

{POWER.CODERS}

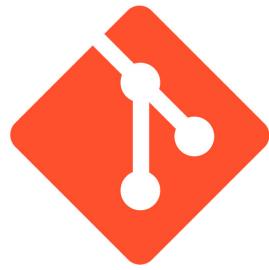
Git Training

Tuesday 25th of June 2019

jeremy.gobet.72@gmail.com

Agenda

1. Discover Git
2. Basic commands
3. Understand Git basics
4. Working remotely
5. Branches & workflows
6. Rewrite history
7. Other commands
8. Advanced notions
9. Conclusion



git

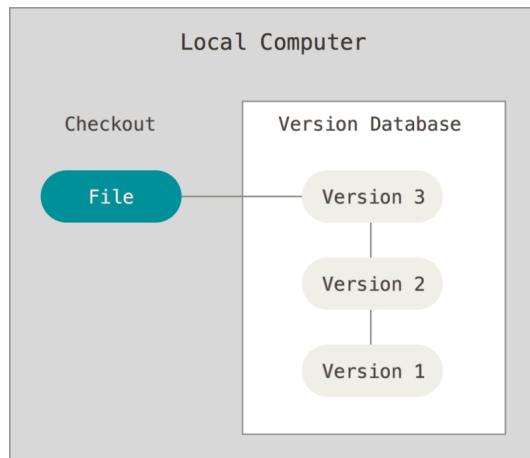
Discover Git

Connect

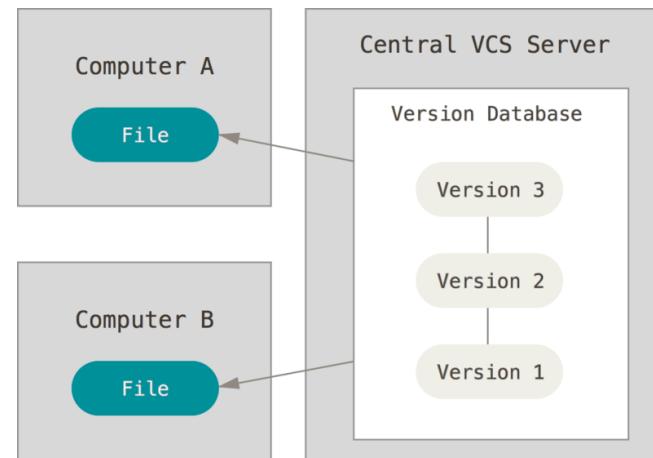
- ▶ Are you using source control ?
- ▶ What great about it ?
- ▶ What not so nice ?
- ▶ What missing and why would I want that ?

Version Control

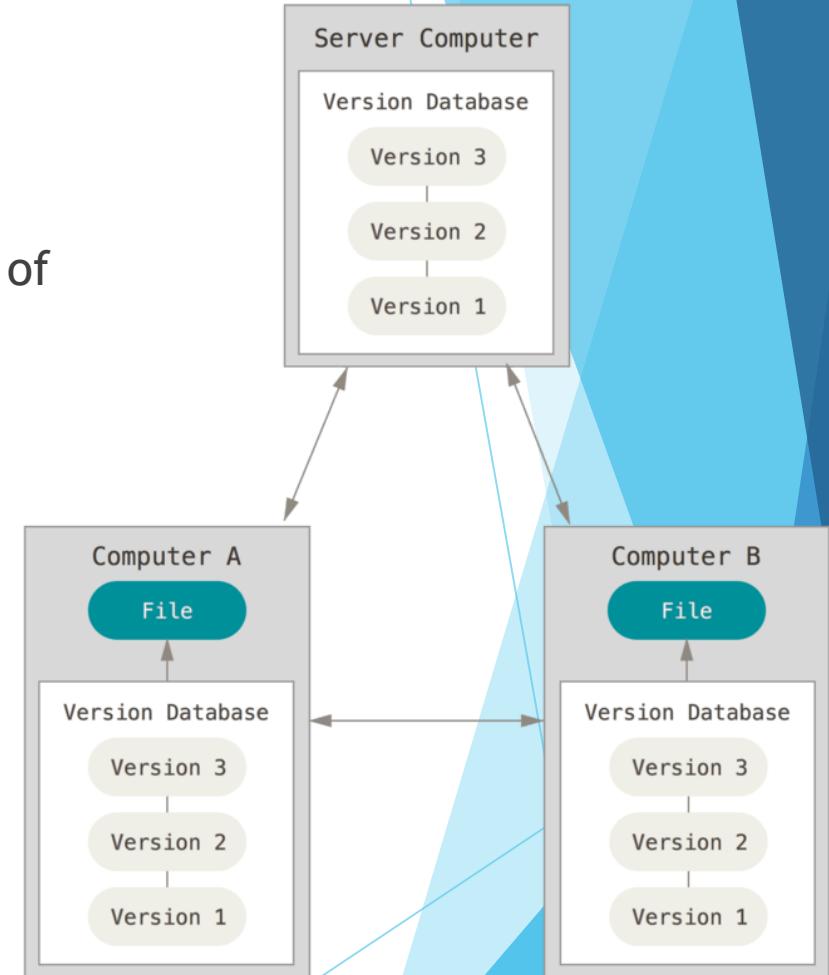
- ▶ Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later



Local Version Control



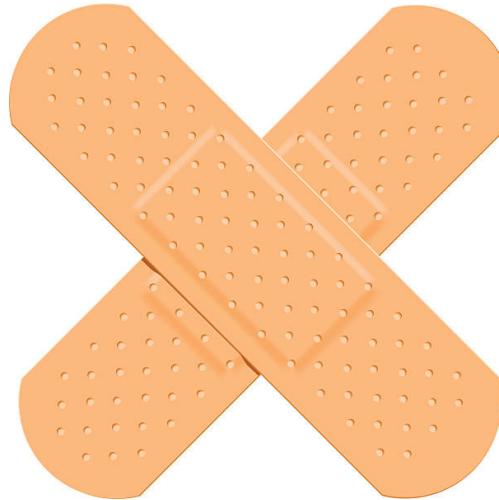
Centralized Version Control



Distributed Version Control

What are the pains for developers about source code ?

- ▶ Do modifications in **isolation**
 - ▶ Do not be impact by changes of my colleagues
- ▶ Have several **versions** of the software
 - ▶ The current development version
 - ▶ To do a patch on an old version
- ▶ **Merge** my work with others
- ▶ Have **backup** of the source code
- ▶ Have an **history** of changes
- ▶ View the **origin** of modifications in a file
- ▶ Have **consistency**
 - ▶ Most of changes concern several files
- ▶ **Compare** versions



Git & Ecosystem

- ▶ Git is the heart, but there are also fundamental tools/sites around
 - ▶ They offer free or paid remote repository
 - ▶ 'Social network' feature about coding
 - ▶ Tasks repository
 - ▶ Collaborative feature : Merge (or Pull request)



<https://github.com/>



<https://gitlab.com/>



<https://bitbucket.org/>

Basic commands

Git config

Setup our local environment to annotate the commits

```
git config --global user.name "Jeremy Gobet"
```

```
git config --global user.email jeremy@gobet.ch
```

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

Show local config

```
git config --list
```

Show and edit config with your file editor

```
git config --global --edit
```

Git help

```
git help
```

Help about a specific command

```
git help <command>
```

```
git help add
```

```
git help commit
```

```
git help status
```

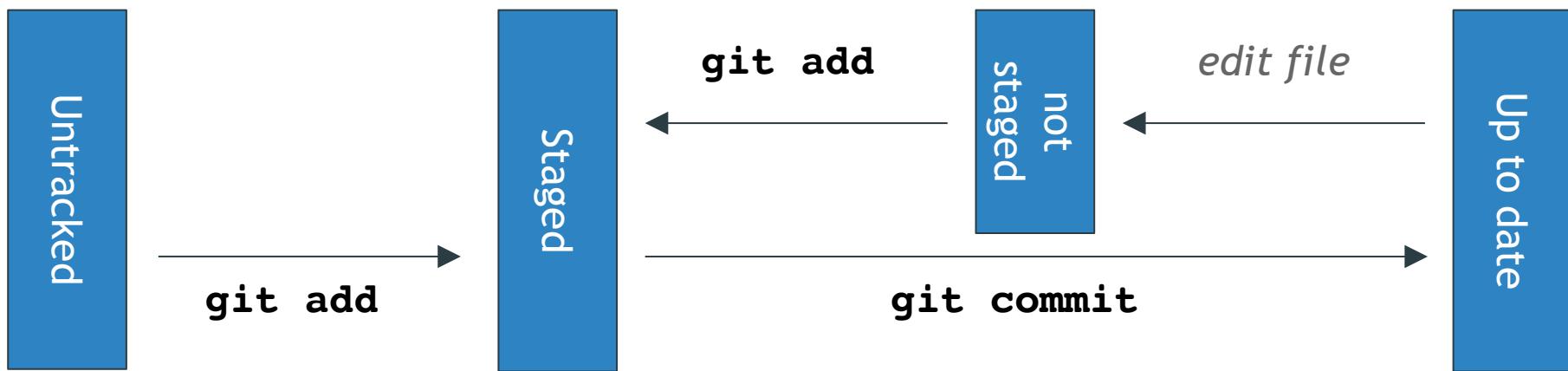
```
git help push
```

```
git help pull
```

Understand Git basics

Four states

1. Untracked
2. Staged
3. Up-to-date
4. Not staged



Create local repository and add file

Create your Git repository

```
mkdir myrepository
```

```
cd myrepository
```

```
git init
```

Create and track a file

```
touche readme.txt
```

```
git status
```

```
git add readme.txt
```

```
git status
```

Git commit

Add not staged files and commit with message

```
git add .
```

```
git commit -m "message"
```

or

```
git commit -am "message"
```

Add staged files to previous commit

```
git commit --amend
```

Edit file

```
vim readme.txt  
git diff  
git status  
git commit -am "edited readme file"  
git status  
git log
```

Vim cheatsheet

mode insert

- i

Exit mode

- ESC

Cancel

- u

Quit

- :q

Save & Quit

- :wq

Force Quit

- :q!

Cancel last changes

```
vim readme.txt  
git status  
git checkout readme.txt  
git status
```

Vim cheatsheet

mode insert

- i

Exit mode

- ESC

Cancel

- u

Quit

- :q

Save & Quit

- :wq

Force Quit

- :q!

Cancel & revert commits

Cancel last commit

```
git reset --soft HEAD^
```

Cancel last commit and unstage changes

```
git reset --mixed HEAD^
```

```
git reset HEAD^
```

Cancel last commit and delete changes

```
git reset --hard HEAD^
```

Commit selection cheatsheet

last commit

- HEAD^
- HEAD~1

2 last commits

- HEAD^^
- HEAD~2

3 last commits

- HEAD~3

Commit number (last 6 characters)

- 4f56b2

Remove file from repository

```
git rm readme.txt
```

```
git status
```

```
git commit -am "removed readme file"
```

```
git status
```

```
git log
```

