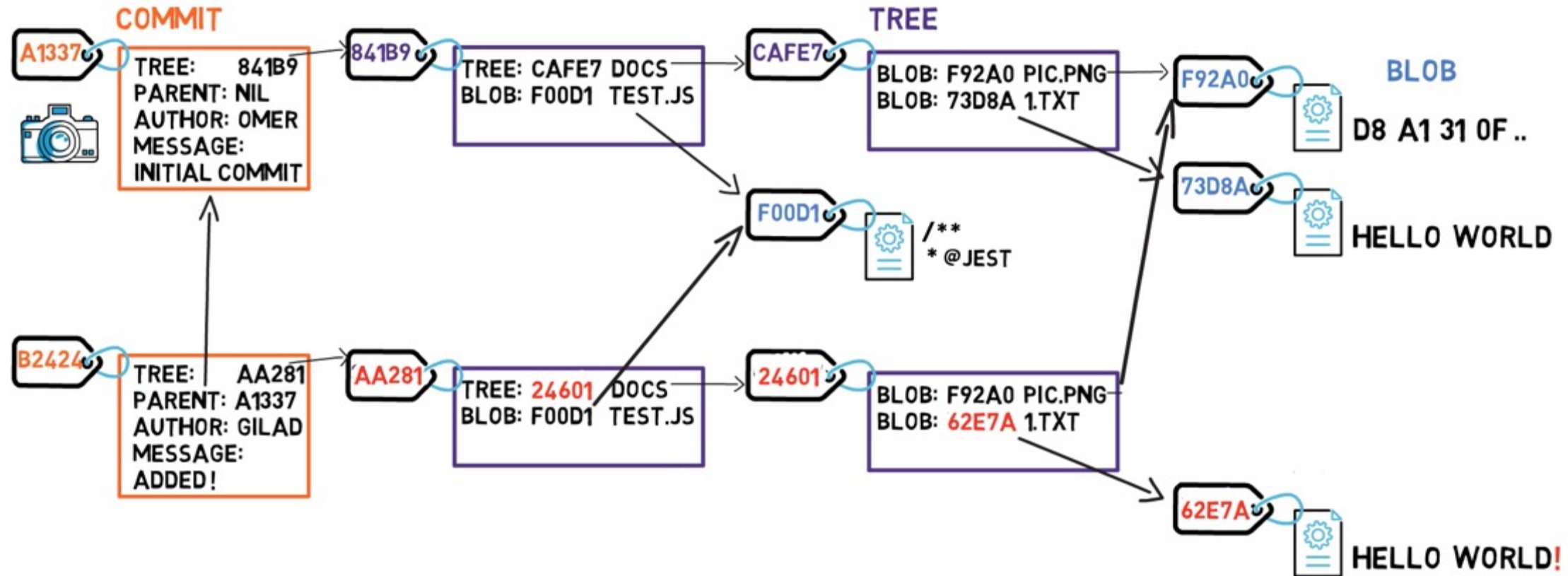


New Commit

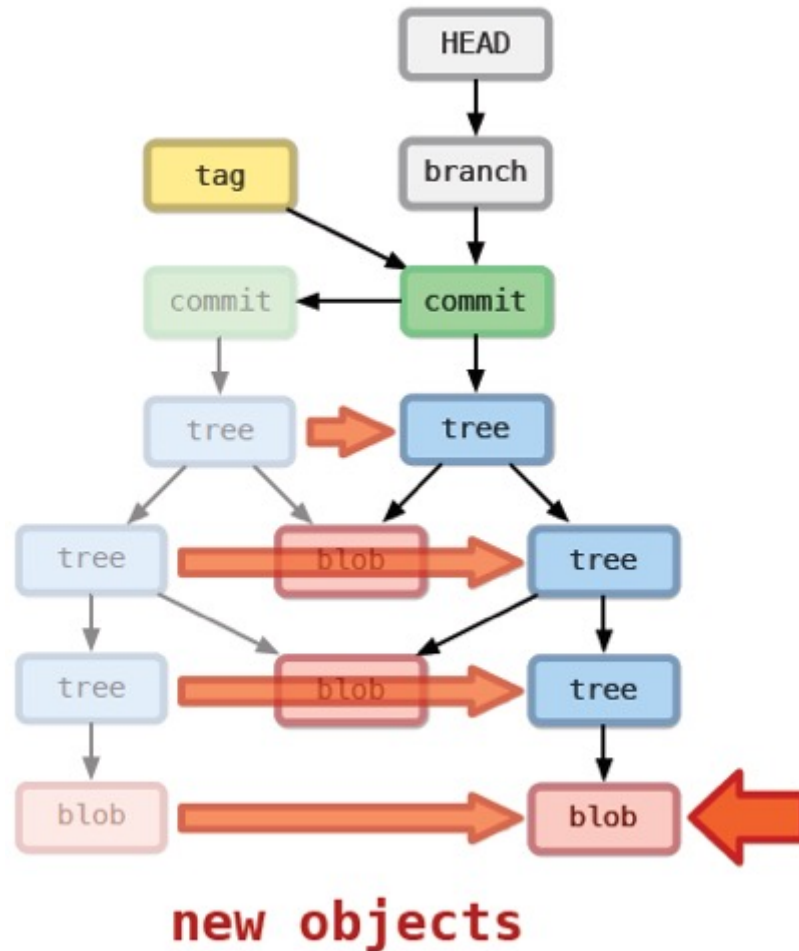




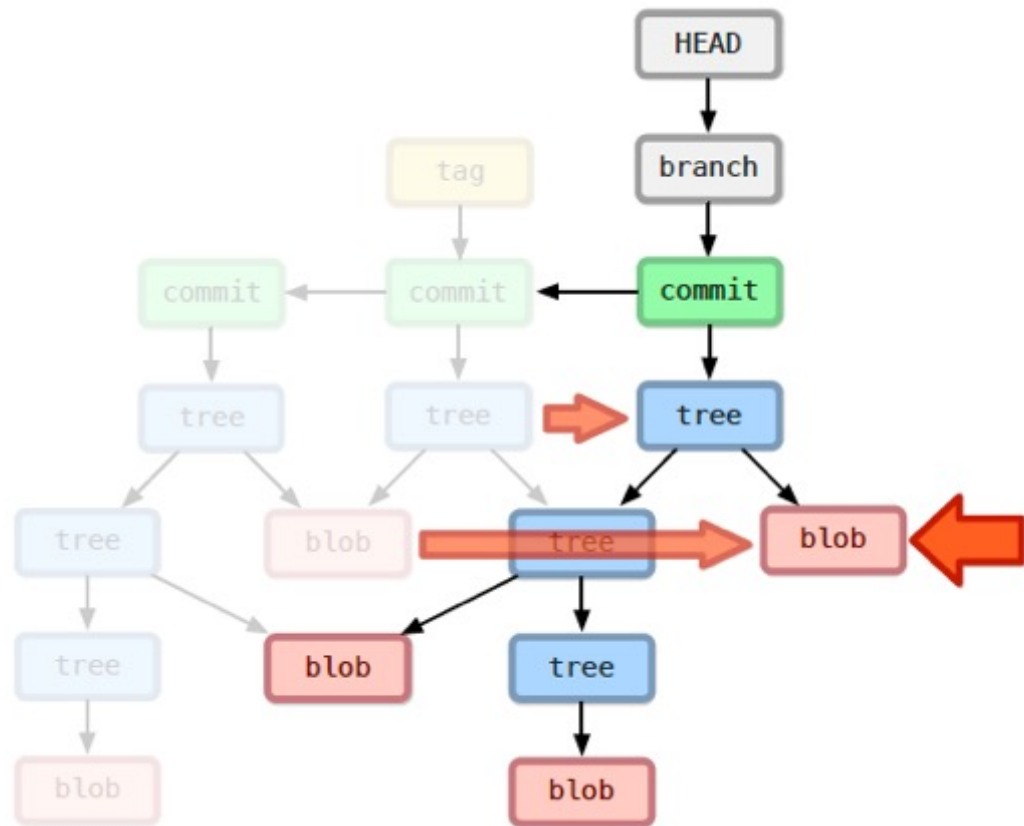
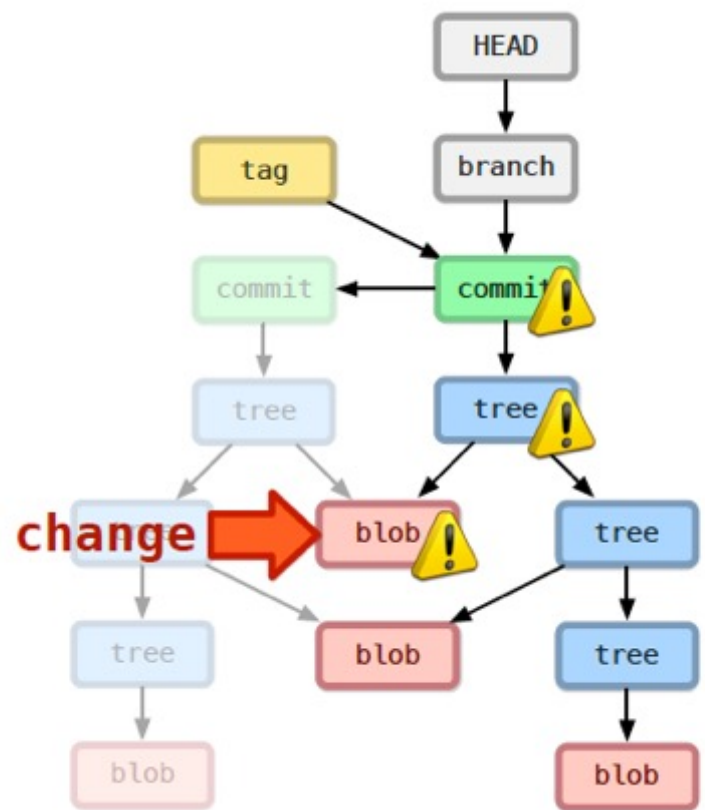
Referenced



New Objects



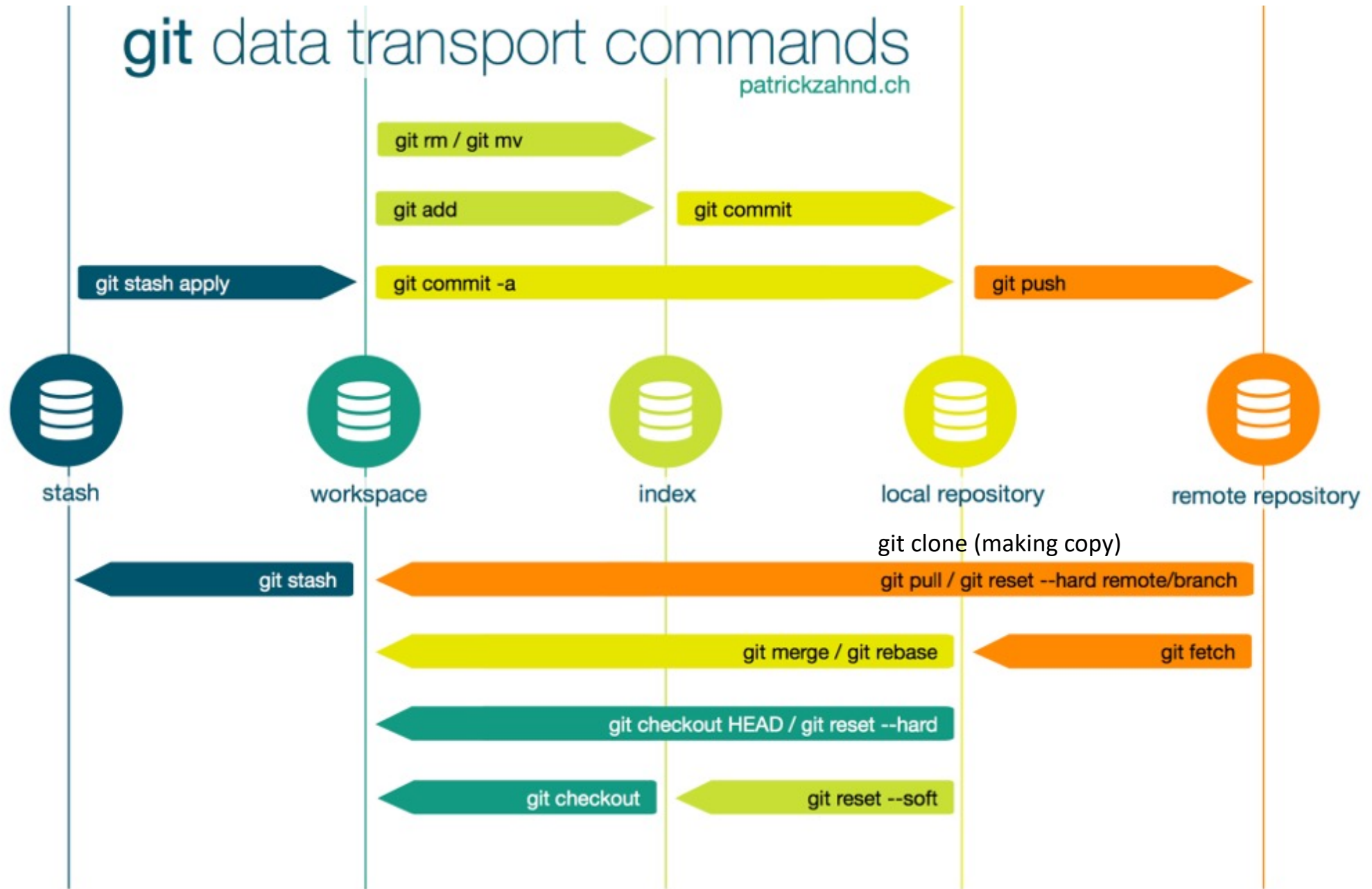
HEAD points out the last commit in the current checkout branch



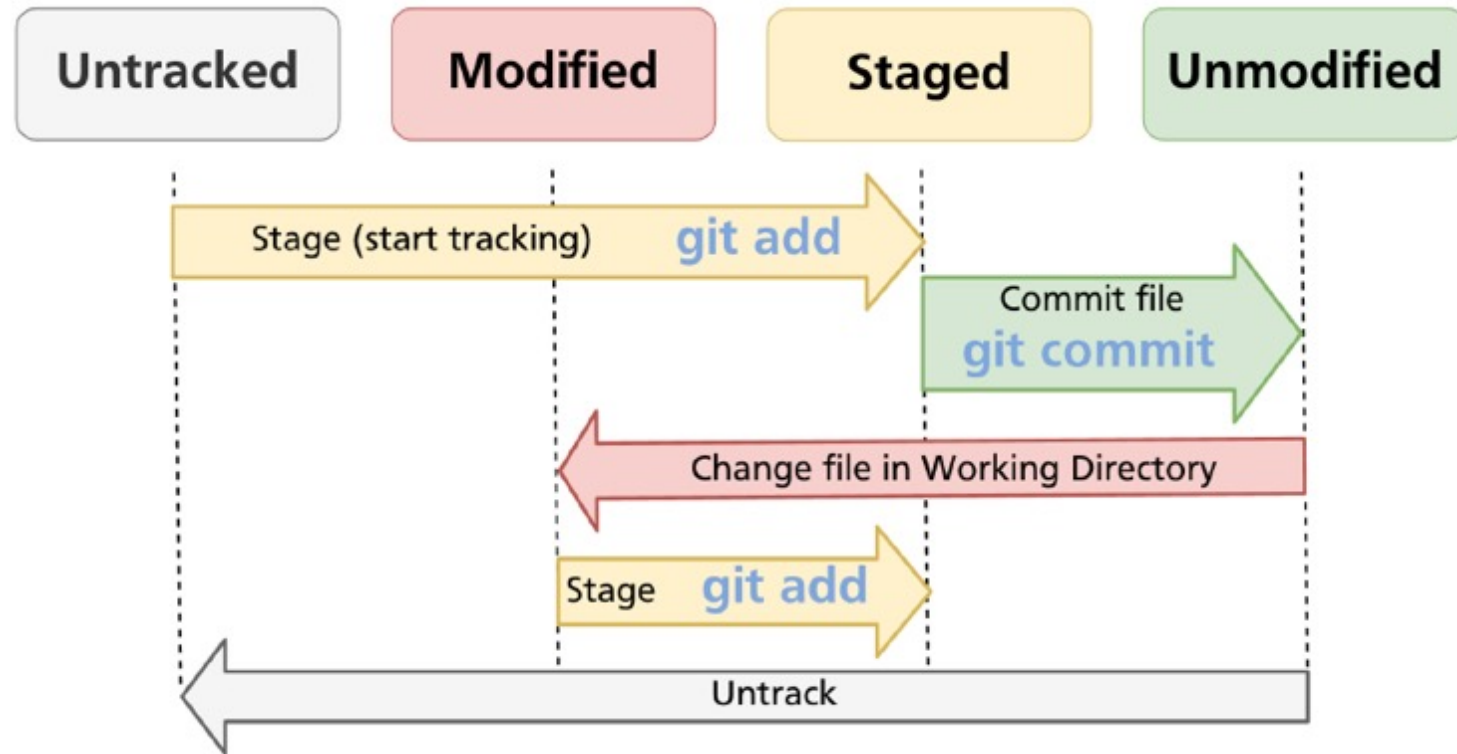


git data transport commands

patrickzahnd.ch

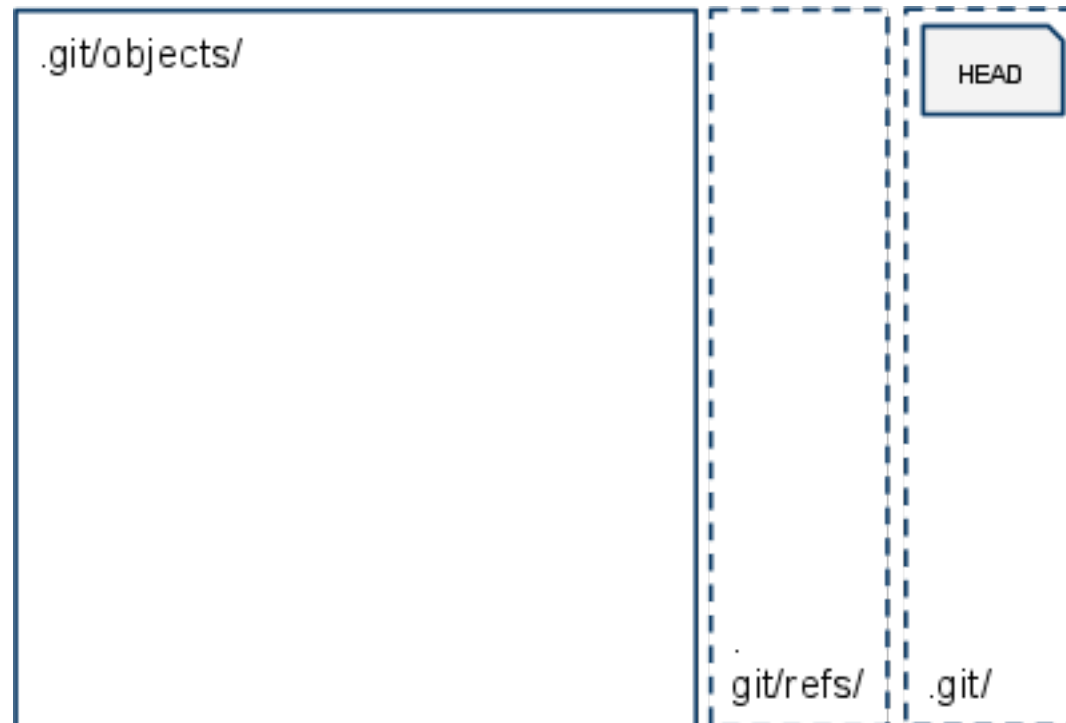


Git Files Lifecycle



git init

- Empty `.git/objects/` and `.git/refs/` created
- No index file yet
- HEAD symbolic reference created. `$ cat .git/HEAD` ref: refs/heads/master

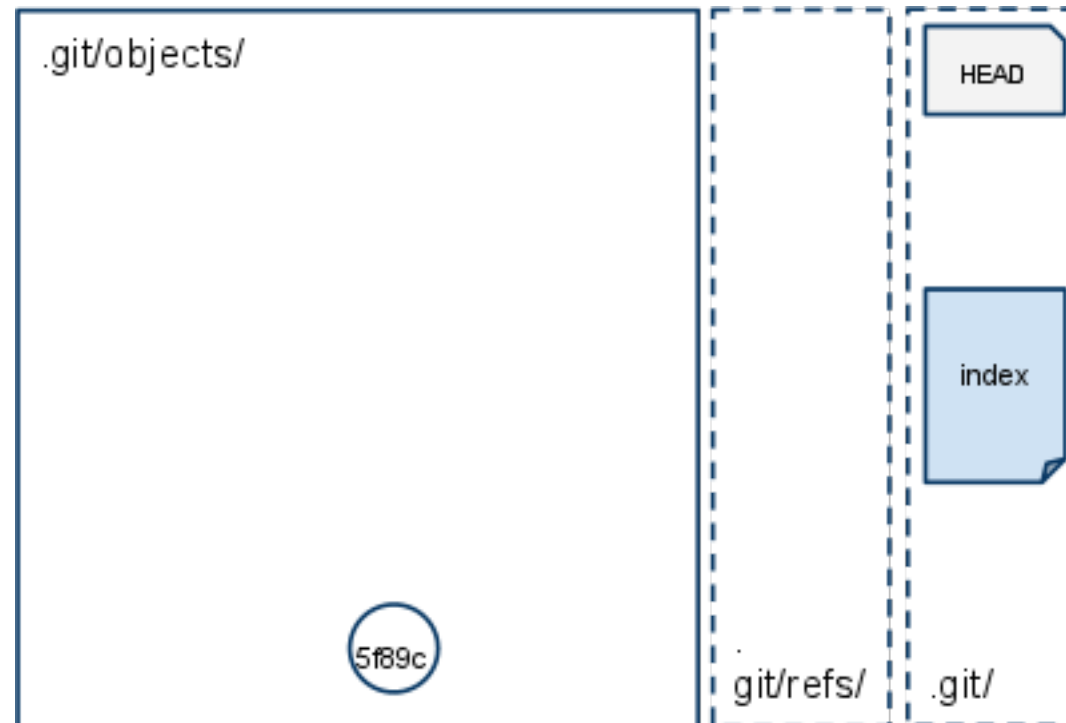


git add

- Index file created: SHA1 hash that points to a blob object
- Blob object created: content of README file is stored

```
$ git ls-files --stage
100644 5f89c6f016cad2d419e865df380595e39b1256db 0 README
```

```
# .git/objects/5f/89c6f016cad2d419e865df380595e39b1256db
$ git cat-file blob 5f89c6
A roti canai project.
```



git add

Documentation

add files to the index

```
add: files
    entries = read_index()
    for each files
        create new index entry
        add it to entries list
    sort entries
    write_index(entries)
```

git status

Documentation

files with differences between the working dir and the index

```
status:
    index = read_index()
    files = work_dir_files()
    for each files
        if file.path in index
            hash = hash-object(file, "blob")
            if hash != entry.hash
                "modified"
        else
            "new"
    for each index entry
        if entry.path != all files path
            "deleted"
```

git diff

Documentation

changes between the working dir and the index

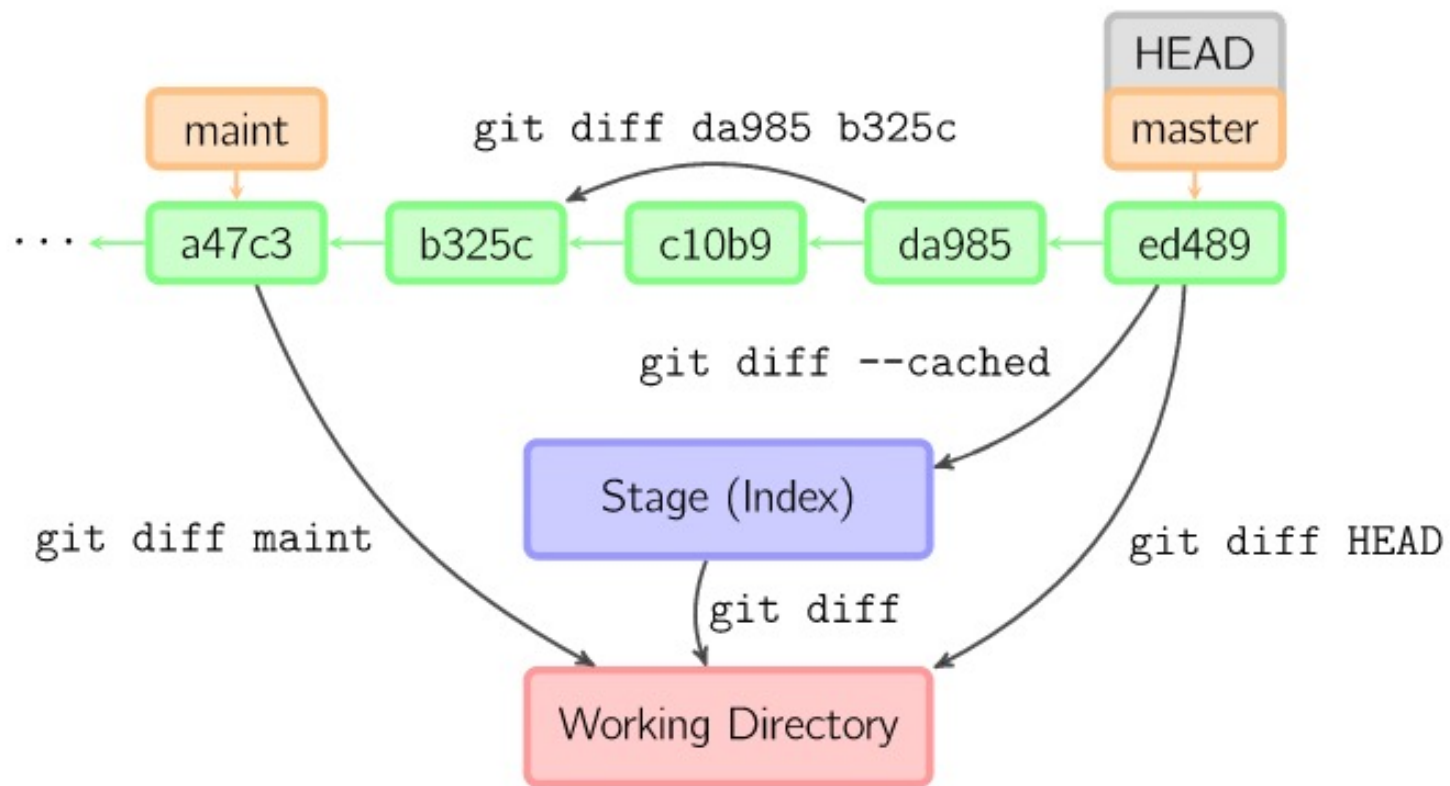
```
diff: paths
    index = read_index()
    for each paths
        stored_file = get_index_entry(path)
        stored_obj = get_object(stored_file.hash)
        current_data = read_file(path)

        print LCS_diff(stored_obj.data, current_data)
```

```
git diff --diff-algorithm=
```

- myers (default): the basic greedy diff algorithm
- minimal: get the smallest possible diff
- patience: try to get more meaningful diff
- histogram: mainly used for its speed

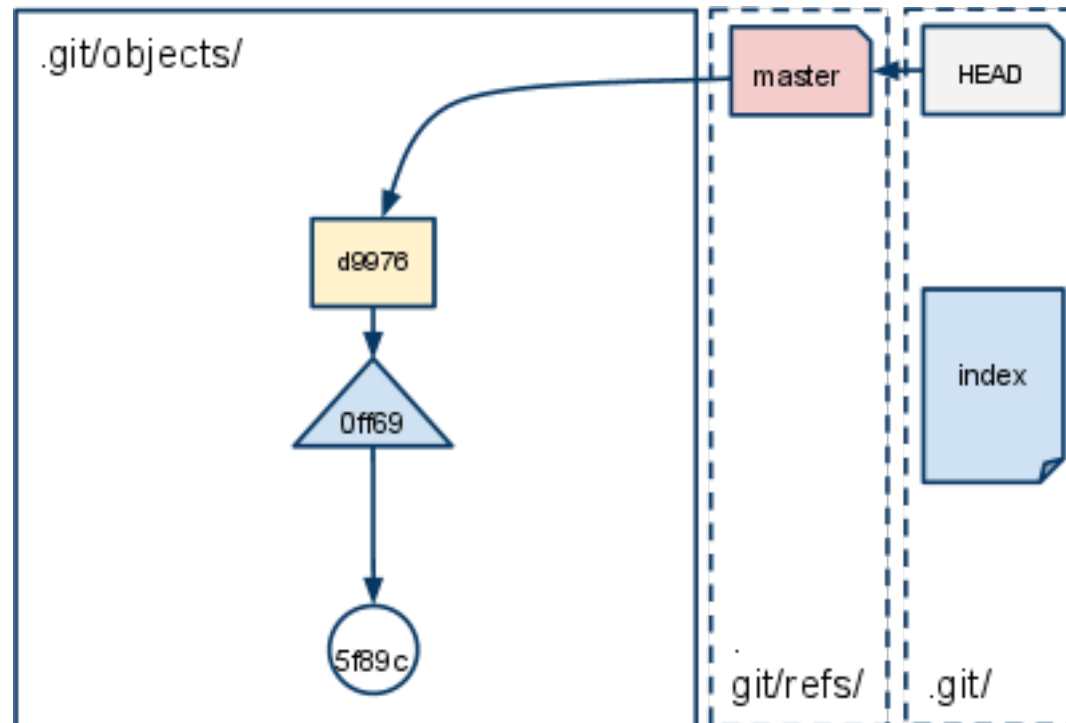
git diff



git commit

- Branch 'master' reference created: points to the latest commit object in 'master' branch
- First commit object created: points to the root tree object
- Tree object created: represents directory

```
# .git/objects/0f/f699bbafc5d17d0637bf058c924ab405b5dcfe  
$ git ls-tree 0ff699  
100644 blob 5f89c6f016cad2d419e865df380595e39b1256db  README
```



git commit

Documentation

stores the index content as a new commit object

```
commit: message
    tree_hash = write-tree()
+ parent_hash = get current commit (HEAD)
    content =
        "tree tree_hash
+     parent parent_hash
        author author_name time
        committer committer_name time
        message"
    hash = hash-object(content, "commit", write=True)
+ update HEAD
    return hash
```

