



Stanislav Böhm

s.bohm@kreatrix.org

github.com/spirali

Solutions that might fix the problem without breaking anything

Essential

Hoping This
Works

O RLY?

@ThePracticalDev

1. Introduction

2. Hands-on:

- Basic operations
- Branching / Merge / Rebase
- Remote repository

3. Common scenarios

4. Repository management

5. *Bonus topics*

1. Introduction

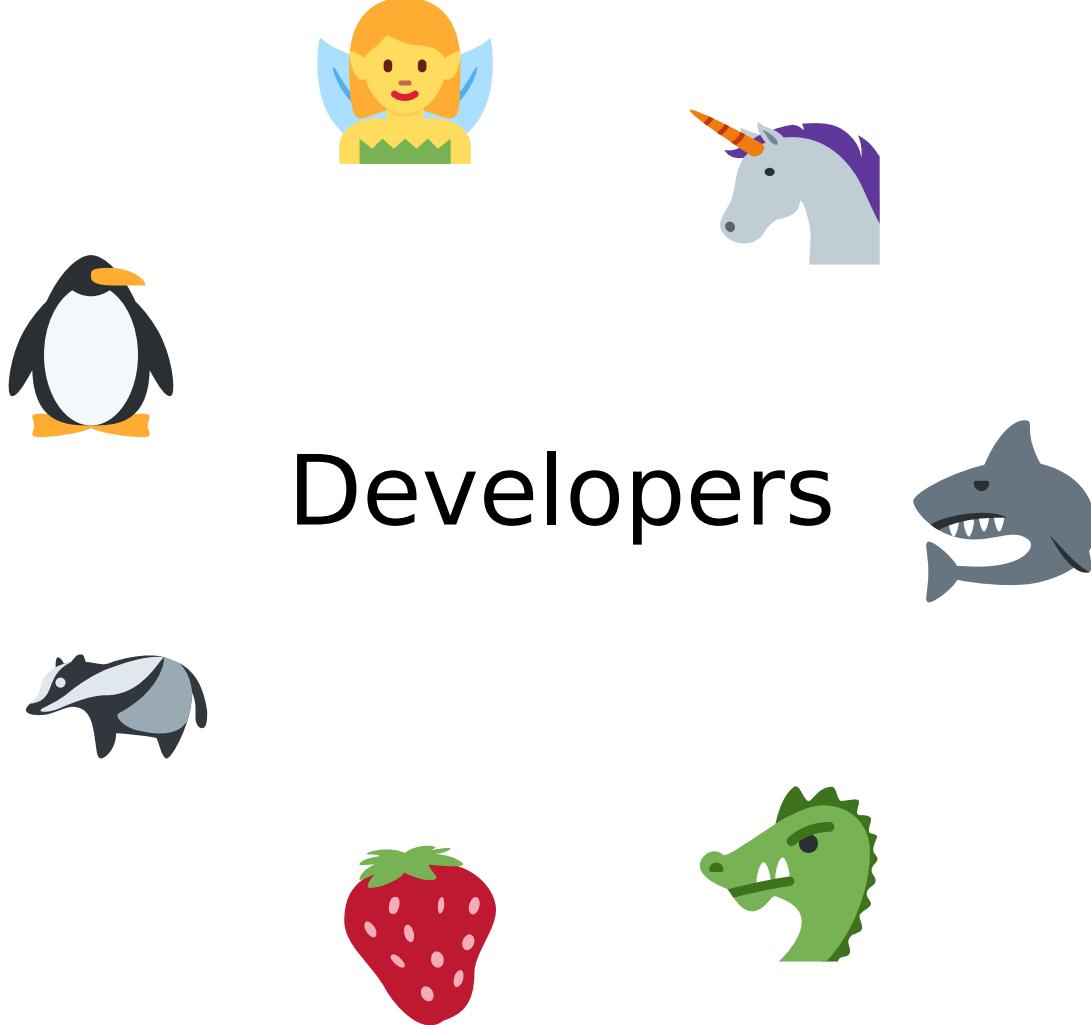
2. Hands-on:

- Basic operations
- Branching / Merge / Rebase
- Remote repository

3. Common scenarios

4. Repository management

5. *Bonus topics*



twemoji.twitter.com

Version Control System

A system used to manage changes to files and track the history of those changes.

Allows multiple users to work on the same codebase simultaneously.

Provides a history of changes, allowing users to revert to previous versions if needed.

Facilitates collaboration between multiple developers working on the same project.

Enables tracking of specific changes made by individual users or teams.

Supports branching and merging of code, allowing for parallel development paths.

Provides automated build and deployment capabilities for continuous integration.

Enables tracking of dependencies and external libraries used in the project.

Facilitates code reviews and provides a platform for discussing changes and improvements.

Provides reporting and analytics to track the performance and usage of the codebase.

Enables tracking of file revisions and changes made to specific lines of code.

Provides a centralized location for storing and managing all project files and documentation.

Enables tracking of file revisions and changes made to specific lines of code.

Provides a centralized location for storing and managing all project files and documentation.

Enables tracking of file revisions and changes made to specific lines of code.

Provides a centralized location for storing and managing all project files and documentation.

Enables tracking of file revisions and changes made to specific lines of code.

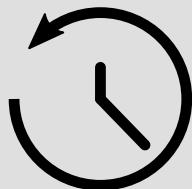
Provides a centralized location for storing and managing all project files and documentation.

Enables tracking of file revisions and changes made to specific lines of code.

Provides a centralized location for storing and managing all project files and documentation.

Version Control System

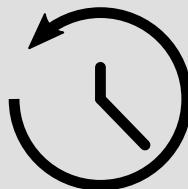
Change Tracking



```
--- print("Hello")
+++ print("Hello world!")
```

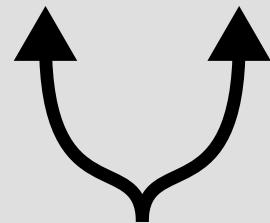
Version Control System

Change Tracking



```
--- print("Hello")
+++ print("Hello world!")
```

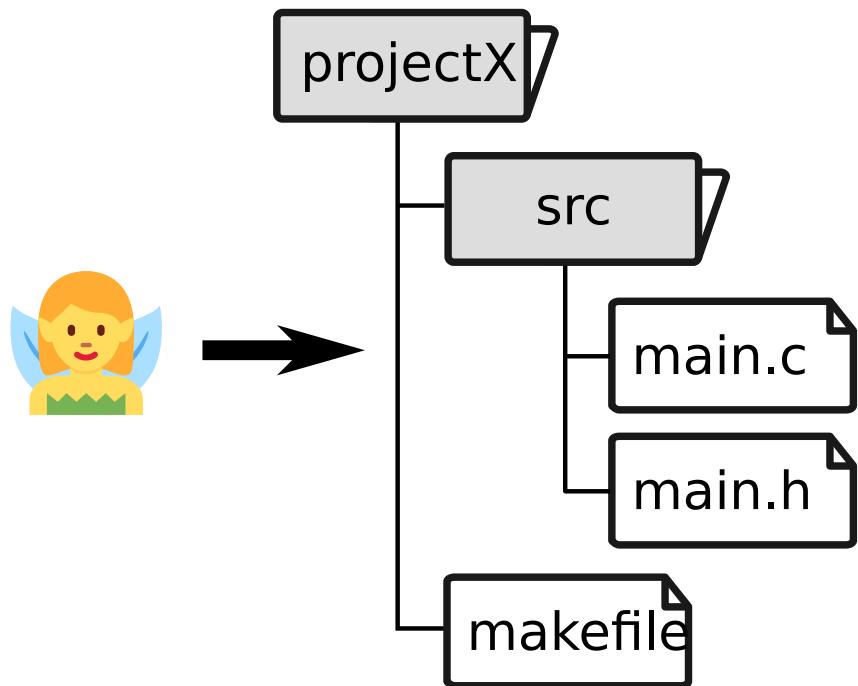
Concurrent Development



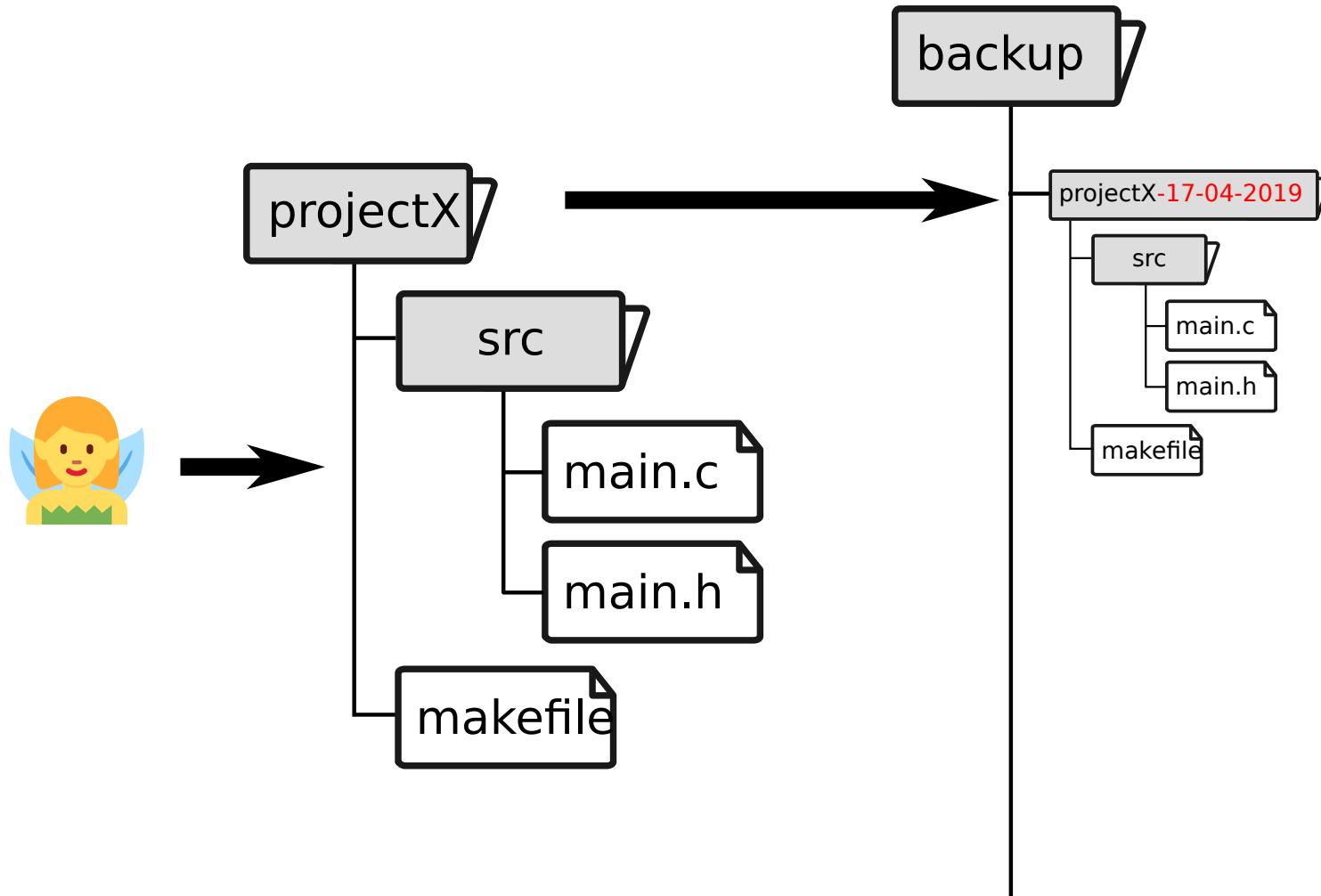
Manual VCS

Manual VCS

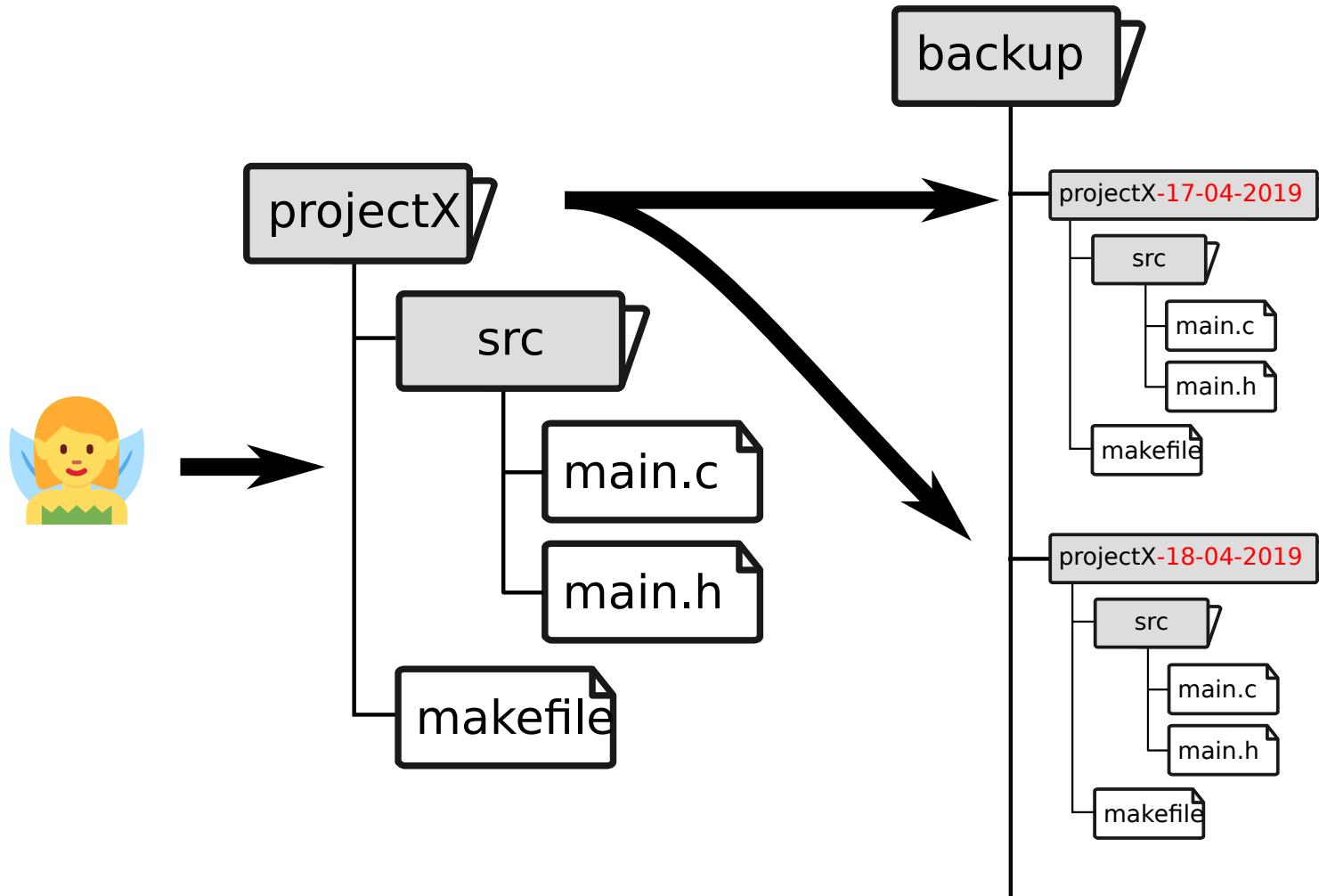




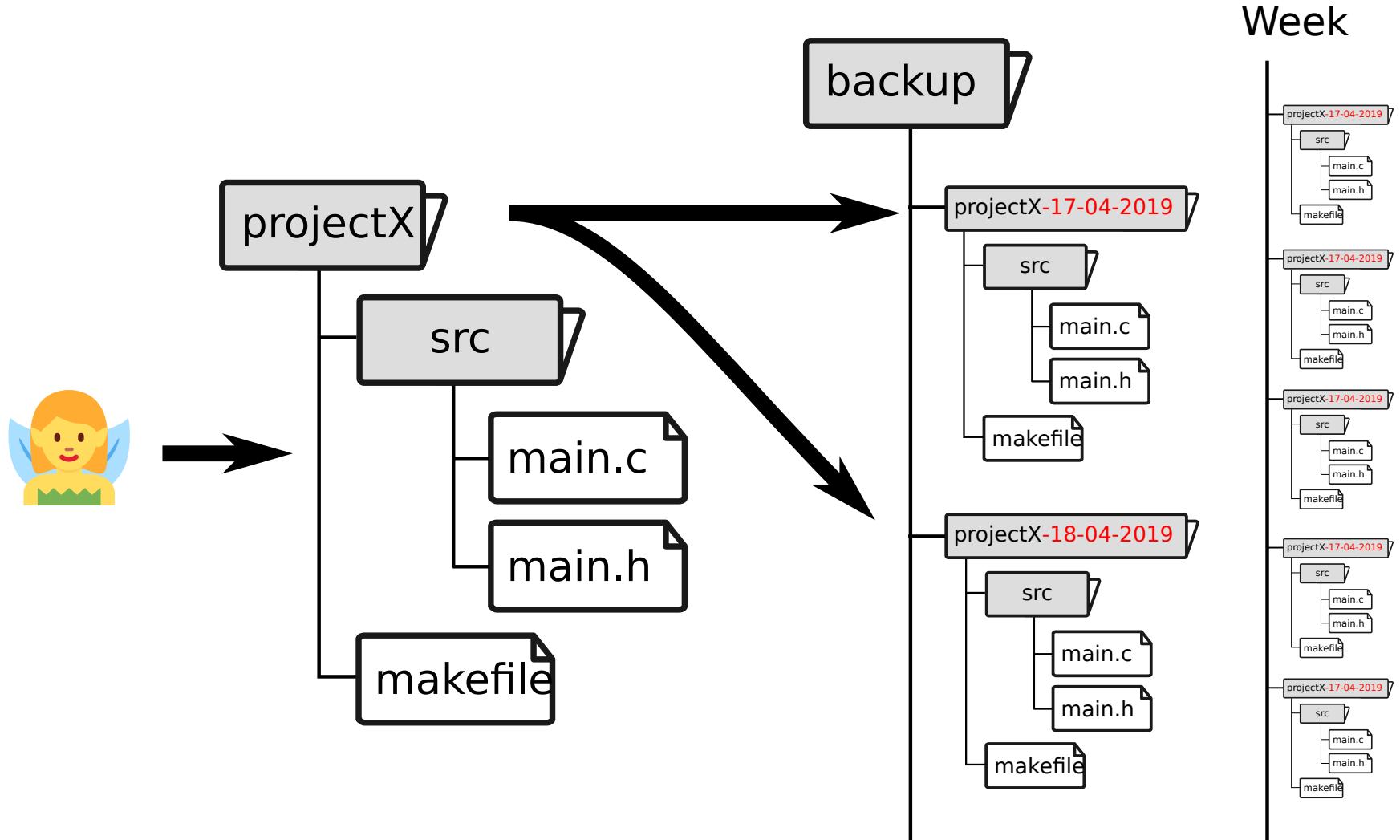
Manual VCS



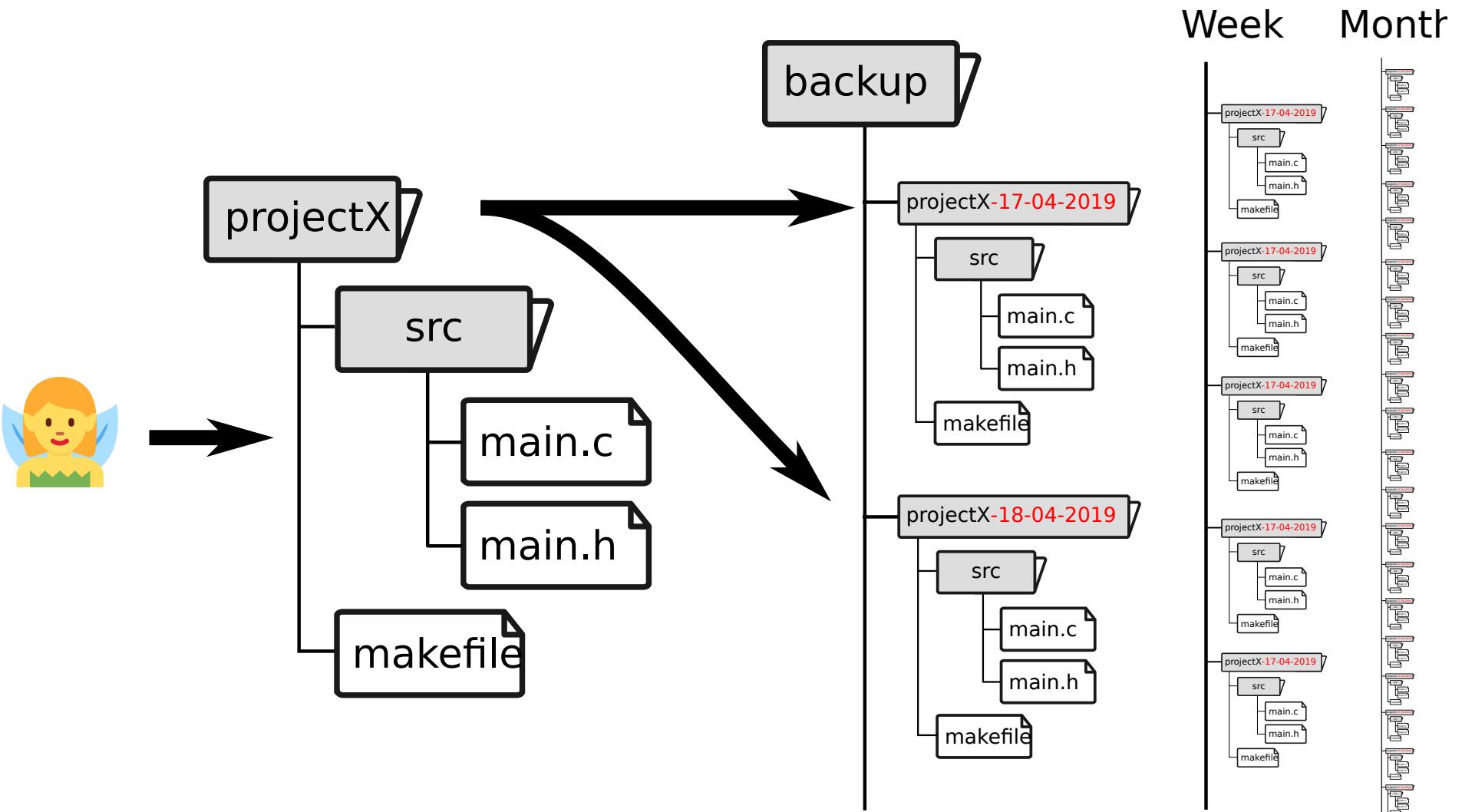
Manual VCS



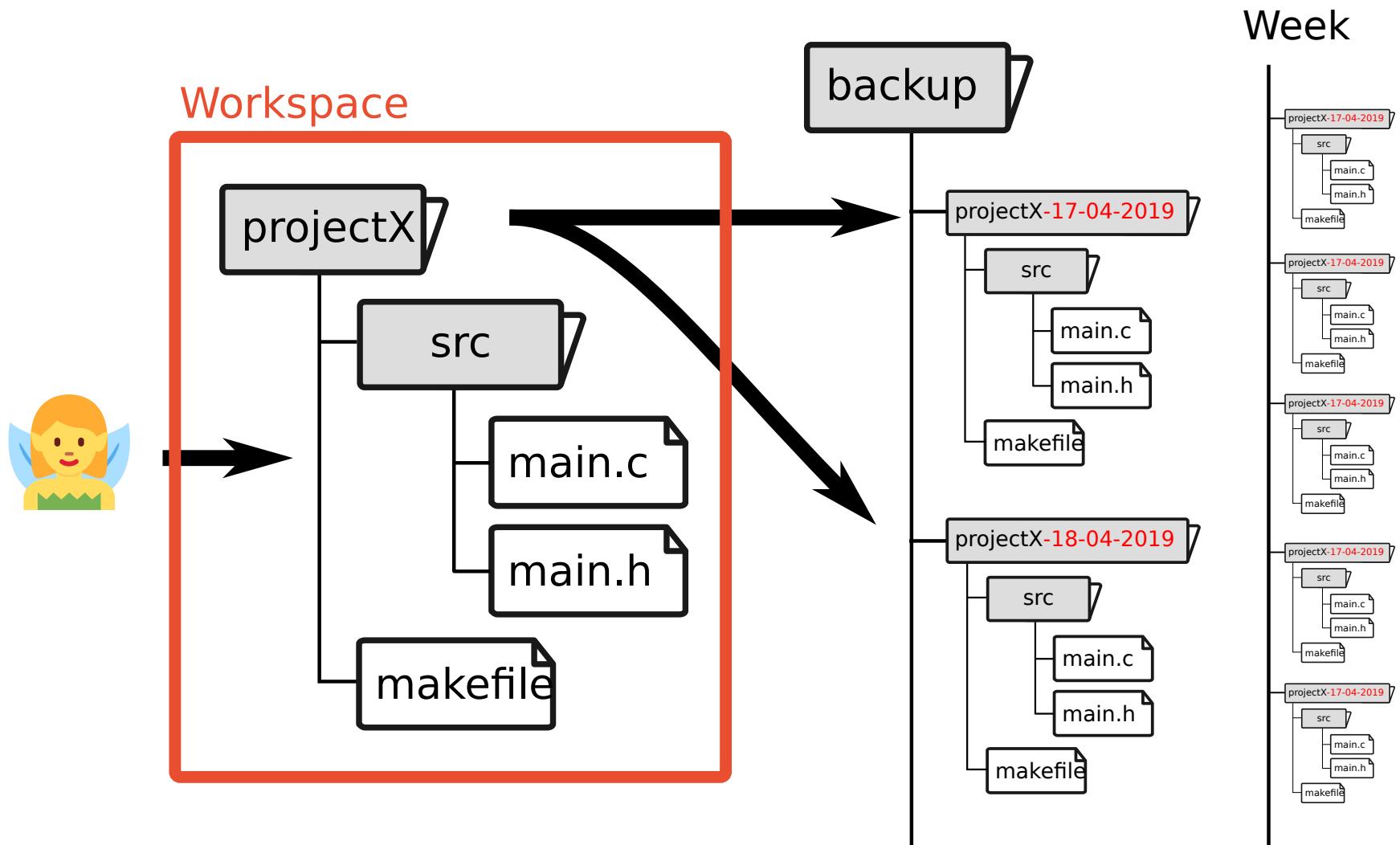
Manual VCS



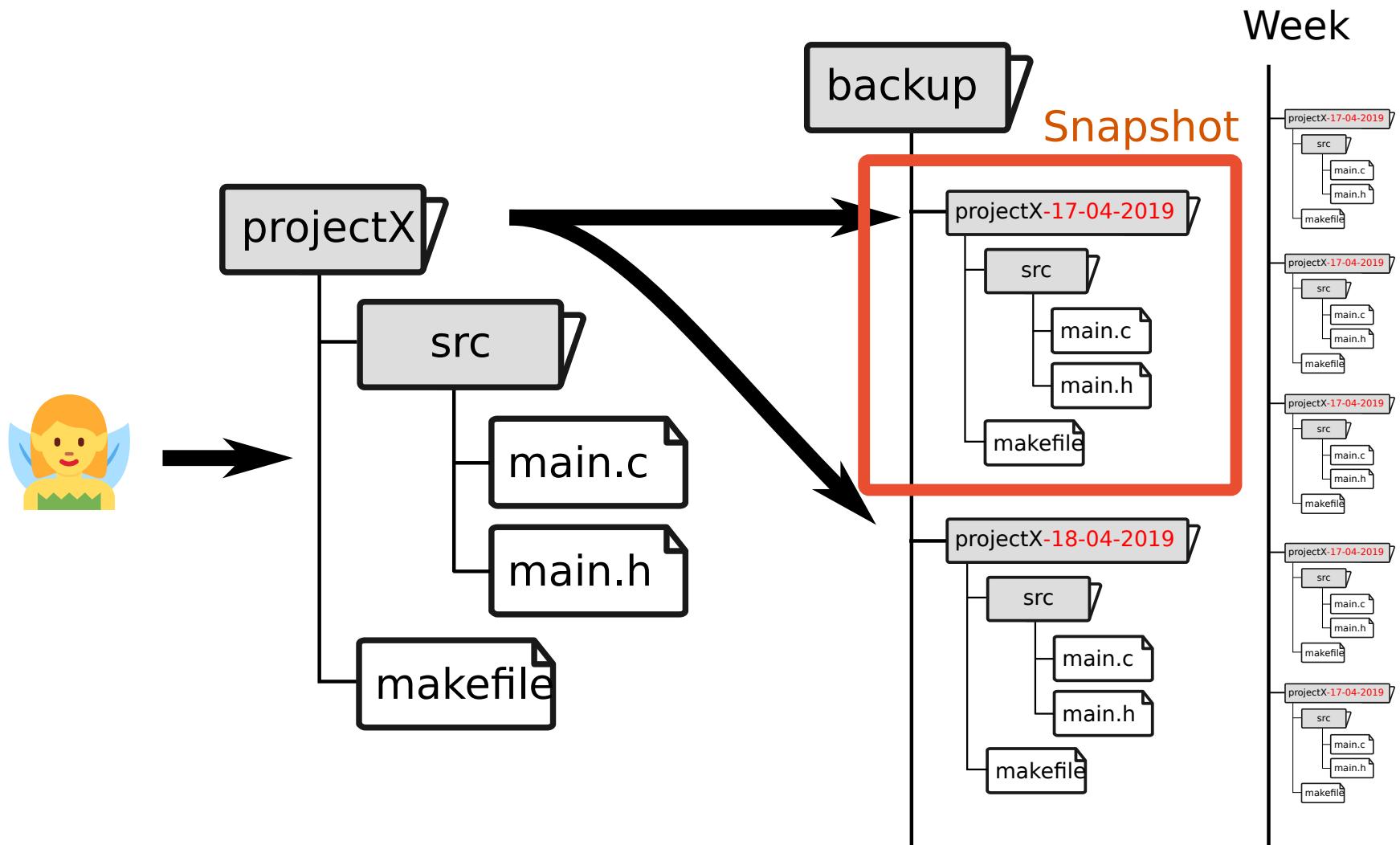
Manual VCS



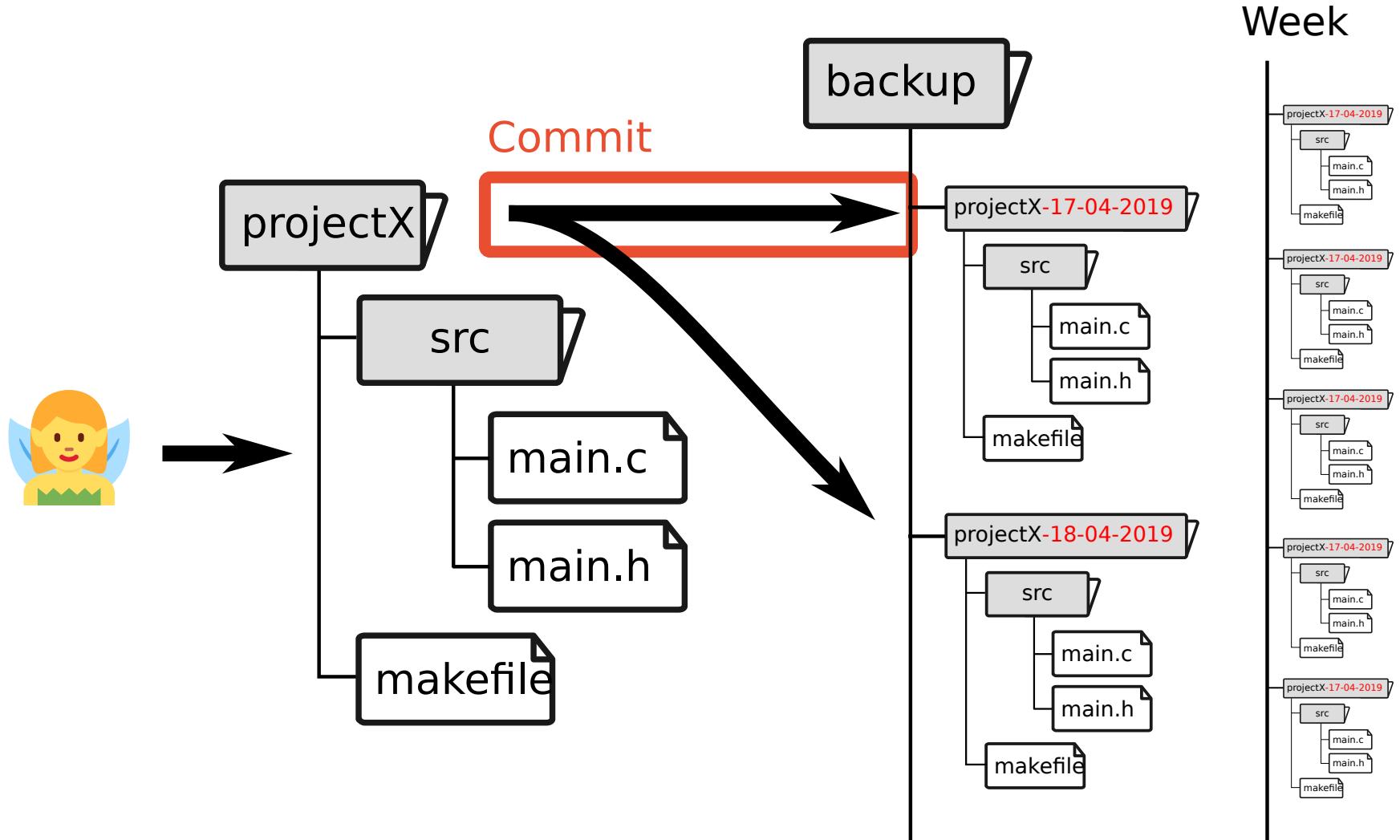
Manual VCS



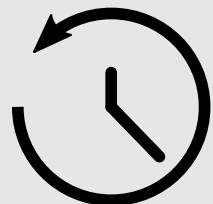
Manual VCS



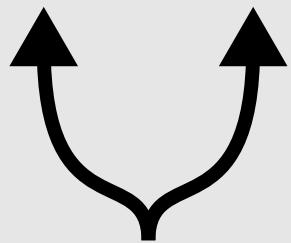
Manual VCS



Manual VCS

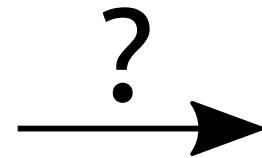


```
--- print("Hello")  
+++ print("Hello world!")
```



Tracking Changes

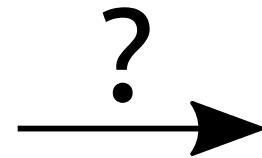
```
def main():
    print("Hello world!\n")
    print("-----")
```



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

Tracking Changes

```
def main():
    print("Hello world!\n")
    print("-----")
```



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

character changes



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

Tracking Changes

```
def main():
    print("Hello world!\n")
    print("-----")
```



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

character changes



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

Tracking Changes

```
def main():
    print("Hello world!\n")
    print("-----")
```



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

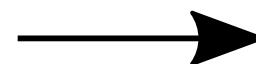
character changes



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

line changes



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

Tracking Changes

```
def main():
    print("Hello world!\n")
    print("-----")
```



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

character changes



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

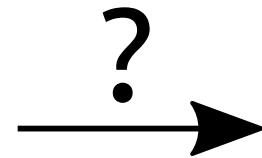
line changes



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

Tracking Changes

```
def main():
    print("Hello world!\n")
    print("-----")
```



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

character changes



```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print("Hello world!\n")
    print("-----")
```

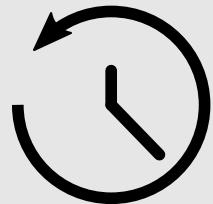
line changes



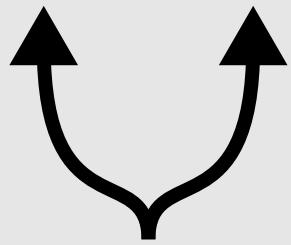
```
def main():
--- print("Hello world!\n")
+++ print_name = "world"
+++ print("Hello f{print_name}!\n")
    print("-----")
```

```
def main():
    print_name = "world"
    print("Hello f{print_name}!\n")
    print("-----")
```

Manual VCS + Diff



```
--- print("Hello")
+++ print("Hello world!")
```



Centralized

CSV

Subversion

ClearCase

[Perforce]

[TFS]