Ignore files and folders

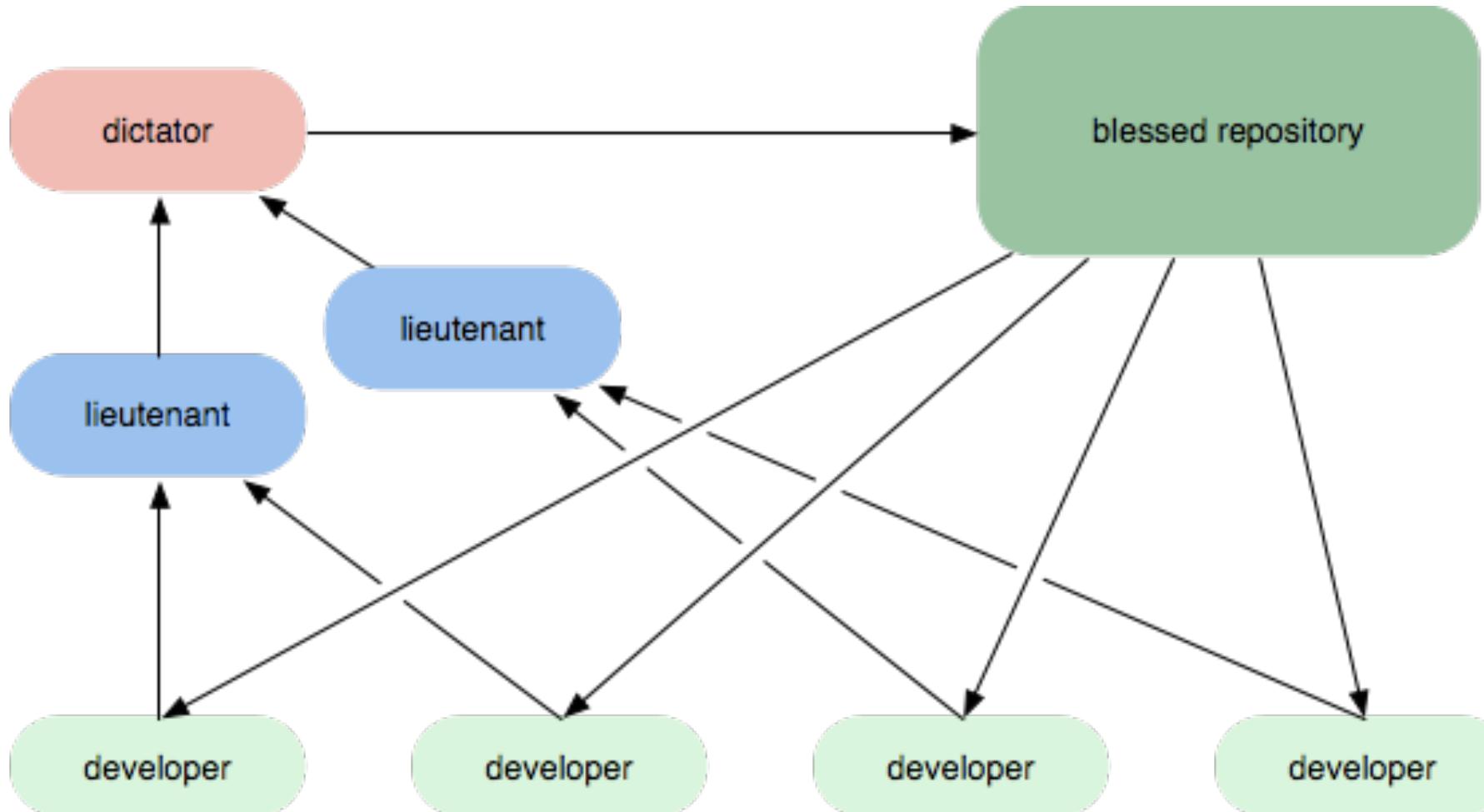
.gitignore

- ▶ Syntax in .gitignore file

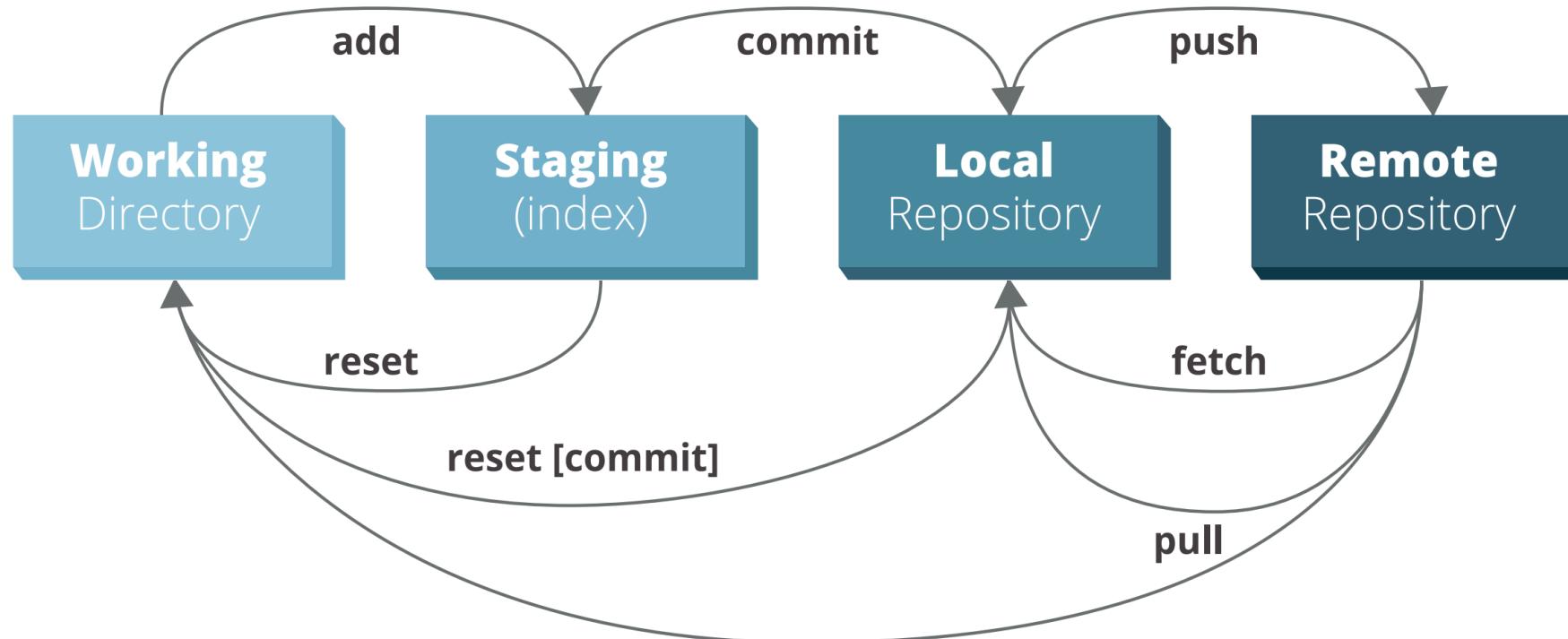
- ▶ `**.txt` Ignore all files with txt extension
- ▶ `*.txt` Ignore all files with txt extension in root folder
- ▶ `/src/data` Ignore all files and folder in folder: /src/data
- ▶ `/app/*.txt` Ignore all files with txt extension in folder: /app
- ▶ `app` Ignore all files and folder named app
- ▶ `!/target/.txt` Ignore all everything except files with txt extension

Working remotely

A decentralized model from Linux



Git workflow



Git pull

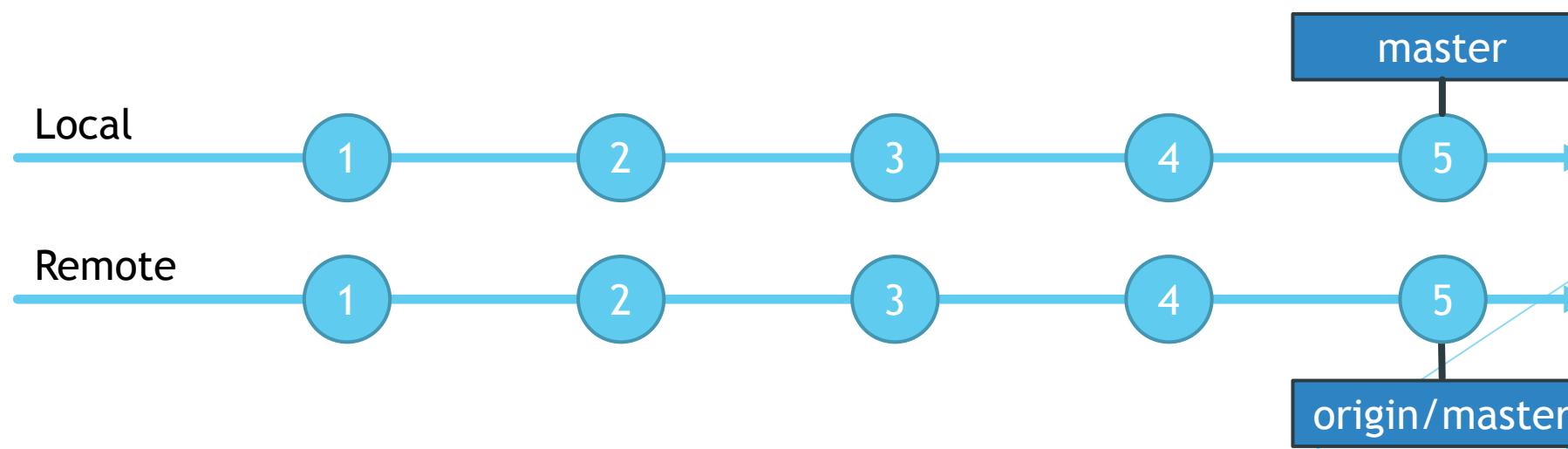
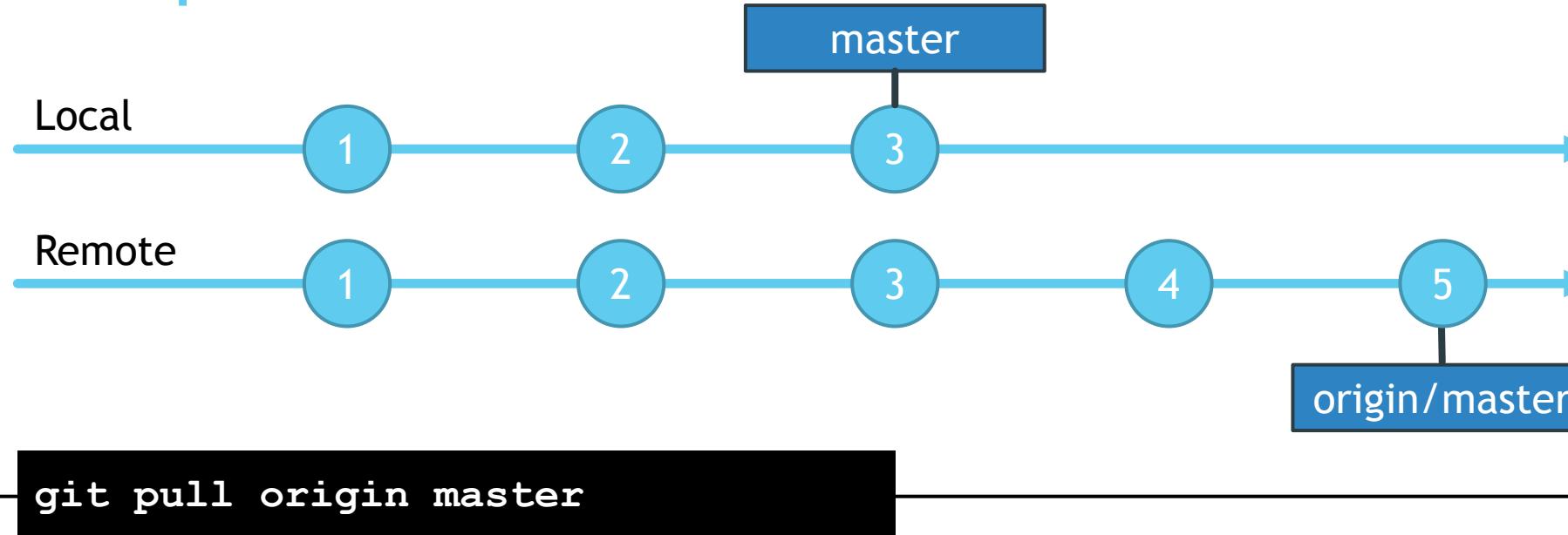
Pull changes from remote repository

```
git pull
```

```
git pull [<remote> <branch>]
```

```
git pull origin master
```

Git pull



Git push

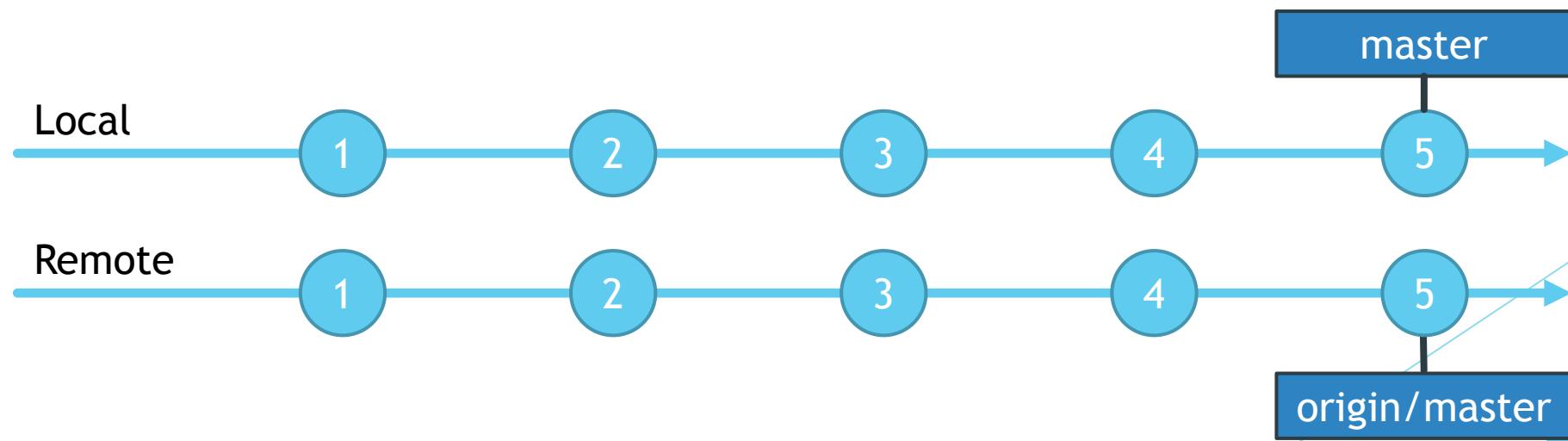
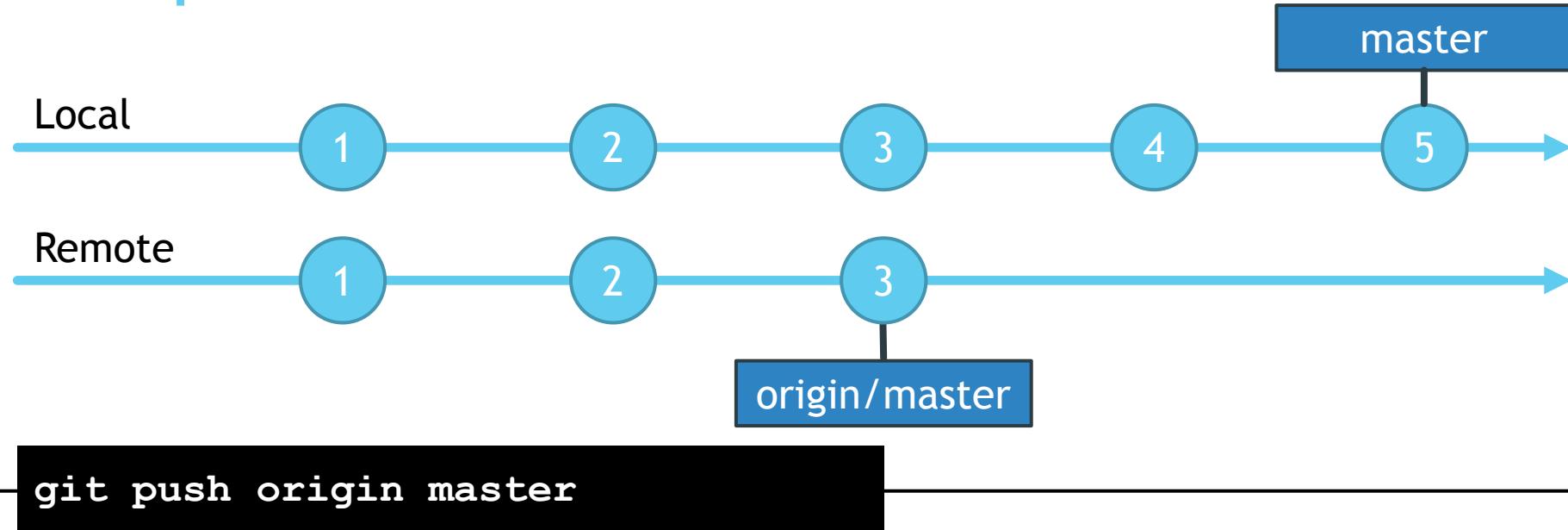
Push changes to remote repository