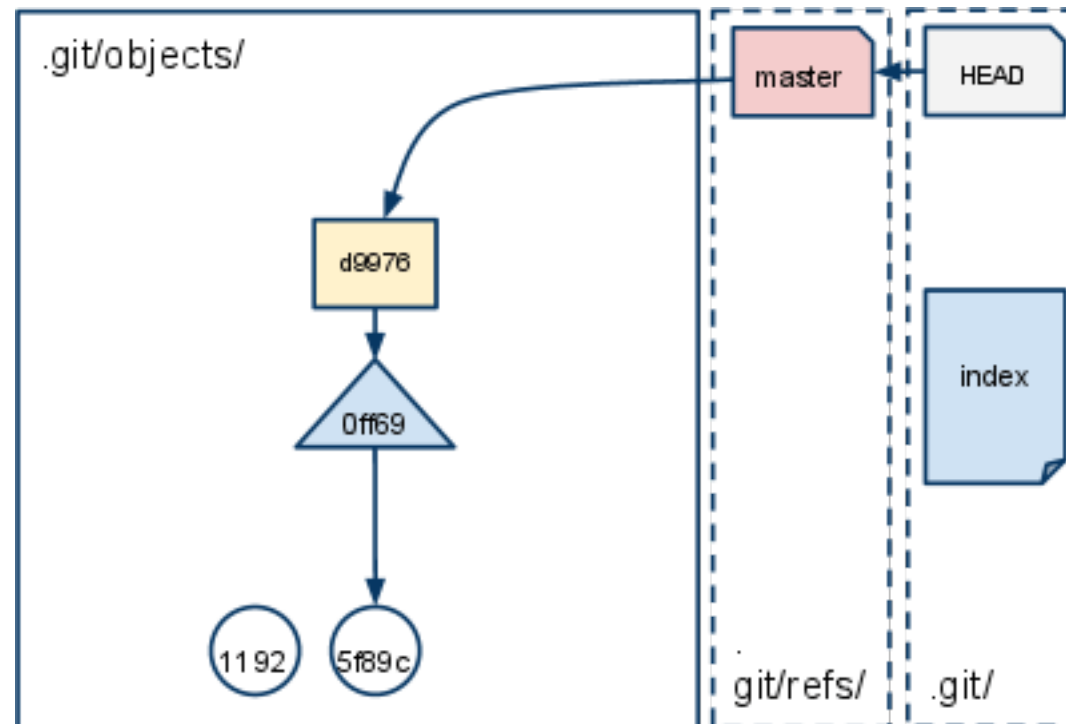
```
write-tree:
    entries = []
    index = read_index()
    for each index entries
        parse entry info
        append new tree entry to entries list
    hash = hash-object(entries, "tree", write=True)
    return hash
```

Add Modified File

- Index file updated: points to a new blob
- Blob object created: entire content is stored as a new blob

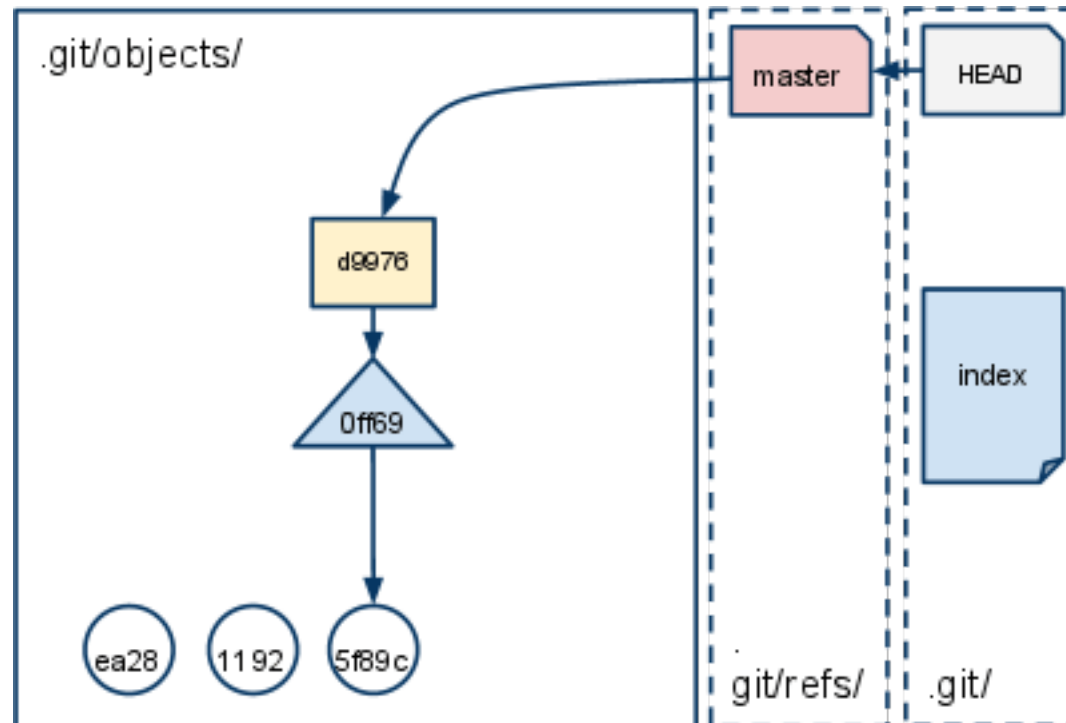
```
$ git ls-files --stage
100644 1192db4c15e019da7fc053225d09dea14bc3ac07 0 README
```

```
# .git/objects/11/92db4c15e019da7fc053225d09dea14bc3ac07
$ git cat-file blob 1192db
A roti canai project.
Welcome everyone.
```



Add File into Subdirectory

- Index file updated
- Blob object created

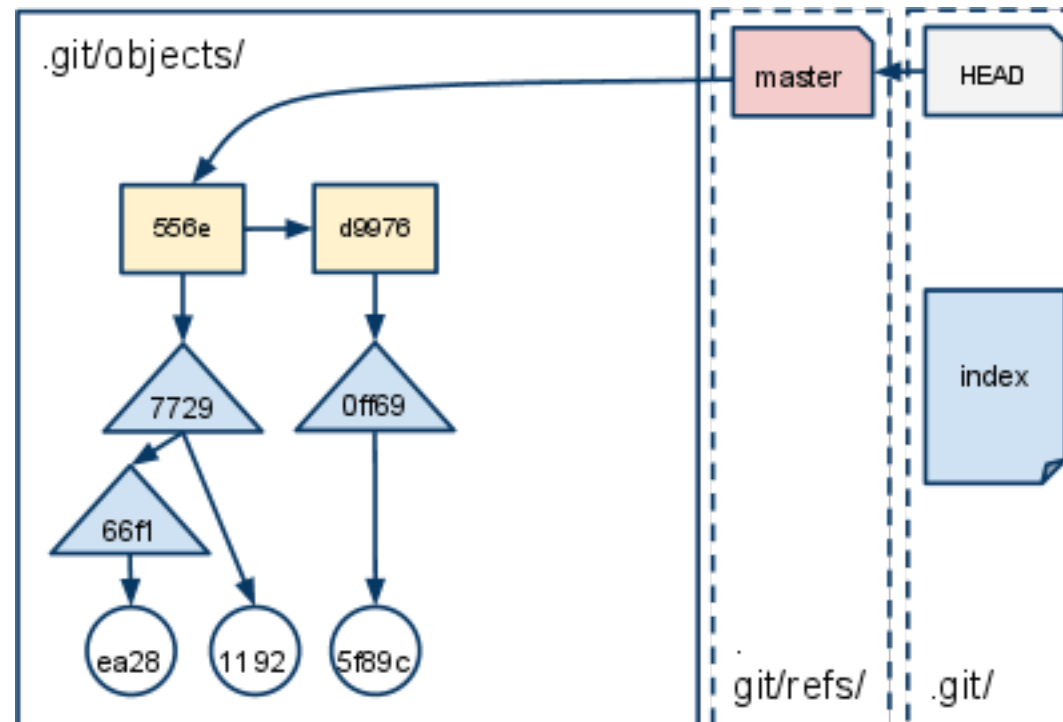


Second Commit

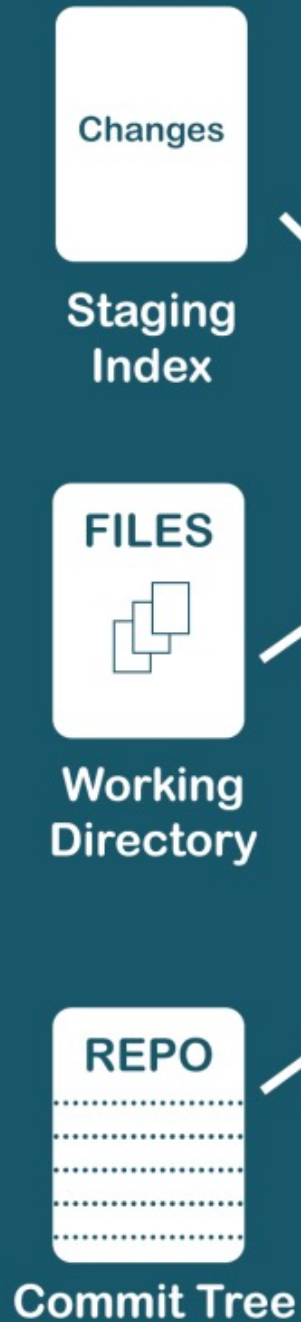
- Branch 'master' reference updated: points to the latest commit
- Commit object created: 'parent' points to the first commit object
- New root tree object and subdir tree object created

```
$ git cat-file commit 556e
tree 7729a8b15b747bce541a9752a8f10d57daf221b6
parent d9976cfe0430557885d162927dd70186d0f521e8
author Huiming Teo <huiming@favoritemedium.com> 1306743598 +0800
committer Huiming Teo <huiming@favoritemedium.com> 1306743598 +0800

second commit
```



git log



git status

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   first_file.txt

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

    second_file.txt
```

git log --oneline

```
$ git log --oneline
a5edf36 (HEAD -> master, origin/master, origin/HEAD) f
bb9479b e
c71d1af Revert "Revert "adds b and c""
8cbdcc3 Revert "adds b and c"
7891e93 adds d
478e7ba adds b and c
0ca0f89 456
6b254ed testing and 123
8cc1b80 Initial commit
(base)
```



Git log format string cheatsheet

Pretty format

```
git log --pretty="format:%H"
```

See the next tables on format variables.

Hash

Commit	
%H	commit hash
%h	(abbrev) commit hash
Tree	
%T	tree hash
%t	(abbrev) tree hash
Parent	
%P	parent hash
%p	(abbrev) parent hash

Commit

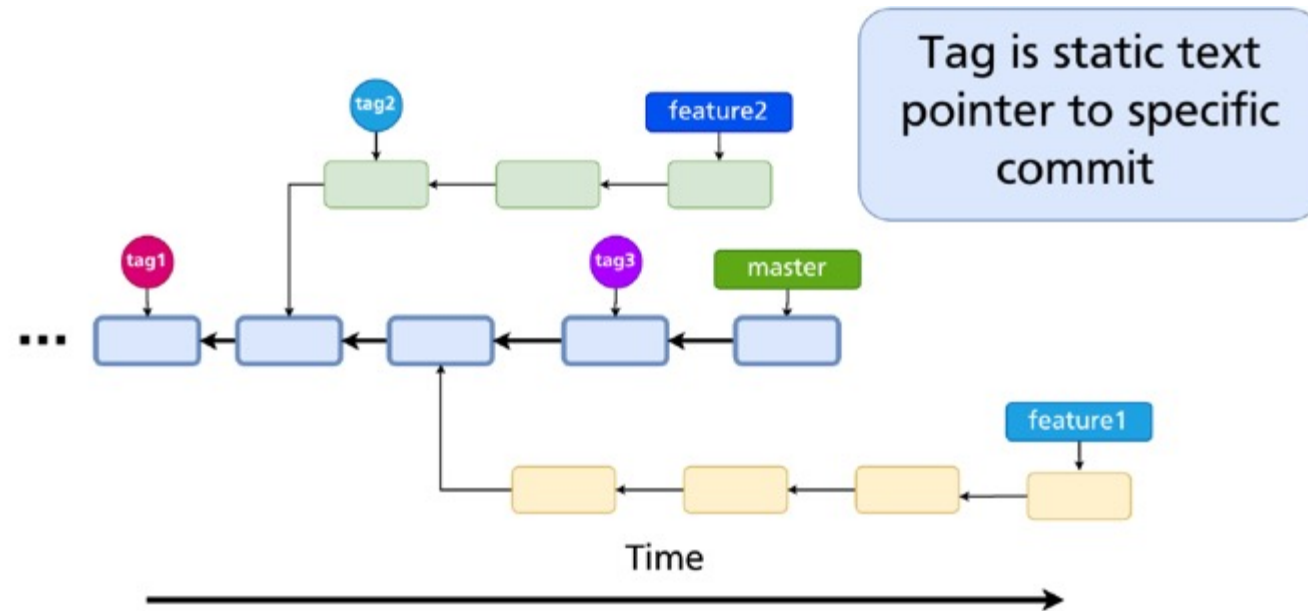
%s	commit subject
%f	commit subject, filename style
%b	commit body
%d	ref names
%e	encoding

<https://devhints.io/git-log-format>

Git Alias Examples

- git l
 - l = log --color --graph --pretty=format:'%Cred%h%Creset - %C(yellow)%d%Creset %s %Cgreen(%cr)%C(bold blue)<%an>%Creset' --abbrev-commit
- git cleanbranch
 - cleanbranch = "!git branch -d \$(git branch --merged | grep -v '^*\\|\\<master\$')"
- git ch
 - ch = "!git checkout \$(git branch | fzf)" (after install fzf)
- git backup
 - b0 = "!git branch | awk '/^*/{print \$2}'" backup = "!git branch backup-`git b0`"

Git Tags



Lightweight

`git tag v1.0.0`

- is stored in the `.git/refs/tags`

Annotated

`git tag -a v1.0.0 -m "New tag"`

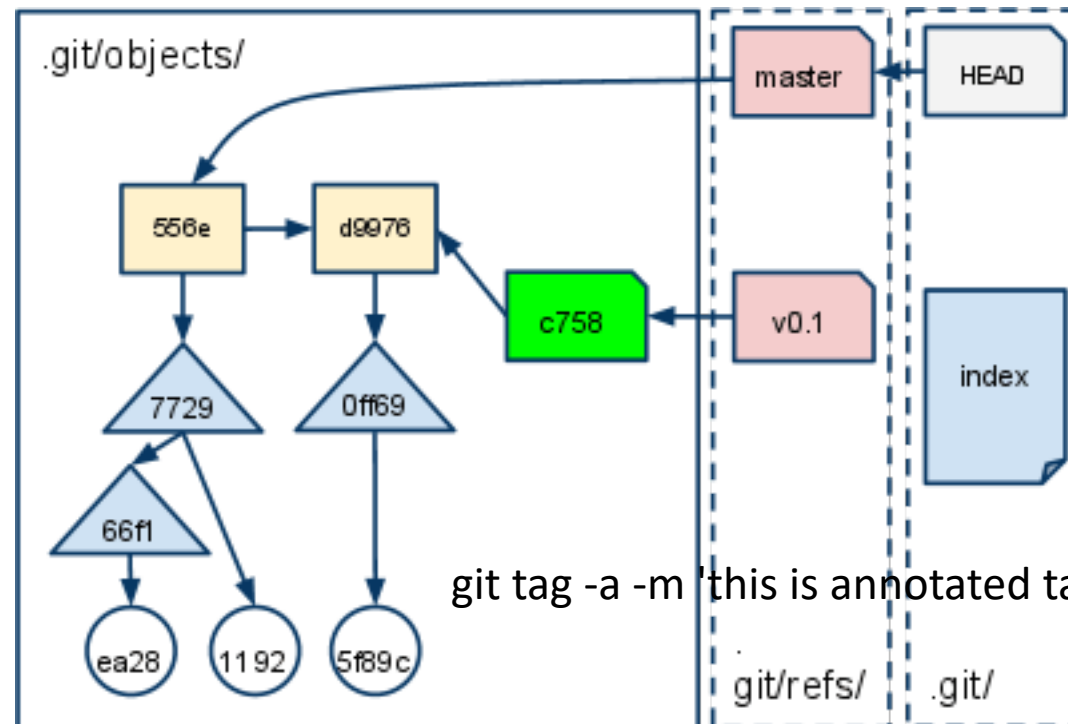
- is stored in the `.git/refs/tags`
- is also stored in the `.git/objects`
- stores tag message
- stores tag author and date

Annotated Tag

- Tag reference created: points to a tag object
- Tag object created

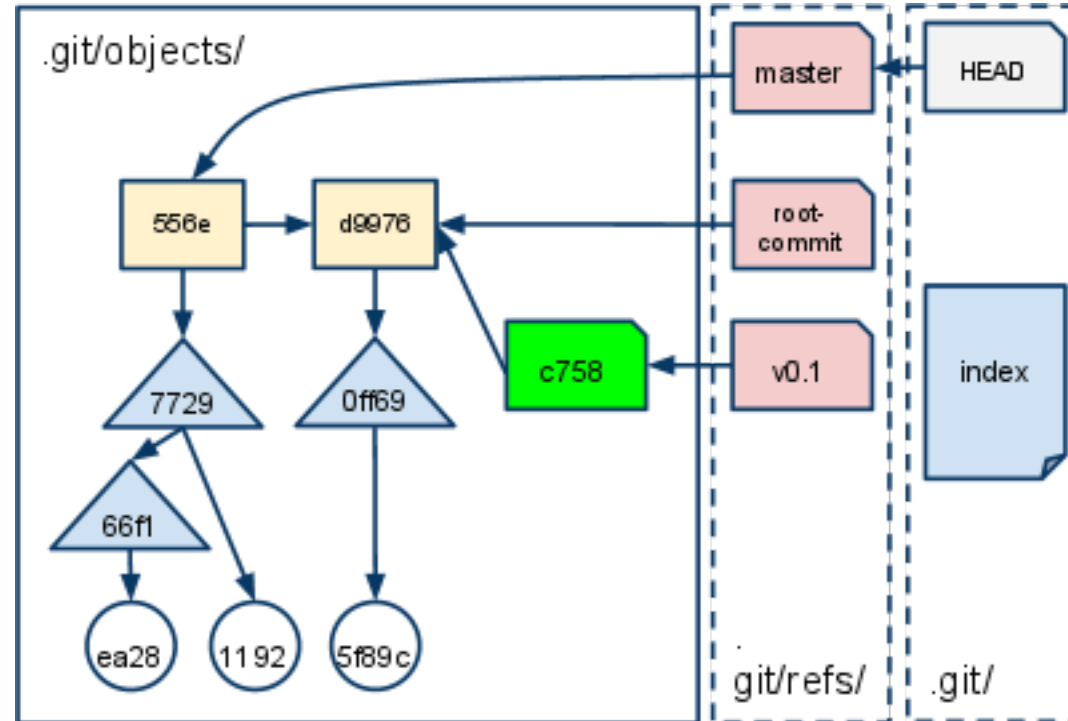
```
$ cat .git/refs/tags/v0.1  
c758f4820f02acf20bb3f6d7f6098f25ee6ed730
```

```
$ git cat-file tag c758  
object d9976cfe0430557885d162927dd70186d0f521e8  
type commit  
tag v0.1  
tagger Huiming Teo <huiming@favoritemedium.com> 1306744918 +0800  
  
this is annotated tag
```



Lightweight Tag

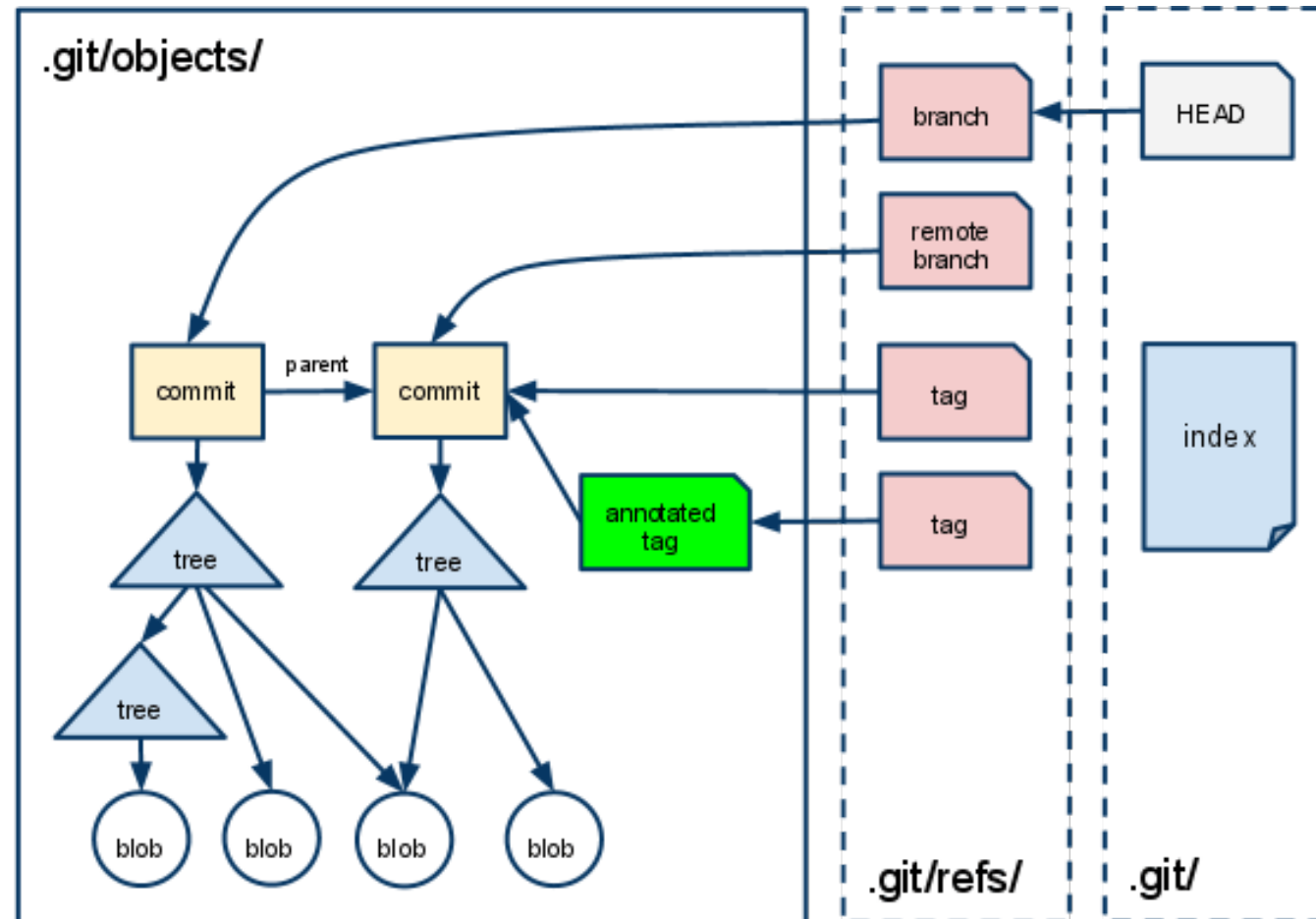
- `git tag root-commit d9976`



*Push tags to remote: `git push --tag`

*`git push --follow-tags`

Git Architecture



.gitignore

Specify files which will be ignored by Git

```
$ cat .gitignore
.svn
.DS_Store
build
*.pbxuser
*.perspective
*.perspectivev3
*.modelv3
*.mode2v3
*.xcuserstate
*.xcworkspace
xcuserdata
```

- build folders like **bin/**
- dependency folders like **node_modules/**
- compiled and log files like ***.pyc, *.log**
- hidden OS files like **Thumbs.db** or **.DS_Store**

gitignore.io

 | gitignore.io



gitignore.io

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요

운영체제, 개발 환경(IDE), 프로그래밍 언어 검색

생성

[소스 코드](#) | [커맨드라인 문서](#)

Branches

Branch is just text reference to the commit

1. Default branch is **master**
2. Multiple branches can exist in the same repository
3. Pointers for all branches are located in **.git/refs/heads** folder
4. Current branch tracks new commits
5. Branch pointer moves automatically after every new commit
6. Change branch **git checkout <branch>**

master

new-branch

DEV-1512

git branch

Documentation

create a new branch

```
branch: name  
    check if repo has at least 1 commit  
    get current commit hash (HEAD)  
    write the hash to .git/refs/heads/name
```

git branch -f master HEAD~3 (move branch)

git checkout

Documentation

switch to a branch Switch branches or restore working tree files