Distributed

Bitkeeper

Arch

Monotone

Darcs

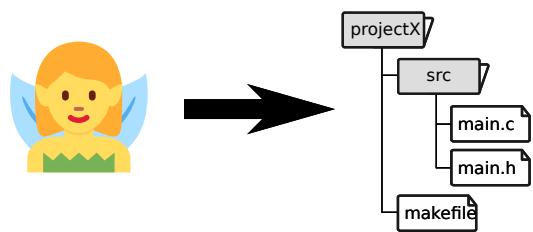
Bazaar

Mercurial

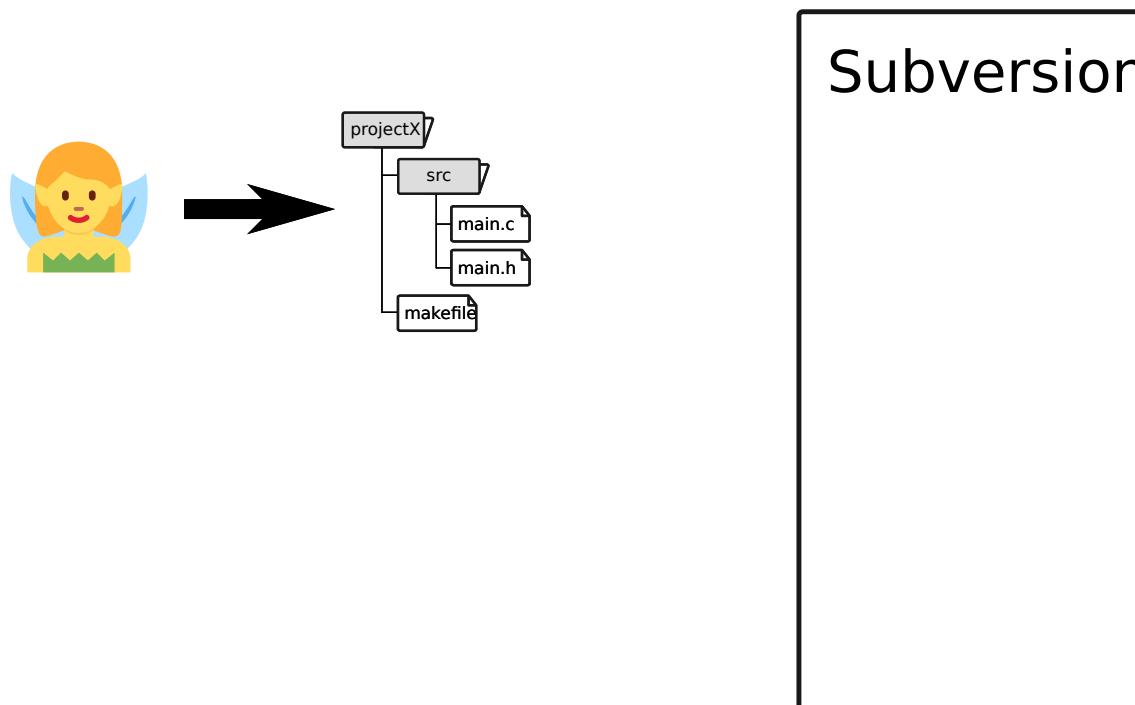
Git

Pijul

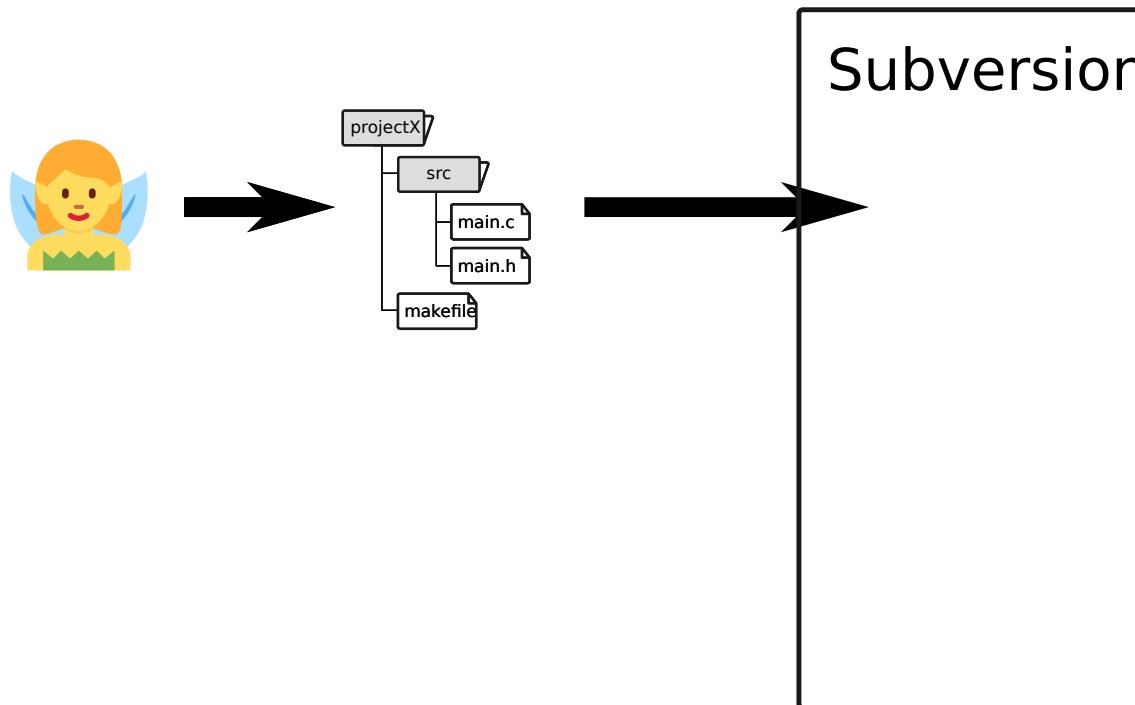
Subversion



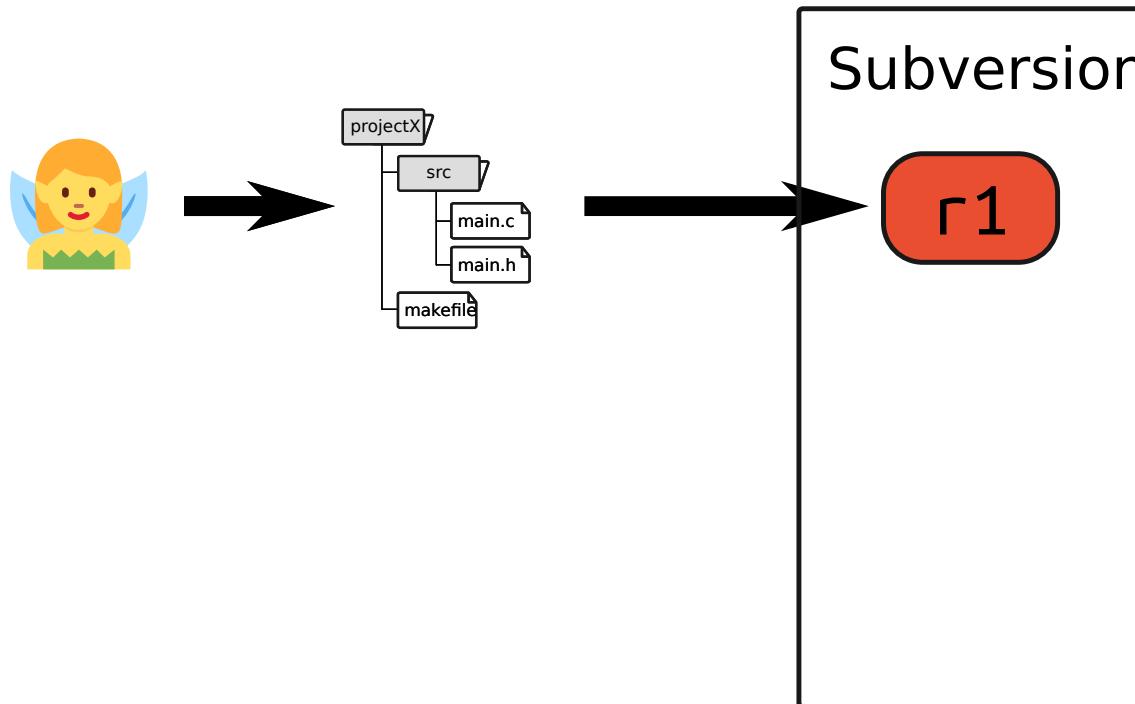
Subversion



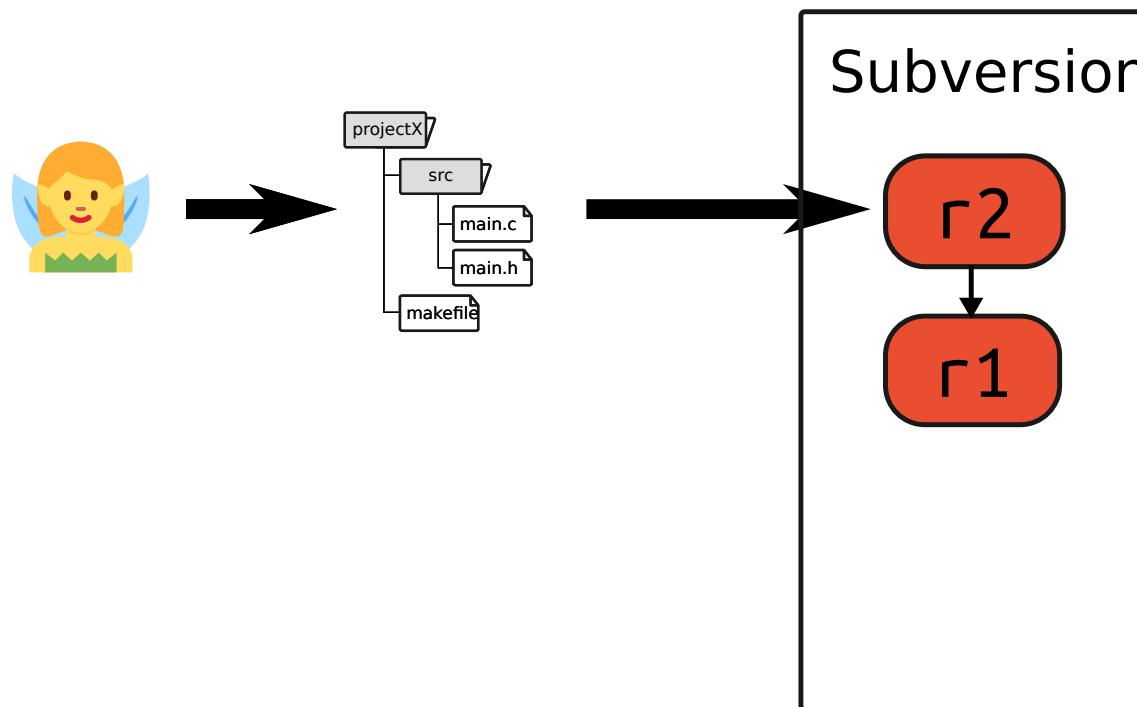
Subversion



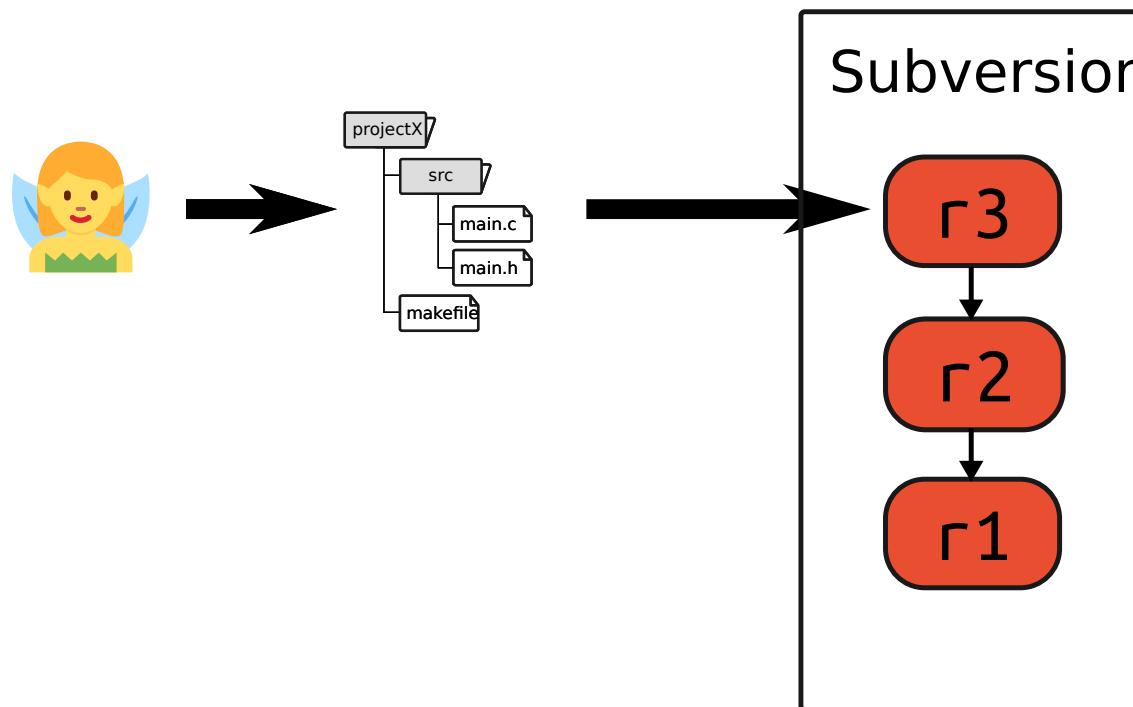
Subversion



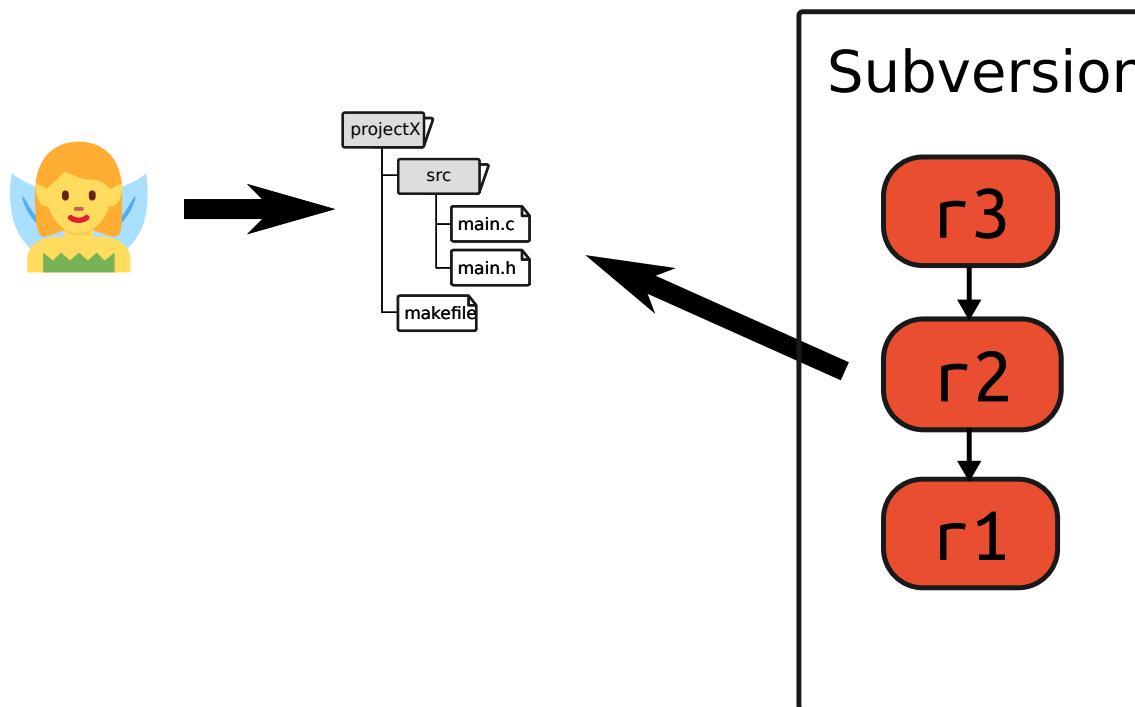
Subversion



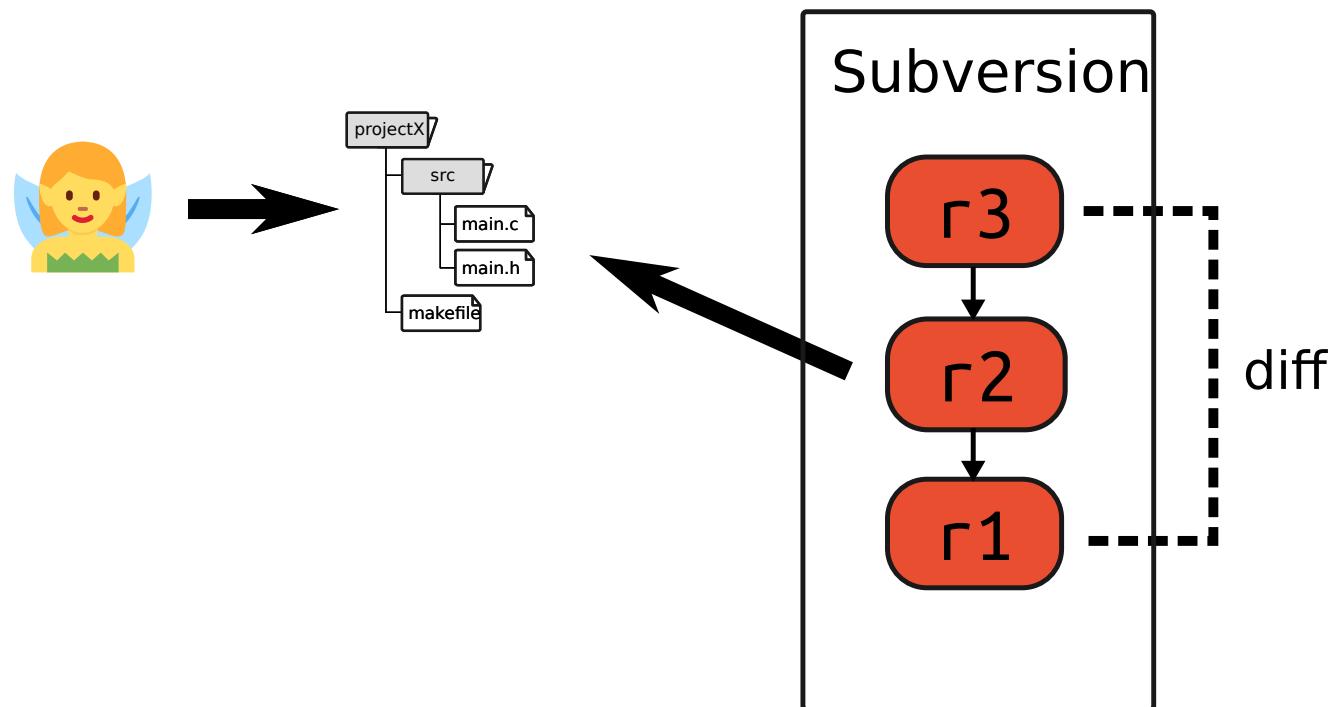
Subversion



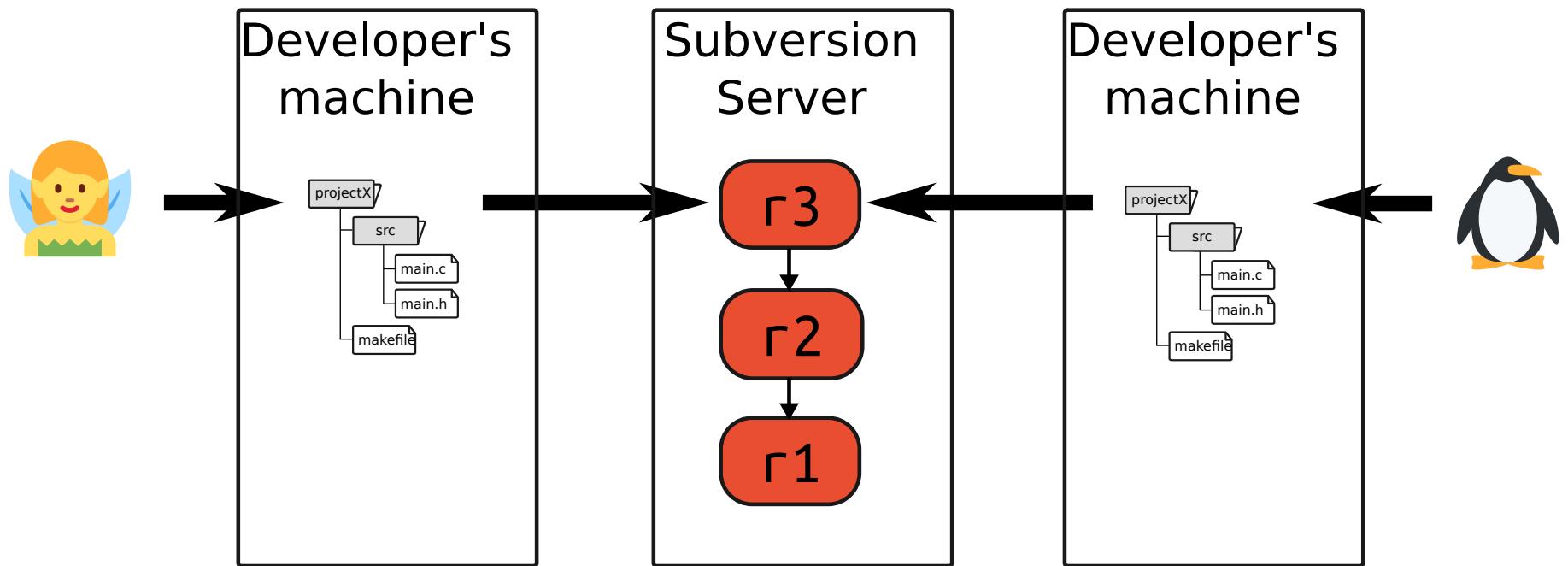
Subversion



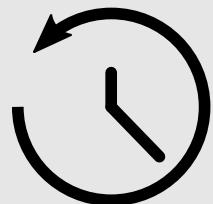
Subversion



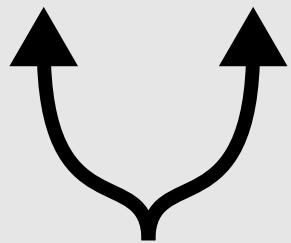
Subversion



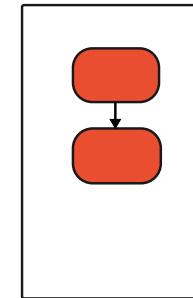
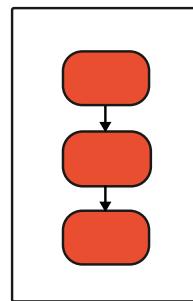
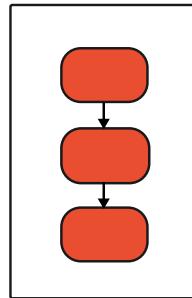
Subversion



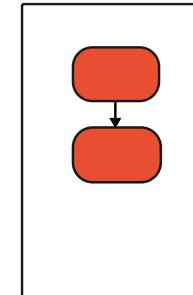
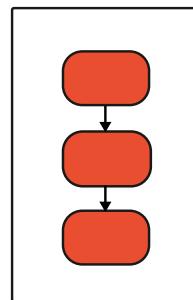
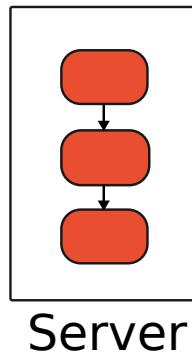
```
--- print("Hello")
+++ print("Hello world!")
```



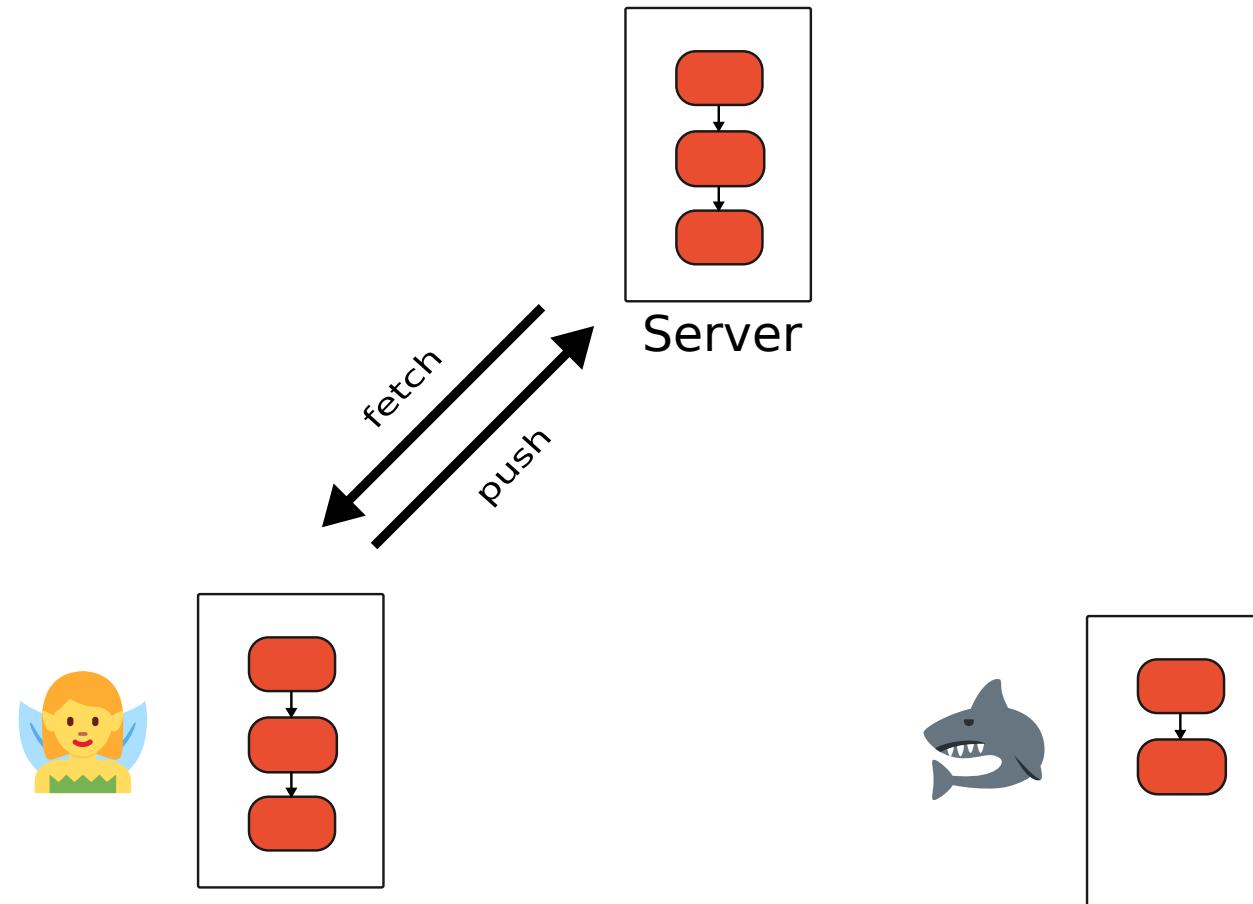
Distributed VCS



Distributed VCS



Distributed VCS



Git

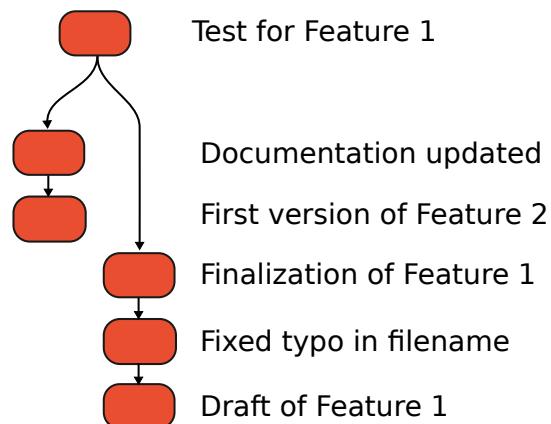
- Distributed VCS

- Distributed VCS
- First-class branching support

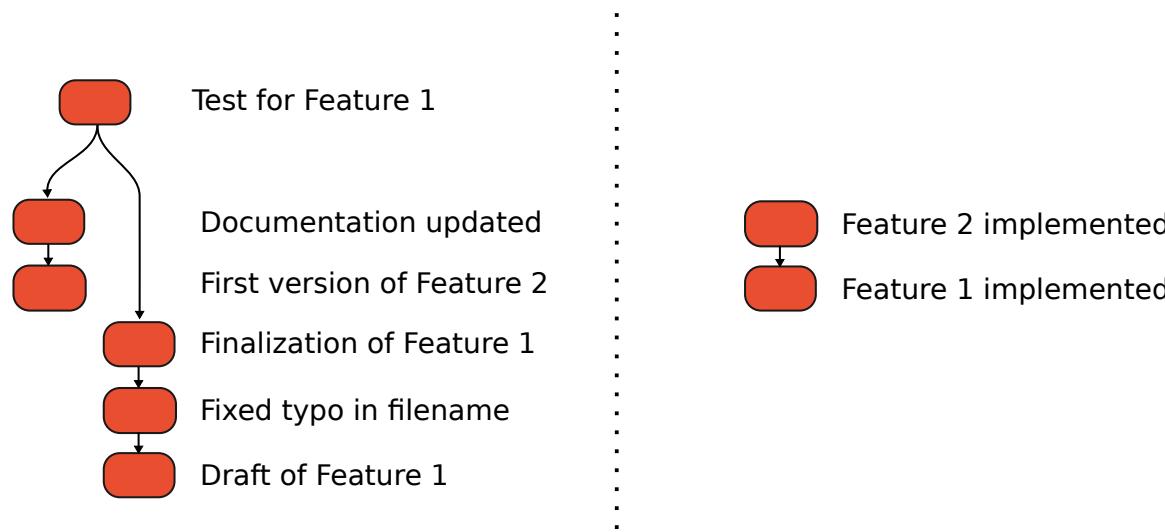
- Distributed VCS
- First-class branching support
- Created by Linus Torvalds in 2005

- Distributed VCS
- First-class branching support
- Created by Linus Torvalds in 2005
- Rewriting history

- Distributed VCS
- First-class branching support
- Created by Linus Torvalds in 2005
- Rewriting history



- Distributed VCS
- First-class branching support
- Created by Linus Torvalds in 2005
- Rewriting history



What to learn?

Graph DB

Versioning Concepts

Graph DB

Commands | GUI | IDE

Versioning Concepts

Graph DB

What to learn?

Commands

GUI

IDE

Versioning Concepts

Graph DB

GitSomething

GitHub

GitHub



GitLab

GitHub



gitk

A screenshot of the gitk graphical interface, which displays a timeline of commits for a repository. The commits are listed in chronological order from top to bottom. Each commit entry includes the author's name and email, the date and time of the commit, and a short log message describing the changes made. The interface has a classic windowed look with a menu bar at the top and a search bar below it. The main area shows the commit list and some commit details.

Author	Date	Log Message
Stanislav Bohm <...>	2018-10-29 22:05:35	[server] Migrated from tokio-0c:codet to t
Stanislav Bohm <...>	2018-10-29 19:33:08	[docs] Link to tutorial Part 3
Stanislav Bohm <...>	2018-10-29 19:33:08	[docs] Fix typos in the tutorial part 3
Stanislav Bohm <...>	2018-10-29 17:20:24	[tooling] Execscale deployment script update
Stanislav Bohm <...>	2018-09-05 16:43:08	Releasing version 0.4.0-pre
Stanislav Bohm <...>	2018-09-05 16:24:45	Fix CI script
Stanislav Bohm <...>	2018-08-29 18:34:08	[test] Fix dummy event in tests
Stanislav Bohm <...>	2018-08-28 19:08:15	[dashboards] Dashboard rebuild
Tomas Gavencik <...>	2018-12-16 12:58:03	[server] Migrated from tokio-0c:codet to t
Tomas Gavencik <...>	2018-12-16 12:30:34	Update README and py package readme
Stanislav Bohm <...>	2018-12-15 15:23:10	[fix] style
Stanislav Bohm <...>	2018-12-15 15:23:10	[docs] Removed #!macro_use
Stanislav Bohm <...>	2018-12-15 15:23:10	[remote design] Remote design