git push

git push [<remote> <branch>]

git push origin master

Git push



Git fetch

Fetch changes from remote repository

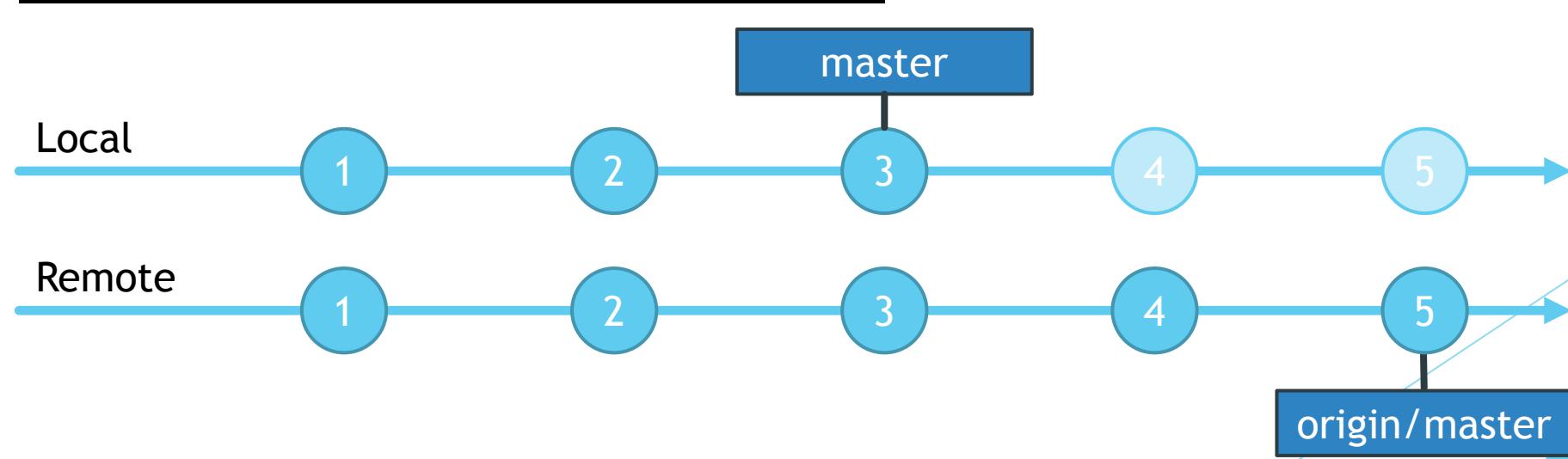
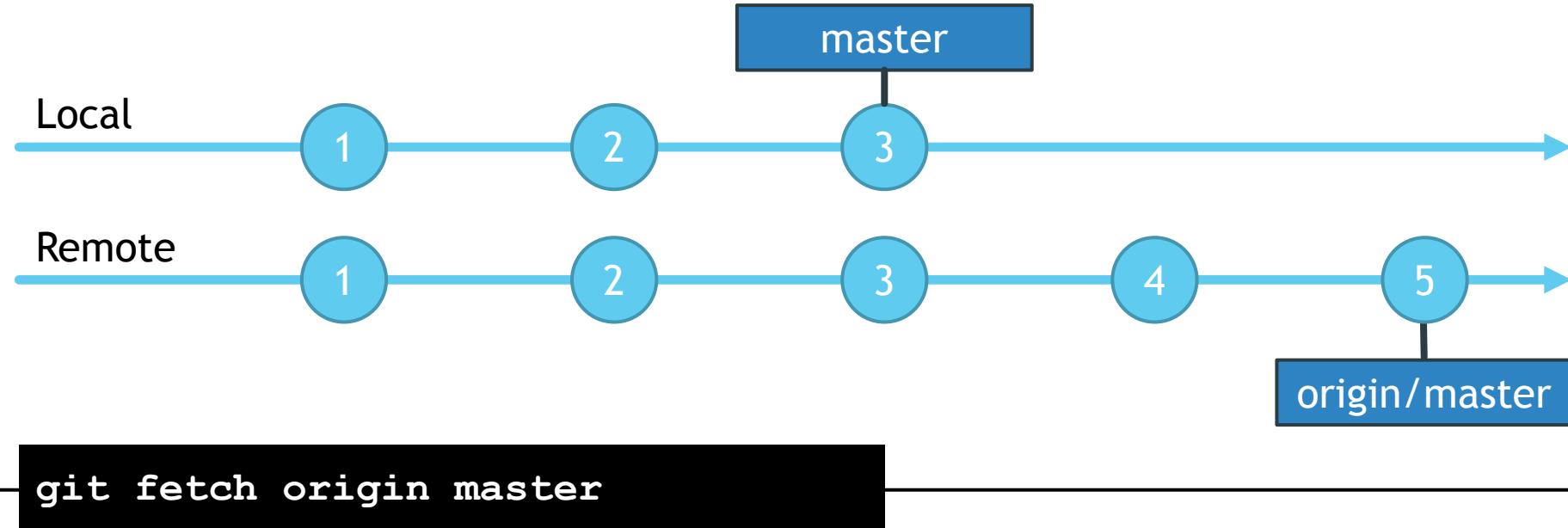
```
git fetch
```

```
git fetch [<remote> <branch>]
```

```
git fetch origin master
```

```
git push = git fetch + git merge
```

Git fetch



Git clone remote repository

```
cd ..  
git clone git@github.com:PimsJay01/git-training.git  
git remote -v  
git pull origin master
```

SSH Keys

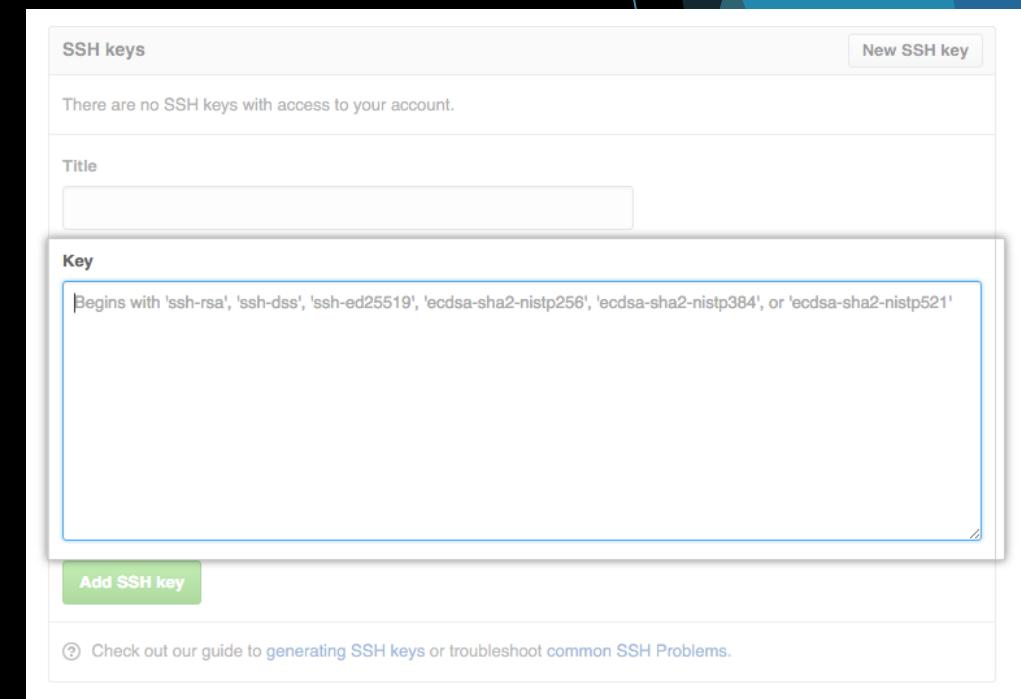
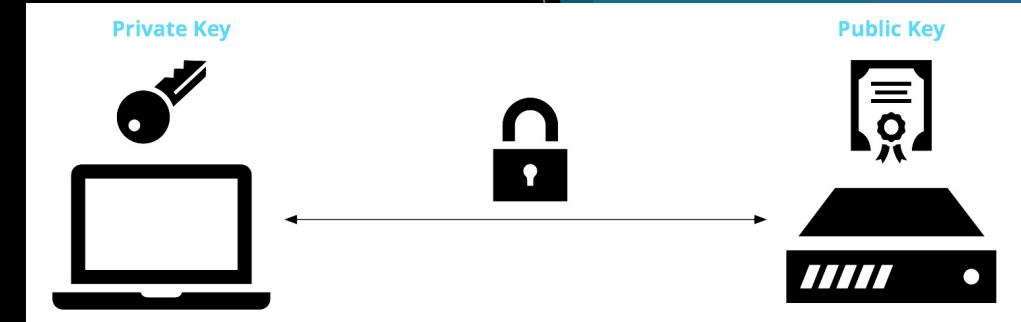
Generate SSH public & private keys

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Copies the contents of the `id_rsa.pub` file
to your clipboard

```
clip < ~/.ssh/id_rsa.pub
```

Paste public key in GitHub



At Your Service



Sturdy Templates

Our templates are updated regularly so they don't break.



Jeremy Jay Gobet

Developer



Ready to Ship

You can use this theme as is, or you can make changes!



Your name here

Developer



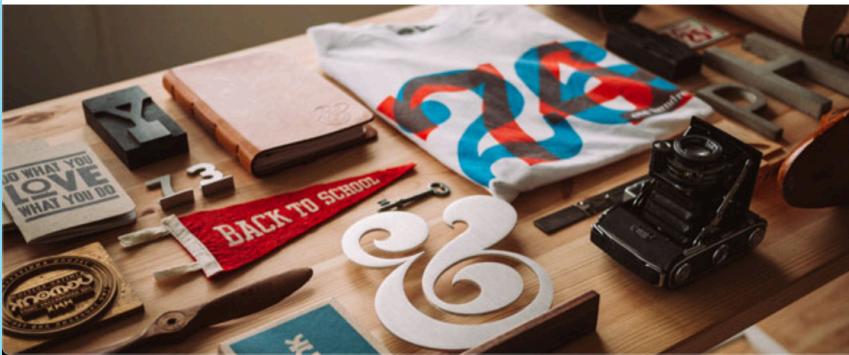
Up to Date

We update dependencies to keep things fresh.



Made with Love

You have to make your websites with love these days!



Push change

Edit file index.html (add your name)

Commit change

Check Web page on your browser

Push change

What happened ?

Vim cheatsheet

mode insert

- i

Exit mode

- ESC

Cancel

- u

Quit

- :q

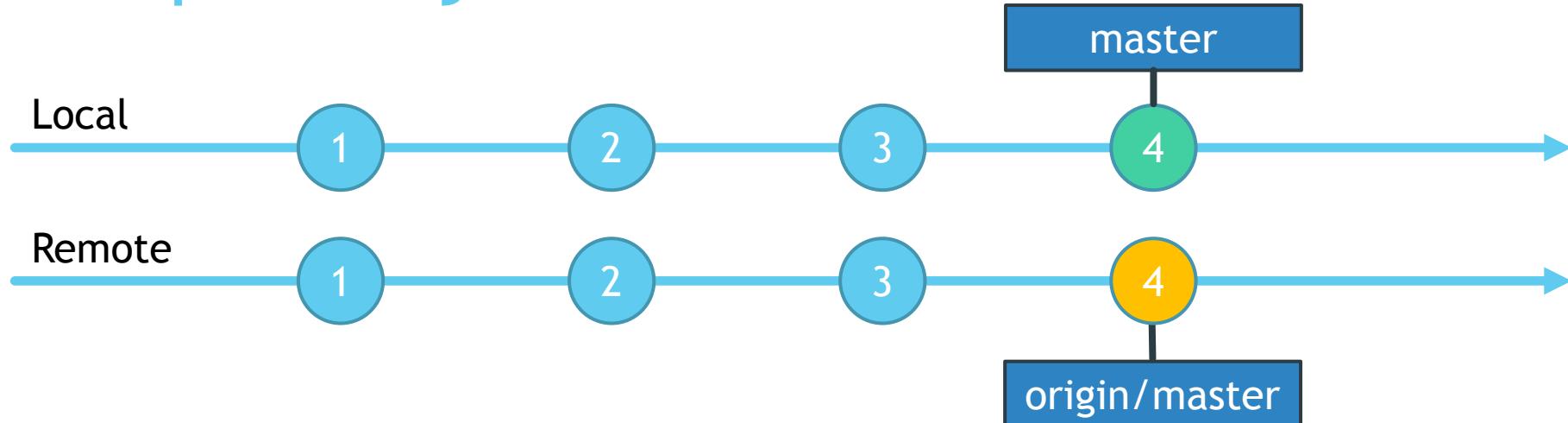
Save & Quit

- :wq

Force Quit

- :q!