```
checkout: ref
```

```
    check if ref is a commit object
```

```
    compare trees between ref commit and HEAD commit
```

```
    for each diff
```

```
        add/modify/delete the file
```

```
    update index
```

```
    update HEAD
```

git checkout -- file



Branch Logs

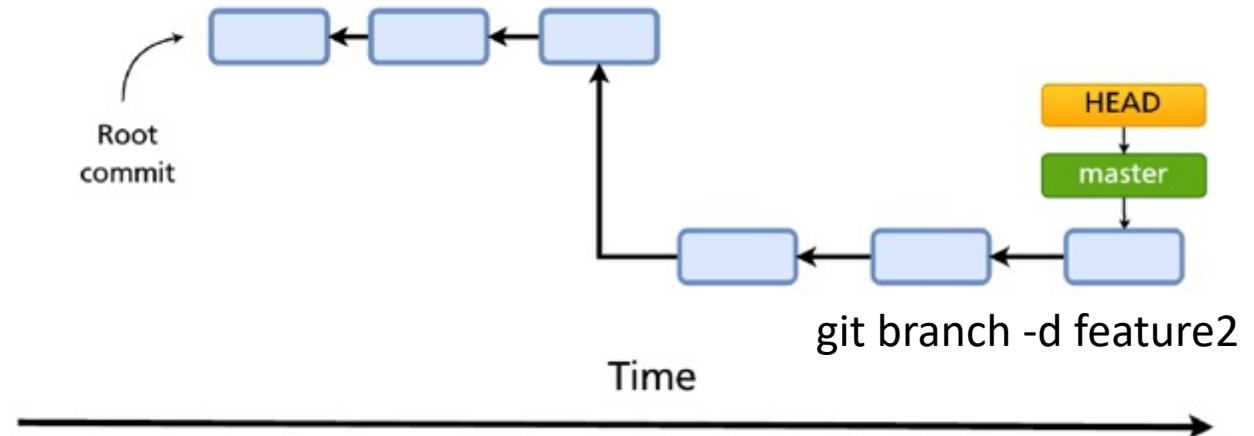
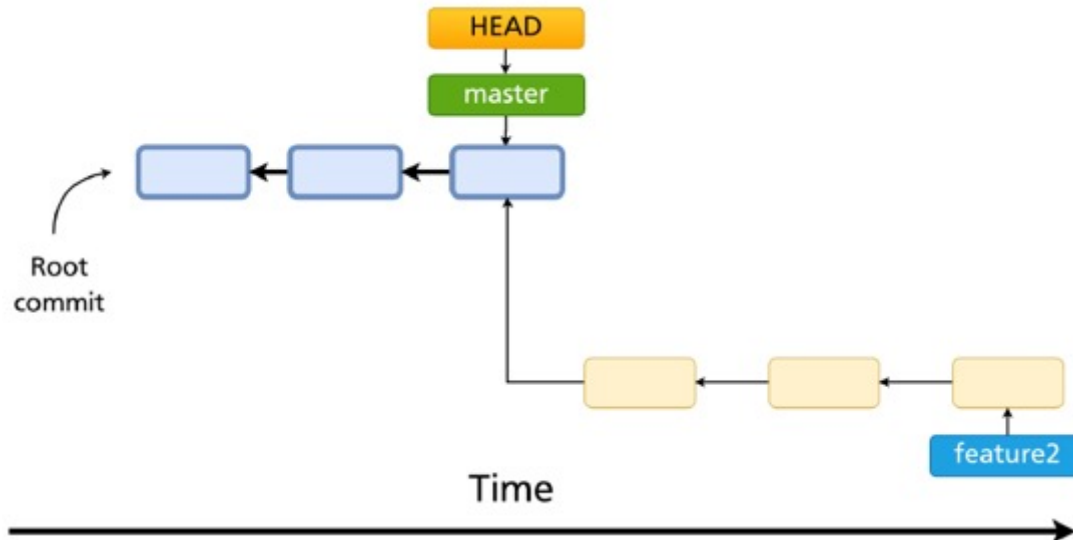
```
$ git reflog HEAD
f165ace HEAD@{0}: commit: hooks
13db5c9 HEAD@{1}: commit: repo layout
d74db73 HEAD@{2}: commit: fixed headers
16180f1 HEAD@{3}: commit: use FETCH_HEAD directly
:      :      :
adb1fb6 HEAD@{16}: merge 810053ec98502f04943c06f6ca3bfaf3d3f2060d: Merge made by the 'recursive'
strategy.
:      :      :
77d7f39 HEAD@{81}: checkout: moving from master to 77d7f39
```


git merge

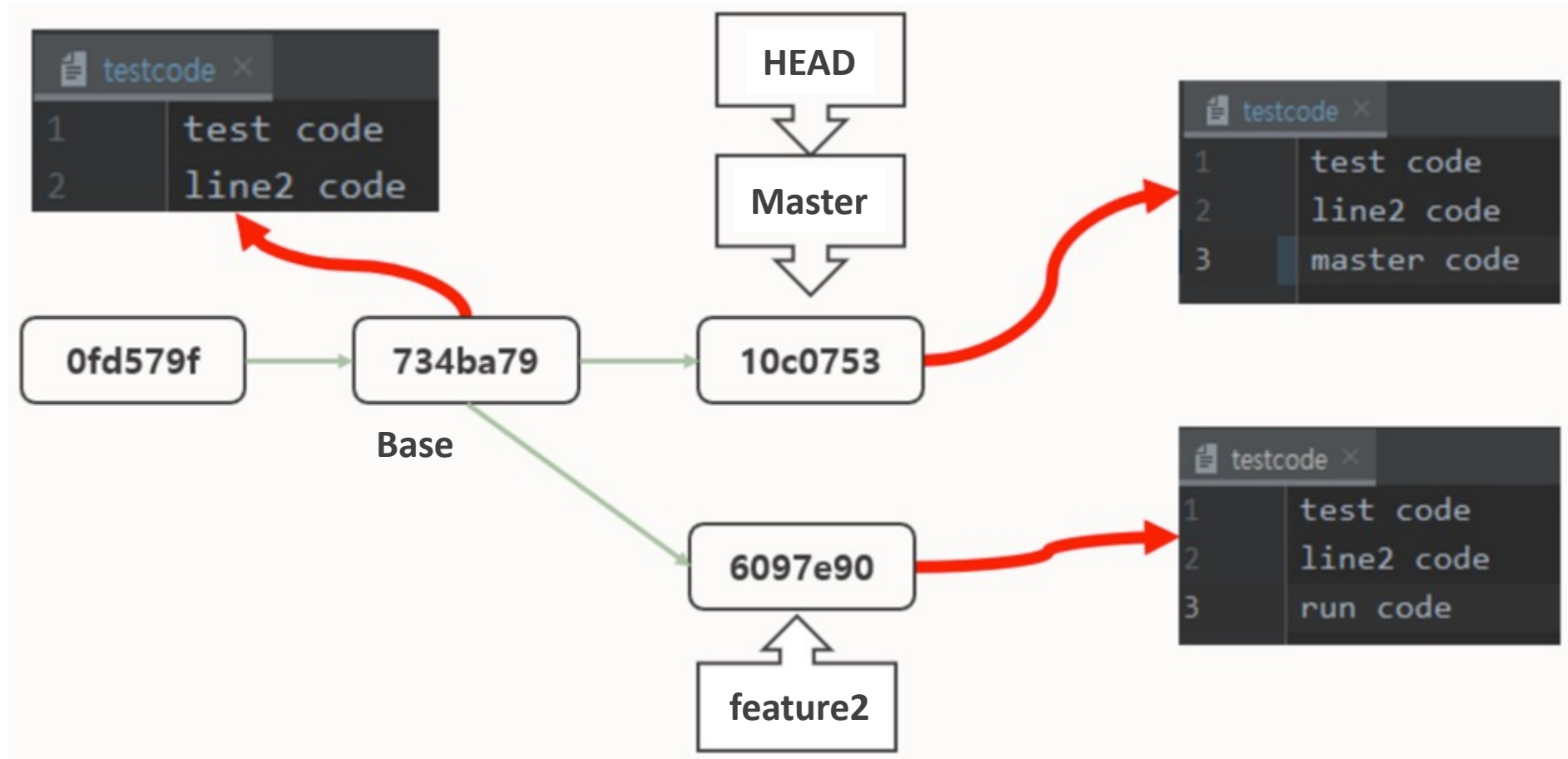
```
merge: ref
    check if ref is a commit object
    // Case 1: fast-forward
    if HEAD is ancestor of ref
        update working dir from ref
        update HEAD
    // Case 2: non fast-forward
    else
        get diffs from common ancestor
        update working dir
        update HEAD
        if conflicts
            "Need to resolve conflicts"
        else
            commit("Merge ... into ...")
```

Fast-Forward Merge

git merge feature2



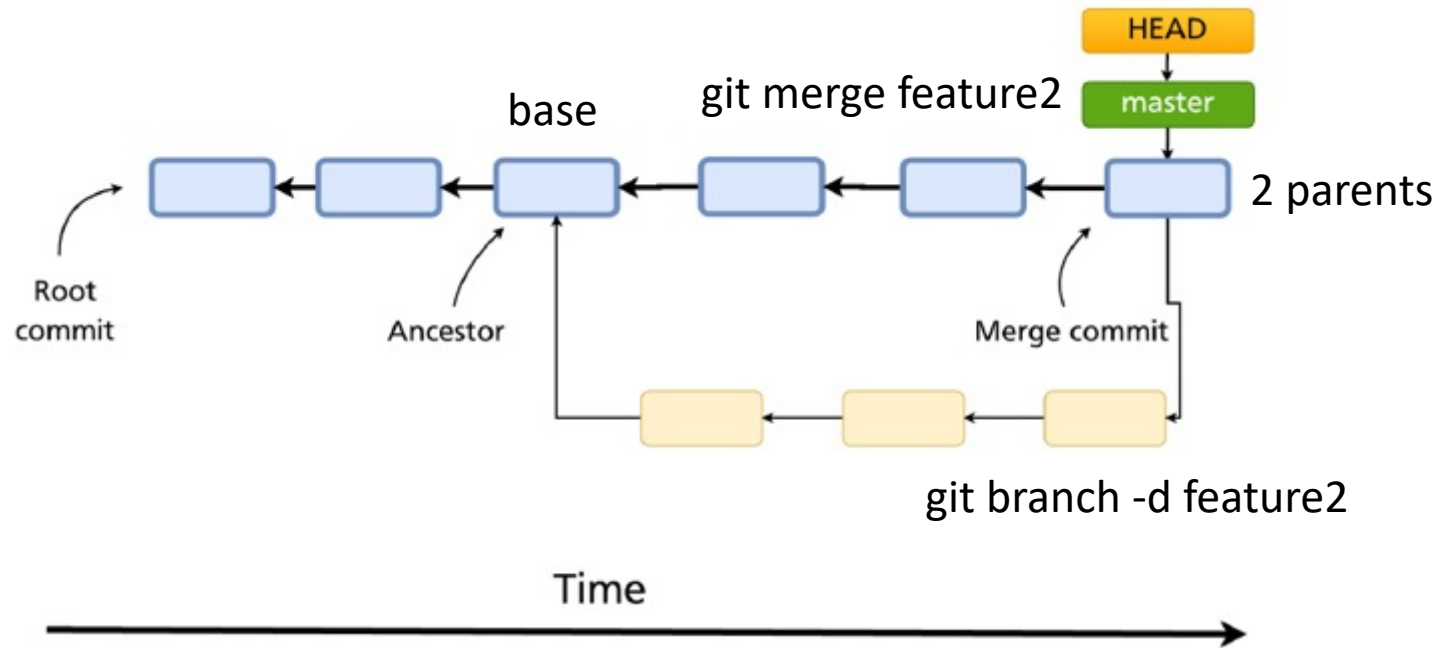
3-Way Merge



2-Way Merge vs 3-Way Merge

New		Feature2	2 way merge	3 way merge
A			?	
B		B	B	B
1		2	?	?
		D	?	

3-Way Merge

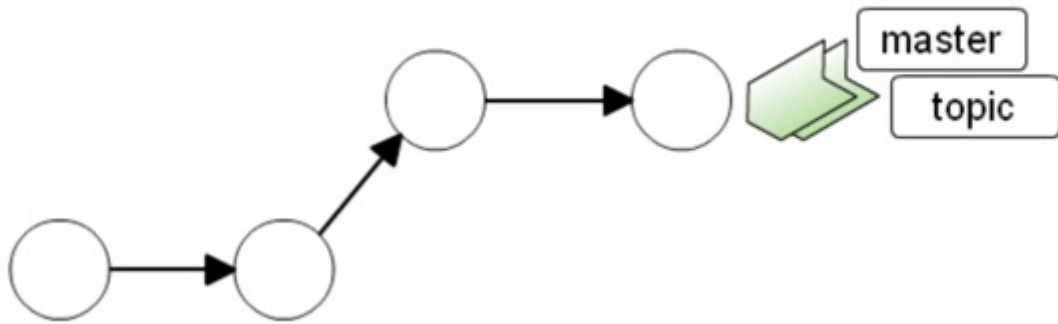


Quiz

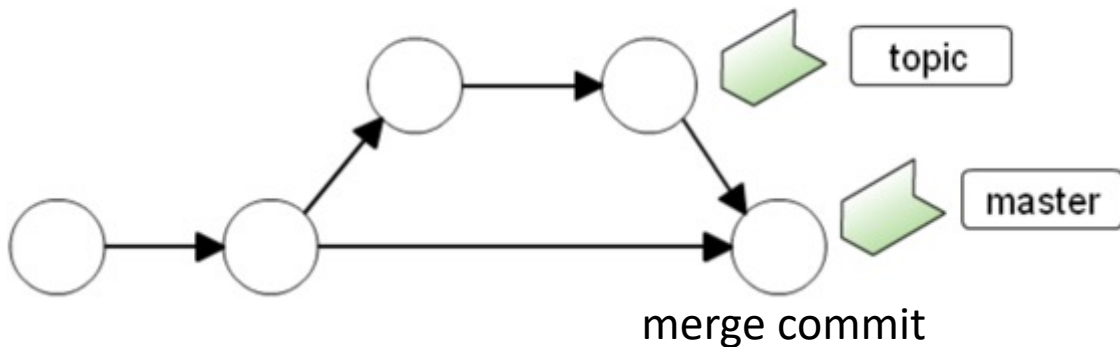
- Which is better?
 - fast-forward merge vs 3-way merge

ff vs no-ff

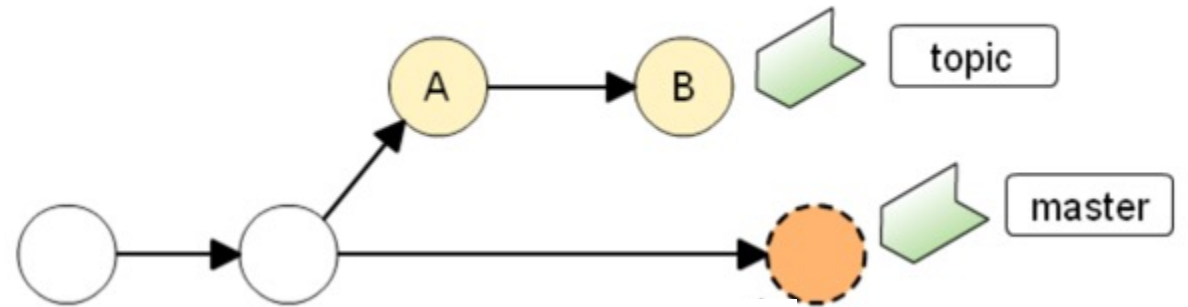
\$ git merge topic



\$ git merge --no-ff topic



\$ git merge --squash topic



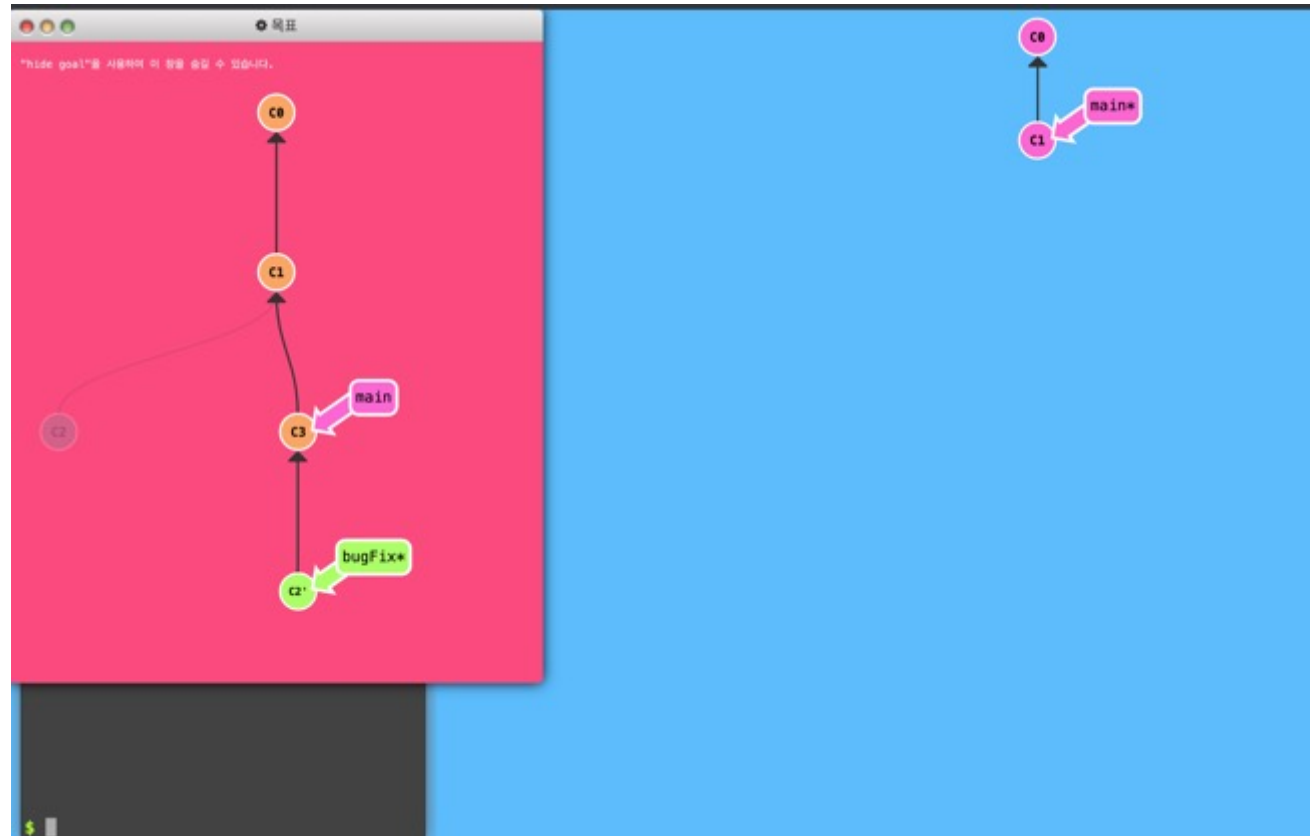
A successful Git branching model

Option 1: \$ git config --global alias.nffmerge "merge --no-ff"

Option 2: \$ git config branch.<name>.mergeoptions "--no-ff"

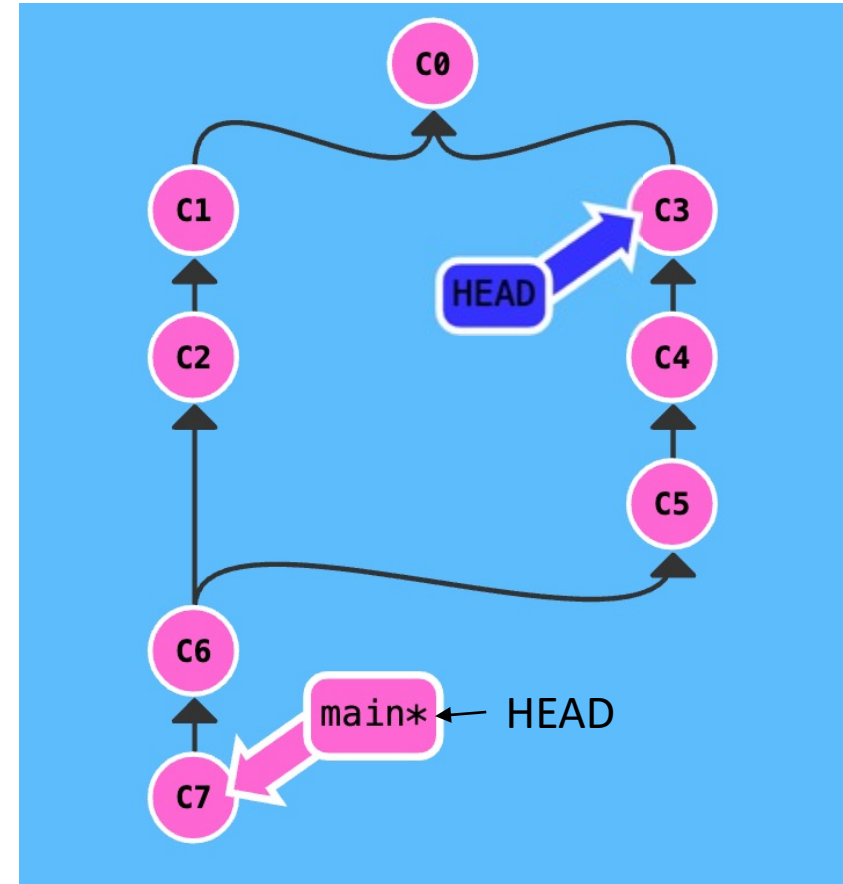
Learning Git Branching

- <https://learngitbranching.js.org/?locale=ko>

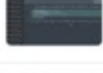


Quiz

- Which commit HEAD points to after `git checkout HEAD~^2~2`

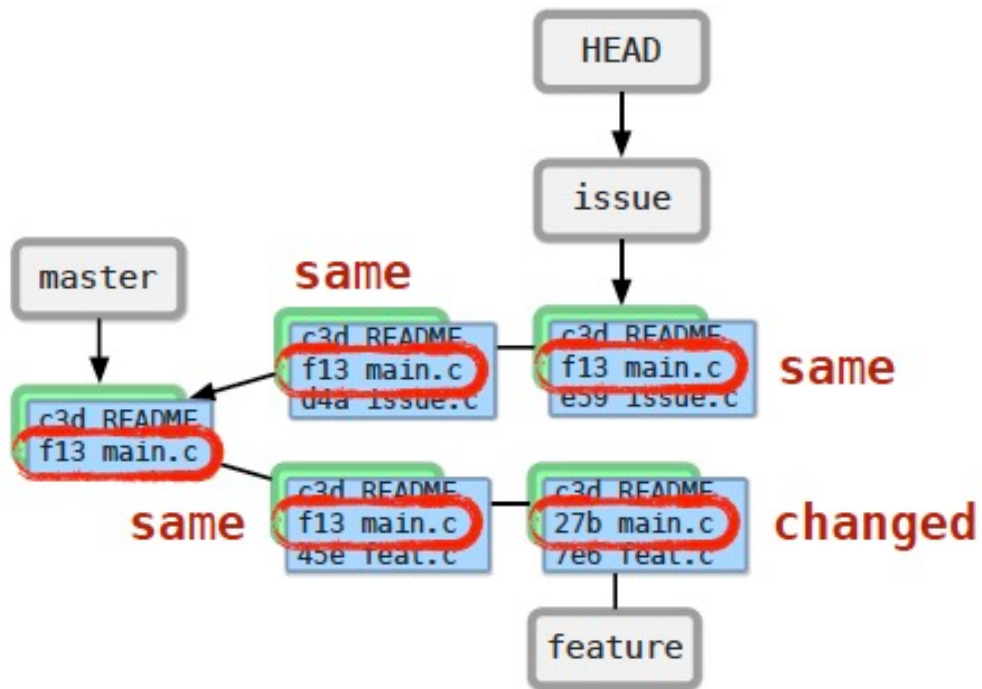


Git GUI

	BEST VISUAL GIT GUIs	PRICE	PLATFORMS	ISSUE TRACKER SUPPORT	PRO FEATURES
92	 SmartGit	FREE/PAID	Linux, Mac, Windows	Atlassian JIRA	Conflict Solver and Com
81	 Git Extensions	FREE	Windows, Linux, MacOS	-	-
80	 GitKraken	\$0-\$99/YR	Windows, macOS, Linux	-	-
75	 Fork	\$50 USD	Windows, macOS	-	-
--	 Visual Studio Code	FREE	Windows, macOS, Linux	-	-
71	 SourceTree	FREE	Windows, macOS	-	-
--	 GitUp	FREE	Mac	-	-
--	 GitAhead	FREE	Windows, Mac, Linux	-	-
69	 Tower	\$69/YR-\$149/YR	Windows, Mac	-	-
69	 TortoiseGit ...	FREE	Windows	-	-
--	 git-cola	-	Windows, macOS, Linux	-	-
--	 Gitg	FREE	-	-	-
--	 Sublime Merge	PAID	Linux, Mac, Windows	-	-
58	 GitHub Desktop	-	-	-	-



Merge Conflict



```
$ git merge feature
```

Auto-merging main.c

CONFLICT (content): Merge conflict in main.c
Automatic merge failed; fix conflicts and then commit the result.