1. Introduction

2. Hands-on:

- Basic operations
- Branching / Merge / Rebase
- Remote repository

3. Common scenarios

4. Repository management

5. *Bonus topics*



Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

Staging area

Commit

Branch

Merge

Conflict

Rebase

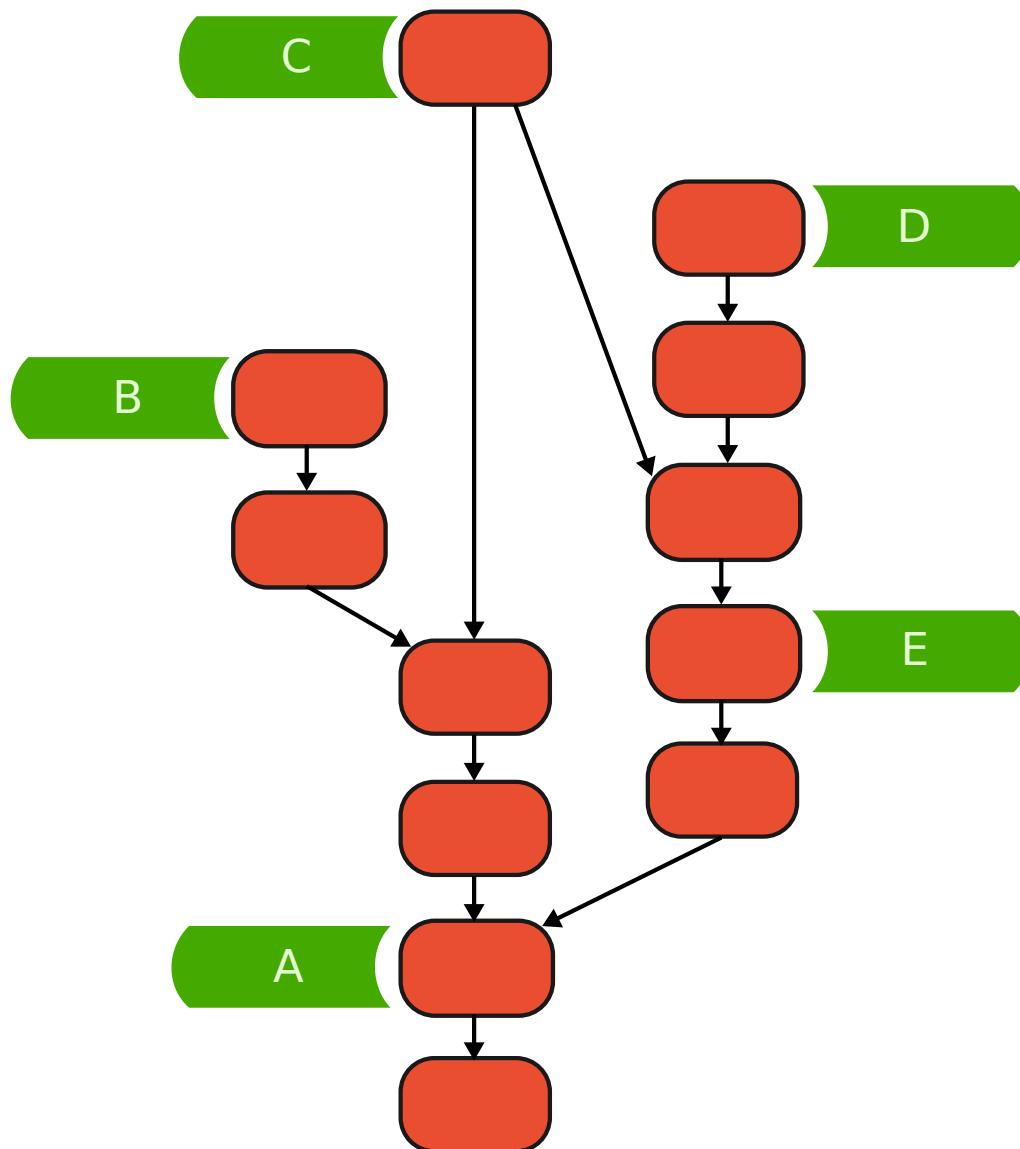
Squash / Reword

Stash

Fetch

Push

Fast-forward



Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

Staging area

Commit

Branch

Merge

Conflict

Rebase

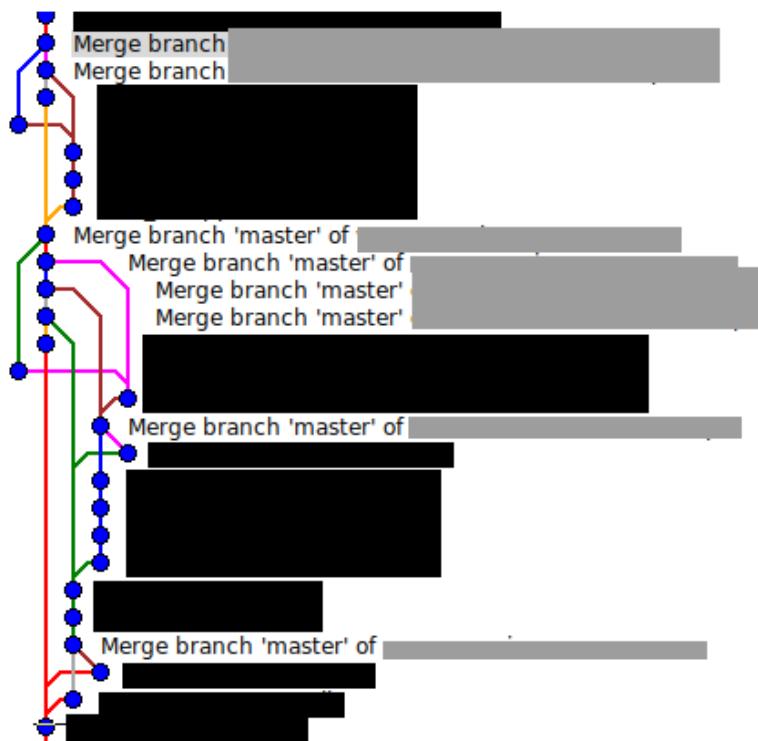
Squash / Reword

Stash

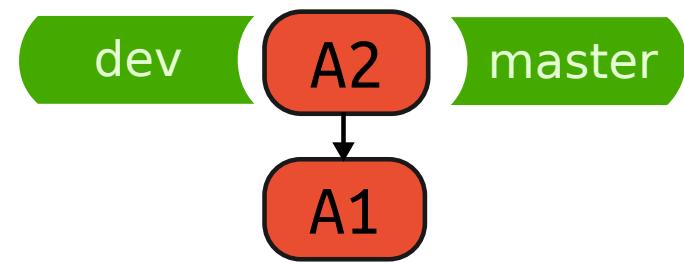
Fetch

Push

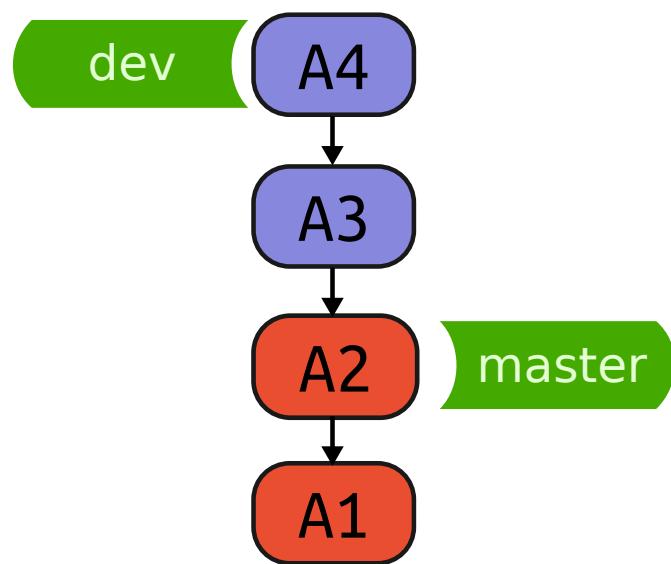
Merging Commits



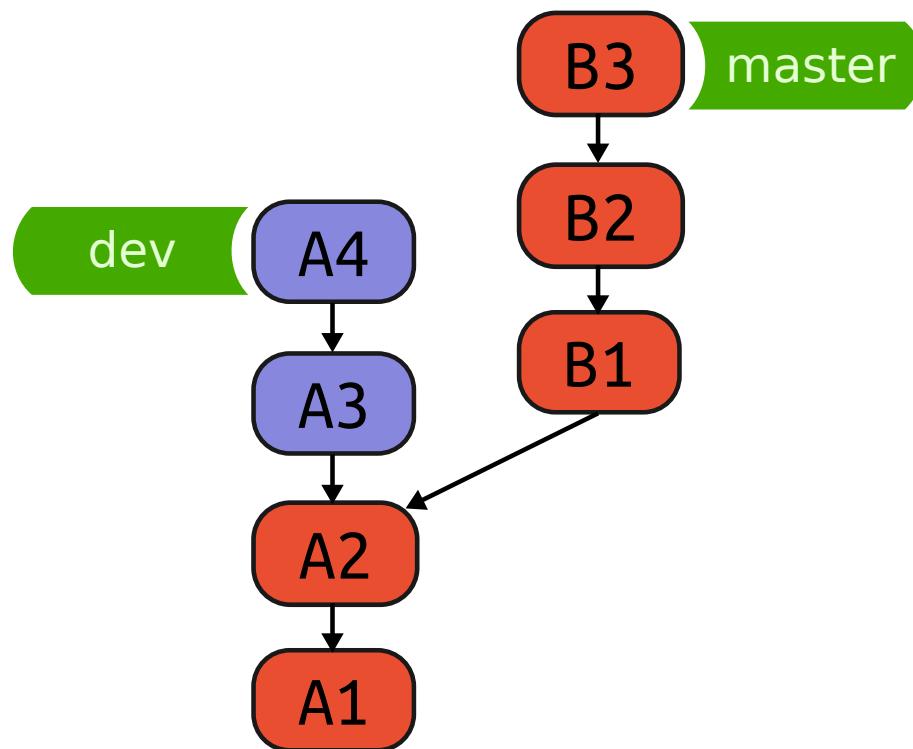
Rebase



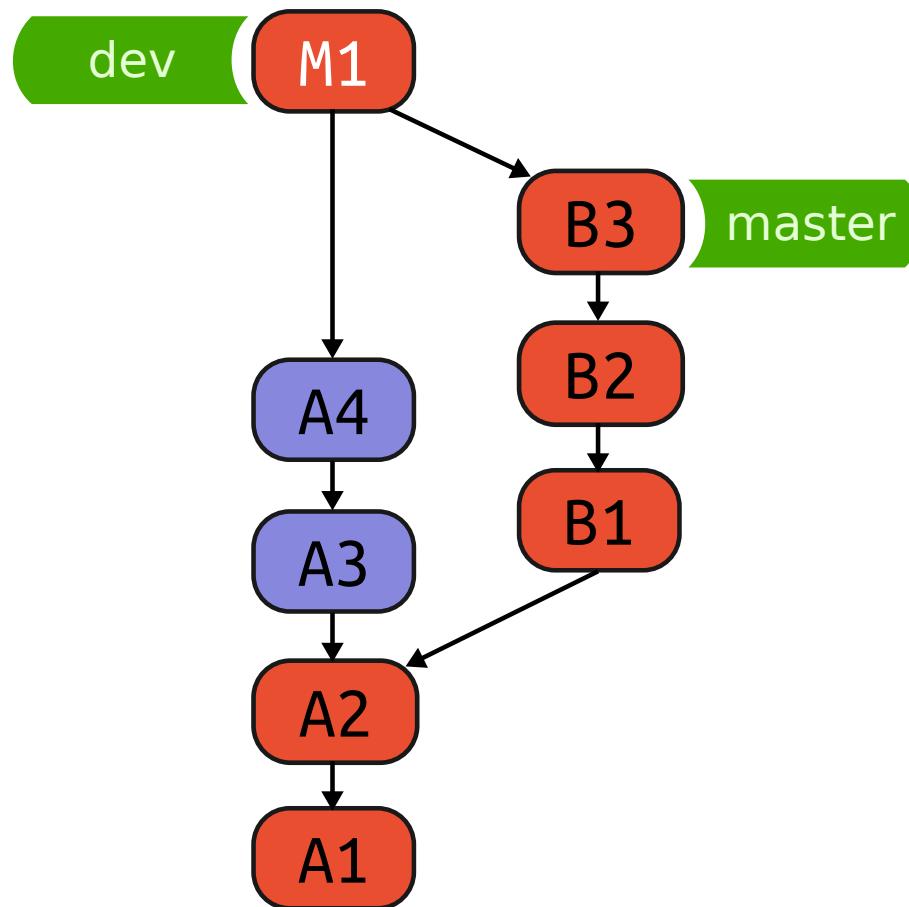
Rebase



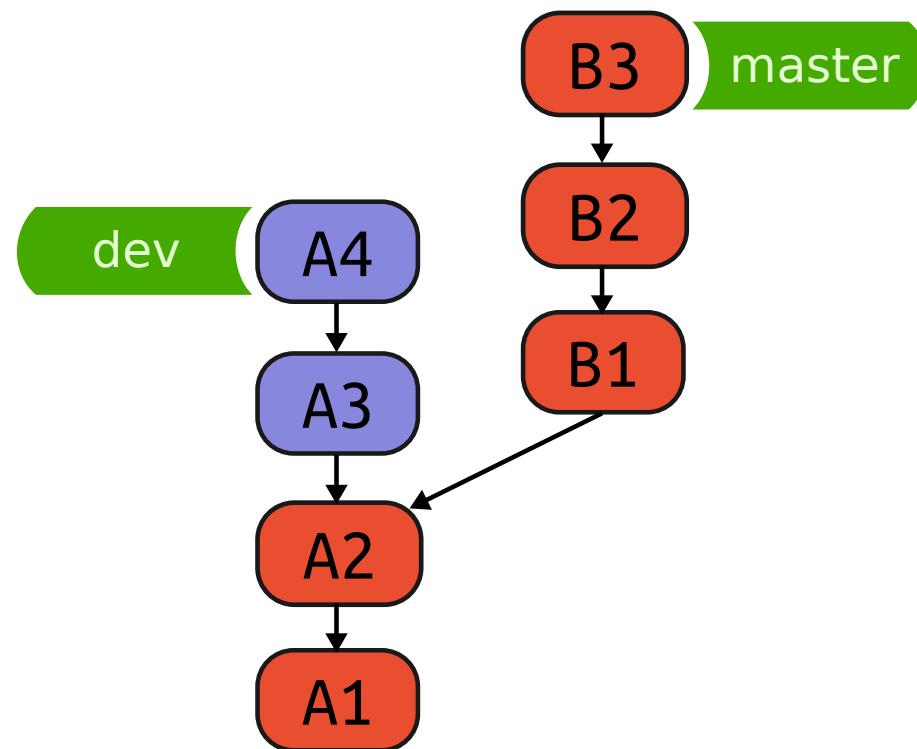
Rebase



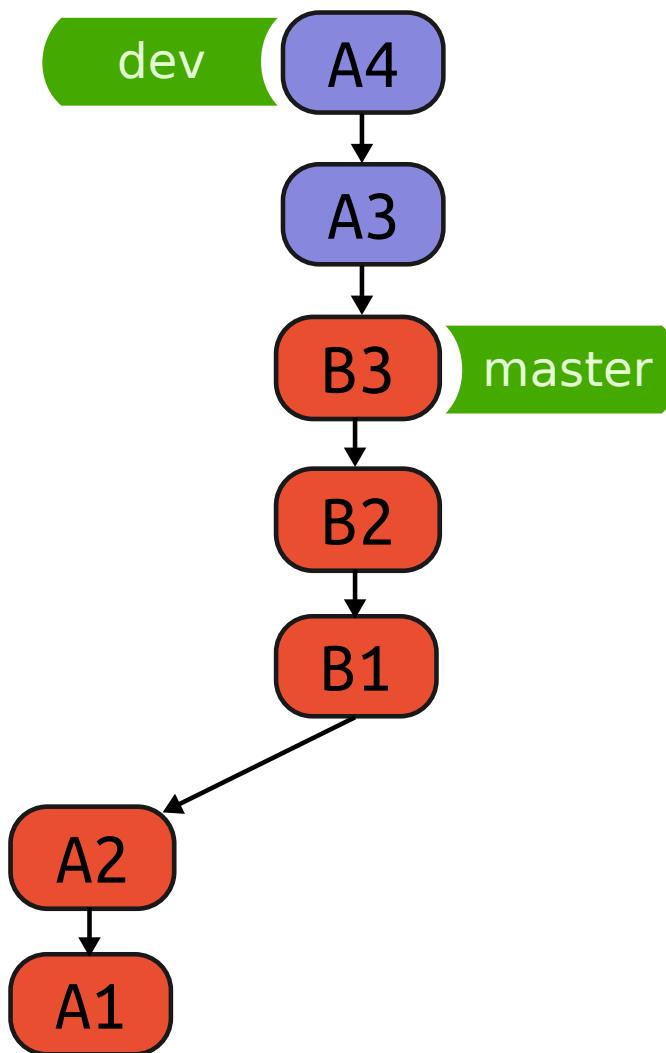
Rebase



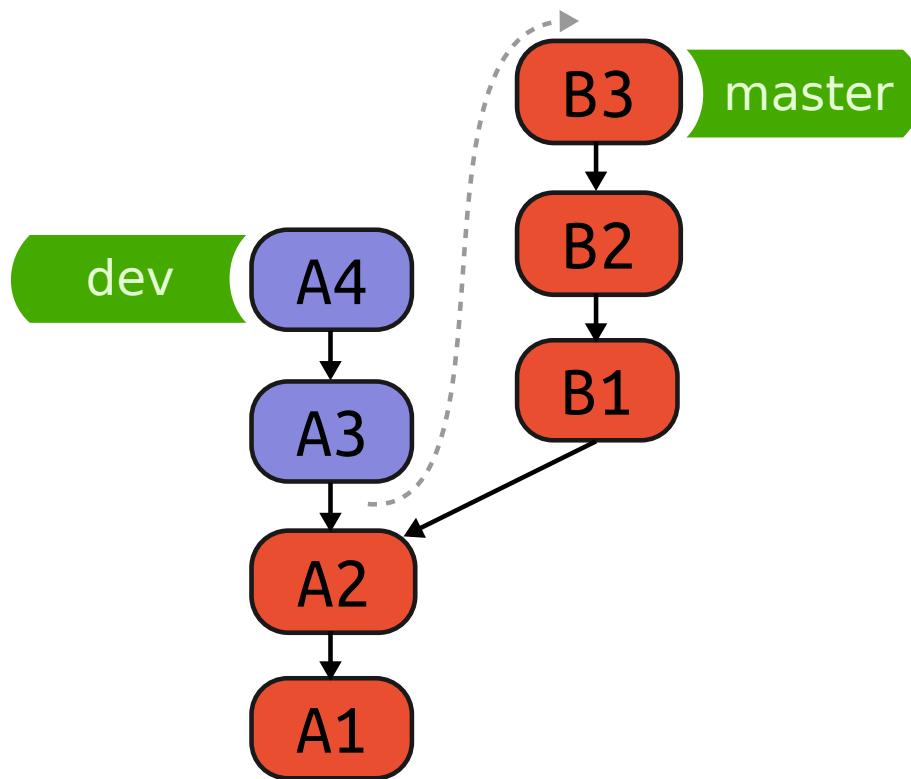
Rebase



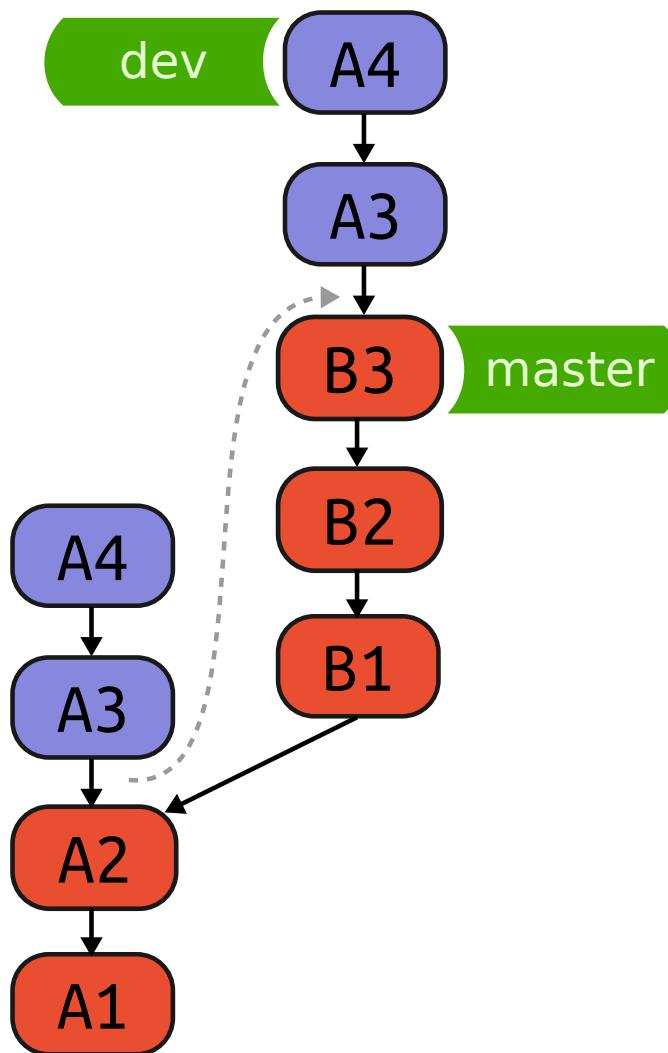
Rebase



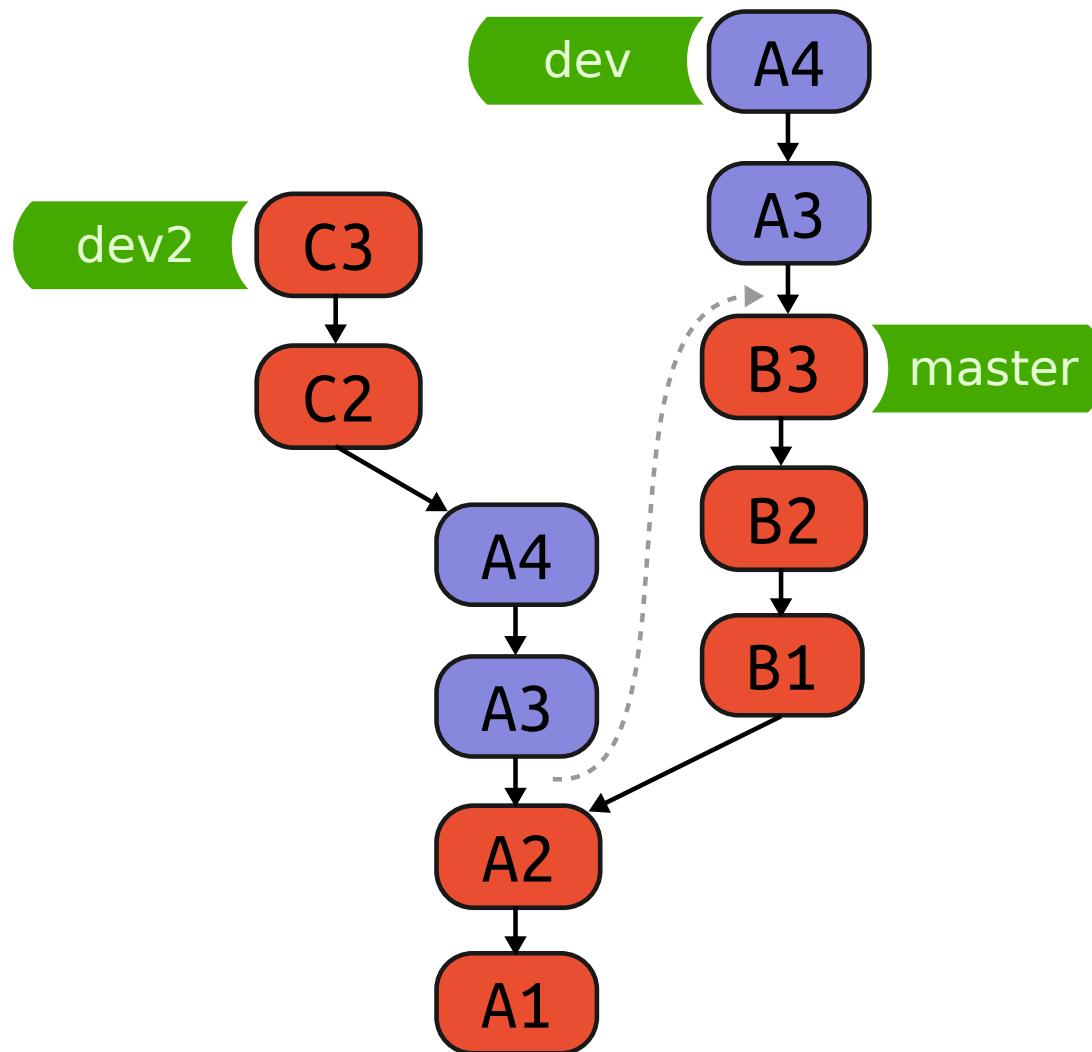
Rebase



Rebase



Rebase



Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

Staging area

Commit

Branch

Merge

Conflict

Rebase

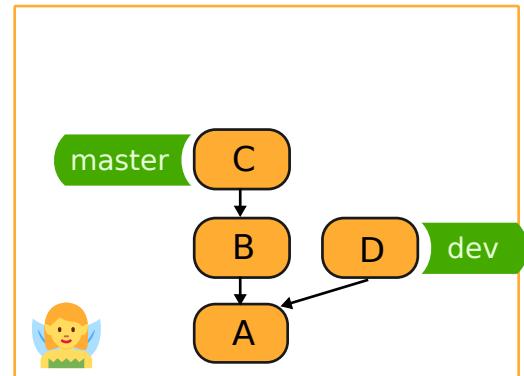
Squash / Reword

Stash

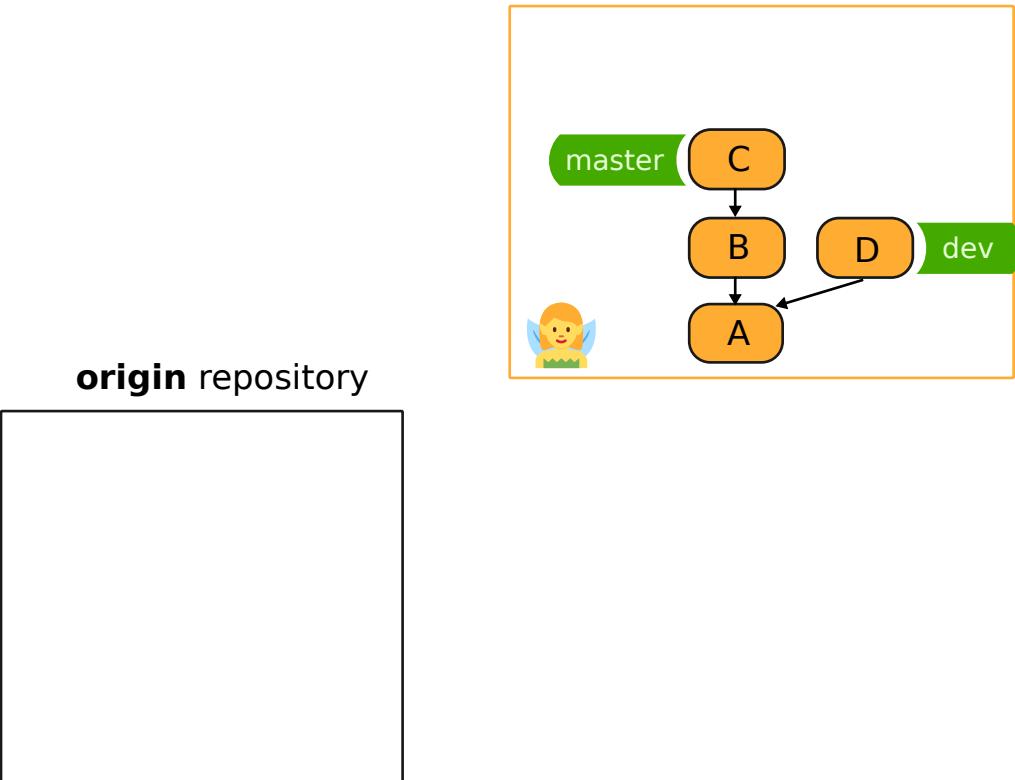
Fetch

Push

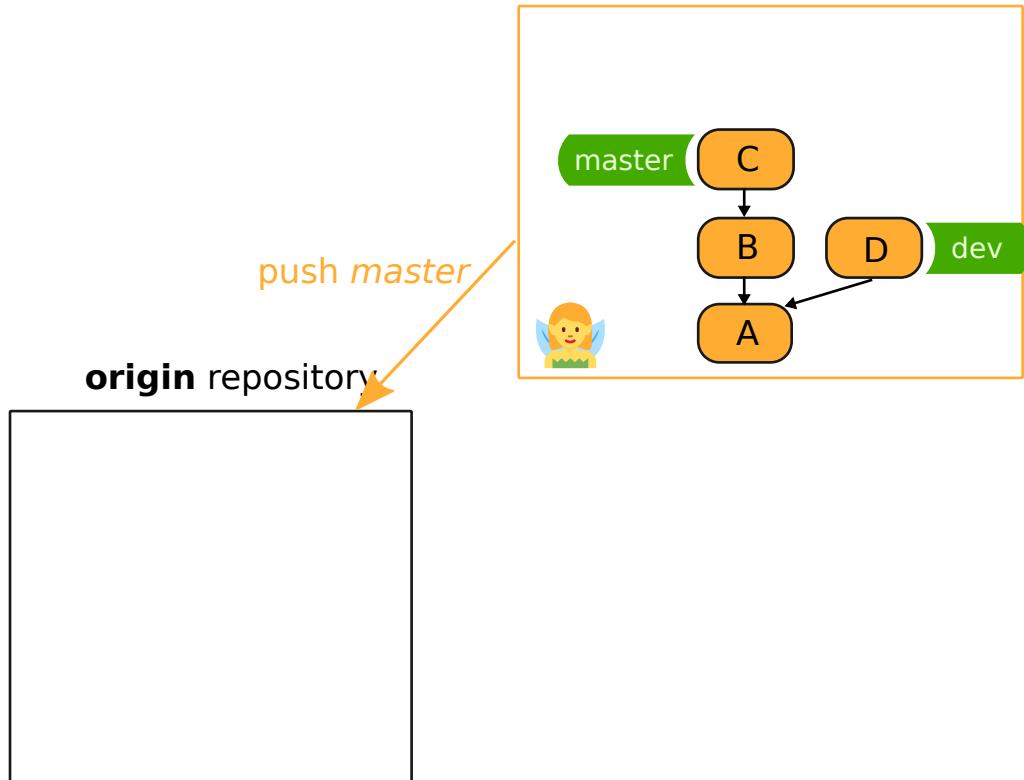