Git push rejected



```
git push origin master
```

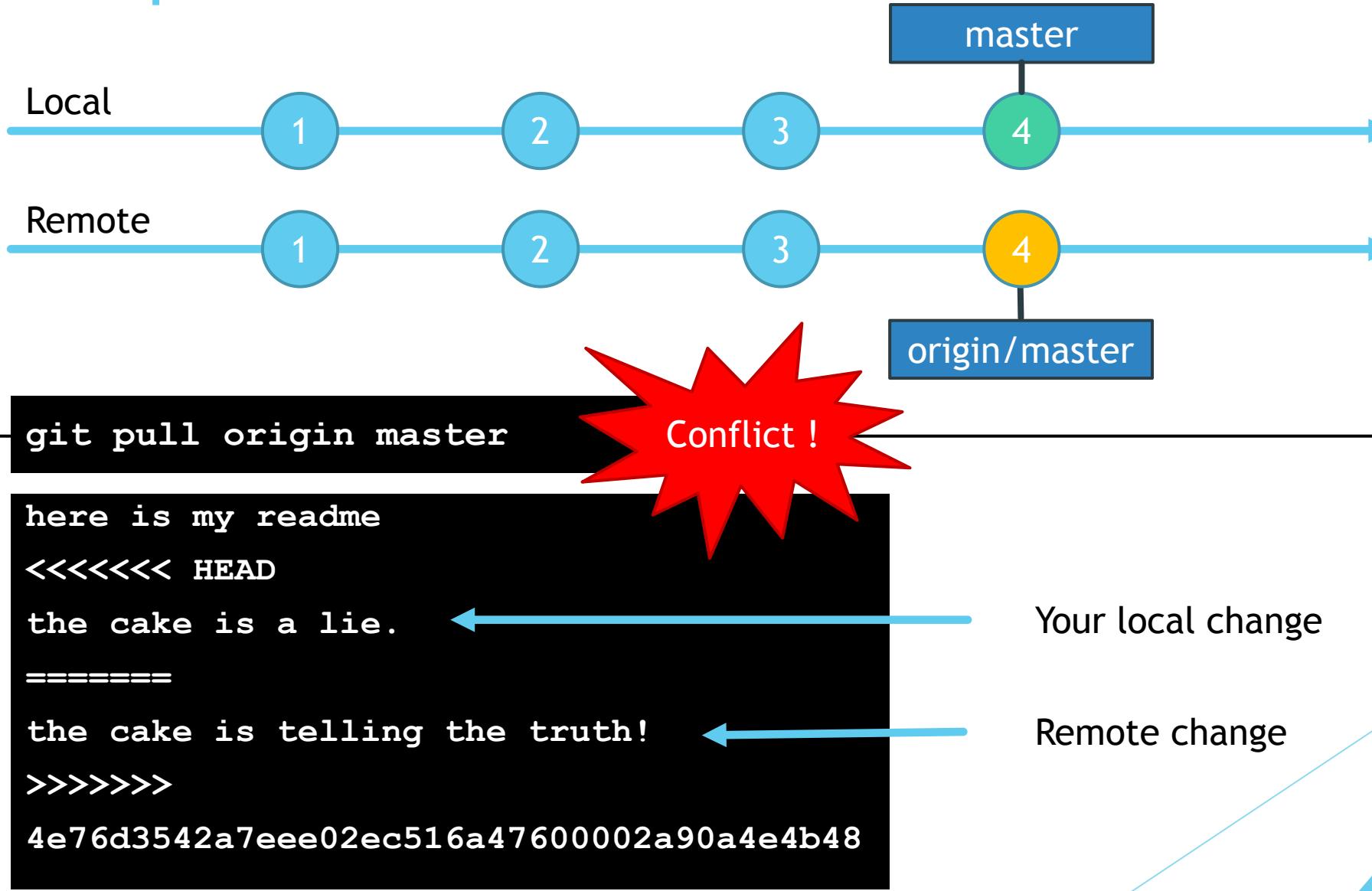
```
Cannot write over Jeremy's commit
To git@github.com:me/remote-repository.git
 ! [rejected]          master -> master (non-fast-forward)
error: failed to push some refs to 'git@github.com:me/remote-repository.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Merge the remote changes (e.g. 'git pull')
hint: before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Push change

Pull remote changes in order to push yours

What happened ?

Git pull in conflict



Visual merge tools

Merge Revisions for D:\Projects\Git\Demo\Nancy\src\Nancy\DependencyContextAssemblyCatalog.cs

5 changes. 1 conflict

Local Changes (Read-only)		Result		Changes from Server (revision 9d4b6795...)	
1	using System.Reflection;	7	7	using System.Reflection;	7
2	using Microsoft.Extensions.Caching.Distributed;	8	8	using Microsoft.Extensions.Caching.Distributed;	8
	X >> 9	9	9	9 << X	
	/// <summary>	10	10	/// <summary>	10
	/// Default implementation for retrieving a dependency.	11	11	/// Default implementation for retrieving a dependency.	11
	/// retrieving <see cref="DependencyContextAssemblyCatalog"/>	12	12	/// retrieving <see cref="DependencyContextAssemblyCatalog"/>	12
	/// </summary>	13	13	/// </summary>	13
3	public class Depender	14	14	public class DependencyContextAssemblyCatalog	14
4	{	15	15	{	15
5	private static readonly DependencyContextAssemblyCatalog _depender;	16	16	private static readonly DependencyContextAssemblyCatalog _depender;	16
6	private readonly IDependencyContextAssemblyCatalog _dependencyContextAssemblyCatalog;	17	17	private readonly IDependencyContextAssemblyCatalog _dependencyContextAssemblyCatalog;	17
7	/// <summary>	18	18	/// <summary>	18
8	/// Initializes a new instance of the <code>Depender</code> class.	19	19	/// Initializes a new instance of the <code>DependencyContextAssemblyCatalog</code> class.	19
9	/// using <see cref="IDependencyContextAssemblyCatalog"/> for dependency resolution.	20	20	/// using <see cref="IDependencyContextAssemblyCatalog"/> for dependency resolution.	20
10	/// </summary>	21	21	/// </summary>	21
11	public DependencyContextAssemblyCatalog(IDependencyContextAssemblyCatalog dependencyContextAssemblyCatalog)	22	22	public DependencyContextAssemblyCatalog(IDependencyContextAssemblyCatalog dependencyContextAssemblyCatalog)	22
12	: this(dependencyContextAssemblyCatalog)	23	23	: this(dependencyContextAssemblyCatalog)	23
13	}	24	X >> 24	var entryAssemblyCatalog = dependencyContextAssemblyCatalog;	24
14	}	25	25	X << 25	this.dependencyContextAssemblyCatalog = entryAssemblyCatalog;
15	X >> 26	26	26	X << 26	this.dependencyContextAssemblyCatalog = entryAssemblyCatalog;
16	/// <summary>	27	27	/// <summary>	27
17	/// Initializes a new instance of the <code>DependencyContextAssemblyCatalog</code> class.	28	28	/// Gets all dependencies for the specified assembly.	28
18	/// using <param name="assemblyName"/> for assembly resolution.	29	29	/// <summary>	29
19	/// </summary>	30	30	/// Gets all dependencies for the specified assembly.	30
20	public DependencyContextAssemblyCatalog(string assemblyName)	31	31	/// <summary>	31
21	: this(assemblyName)	32	32	/// Gets all dependencies for the specified assembly.	32
22	}	33	33	/// <returns>An <code>IReadOnlyCollection<DependencyContextAssemblyCatalog></code> containing all dependencies for the specified assembly.	33
23	dependencyContextAssemblyCatalog = new Dictionary<string, DependencyContextAssemblyCatalog>();	34	34	public IReadOnlyCollection<DependencyContextAssemblyCatalog> GetDependencies(string assemblyName)	34
24	}	35	35	{	35
25		36	36	var results = new List<DependencyContextAssemblyCatalog>();	36
26		37	37	foreach (var entryAssemblyCatalog in dependencyContextAssemblyCatalog)	37
27		38	38	results.Add(entryAssemblyCatalog);	38
28		39	39	return results;	39
29		40	40	}	40
30		41	41		41
31		42	42		42
32		43	43		43
33		44	44		44
34		45	45		45
35		46	46		46
36		47	47		47
37		48	48		48
38		49	49		49
39		50	50		50
40		51	51		51
41		52	52		52
42		53	53		53
43		54	54		54
44		55	55		55
45		56	56		56
46		57	57		57
47		58	58		58
48		59	59		59
49		60	60		60
50		61	61		61
51		62	62		62
52		63	63		63
53		64	64		64
54		65	65		65
55		66	66		66
56		67	67		67
57		68	68		68
58		69	69		69
59		70	70		70
60		71	71		71
61		72	72		72
62		73	73		73
63		74	74		74
64		75	75		75
65		76	76		76
66		77	77		77
67		78	78		78
68		79	79		79
69		80	80		80
70		81	81		81
71		82	82		82
72		83	83		83
73		84	84		84
74		85	85		85
75		86	86		86
76		87	87		87
77		88	88		88
78		89	89		89
79		90	90		90
80		91	91		91
81		92	92		92
82		93	93		93
83		94	94		94
84		95	95		95
85		96	96		96
86		97	97		97
87		98	98		98
88		99	99		99
89		100	100		100
90		101	101		101
91		102	102		102
92		103	103		103
93		104	104		104
94		105	105		105
95		106	106		106
96		107	107		107
97		108	108		108
98		109	109		109
99		110	110		110
100		111	111		111
101		112	112		112
102		113	113		113
103		114	114		114
104		115	115		115
105		116	116		116
106		117	117		117
107		118	118		118
108		119	119		119
109		120	120		120
110		121	121		121
111		122	122		122
112		123	123		123
113		124	124		124
114		125	125		125
115		126	126		126
116		127	127		127
117		128	128		128
118		129	129		129
119		130	130		130
120		131	131		131
121		132	132		132
122		133	133		133
123		134	134		134
124		135	135		135
125		136	136		136
126		137	137		137
127		138	138		138
128		139	139		139
129		140	140		140
130		141	141		141
131		142	142		142
132		143	143		143
133		144	144		144
134		145	145		145
135		146	146		146
136		147	147		147
137		148	148		148
138		149	149		149
139		150	150		150
140		151	151		151
141		152	152		152
142		153	153		153
143		154	154		154
144		155	155		155
145		156	156		156
146		157	157		157
147		158	158		158
148		159	159		159
149		160	160		160
150		161	161		161
151		162	162		162
152		163	163		163
153		164	164		164
154		165	165		165
155		166	166		166
156		167	167		167
157		168	168		168
158		169	169		169
159		170	170		170
160		171	171		171
161		172	172		172
162		173	173		173
163		174	174		174
164		175	175		175
165		176	176		176
166		177	177		177
167		178	178		178
168		179	179		179
169		180	180		180
170		181	181		181
171		182	182		182
172		183	183		183
173		184	184		184
174		185	185		185
175		186	186		186
176		187	187		187
177		188	188		188
178		189	189		189
179		190	190		190
180		191	191		191
181		192	192		192
182		193	193		193
183		194	194		194
184		195	195		195
185		196	196		196
186		197	197		197
187		198	198		198
188		199	199		199
189		200	200		200
190		201	201		201
191		202	202		202
192		203	203		203
193		204	204		204
194		205	205		205
195		206	206		206
196		207	207		207
197		208	208		208
198		209	209		209
199		210	210		210
200		211	211		211
201		212	212		212
202		213	213		213
203		214	214		214
204		215	215		215
205		216	216		216
206		217	217		217
207		218	218		218
208		219	219		219
209		220	220		220
210		221	221		221
211		222	222		222
212		223	223		223
213		224	224		224
214		225	225		225
215		226	226		226
216		227	227		227
217		228	228		228
218		229	229		229
219		230	230		230
220		231	231		231
221		232	232		232
222		233	233		233
223		234	234		234
224		235	235		235
225		236	236		236
226		237	237		237
227		238	238		238
228		239	239		239
229		240	240		240
230		241	241		241
231		242	242		242
232		243	243		243
233		244	244		244
234		245	245		245
235		246	246		246
236		247	247		247
237		248	248		248
2					

Push change

Resolve conflict