```
git config --global merge.tool vimdiff
```

```
git mergetool [--tool=<tool>] [-y | --[no-]prompt] [<file>...]
```



Meld

FREE

Linux; Windows; Mac (unofficial)

Free and Open Source



P4Merge

FREE

Windows; Mac; Linux

Free



Beyond Compare

PAID

Windows; Mac; Linux

Proprietary



KDiff3

-

-

-



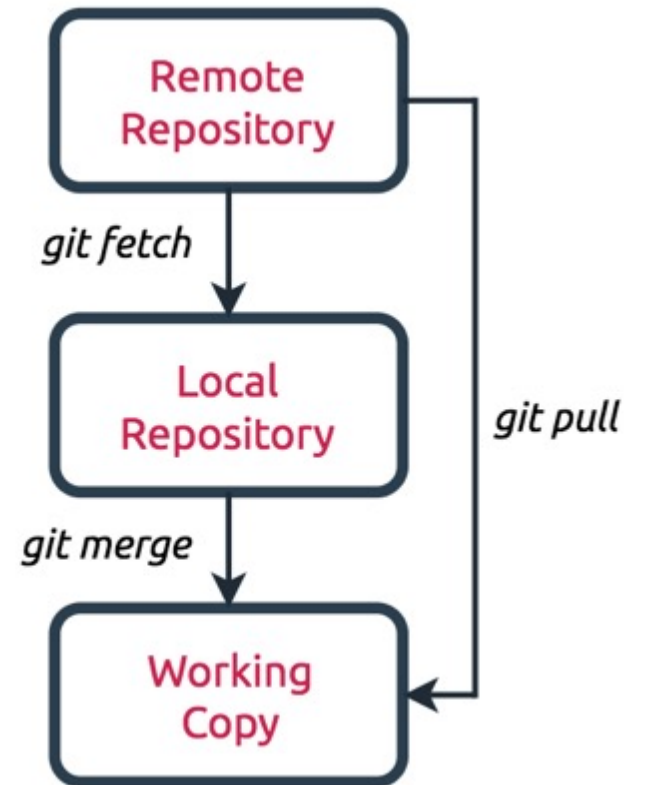
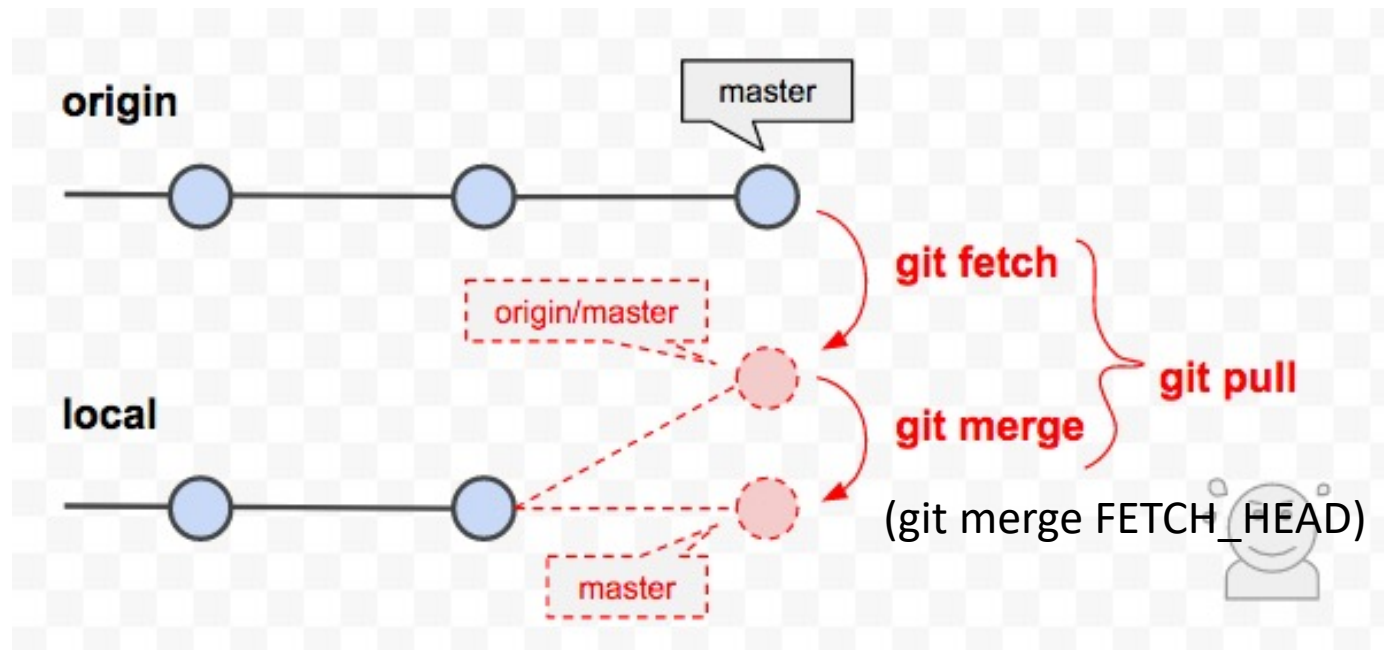
TortoiseMerge

FREE

Windows

Free

git fetch, git pull, git merge



Quiz

- What happens?
 - git push origin :branch_name
 - git fetch origin :branch_name

Remote Push/Fetch Commands

- Update local branches when remote branches updated
 - `git fetch` (and `git checkout branch_name` for repository update)
- Update local branches when remote branches deleted
 - `git remote update origin --prune` (or `git remote prune origin`)
- Add remote branch
 - `git push --set-upstream-to` (or `-u`) `origin branch_name`
- Update local branch from remote branch
 - `git fetch origin remote_branch:local_branch`
- Update remote branch from local branch
 - `git push origin local_branch:remote_branch`
- Delete remote branch
 - `git push origin -d branch_name` (or `:branch_name`)
- Add local branch
 - `git branch branch_name` (or `git checkout -b branch_name` or `git fetch origin :branch_name`)

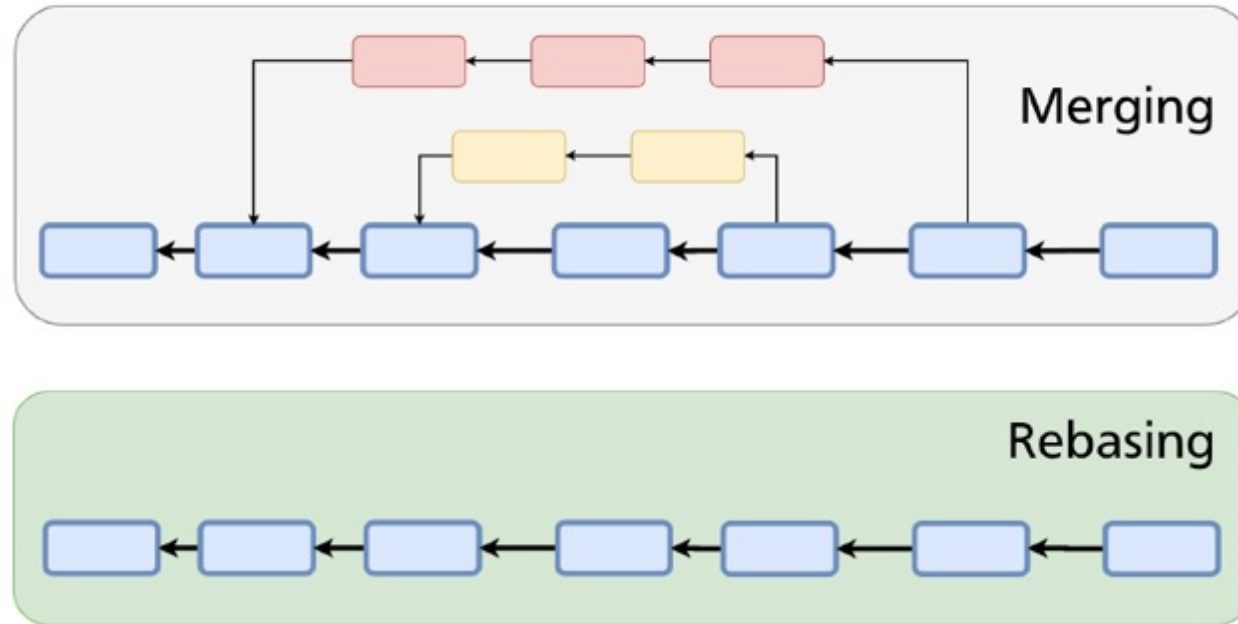


Special Refs

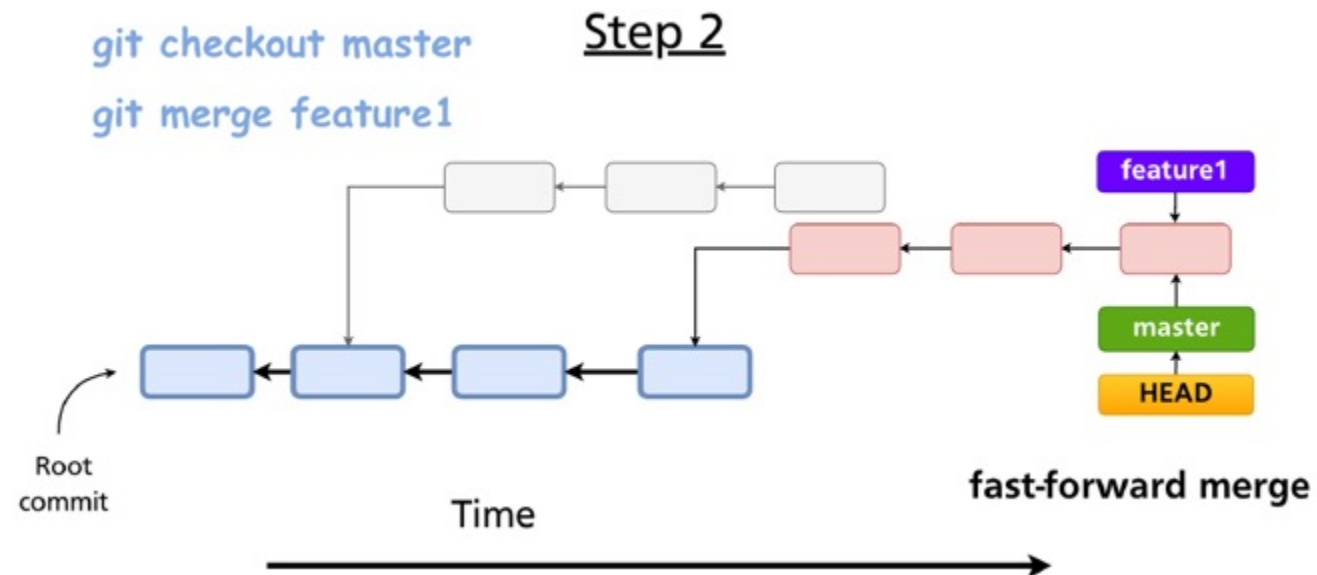
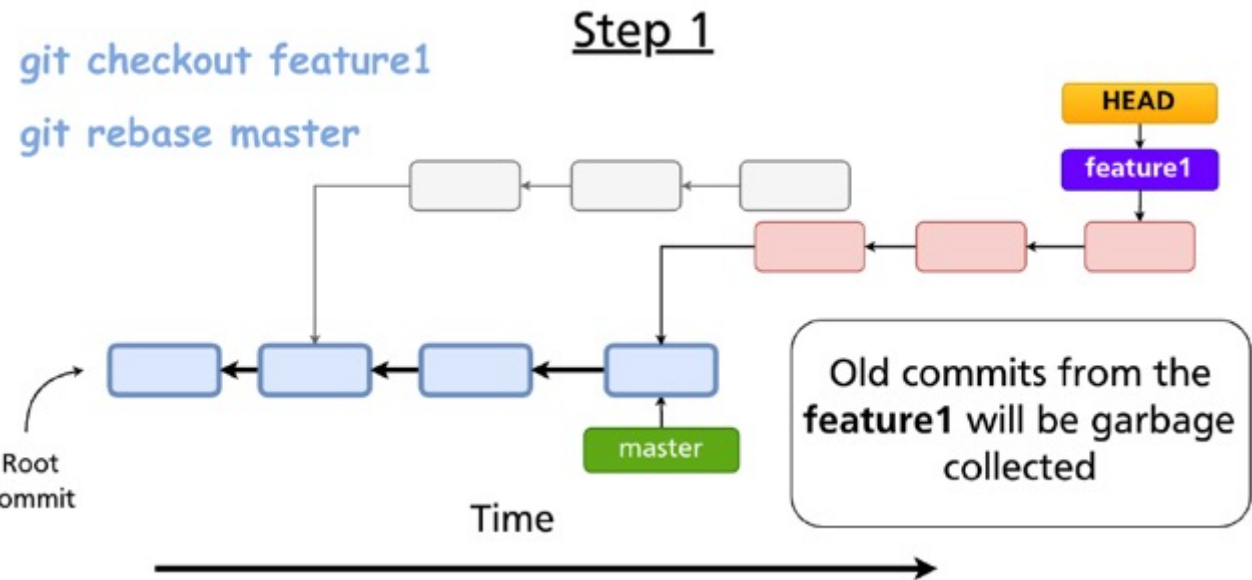
HEAD	base of the working tree	
FETCH_HEAD	upstream head from the last <code>git fetch</code> ³	
ORIG_HEAD	last HEAD before commands doing drastic changes (eg: <code>merge</code> , <code>rebase</code>)	→ Able to rollback after <code>git reset --hard</code> , merge, rebase mistake (<code>git reset --hard ORIG_HEAD</code>)
MERGE_HEAD	commit being merged ³ (<i>when resolving a conflict</i>)	HEAD@{1}, HEAD^, HEAD~
CHERRY_PICK_HEAD	commit being cherry-picked (<i>when resolving a conflict</i>)	

Merge vs Rebase

Rebasing vs merging



- rewrites history
- history becomes linear
- doesn't keep entire history of all commits



Nodejs/node

Taeung/node

fork



Commit 3

Commit 2

Commit 1

Commit 3

Commit 2

Commit 1

Nodejs/node

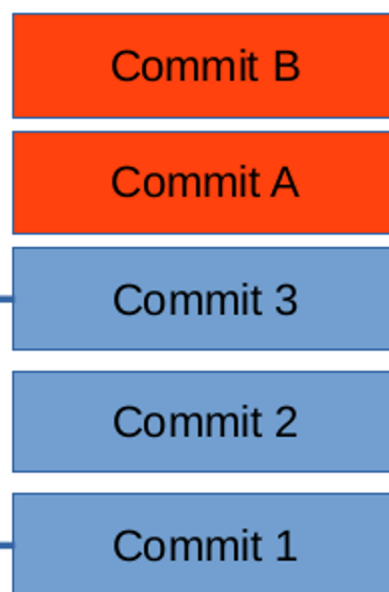
Taeung/node

fork

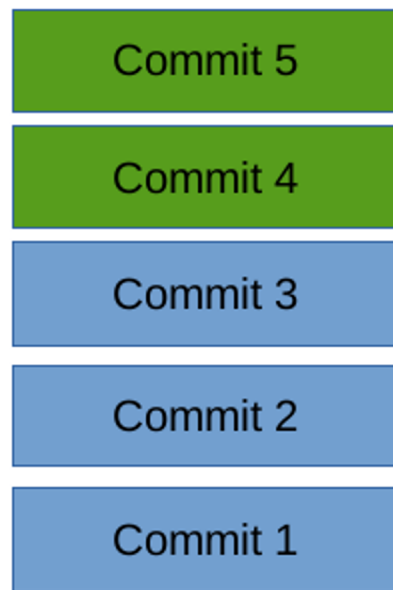


clone

Base



Nodejs/node



fork



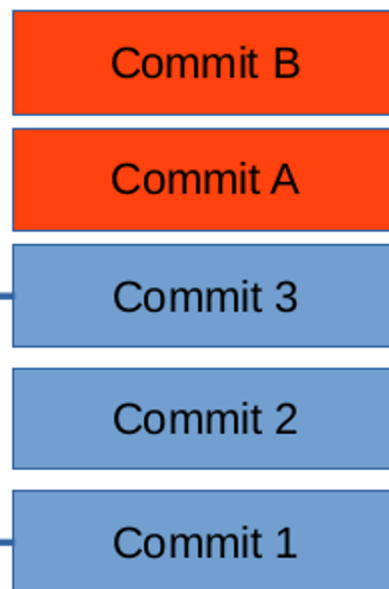
Taeung/node



clone

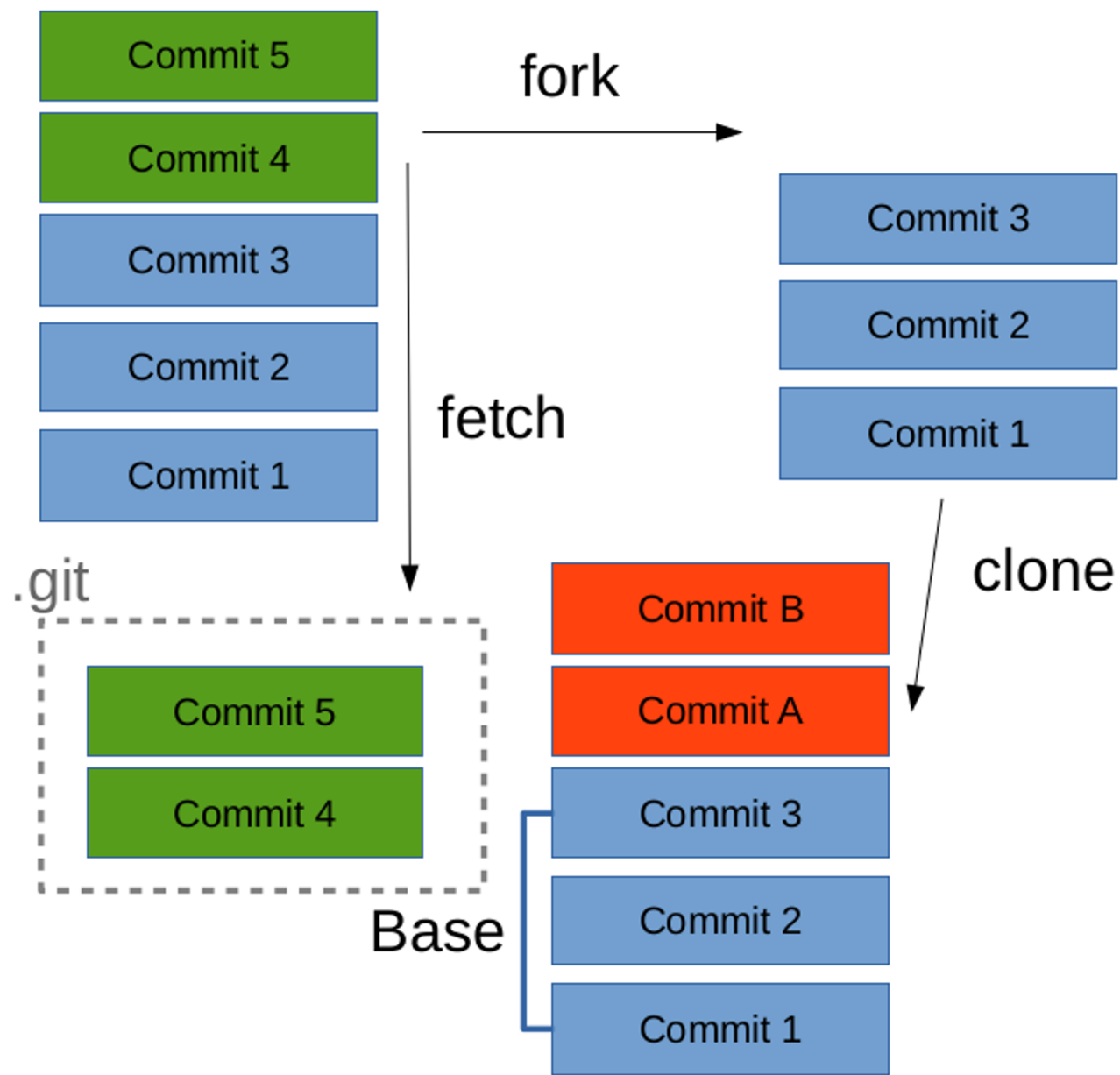


Base



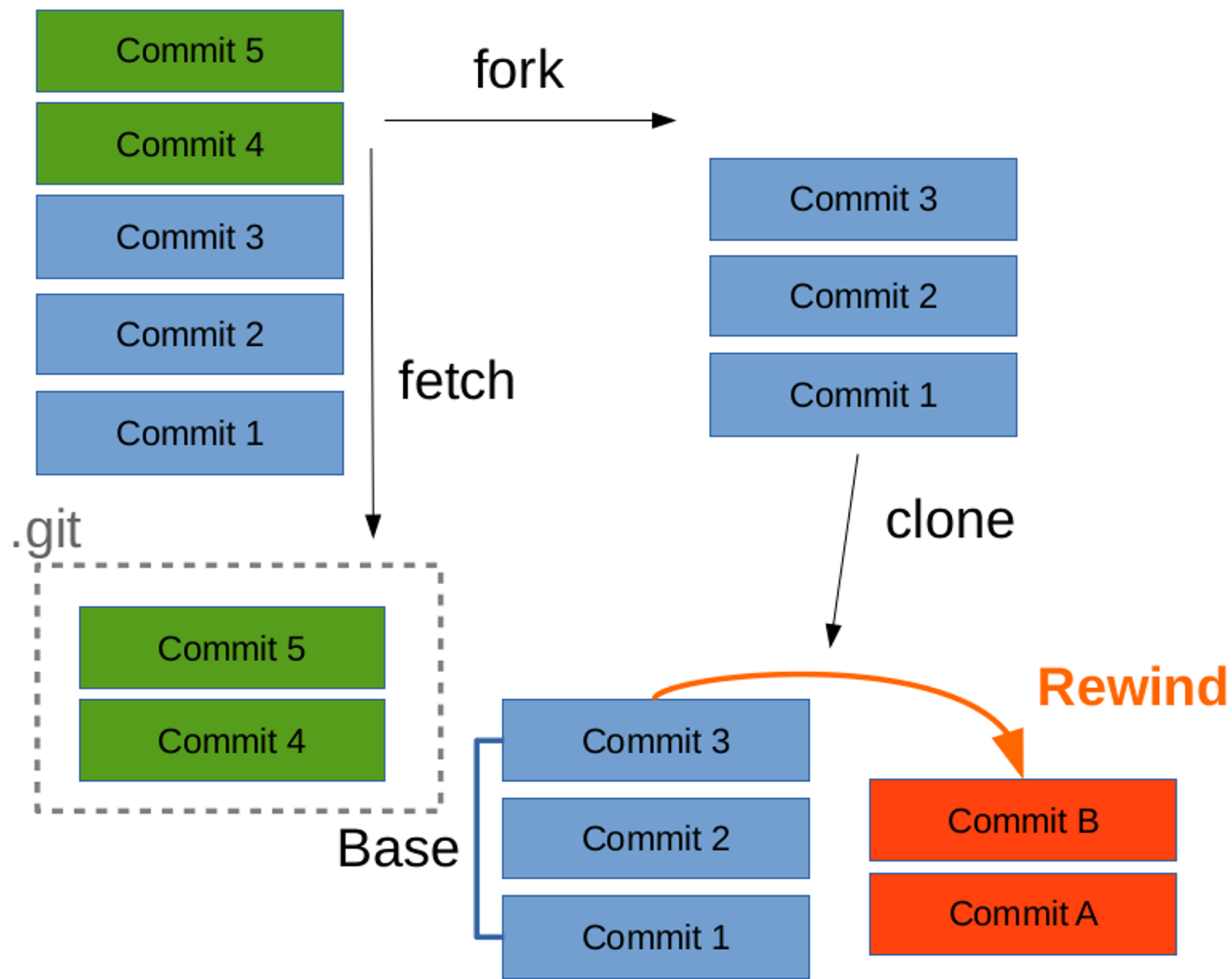
Nodejs/node

Taeung/node



Nodejs/node

Taeung/node



Nodejs/node



fork



Taeung/node



clone



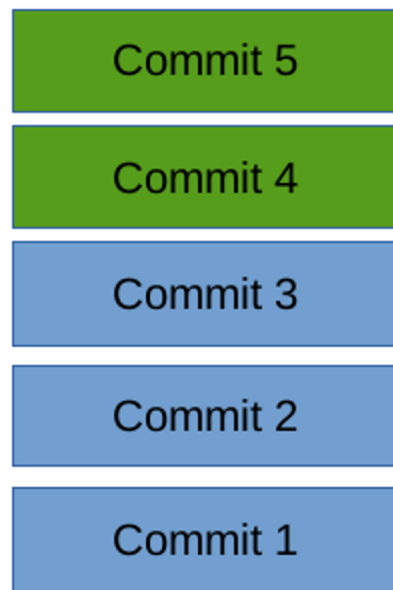
Rebase



Rewind



Nodejs/node



fork



Taeung/node



clone

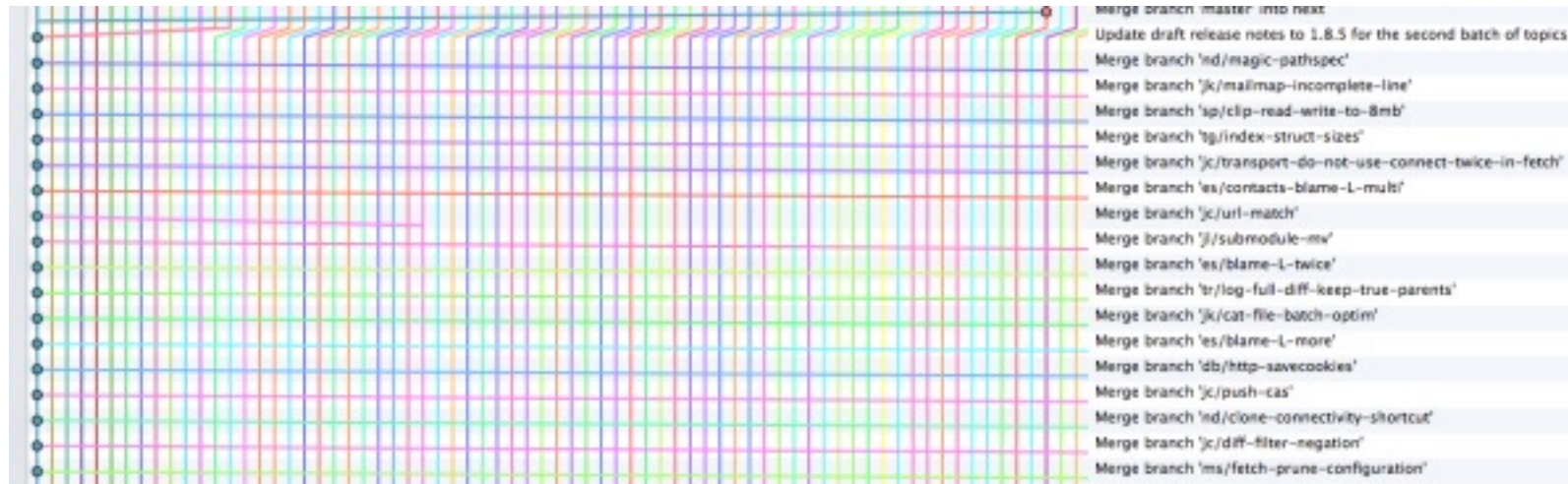


Rebase



Merge vs Rebase

- Update local with remote changes with merge
 - git pull (git fetch -> git merge)
 - Pros: traceability, Cons: complicated



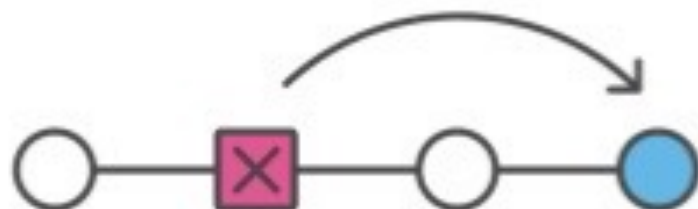
- Update local with remote changes with rebase
 - git pull --rebase (git fetch -> git rebase)
 - Pros: readability, Cons: rewrite history (dangerous)

Quiz

- Which command doesn't change history?
 - Revert
 - Reset

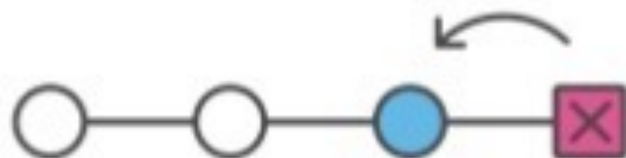
revert vs reset

Reverting



- Don't alter history
- Safe way to undo but generate more commit
- For undo shared/public commits

Resetting



- Alter history
- unsafe way to undo shared/public commits
- Cleaner history
- For undo private (non-public) commits

git reset

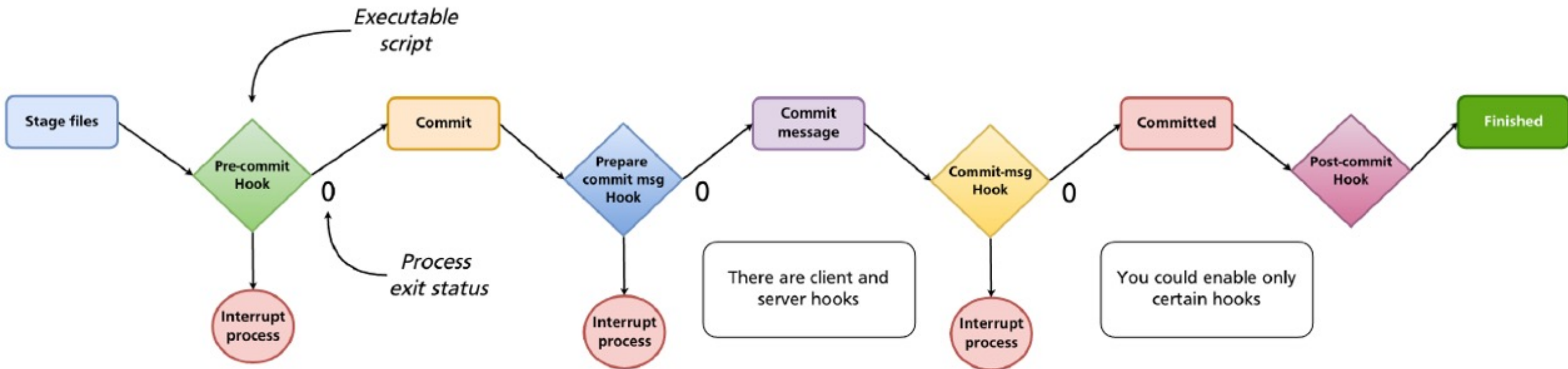
- Modify HEAD, index, working files, (logs)
 - --mixed (default): reset HEAD, index
 - --soft: reset HEAD ("Changes to be committed")
 - --hard: reset HEAD, index, working files

	working directory working tree working copy	index staging area cache	repository history tree
			git reset --soft
		git reset --mixed	
	git reset --hard		

Quiz

- Which worked same as `git branch -f master HEAD~` (current branch is not master)
 - `git reset --soft HEAD~`
 - `git reset --mixed HEAD~`
 - `git reset --hard HEAD~`

Git Hooks



Hooks

applypatch-msg prepare-commit-msg commit-msg	customise the default commit messages
pre-applypatch post-applypatch	before/after applying a patch
pre-commit post-commit	before/after commit
post-checkout pre-rebase post-merge post-rewrite	after checkout before rebase after merge after rewriting the history
pre-receive update post-receive post-update	before/during/after push (server-side)
pre-auto-gc	before automatic garbage collection (gc)

How Git Hook Used

Verify or edit commit message

Run application tests before each commit or before pushing to remote

Verify and potentially fix code syntax mistakes using linters like ESLint, Pylint

Generate notifications after successful committing

Deploy updated application to production on the server



Quiz) Troubleshooting #1

- 실수로 `git reset --hard` 를 하였다 원상 복구 하려면?
 - `git reset --hard ORIG_HEAD` (`HEAD@{1}`, `HEAD^`, `HEAD~`)
 - Better way
 - `git reflog`

Quiz) Troubleshooting #2

- 실수로 remote에 불필요한 파일을 push 했다
 - Local & remote 모두 지우고 싶으면?
 - `git rm file`
 - Remote만 지우고 싶으면?
 - `git rm --cached file`

Quiz) Troubleshooting #3

- 실수로 feature 브랜치가 아닌 master에서 작업하고 push함, master는 rollback하고 작업한 코드는 남기고 싶으면?
 - Simple
 - `git branch tmp` (backup)
 - `git reset --hard HEAD~` (reset)
 - `git push origin -f master` (force push)
 - `git checkout tmp` (restore)
 - Better
 - `git checkout feature-branch` (checkout)
 - `git merge master` (backup merge)
 - `git checkout master, git revert --no-commit hash, git commit` (revert & commit)
 - `git checkout feature-branch, git merge master` (restore & merge)
 - `git push origin master` (push)

Quiz) Troubleshooting #4

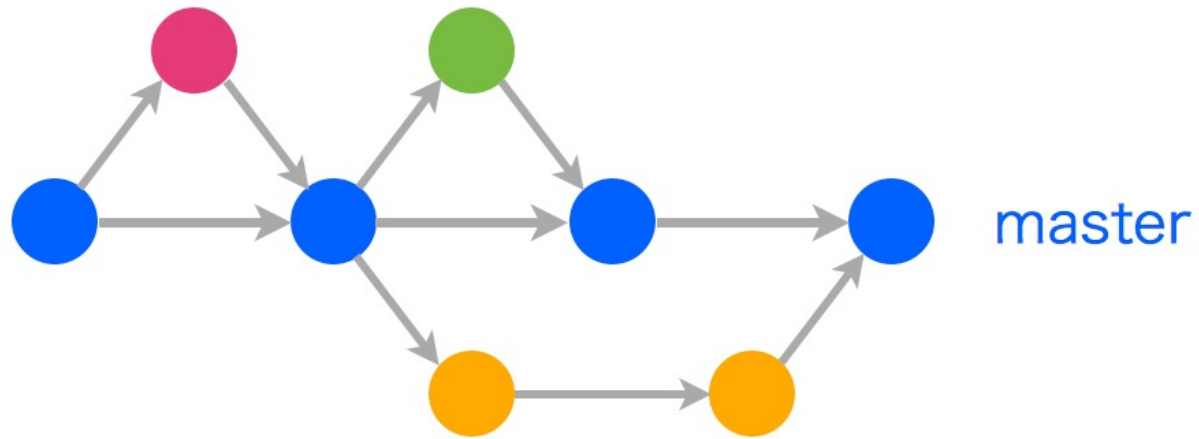
- Hotfix 해서 Tag 1.2.1 배포 후 예기치 못한 장애가 발생하여 1.2.0으로 rollback 하고 재배포하자 (Hotfix 코드는 남기고 싶다)
 - `git branch tmp` (backup)
 - `git reset --hard 1.2.0` (reset)
 - `git push origin -f master` (push)
 - `git checkout tmp` (restore)

Quiz) Troubleshooting #5

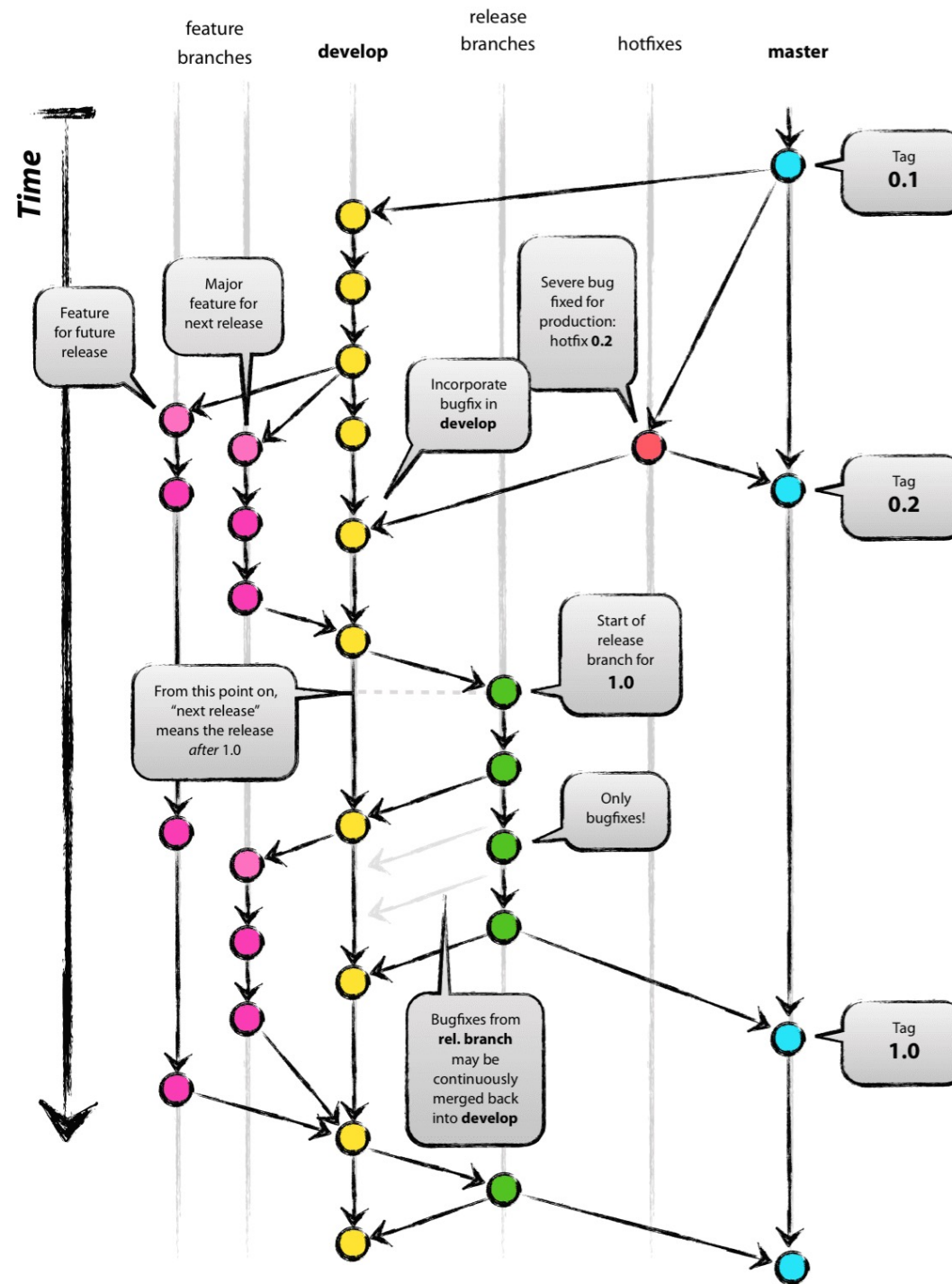
- 실수로 삭제한 브랜치를 (-D) 살리고 싶으면?
 - `git reflog` (check branch hash)
 - `git checkout -b deleted_branch hash` (restore)

Git Branch Model

GitHub flow



GitFlow



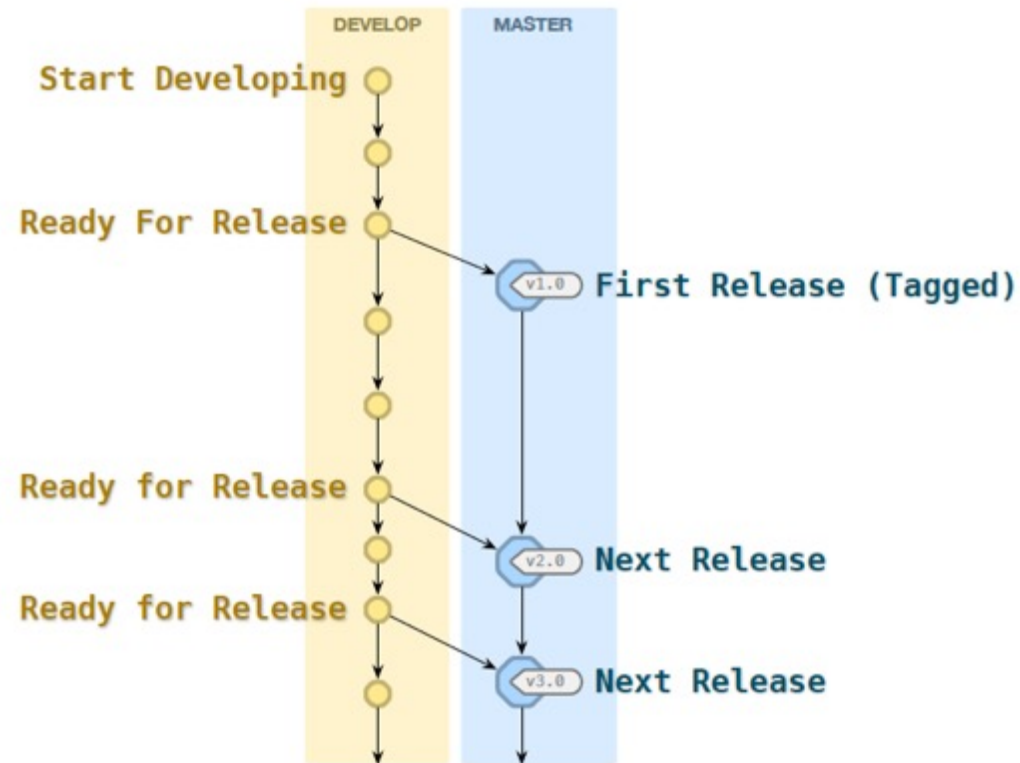
[Vincent Driessen](#)



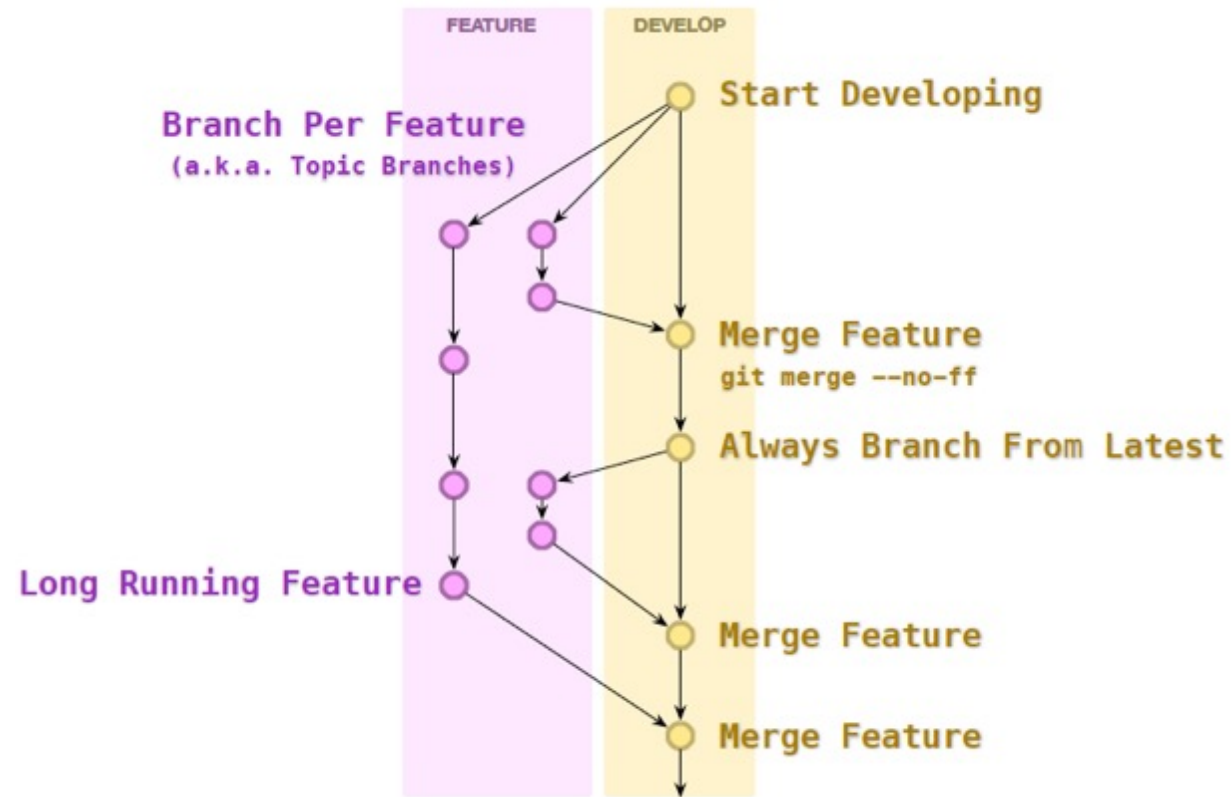
GitFlow Commands