Remotes



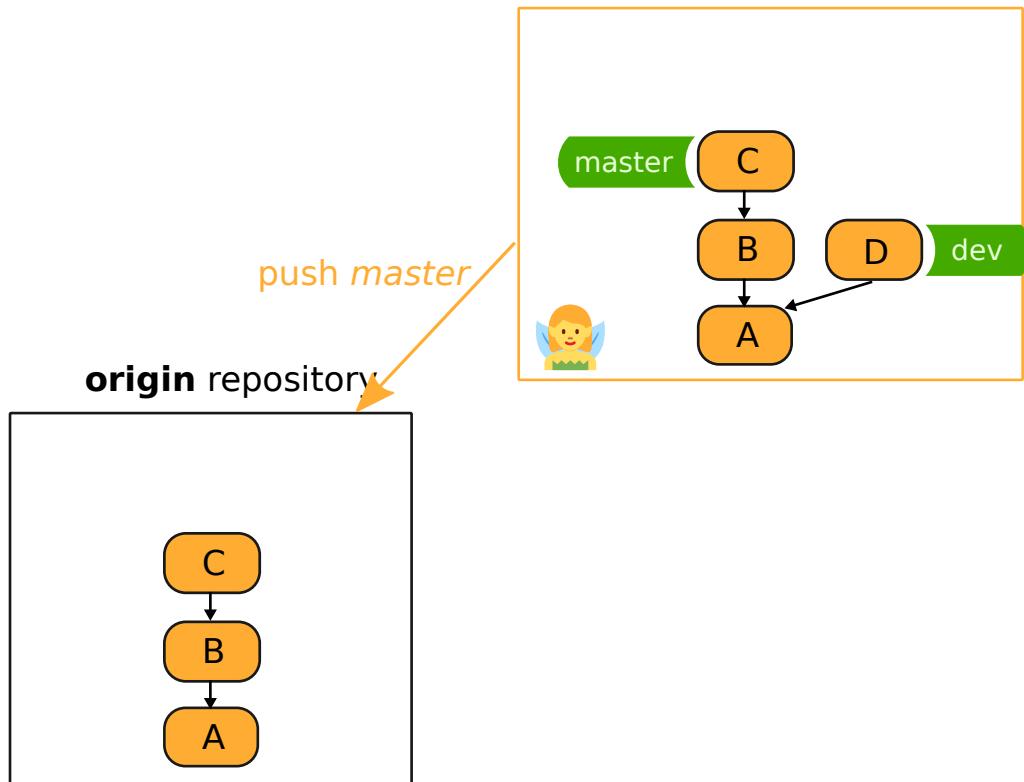
Remotes



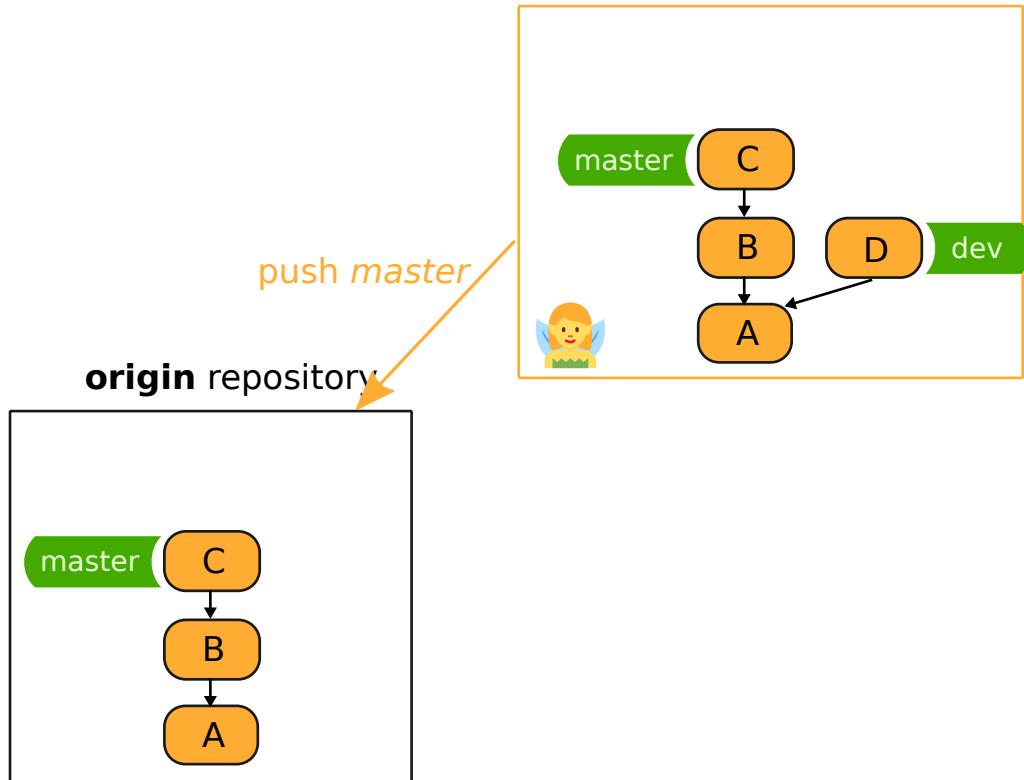
Remotes



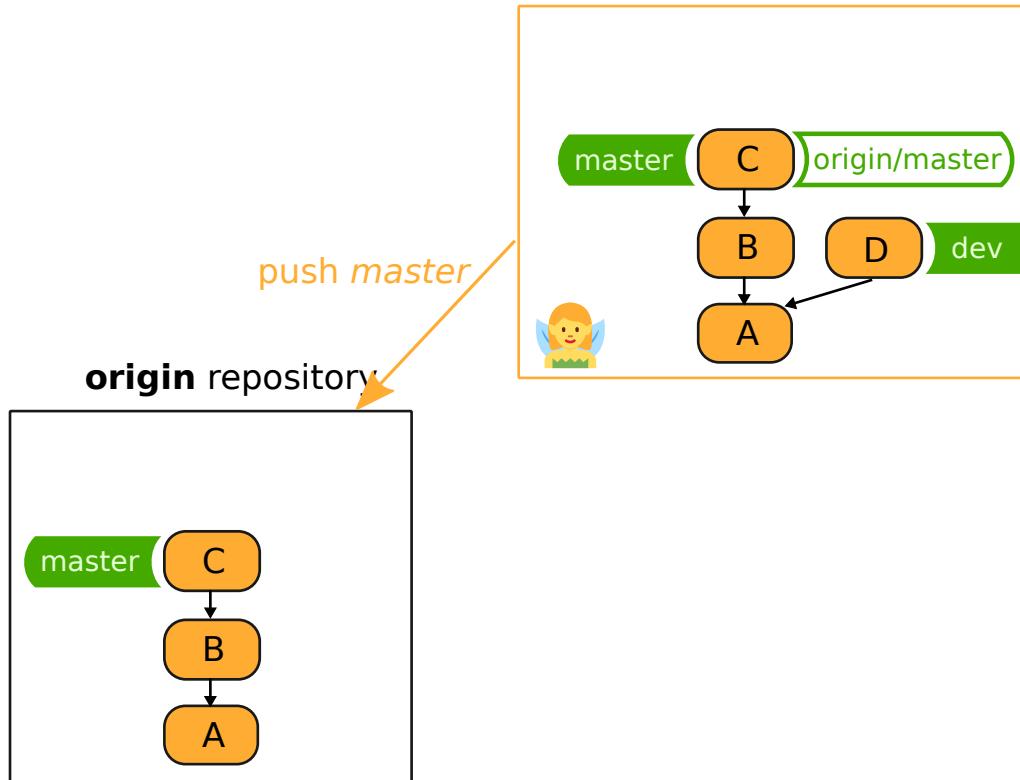
Remotes



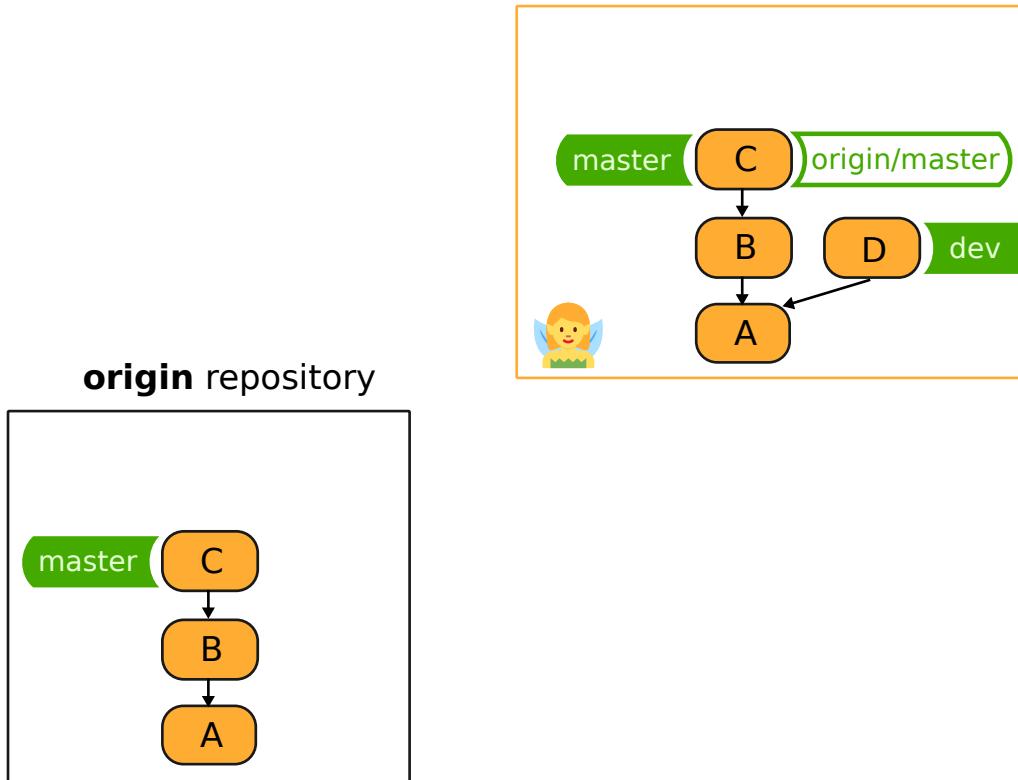
Remotes



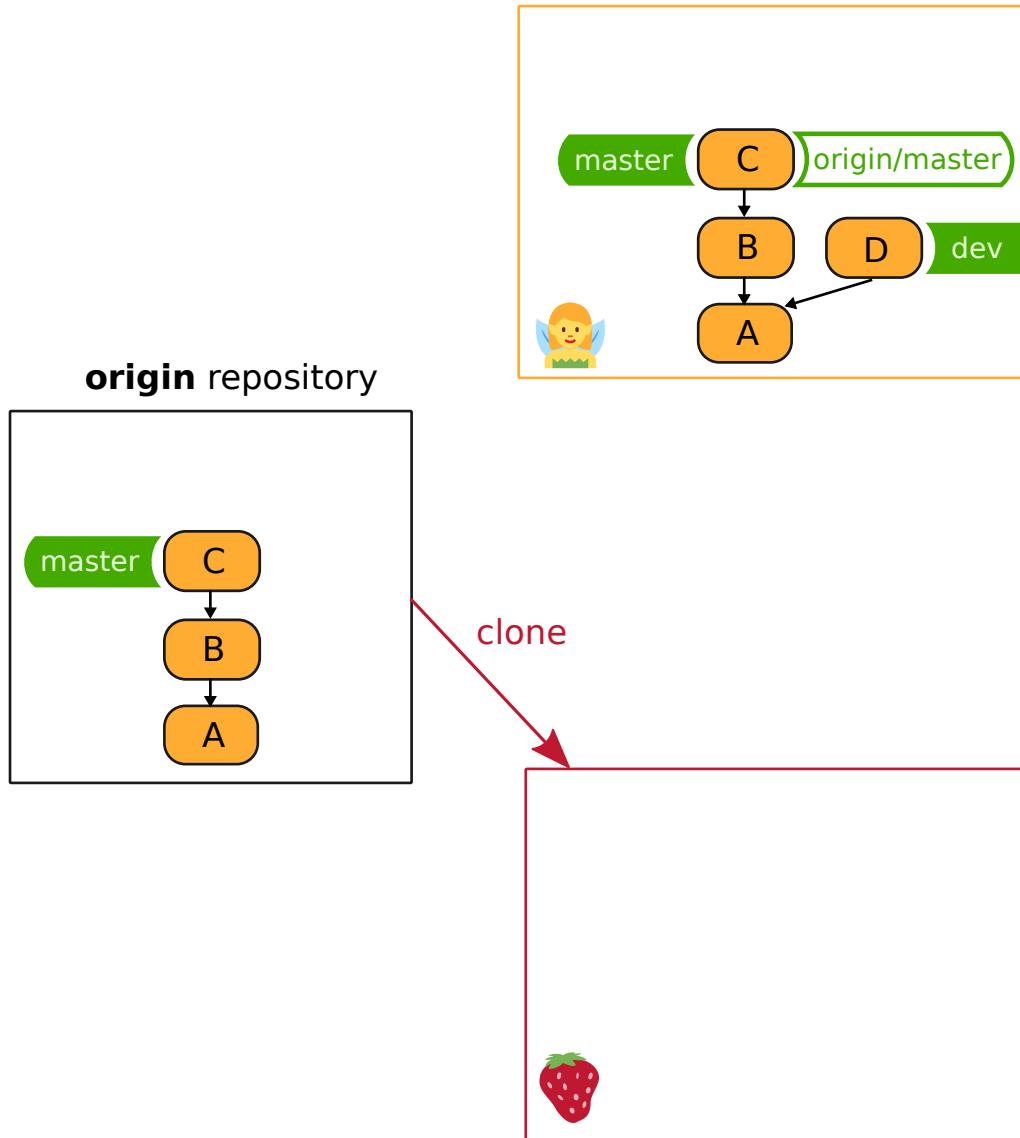
Remotes



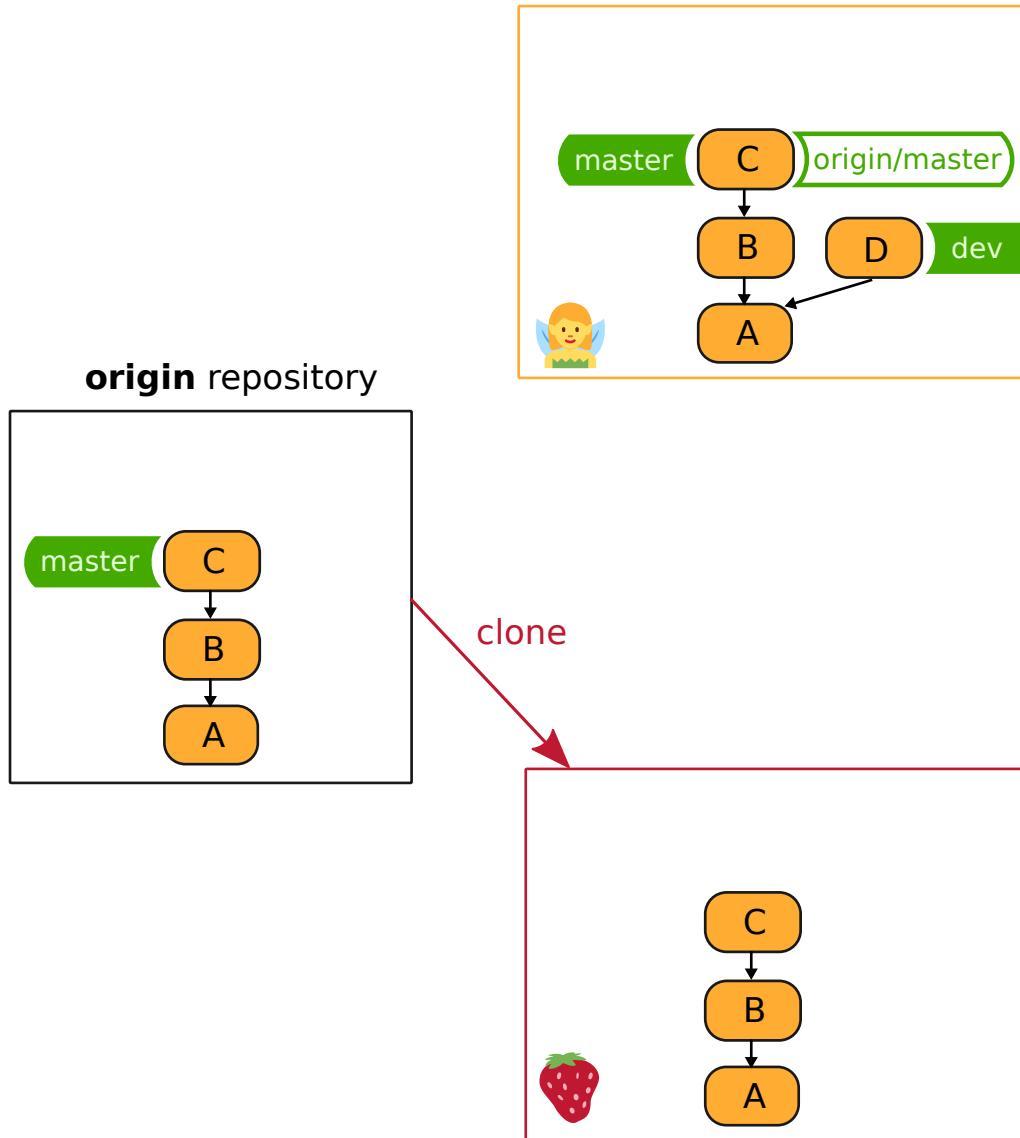
Remotes



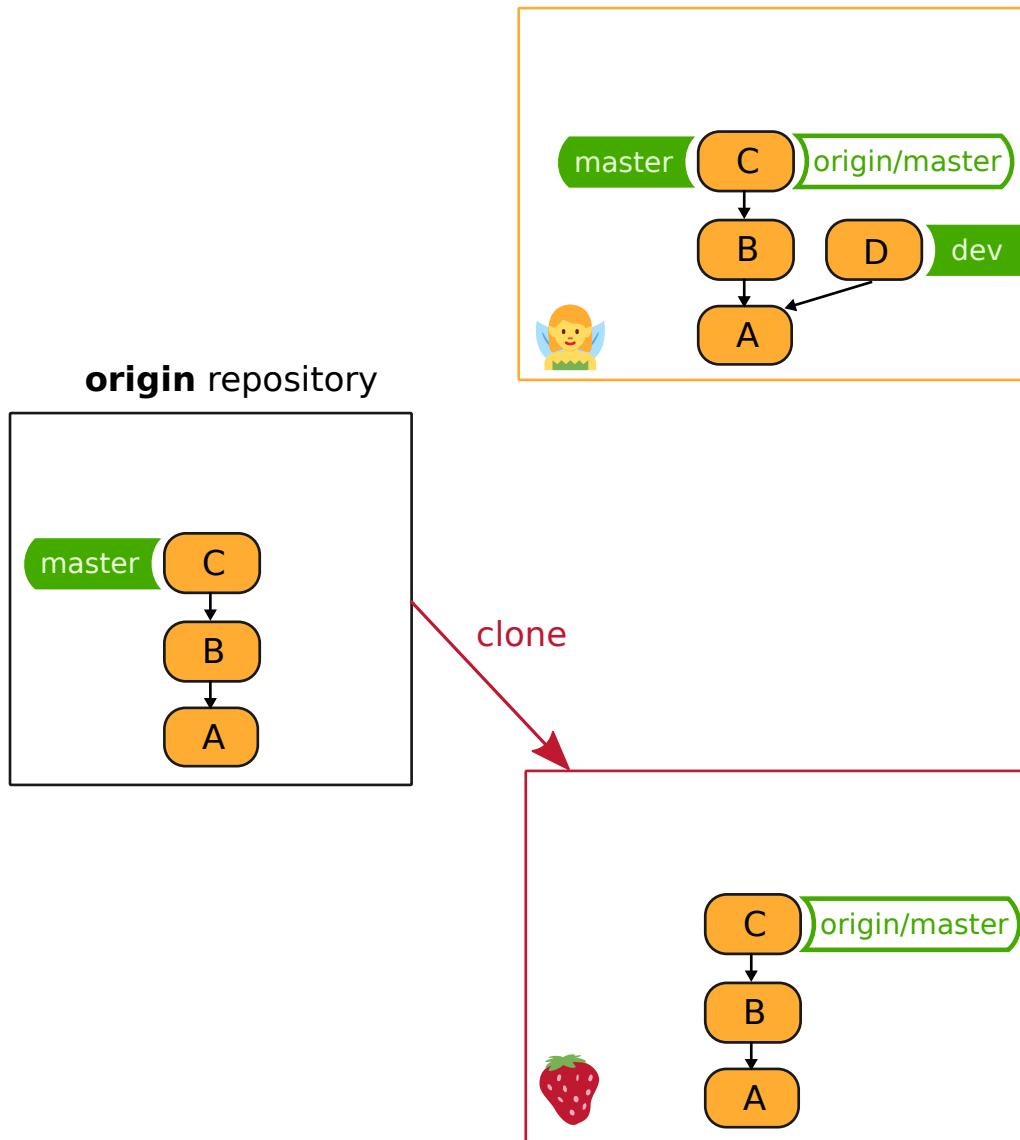
Remotes



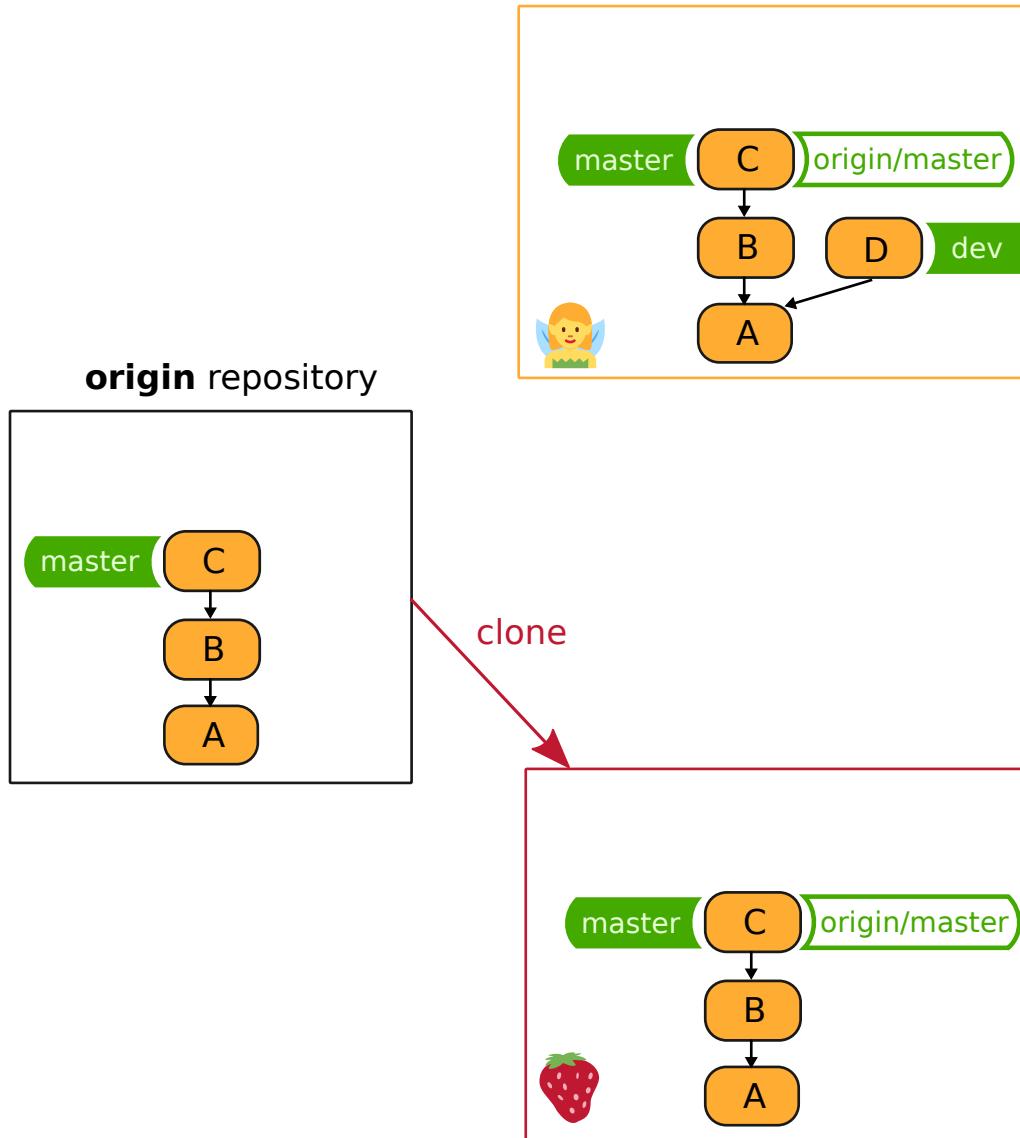
Remotes



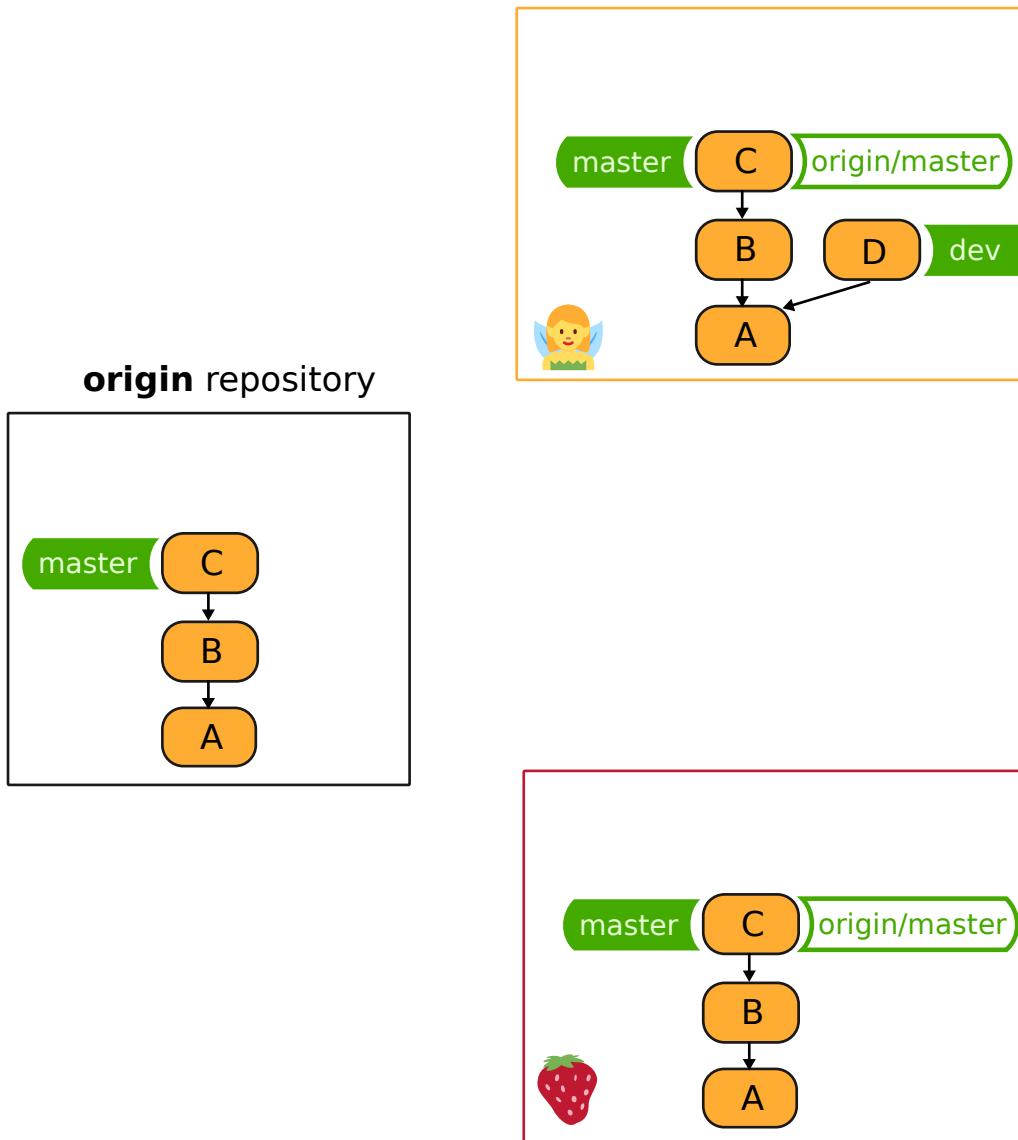
Remotes



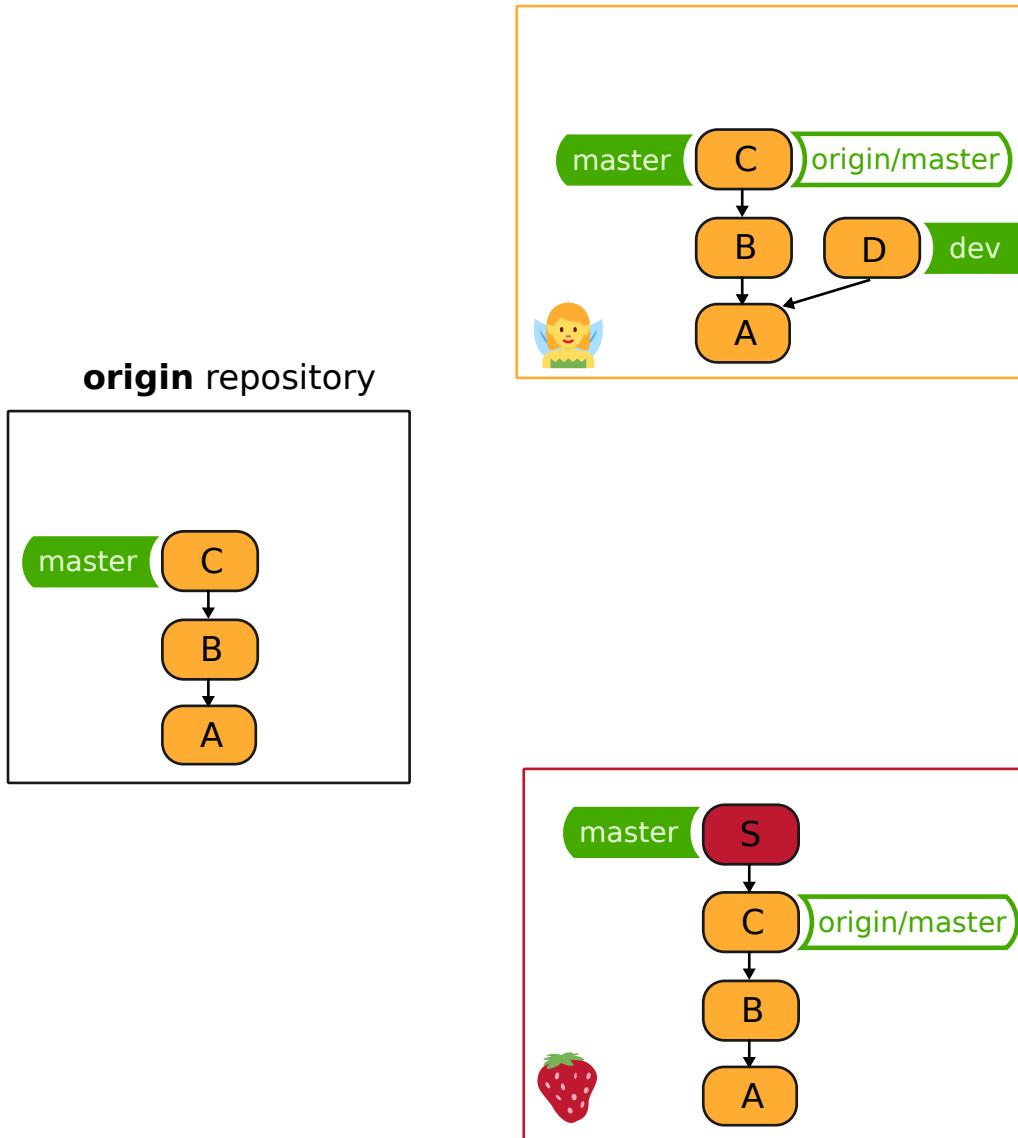
Remotes



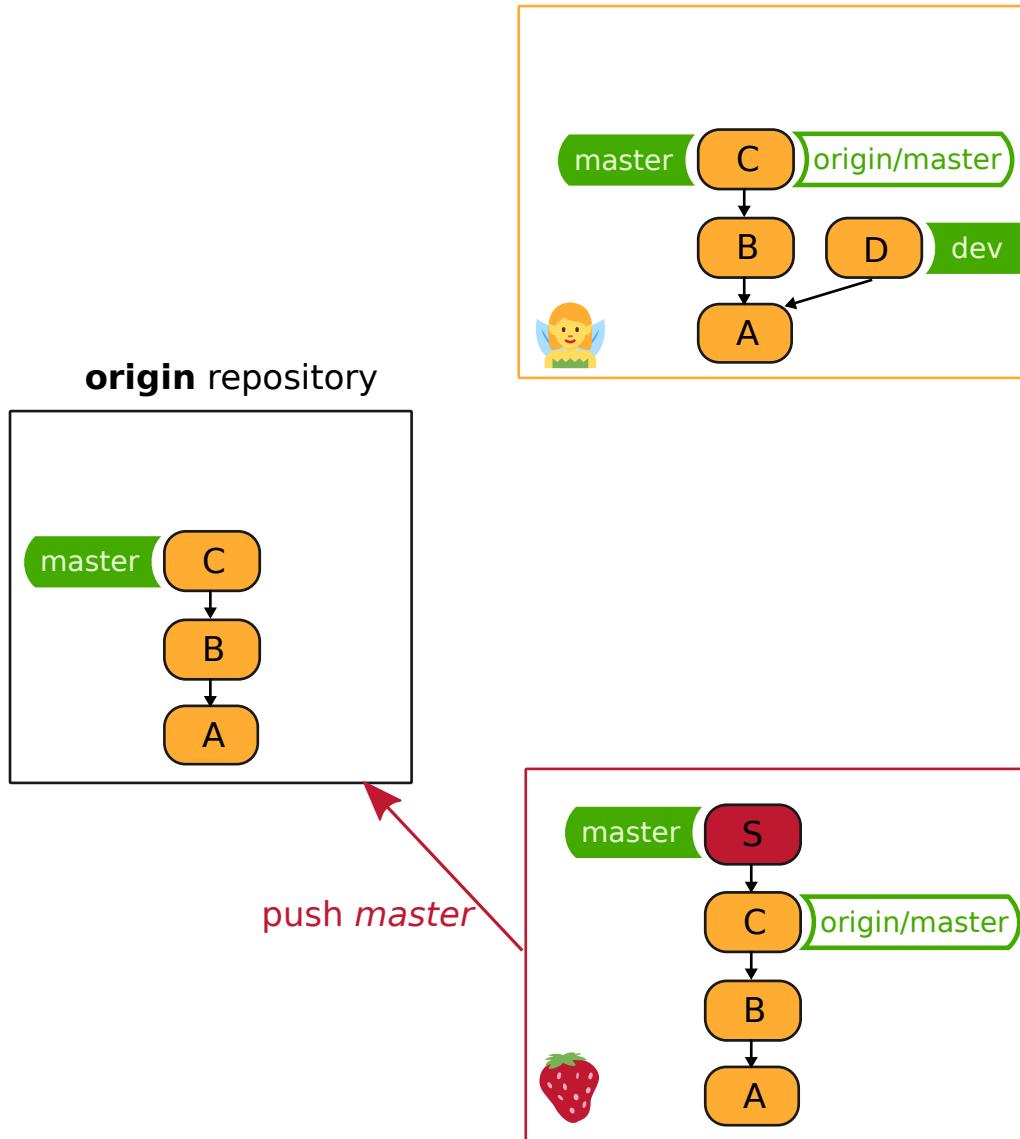
Remotes



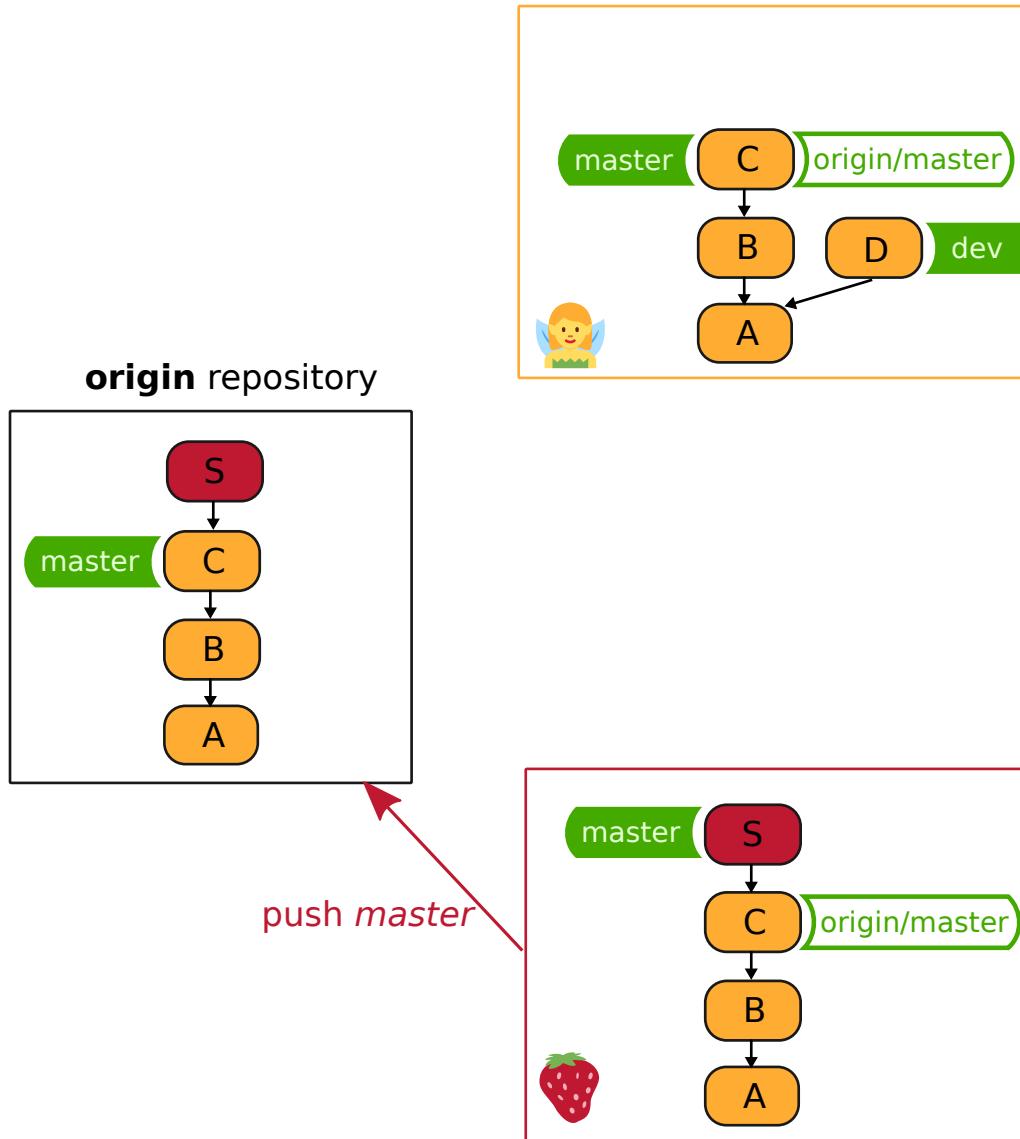
Remotes



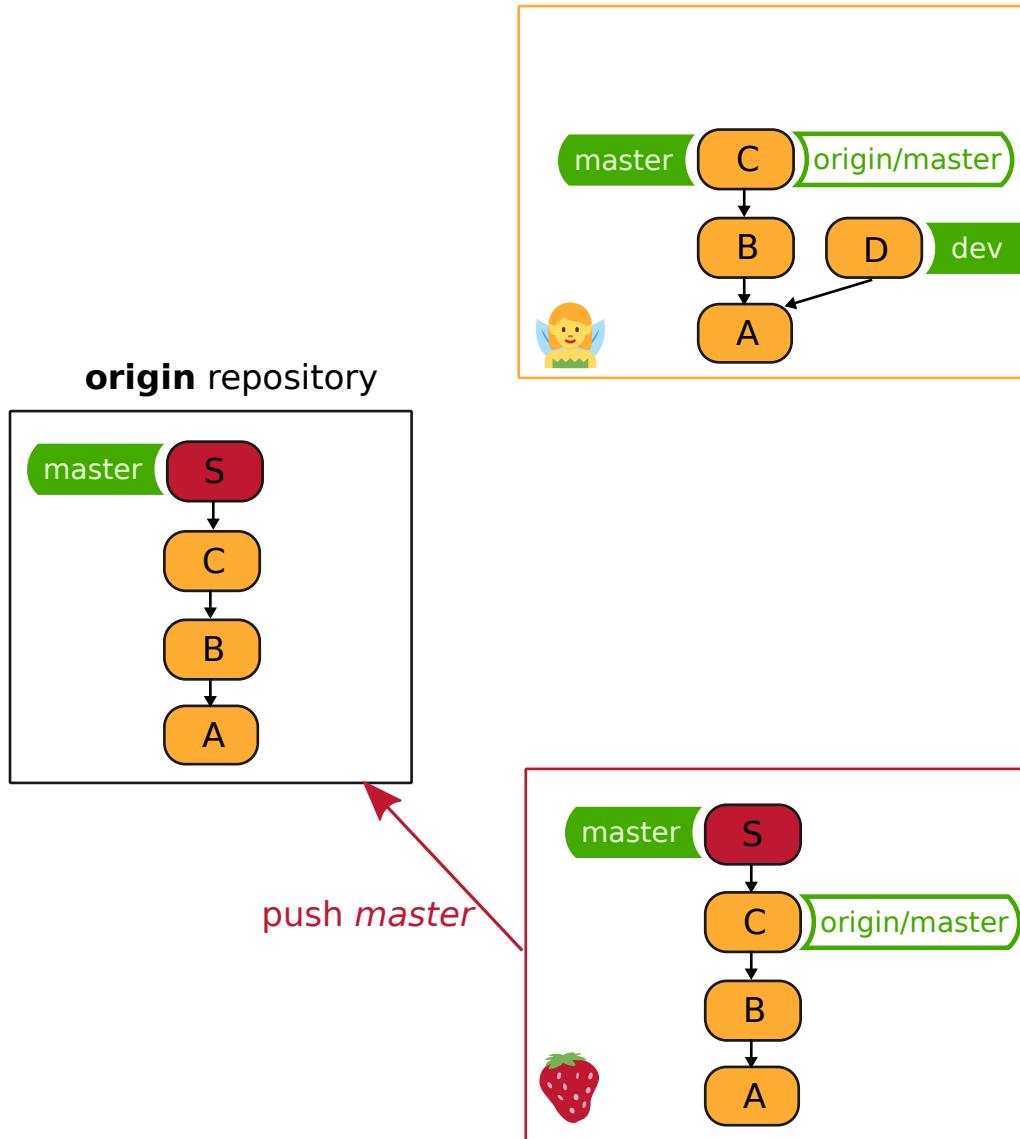
Remotes



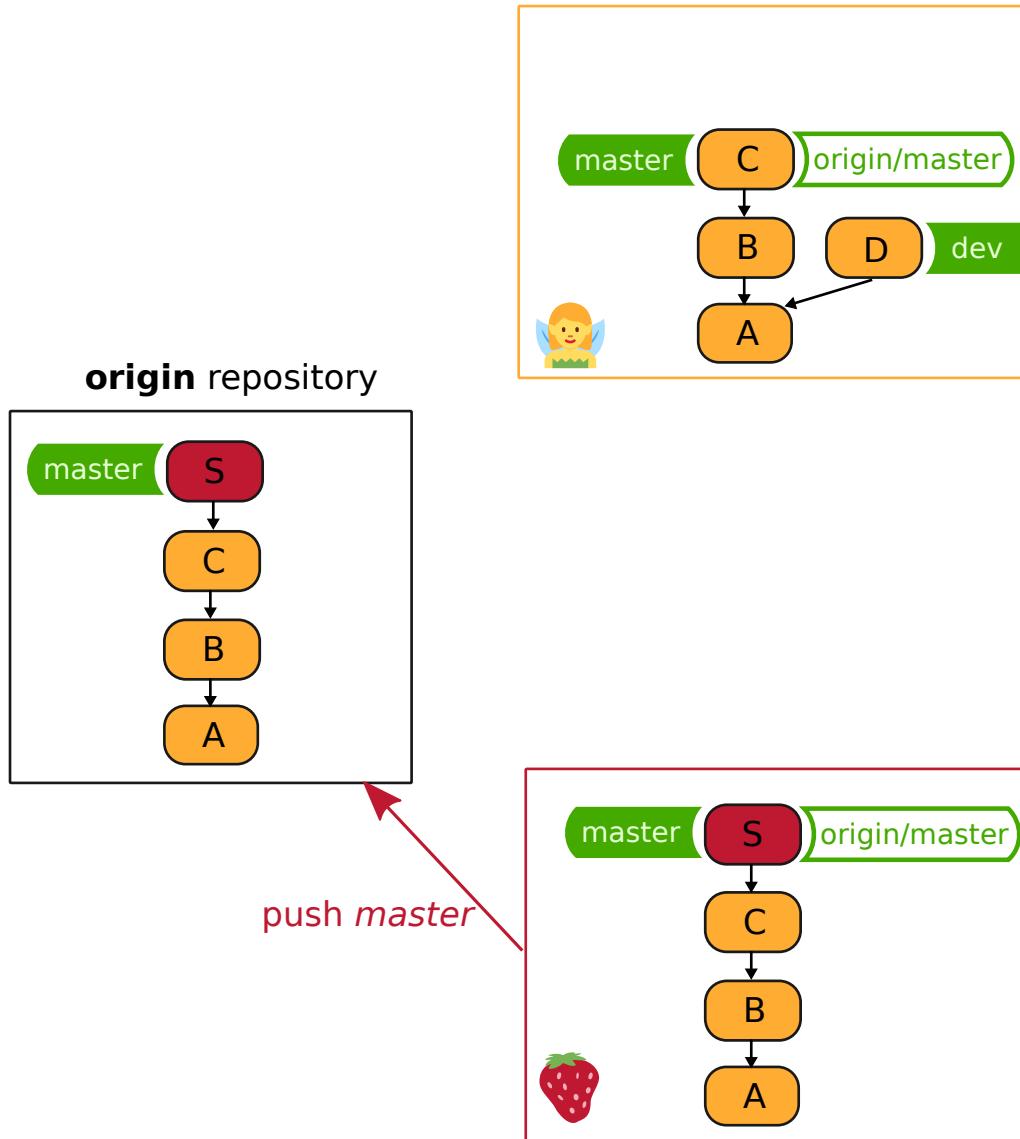
Remotes



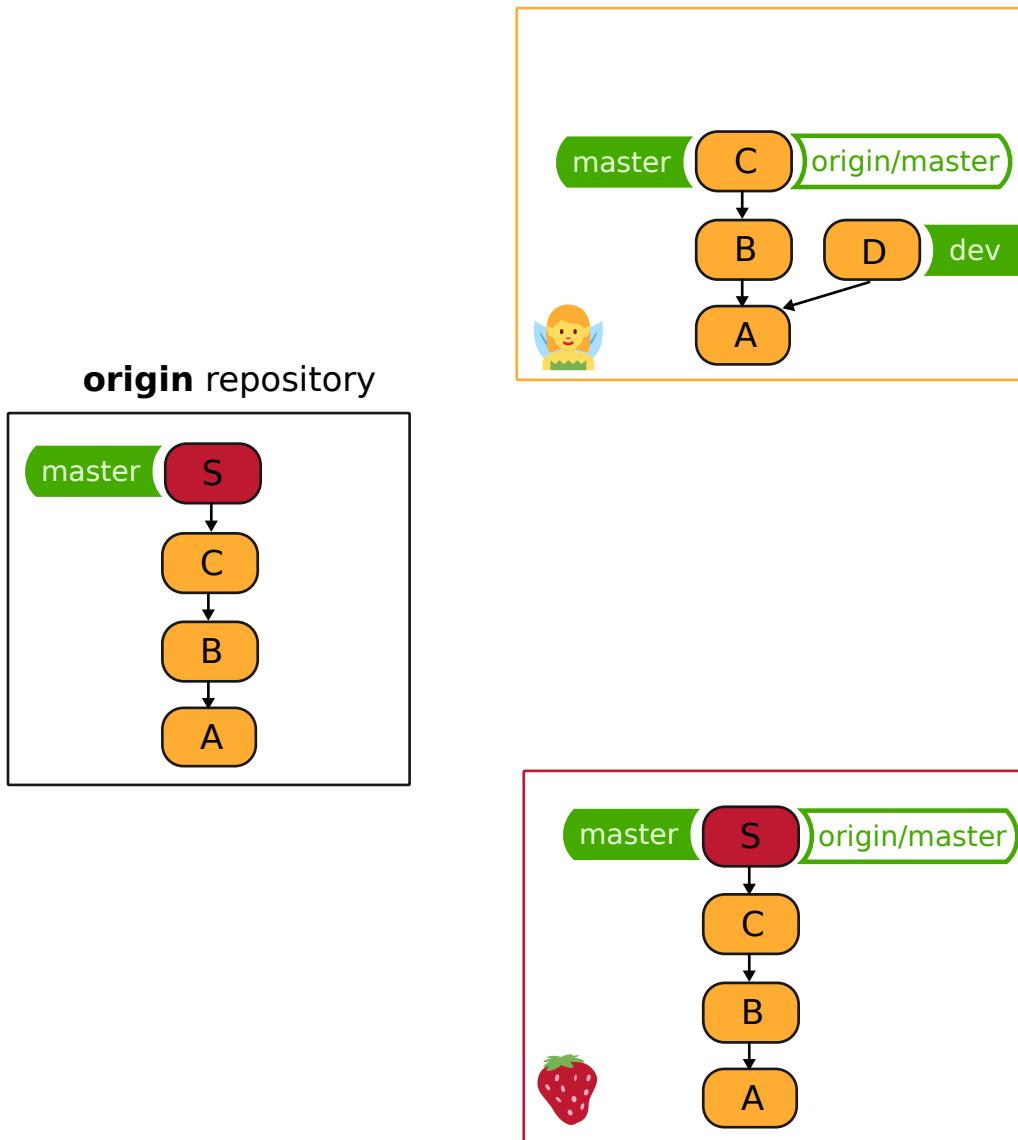
Remotes



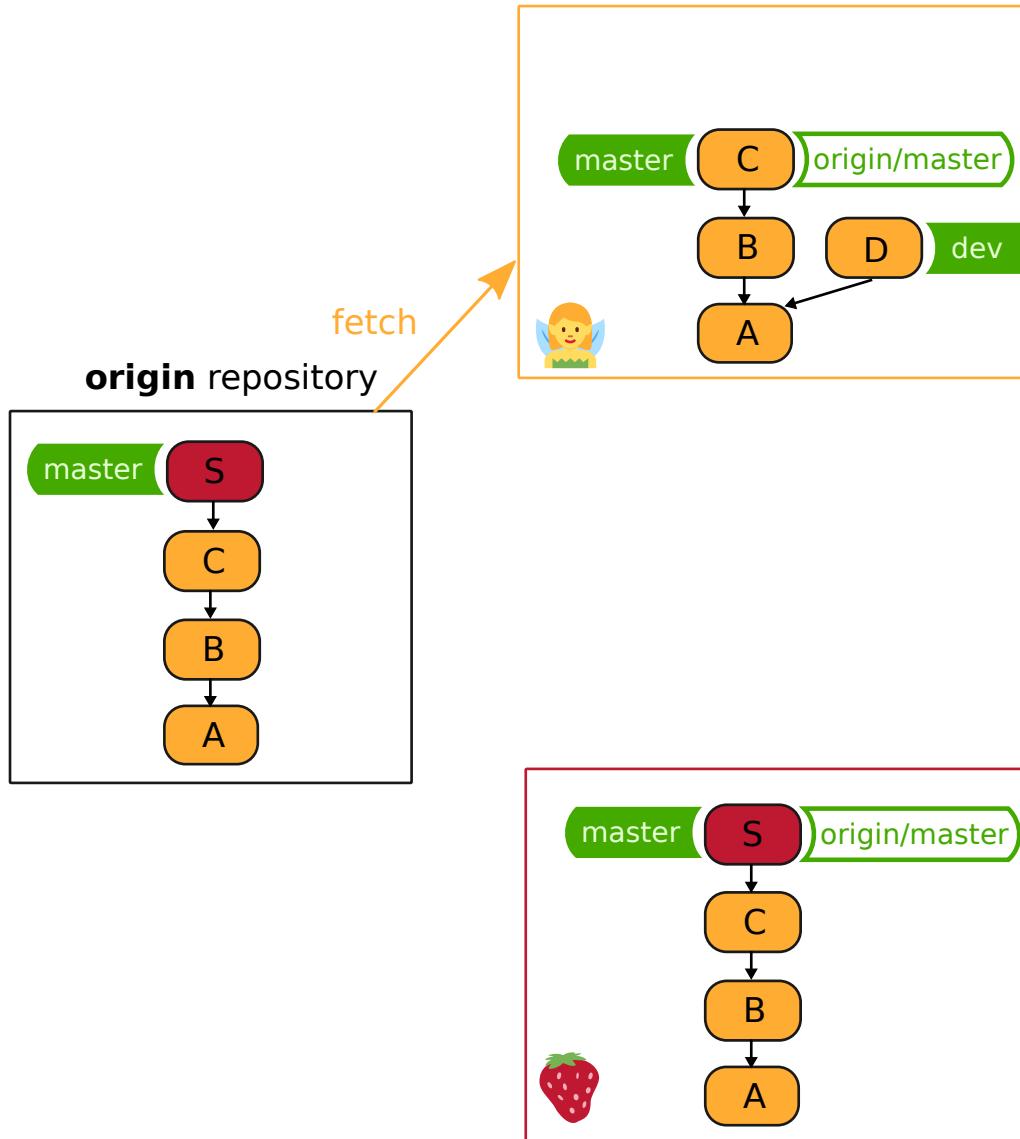
Remotes



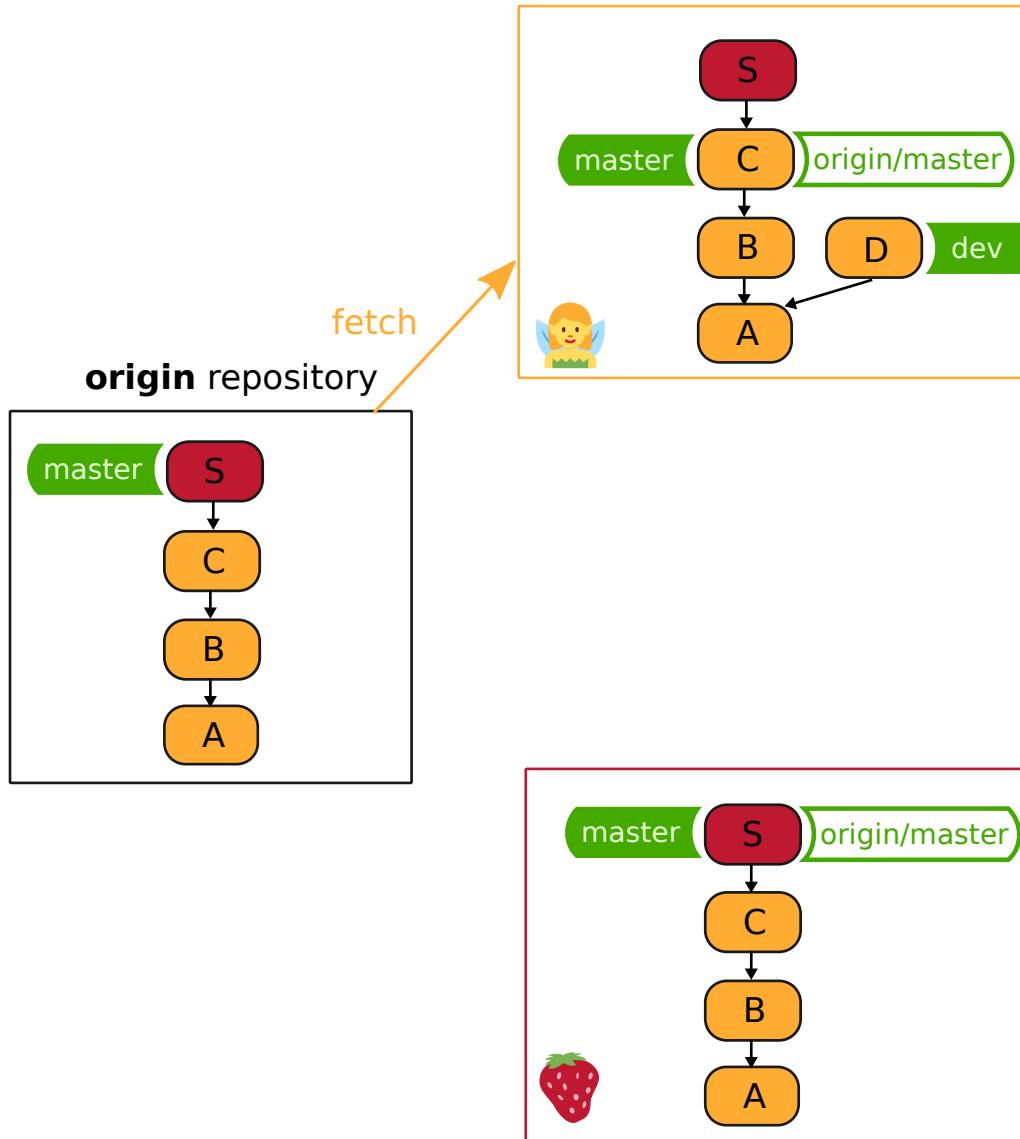
Remotes



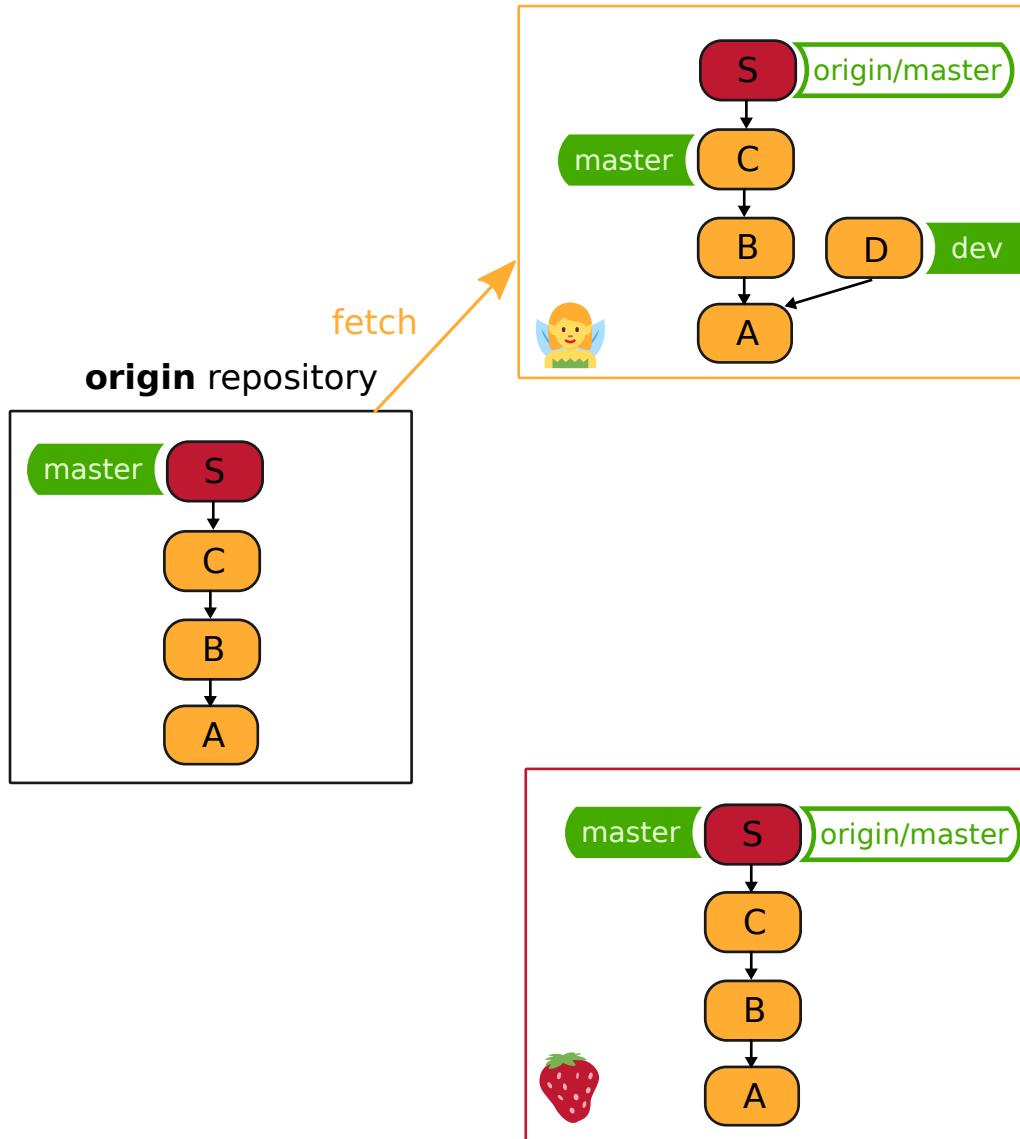
Remotes



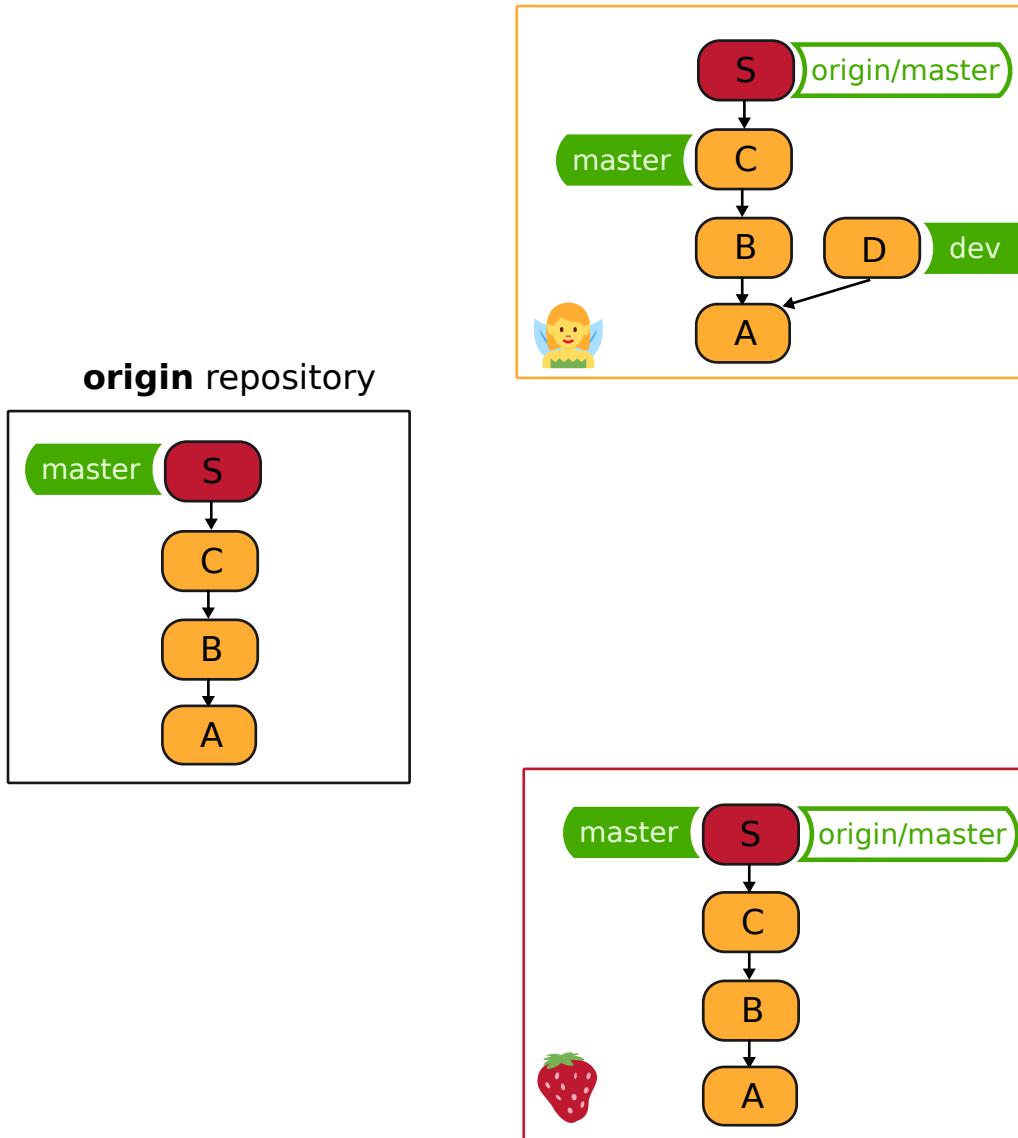
Remotes



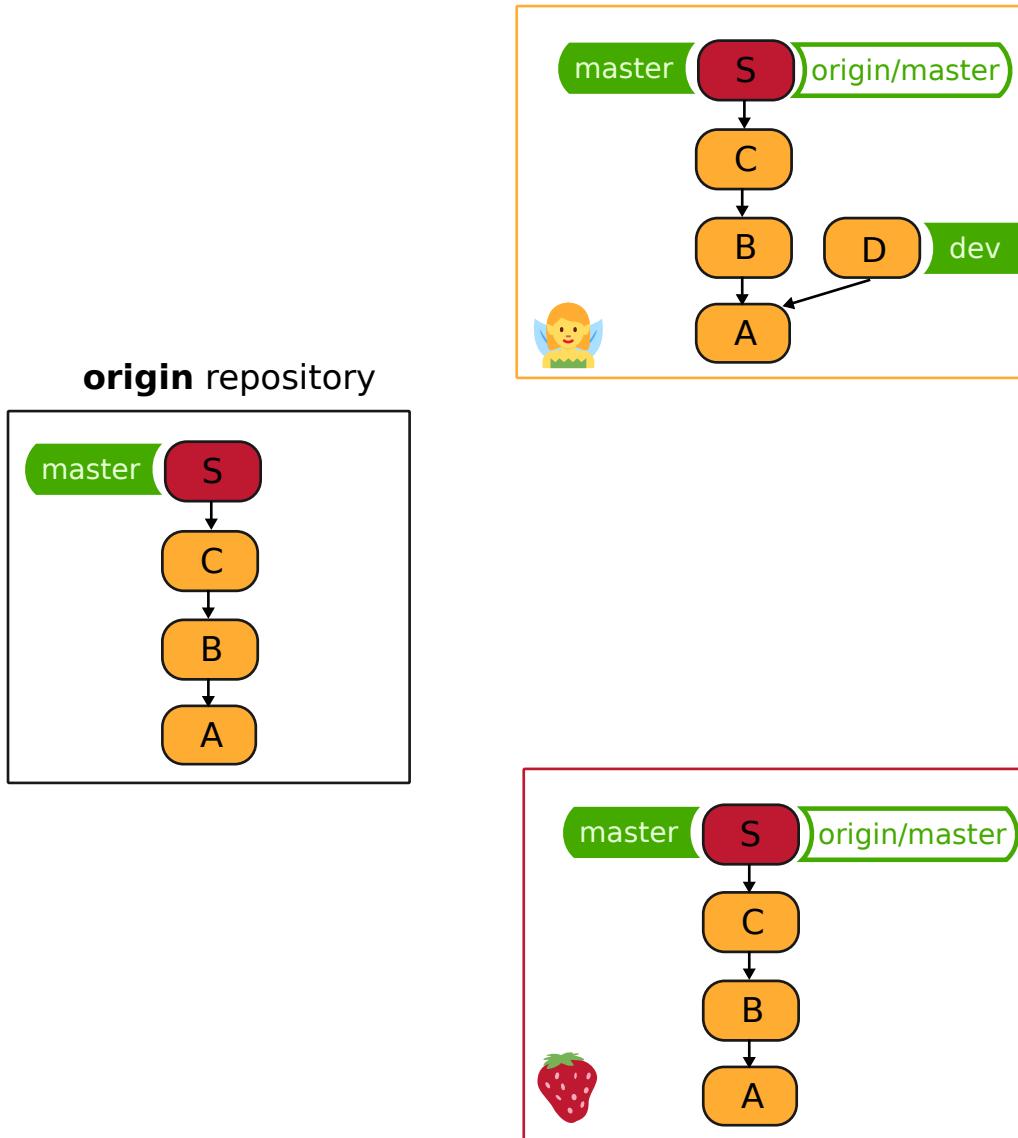
Remotes



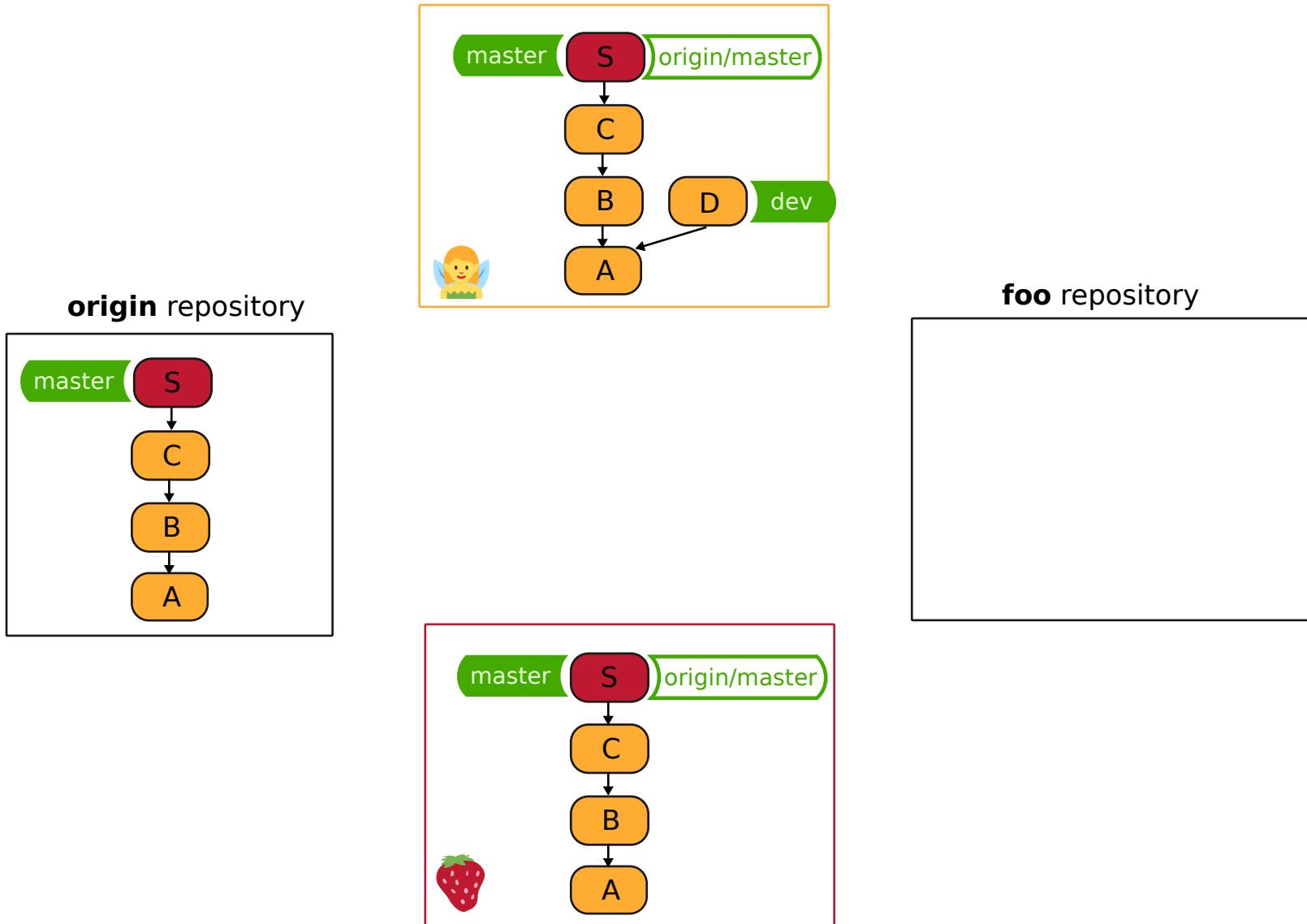
Remotes



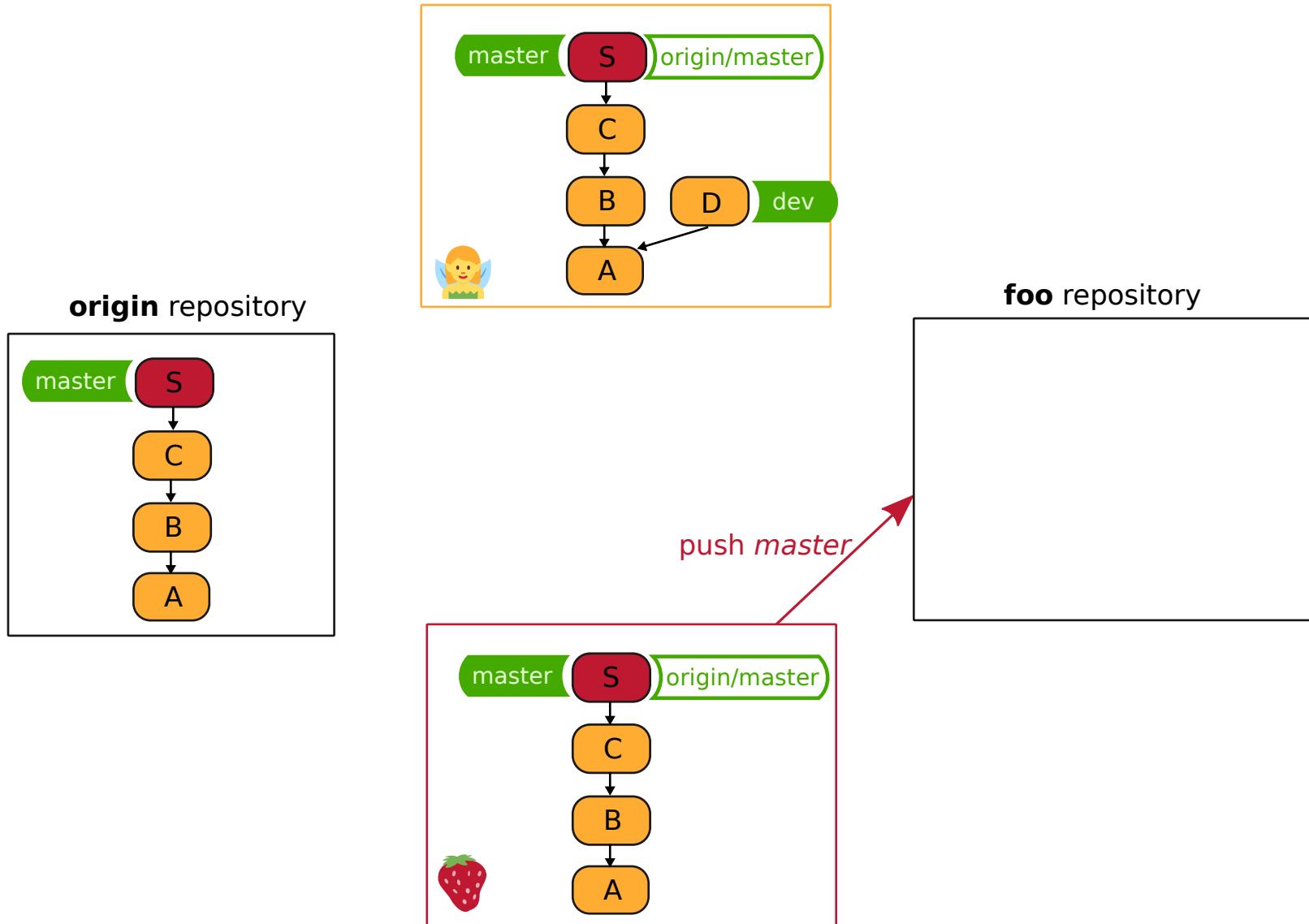
Remotes



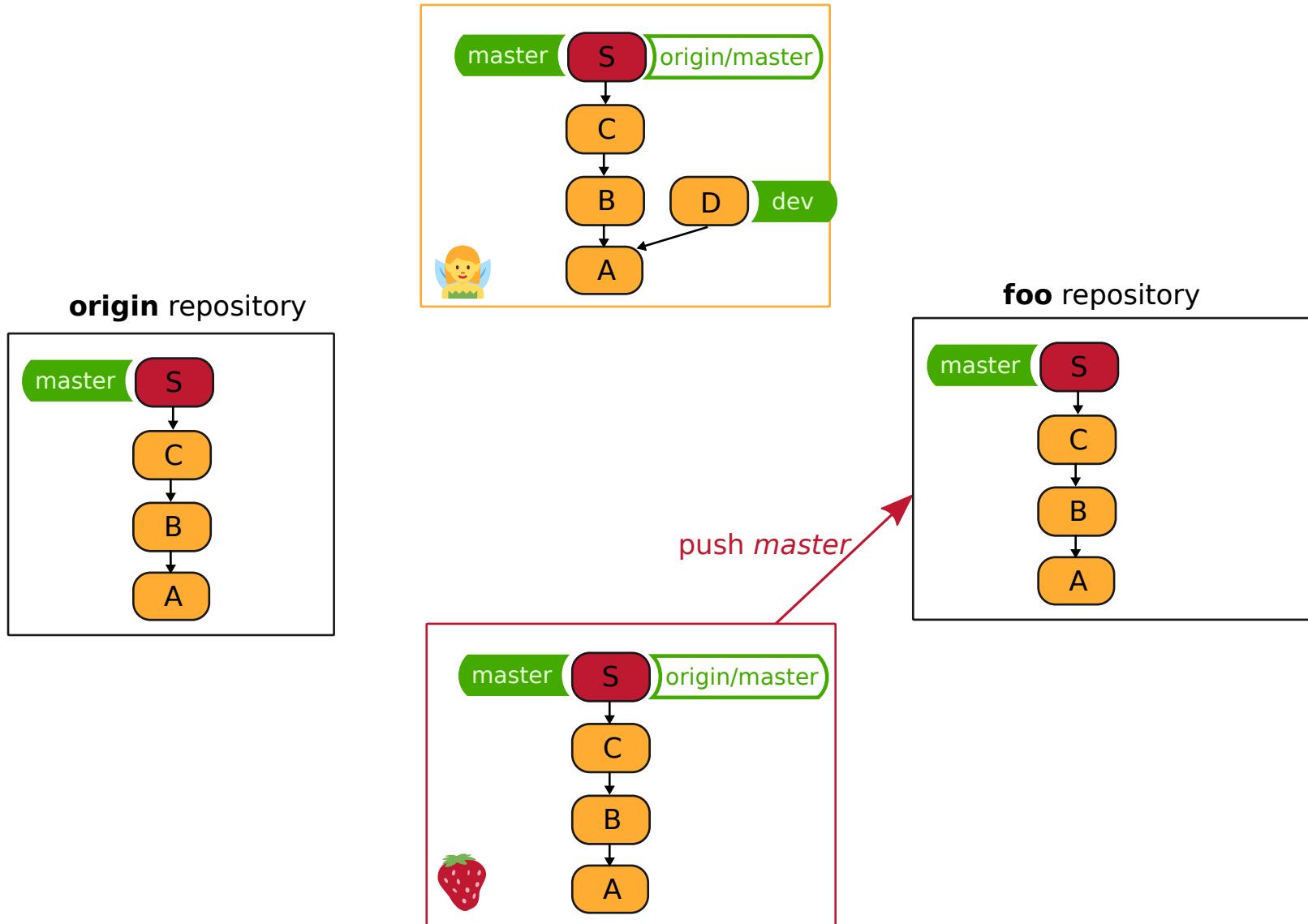
Remotes



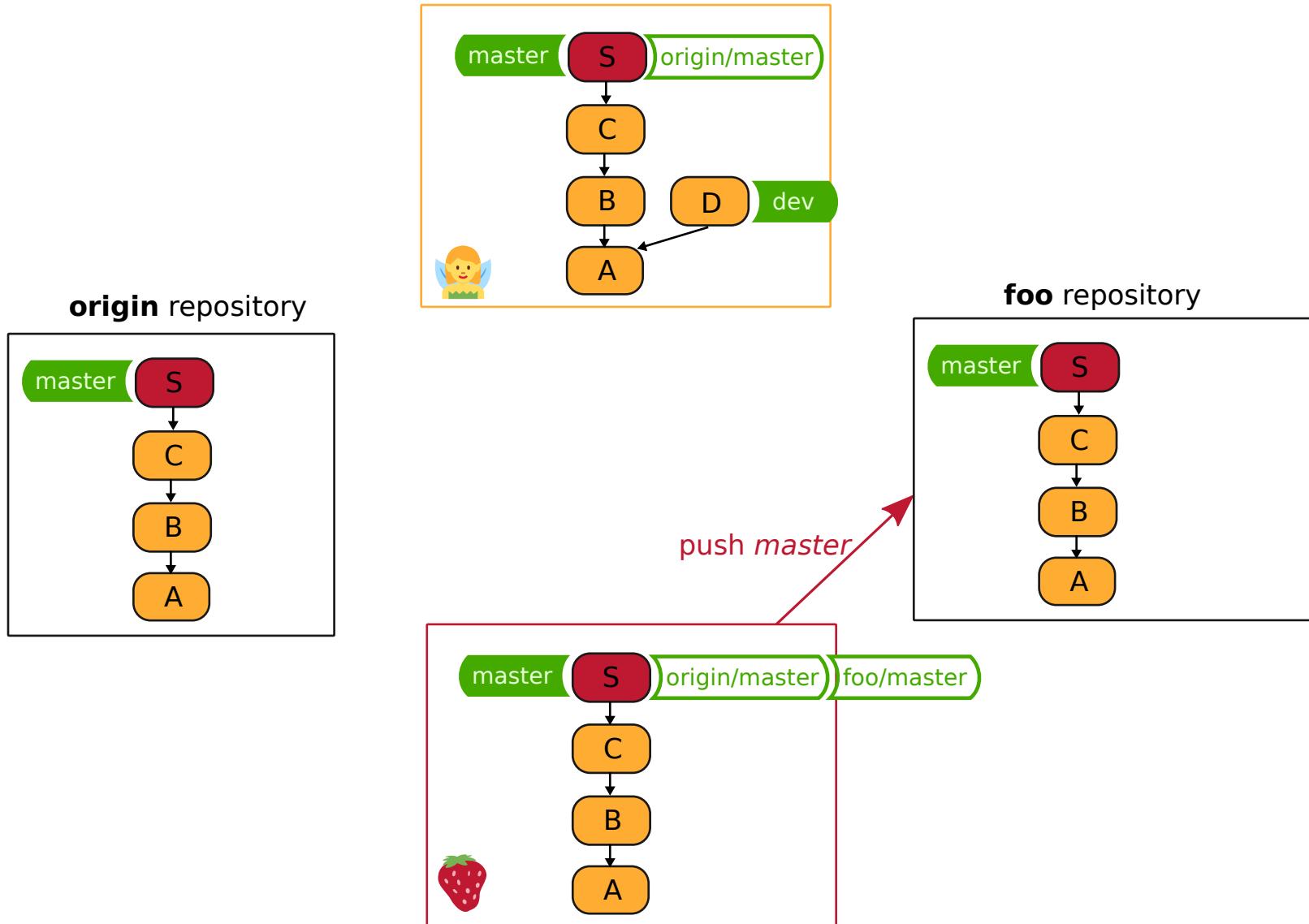
Remotes



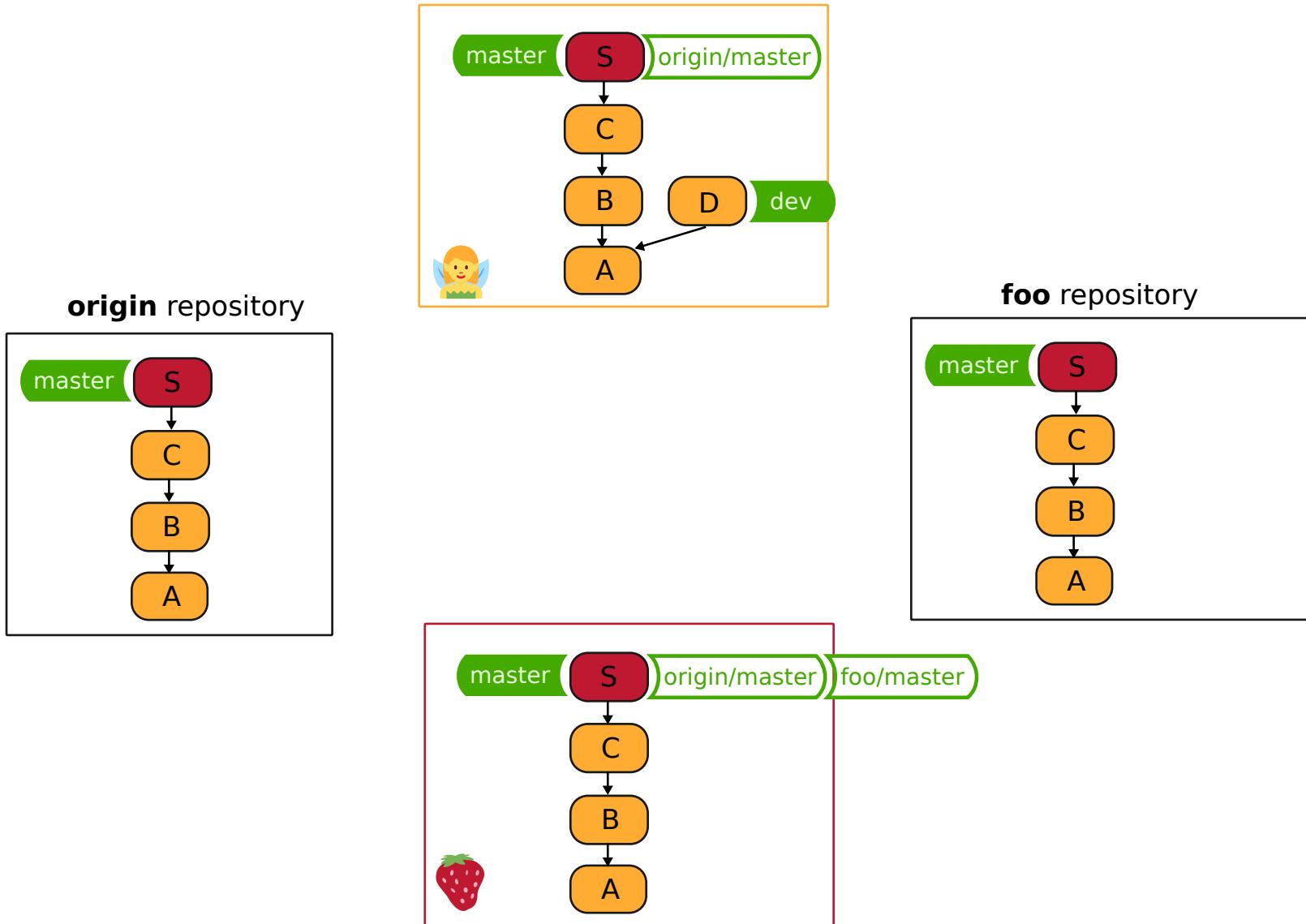