```
vim index.html (or use Visual Studio)
```

```
add index.html
```

```
git commit -m "Resolved conflict"
```

Vim cheatsheet

mode insert

- i

Exit mode

- ESC

Cancel

- u

Quit

- :q

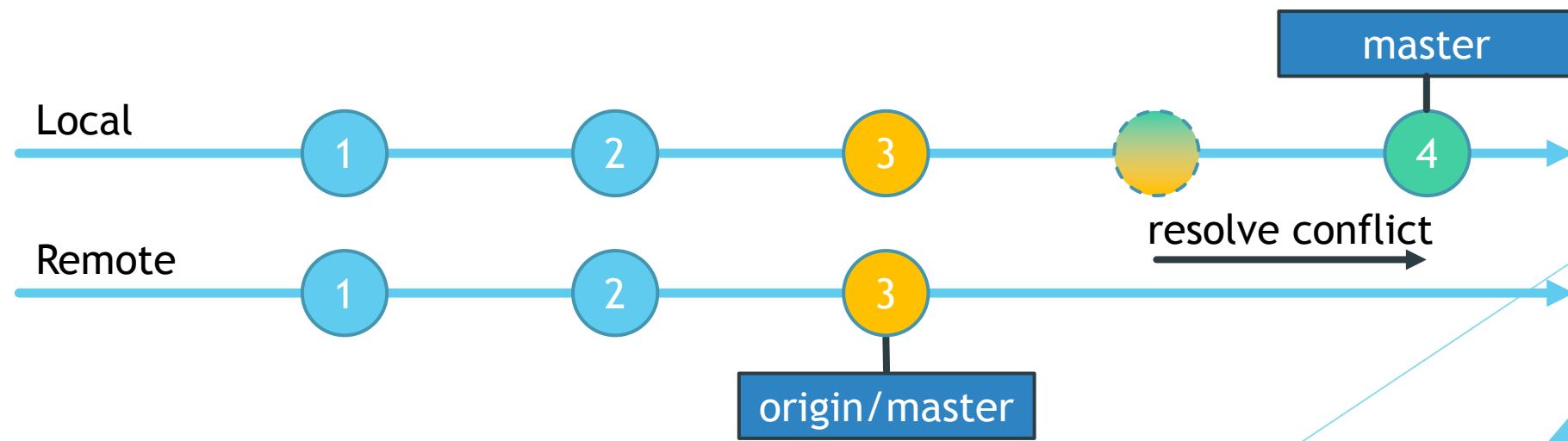
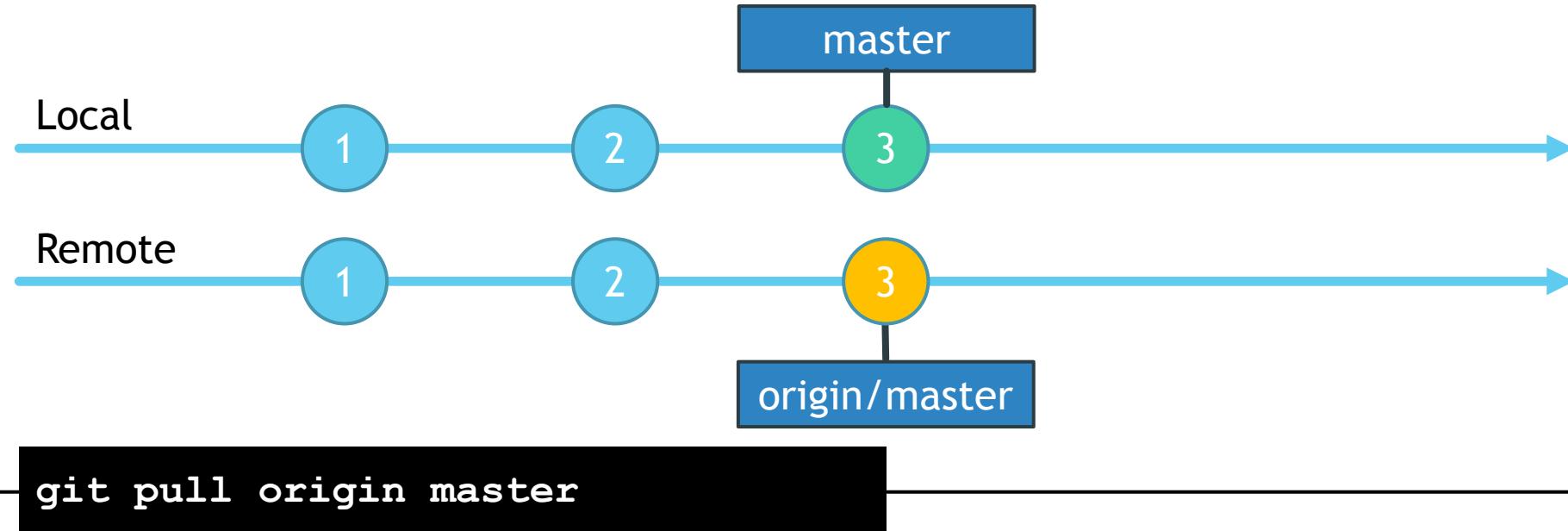
Save & Quit

- :wq

Force Quit

- :q!

Conflict resolved



Git remote

Add reference to remote repository

```
git remote add myrepo git@github.com:<user>/<repo>.git
```

```
git remote add myrepo https://github.com/<user>/<repo>.git
```

Display references

```
git remote -v
```

origin git@github.com:PimsJay01/git-training.git	(fetch)
origin git@github.com:PimsJay01/git-training.git	(push)

Delete reference

```
git remote remove myrepo
```

Edit remote URL

Create new personal repository on GitHub

Push local repository to the new repository

```
git remote add origin git@github.com:<user>/<repo>.git
```

```
git remote -v
```

```
git push origin master
```

```
git status
```

branches & workflows

Git branch

Create a local branch

```
git branch my-branch
```

Switch your work environment to another branch

```
git checkout my-branch
```

Show local branches

```
git branch
```

Remove branch

```
git branch -D my-branch
```

Create feature branch

Clone remote repository with a custom name

```
cd ..
```

```
git clone git@github.com:PimsJay01/git-training.git mynewrepo
```

```
cd mynewrepo
```

Create branch with your name

```
git branch add_name-<your-name>
```

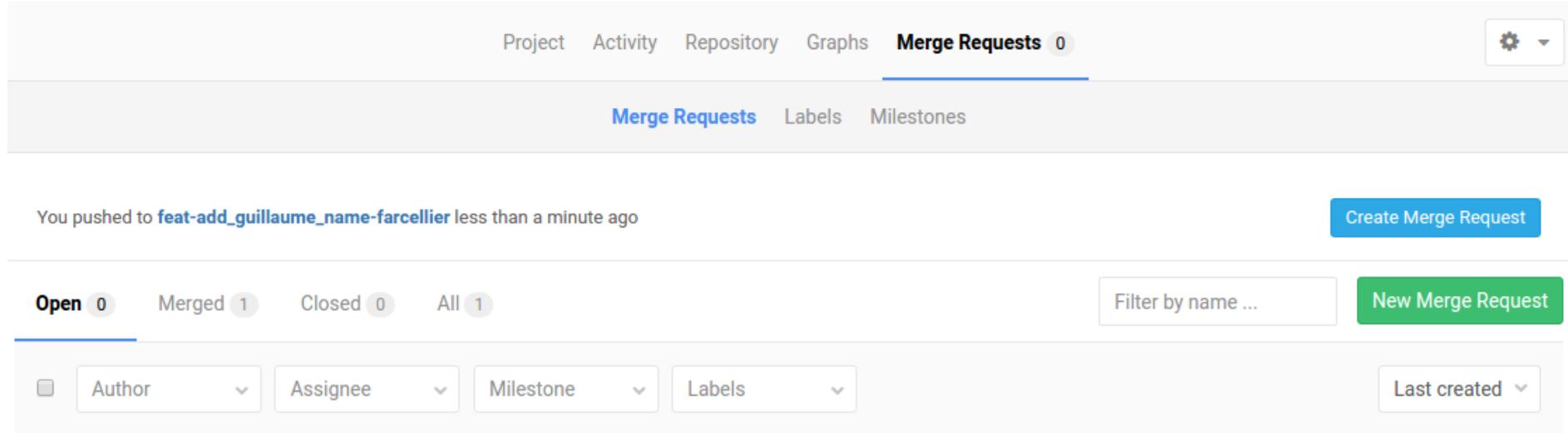
```
git checkout add_name-<your-name>
```

Edit file index.html to add you name

Commit and push change