```
$> git flow init
$> git push origin --all
$> git flow feature start <feature-branch>
$> git flow feature list
$> git add .
$> git commit -m
# push & pull-request(to review)
$> git flow feature finish <feature-branch>
$ other-feature> git merge --no-ff develop
$> git flow release start <tag>
$> git flow release finish <tag>
$> git tag
$> git push origin --all --follow-tags
$> git flow hotfix start <tag>
$> git flow hotfix finish <tag>
```

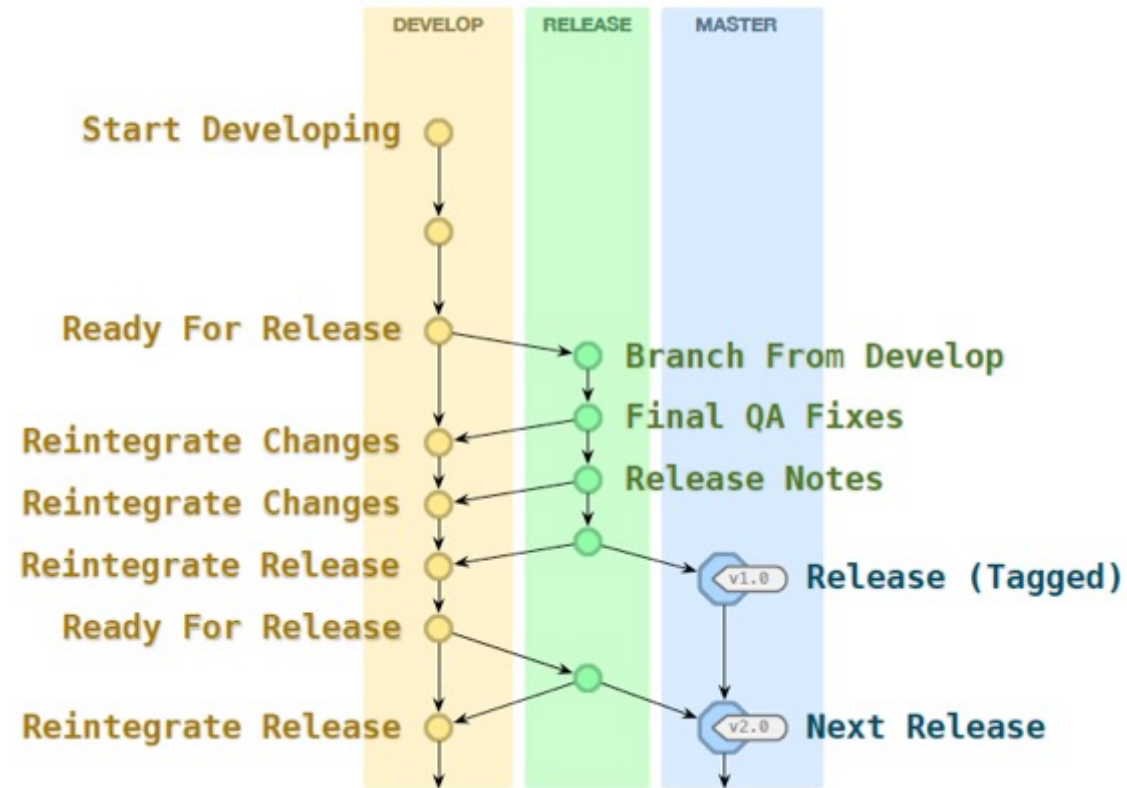
Primary Branches



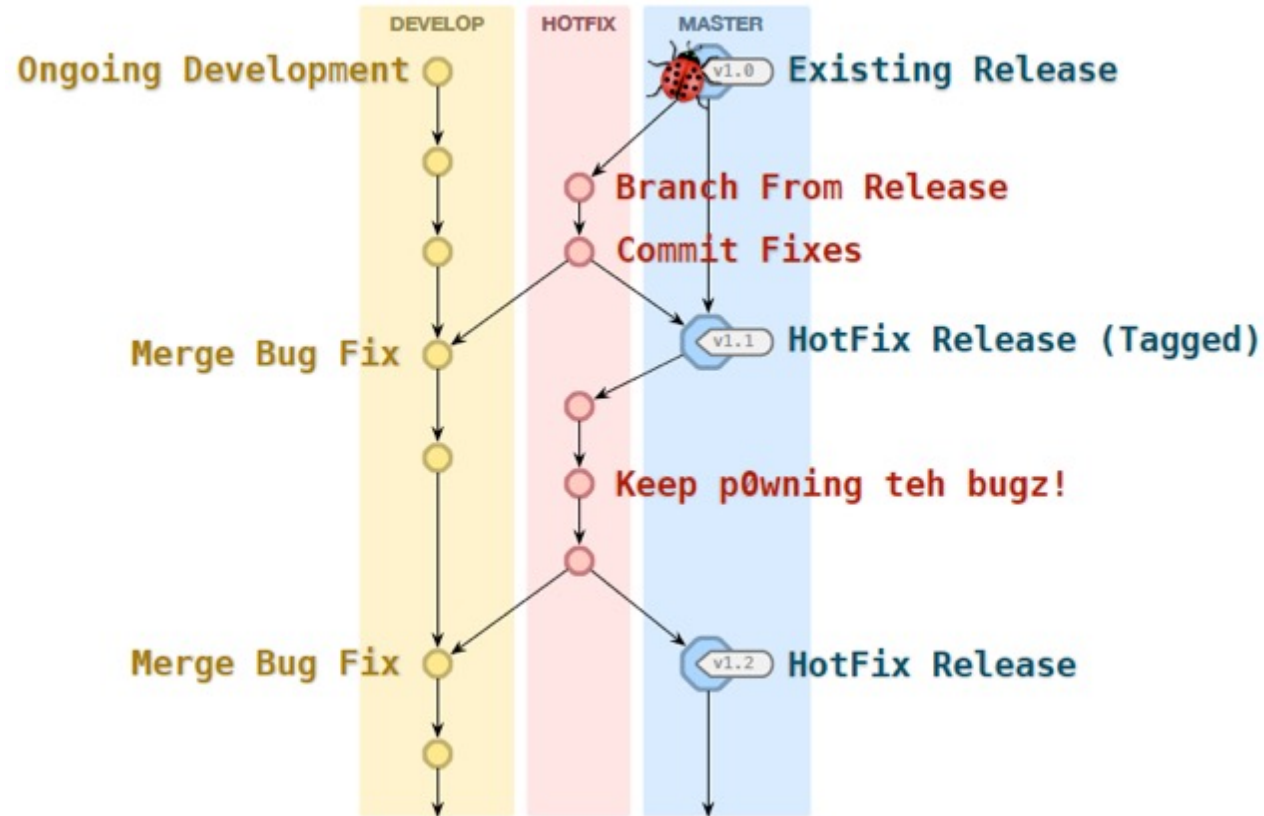
Feature Branches



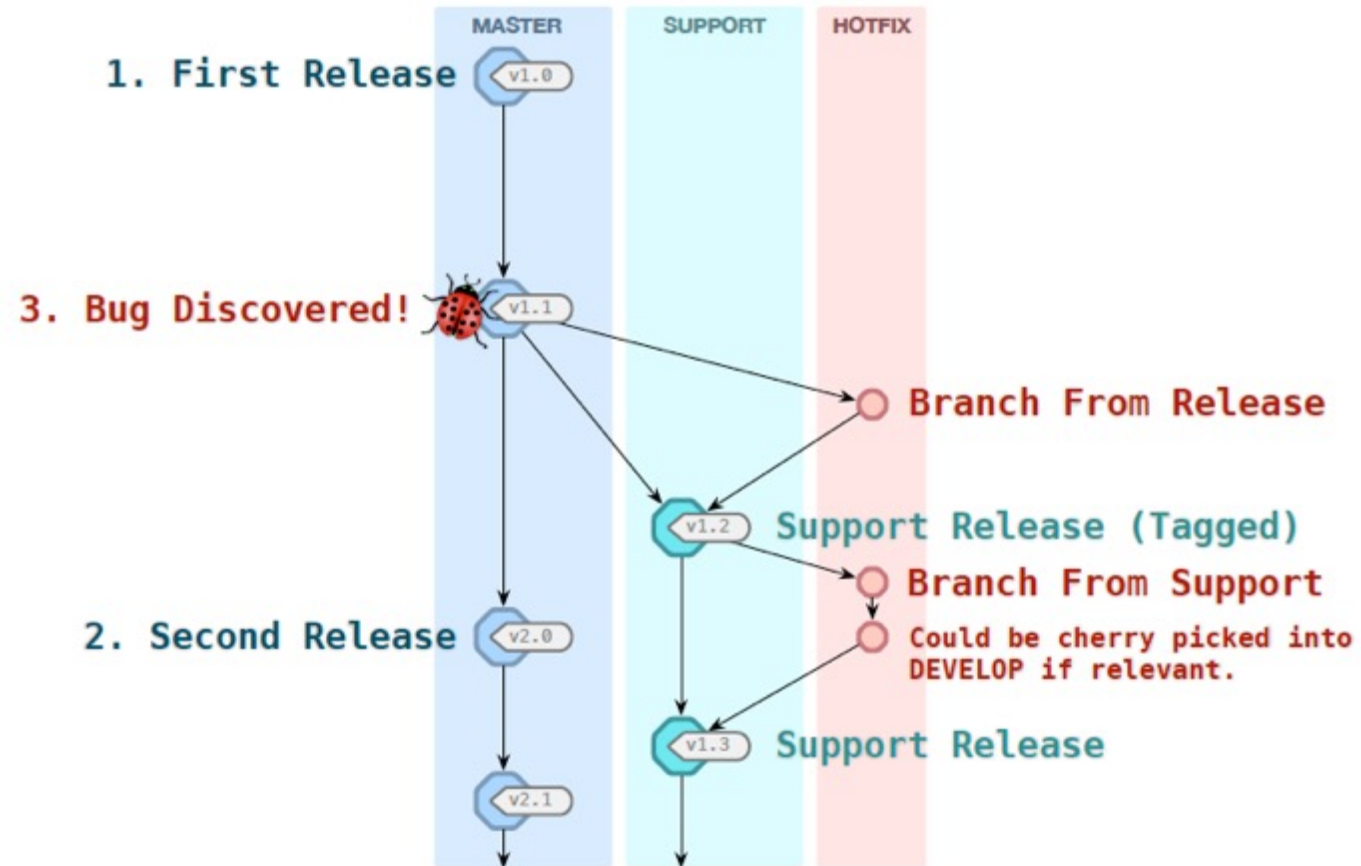
Release Branch



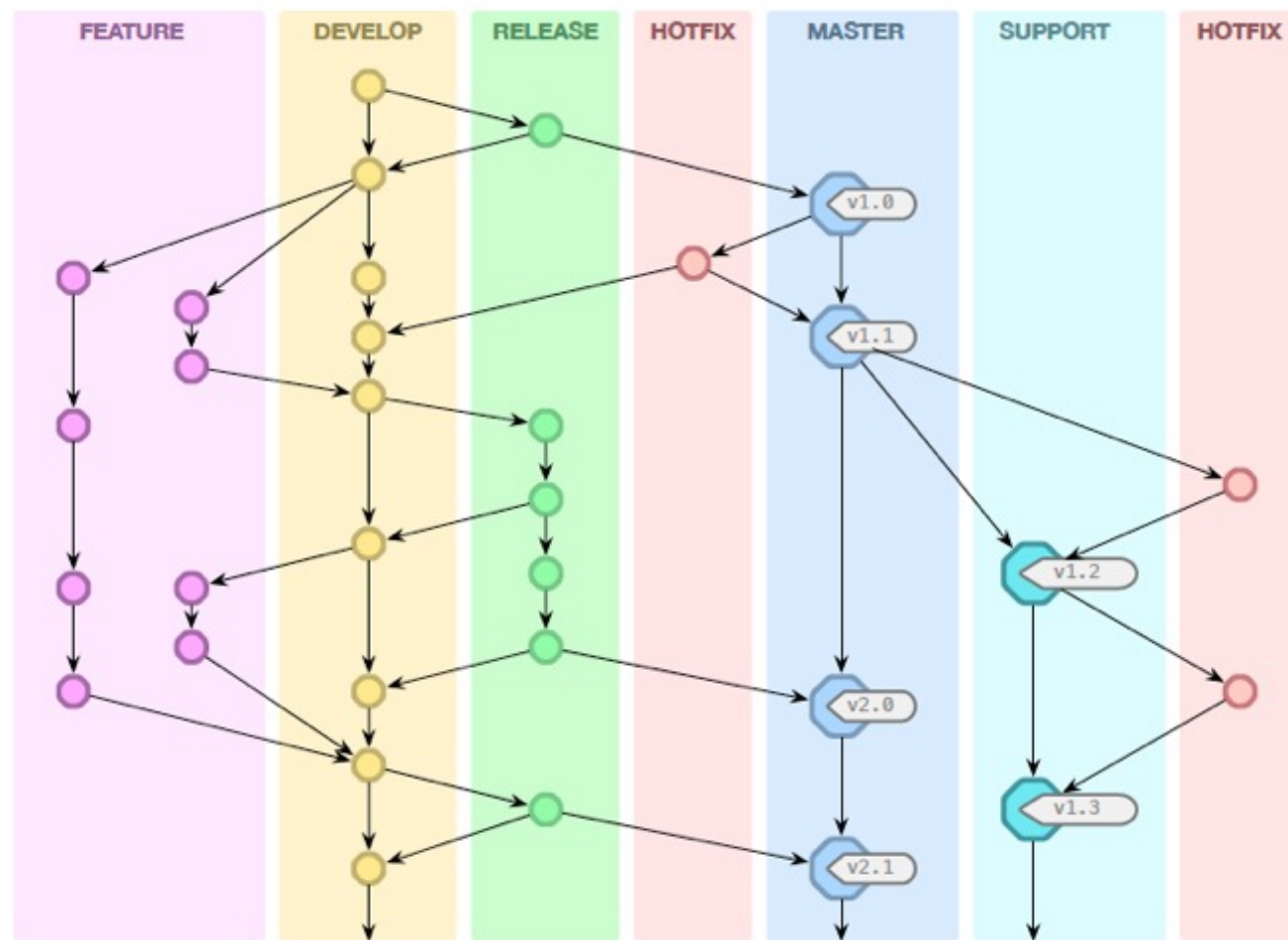
Hotfix Branch



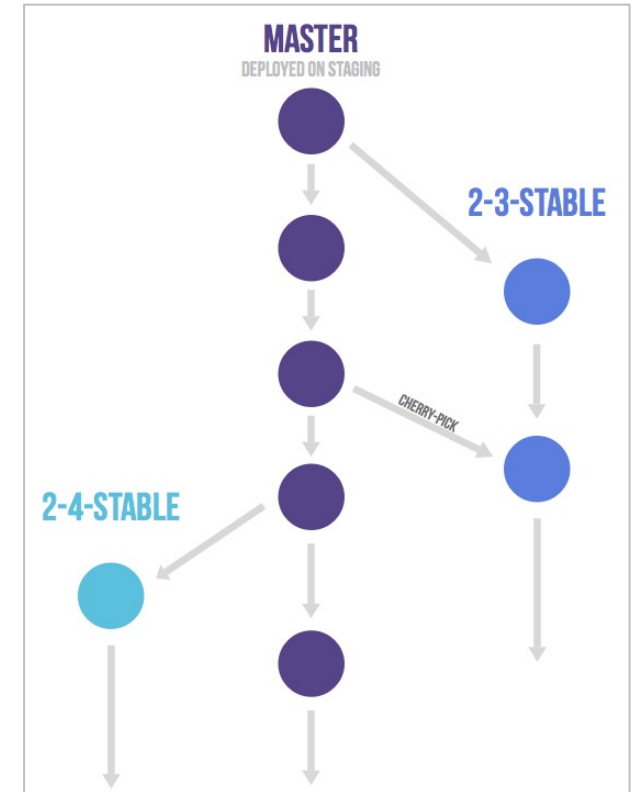
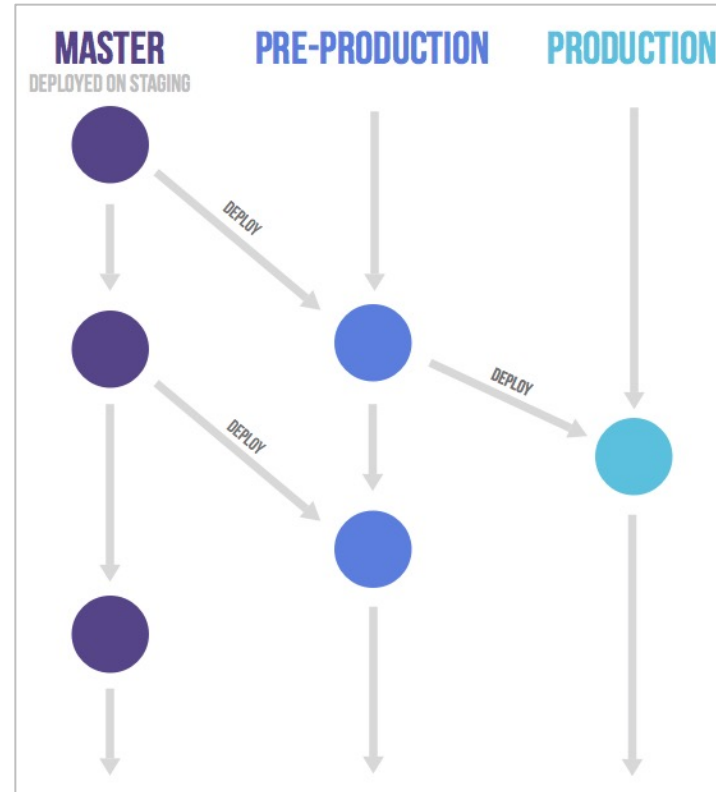
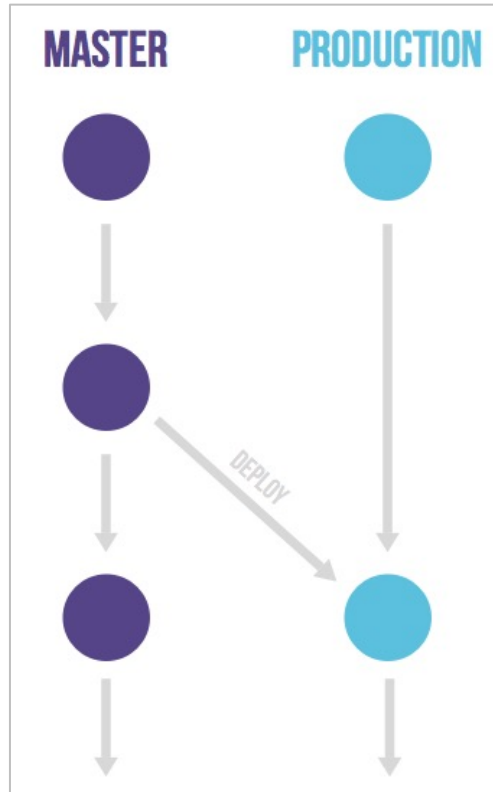
Support Branch



All Together

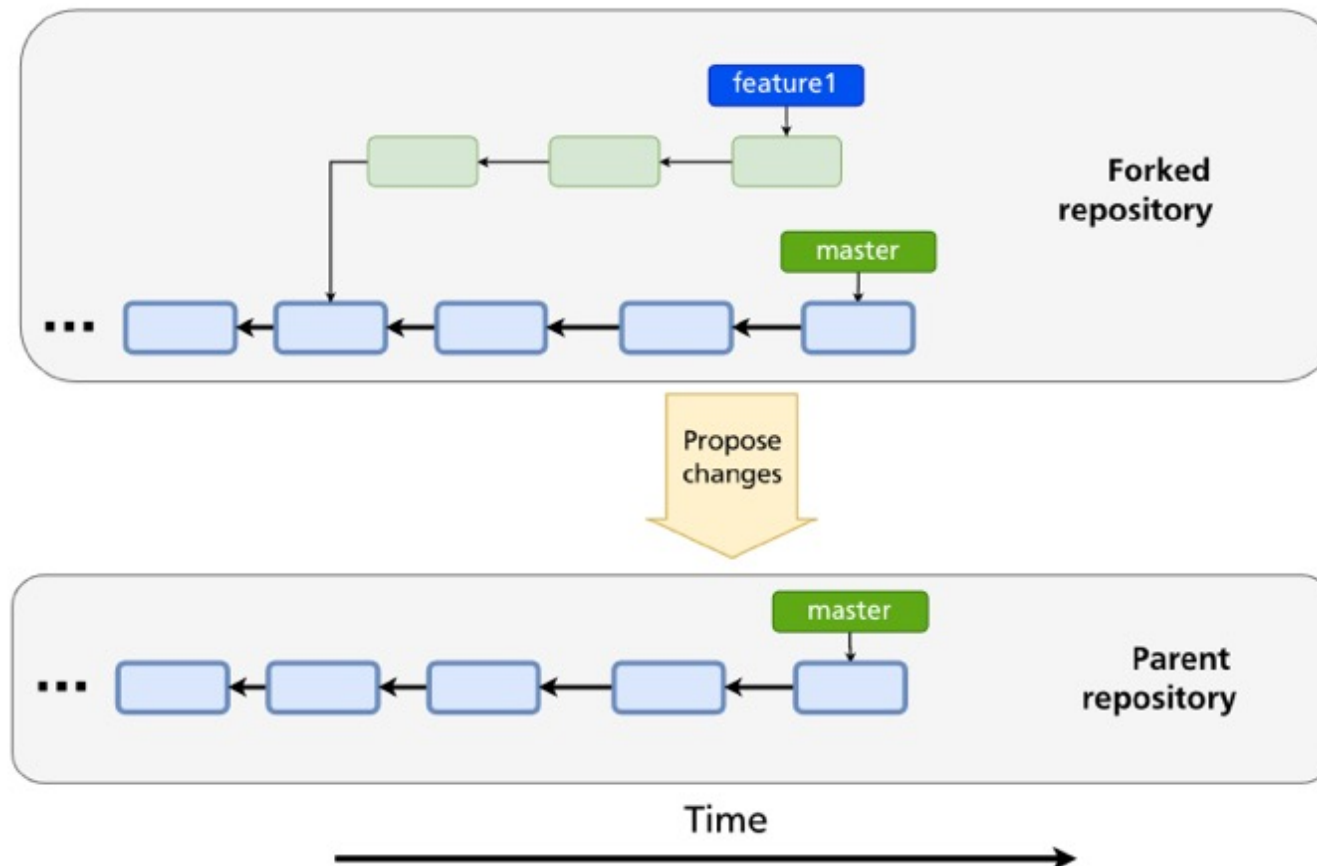


Gitlab Flow



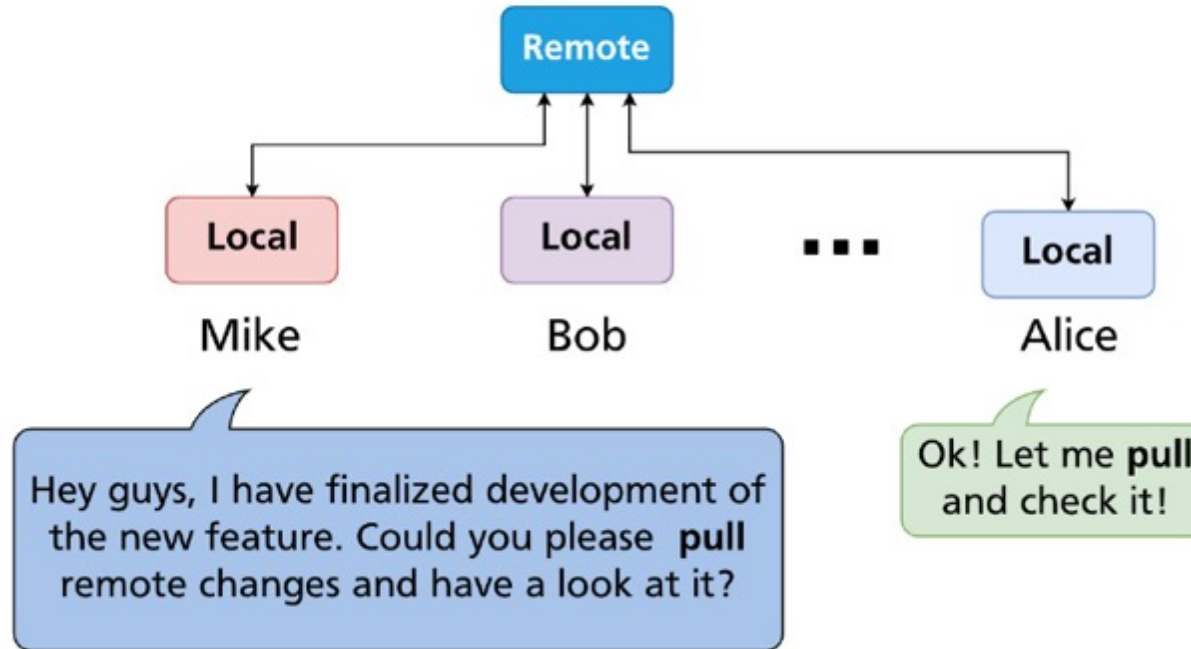
Pull Request (PR)

Pull Request: proposal of the potential changes



Why Pull Request

Why "**pull**" request?



Merge Request (MR)

 GitLab Community Edition

Overview

Repository

Issues8,730

Merge Requests472

CI / CD

Wiki

Snippets

Settings

GitLab /  GitLab.org /  GitLab Community Edition

Merge Requests

Edit Merge Requests

New merge request

Open472Merged11,188Closed1,969All13,629

Search or filter results...

Last created

test MR

!13679 · opened 17 minutes ago by Mike Greiling

1updated 14 minutes ago

Add docs for group issues page and group merge requests page

!13678 · opened 20 minutes ago by Victor Wu

0updated 2 minutes ago

Docs update links guideline to inline links

!13677 · opened 36 minutes ago by Marcia Ramos

10.0

Documentation

docs-update

0updated 34 minutes ago

WIP: Clean up new dropdown styles

0 of 1 task completed

!13676 · opened 50 minutes ago by Winnie Hellmann

10.0

Deliverable

UI polish

frontend

3updated 15 minutes ago

Greatly reduce test duration for git_access_spec

!13675 · opened 58 minutes ago by Robert Speicher

10.0

Edge

backstage

performance

technical debt

test

2updated 20 minutes ago

Implement new system note icons

0 of 11 tasks completed

!13673 · opened about 3 hours ago by Bryce Johnson

10.0

Deliverable

frontend

1updated less than a minute ago

WIP: Prepare 9.5 RC6

!13672 · opened about 3 hours ago by Jose Ivan Vargas Lopez

9-5-stable

Release

0updated about an hour ago

Use Gitaly 0.33.0

0 of 11 tasks completed

!13671 · opened about 4 hours ago by Jacob Vosmaer (GitLab)

9.5

Gitaly

Pick into Stable

4updated about 4 hours ago

[WIP] Make the import take subgroups into account

0 of 9 tasks completed

!13670 · opened about 5 hours ago by Bob Van Landuyt

10.0

Platform

0updated about 5 hours ago

Branch protection rule

Branch name pattern

Protect matching branches

☐ **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

☐ **Require status checks to pass before merging**

Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

☐ **Require signed commits**

Commits pushed to matching branches must have verified signatures.

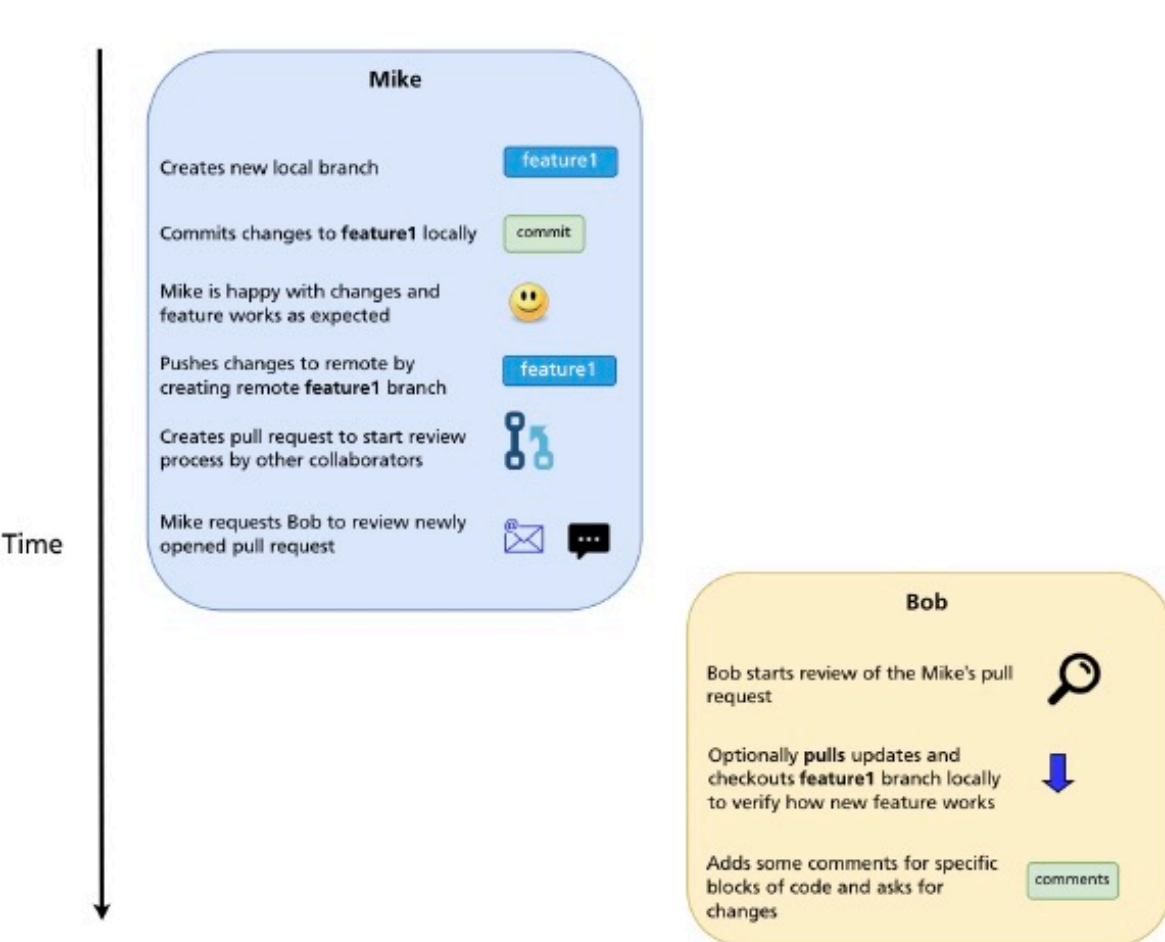
☐ **Require linear history**

Prevent merge commits from being pushed to matching branches.

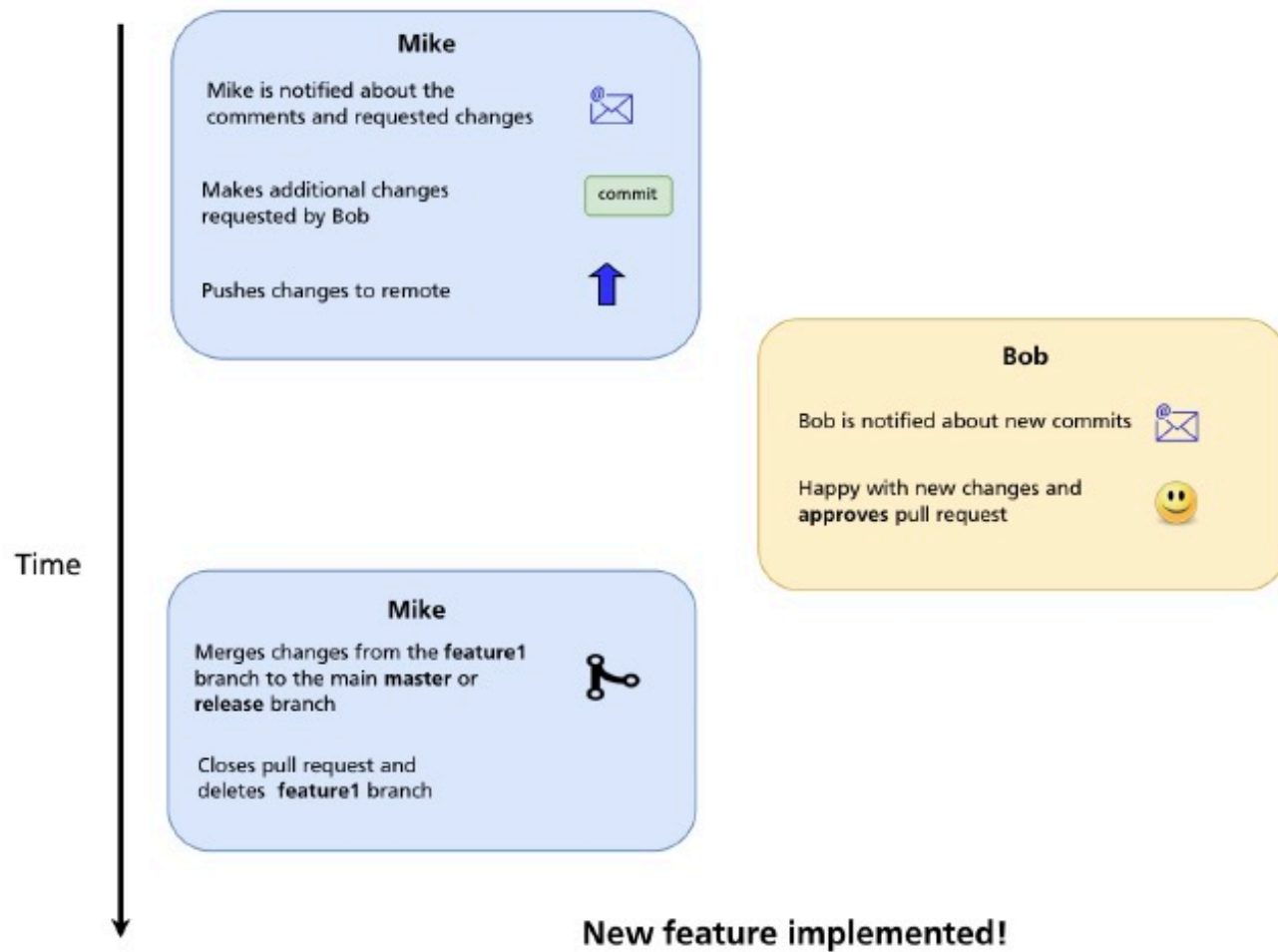
☐ **Include administrators**

Enforce all configured restrictions above for administrators.

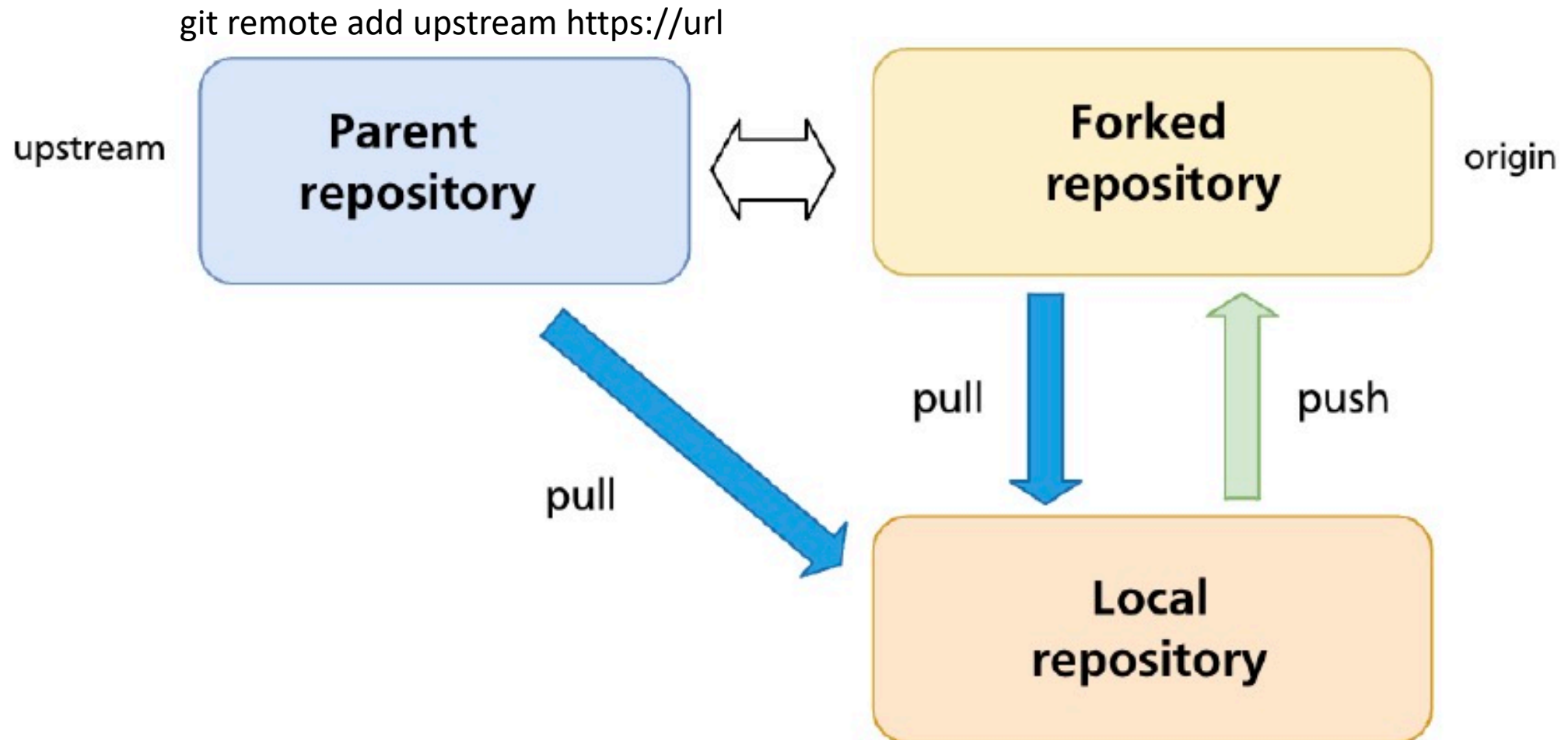
Pull request process



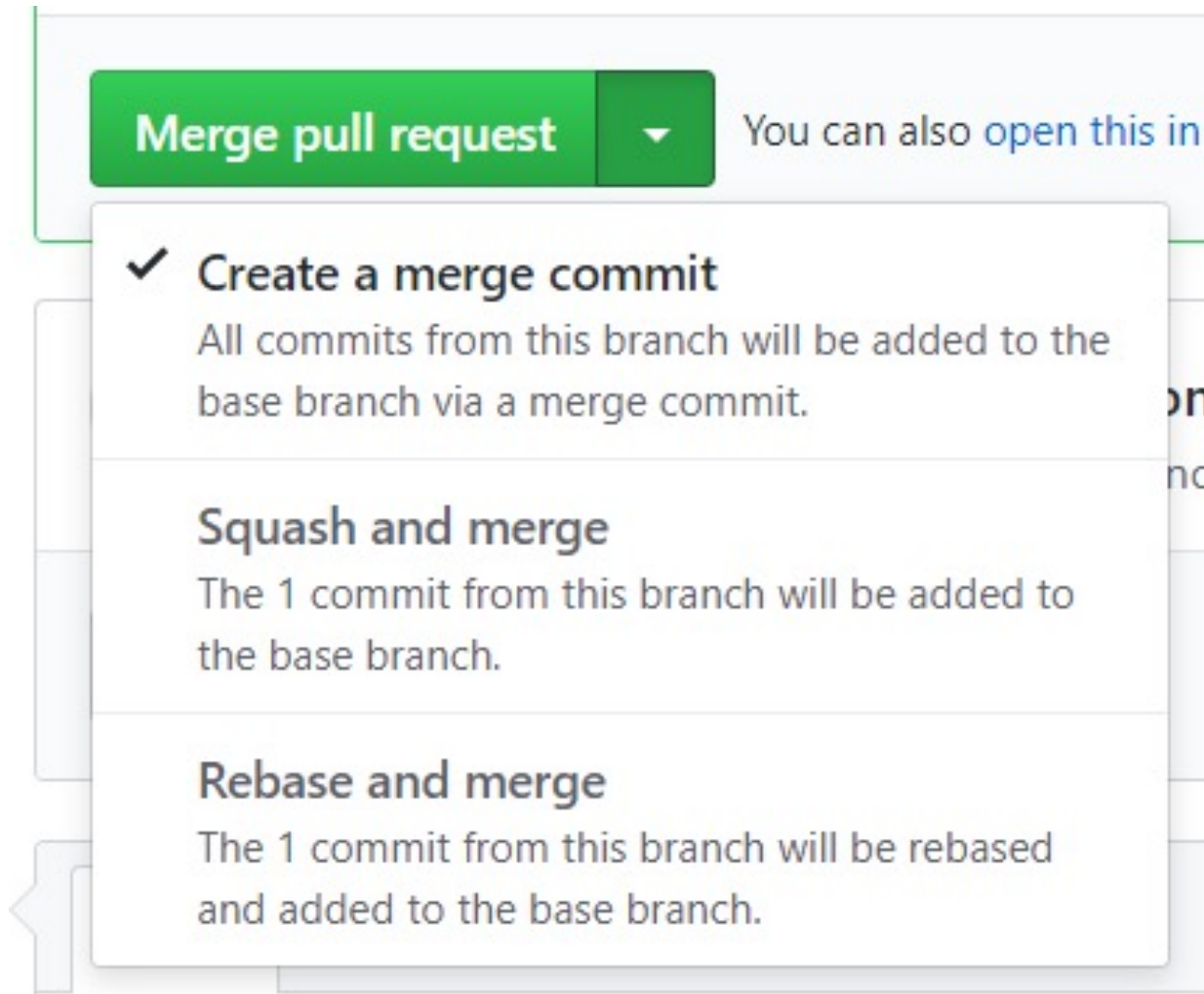
Pull request process (Cont.)



Sync Changes from Upstream



How to Merge PR?



filter: Support a float time value for trigger #76

[Edit](#)

Merged namhyung merged 3 commits into `namhyung:master` from `Taeung:time-trigger/float-threshold` on Mar 23

Conversation 11

Commits 3

Files changed 26

+147 -137



Taeung commented on Mar 20

Collaborator



The time tigger cannot handle a float value
so fix it using existing `parse_time()`.

@namhyung

I moved `parse_time()` function into `utils/utils.c`
But if you don't agree the movement,
I'll resend v2 fixing it.

Before:

```
$ uftrace record ./hello
$ uftrace replay -T main@time=1.79us
# DURATION    TID     FUNCTION
   1.013 us [19810] | __monstartup();
   0.649 us [19810] | __cxa_atexit();
                [19810] | main() {
  11.739 us [19810] | puts();
                [19810] | foo() {
   1.785 us [19810] | puts();
   2.357 us [19810] | } /* foo */
  15.040 us [19810] | } /* main */
```

After:

```
$ uftrace replay -T main@time=1.79us
# DURATION    TID     FUNCTION
   1.013 us [19810] | __monstartup();
   0.649 us [19810] | __cxa_atexit();
                [19810] | main() {
  11.739 us [19810] | puts();
   2.357 us [19810] | foo();
  15.040 us [19810] | } /* main */
```

Reviewers



namhyung



Assignees



No one—assign yourself

Labels



None yet

Milestone



No milestone

Notifications

Unsubscribe

You're receiving notifications
because you authored the thread.

2 participants



Lock conversation

☐ Allow edits from maintainers.

[Learn more](#)

commit

torvalds / linux

Watch

5,453

Unstar

45,394

Fork

17,170

Code

Pull requests 142

Projects 0

Insights

perf annotate: Use stripped line instead of raw disassemble line

Browse files

When parsing disassemble lines for source line number, use a stripped line instead of raw line.

Signed-off-by: Taeung Song <treeze.taeung@gmail.com>

Cc: Andi Kleen <ak@linux.intel.com>

Cc: Jiri Olsa <jolsa@kernel.org>

Cc: Namhyung Kim <namhyung@kernel.org>

Link: <http://lkml.kernel.org/r/1491612748-1605-3-git-send-email-treeze.taeung@gmail.com>

Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>

master v4.12-rc2 v4.12-rc1

Taeung committed with Arnaldo Carvalho de Melo on Apr 8

1 parent 4597cf0

commit 986a5bc028a84d487c354a529730b48682d1fb41

Showing 1 changed file with 2 additions and 2 deletions.

Unified

Split

4 tools/perf/util/annotate.c

View

```
@@ -1159,8 +1159,8 @@ static int symbol__parse_objdump_line(struct symbol *sym, struct map *map,
1159 1159         parsed_line = rtrim(line);
1160 1160
1161 1161         /* /filename:linenr ? Save line number and ignore. */
1162 - if (regexexec(&file_lineno, line, 2, match, 0) == 0) {
1163 -     *line_nr = atoi(line + match[1].rm_so);
1162 + if (regexexec(&file_lineno, parsed_line, 2, match, 0) == 0) {
1163 +     *line_nr = atoi(parsed_line + match[1].rm_so);
1164 1164         return 0;
1165 1165     }
```

How to Write Commit Message

- Specify the type of commit:
 - feat: The new feature you're adding to a particular application
 - fix: A bug fix
 - style: Feature and updates related to styling
 - refactor: Refactoring a specific section of the codebase
 - test: Everything related to testing
 - docs: Everything related to documentation
 - chore: Regular code maintenance.
- Use the body to explain what changes you have made and why you made them.
- Do not assume the reviewer understands what the original problem was, ensure you add it
- Do not think your code is self-explanatory
- Follow the commit convention defined by your team

commitlint

```
~/Projects/labs/commitlint master*  
> echo "foo" | commitlint  
✖ input: foo  
✖ message may not be empty [subject-empty]  
✖ type may not be empty [type-empty]  
✖ found 2 problems, 0 warnings  
  
~/Projects/labs/commitlint master*  
> echo "foo: bar" | commitlint  
✖ input: foo: bar  
✖ type must be one of [build, chore, ci, docs, feat, fix, perf, refactor, revert, style, test] [type-enum]  
✖ found 1 problems, 0 warnings  
  
~/Projects/labs/commitlint master*  
> echo "foo: bar" | commitlint
```

[Code](#)[Issues](#) 5[Pull requests](#) 1[Wiki](#)[Insights](#) ▾

filter: Support a float time value for trigger #76

[Edit](#)

Merged namhyung merged 3 commits into `namhyung:master` from `Taeung:time-trigger/float-threshold` on Mar 23

[Conversation](#) 11[Commits](#) 3[Files changed](#) 26

+147 -137

Commits on Mar 23, 2017

**filter: Support a float time value for trigger** ...

Taeung committed on Mar 21



a3e3a04

**test: Add second test case for time trigger** ...

Taeung committed on Mar 21



f51fde2

**utils: Remove string.h and ctype.h if including utils.h** ...

Taeung committed on Mar 21 ✓



c9eabc6





namhyung commented on Mar 22

Owner



Oh, I missed that. Will check!



namhyung requested changes on Mar 22

[View changes](#)

I'm not sure about the header file changes, but it seems to help writing C files with fewer lines. So I'll take it (and maybe move others later).

tests/t135_trigger_time2.py

Hide outdated

```
34 + p = sp.Popen(replay_cmd, shell=True, stdout=sp.PIPE, stderr=sp.PIPE)
35 + r = p.communicate()[0].decode()
36 + START, unit = r.split('\n')[1].split()[0:2] # skip header
37 + START = str(float(START) + 0.001) # for time filtering
```



namhyung on Mar 22

Owner

I don't think converting float into str is a good idea here. Also 'START' is not a good name, what about THRESHOLD or something?



Taeung on Mar 22 • edited

Collaborator

Okey, will change them.