Remotes



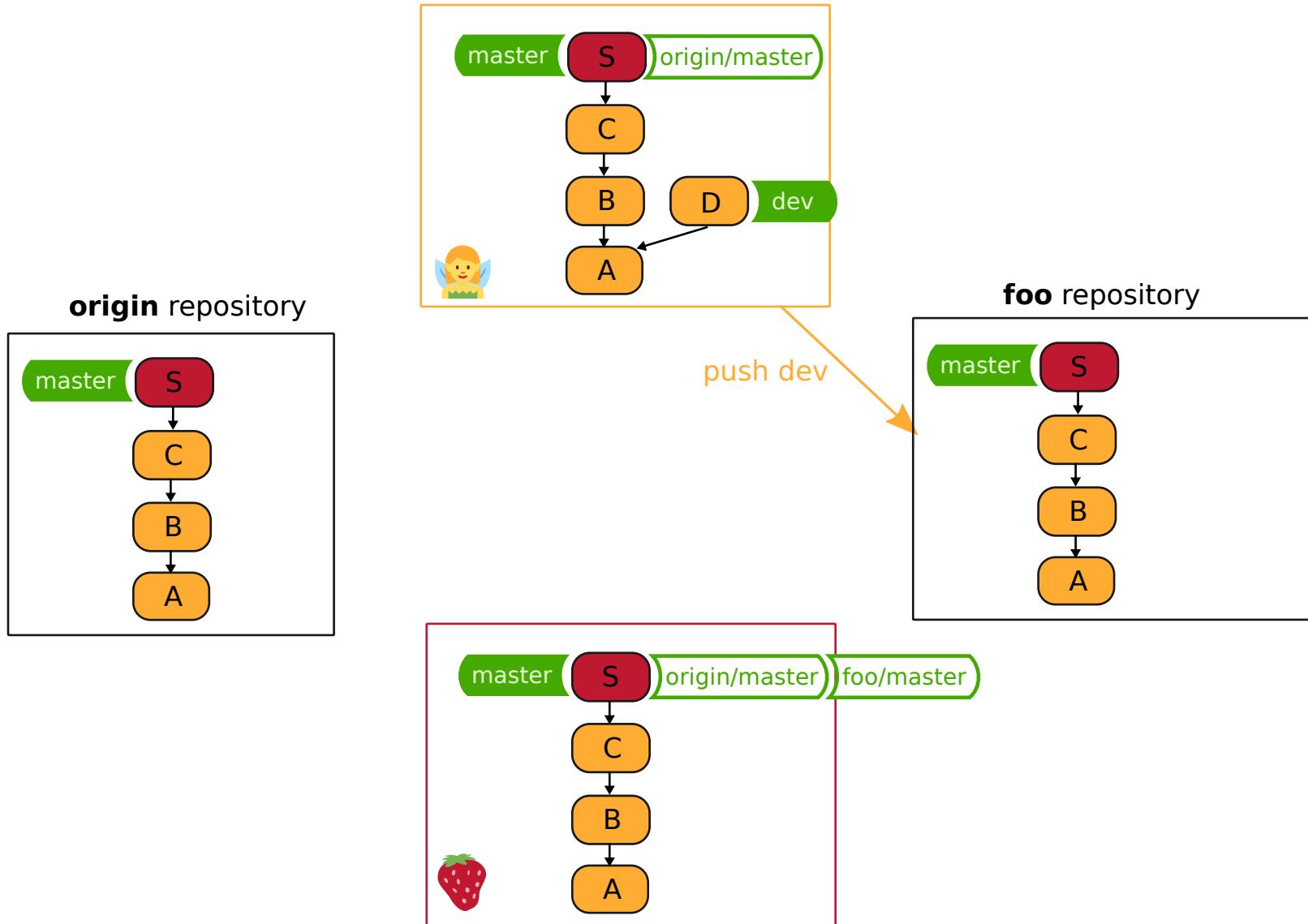
Remotes



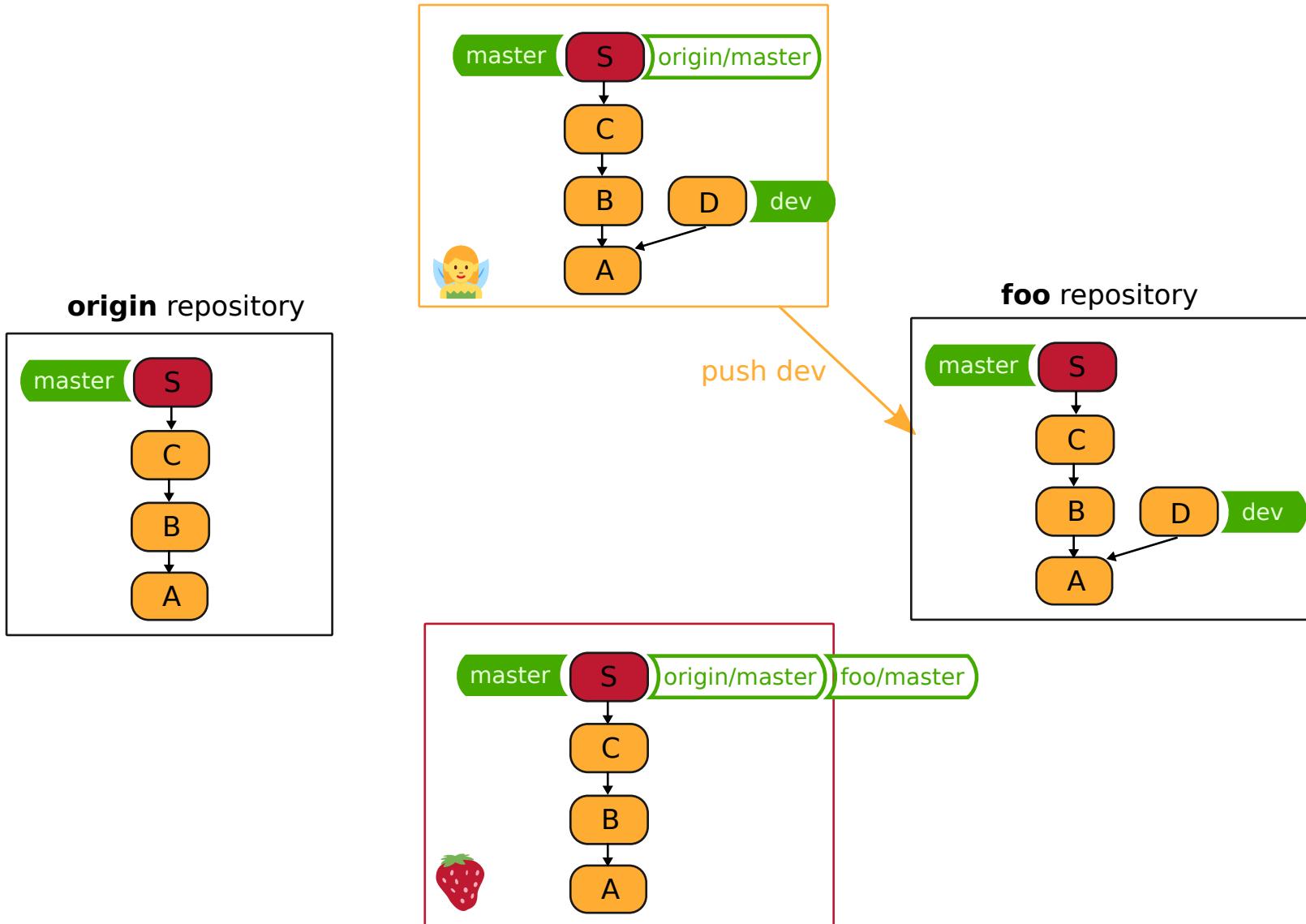
Remotes



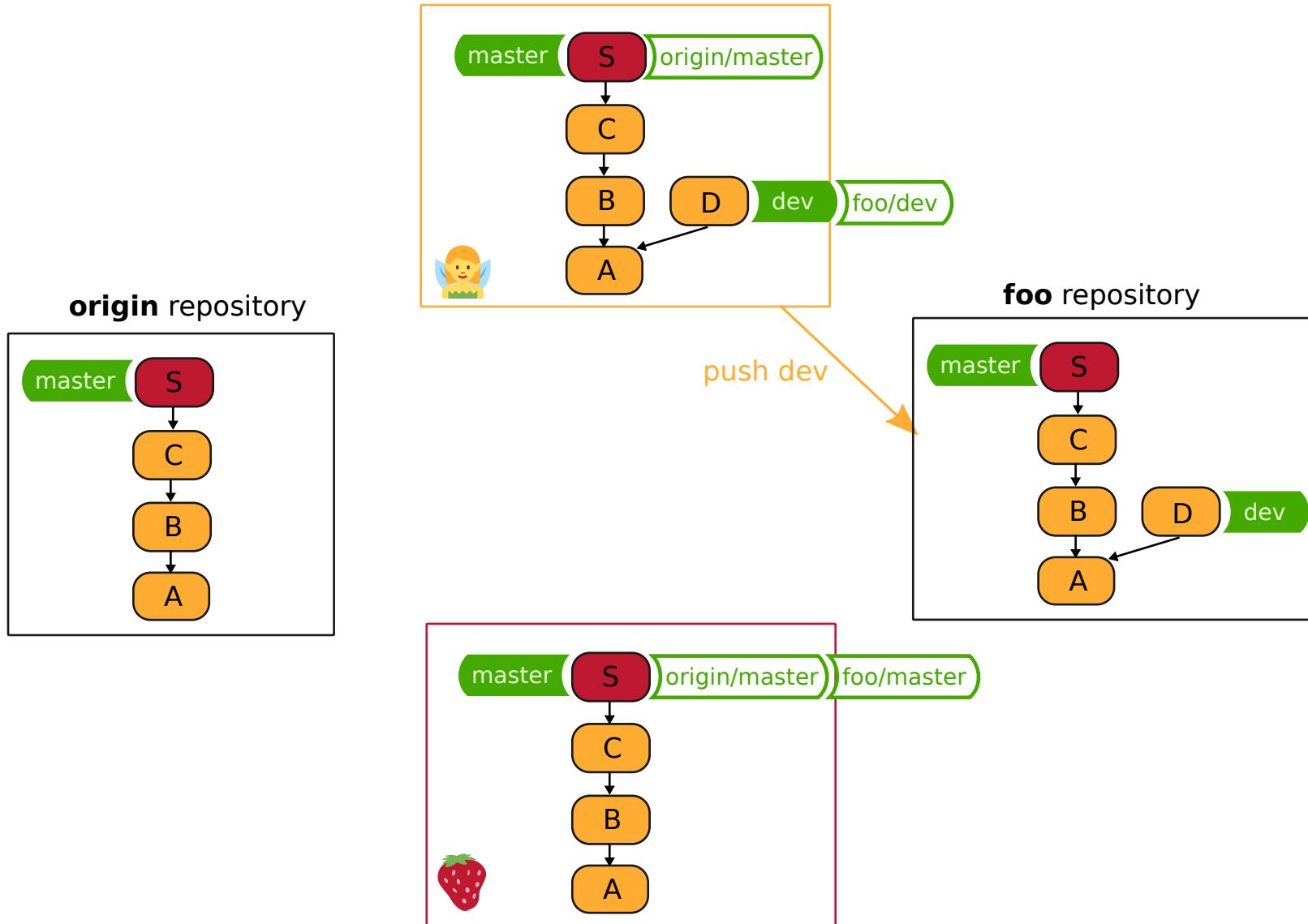
Remotes



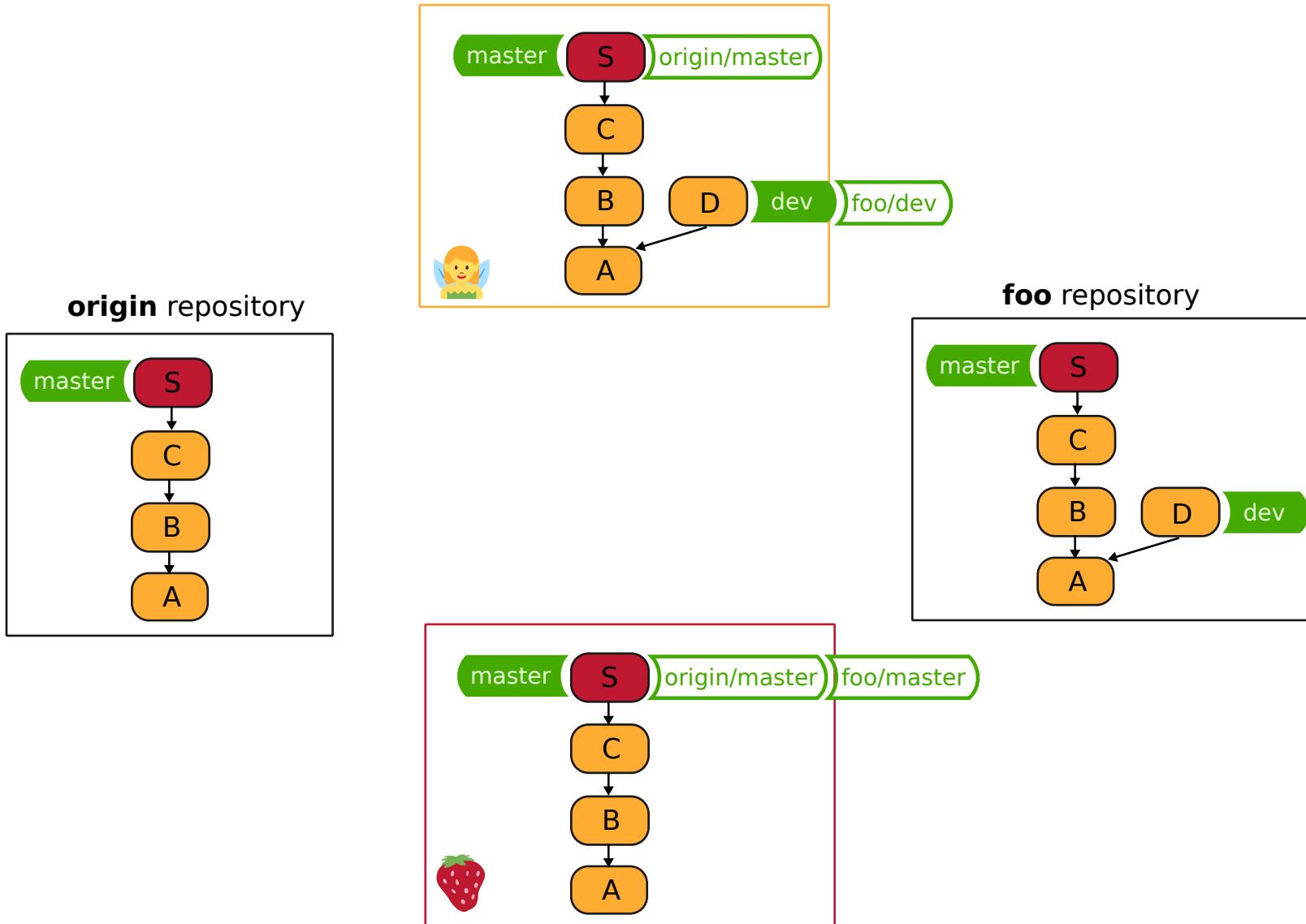
Remotes



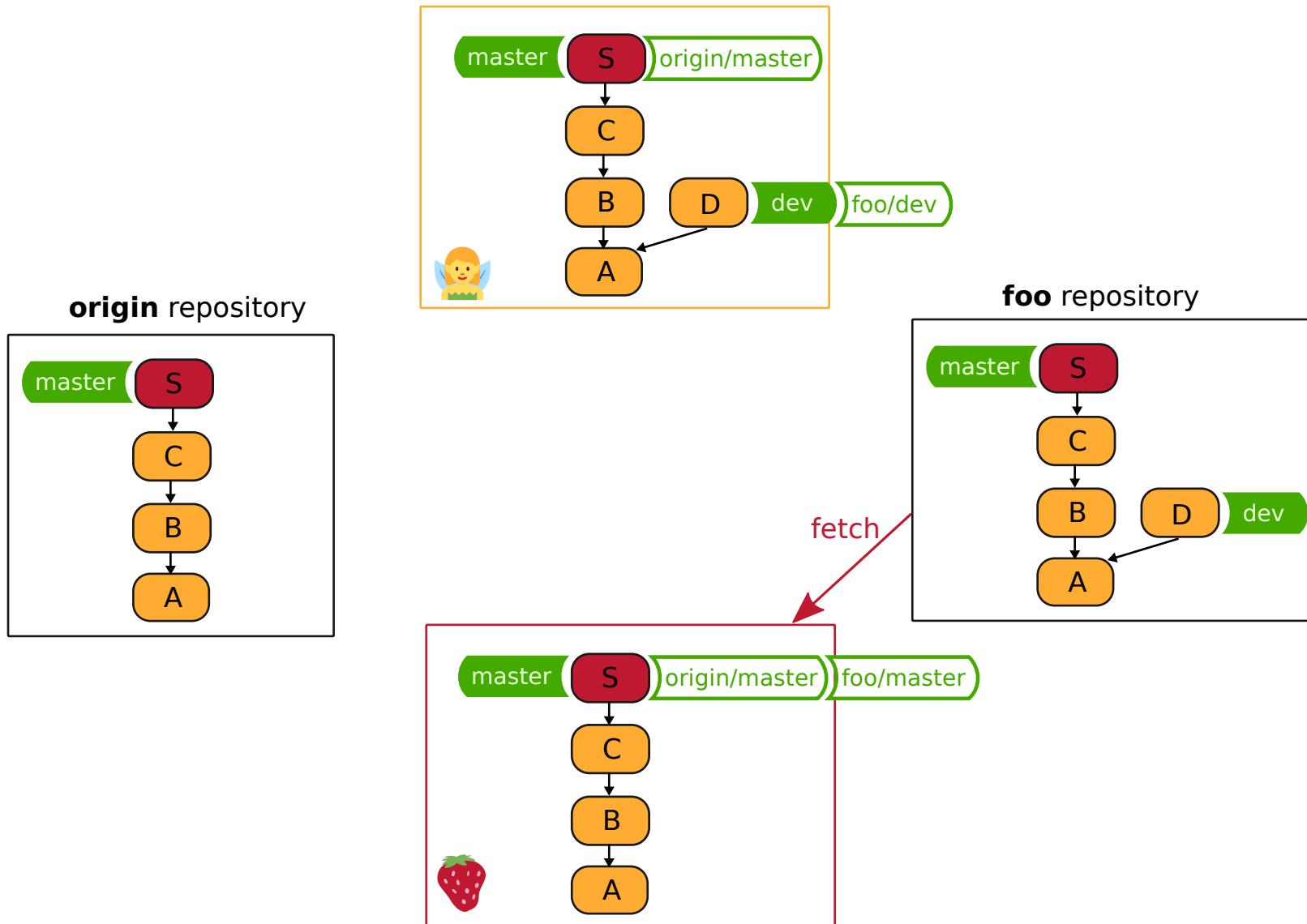
Remotes



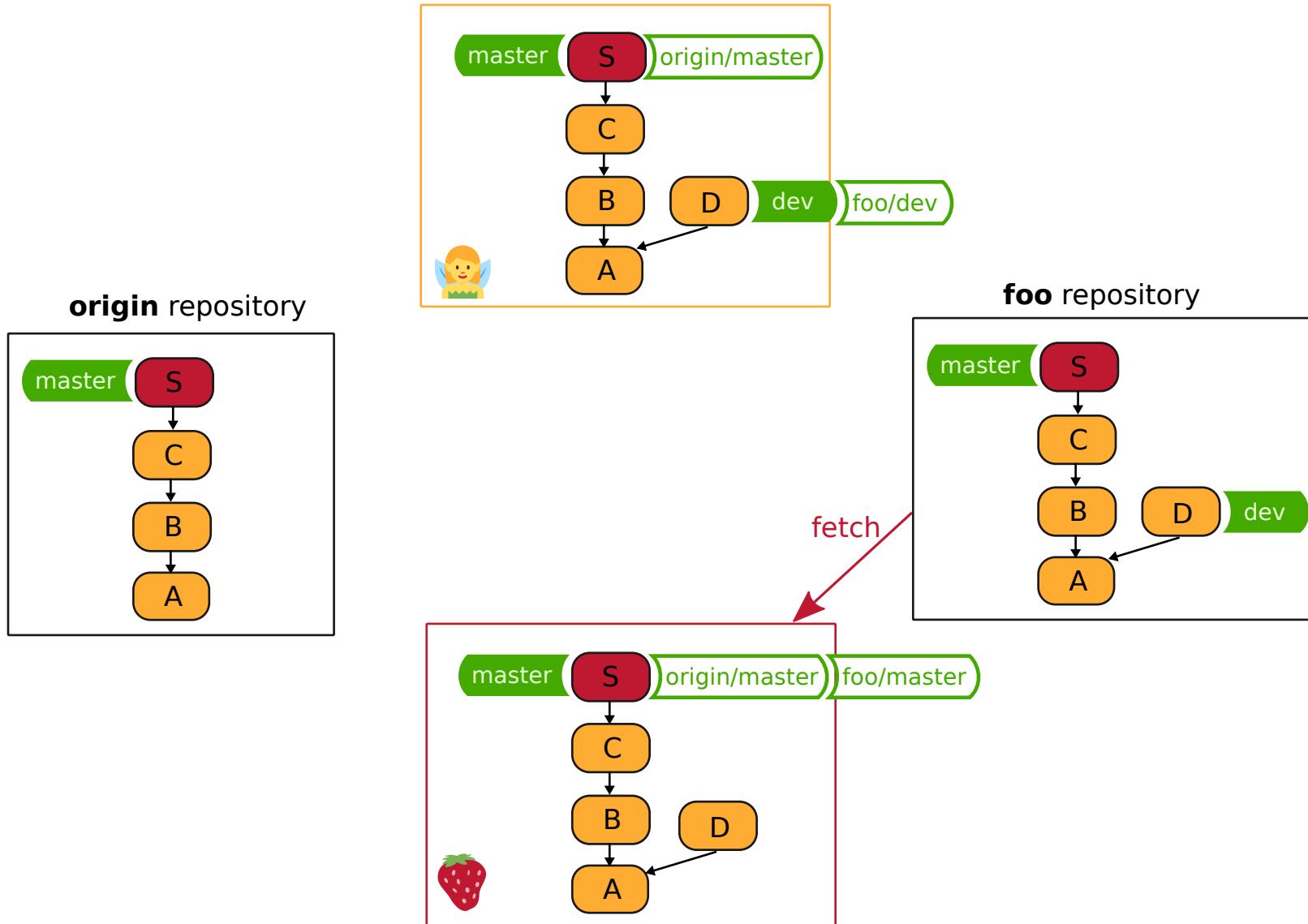
Remotes



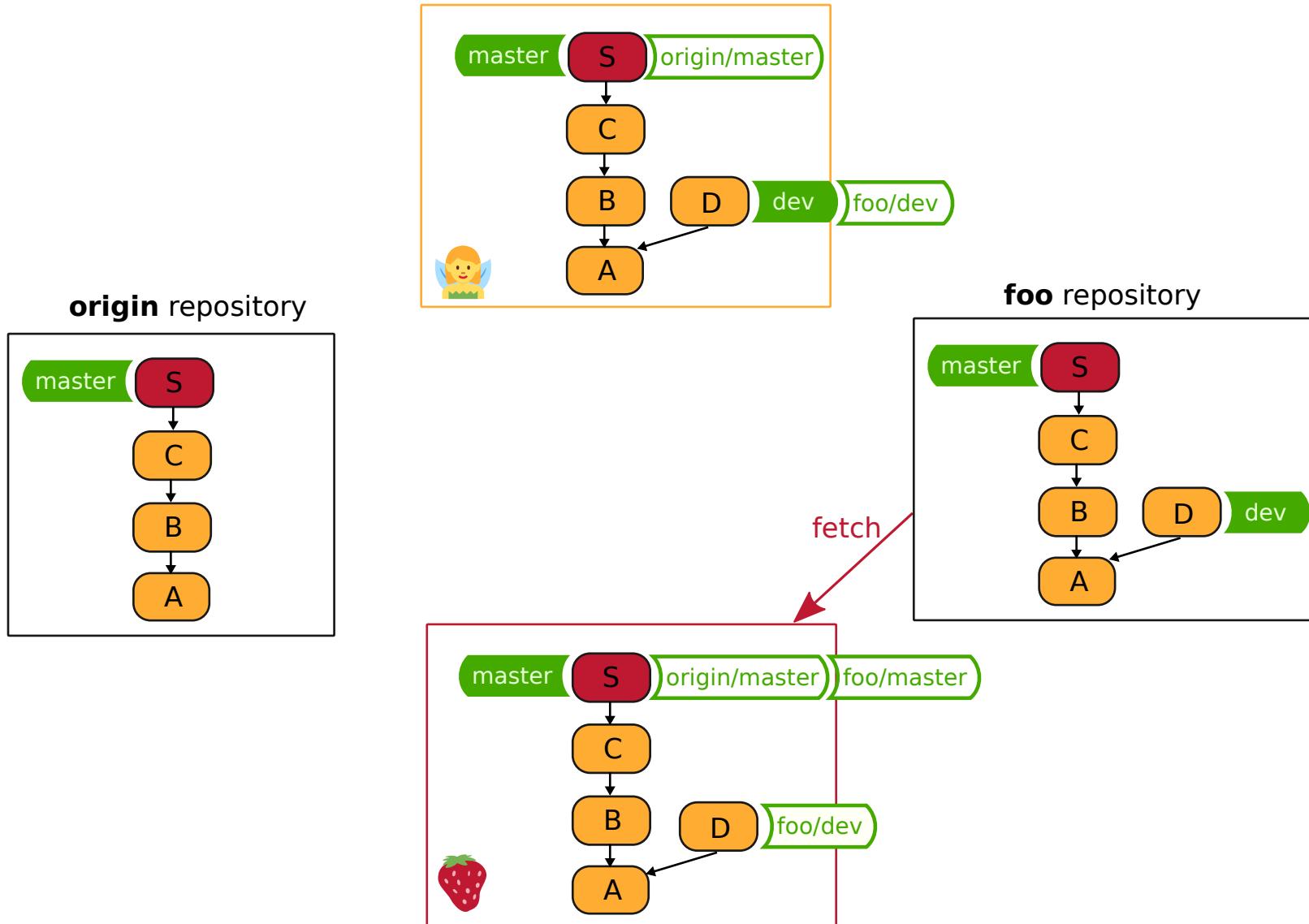
Remotes



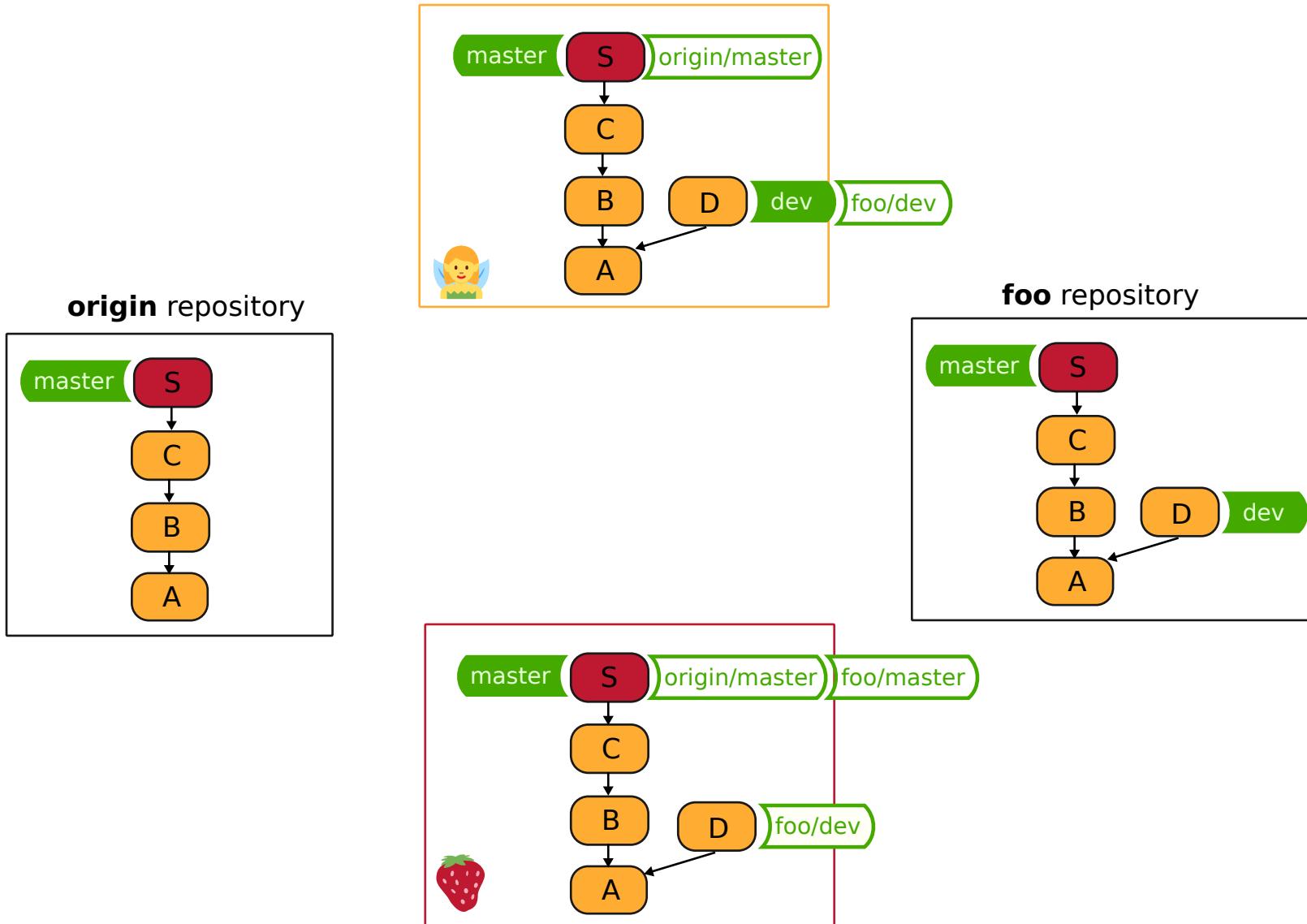
Remotes



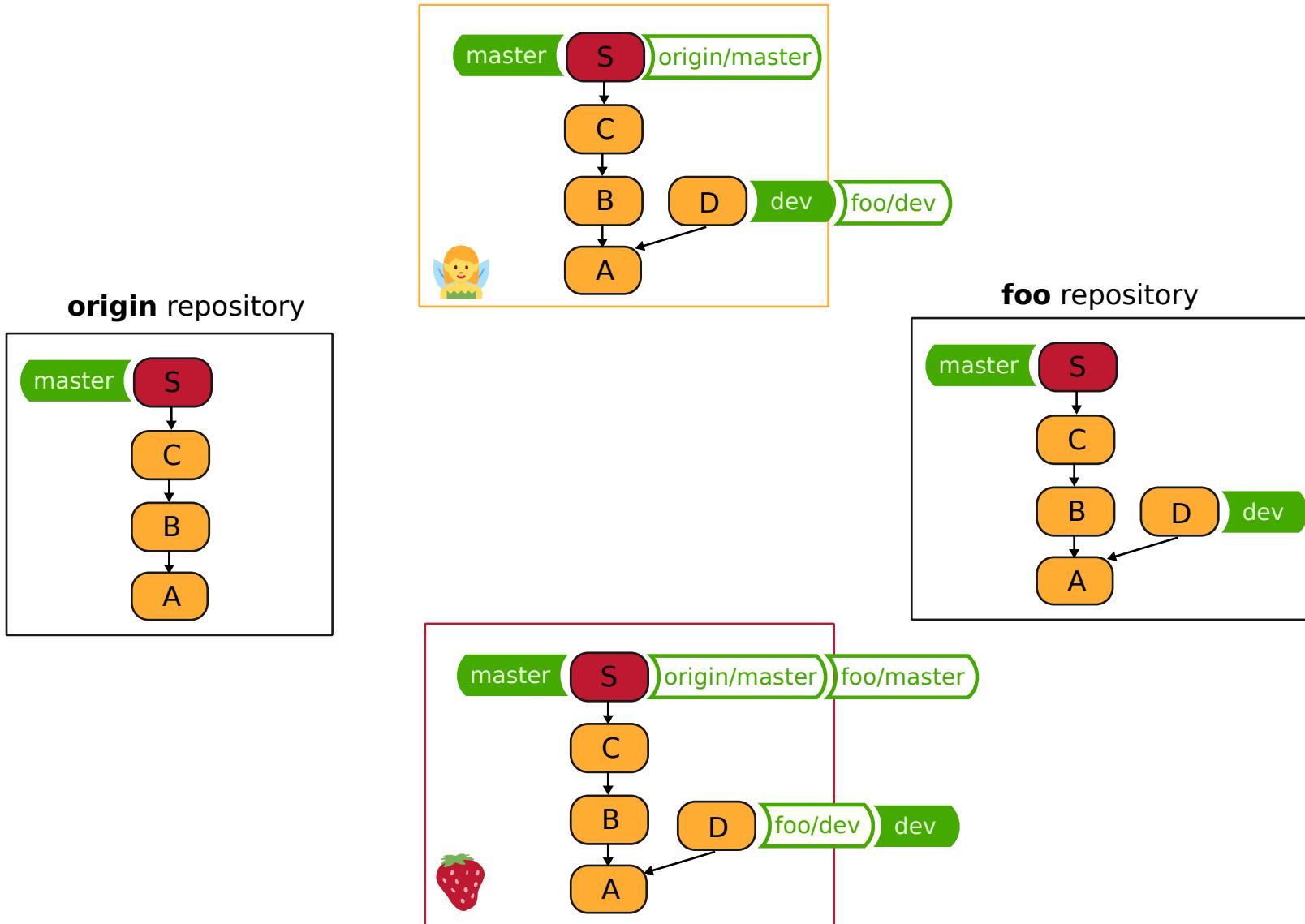
Remotes



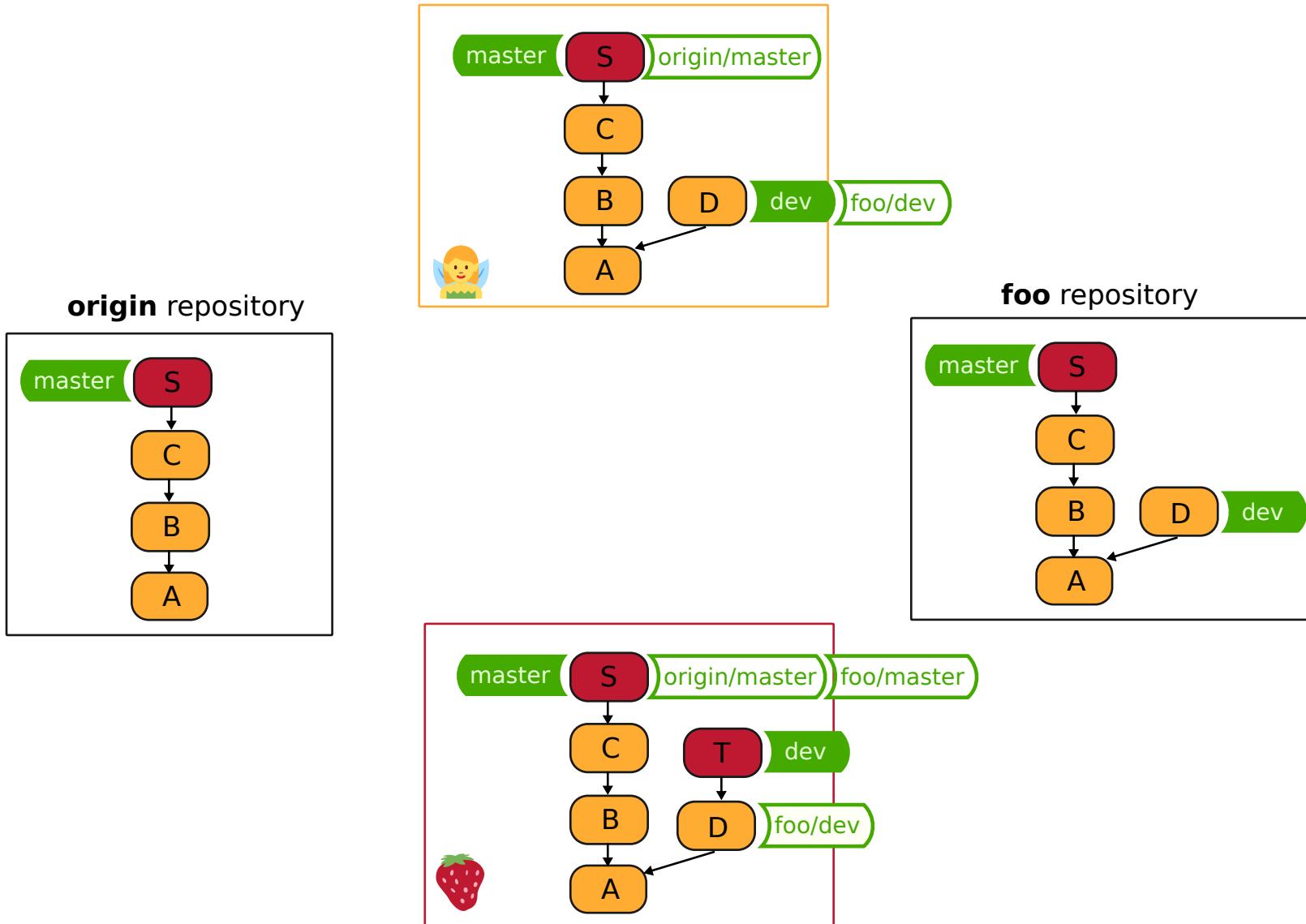
Remotes



Remotes



Remotes



Staging area

Commit

Branch

Merge

Conflict

Rebase

Squash / Reword

Stash

Fetch

Push

1. Introduction

2. Hands-on:

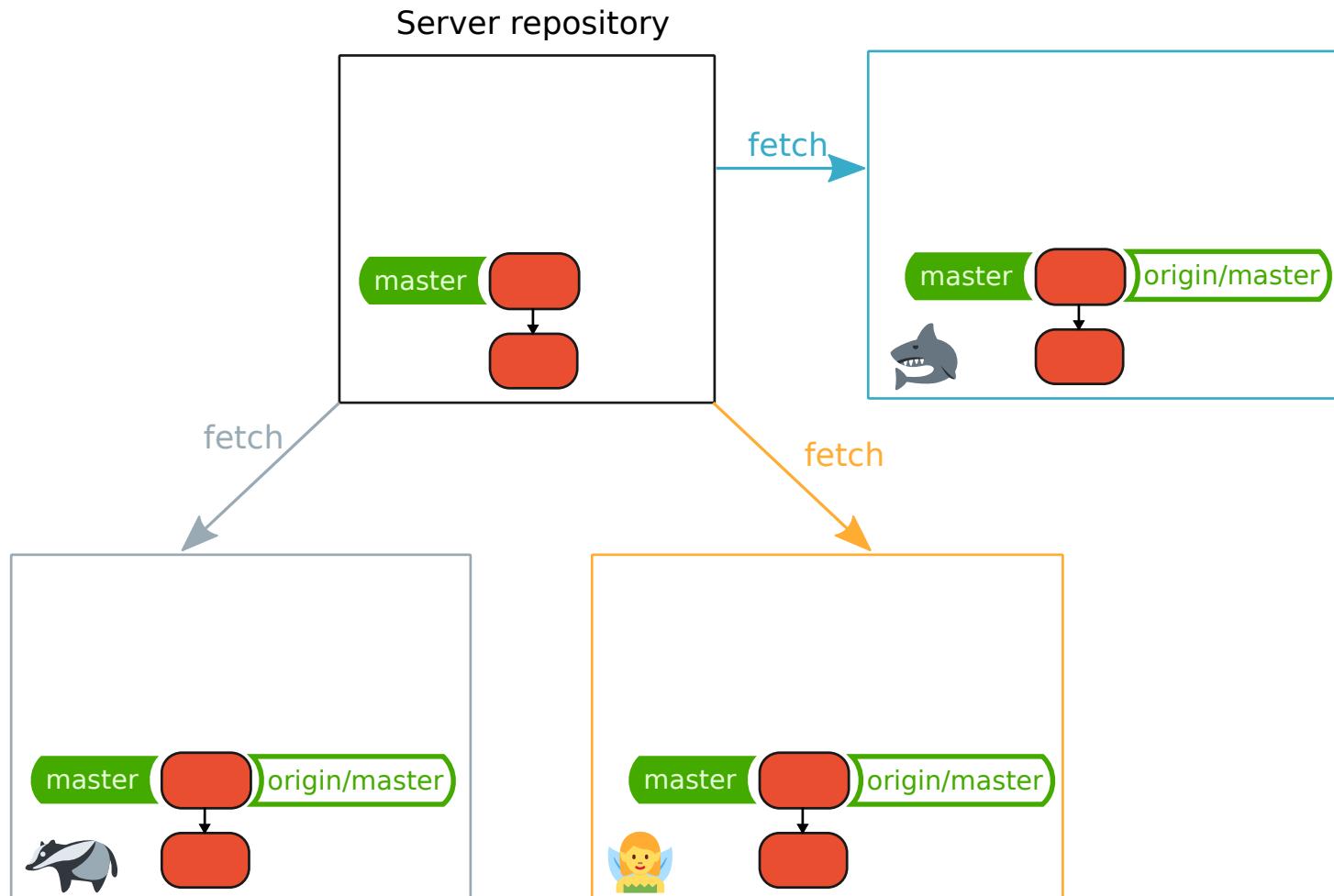
- Basic operations
- Branching / Merge / Rebase
- Remote repository

3. Common scenarios

4. Repository management

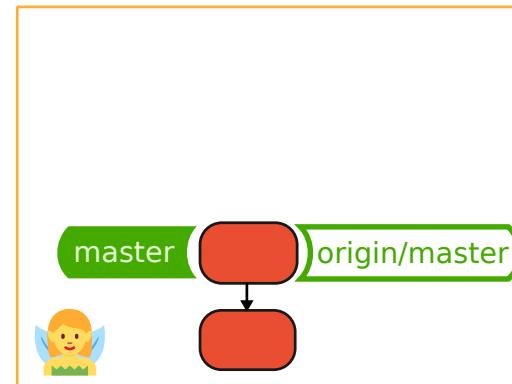
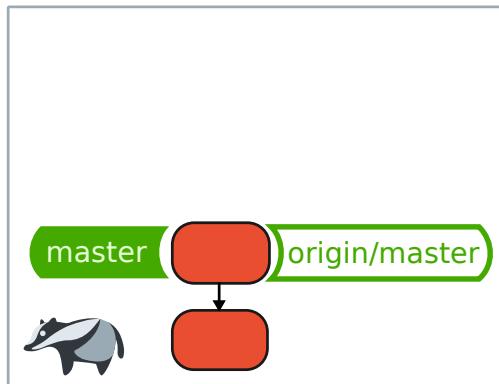
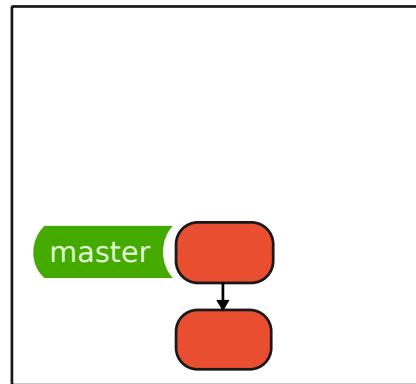
5. *Bonus topics*

Commit Race



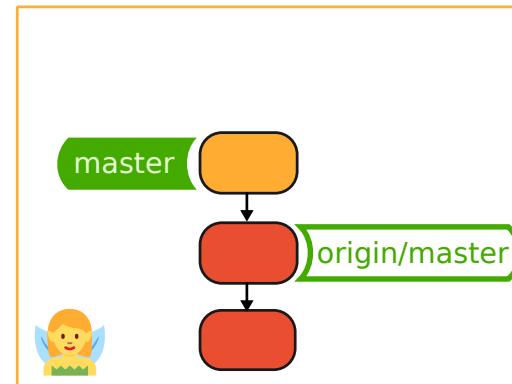
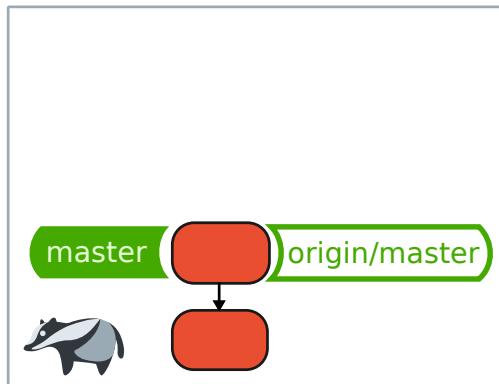
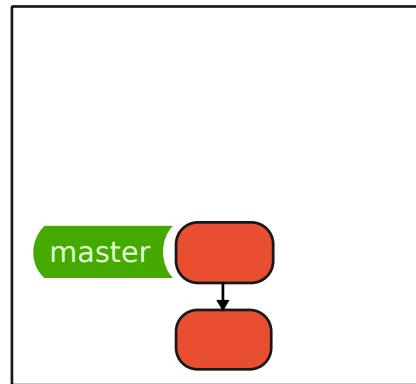
Commit Race

Server repository



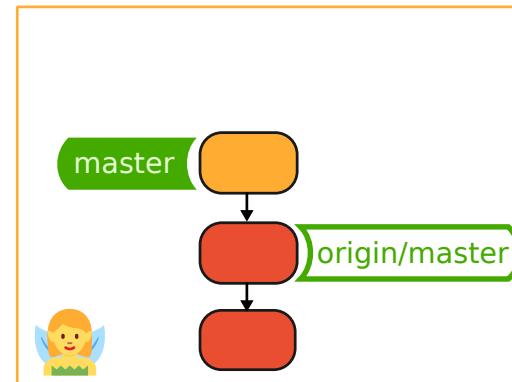
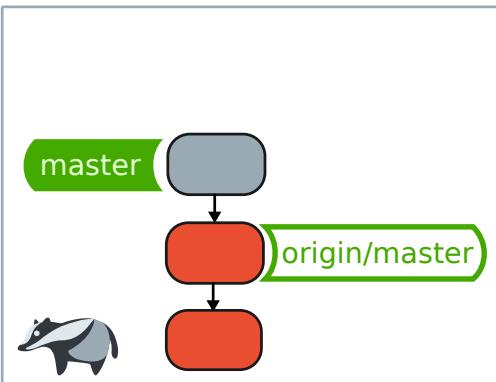
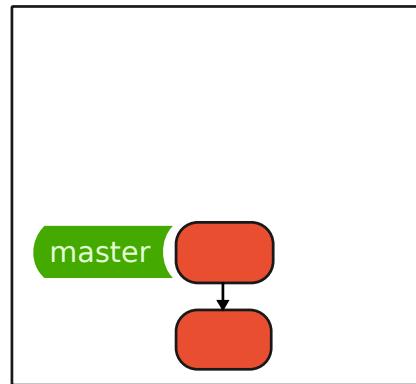
Commit Race