```
git push --set-upstream origin add_name-<your-name>
```

Create a merge-request on GitHub



The screenshot shows the GitHub interface for managing merge requests. At the top, there is a navigation bar with tabs: Project, Activity, Repository, Graphs, and Merge Requests (which is currently selected). Below the navigation bar is a secondary navigation bar with links: Merge Requests, Labels, and Milestones. A message at the top left indicates a recent push to a branch named `feat-add_guillaume_name-farcellier`. On the right side of the top bar is a "Create Merge Request" button. Below the top bar, there are filters for Open (0), Merged (1), Closed (0), and All (1) merge requests. There are also dropdowns for Author, Assignee, Milestone, Labels, and a sorting dropdown set to "Last created". A search bar labeled "Filter by name ..." is positioned above the main list area. The main content area displays the message "No merge requests to show".

- ▶ Demo

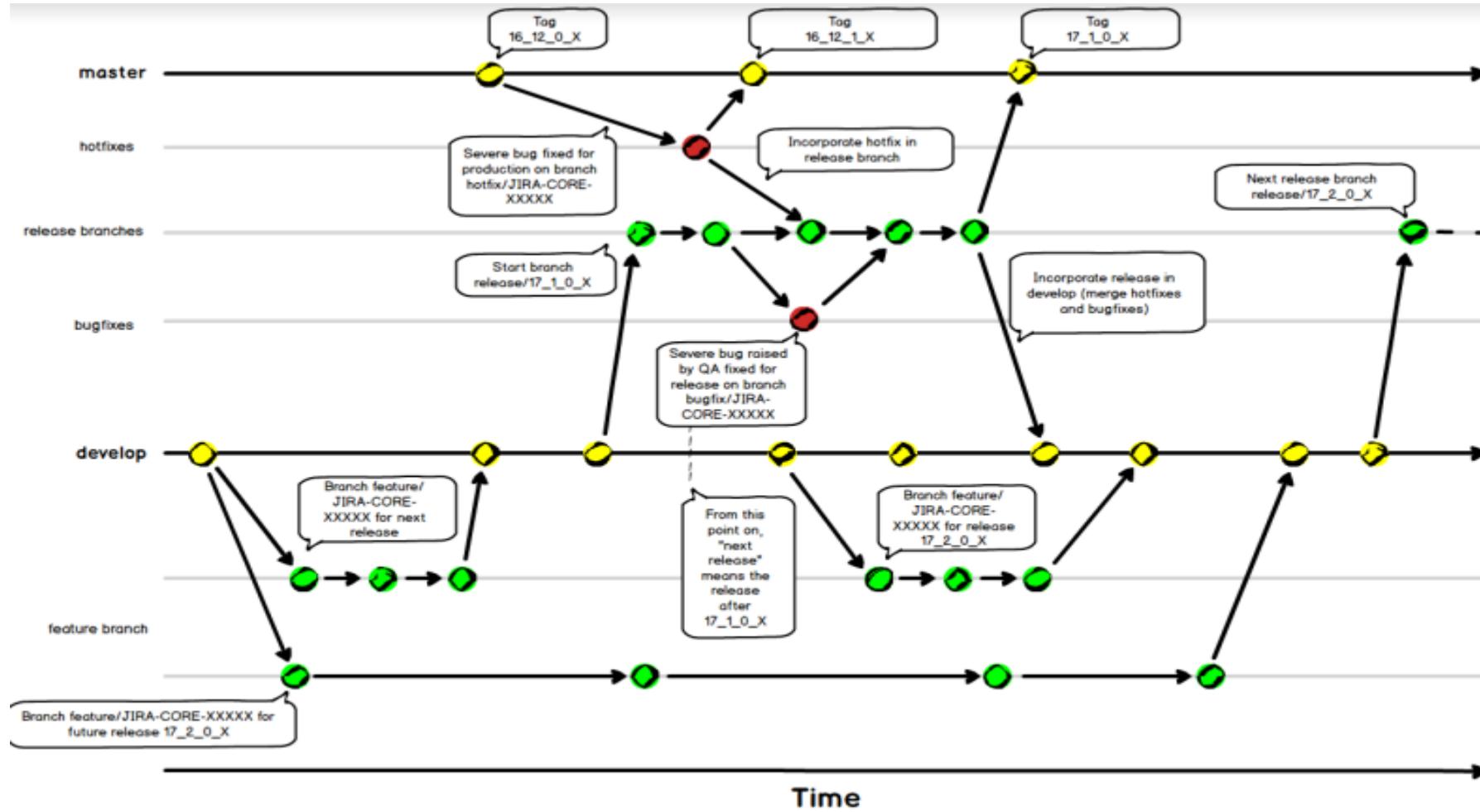
Feature branch & merge request

- ▶ A clean way to work with Git & branches but merge create additional commits



?	Nicholas C. Za...	8740984	Merge pull request #303 from ilyavolodin/eslint_errors
?	Ilya Volodin	477b420	Fixing ESLint errors
?	Nicholas C. Za...	03090fd	Merge pull request #297 from ilyavolodin/upgrade-escape
?	Ilya Volodin	7913cbe	Upgrading escape version and fixing related bugs
?	Nicholas C. Za...	4b42681	Merge pull request #299 from ilyavolodin/no-unused-vars
?	Ilya Volodin	0cfb18b	Fixing assignment during initialization issue

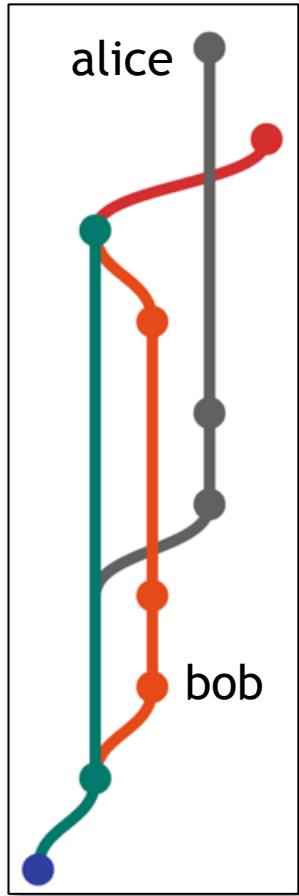
Workflow GitFlow



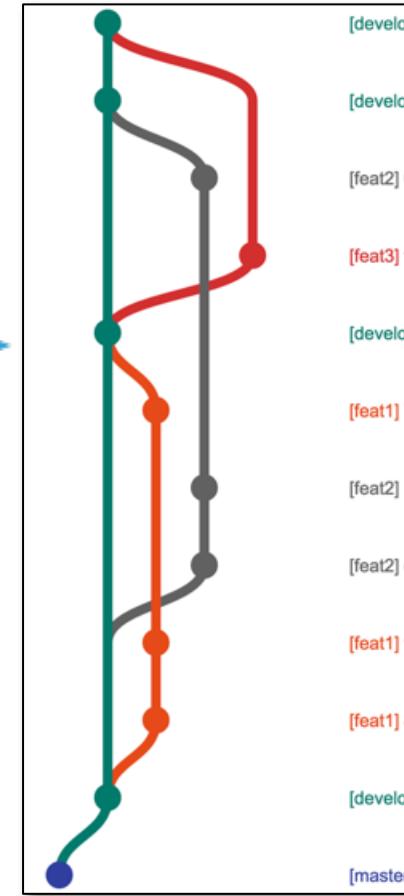
Rewrite history

Classic merge flow vs rebase flow

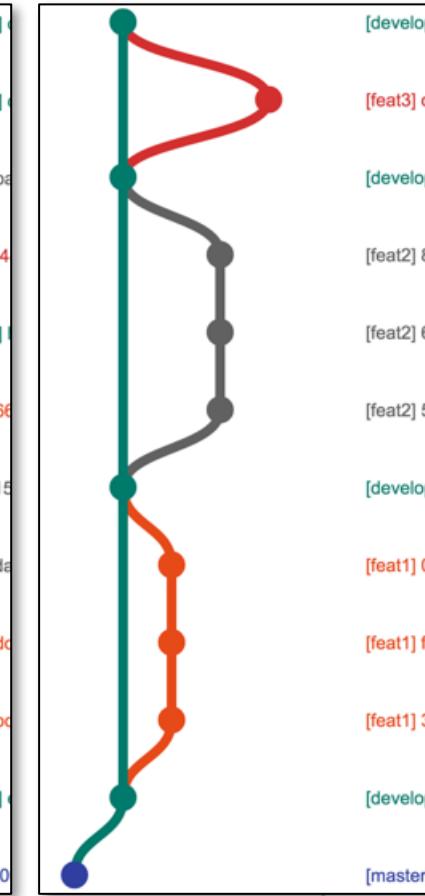
Real commits order



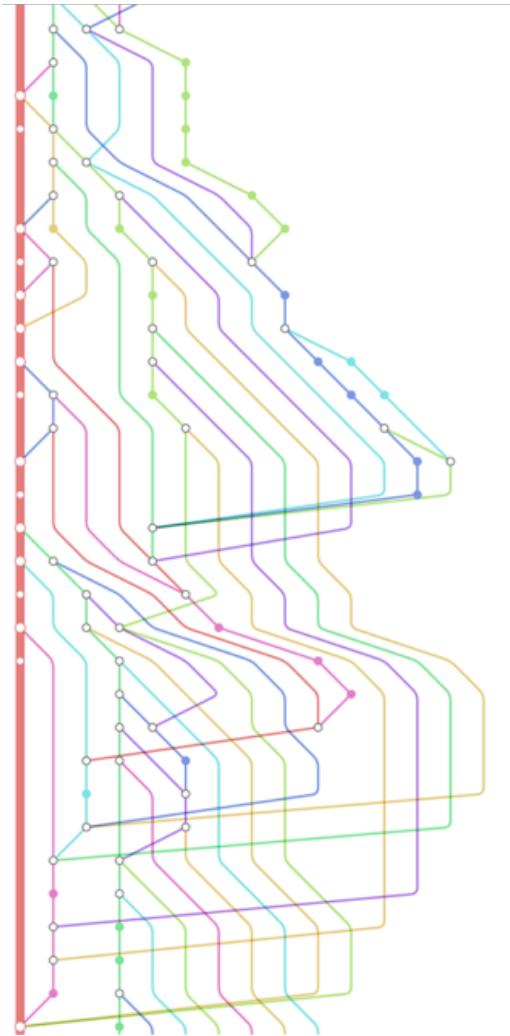
Merge flow



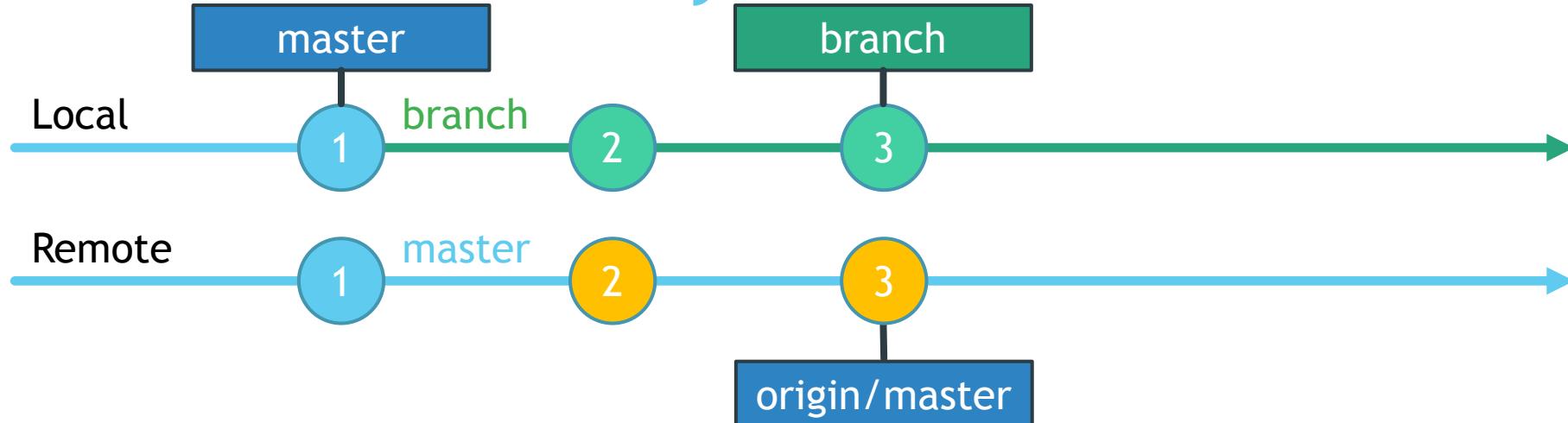
Rebase flow



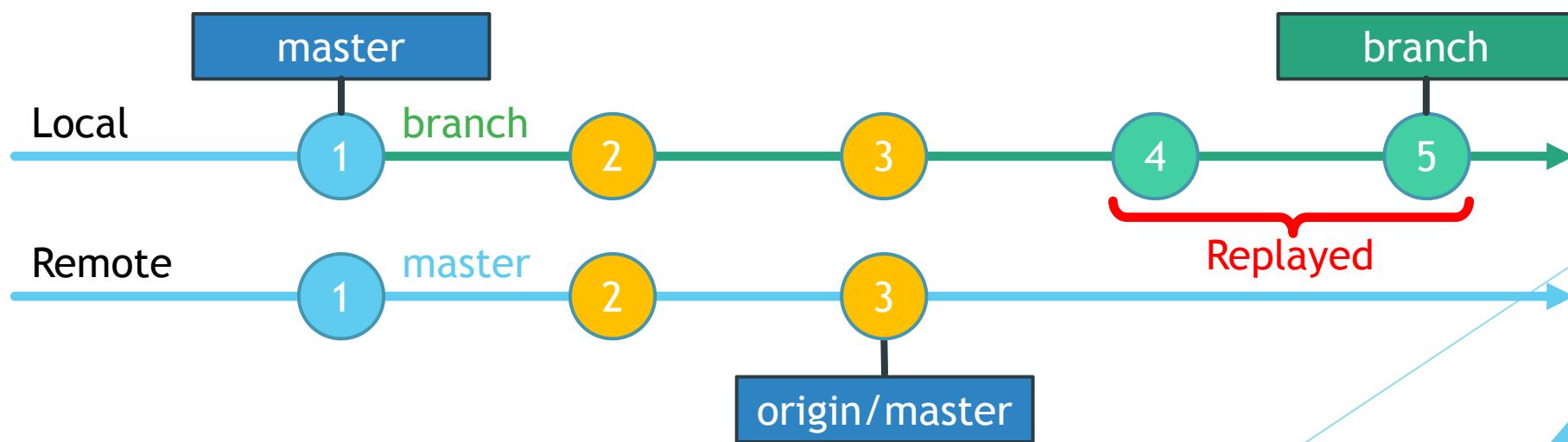
Why rebasing ?



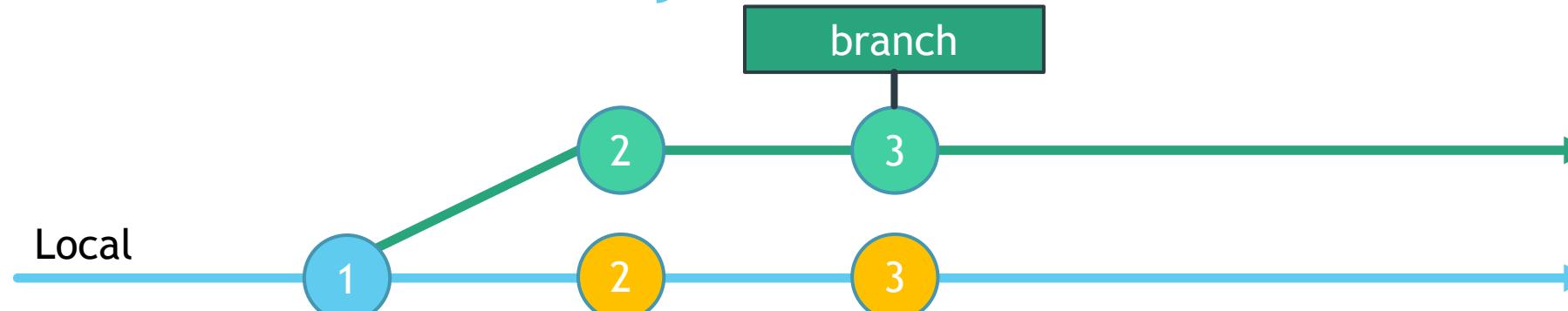
Git rebase remotely



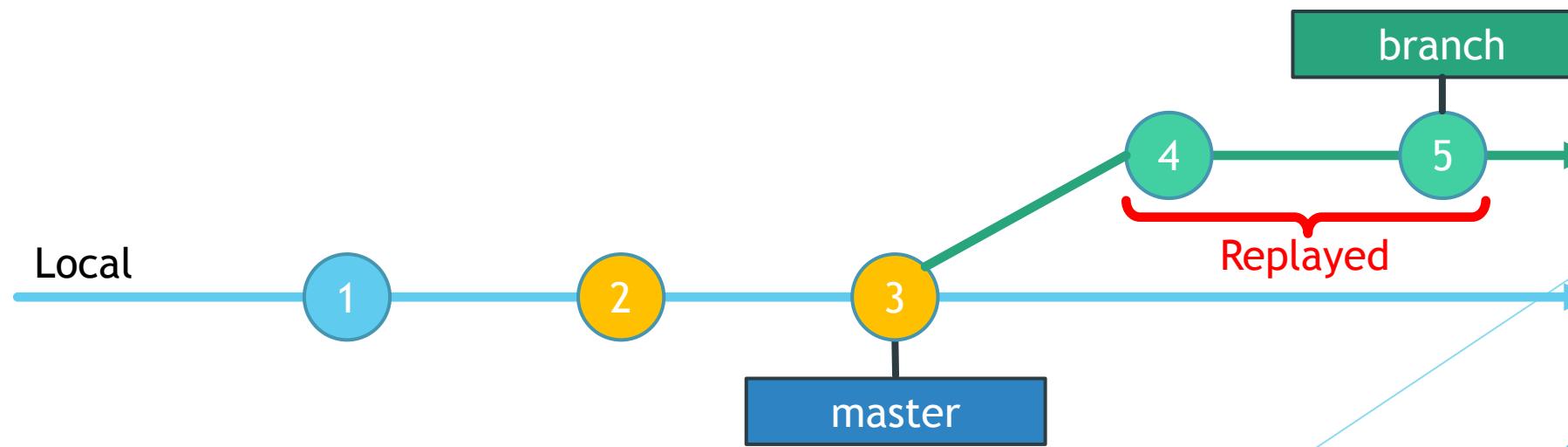
```
git pull --rebase origin master
```



Git rebase locally



git rebase master



Rebase your branch from origin master