```
41 +         return TestBase.TEST_SUCCESS
42 +
43 +     def runcmd(self):
44 +         return '%s replay -T main@time=%s -d %s' % (TestBase.ftrace, START,
```



namhyung on Mar 22 Owner

Here the time filter value might be longer than 3 digits. For my (failed) test case, it was `-T main@time=1.2799999999999998us` so I think it's better to print it as float with a precision limit:

```
diff --git a/tests/t135_trigger_time2.py b/tests/t135_trigger_time2.py
index f16230d..268f844 100644
--- a/tests/t135_trigger_time2.py
+++ b/tests/t135_trigger_time2.py
@@ -4,7 +4,8 @@ from runtest import TestBase
import subprocess as sp

TDIR='xxx'
-START=0
+TIME=0
+UNIT=''

class TestCase(TestBase):
    def __init__(self):
@@ -24,7 +25,7 @@ class TestCase(TestBase):
    """', sort='simple')

    def pre(self):
-        global START
+        global TIME, UNIT

        record_cmd = '%s record -F main -d %s %s' % (TestBase.ftrace, TDIR, '
        sp.call(record_cmd.split())
@@ -33,15 +34,14 @@ class TestCase(TestBase):
```




Taeung added some commits on Mar 21



filter: Support a float time value for trigger ...

a3e3a04



test: Add second test case for time trigger ...

f51fde2



Taeung commented on Mar 23 • edited

Collaborator



@**namhyung** I sent v3 according to your changes including 'signed off by you'. 🐼



utils: Remove string.h and ctype.h if including utils.h ...



c9eabc6



Taeung commented on Mar 23

Collaborator



@**namhyung** Oops, I missed changing the commit ID in the change log ([Taeung/uftace@ c9eabc6](#)), so fixed it now !



namhyung merged commit **4c0def3** into **namhyung:master** on Mar 23

1 check passed

[View details](#)

[Revert](#)



Pull request successfully merged and closed

You're all set—the **Taeung:time-trigger/float...** branch can be safely deleted.

[Delete branch](#)

dump: support ctf output format #80

[Edit](#)[Open](#) abenbachir wants to merge 1 commit into [namhyung:master](#) from [abenbachir:support-ctf-format](#)[Conversation](#) 11[Commits](#) 1[Files changed](#) 11

+352 -4



abenbachir commented on Apr 2 • edited

[Contributor](#) + 🗨️ ✎️

Hi @namhyung,

This pull request introduce a new dump format: [Comment Trace Format \(CTF\)](#), which is a binary trace format mostly used by a powerfull tracer known as Lttng.

One dependency would be added in order to use CTF features: libbabeltrace-ctf (1.5 or 2.0)

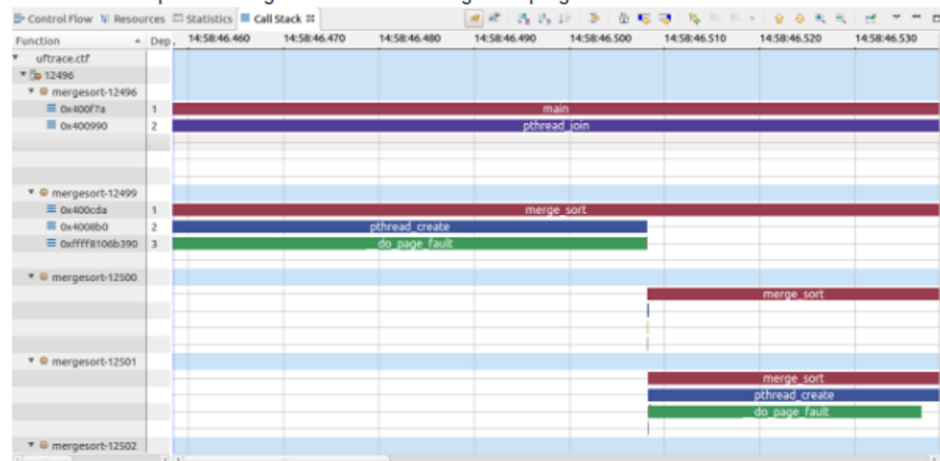
This CTF conversion will provide developers the possibility of using [Trace Compass](#), an application for viewing and analyzing traces which support CTF.

Trace Compass is powerfull and very optimized for analyzing very big traces as 10 GB of traces data, but it also comes with many configurable and builtin views like CPU, Memory usage, Control flow and, stack view and so on.

The call stack view will be used to display the callgraph dumped from ufttrace.

I intent to write documentations on how to use Trace Compass features with ufttrace.

I made an example of tracing a multithreaded merge sort program :



namhyung commented on Apr 3 • edited

[Owner](#) + 🗨️ ✎️ ✕

@abenbachir Thanks for contributing this!! I'm really happy to see ufttrace supports the CTF format.

However I think the babeltrace library is not very popular so we need to opt-out if it's not installed on the system. Also my system has an old version of the library so I see an error like following:

[Reviewers](#) ⚙️[Suggestions](#)[namhyung](#) ⓘ[Assignees](#) ⚙️

No one—assign yourself

[Labels](#) ⚙️

None yet

[Milestone](#) ⚙️

No milestone

[Notifications](#)[Unsubscribe](#)

You're receiving notifications because you're subscribed to this repository.

4 participants

[Lock conversation](#)

Make the configure script more packaging-friendly #98

[Edit](#)

Merged namhyung merged 6 commits into `namhyung:master` from `dridi:configure` 8 days ago

Conversation 19

Commits 6

Files changed 1

+26 -23



Dridi commented 9 days ago

Contributor + 😊 ✎

Hello,

Someone's trying to introduce uftrace on Fedora [1] and I'd like to see this move forward. One problem is that the configure script doesn't meet packaging expectations, which is fixed in [77dcd9c](#) [ca9d704](#) . That should make downstream maintainers lives easier since they'd less likely need to manually update the packaging whenever a guideline changes.

I also polished the script a bit.

Dridi

[1] https://bugzilla.redhat.com/show_bug.cgi?id=1398922



namhyung commented 9 days ago • edited

Owner + 😊 ✎ ✕

Thank you for fixing this, all look good to me.

@thepaul @bkircher @ganadist Are you ok with this change?



thepaul reviewed 9 days ago

[View changes](#)

configure

Show outdated



thepaul commented 9 days ago

Contributor + 😊 ✎ ✕

Ok with the change other than my quibble about `${foo%x}` vs `${foo%*}` . This won't affect the deb packaging much.

Reviewers



thepaul



namhyung



Assignees



No one—assign yourself

Labels



None yet

Milestone



No milestone

Notifications

🔊 Unsubscribe

You're receiving notifications because you're subscribed to this repository.

4 participants



🔒 Lock conversation

Comments Acronyms

- **LGTM: “Looks Good To Me”**
- **WFM: “Works For Me”**
- **WIP: “Work In Progress”**
- **AFAIK: “As Far As I Know”**
- **IMO: “In My Opinion”**
- **r?: review please, r+: reviewed + approved**
- **Principles**
 - **KISS: Keep It Simple and Stupid**
 - **DRY: Don’t Repeat Yourself**
 - **YAGNI: You Aren’t Gonna Need It**

Contributing Opensource Projects

- **tensorflow**
- **react**
- **elasticsearch**
- **node.js**
- **spring-boot**

tensorflow / tensorflow

Watch

5,192

★ Unstar

58,457

Fork

27,953

Code

Issues 1,134

Pull requests 56

Projects 0

Insights

<https://github.com/tensorflow/tensorflow>

Computation using data flow graphs for scalable machine learning <http://tensorflow.org>

tensorflow

machine-learning

python

deep-learning

deep-neural-networks

neural-network

ml

distributed

17,892 commits

15 branches

30 releases

852 contributors

Apache-2.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

maciekc committed on GitHub Merge pull request #10181 from maciekc/master

Latest commit a7bf76b 2 hours ago

tensorflow	Fix Rnn tests broken by move.	15 hours ago
third_party	Updated libxsmm kernels (#10112)	3 days ago
tools	Merge changes from github.	14 days ago
util/python	Remove deleted files.	20 days ago
.gitignore	Merge changes from github.	14 days ago
ACKNOWLEDGMENTS	TensorFlow: Improve performance of Alexnet	2 years ago
ADOPTERS.md	Internal file cleanup.	7 months ago
AUTHORS	Merge changes from github.	11 months ago
BUILD	Depend on protobuf's header only library when building custom ops	3 months ago
CONTRIBUTING.md	Merge changes from github.	14 days ago
ISSUE_TEMPLATE.md	Merge changes from github.	20 days ago

 **chris-chris** Patch/170506 misspell (#9725)

4be052a 19 days ago

5 contributors 

167 lines (116 sloc) | 7.28 KB

Raw

Blame

History



Contributing guidelines

How to become a contributor and submit your own code

Contributor License Agreements

We'd love to accept your patches! Before we can take them, we have to jump a couple of legal hurdles.

Please fill out either the individual or corporate Contributor License Agreement (CLA).

- If you are an individual writing original source code and you're sure you own the intellectual property, then you'll need to sign an [individual CLA](#).
- If you work for a company that wants to allow you to contribute your work, then you'll need to sign a [corporate CLA](#).

Follow either of the two links above to access the appropriate CLA and instructions for how to sign and return it. Once we receive it, we'll be able to accept your pull requests.

NOTE: Only original source code from you and other people that have signed the CLA can be accepted into the main repository.

Contributing code

If you have improvements to TensorFlow, send us your pull requests! For those just getting started, Github has a [howto](#).

If you want to contribute but you're not sure where to start, take a look at the [issues with the "contributions welcome" label](#). These are issues that we believe are particularly well suited for outside contributions, often because we probably won't get to them right now. If you decide to start on an issue, leave a comment so that other people know that you're working on it. If you want to help out. but not alone. use the issue comment thread to coordinate.

Products > Open Source Programs Office

Contributor License Agreements

Google Individual Contributor License Agreement

In order to clarify the intellectual property license granted with Contributions from any person or entity, Google Inc. ("Google") must have a Contributor License Agreement ("CLA") on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of Google; it does not change your rights to use your own Contributions for any other purpose.

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to Google. Except for the license granted herein to Google and recipients of software distributed by Google, You reserve all right, title, and interest in and to Your Contributions.

1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with Google. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to Google for inclusion in, or documentation of, any of the products owned or managed by Google (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to Google or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, Google for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

2. Grant of Copyright License. Subject to the terms and conditions of this Agreement, You hereby grant to Google and to recipients of software distributed by Google a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

<https://cla.developers.google.com/about/google-individual?csw=1>



Contributing guidelines

How to become a contributor and submit your own code

Contributor License Agreements

We'd love to accept your patches! Before we can take them, we have to jump a couple of legal hurdles.

Please fill out either the individual or corporate Contributor License Agreement (CLA).

- If you are an individual writing original source code and you're sure you own the intellectual property, then you'll need to sign an [individual CLA](#).
- If you work for a company that wants to allow you to contribute your work, then you'll need to sign a [corporate CLA](#).

Follow either of the two links above to access the appropriate CLA and instructions for how to sign and return it. Once we receive it, we'll be able to accept your pull requests.

NOTE: Only original source code from you and other people that have signed the CLA can be accepted into the main repository.

Contributing code

If you have improvements to TensorFlow, send us your pull requests! For those just getting started, Github has a [howto](#).

If you want to contribute but you're not sure where to start, take a look at the [issues with the "contributions welcome" label](#). These are issues that we believe are particularly well suited for outside contributions, often because we probably won't get to them right now. If you decide to start on an issue, leave a comment so that other people know that you're working on it. If you want to help out. but not alone. use the issue comment thread to coordinate.

Code

Issues 1,130

Pull requests 54

Projects 0

Insights

Filters

is:open label:"stat:contributions welcome"

Labels

Milestones

New issue

Clear current search query, filters, and sorts

358 Open 239 Closed

Author

Labels

Projects

Milestones

Assignee

Sort

! compilation errors due to missing op classes when using selective registration (cmake windows build) stat:contributions welcome

#10080 opened 4 days ago by adennie

1

! Cannot build jemalloc support using CMake on Linux (fails trying to include <windows.h>)

stat:contributions welcome type:build/install

#10071 opened 5 days ago by HFTrader

5

! Class weighting in tf.losses.softmax_cross_entropy stat:contributions welcome type:feature

#10021 opened 7 days ago by w4nderlust

1

! [feature] sparse_multiply stat:contributions welcome type:feature

#10003 opened 7 days ago by asiakaczmar

1

! [feature] sparse_reduce_max and sparse_reduce_max_sparse stat:contributions welcome

type:feature

#10002 opened 7 days ago by asiakaczmar

3

! Convolution_transpose layer now gives an error (Tensorflow 1.0.0). stat:contributions welcome

3

<https://github.com/tensorflow/tensorflow/labels/stat%3Acontributions%20welcome>

Contribution guidelines and standards

Before sending your pull request for [review](#), make sure your changes are consistent with the guidelines and follow the TensorFlow coding style.

General guidelines and philosophy for contribution

- Include unit tests when you contribute new features, as they help to a) prove that your code works correctly, b) guard against future breaking changes to lower the maintenance cost.
- Bug fixes also generally require unit tests, because the presence of bugs usually indicates insufficient test coverage.
- Keep API compatibility in mind when you change code in core TensorFlow, e.g., code in [tensorflow/core](#) and [tensorflow/python](#). TensorFlow has reached version 1 and hence cannot make non-backward-compatible API changes without a major release. Reviewers of your pull request will comment on any API compatibility issues.
- When you contribute a new feature to TensorFlow, the maintenance burden is (by default) transferred to the TensorFlow team. This means that benefit of contribution must be compared against the cost of maintaining the feature.
- Full new features (e.g., a new op implementing a cutting-edge algorithm) typically will live in [tensorflow/contrib](#) to get some airtime before decision is made regarding whether they are to be migrated to the core.

License

Include a license at the top of new files.

- [C/C++ license example](#)
- [Python license example](#)
- [Java license example](#)
- [Go license example](#)
- [Bash license example](#)
- [HTML license example](#)
- [JavaScript/TypeScript license example](#)

Bazel BUILD files also need to include a license section, e.g., [BUILD example](#).

C++ coding style

Changes to TensorFlow C++ code should conform to [Google C++ Style Guide](#).

Google C++ Style Guide

Table of Contents

Header Files	Self-contained Headers The #define Guard Forward Declarations Inline Functions Names and Order of Includes
Scoping	Namespaces Unnamed Namespaces and Static Variables Nonmember, Static Member, and Global Functions Local Variables Static and Global Variables
Classes	Doing Work in Constructors Implicit Conversions Copyable and Movable Types Structs vs. Classes Inheritance Multiple Inheritance Interfaces Operator Overloading Access Control Declaration Order
Functions	Parameter Ordering Write Short Functions Reference Arguments Function Overloading Default Arguments Trailing Return Type Syntax
Google-Specific Magic	Ownership and Smart Pointers cpplint
Other C++ Features	Rvalue References Friends Exceptions Run-Time Type Information (RTTI) Casting Streams Preincrement and Predecrement Use of const Use of constexpr Integer Types 64-bit Portability Preprocessor Macros 0 and nullptr/NULL sizeof auto Braced Initializer List Lambda expressions Template metaprogramming Boost std::hash C++11 Nonstandard Extensions Aliases
Naming	General Naming Rules File Names Type Names Variable Names Constant Names Function Names Namespace Names Enumerator Names Macro Names Exceptions to Naming Rules
Comments	Comment Style File Comments Class Comments Function Comments Variable Comments Implementation Comments Punctuation, Spelling and Grammar TODO Comments Deprecation Comments

<https://google.github.io/styleguide/cppguide.html>

Running sanity check

If you have Docker installed on your system, you can perform a sanity check on your changes by running the command:

```
tensorflow/tools/ci_build/ci_build.sh CPU tensorflow/tools/ci_build/ci_sanity.sh
```

This will catch most license, Python coding style and BUILD file issues that may exist in your changes.

Running unit tests

There are two ways to run TensorFlow unit tests.

1. Using tools and libraries installed directly on your system.

Refer to the [CPU-only developer Dockerfile](#) and [GPU developer Dockerfile](#) for the required packages. Alternatively, use the said [Docker images](#), e.g., `tensorflow/tensorflow:nightly-devel` and `tensorflow/tensorflow:nightly-devel-gpu` for development to avoid installing the packages directly on your system.

Once you have the packages installed, you can run a specific unit test in bazel by doing as follows:

If the tests are to be run on GPU, add CUDA paths to `LD_LIBRARY_PATH` and add the `cuda` option flag

```
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64:  
export flags="--config=opt --config=cuda -k"
```

For example, to run all tests under tensorflow/python, do:

```
bazel test ${flags} //tensorflow/python/...
```

2. Using Docker and TensorFlow's CI scripts.

See [TensorFlow Builds](#) for details.

 **skye** Update "Using TensorFlow via Docker" README to mention memory limits.

fcfb548 24 days ago

8 contributors



116 lines (79 sloc) | 4.48 KB

Raw

Blame

History



TensorFlow Builds

This directory contains all the files and setup instructions to run all the important builds and tests. **You can trivially run it yourself!** It also run continuous integration ci.tensorflow.org.

Run It Yourself

1. Install [Docker](#). Follow instructions [on the Docker site](#).

You can run all the jobs **without docker** if you are on mac or on linux and you just don't want docker. Just install all the dependencies from [Installing TensorFlow](#). Then run any of the one liners below without the `tensorflow/tools/ci_build/ci_build.sh` in them.

2. Clone tensorflow repository.

```
git clone https://github.com/tensorflow/tensorflow.git
```

3. Go to tensorflow directory

```
cd tensorflow
```

4. Build what you want, for example

```
tensorflow/tools/ci_build/ci_build.sh CPU bazel test //tensorflow/...
```


Update mnist.py #10188

Merged

maciekcc merged 1 commit into tensorflow:master from RyanKung:patch-2 an hour ago

Conversation 6

Commits 1

Files changed 1

+7 -11



RyanKung commented 9 hours ago

Contributor +😊

I am reading/learning the code, Read by Refactoring. For clean and clear.



Update mnist.py ...

✓ 8e195d9



tensorflow-jenkins commented 9 hours ago

Collaborator +😊

Can one of the admins verify this patch?



googlebot commented 9 hours ago

+😊

Thanks for your pull request. It looks like this may be your first contribution to a Google open source project. Before we can look at your pull request, you'll need to sign a Contributor License Agreement (CLA).

📄 Please visit <https://cla.developers.google.com/> to sign.

Once you've signed, please reply here (e.g. `I signed it!`) and we'll verify. Thanks.

- If you've already signed a CLA, it's possible we don't have your GitHub username or you're using a different email address. Check [your existing CLA data](#) and verify that your [email is set on your git commits](#).

Reviewers

maciekcc ✓

Assignees

No one assigned

Labels

cla: yes

Projects

None yet

Milestone

No milestone

Notifications

🔊 Subscribe

You're not receiving notifications from this thread.

4 participants



Changes from all commits ▾ 1 file ▾ +7 -11 ■■■■

Unified

Split

Review changes ▾

Update mnist.py

clean and clear



RyanKung committed on GitHub 9 hours ago

commit 8e195d991452af81b467192e538fe8f459d7c9c0

18 ■■■■ tensorflow/contrib/learn/python/learn/datasets/mnist.py

View ▾

✱	@@ -261,17 +261,13 @@ def fake():
261	261 train_images = train_images[validation_size:]
262	262 train_labels = train_labels[validation_size:]
263	263
264	- train = DataSet(
265	- train_images, train_labels, dtype=dtype, reshape=reshape, seed=seed)
266	- validation = DataSet(
267	- validation_images,
268	- validation_labels,
269	- dtype=dtype,
270	- reshape=reshape,
271	- seed=seed)
272	- test = DataSet(
273	- test_images, test_labels, dtype=dtype, reshape=reshape, seed=seed)
274	-
264	+ options = dict(dtype=dtype, reshape=reshape, seed=seed)
265	+ <u>train = DataSet(train_images, train_labels, **options)</u>
266	+ <u>validation = DataSet(validation_images, validation_labels, **options)</u>
267	+ <u>test = DataSet(test_images, test_labels, **options)</u>
268	+
269	+
270	+
275	271 return base.Datasets(train=train, validation=validation, test=test)
276	272
277	273



facebook / react

Watch

4,471

Star

67,492

Fork

12,523

Code

Issues 598

Pull requests 125

Projects 0

Wiki

Insights

<https://github.com/facebook/react>

A declarative, efficient, and flexible JavaScript library for building user interfaces. <https://facebook.github.io/react/>

8,627 commits

39 branches

61 releases

1,009 contributors

BSD-3-Clause

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



bvaughn committed on GitHub Prevent fiber from leaking into RN stack renderer (#9775) ...

Latest commit 824d22c 33 minutes ago

.github

Update JSFiddle for issue template (#9289)

2 months ago

docs

Bind handleChange Instead of Calling (#9764)

a day ago

eslint-rules

Make nodeType into constant for readability and reuse (#9113)

a month ago

fixtures

Remove invalid css from DOMFixtures (#9600)

8 days ago

flow

Show more meaningful stack trace for ReactNative errors (#9291)

2 months ago

mocks

Update tests to use plain JavaScript classes

2 months ago

packages

16.0.0-alpha.12

23 days ago

scripts

Prevent fiber from leaking into RN stack renderer (#9775)

33 minutes ago

src

Warn it only in DEV (#9772)

3 hours ago

.babelrc

Convert current build system to Rollup and adopt flat bundles (#9327)

2 months ago

.editorconfig

Add insert_final_newline to editorconfig

2 years ago

.eslintignore

Add React benchmarking infrastructure (#9465)

16 days ago

 **gaelaron** Move How to Contribute to documentation and update it (#7817)

c82bcef on Sep 28, 2016

15 contributors



6 lines (3 sloc) | 229 Bytes

Raw

Blame

History



Contributing to React

Want to contribute to React? There are a few things you need to know.

We wrote a [contribution guide](#) to help you get started.

QUICK START

- Installation
- Hello World
- Introducing JSX
- Rendering Elements
- Components and Props
- State and Lifecycle
- Handling Events
- Conditional Rendering
- Lists and Keys
- Forms
- Lifting State Up
- Composition vs Inheritance
- Thinking In React

ADVANCED GUIDES

- JSX In Depth
- Typechecking With PropTypes
- Refs and the DOM
- Uncontrolled Components
- Optimizing Performance
- React Without ES6
- React Without JSX
- Reconciliation
- Context
- Web Components
- Higher-Order Components
- Integrating with Other Libraries

How to Contribute

React is one of Facebook's first open source projects that is both under very active development and is also being used to ship code to everybody on facebook.com. We're still working out the kinks to make contributing to this project as easy and transparent as possible, but we're not quite there yet. Hopefully this document makes the process for contributing clear and answers some questions that you may have.

Code of Conduct

Facebook has adopted a Code of Conduct that we expect project participants to adhere to. Please read [the full text](#) so that you can understand what actions will and will not be tolerated.

Open Development

All work on React happens directly on [GitHub](#). Both core team members and external contributors send pull requests which go through the same review process.

Branch Organization

We will do our best to keep the `master` branch in good shape, with tests passing at all times. But in order to move fast, we will make API changes that your application might not be compatible with. We recommend that you use [the latest stable version of React](#).

If you send a pull request, please do it against the `master` branch. We maintain stable branches for major versions separately but we don't accept pull requests to them directly. Instead, we cherry-pick non-breaking changes from master to the latest stable major version.

Semantic Versioning

React follows [semantic versioning](#). We release patch versions for bugfixes, minor versions for new features, and major versions for any breaking changes. When we make breaking changes, we also introduce deprecation warnings in a minor version so that our users learn about the upcoming changes and migrate their code in advance.

Your First Pull Request

Working on your first Pull Request? You can learn how from this free video series:

How to Contribute to an Open Source Project on GitHub

To help you get your feet wet and get you familiar with our contribution process, we have a list of [beginner friendly issues](#) that contain bugs which are fairly easy to fix. This is a great place to get started.

If you decide to fix an issue, please be sure to check the comment thread in case somebody is already working on a fix. If nobody is working on it at the moment, please leave a comment stating that you intend to work on it so other people don't accidentally duplicate your effort.

If somebody claims an issue but doesn't follow up for more than two weeks, it's fine to take over it but you should still leave a comment.

Sending a Pull Request

The core team is monitoring for pull requests. We will review your pull request and either merge it, request changes to it, or close it with an explanation. For API changes we may need to fix our internal uses at Facebook.com, which could cause some delay. We'll do our best to provide updates and feedback throughout the process.

Before submitting a pull request, please make sure the following is done:

1. Fork [the repository](#) and create your branch from `master`.
2. If you've added code that should be tested, add tests!
3. If you've changed APIs, update the documentation.
4. Ensure the test suite passes (`npm test`).
5. Make sure your code lints (`npm run lint`).
6. Format your code with [prettier](#) (`npm run prettier`).
7. Run the [Flow](#) typechecks (`npm run flow`).
8. If you added or removed any tests, run `./scripts/fiber/record-tests` before submitting the pull request, and commit the resulting changes.
9. If you haven't already, complete the CLA.

✕ Clear current search query, filters, and sorts

🔔 12 Open ✓ 176 Closed

Author ▾

Labels ▾

Projects ▾

Milestones ▾

Assignee ▾

Sort ▾

- 🔔

Throw a warning when value is provided to a checkbox input

Difficulty: beginner

#9446 opened on Apr 17 by srph

7
- 🔔

improving doc

Component: Documentation & Website

Difficulty: beginner

#9317 opened on Apr 2 by giorgim

8
- 🔔

Make sure that jest tests fail if an error is thrown within a jsdom event handler

Difficulty: beginner

#8260 opened on Nov 11, 2016 by spicyj

7
- 🔔

[New Docs] Wanted Guides

Component: Documentation & Website

Difficulty: beginner

#8060 opened on Oct 23, 2016 by gaearon

1 of 13

49
- 🔔

how remove debug info from react.min.js file when publish project

Difficulty: beginner

#7990 opened on Oct 17, 2016 by uxitten

12
- 🔔

Should React warn when controlled <select> components have duplicate values?

Difficulty: beginner

Type: Enhancement

#6959 opened on Jun 4, 2016 by jbinto

14

Contributor License Agreement (CLA)

In order to accept your pull request, we need you to submit a CLA. You only need to do this once, so if you've done this for another Facebook open source project, you're good to go. If you are submitting a pull request for the first time, just let us know that you have completed the CLA and we can cross-check with your GitHub username.

[Complete your CLA here.](#)

Contribution Prerequisites

- You have `node` installed at v4.0.0+ and `npm` at v2.0.0+.
- You have `gcc` installed or are comfortable installing a compiler if needed. Some of our `npm` dependencies may require a compilation step. On OS X, the Xcode Command Line Tools will cover this. On Ubuntu, `apt-get install build-essential` will install the required packages. Similar commands should work on other Linux distros. Windows will require some additional steps, see the [node-gyp installation instructions](#) for details.
- You are familiar with `npm` and know whether or not you need to use `sudo` when installing packages globally.
- You are familiar with `git`.

Development Workflow

After cloning React, run `npm install` to fetch its dependencies. Then, you can run several commands:

- `npm run lint` checks the code style.
- `npm test` runs the complete test suite.
- `npm test -- --watch` runs an interactive test watcher.
- `npm test <pattern>` runs tests with matching filenames.
- `npm run flow` runs the [Flow](#) typechecks.
- `npm run build` creates a `build` folder with all the packages.

Contributing to Facebook Projects

We welcome you to follow, fork, and work on, our open source projects. If you want to contribute back to our repositories, we need to ask you to complete the Contributor License Agreement (CLA) below. If you will be contributing on behalf of your employer, you will need to complete the corporate CLA for your company.

This agreement clarifies the ability for us to incorporate your contributions. You only need to complete this form once, and you'll then be able to contribute to any of our projects.

GitHub Username *

Name *

Email *

Secondary Email

Country

Select a Country ▼

Facebook, Inc. Individual Contributor License Agreement ("Agreement"), v1.0

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to Facebook, Inc. ("Facebook"). Except for the license granted herein to Facebook and recipients of software distributed by Facebook, You reserve all right, title, and interest in and to Your Contributions.

1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with Facebook. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

Style Guide

Our linter will catch most styling issues that may exist in your code. You can check the status of your code styling by simply running `npm run lint`.

However, there are still some styles that the linter cannot pick up. If you are unsure about something, looking at [Airbnb's Style Guide](#) will guide you in the right direction.

Code Conventions

- Use semicolons `;`
- Commas last `,`
- 2 spaces for indentation (no tabs)
- Prefer `'` over `"`
- `'use strict';`
- 120 character line length (**except documentation**)
- Write "attractive" code
- Do not use the optional parameters of `setTimeout` and `setInterval`

Introductory Video

You may be interested in watching [this short video](#) (26 mins) which gives an introduction on how to contribute to React.

Meeting Notes

React team meets once a week to discuss the development of React, future plans, and priorities. You can find the meeting notes in a [dedicated repository](#).

License

By contributing to React, you agree that your contributions will be licensed under its BSD license.

What Next?

Read the [next section](#) to learn how the codebase is organized.

ReactLink still using React.PropTypes. #9755

[New issue](#)

🔔 Open

Irojas94 opened this issue 2 days ago · 1 comment



Irojas94 commented 2 days ago • edited



Do you want to request a *feature* or report a *bug*?

BUG

What is the current behavior?

React is using React.PropTypes even though it states it's going to be deprecated.

If the current behavior is a bug, please provide the steps to reproduce and if possible a minimal demo of the problem via <https://jsfiddle.net> or similar (template: <https://jsfiddle.net/84v837e9/>).
<https://github.com/facebook/react/blob/15-stable/src/addons/link/ReactLink.js#L62>

What is the expected behavior?

Should use PropTypes from the `prop-types` package.

Which versions of React, and which browser / OS are affected by this issue? Did this work in previous versions of React?

15.5.4



flarnie self-assigned this 2 days ago



flarnie added this to the **15.6** milestone 2 days ago

Assignees



flarnie

Labels

None yet

Projects

None yet

Milestone



15.6

Notifications

Subscribe

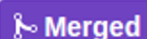
You're not receiving notifications from this thread.

2 participants



Remove stray reference to React.PropTypes in ReactLink

#9757



Merged

flarnie merged 1 commit into facebook:15.6-dev from flarnie:removeReferenceToReactPropTypesInReactLink 2 days ago



Conversation 2



Commits 1



Files changed 1

+4 -3



flarnie commented 2 days ago

Contributor

**what is the change?:**It looks like we missed updating this callsite in [12a96b9](#)**why make this change?:**

We are deprecating the `React.PropTypes` syntax and splitting that functionality into [a separate module](#).
@acdlite please correct me if there is a reason we left this here.

test plan:`yarn test`**issue:**fixes [#9755](#)

Reviewers



spicyj



gaearon



acdlite



Assignees

No one assigned

Labels

CLA Signed

Projects

None yet



Remove stray reference to React.PropTypes in ReactLink ...

✓ 959c84d

Changes from all commits 1 file +4 -3

Unified Split

Review changes

Remove stray reference to React.PropTypes in ReactLink

what is the change?

It looks like we missed updating this callsite in [12a96b9](#)

why make this change?

We are deprecating the `React.PropTypes` syntax and splitting that functionality into [a separate module](12a96b9). @acdliite please correct me if there is a reason we left this here.`

test plan:

``yarn test``

issue:

[#9755](#)

 flarnie committed 2 days ago

commit 959c84df0a7a5b34847b786a81cd9e7eb5661221

7 src/addons/link/ReactLink.js

View

```

34 34  * consumption of ReactLink easier; see LinkValueUtils and LinkStateMixin.
35 35  */
36 36
37 +var PropTypes = require('prop-types');
37 38  var React = require('React');
38 39
39 40  /**
37 38  -  ? React.PropTypes.any.isRequired
37 38  +  ? PropTypes.any.isRequired
63 64  : linkType.isRequired,
64 65  -  requestChange: React.PropTypes.func.isRequired,
64 65  +  requestChange: PropTypes.func.isRequired,
65 66  };
66 67  -  return React.PropTypes.shape(shapes);
66 67  +  return PropTypes.shape(shapes);
67 68  }
68 69
69 70  ReactLink.PropTypes = {

```

elastic / elasticsearch

Watch

1,954

★ Star

22,858

Fork

8,029

<> Code

Issues 1,186

Pull requests 159

Projects 1

Insights

Open Source, Distributed, RESTful Search Engine <https://www.elastic.co/products/elast...>

elasticsearch

java

search-engine

27,618 commits

97 branches

178 releases

837 contributors

Apache-2.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

talevy committed on **GitHub** add docs example for Ingest scripts manipulating document metadata (#... [...](#)) Latest commit dfe2eca 3 hours ago

 .github	Rewrote the github issue template to be shorter and more likely to be...	21 days ago
 benchmarks	Limit AllocationService dependency injection hack (#24479)	20 days ago
 buildSrc	Add magic \$_path stash key to docs tests (#24724)	2 days ago
 client	Deleting a document from a non-existing index creates the should not ...	3 days ago
 core	Fix error message if an incompatible node connects (#24884)	5 hours ago
 dev-tools	Remove Deprecated Script Settings (#24756)	7 days ago
 distribution	Add BWC packaging distributions	a day ago
 docs	add docs example for Ingest scripts manipulating document metadata (#...	3 hours ago
 modules	Scripting: Make ScriptEngine.compile generic on the script context (#...	15 hours ago
 plugins	Scripting: Make ScriptEngine.compile generic on the script context (#...	15 hours ago
 qa	Scripting: Simplify ScriptContext (#24818)	3 days ago

Branch: master

elasticsearch / CONTRIBUTING.md

Find file

Copy path



imotov Document work-around for jar hell in idea_rt.jar file (#24523)

7e437f2 17 days ago

20 contributors



175 lines (123 sloc) 9.96 KB

Raw

Blame

History



Contributing to elasticsearch

Elasticsearch is an open source project and we love to receive contributions from our community — you! There are many ways to contribute, from writing tutorials or blog posts, improving the documentation, submitting bug reports and feature requests or writing code which can be incorporated into Elasticsearch itself.

Bug reports

If you think you have found a bug in Elasticsearch, first make sure that you are testing against the [latest version of Elasticsearch](#) - your issue may already have been fixed. If not, search our [issues list](#) on GitHub in case a similar issue has already been opened.

It is very helpful if you can prepare a reproduction of the bug. In other words, provide a small test case which we can run to confirm your bug. It makes it easier to find the problem and to fix it. Test cases should be provided as `curl` commands which we can copy and paste into a terminal to run it locally, for example:

```
# delete the index
curl -XDELETE localhost:9200/test

# insert a document
curl -XPUT localhost:9200/test/test/1 -d '{
  "title": "test document"
}'

# this should return XXXX but instead returns YYY
curl ....
```

Contributing code and documentation changes

If you have a bugfix or new feature that you would like to contribute to Elasticsearch, please find or open an issue about it first. Talk about what you would like to do. It may be that somebody is already working on it, or that there are particular issues that you should know about before implementing the change.

We enjoy working with contributors to get their code accepted. There are many approaches to fixing a problem and it is important to find the best approach before writing too much code.

The process for contributing to any of the [Elastic repositories](#) is similar. Details for individual projects can be found below.

Fork and clone the repository

You will need to fork the main Elasticsearch code or documentation repository and clone it to your local machine. See [github help page](#) for help.

Further instructions for specific projects are given below.

Submitting your changes

Once your changes and tests are ready to submit for review:

1. Test your changes

Run the test suite to make sure that nothing is broken. See the [TESTING](#) file for help running tests.

2. Sign the Contributor License Agreement

Please make sure you have signed our [Contributor License Agreement](#). We are not asking you to assign copyright to us, but to give us the right to distribute your code without restriction. We ask this of all contributors in order to assure our users of the origin and continuing existence of the code. You only need to sign the CLA once.

3. Rebase your changes

Update your local repository with the most recent code from the main Elasticsearch repository, and rebase your branch on top of the latest master branch. We prefer your initial changes to be squashed into a single commit. Later, if we ask you to make changes, add them as separate commits. This makes them easier to review. As a final step before merging we will either ask you to squash all commits yourself or we'll do it for you.

4. Submit a pull request

Push your local changes to your forked copy of the repository and [submit a pull request](#). In the pull request, choose a title which sums up the changes that you have made, and in the body provide more details about what your changes do. Also mention the number of the issue where discussion has taken place, eg "Closes #123".

At Elastic, we are passionate about our products, its users, and the community. You are important to us, so we feel it is only right to ensure we have the agreement of our contributors to use their work, whether it be code or documentation.

Therefore, we are asking all contributors to sign a contributor license agreement (CLA). CLAs are generally common and accepted in most Open Source software projects. We all wish for Elasticsearch to be used and distributed as widely as possible, and for its users to be confident about the origins and continuing existence of the code. The CLA help us achieve that goal.

The CLA is a license agreement, which enables Elastic to distribute your code without restriction. It doesn't require you to assign to us any copyright you have, the ownership of which remains in full with you. You cannot withdraw permission for its use at a later date.

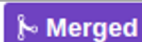
The next step is for you to select the appropriate agreement below, read the terms, and once comfortable, complete the electronic signature. After this, you will receive a copy of the agreement by email, and we can start to use your valued contribution within the different Elasticsearch projects.

Please select the appropriate description that reflects why you are signing up:

- ☐ **An individual contributor**
- ☐ **An individual contributing under an existing company contributor agreement**
- ☐ **A company wanting its employees to contribute**

[Sign Agreement](#)

Remove AlreadyExpiredException #24857



ywelsch merged 1 commit into elastic:master from ywelsch:enhance/remove-alreadyexpiredexception a day ago

Conversation 1

Commits 1

Files changed 5

+35 -194



ywelsch commented a day ago • edited

Contributor +

This is a relict from the TTL functionality that has been removed.

Relates to #21670

Remove AlreadyExpiredException ✓ 9087ad5

ywelsch added :Internal non-issue v6.0.0 labels a day ago

ywelsch requested a review from s1monw a day ago



bleskes approved these changes a day ago

View changes

ywelsch merged commit b5adb3c into elastic:master a day ago
2 checks passed

View details

Reviewers

bleskes ✓

s1monw ●

Assignees

No one assigned

Labels

:Internal

non-issue

v6.0.0

Projects

None yet

Milestone

No milestone

Remove 2.x backward compatibility of mappings. #21670

Merged jpountz merged 1 commit into elastic:master from jpountz:remove_mapping_bwc on Nov 30, 2016

Conversation 5

Commits 1

Files changed 279

+466 -24,538



jpountz commented on Nov 19, 2016

Contributor +

For the record, I also had to remove the geo-hash cell and geo-distance range queries to make the code compile. These queries already throw an exception in all cases with 5.x indices, so that does not hurt any more.

I also had to rename all 2.x bwc indices from `index-${version}` to `unsupported-${version}` to make `oldIndexBackwardCompatibilityIT` happy.



jpountz added **:Mapping** **non-issue** **v6.0.0** labels on Nov 19, 2016



jpountz commented on Nov 23, 2016

Contributor +

@elasticmachine retest this please

Reviewers

No reviews

Assignees

s1monw

Labels

:Mapping

non-issue

v6.0.0-alpha1

Projects

None yet

Changes from all commits 5 files +35 -194

Unified

Split

Review changes

3 core/src/main/java/org/elasticsearch/ElasticsearchException.java

View

```

@@ -877,8 +877,7 @@ public String toString() {
    877      org.elasticsearch.transport.ReceiveTimeoutTransportException::new, 83, UNKNOWN_VERSION_ADDED),
    878      NODE_DISCONNECTED_EXCEPTION(org.elasticsearch.transport.NodeDisconnectedException.class,
    879      org.elasticsearch.transport.NodeDisconnectedException::new, 84, UNKNOWN_VERSION_ADDED),
    880      - ALREADY_EXPIRED_EXCEPTION(org.elasticsearch.index.AlreadyExpiredException.class,
    881      - org.elasticsearch.index.AlreadyExpiredException::new, 85, UNKNOWN_VERSION_ADDED),
    880      + // 85 used to be for AlreadyExpiredException
    882      AGGREGATION_EXECUTION_EXCEPTION(org.elasticsearch.search.aggregations.AggregationExecutionException.class,
    883      org.elasticsearch.search.aggregations.AggregationExecutionException::new, 86, UNKNOWN_VERSION_ADDED),
    884      // 87 used to be for MergeMappingException

```

91 core/src/main/java/org/elasticsearch/index/AlreadyExpiredException.java

View

```

... -1,91 +0,0 @@
1  -/*
2   - * Licensed to Elasticsearch under one or more contributor
3   - * license agreements. See the NOTICE file distributed with
4   - * this work for additional information regarding copyright
5   - * ownership. Elasticsearch licenses this file to you under
6   - * the Apache License, Version 2.0 (the "License"); you may
7   - * not use this file except in compliance with the License.
8   - * You may obtain a copy of the License at
9   - *
10  - *   http://www.apache.org/licenses/LICENSE-2.0
11  - *
12  - * Unless required by applicable law or agreed to in writing,
13  - * software distributed under the License is distributed on an
14  - * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
15  - * KIND, either express or implied. See the License for the
16  - * specific language governing permissions and limitations
17  - * under the License.
18  - */
19  -
20  -package org.elasticsearch.index:

```

nodejs / node

Watch

2,205

★ Unstar

35,245

Fork

6,772

Code

Issues 737

Pull requests 309

Projects 5

Wiki

Insights

Node.js JavaScript runtime <https://nodejs.org>

nodejs

javascript

node

js

runtime

mit

linux

macos

windows

17,578 commits

126 branches

426 releases

1,363 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



TimothyGu url: expose WHATWG url.origin as ASCII ...

Latest commit 413691f 5 days ago

.github

doc: fix commit guideline url

20 days ago

benchmark

stream: improve readable push performance

2 days ago

deps

deps: backport 6d38f89d from upstream V8

2 hours ago

doc

src: allow --tls-cipher-list in NODE_OPTIONS

2 hours ago

lib

url: expose WHATWG url.origin as ASCII

24 minutes ago

src

src: add linux getauxval(AT_SECURE) in SafeGetenv

an hour ago

test

url: expose WHATWG url.origin as ASCII

24 minutes ago

tools

tools: update test-npm.sh paths

2 days ago

.editorconfig

tools: disallow trailing whitespace for markdown

6 months ago

.eslintignore

doc: add eslint-plugin-markdown

a month ago

.eslintrc.yaml

tools: relax lint rule for regexps

20 days ago

.gitattributes

src: limit .gitattributes eol to vcbuild.bat

2 years ago

.gitignore

tools: ignore node_trace.*.log

24 days ago

 **kysnm** doc: Change options at STEP 5 in CONTRIBUTING.md

d7d16f7 14 days ago

45 contributors



346 lines (250 sloc) | 11.3 KB

Raw

Blame

History



Contributing to Node.js

Code of Conduct

Please read the [Code of Conduct](#) which explains the minimum behavior expectations for Node.js contributors.

Issue Contributions

When opening issues or commenting on existing issues, please make sure discussions are related to concrete technical issues with Node.js.

- For general help using Node.js, please file an issue at the [Node.js help repository](#).
- Discussion of non-technical topics (such as intellectual property and trademark) should use the [Technical Steering Committee \(TSC\) repository](#).

Code Contributions

This section will guide you through the contribution process.

Step 1: Fork

Fork the project [on GitHub](#) and clone your fork locally.

```
$ git clone git@github.com:username/node.git
```

stream: remove dup property #13216

🔗 Merged calvinmetcalf merged 1 commit into nodejs:master from calvinmetcalf:stream-final-dup-prop 7 hours ago

💬 Conversation 9

🔗 Commits 1

📄 Files changed 1

+0 -3



calvinmetcalf commented 7 hours ago

Member



looks like [#12828](#) accidentally got a double property definition, as mentioned [here](#), this removes that



nodejs-github-bot added the **stream** label 7 hours ago



✅ lpinca approved these changes 7 hours ago

[View changes](#)



✅ cjhrrig approved these changes 7 hours ago

[View changes](#)



calvinmetcalf commented 7 hours ago

Member



considering how trivial this is I'll plan on landing this in the next hour or 2 unless anyone objects

👍 1

Reviewers

addaleax



lpinca



cjhrrig



Assignees

No one assigned

Labels

stream

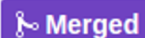
Projects

None yet

Milestone

No milestone

stream: remove dup property #13216



Merged

calvinmetcalf merged 1 commit into nodejs:master from calvinmetcalf:stream-final-dup-prop 7 hours ago



Conversation 9



Commits 1



Files changed 1

Changes from all commits ▾ 1 file ▾ +0 -3 ■■■■

Unified

Split

Review changes ▾

stream: remove dup property

PR-URL: #13216

Reviewed-By: Colin Ihrig <cjihrig@gmail.com>

Reviewed-By: Luigi Pinca <luigipinca@gmail.com>

Reviewed-By: Anna Henningsen <anna@addaleax.net>



calvinmetcalf committed 8 hours ago

Verified

commit 155687cb7ee23d7777d1072c4c2956564bffa8ac4

3 ■■■■ lib/_stream_writable.js

View



@@ -61,9 +61,6 @@ function WritableState(options, stream) {

61 61 // if _final has been called

62 62 this.finalCalled = false;

63 63

64 - // if _final has been called

65 - this.finalCalled = false;

66 -

67 64 // drain event flag.

68 65 this.needDrain = false;

69 66 // at the start of calling end()



spring-projects / spring-boot



Watch

1,574



Star

12,841



Fork

11,529

Code

Issues 523

Pull requests 50

Wiki

Insights

Spring Boot <http://projects.spring.io/spring-boot>

12,161 commits

12 branches

76 releases

344 contributors

Apache-2.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



mbhave Replace usages of EnvironmentTestUtils

Latest commit d745b69 2 days ago

.github	Merge branch '1.3.x'	10 months ago
.mvn	Upgrade to Apache Maven 3.5.0	a month ago
eclipse	Introduce spring-boot-autoconfigure-processor	4 months ago
spring-boot-actuator-docs	Polish "Use try-with-resources to close resources automatically"	2 days ago
spring-boot-actuator	Replace usages of EnvironmentTestUtils	20 hours ago
spring-boot-autoconfigure	Replace usages of EnvironmentTestUtils	20 hours ago
spring-boot-cli	Polish "Use try-with-resources to close resources automatically"	2 days ago
spring-boot-dependencies	Upgrade to Sqlite Jdbc 3.18.0	a day ago
spring-boot-deployment-tests	Revert "Work around https://issues.jboss.org/browse/WFLY-8515"	2 months ago

snicoll Document `fast` profile in contributing documentation

1d9520b on Mar 3

12 contributors



Executable File | 229 lines (161 sloc) | 9.2 KB

Raw

Blame

History



Contributing to Spring Boot

Spring Boot is released under the Apache 2.0 license. If you would like to contribute something, or simply want to hack on the code this document should help you get started.

Code of Conduct

This project adheres to the Contributor Covenant [code of conduct](#). By participating, you are expected to uphold this code. Please report unacceptable behavior to spring-code-of-conduct@pivotal.io.

Using GitHub issues

We use GitHub issues to track bugs and enhancements. If you have a general usage question please ask on [Stack Overflow](#). The Spring Boot team and the broader community monitor the [spring-boot](#) tag.

If you are reporting a bug, please help to speed up problem diagnosis by providing as much information as possible. Ideally, that would include a small [sample project](#) that reproduces the problem.

Sign the Contributor License Agreement

Before we accept a non-trivial patch or pull request we will need you to [sign the Contributor License Agreement](#). Signing the contributor's agreement does not grant anyone commit rights to the main repository, but it does mean that we can accept your contributions, and you will get an author credit if we do. Active contributors might be asked to join the core team, and given the ability to merge pull requests.



Authorize Pivotal CLA



Pivotal CLA by [pivotal-issuemaster](#)
wants to access your **Taeung** account



Personal user data
Email addresses (read-only)



Authorizing will redirect to
<https://cla.pivotal.io>

[Authorize pivotal-issuemaster](#)

Not owned or
operated by GitHub

Created 2 years ago

More than 1K
GitHub users

[Learn more about OAuth](#)

Code Conventions and Housekeeping

None of these is essential for a pull request, but they will all help. They can also be added after the original pull request but before a merge.

- Use the Spring Framework code format conventions. If you use Eclipse and you follow the 'Importing into eclipse' instructions below you should get project specific formatting automatically. You can also import formatter settings using the `eclipse-code-formatter.xml` file from the `eclipse` folder. If using IntelliJ IDEA, you can use the [Eclipse Code Formatter Plugin](#) to import the same file.
- Make sure all new `.java` files to have a simple Javadoc class comment with at least an `@author` tag identifying you, and preferably at least a paragraph on what the class is for.
- Add the ASF license header comment to all new `.java` files (copy from existing files in the project)
- Add yourself as an `@author` to the `.java` files that you modify substantially (more than cosmetic changes).
- Add some Javadocs.
- A few unit tests would help a lot as well — someone has to do it.
- If no-one else is using your branch, please rebase it against the current master (or other target branch in the main project).
- When writing a commit message please follow [these conventions](#), if you are fixing an existing issue please add `Fixes gh-XXXX` at the end of the commit message (where `XXXX` is the issue number).

Working with the code

If you don't have an IDE preference we would recommend that you use [Spring Tools Suite](#) or [Eclipse](#) when working with the code. We use the [M2Eclipse](#) eclipse plugin for maven support. Other IDEs and tools should also work without issue.

Building from source

To build the source you will need to install [Apache Maven](#) v3.2.3 or above and JDK 1.8.

Default build

The project can be built from the root directory using the standard maven command:

```
$ ./mvnw clean install
```

Note	You may need to increase the amount of memory available to Maven by setting a <code>MAVEN_OPTS</code> environment variable with the value <code>-Xmx512m</code>
------	---

If you are rebuilding often, you might also want to skip the tests and the execution of checkstyle until you are ready to submit a pull request:

```
$ ./mvnw clean install -DskipTests -Pfast
```

Full Build

Multi-module Maven builds cannot directly include maven plugins that are part of the reactor unless they have previously been built. Unfortunately this restriction causes some compilations for Spring Boot as we include a maven plugin and use it within the samples. The standard build works around this restriction by launching the samples via the `maven-invoker-plugin` so that they are not part of the reactor. This works fine most of the time, however, sometimes it's useful to run a build that includes all modules (for example when using `maven-versions-plugin`). We use the full build on our CI servers and during the release process.

When filter init throw exception, embedded tomcat started success, but all request return 404 #9094

[New issue](#)

Closed

hengyunabc opened this issue 21 days ago · 1 comment



hengyunabc commented 21 days ago

Contributor



version: 1.4.x

Add a Filter:

```
@Bean
FilterRegistrationBean testFilter() {
    FilterRegistrationBean filterRegistrationBean = new FilterRegistrationBean(
        filterRegistrationBean.setFilter(new Filter() {
            @Override
            public void init(FilterConfig filterConfig) throws ServletException
                throw new ServletException();
        })

        @Override
        public void doFilter(ServletRequest request, ServletResponse response
            throws IOException, ServletException {

        @Override
        public void destroy() {

    });
    filterRegistrationBean.setUrlPatterns(Arrays.asList("/test"));
    return filterRegistrationBean;
}
```

Assignees

No one assigned

Labels

status: duplicate

Projects

None yet

Milestone

No milestone

Notifications

Subscribe

You're not receiving notifications from this thread.

3 participants



Tomcat print error log:

```
~[classes/:na]
j.initFilter(ApplicationFilterConfig.java:279) ~[tomcat-embed-core-8.5.6.jar:8.5.6]
j.<init>(ApplicationFilterConfig.java:109) ~[tomcat-embed-core-8.5.6.jar:8.5.6]
start(StandardContext.java:4572) [tomcat-embed-core-8.5.6.jar:8.5.6]
internal(StandardContext.java:5215) [tomcat-embed-core-8.5.6.jar:8.5.6]
recycleBase.java:150) [tomcat-embed-core-8.5.6.jar:8.5.6]
d.call(ContainerBase.java:1403) [tomcat-embed-core-8.5.6.jar:8.5.6]
d.call(ContainerBase.java:1393) [tomcat-embed-core-8.5.6.jar:8.5.6]
ava:266) [na:1.8.0_112]
er(ThreadPoolExecutor.java:1142) [na:1.8.0_112]
un(ThreadPoolExecutor.java:617) [na:1.8.0_112]
_112]

o.apache.catalina.core.StandardContext : One or more Filters failed to start. Full details v
o.apache.catalina.core.StandardContext : Context [] startup failed due to previous errors
```

But spring boot started success:

```
20:29:24.521 INFO 27483 --- [          main] s.b.c.e.t.TomcatEmbeddedServletContainer : To
20:29:24.531 INFO 27483 --- [          main] com.test.Application : Started Applic
```

All http request will return 404.

 **spring-issuemaster** added the **status: waiting-for-triage** label 21 days ago

 **hengyunabc** added a commit to hengyunabc/spring-boot that referenced this issue 21 days ago

 **fix #9094** throw exception, when tomcat context state not 0e0e56a
STARTED.



philwebb commented 8 days ago

Owner



Closing in favor of PR [#9095](#)

fix #9094 throw exception, when tomcat context state not STARTED. #9095

Closedhengyunabc wants to merge 1 commit into `spring-projects:1.4.x` from `hengyunabc:1.4.x`

💬 Conversation 0

🔗 Commits 1

📄 Files changed 2

+40 -0



hengyunabc commented 21 days ago

Contributor +😊

No description provided.

🔗 fix #9094 throw exception, when tomcat context state not STARTED.

✓ 0e0e56a

🏷️ spring-issuemaster added the `status: waiting-for-triage` label 21 days ago

📖 This was referenced 10 days ago

Undertow initialises filters lazily so failures are only discovered when the first request is made #9232**Closed****When filter init throw exception, embedded tomcat started success, but all request return 404 #9094****Closed**🏷️ philwebb added `priority: normal` `type: bug` and removed `status: waiting-for-triage` labels 8 days ago

Reviewers

No reviews

Assignees

👤 wilkinsona

Labels

`priority: normal``type: bug`

Projects

None yet

Milestone

1.5.4

Conversation 0

Commits 1

Files changed 2

Changes from all commits 2 files +40 -0

Unified

Split

Review changes

8 ...java/org/springframework/boot/context embedded/tomcat/TomcatEmbeddedServletContainer.java

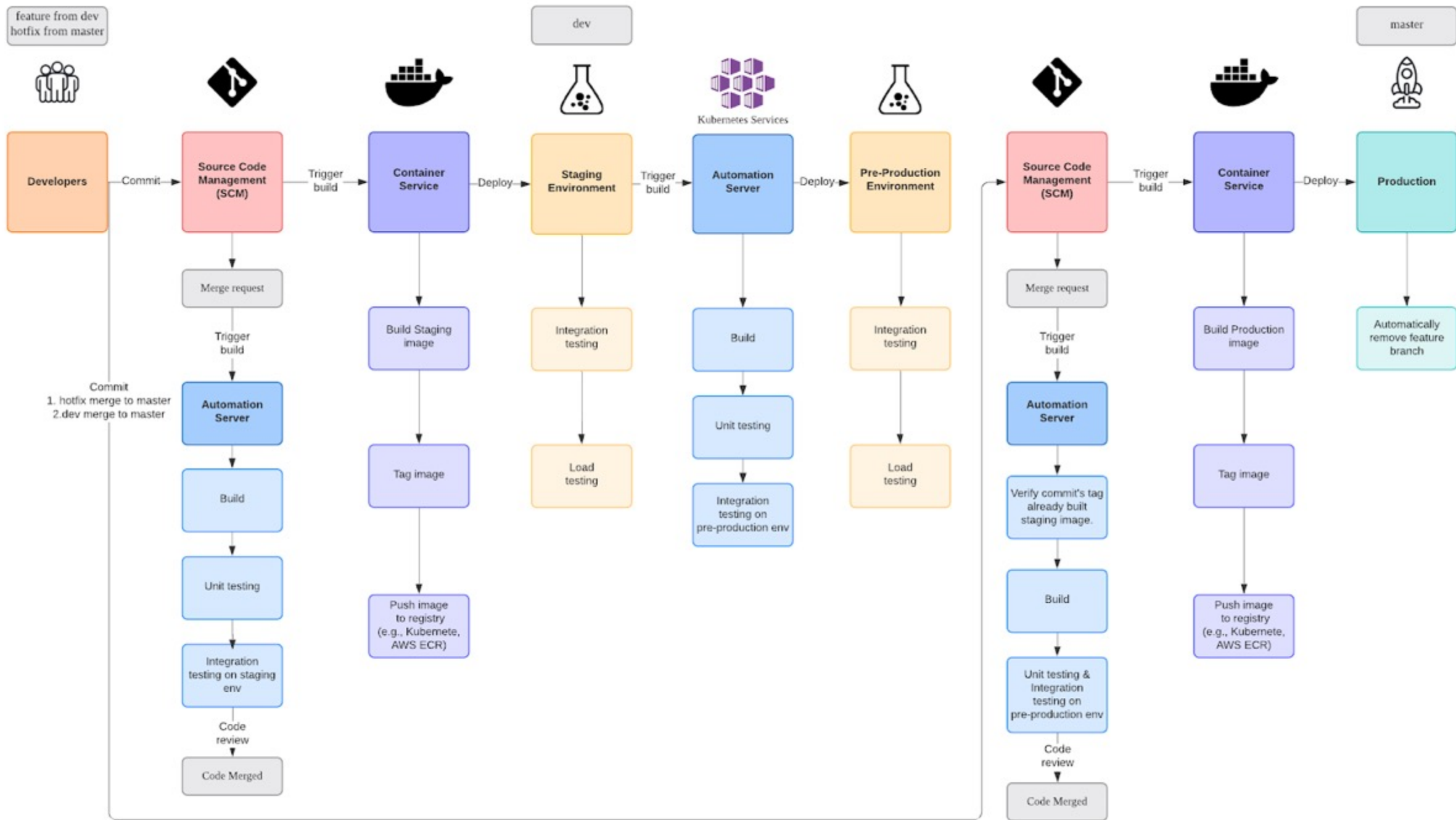
View

```
@@ -192,6 +192,7 @@ public void start() throws EmbeddedServletContainerException {
192 192         if (connector != null && this.autoStart) {
193 193             startConnector(connector);
194 194         }
195 +         checkThatContextHaveStarted();
195 196         checkThatConnectorsHaveStarted();
196 197         this.started = true;
197 198         TomcatEmbeddedServletContainer.logger

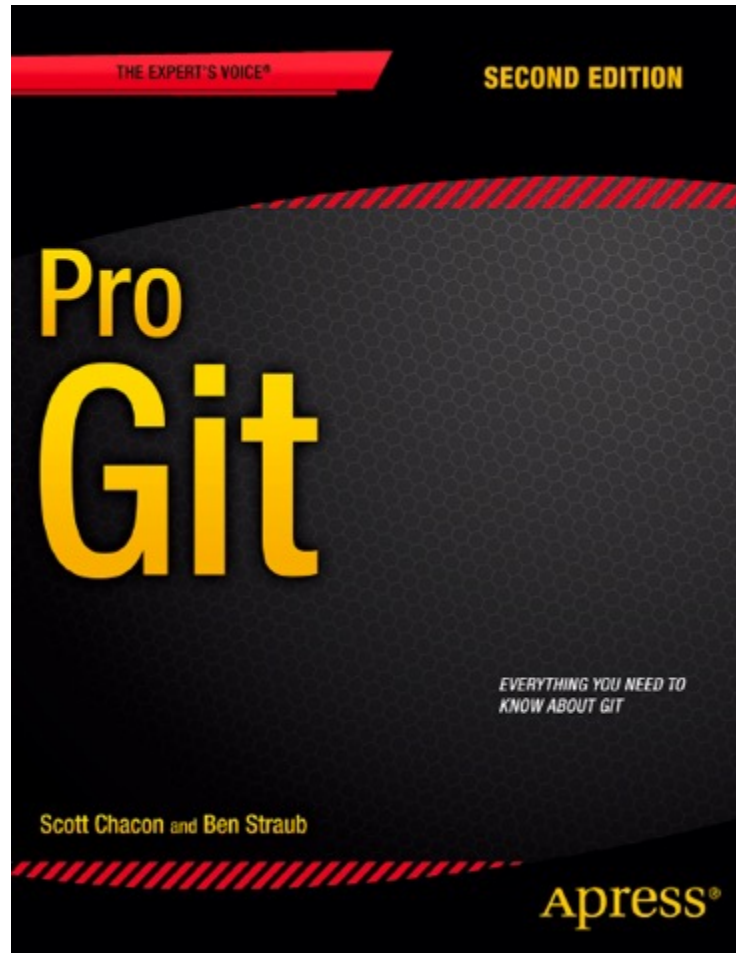
@@ -221,6 +222,13 @@ private void checkThatConnectorsHaveStarted() {
221 222     }
222 223 }
223 224

225 + private void checkThatContextHaveStarted() {
226 +     LifecycleState state = this.findContext().getState();
227 +     if (!LifecycleState.STARTED.equals(state)) {
228 +         throw new EmbeddedServletContainerException("Context state expect STARTED, but " + state, null);
229 +     }
230 + }
231 +

224 232 private void stopSilently() {
225 233     try {
226 234         stopTomcat();
```

Pro Git



<https://git-scm.com/book/ko/v2>

<https://opentutorials.org/module/2676>

지옥에서 온 Git

지옥에서 온 Git

공동공부 (739명)

커버 페이지

토픽 목록

버전관리의 본질

- 설치 및 실습방법

- 저장소 만들기

- git에 관리할 대상으로 파일 등록

- 버전 만들기 (commit)

- Stage area

- 변경사항 확인하기

- 과거의 버전으로 돌아가기

- 명령의 빈도와 메뉴얼 보는 방법

- 수련해봅시다.

git의 원리

- gistory 설치

- git add의 원리

- objects 파일명의 원리

- commit의 원리

- status의 원리

git의 혁신 - branch

- branch 만들기

- branch 관리하기

이 수업은 GiTn 시리즈로 완전히 대체 되었습니다. GiTn은 보다 많은 내용을 적은 단위로 쪼개서 선택적으로 공부하실 수 있도록 제작된 수업입니다. 아래 주소를 통해서 GiTn 을 접할 수 있습니다.

GiTn의 입구수업 : <https://opentutorials.org/module/3733>

지식지도 : <https://seomal.org/?i=GIT1>

수업소개

이 수업은 Git의 초심자에게는 기본적인 사용법을 중급자는 Git이 동작하는 원리를 소개해드리기 위한 수업입니다.

이 수업에서는 명령어를 통해서 Git을 다루는 방법을 소개합니다. 명령어 기반 Git은 많은 시스템에 기본적으로 설치되어 있기 때문에 명령어로 Git을 다룰 수 있게 되면 많은 곳에서 특별한 설치 없이 Git을 사용할 수 있습니다.

명령어로 Git을 다루는 것이 어렵게 느껴지신다면 GUI로 Git을 다루는 수업을 추천드립니다.

<https://opentutorials.org/course/1492>



지옥에서 온 Git - 수업소개



마지막 시청



공유