Server repository

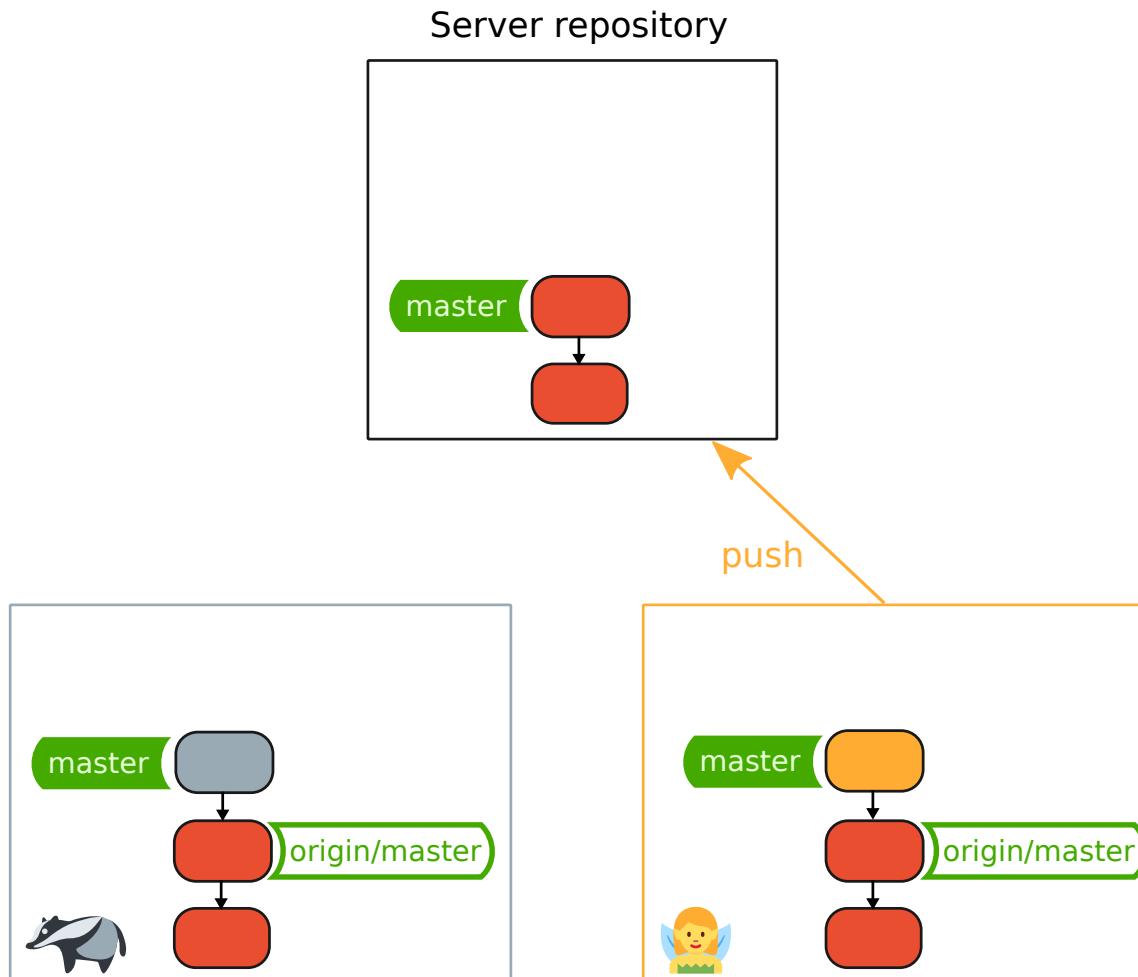


Commit Race

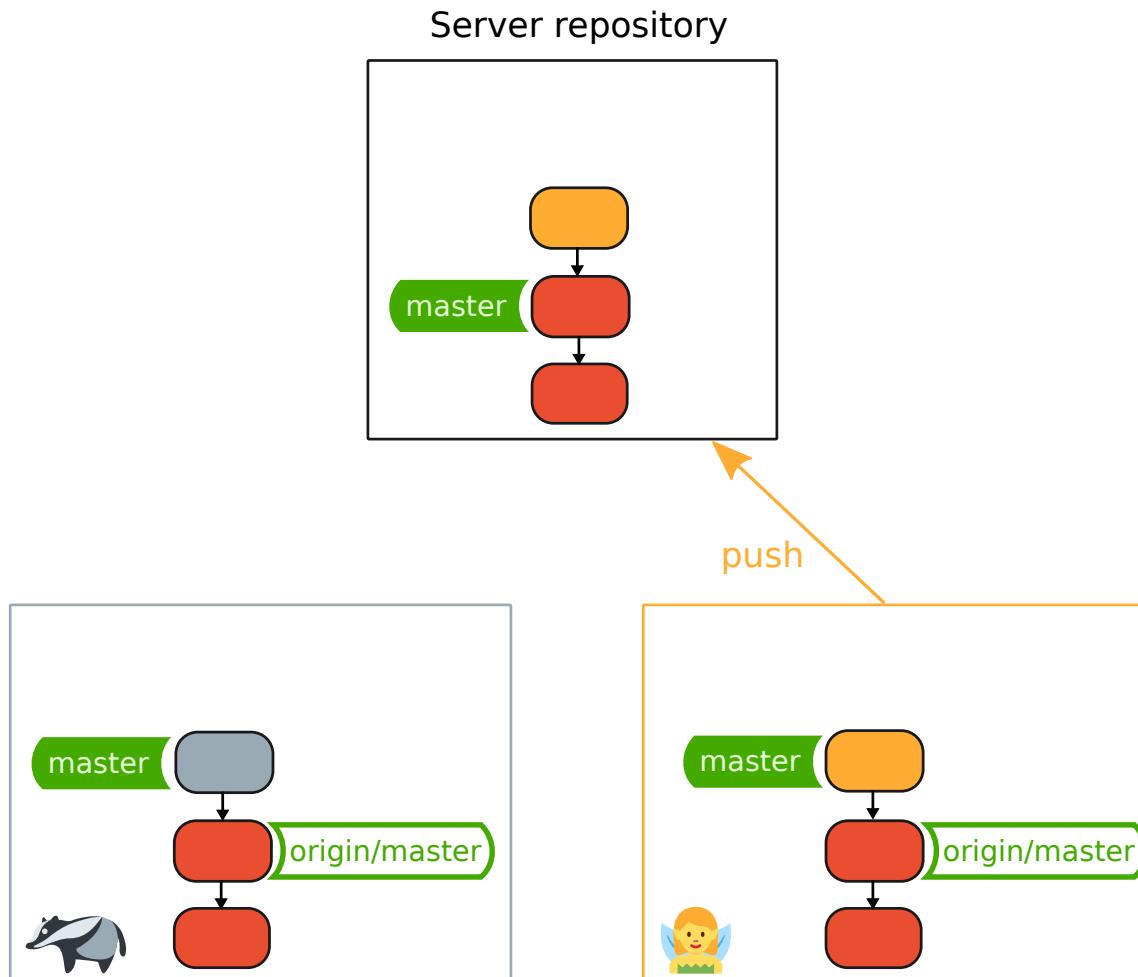
Server repository



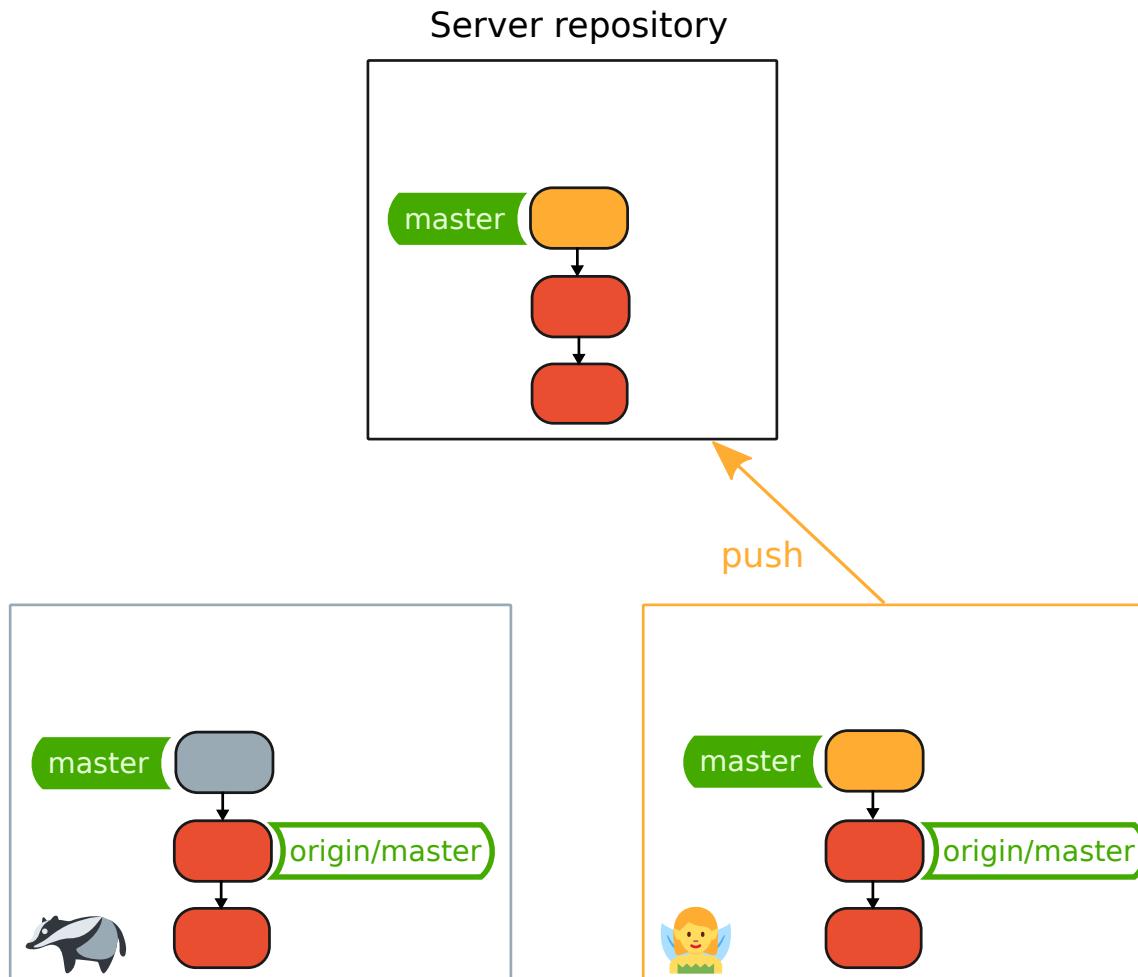
Commit Race



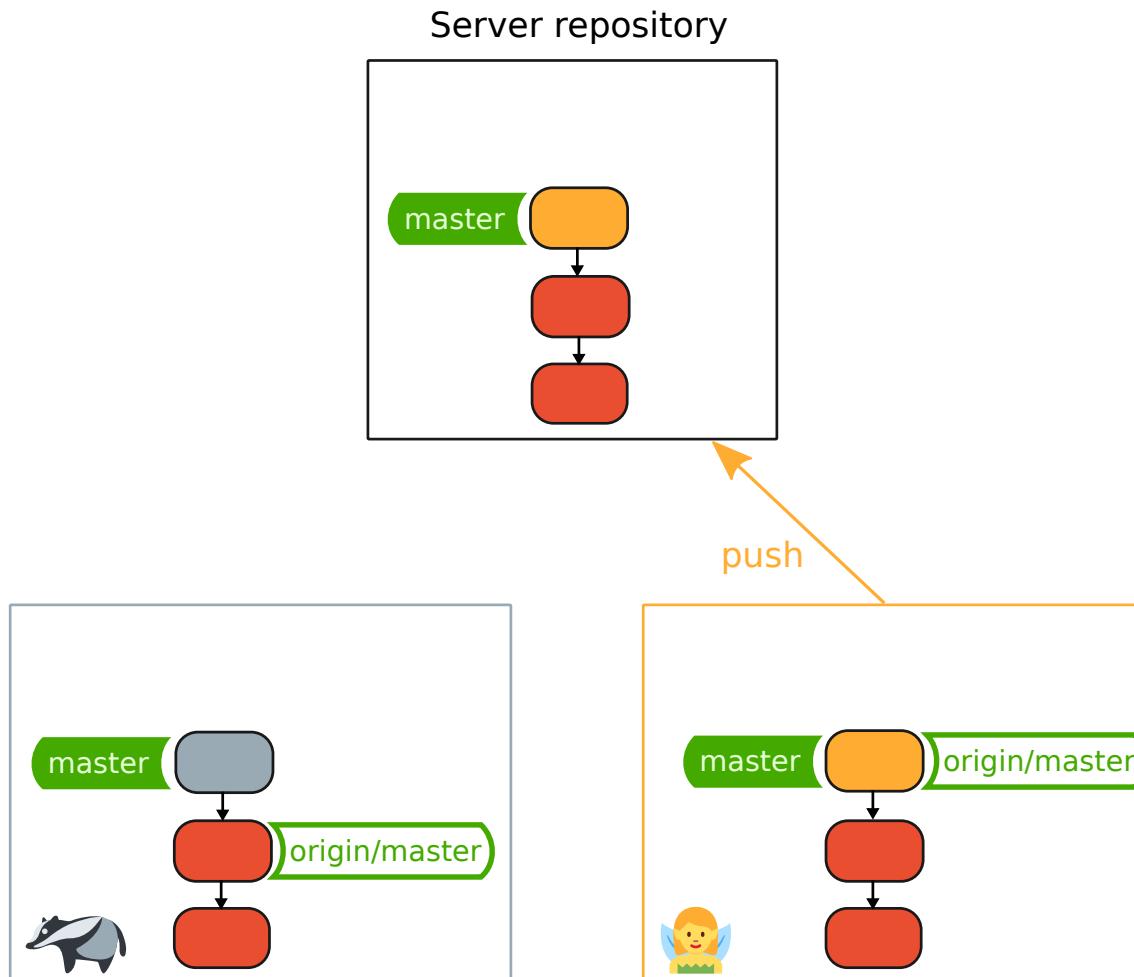
Commit Race



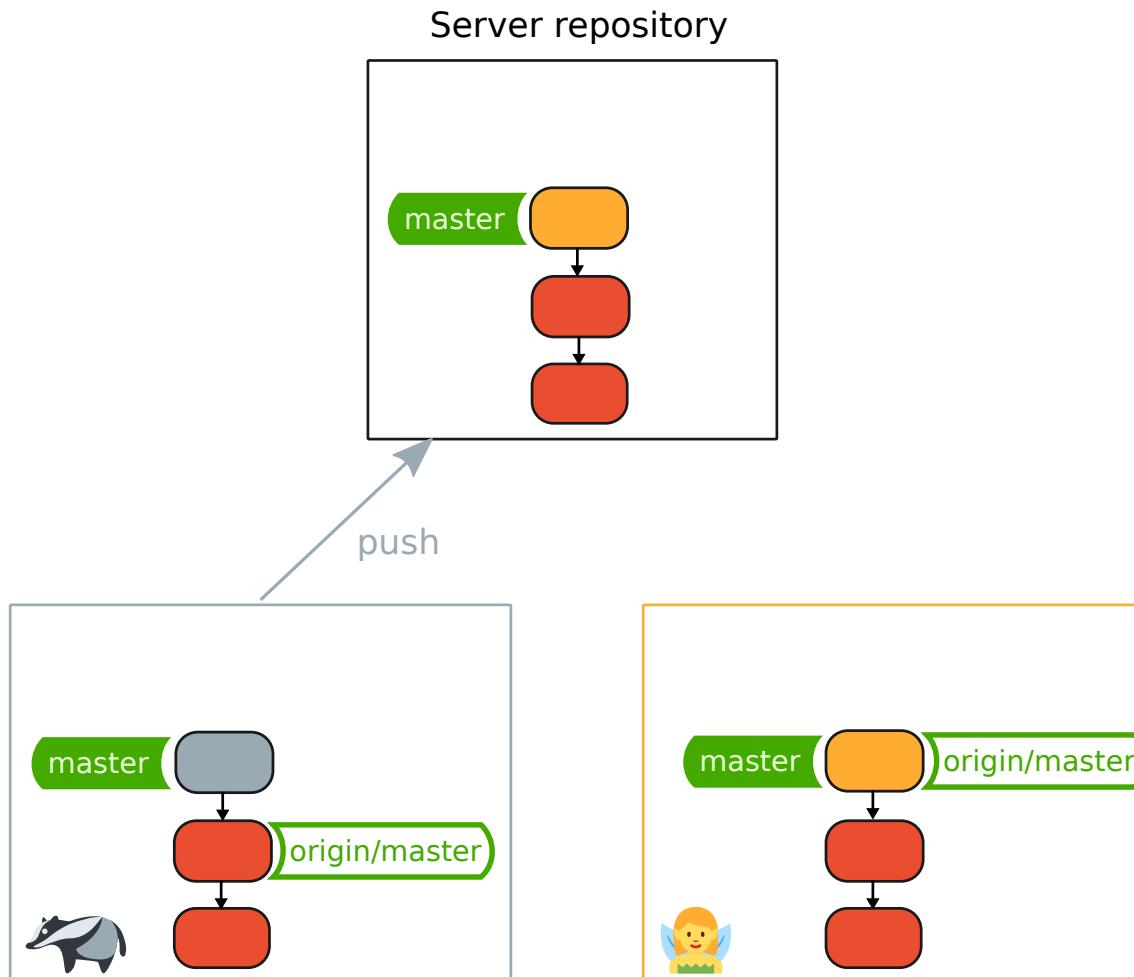
Commit Race



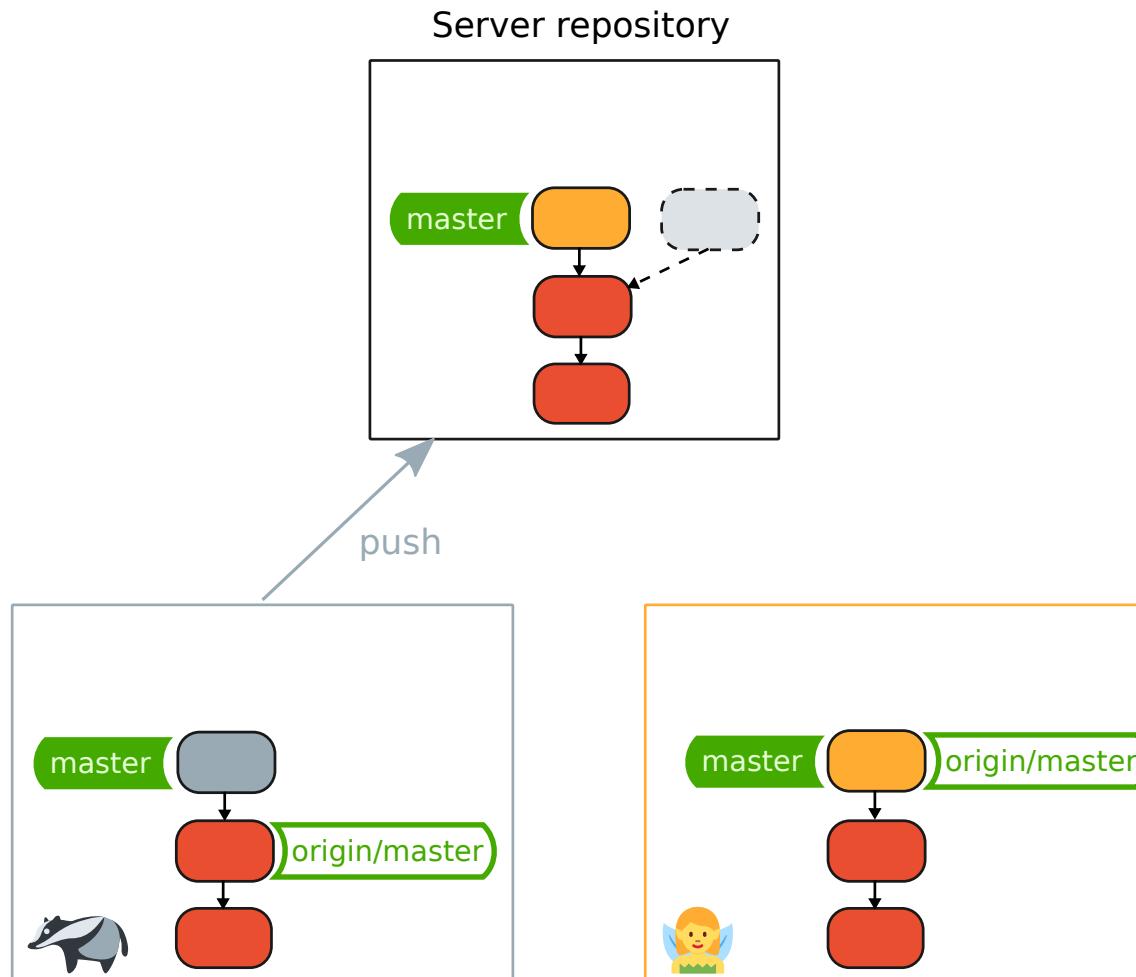
Commit Race



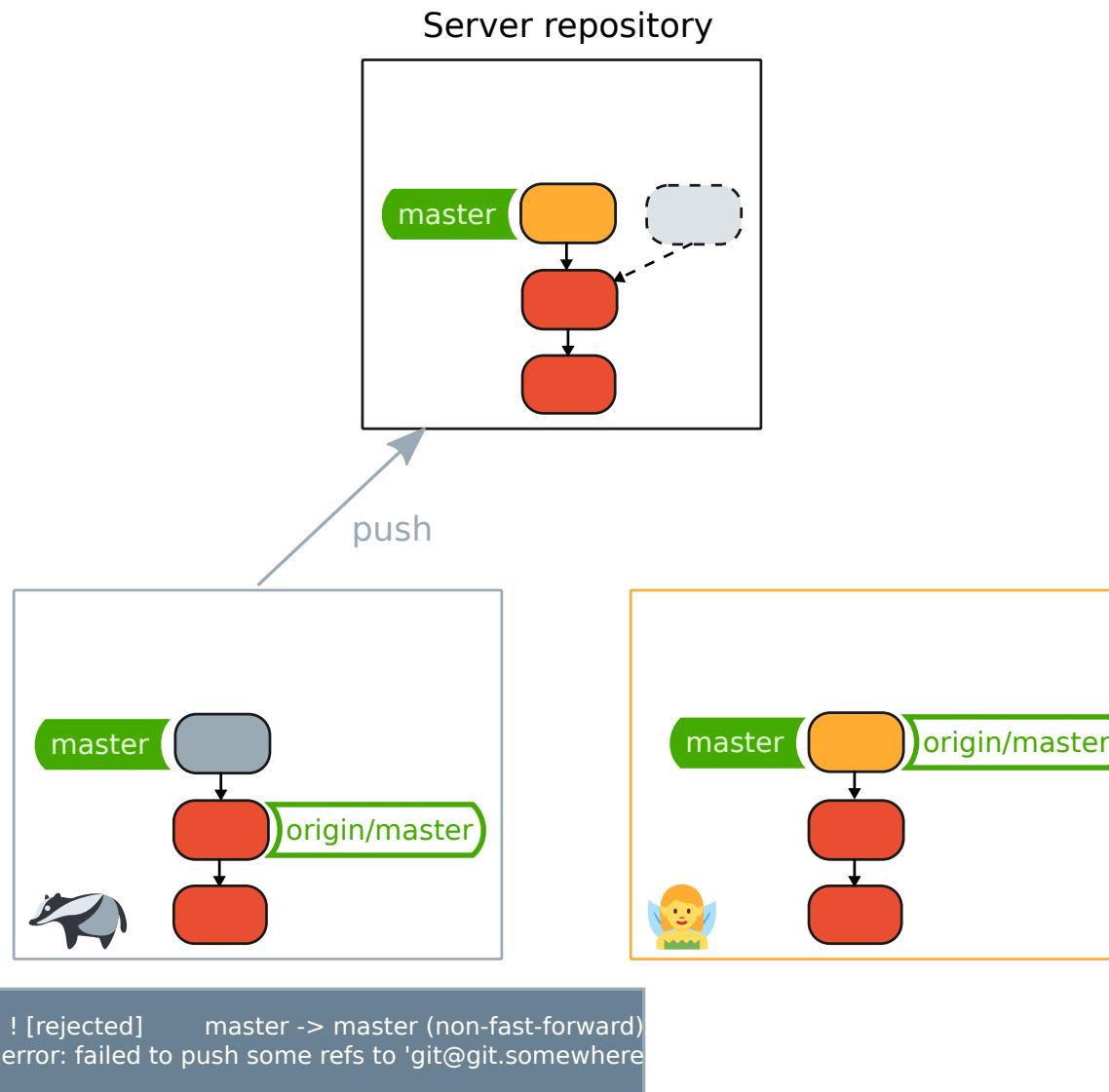
Commit Race



Commit Race

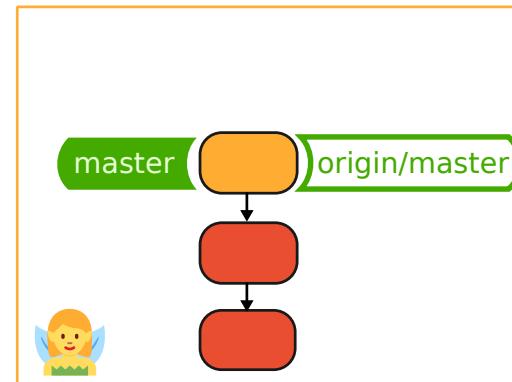
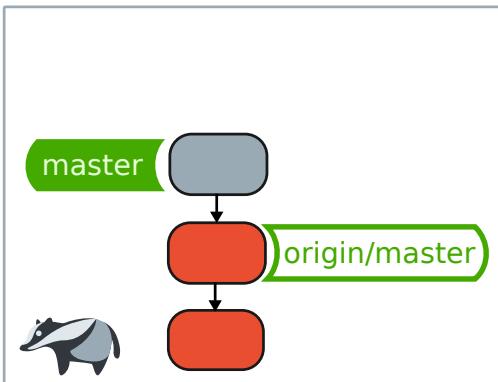
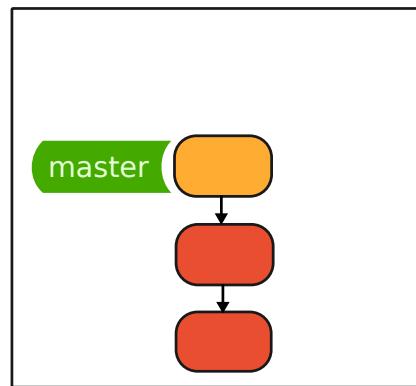


Commit Race

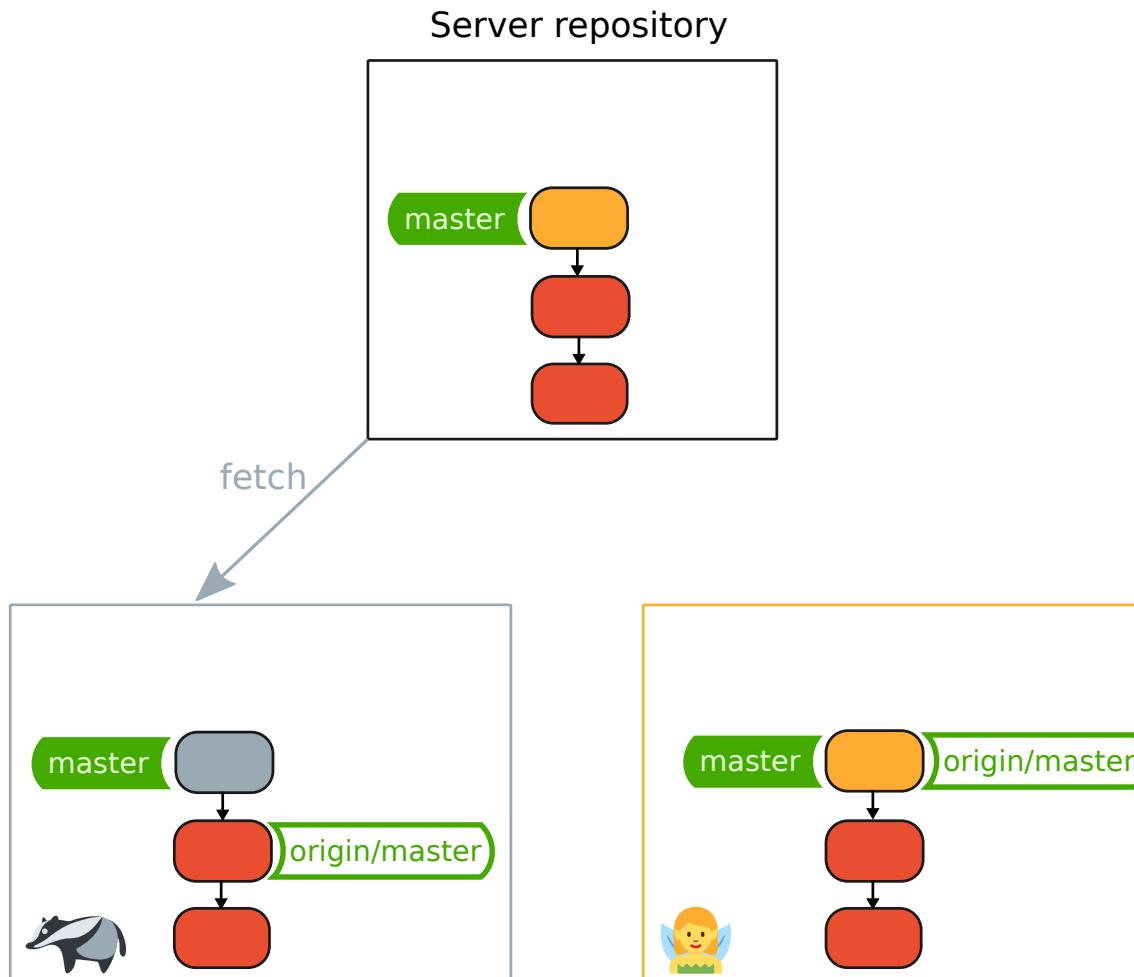


Commit Race

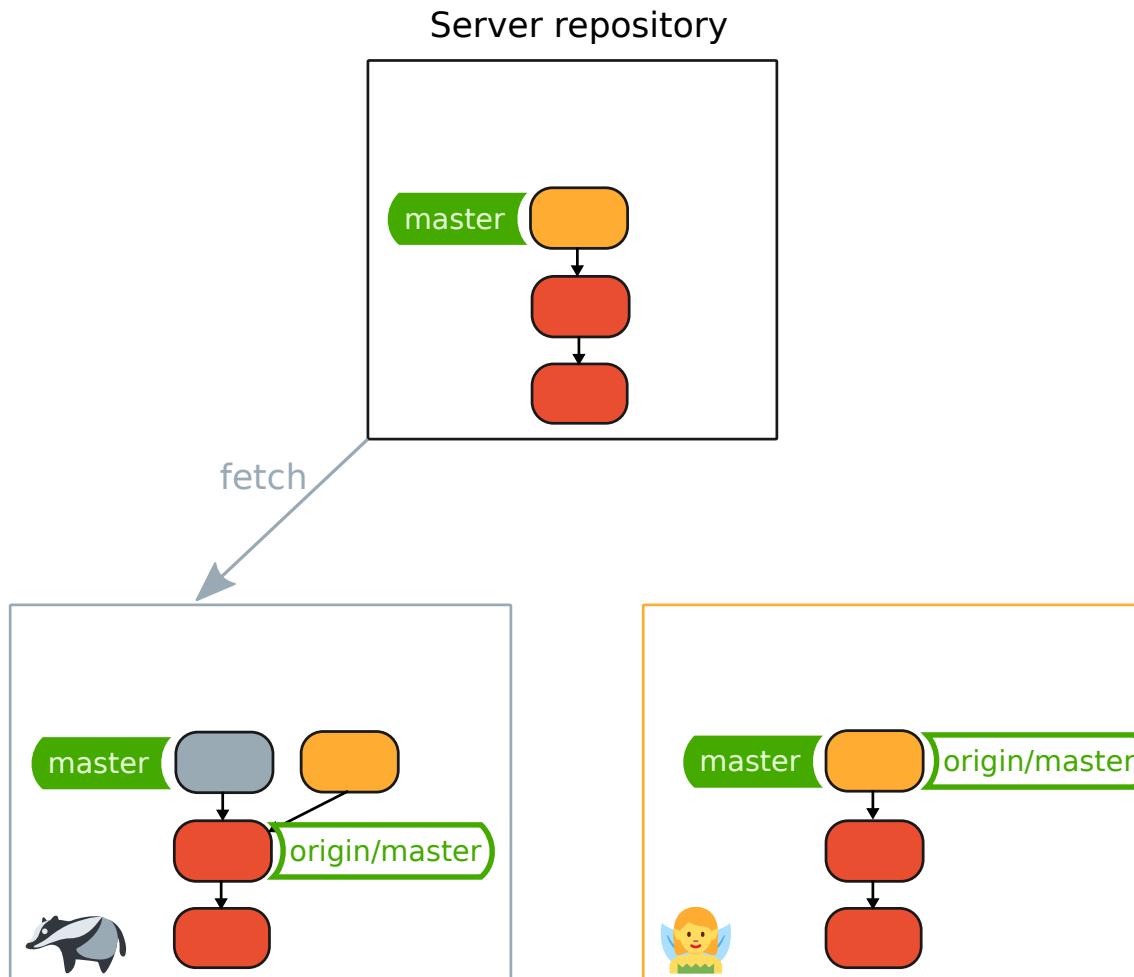
Server repository



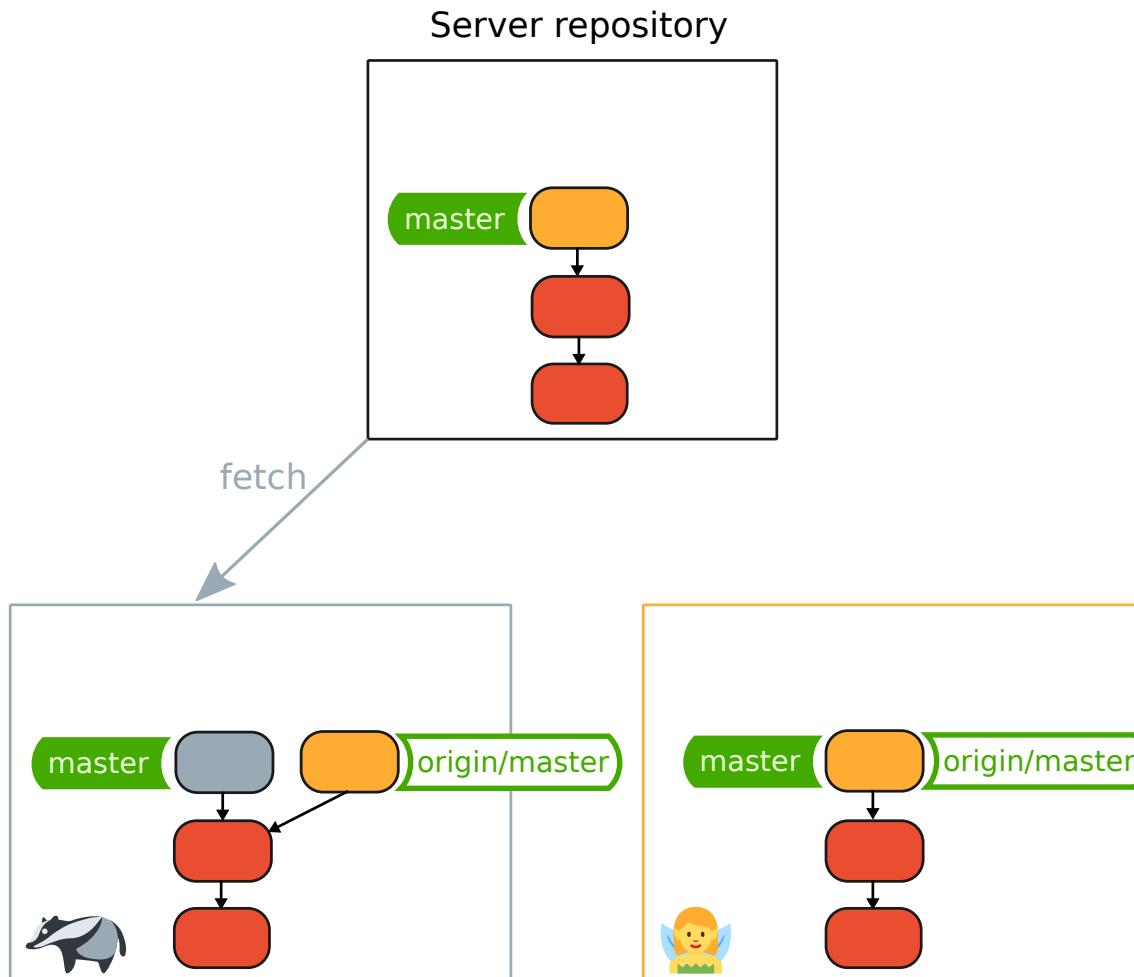
Commit Race



Commit Race

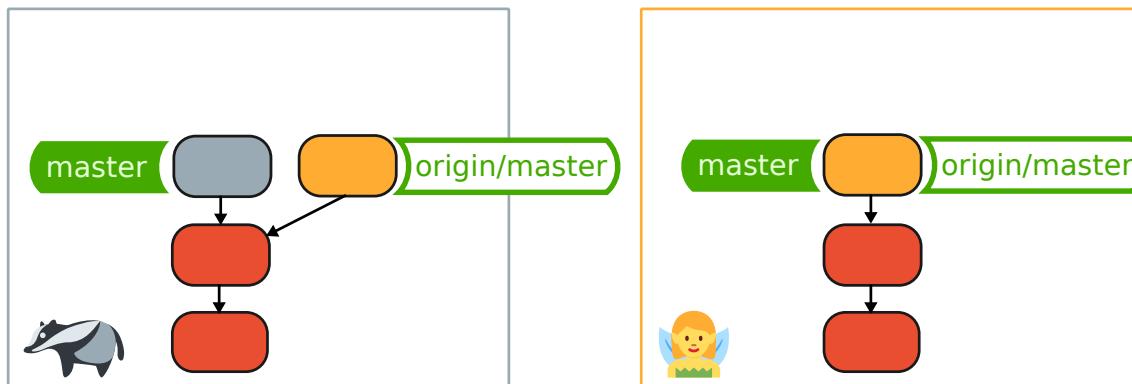
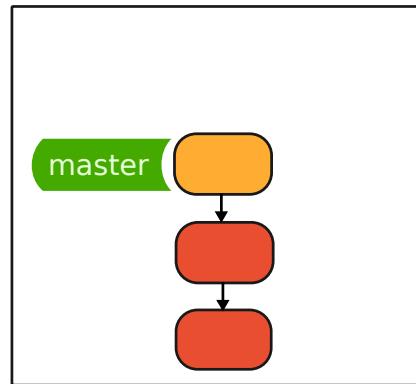


Commit Race



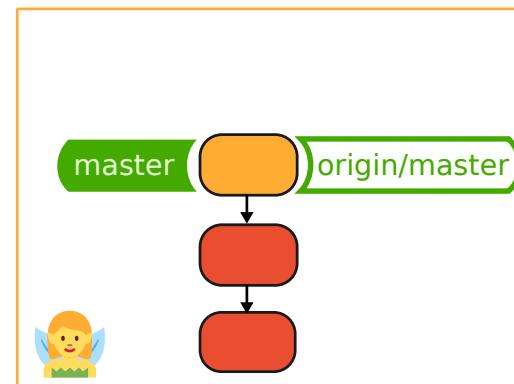
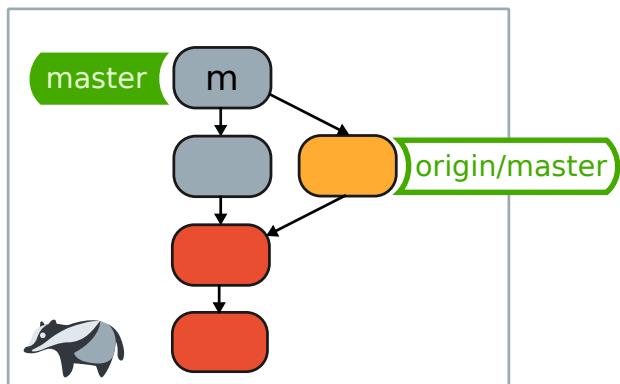
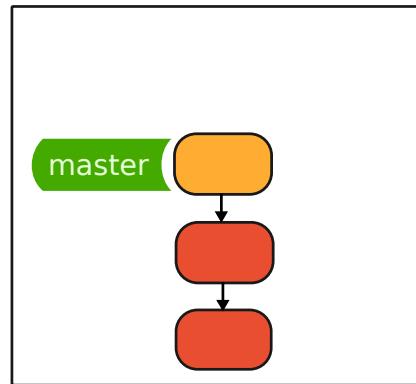
Commit Race

Server repository

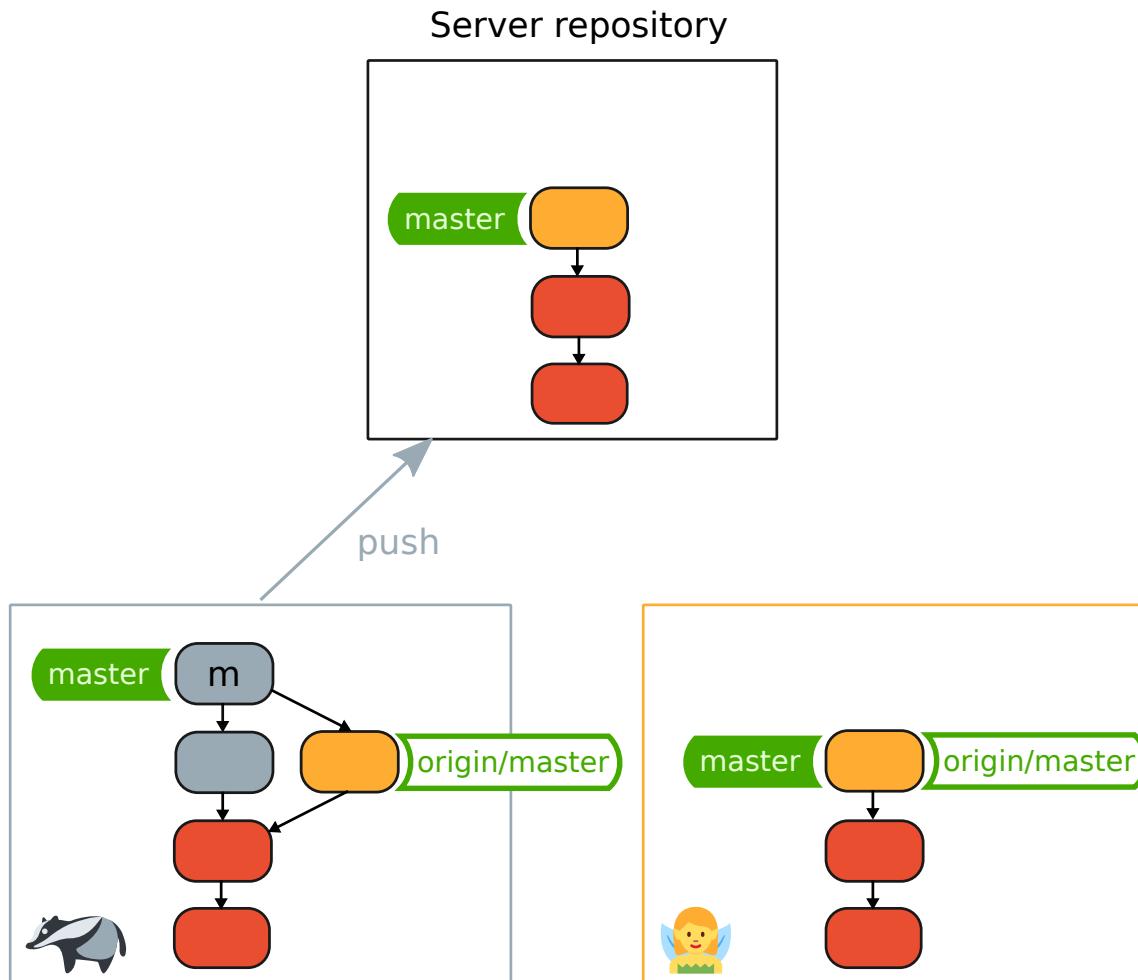


Commit Race

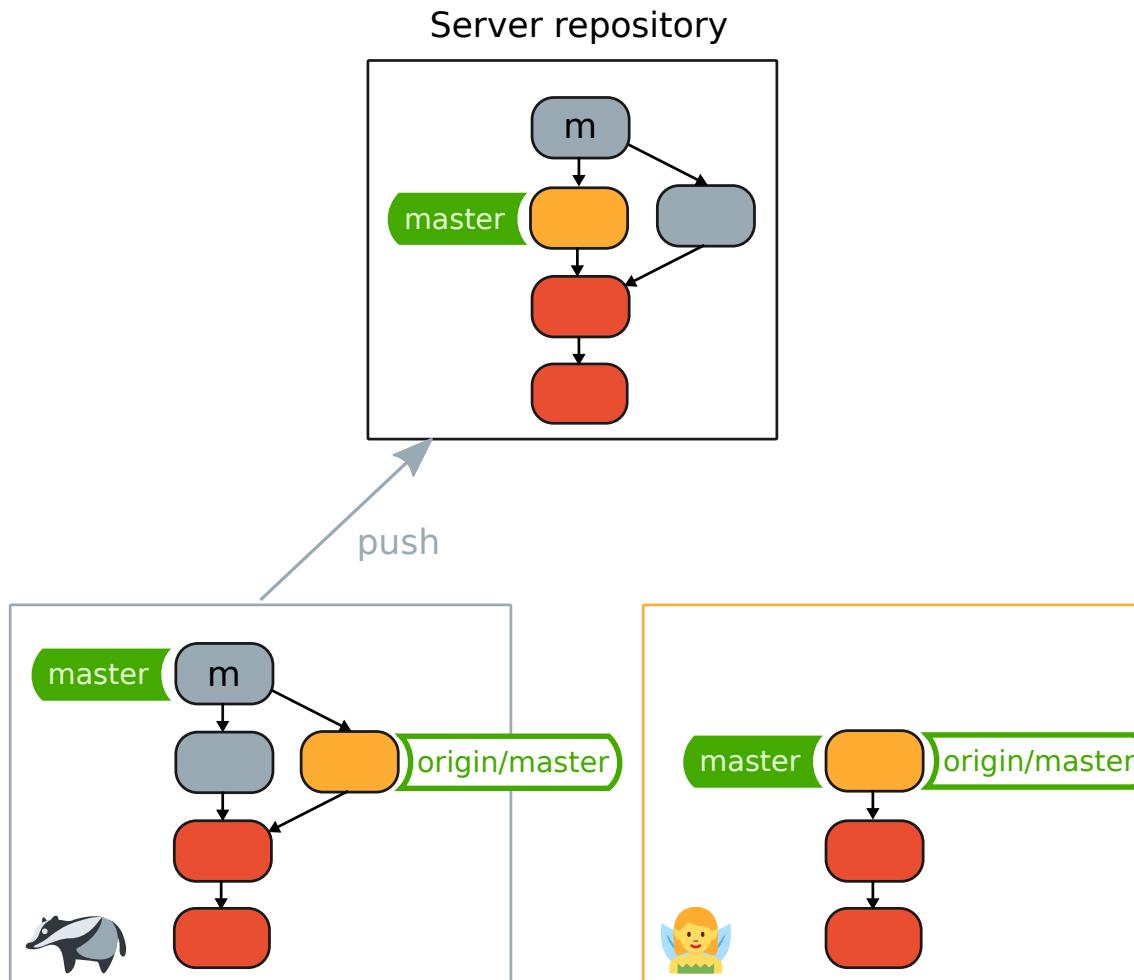
Server repository



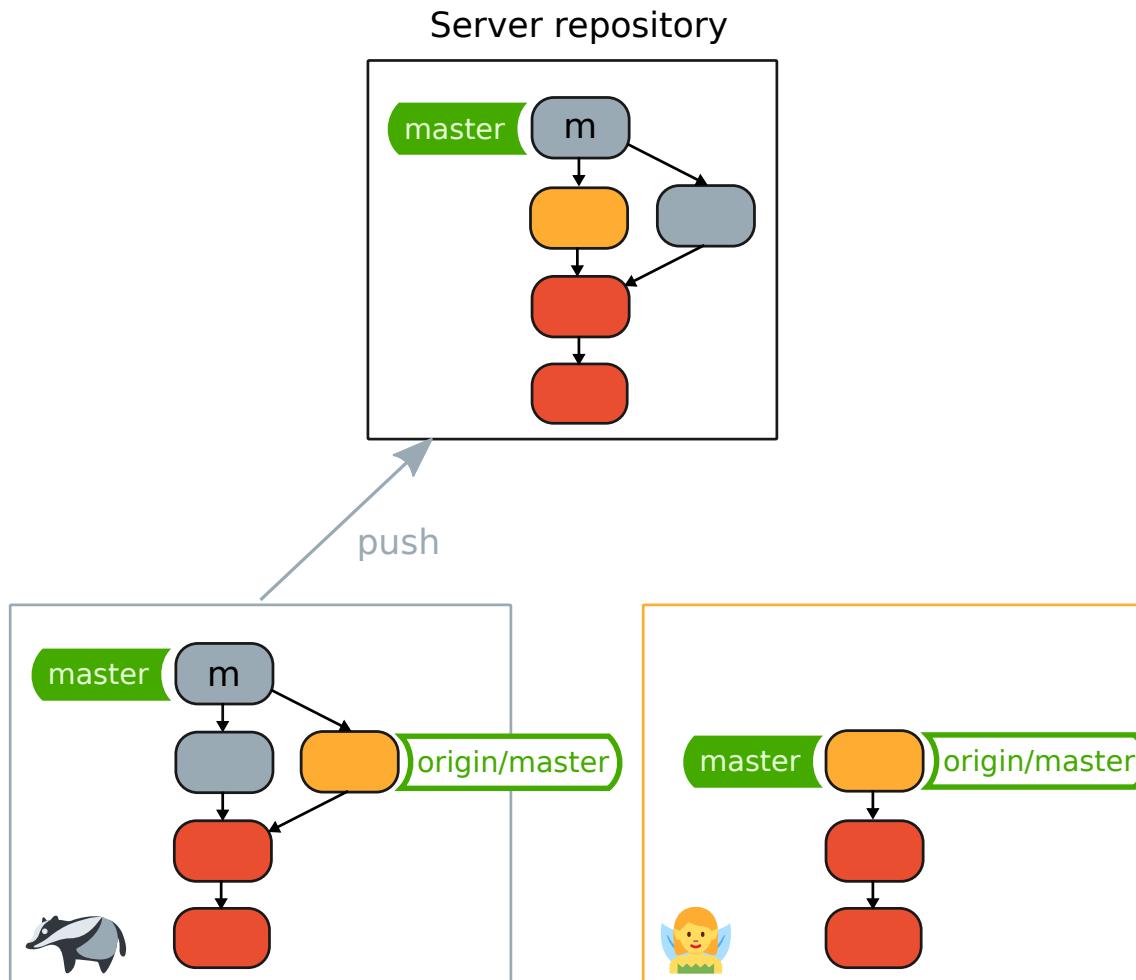
Commit Race



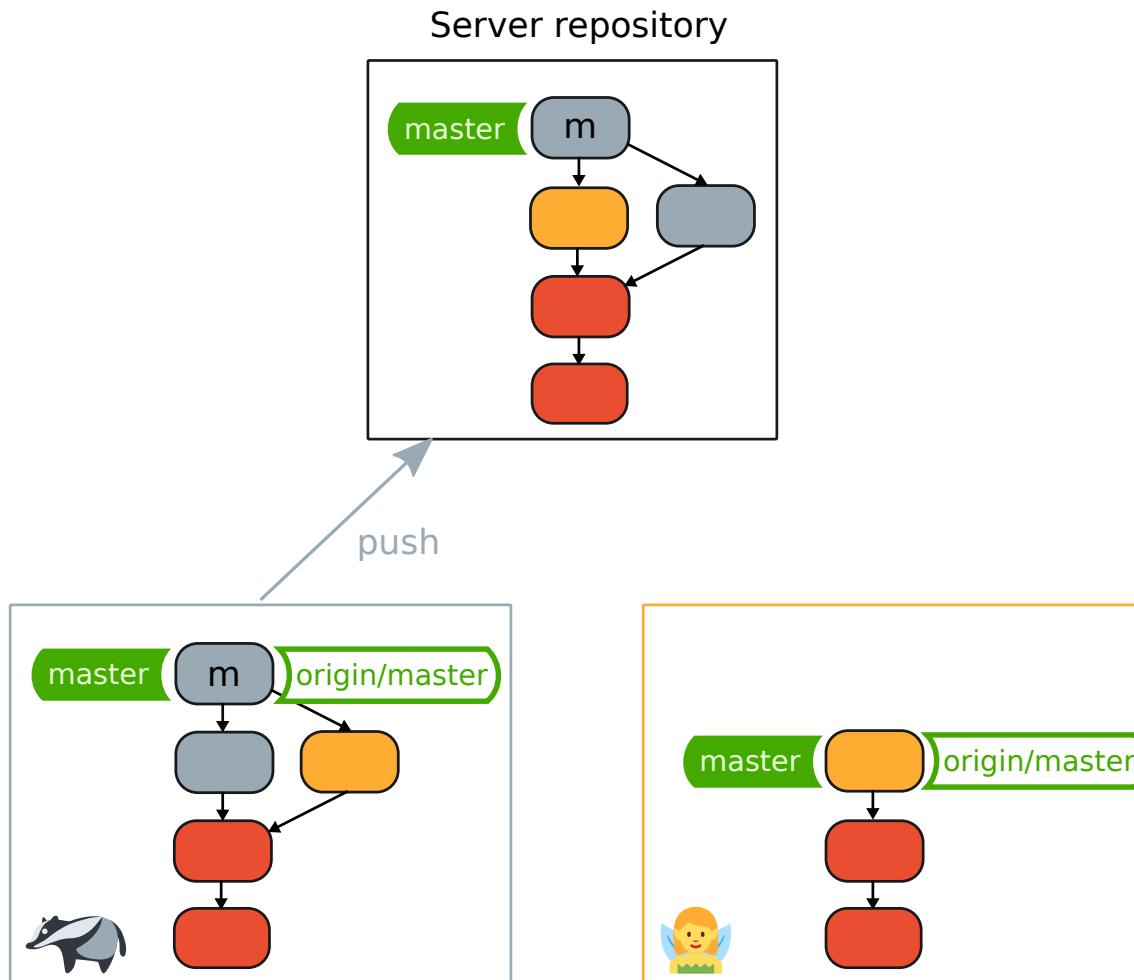
Commit Race



Commit Race

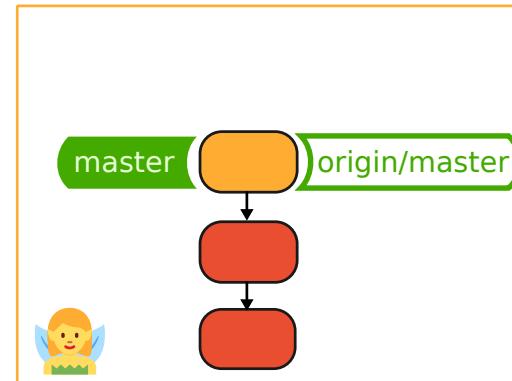
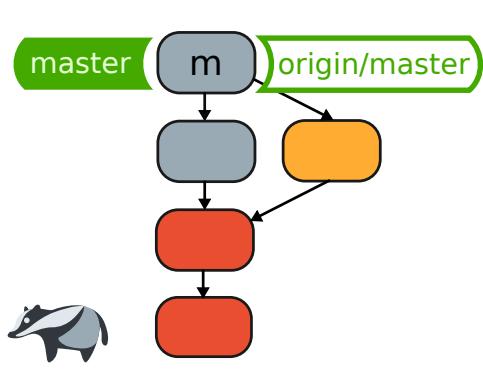
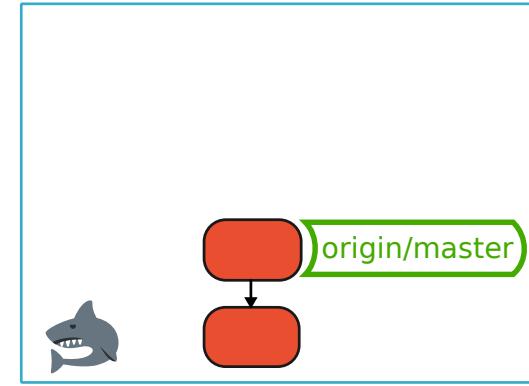
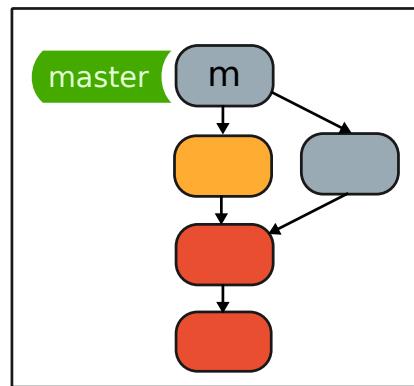


Commit Race



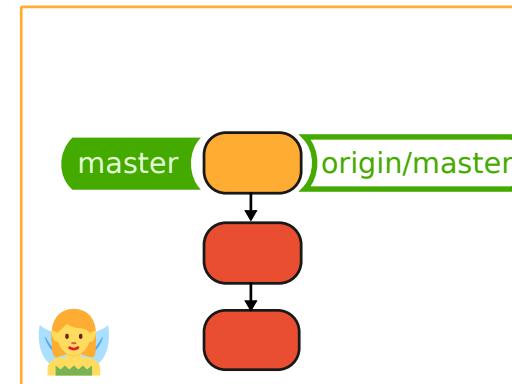
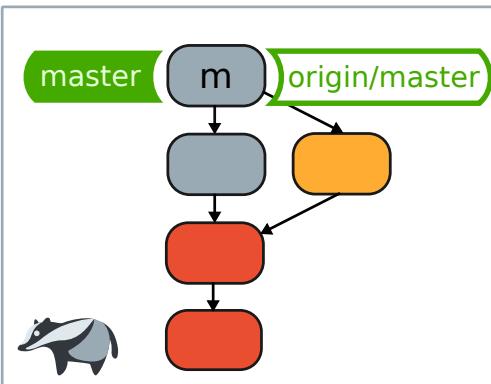
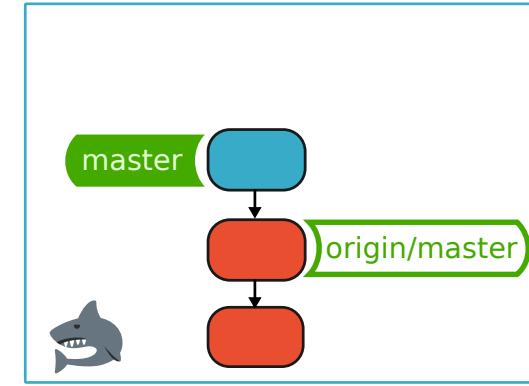
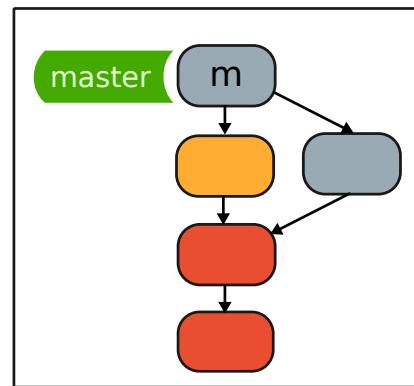
Commit Race

Server repository



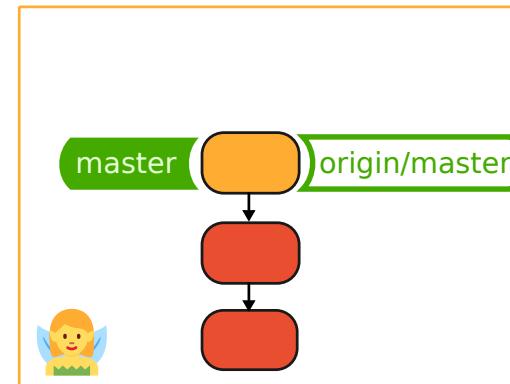
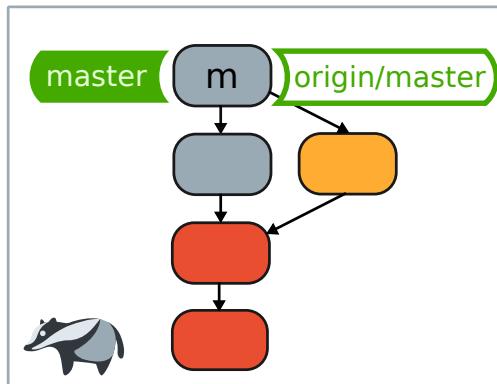
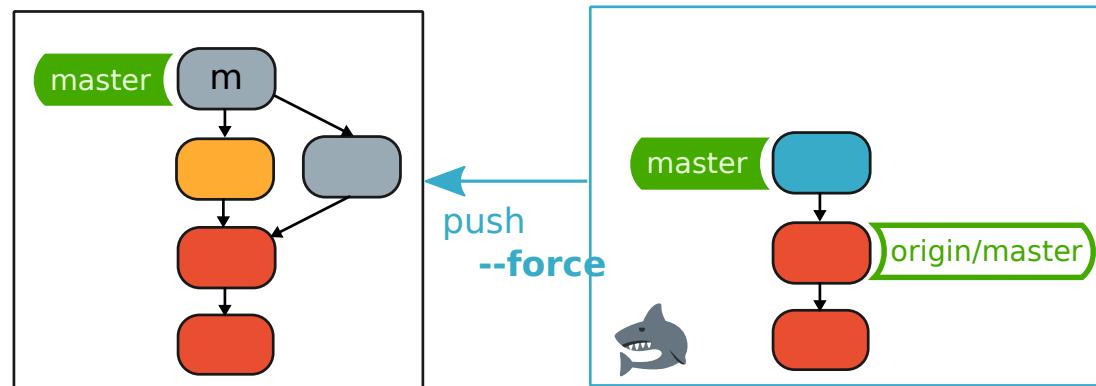
Commit Race