```
git pull --rebase origin master
```

Resolve conflict

```
vim index.html (or use Visual Studio)
```

```
git commit -am "Resolved conflict"
```

```
git status
```

Push your change

```
git push origin add_name-<your-name>
```

What happened ?

Vim cheatsheet

mode insert

- i

Exit mode

- ESC

Cancel

- u

Quit

- :q

Save & Quit

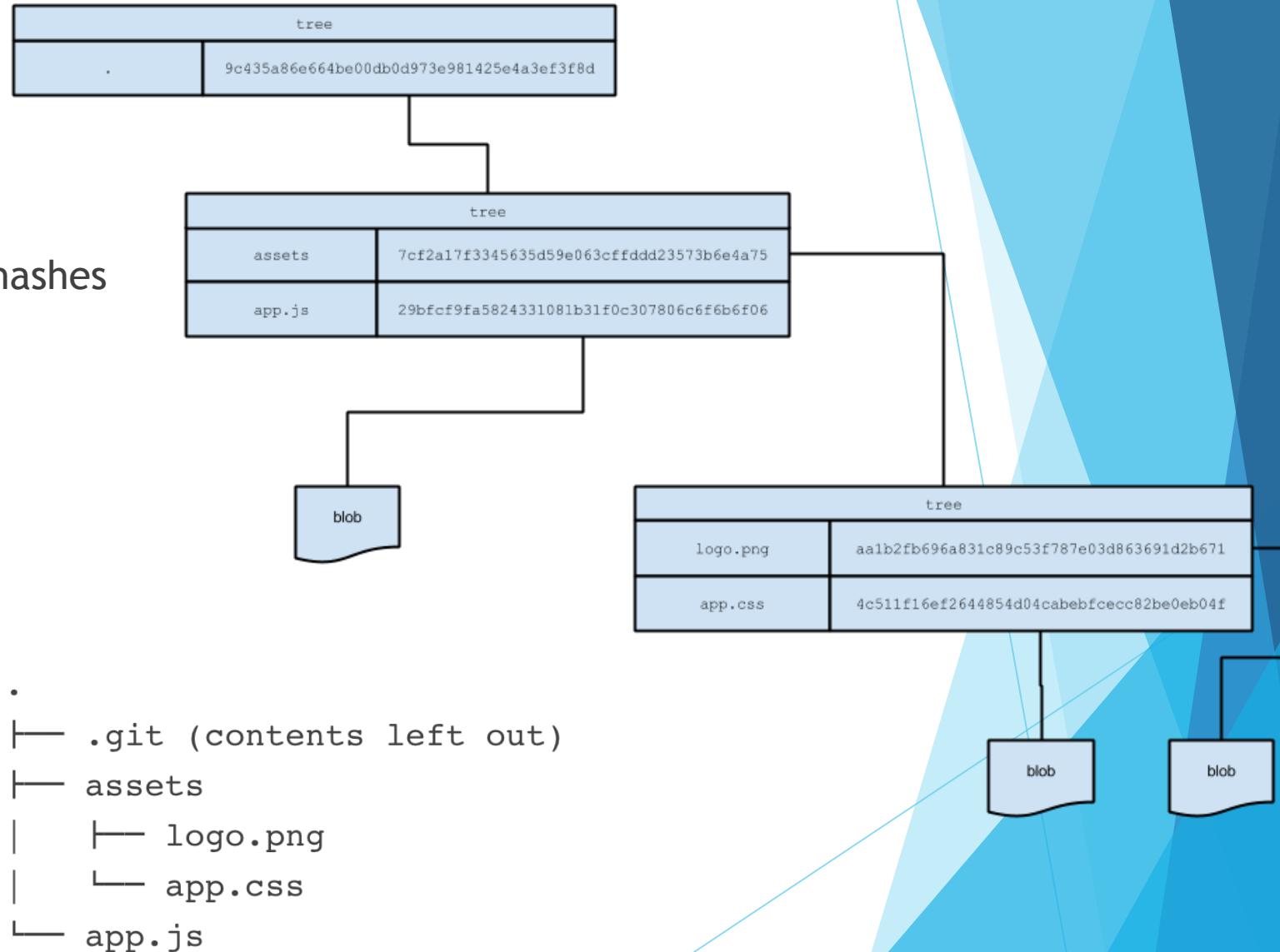
- :wq

Force Quit

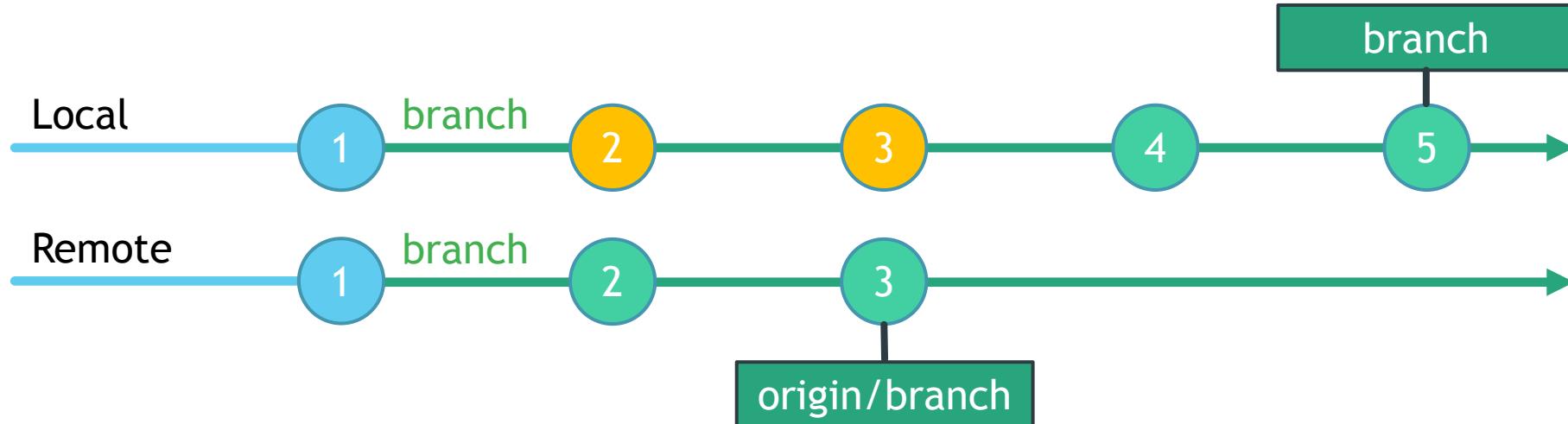
- :q!

Hash

- ▶ SHA-1 is used to create unique hashes from files in your repository
- ▶ This hash allows to check
 - ▶ Identity
 - ▶ Integrity
 - ▶ Checksum



Git rebase



```
git push origin master
```

```
Cannot write over Jeremy's commit  
To git@github.com:me/remote-repository.git  
! [rejected]          master -> master (non-fast-forward)
```

```
git push --force origin master
```



Push changes on your branch

```
git push origin add_name-<your-name>
```

Rejected !

```
git push --force origin add_name-<your-name>
```

```
git log
```

**PUSH REJECTED, REBASE OR
MERGE**



GIT PUSH --FORCE

Feature branch on a daily basis

- ▶ Create a working branch
- ▶ Commit frequently
- ▶ Synchronize frequently → Git rebase
- ▶ Push changes on your branch every day
- ▶ When the new feature is done → merge request
 - ▶ Find a colleague and valid merge request together

Other commands

Git log

Display logs

```
git log
```

Display logs as a graph

```
git log --graph
```

Display all commits concerning a file

```
git log -p README.txt
```

```
| * commit 59efe4110cdedd48db7477a76d275e014e2bf713 (origin/bug/158-align_picto, bug/158-align_picto)
| | Author: Jérémie Gobet <jgobet@octo.com>
| | Date: Mon Jun 17 15:04:47 2019 +0200
| |
| | Remove parameter avatarUrl when using component avatar in profile-summary (use default avatar until implementation)
|
| * commit d451c2a17eddac4d27295e02ed1d37a7739b79ea
| | Author: Jérémie Gobet <jgobet@octo.com>
| | Date: Mon Jun 17 14:43:03 2019 +0200
| |
| | Use same avatar in the profile page than the one in the menu
|
| * commit 717b072b437c699167d193d954a3d8ec3f728b8e
| / Author: Jérémie Gobet <jgobet@octo.com>
| | Date: Mon Jun 17 12:14:09 2019 +0200
| |
| | Set menu's header item not clickable
|
* commit 6537086e0a3670a01c91500fa9c42719c8bfcfbe
|\ Merge: 1500dc7b e5694682
| | Author: Alban Seurat <alban.seurat@me.com>
| | Date: Mon Jun 17 07:57:15 2019 +0000
| |
| | Merge branch 'fix/154-tri-offres' into 'master'
| |
| | offer sort : if same collect end date, sort by offer end date
| |
| | See merge request trivert/trivert!235
| |
| * commit e56946825e32369f1567be666538ecc598c6b054
| / Author: Mathieu Brun <mbrun@octo.com>
| | Date: Sun Jun 16 15:52:08 2019 +0200
| |
| | offer sort : if same collect end date, sort by offer end date
|
* commit 1500dc7b0b22669736529c3b19a6899067861190
|\ Merge: 13e6434e 7d2e2eb8
| | Author: Alban Seurat <alban.seurat@me.com>
| | Date: Mon Jun 17 07:47:36 2019 +0000
| |
| | Merge branch 'bugfix/feedback-payment' into 'master'
| |
| | bugfix feedback for payment
| |
| | See merge request trivert/trivert!234
| |
| * commit 7d2e2eb88565ad04aa0e8c1edba001a7dad5032f
| / Author: Alban Seurat <alban.seurat@me.com>
| | Date: Mon Jun 17 08:55:25 2019 +0200
| |
| | bugfix feedback for payment
```

Git stash

Save your work temporarily

`git stash`

Retrieve your work

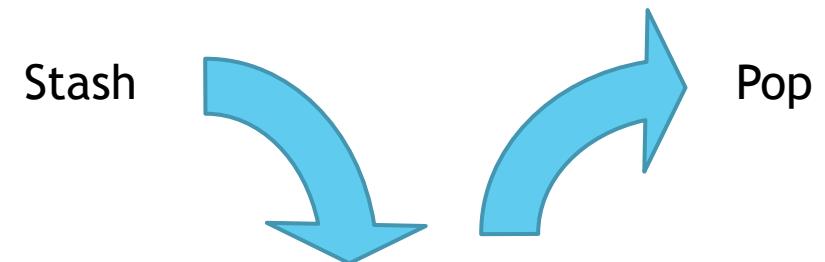
`git pop`

Show all stashes

`git stash list`

Stash list is FILO (First In Last Out)

Stash & Pop



Stash 2

Stash 1

Stash 0

Git tag

A tag is a way to mark and name a commit

Add a tag

```
git tag -a v0.2 -m "version 0.2"
```

Checkout a tag

```
git checkout v0.2
```

Git revert

Cancel last commit by creating a new one

```
git revert HEAD^
```

To avoid doing push force after a Git reset

```
git commit ...
```

```
git push
```

```
git reset --hard HEAD^
```

```
git push -f
```

Git blame

Display origin of lines

git blame <file>

```
2223b776b (Arnaud Béguin 2019-05-21 19:45:58 +0200 1) <div *ngIf="profile" class="main-container profile horizontal-content-container">
2223b776b (Arnaud Béguin 2019-05-21 19:45:58 +0200 2)   <app-profile-summary [data]="profile.summary"></app-profile-summary>
2223b776b (Arnaud Béguin 2019-05-21 19:45:58 +0200 3)   <app-coordinates [data]="profile.coordinates"></app-coordinates>
b3decba66 (Arnaud Béguin 2019-05-24 16:54:05 +0200 4)   <app-phone-information
b3decba66 (Arnaud Béguin 2019-05-24 16:54:05 +0200 5)     [data]="profile.phone"
b3decba66 (Arnaud Béguin 2019-05-24 16:54:05 +0200 6)     (shareMyNumberChanged)="shareMyNumberChanged($event)">
b3decba66 (Arnaud Béguin 2019-05-24 16:54:05 +0200 7)   </app-phone-information>
42d35d2df (Bertrand 2019-05-22 21:52:48 +0200 8)   <app-notifications
b3decba66 (Arnaud Béguin 2019-05-24 16:54:05 +0200 9)     (newsLetterChanged)="newsLetterChanged($event)"
b3decba66 (Arnaud Béguin 2019-05-24 16:54:05 +0200 10)    (ownDemandsOrOffersNotificationChanged)="ownDemandsOrOffersNotificationChanged($event)"
b3decba66 (Arnaud Béguin 2019-05-24 16:54:05 +0200 11)   [data]="profile.notifications">
42d35d2df (Bertrand 2019-05-22 21:52:48 +0200 12) </app-notifications>
a5f42e3c0 (Jérémie Gobet 2019-06-18 12:00:29 +0200 13) <app-service-connect [data]="profile.serviceConnect" (buttonClick)="updateConnectServiceState($event)">
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 14) </app-service-connect>
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 15) <mat-divider></mat-divider>
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 16) <div class="button-container" fxLayout="row" fxLayoutAlign="space-between center">
2b203e320 (Arnaud Béguin 2019-05-24 20:28:17 +0200 17)   <button mat-flat-button color="primary" [routerLink]="'['']'">
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 18)     Retour
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 19)   </button>
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 20)   <!-- Cancel button -->
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 21)   <button mat-flat-button i18n="@account.profile.delete" color="warn" (click)="onDisableProfile()">
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 22)     Désactiver mon compte
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 23)   </button>
0d42f86ba (Arnaud Béguin 2019-05-17 14:05:25 +0200 24) </div>
dd0ded051 (Alban Seurat 2019-05-29 22:59:13 +0200 25) </div>
```

Advanced notions

Interactive rebase

Rebasing of the four last commits

git rebase -i HEAD~4

Rebasing of the last commits

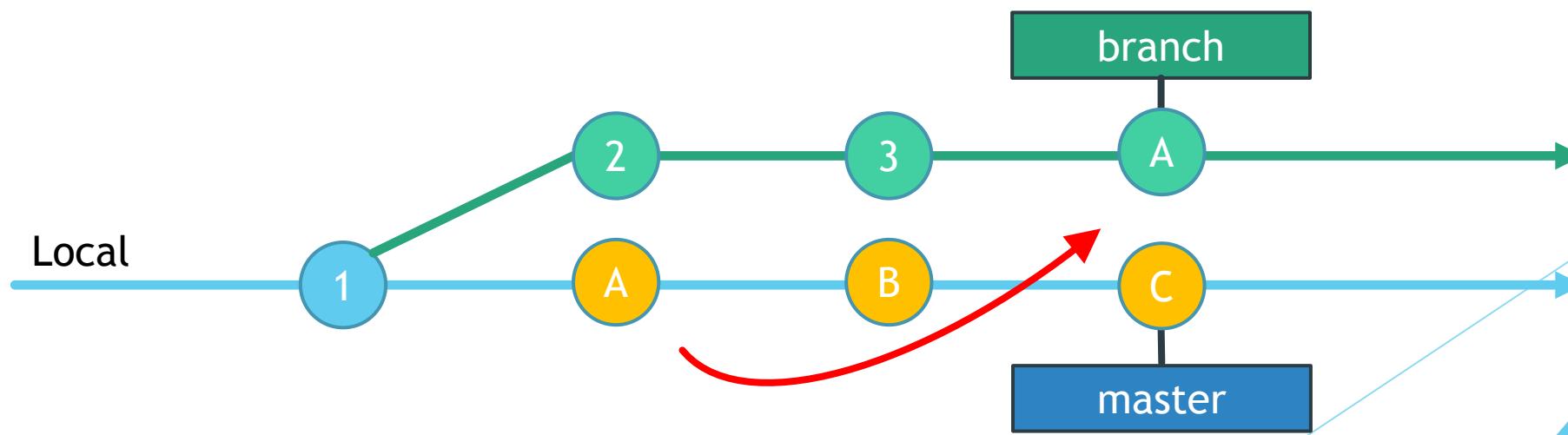
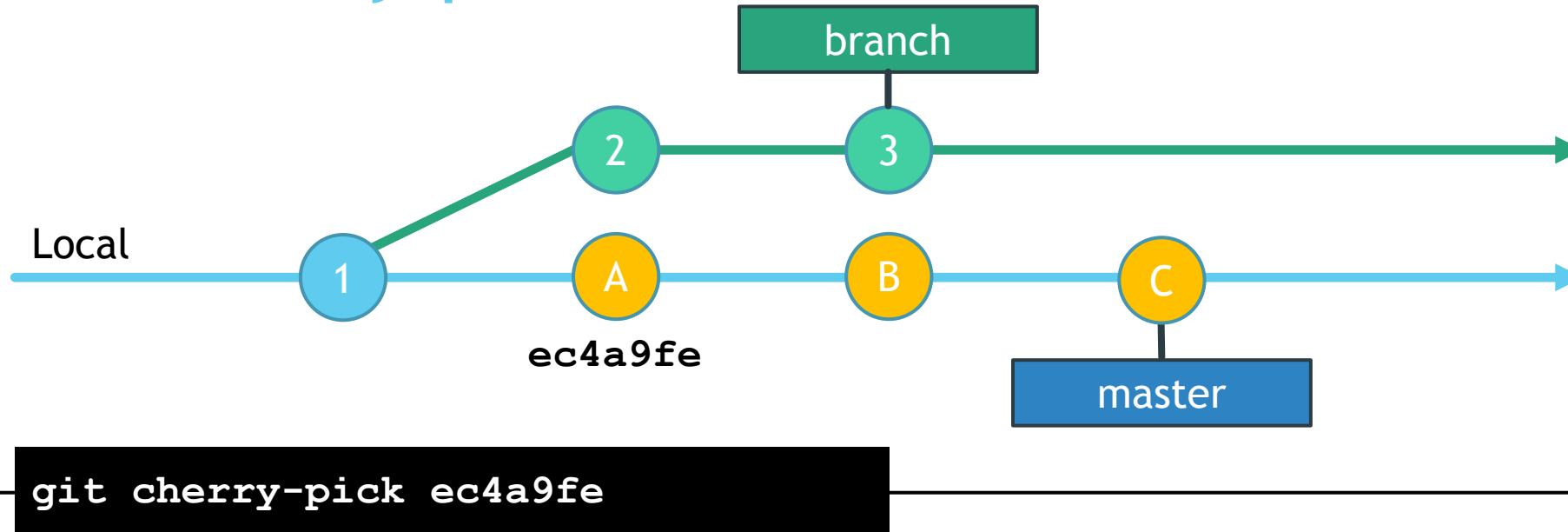
until ec4a9fe

git rebase -i ec4a9fe