Server repository



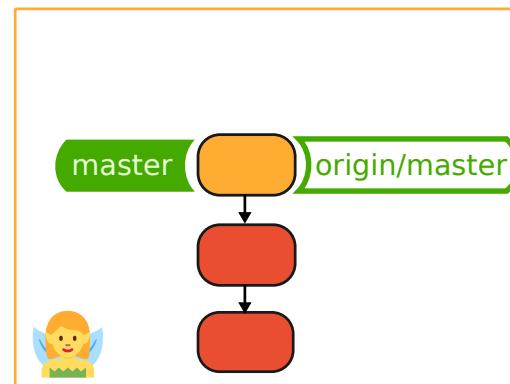
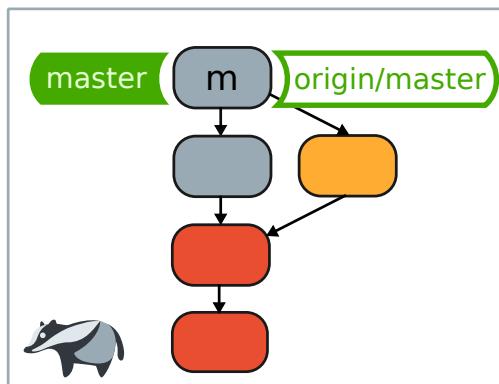
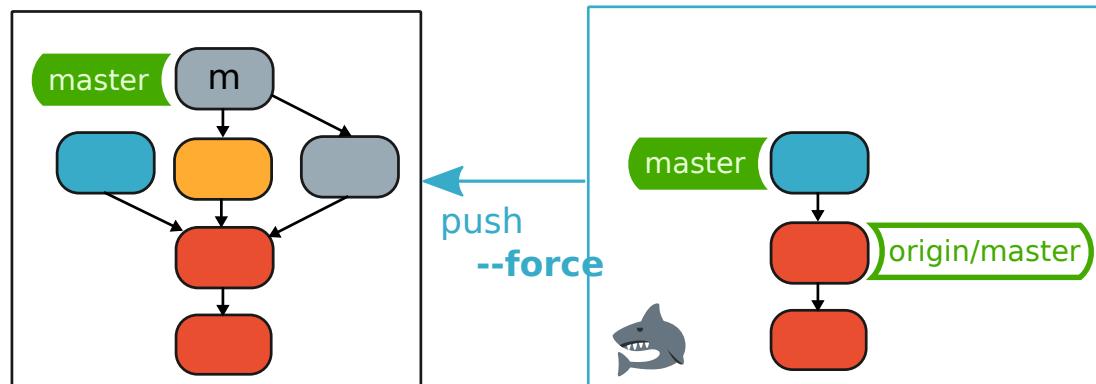
Commit Race

Server repository



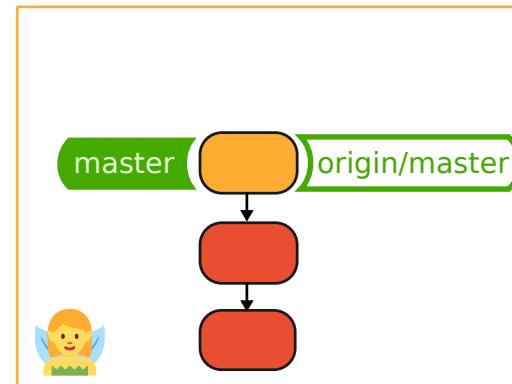
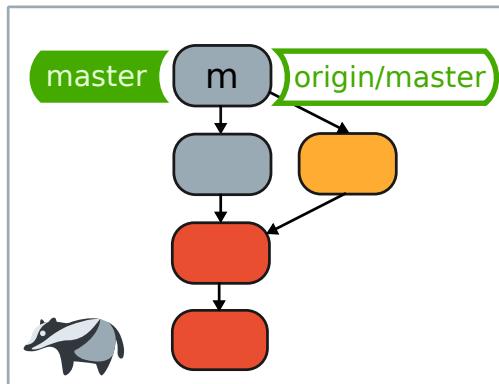
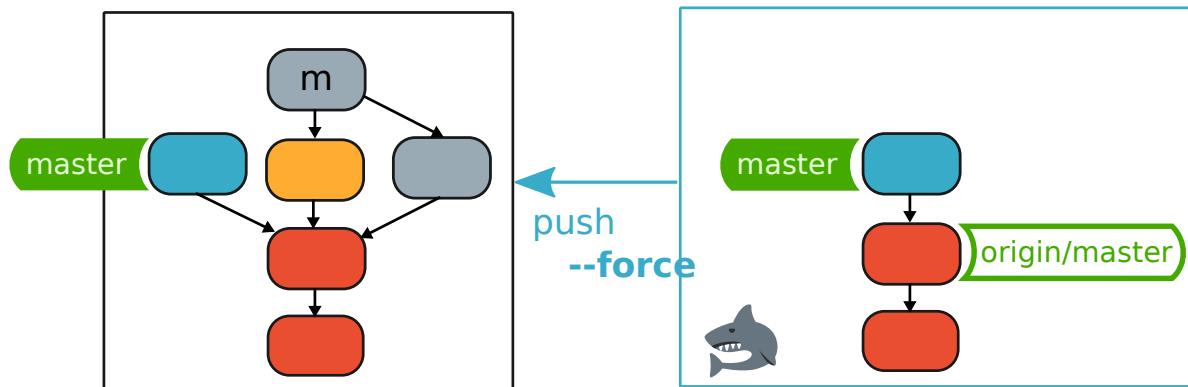
Commit Race

Server repository



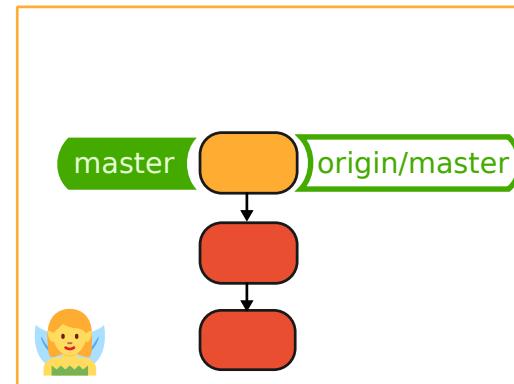
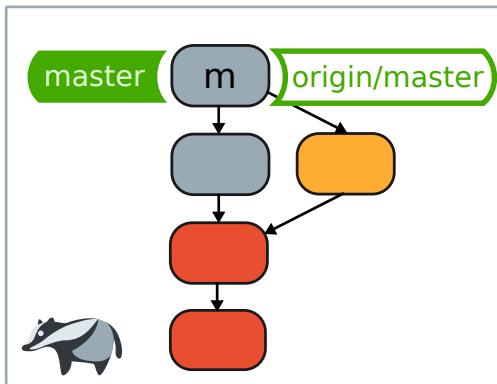
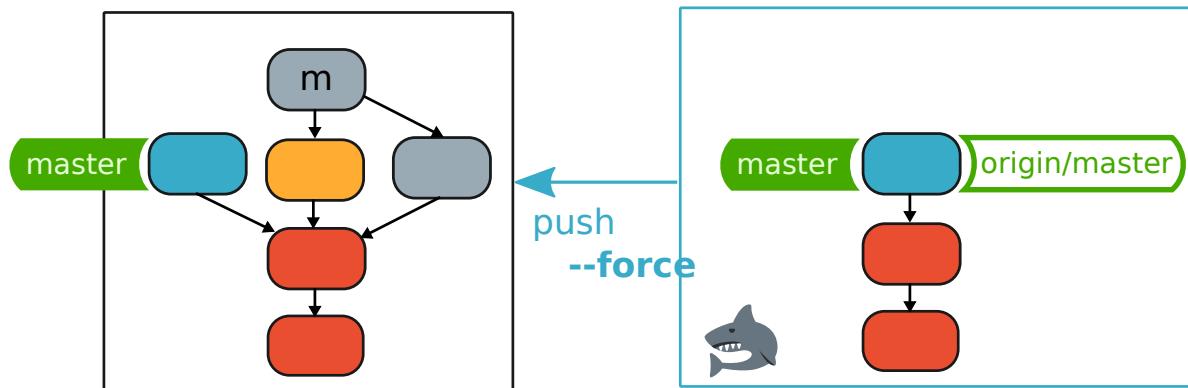
Commit Race

Server repository

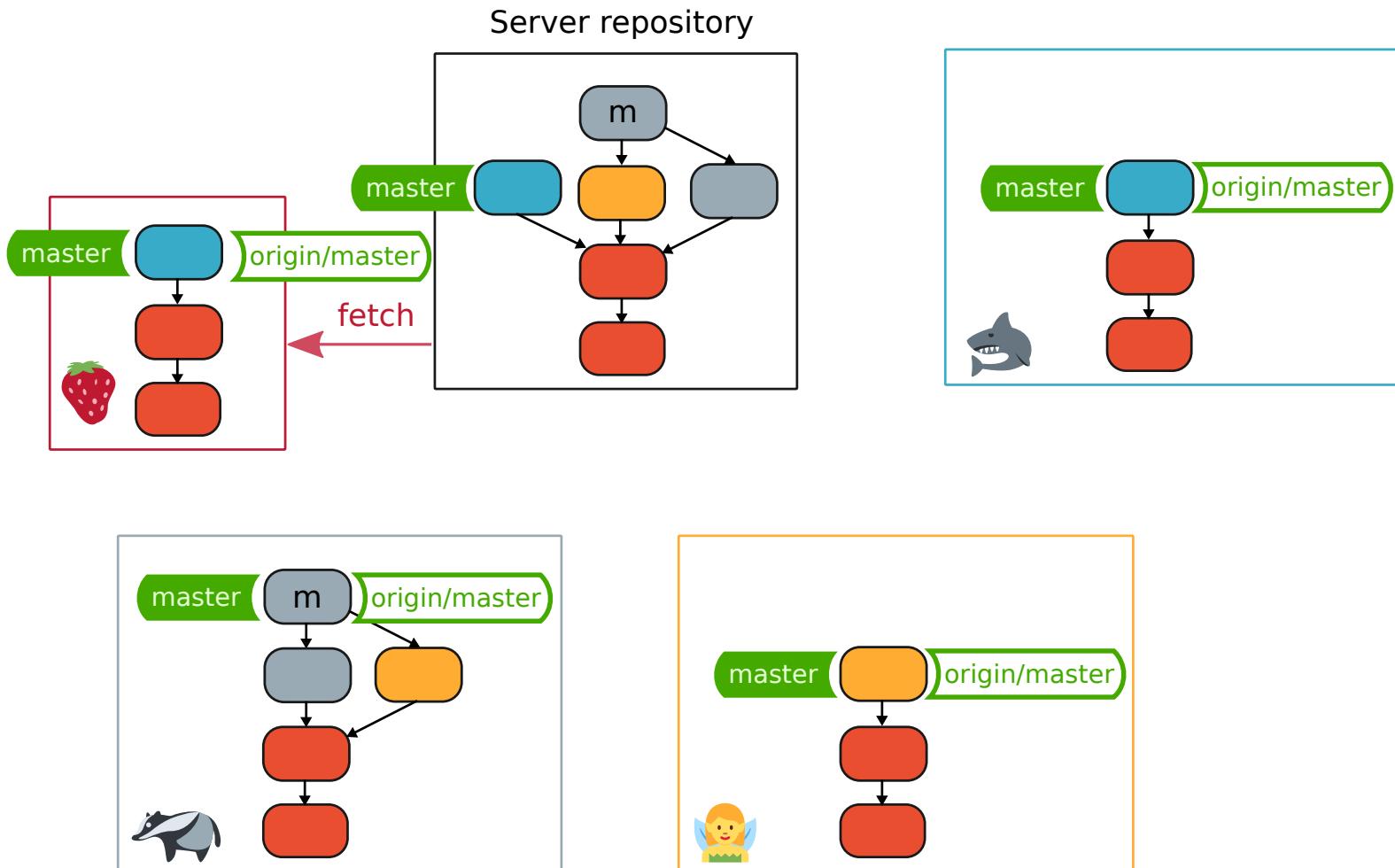


Commit Race

Server repository

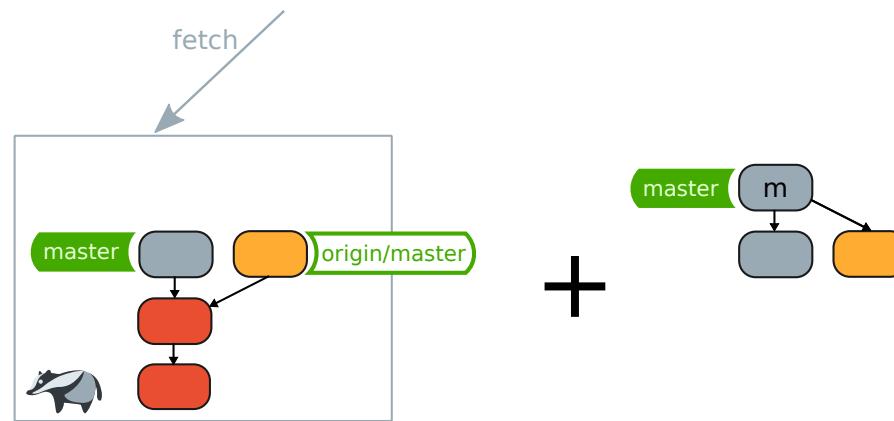


Commit Race

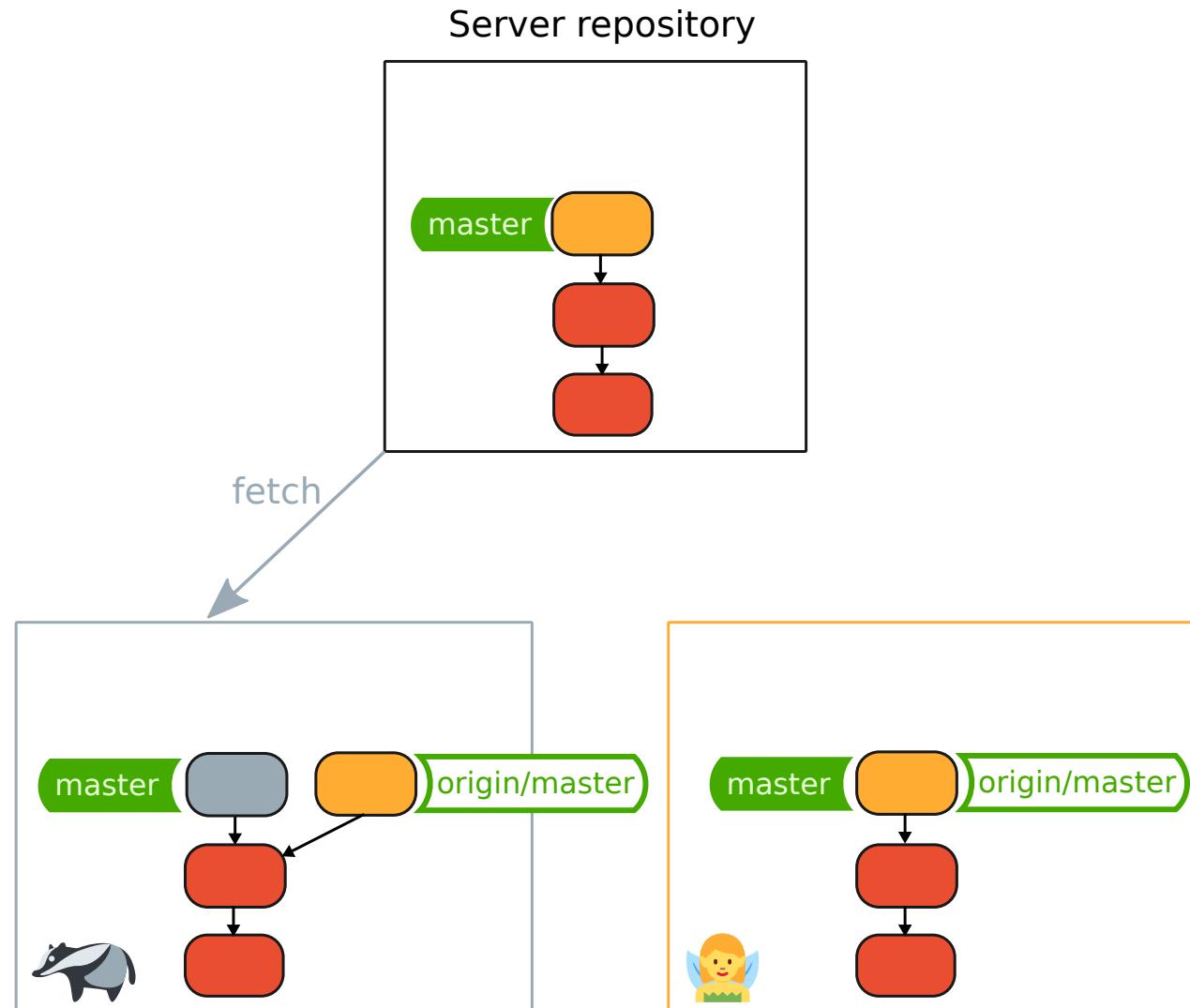


pull = fetch + merge

pull = fetch + merge

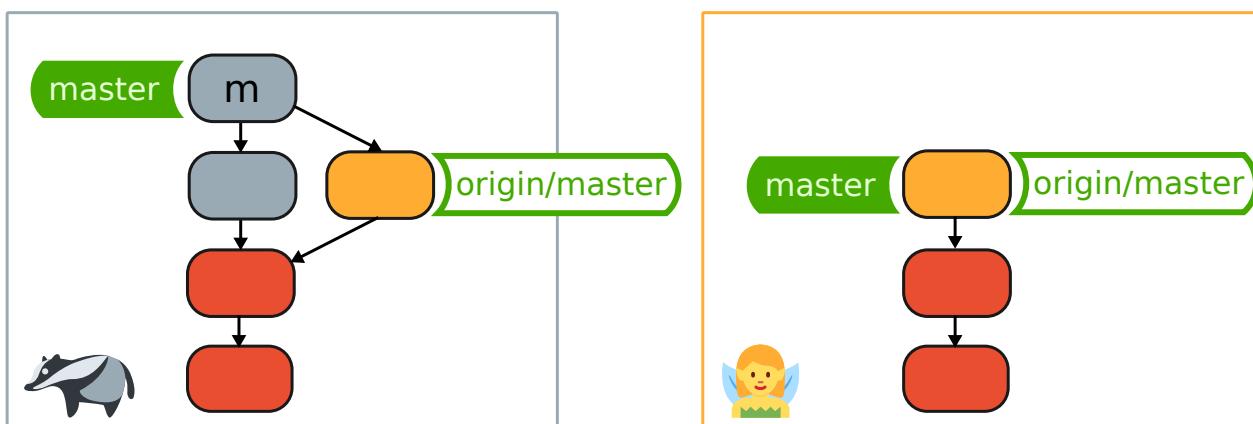
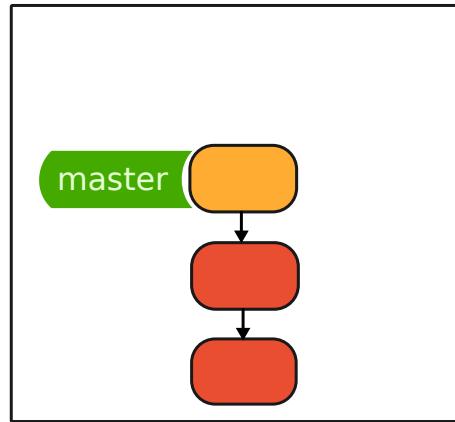


Rebase



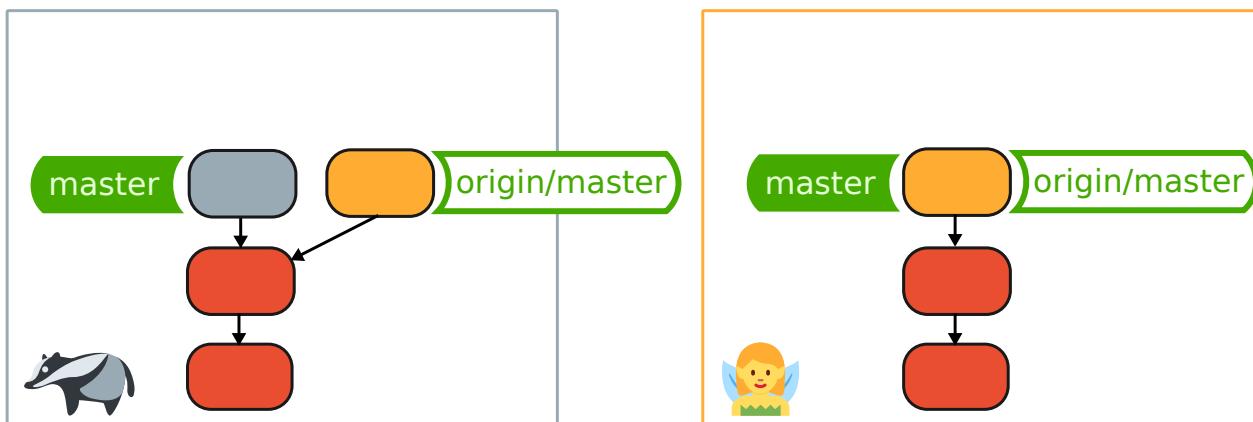
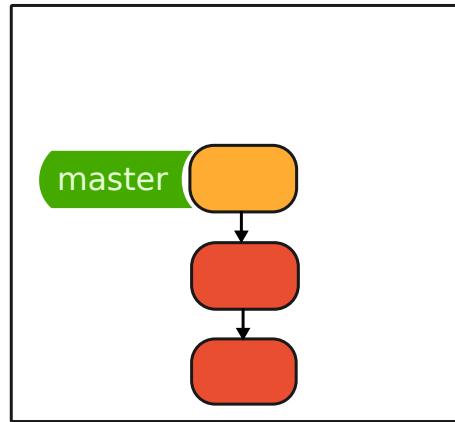
Rebase

Server repository



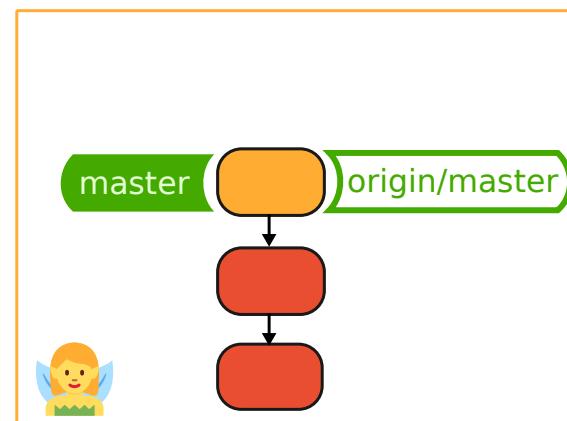
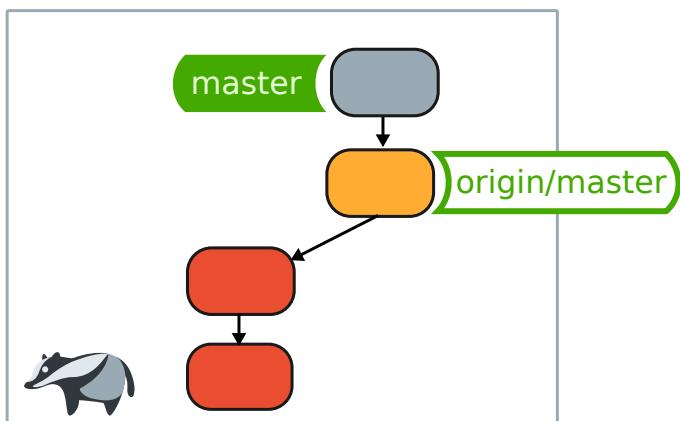
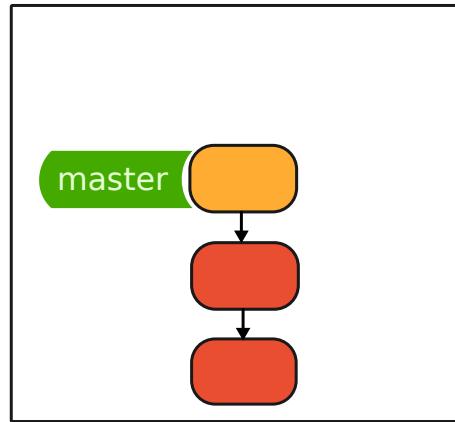
Rebase

Server repository

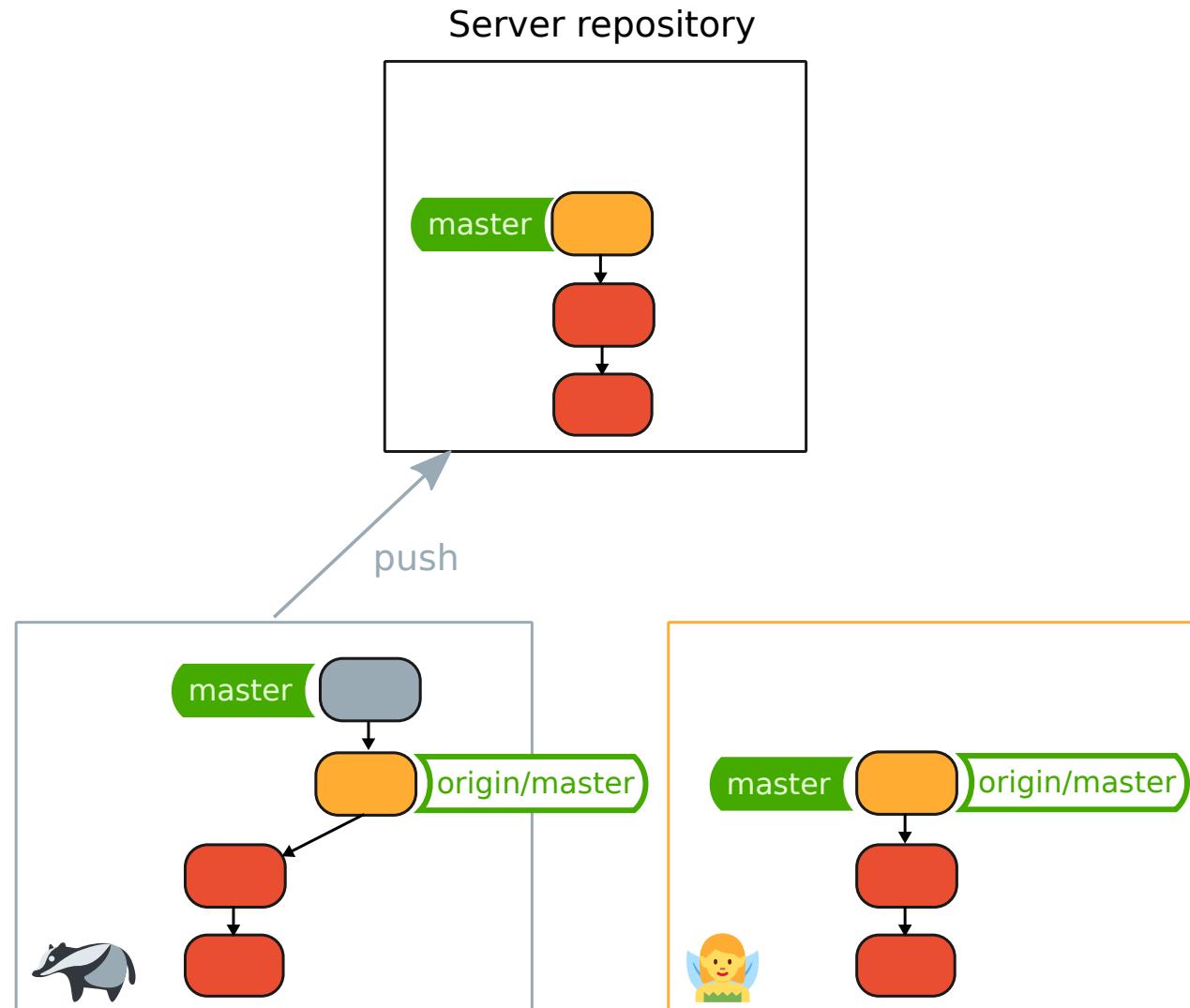


Rebase

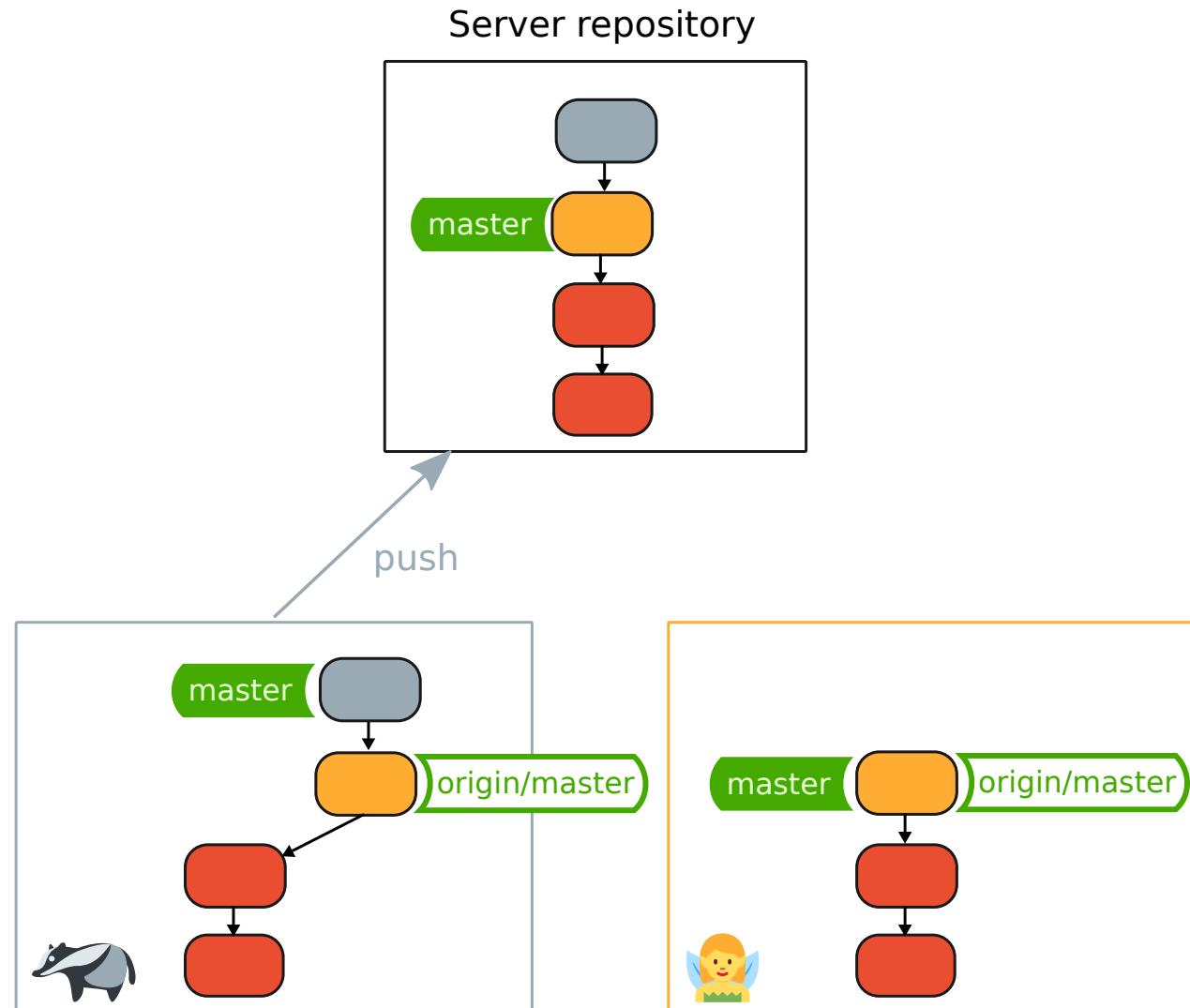
Server repository



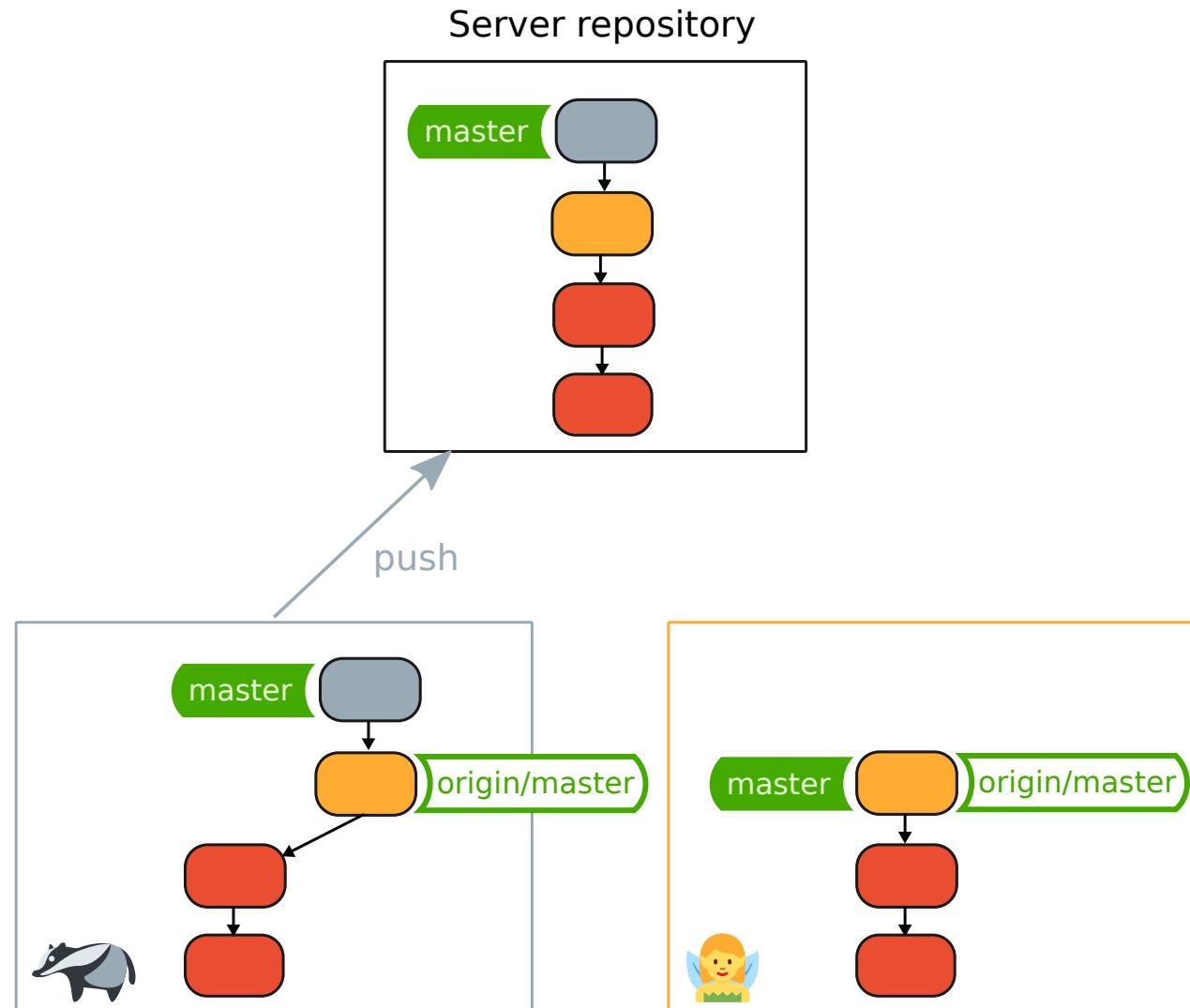
Rebase



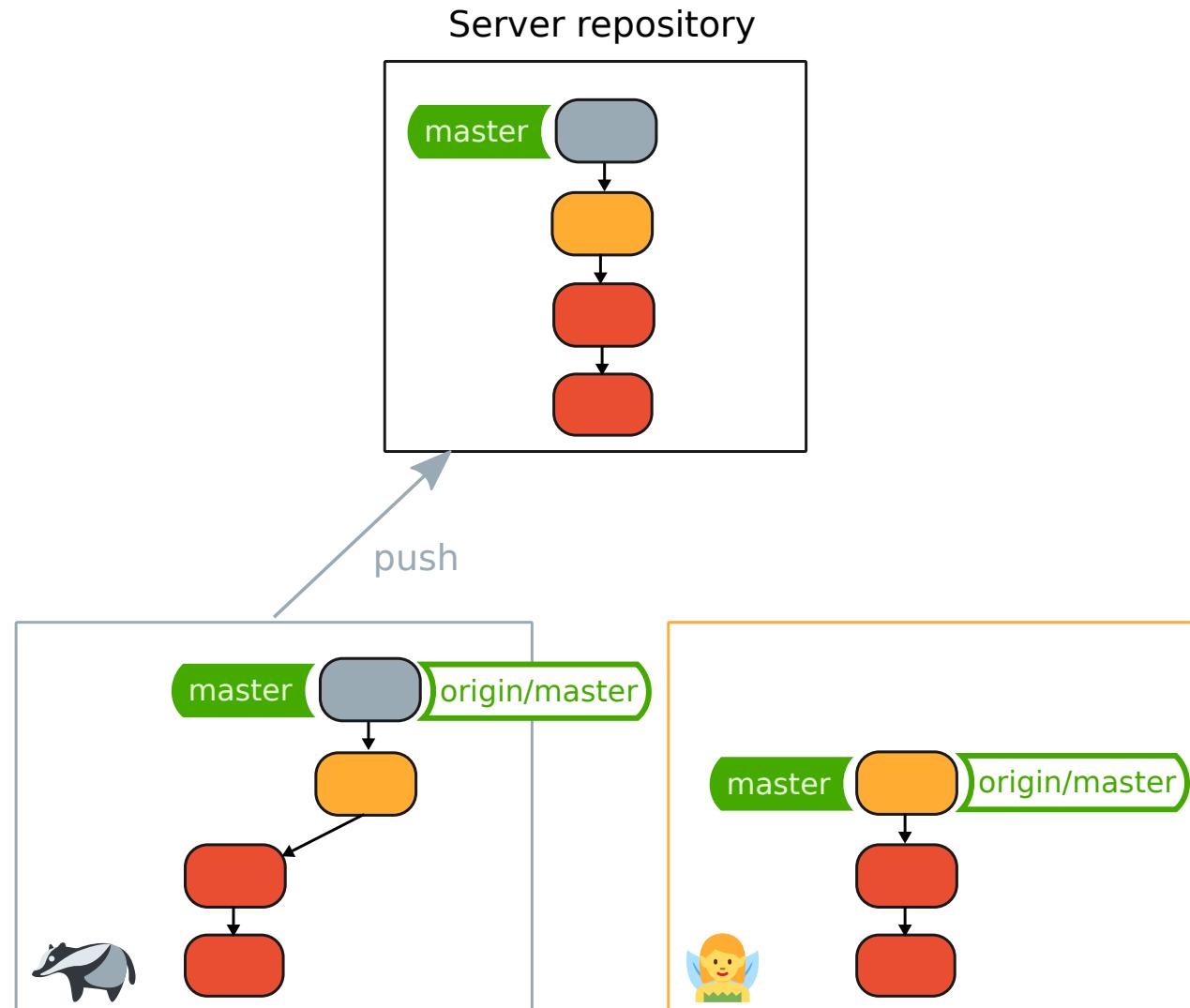
Rebase



Rebase

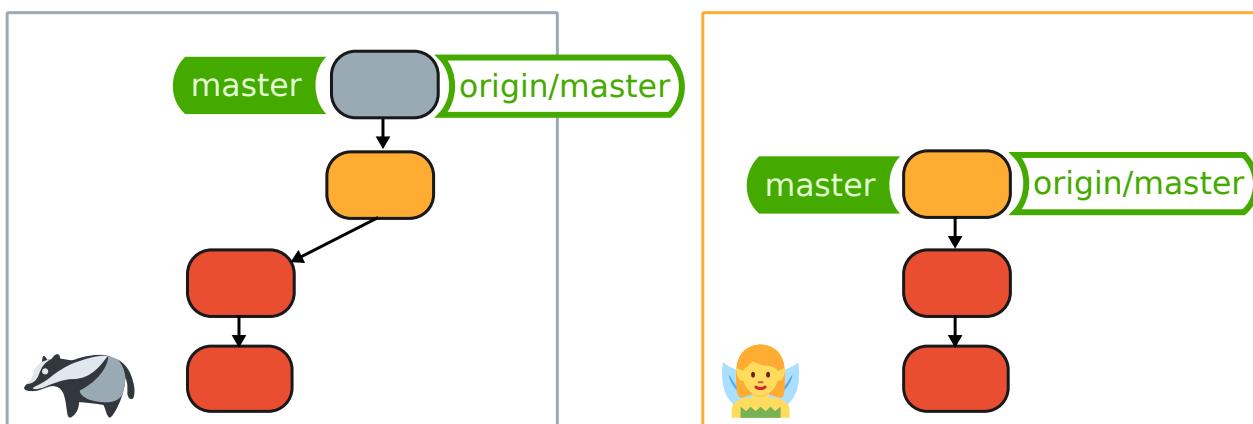
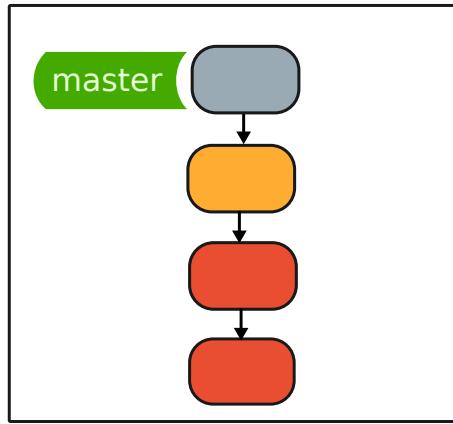


Rebase



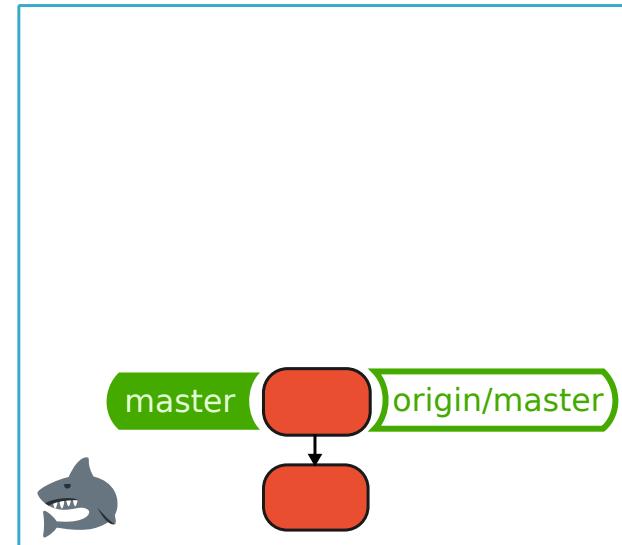
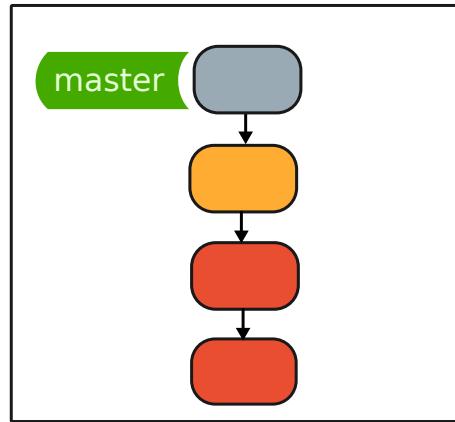
Rebase

Server repository

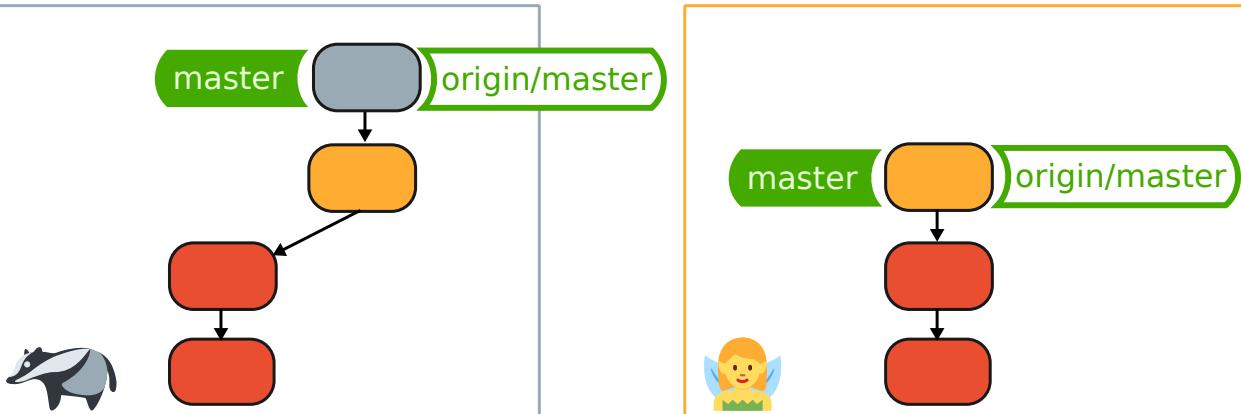
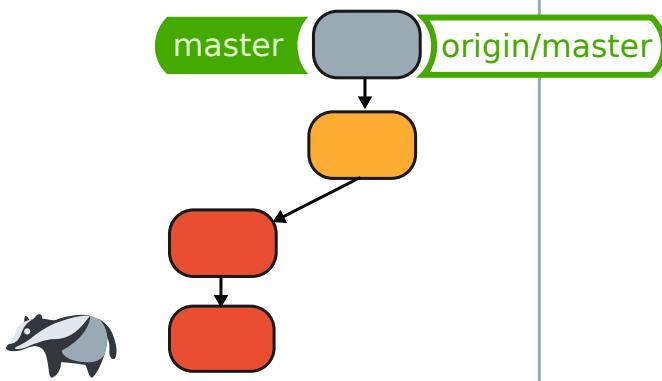


Rebase

Server repository

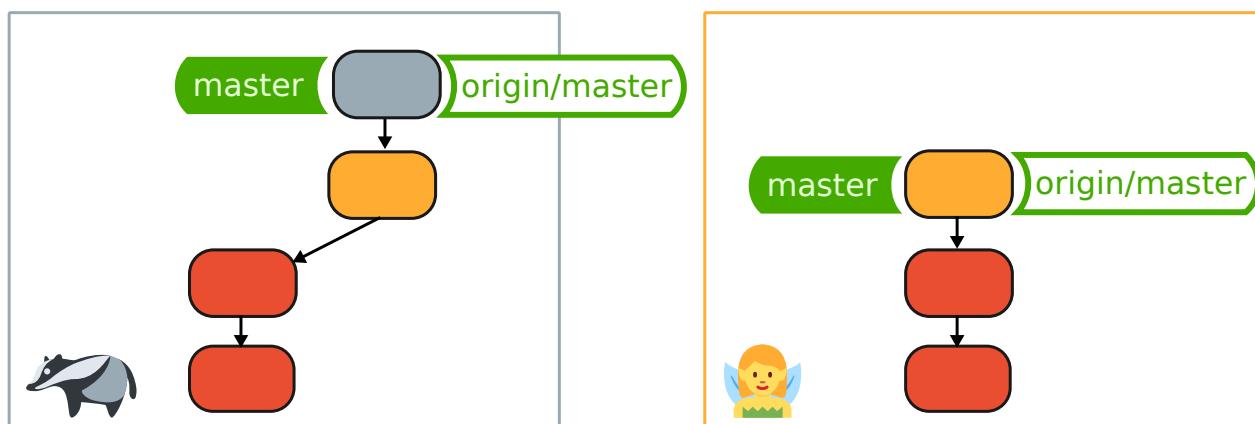
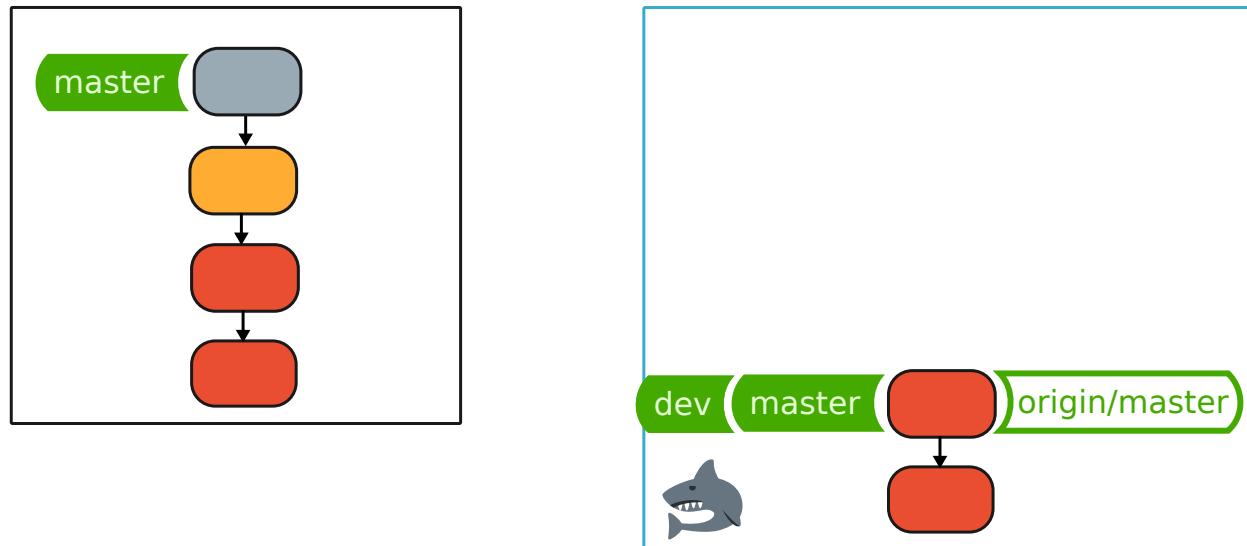


master origin/master



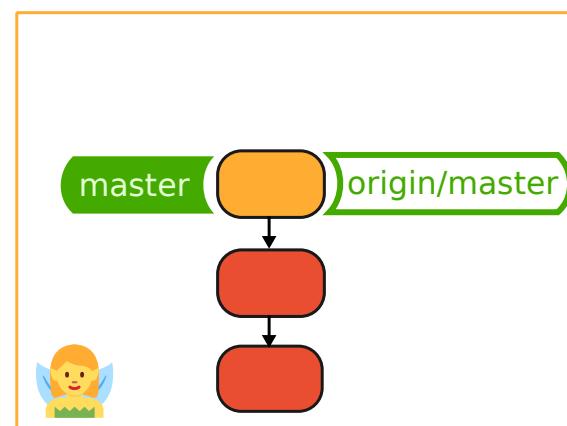
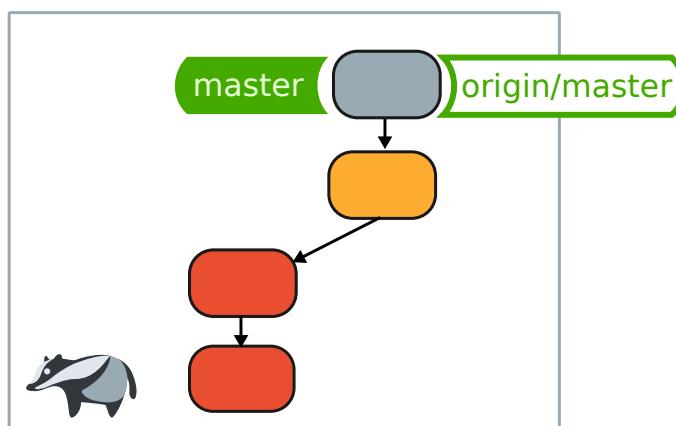
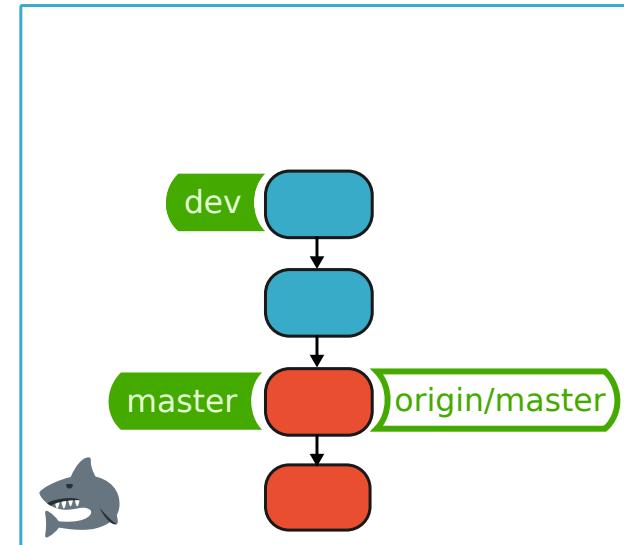
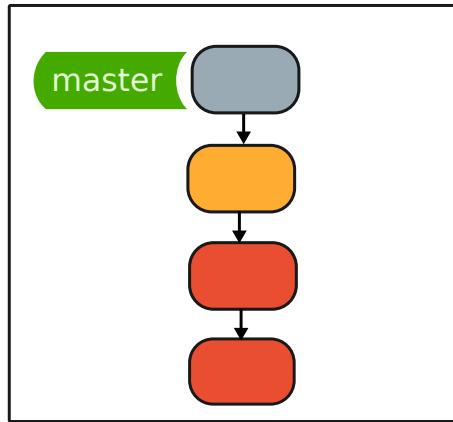
Rebase