```
pick ec4a9fe added gitignore file
pick 9f6cf6c included webpage
pick 9c438a8 added Matteo Mazzaeri
pick 15436bc added Salah Missri

# Rebase 5100f76..15436bc onto 5100f76 (4 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [<oneline>]
# .      create a merge commit using the original merge commit's
# .      message (or the oneline, if no original merge commit was
# .      specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

Git cherry-pick

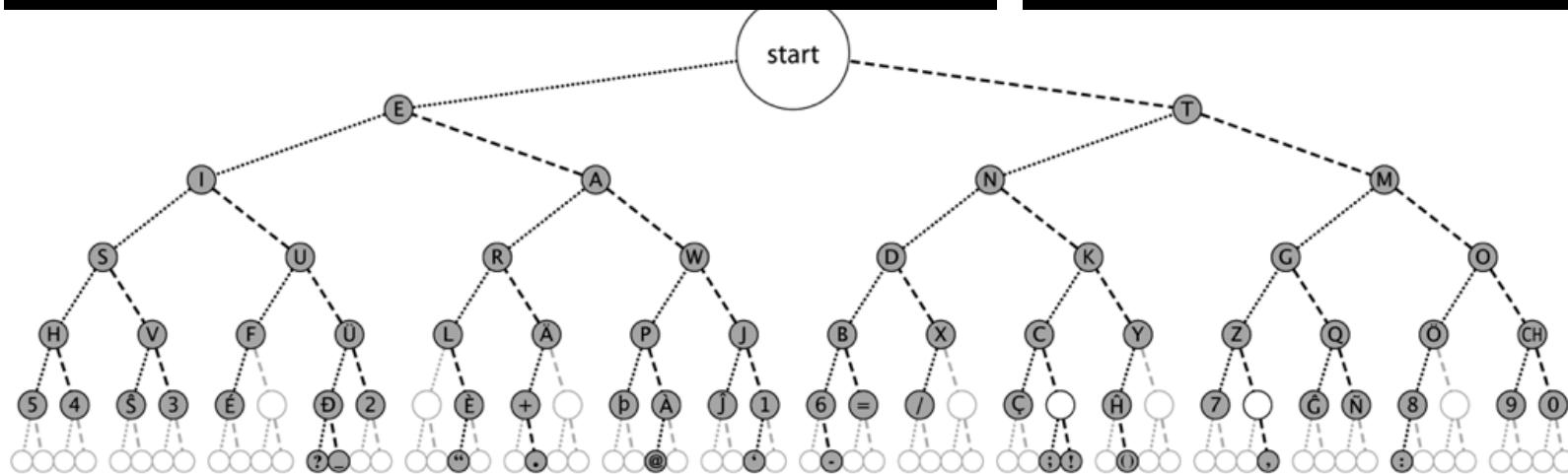


Git bissect

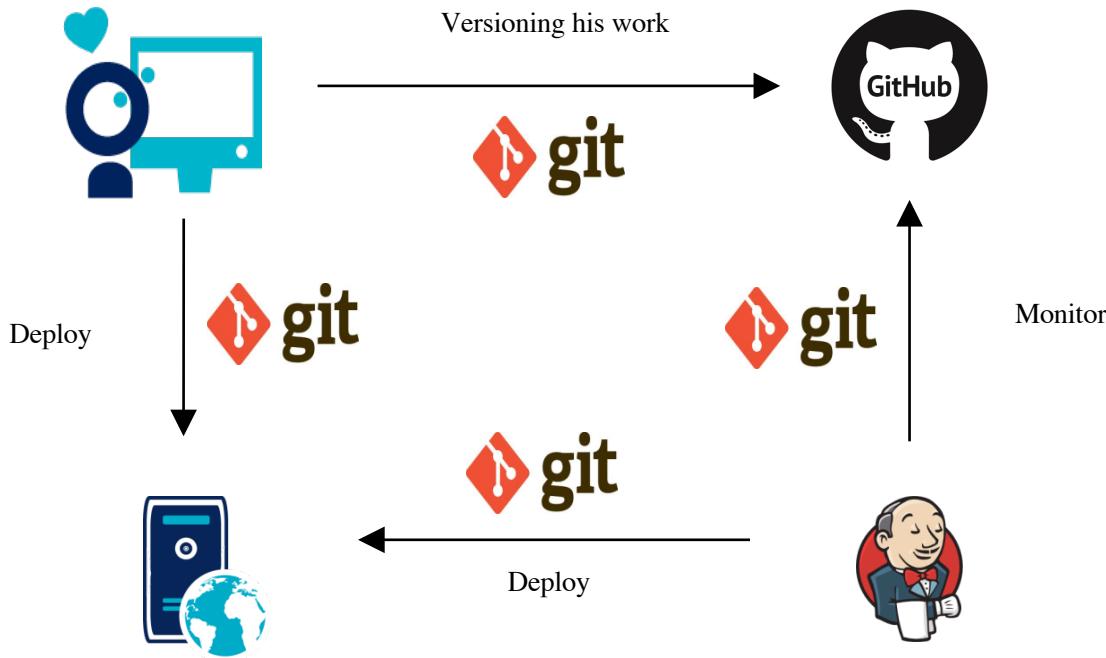
- ▶ Find the commit where a bug was introduced using Git only!
 - ▶ Works by dichotomy: compares the result of a test using different commits.
 - ▶ Finds the first commit where the test did not pass

git bisect start

```
git bisect run <cmd>...
```

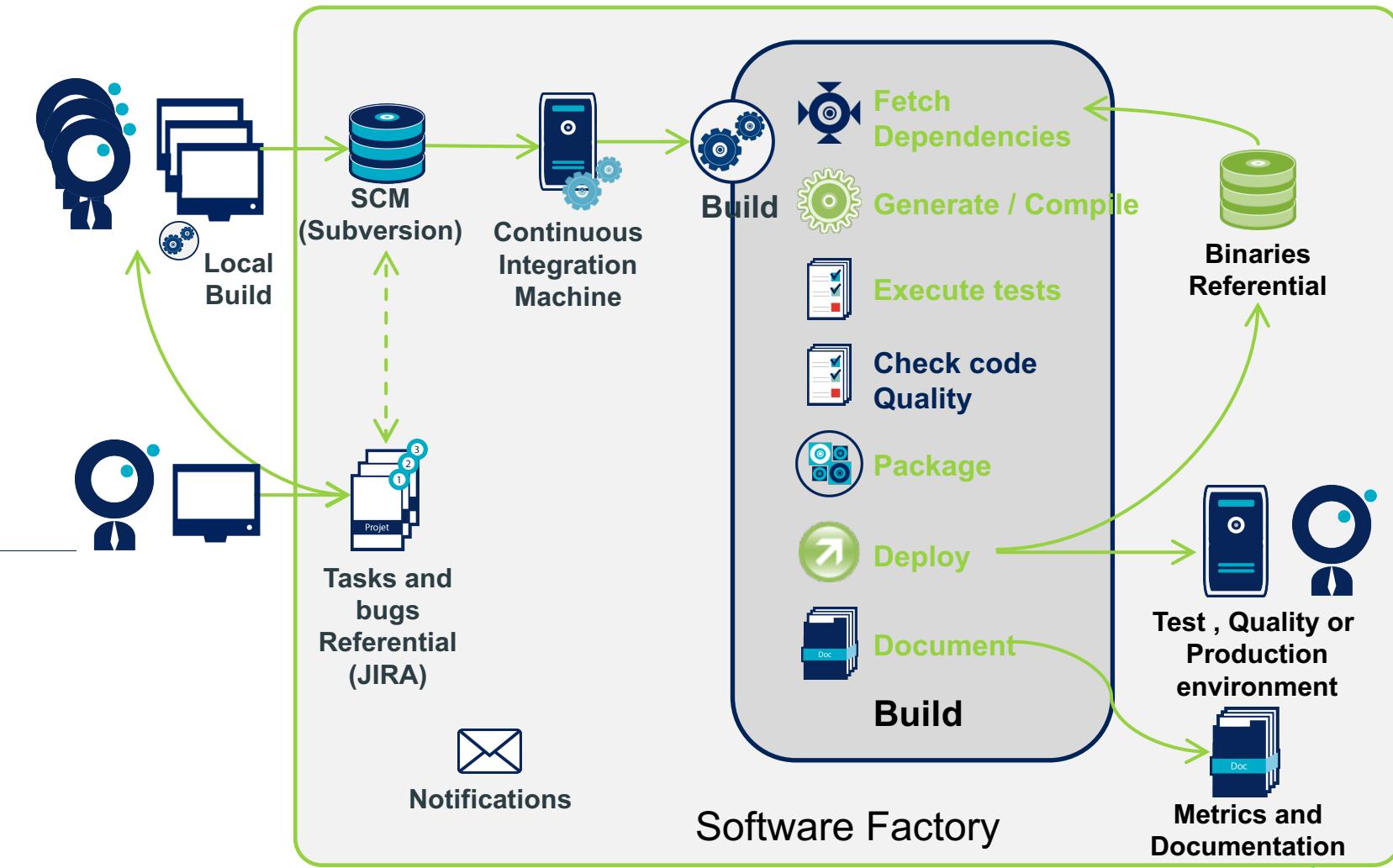


Git, a protocol and a versatile tooling



- ▶ Git protocol permits to synchronize your work with other developers
- ▶ Some actions can be triggered during some commands (hook)
- ▶ You can use forges (GitLab, GitHub, Bitbucket) as central repositories
- ▶ You can even deploy an application with Git (have a look at Heroku)

Continuous integration



Alias

[alias]

st = status

stp = status --porcelain

ci = commit

br = branch

co = checkout

rz = reset --hard HEAD

pullr = pull --rebase

unstage = reset HEAD

lol = log --graph --decorate --pretty=oneline --abbrev-commit (log mieux présenté)
lpush = "!git --no-pager log origin/\$(git currentbranch)..HEAD --oneline » (ce qu'il reste à pusher)

lpull = "!git --no-pager log HEAD..origin/\$(git currentbranch) --oneline » (ce qui va venir au prochain pull)

whatsnew = "!git diff origin/\$(git currentbranch)...HEAD »

whatscoming = "!git diff HEAD...origin/\$(git currentbranch)"

currentbranch = "!git branch | grep \"^*\" | cut -d \" \" -f 2"

Conclusion

- ▶ Git is complex but Git is good
- ▶ Do not learn commands but understand graph manipulation
 - ▶ <http://learngitbranching.js.com>
 - ▶ <http://gitimmersion.com>
- ▶ Git commits are easy and fast → Commit frequently (and deal with it later)
- ▶ Git branches are cheap → Use as many as you need!
- ▶ Git asks less manual merges → The merge engine is powerful
- ▶ Git is standalone → You don't need a server to work
- ▶ Questions ?