Server repository



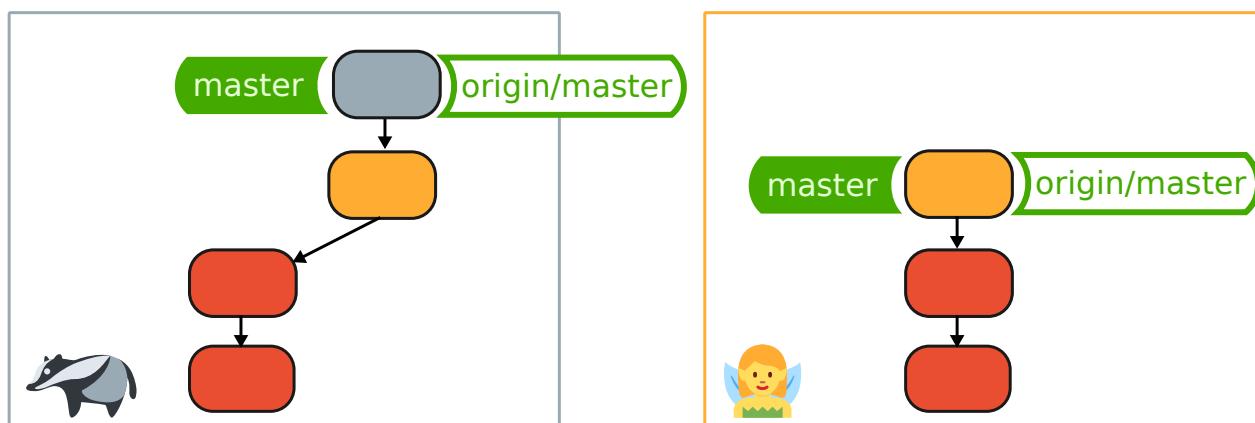
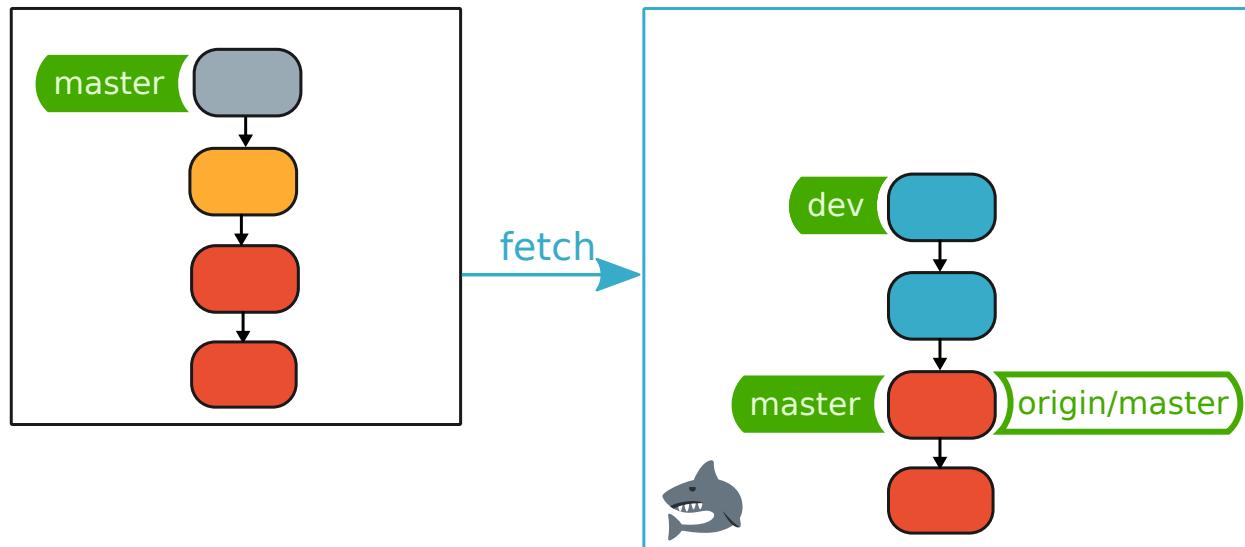
Rebase

Server repository



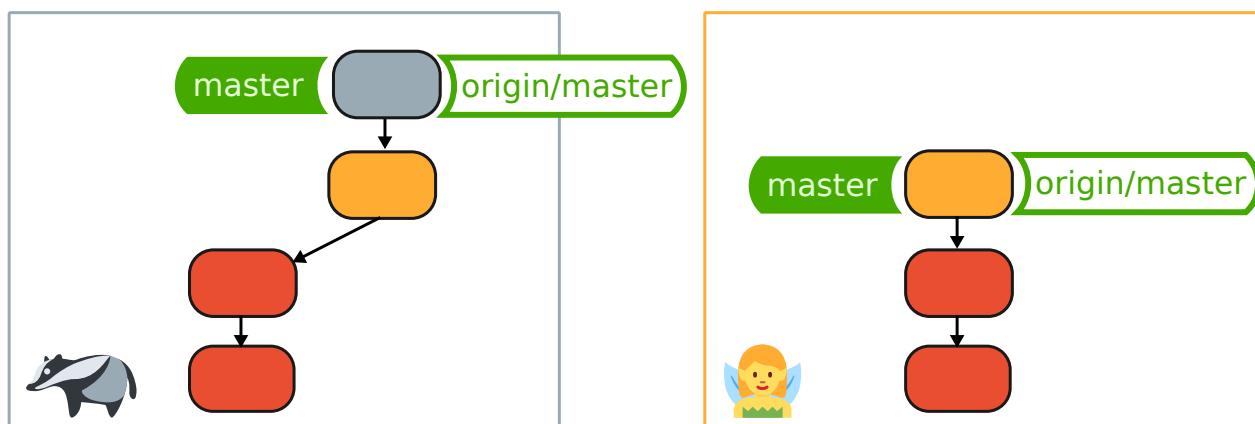
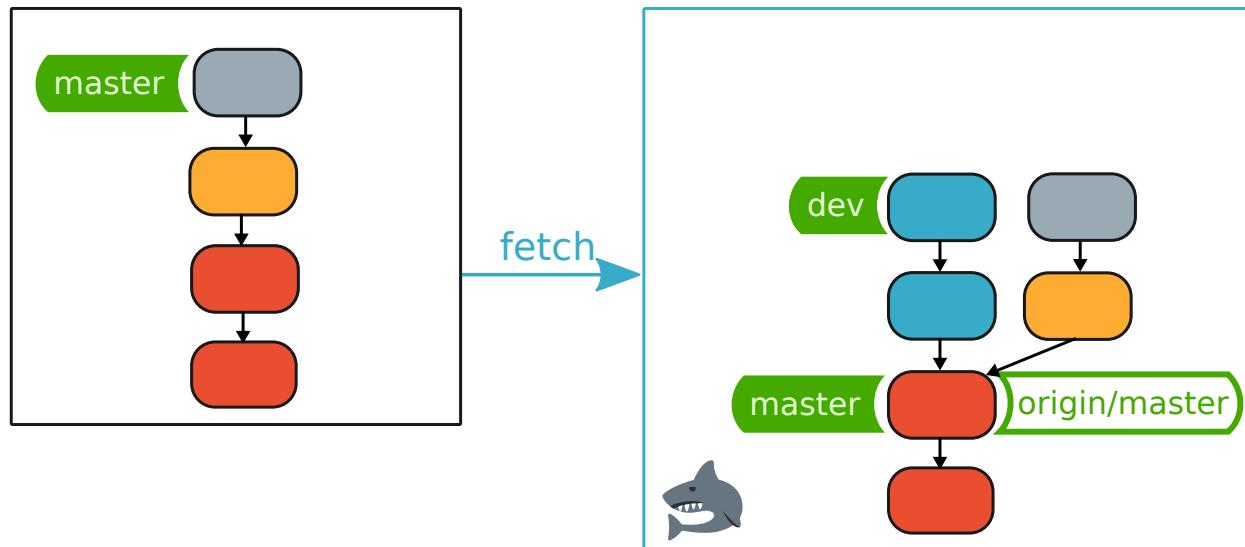
Rebase

Server repository



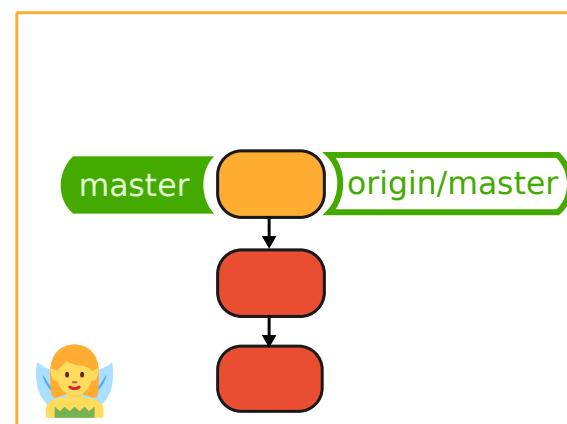
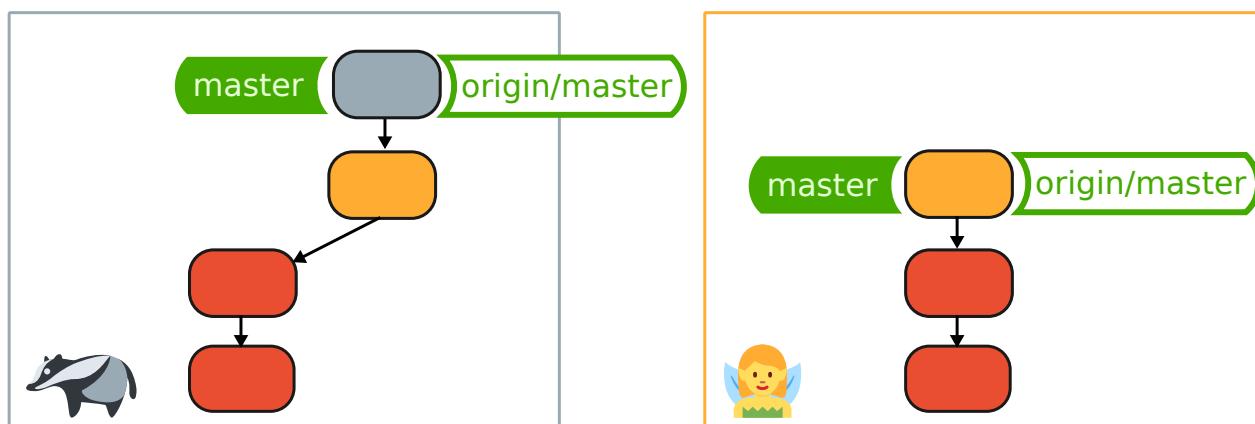
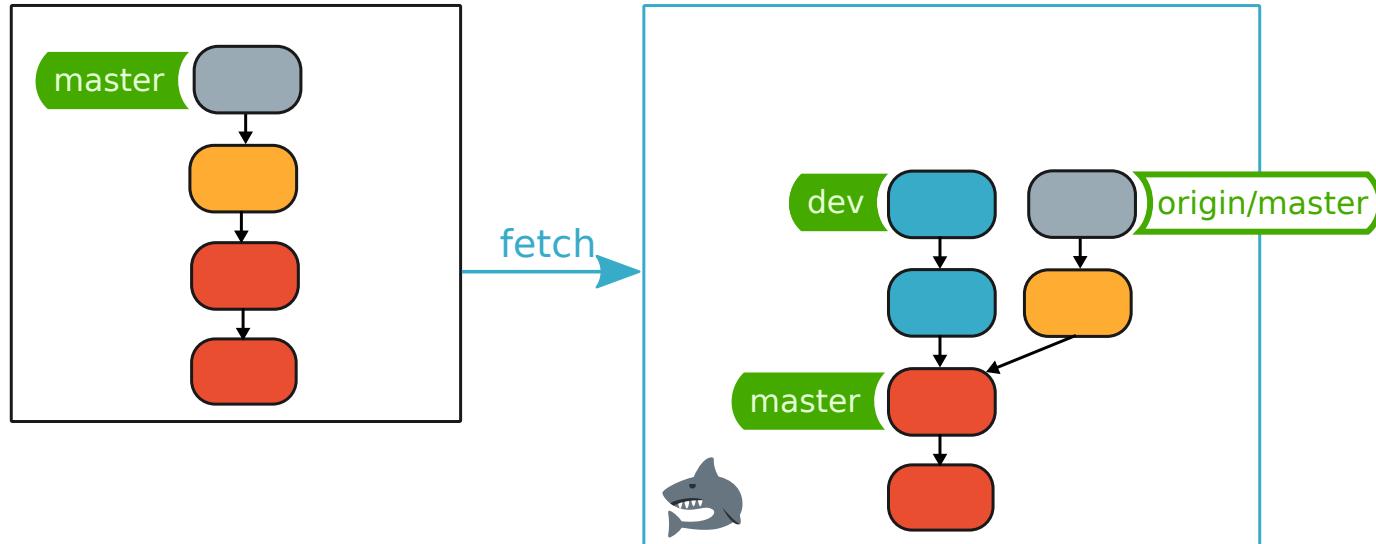
Rebase

Server repository



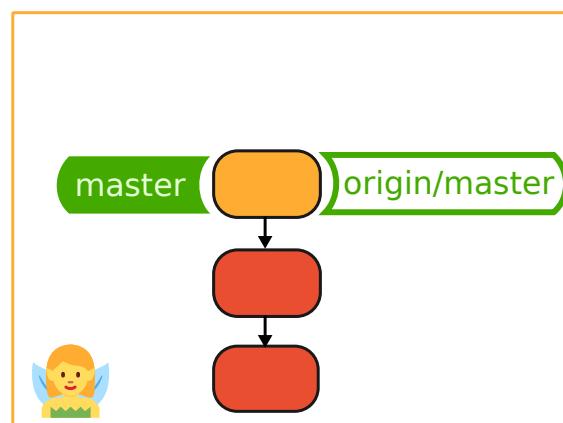
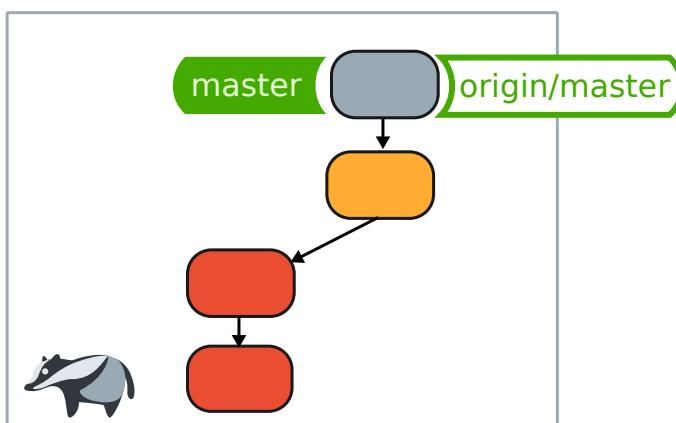
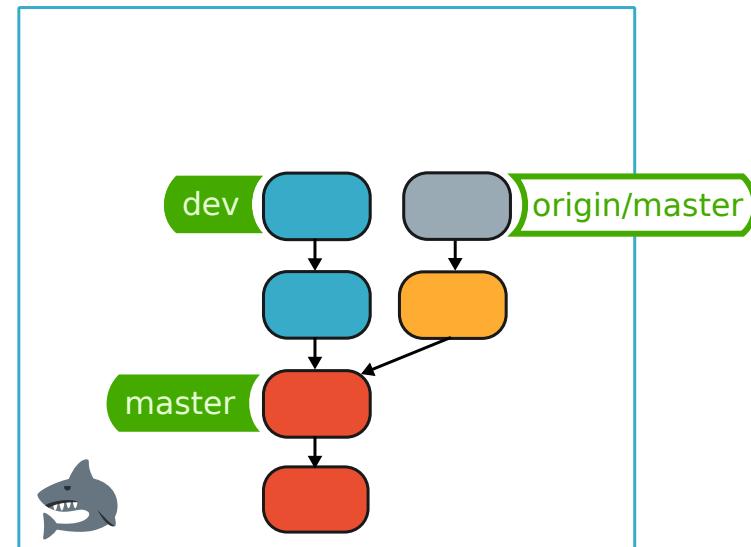
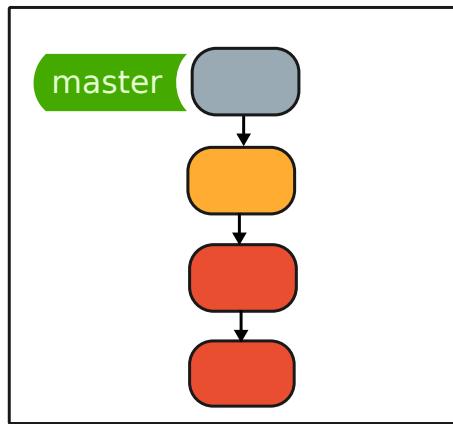
Rebase

Server repository



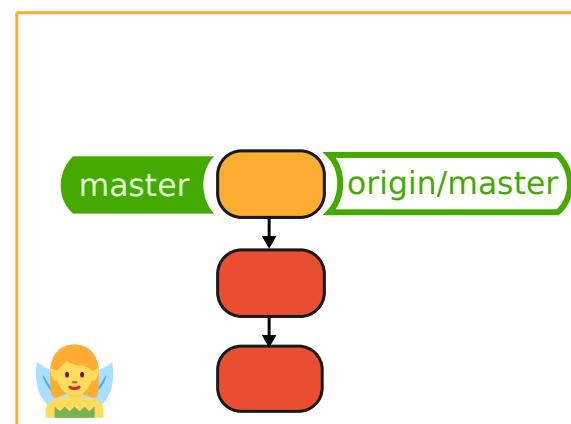
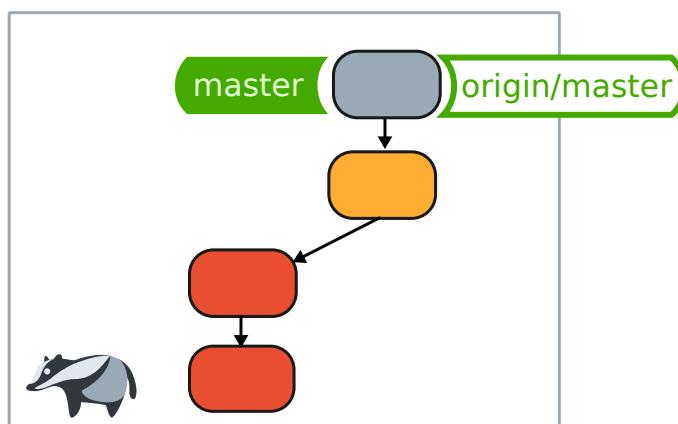
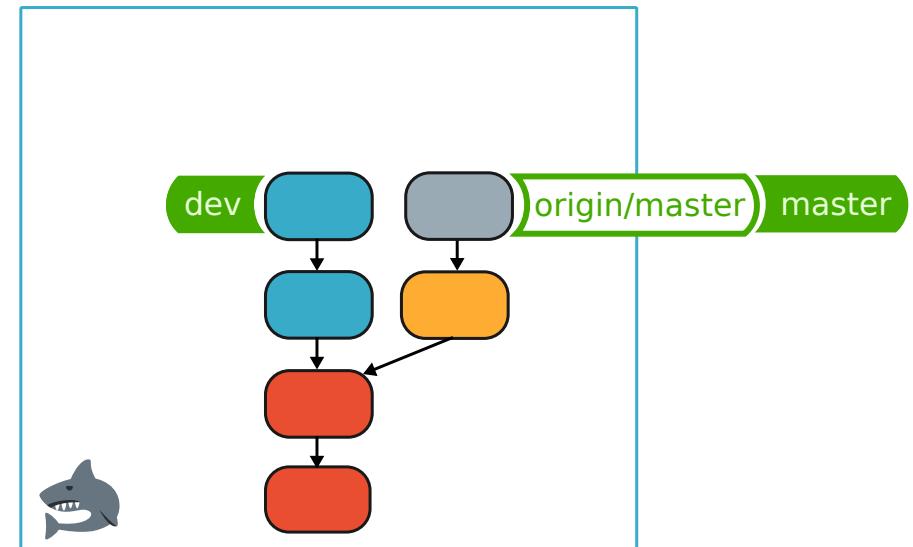
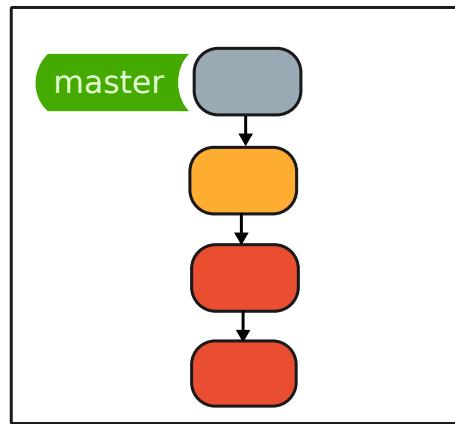
Rebase

Server repository



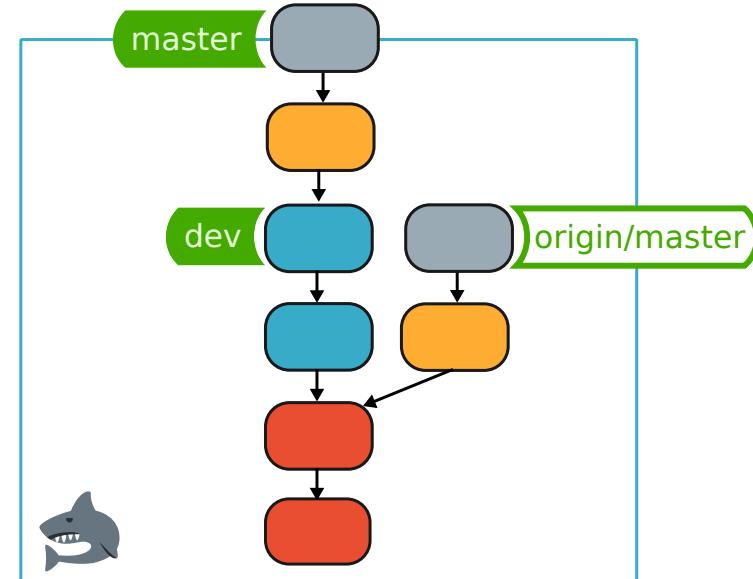
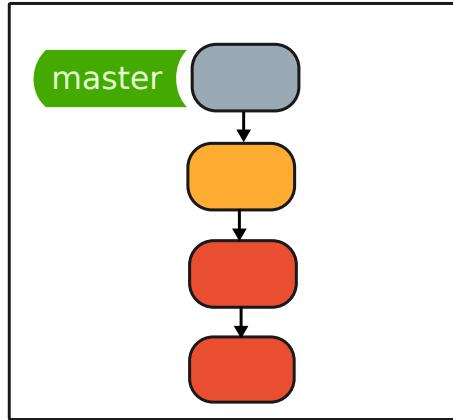
Rebase

Server repository



Rebase

Server repository



master origin/master

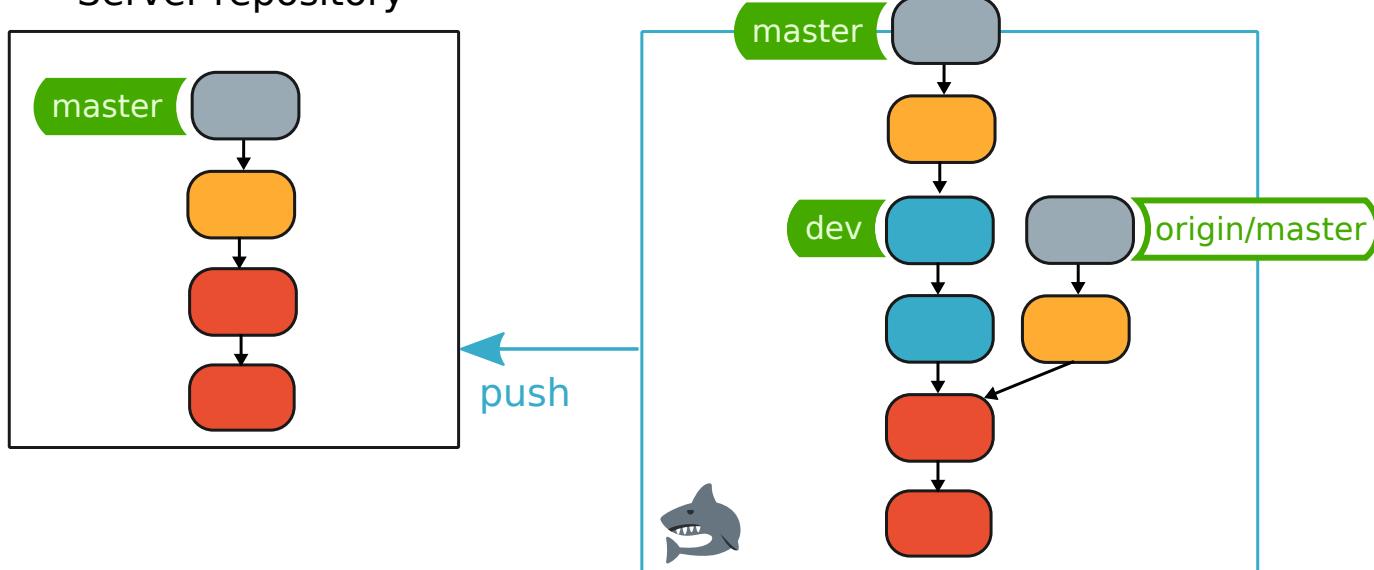


master origin/master

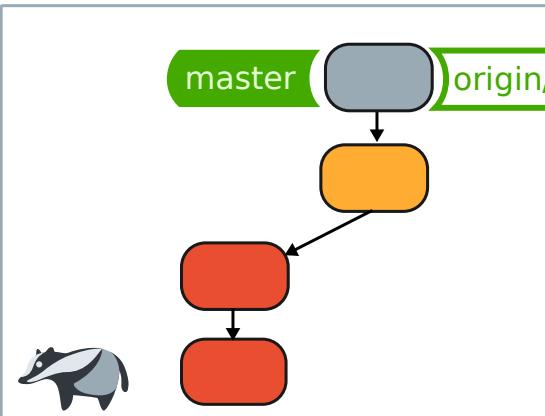


Rebase

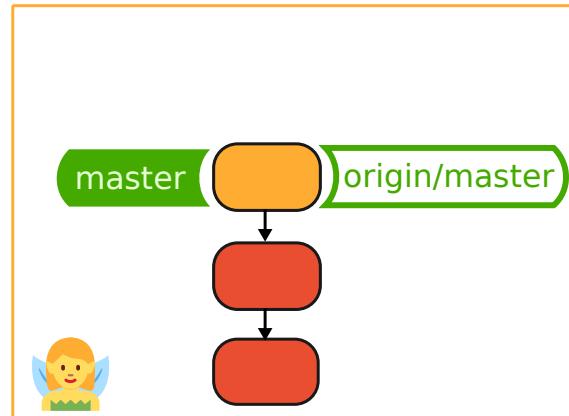
Server repository



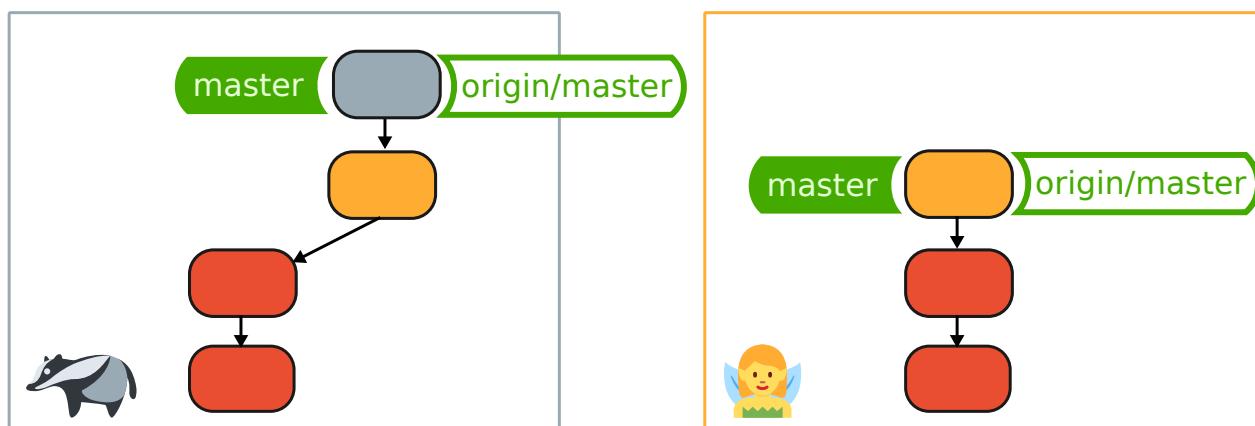
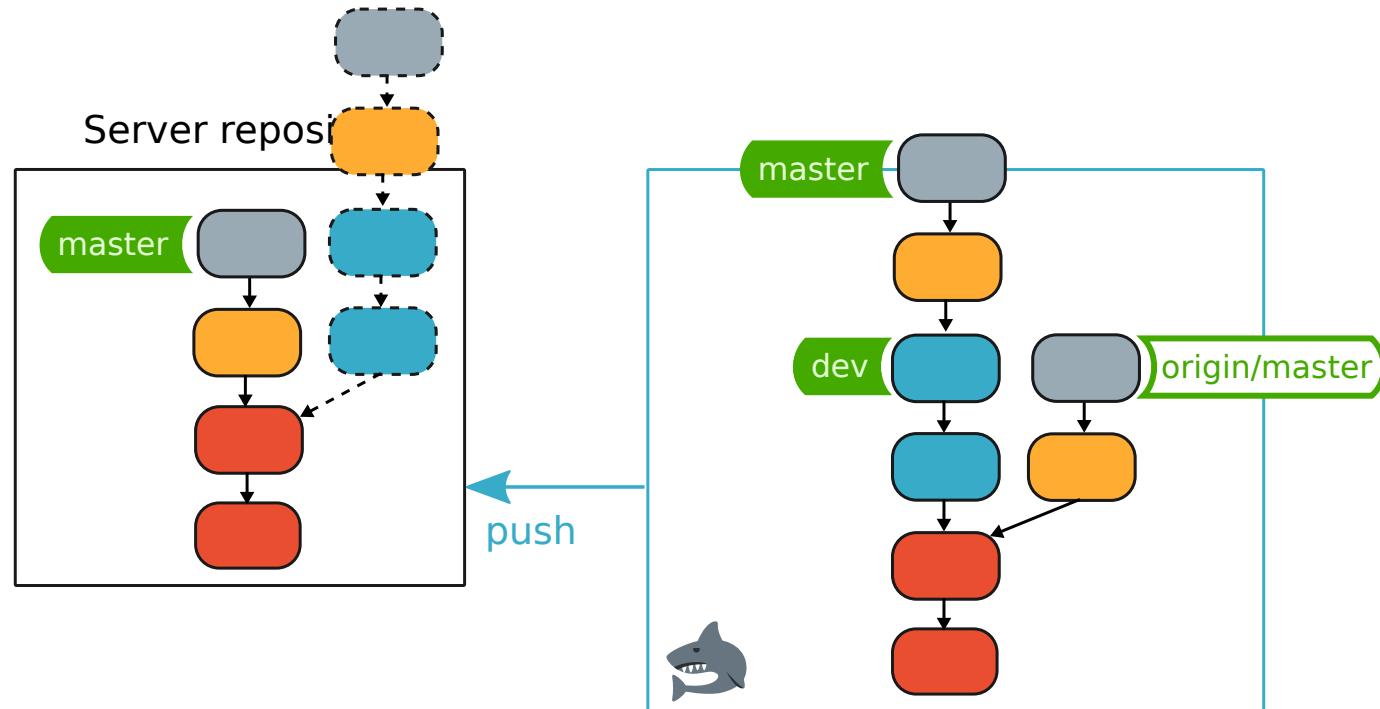
master origin/master



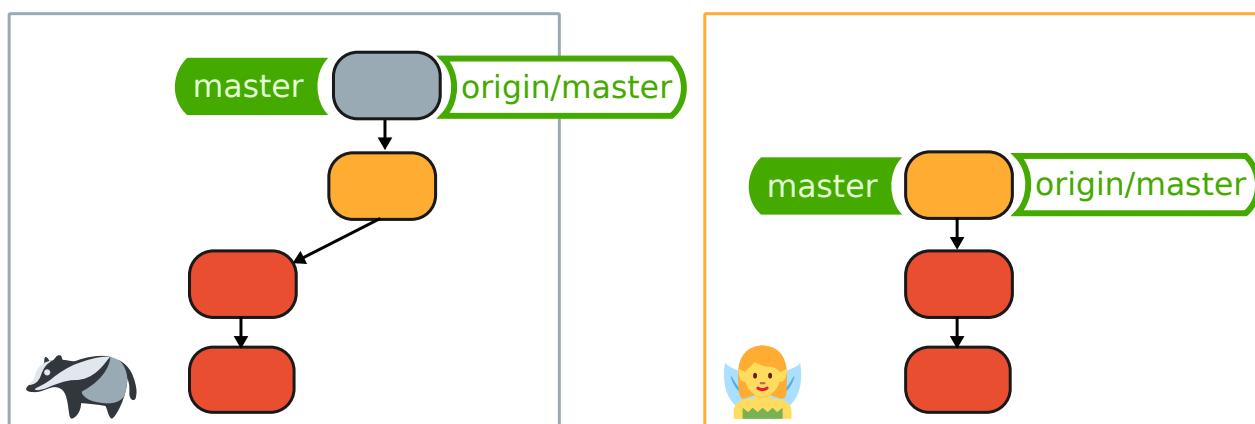
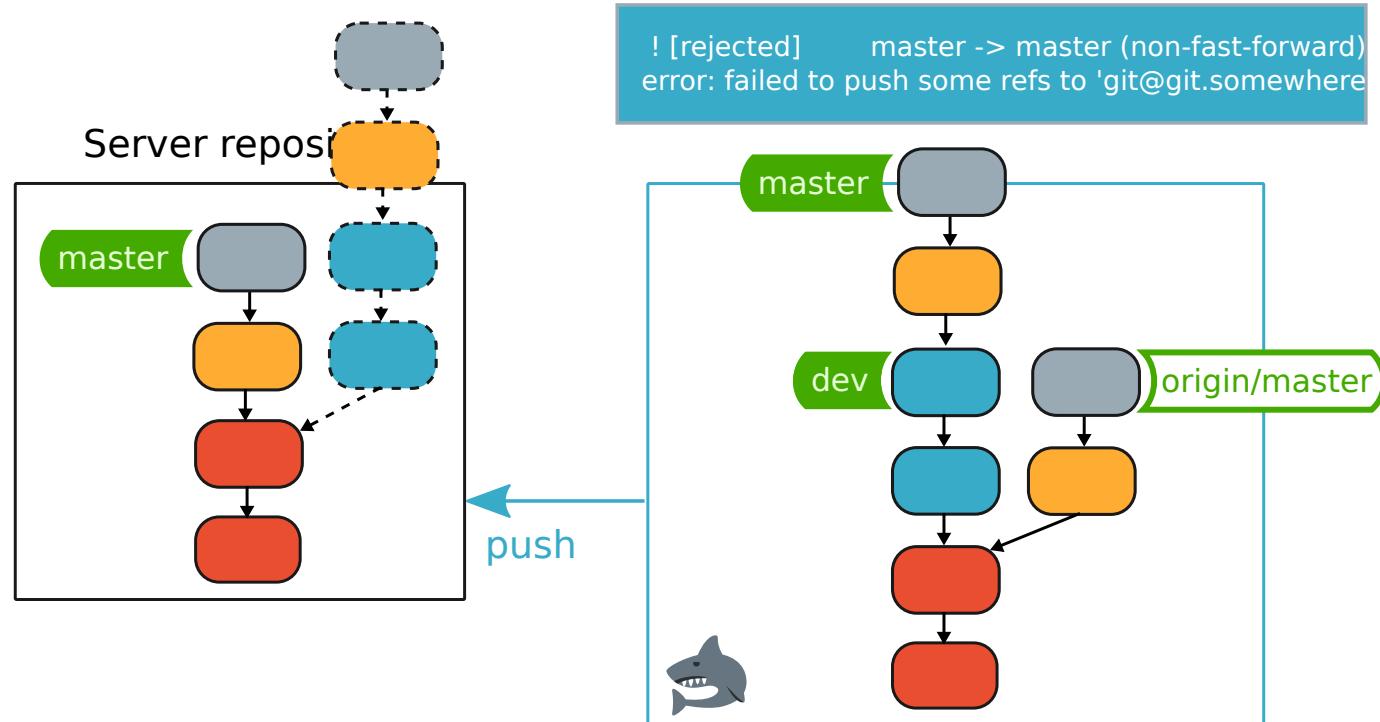
master origin/master



Rebase

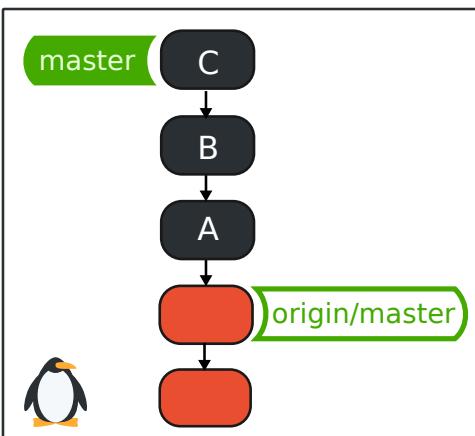
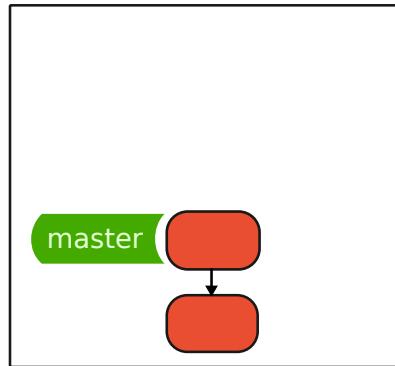


Rebase



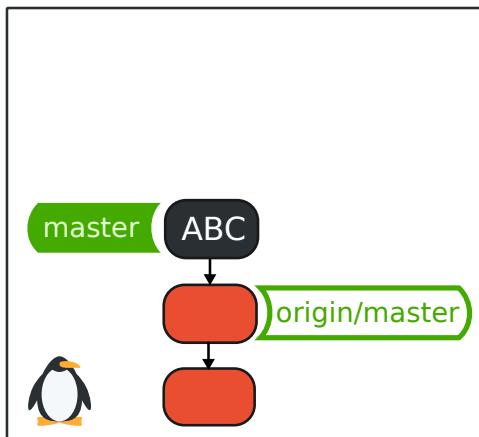
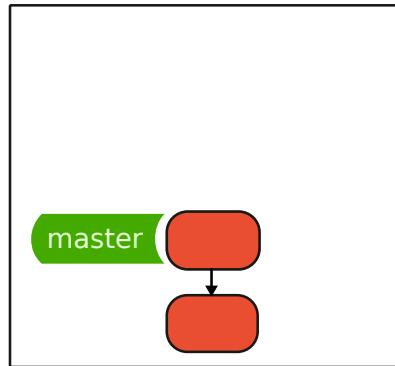
Squash

Server repository

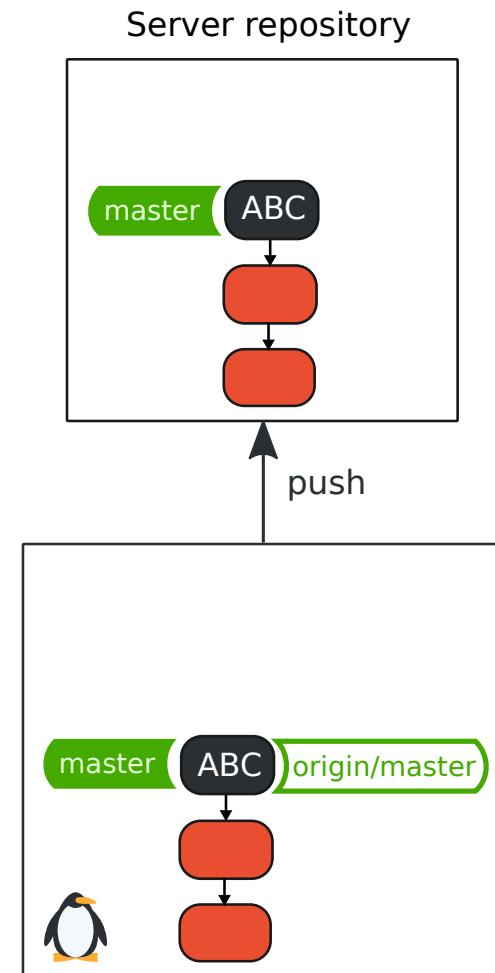


Squash

Server repository

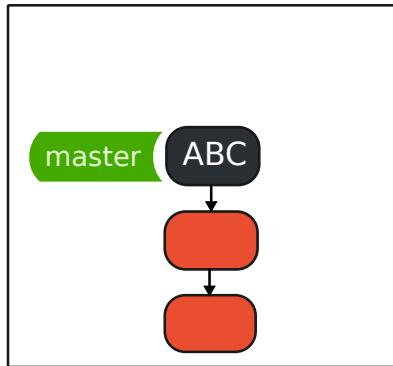


Squash

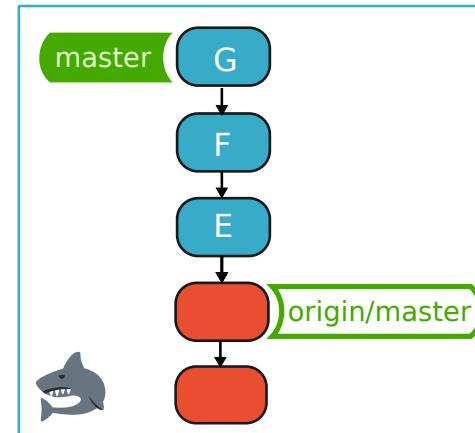
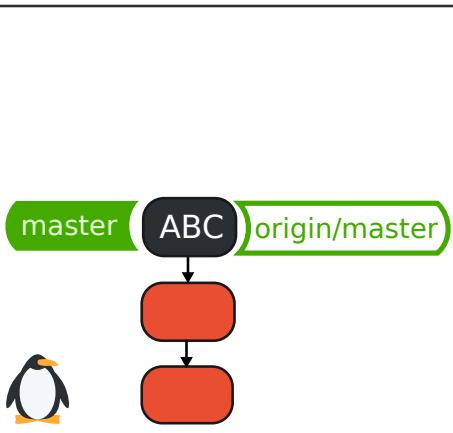
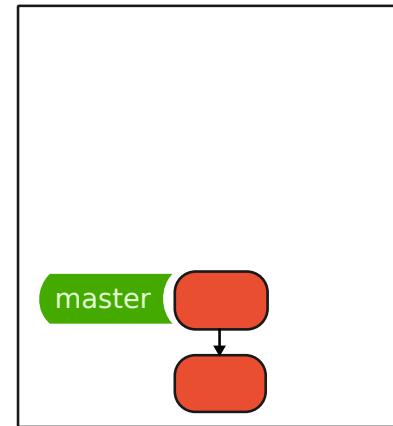


Squash

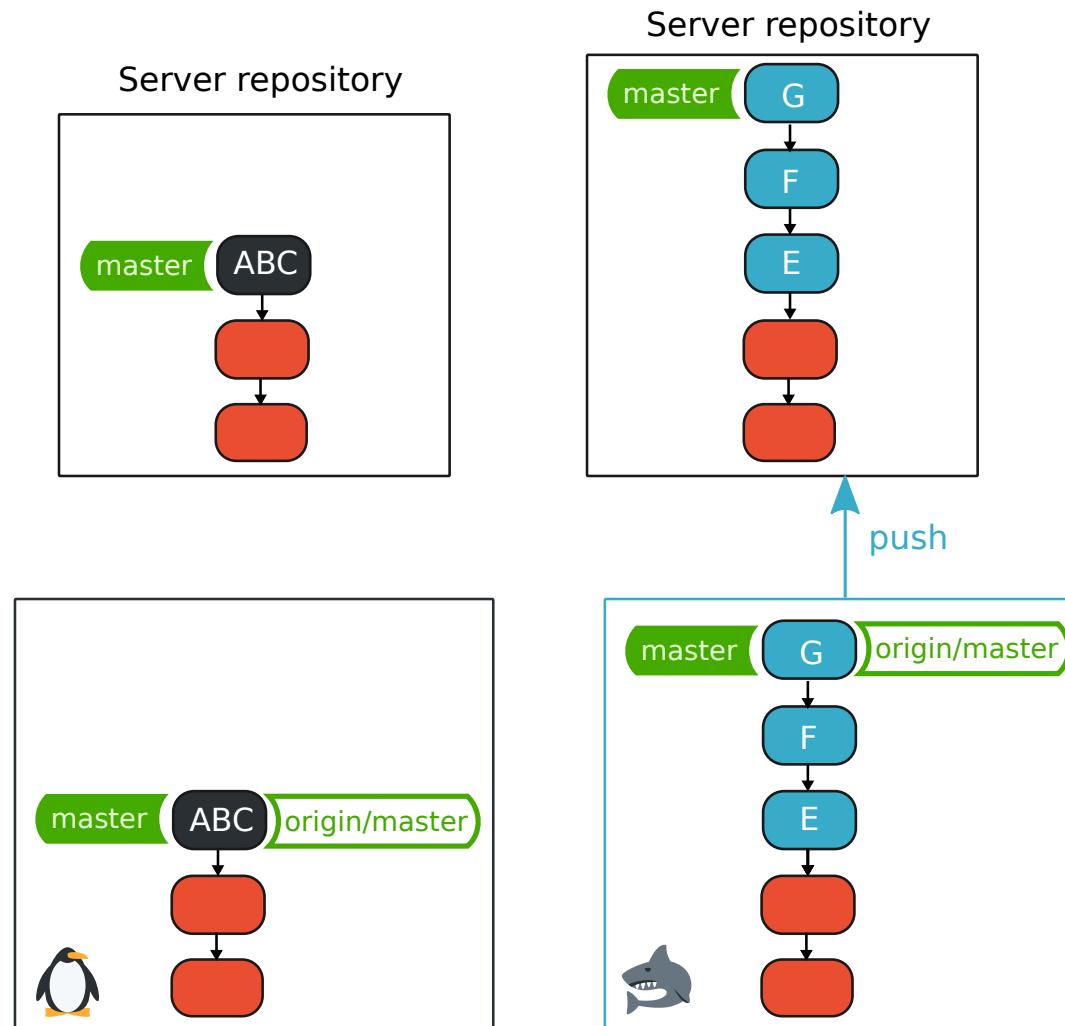
Server repository



Server repository

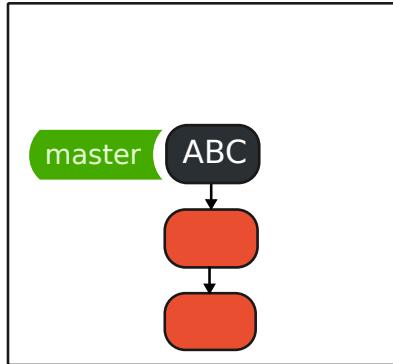


Squash

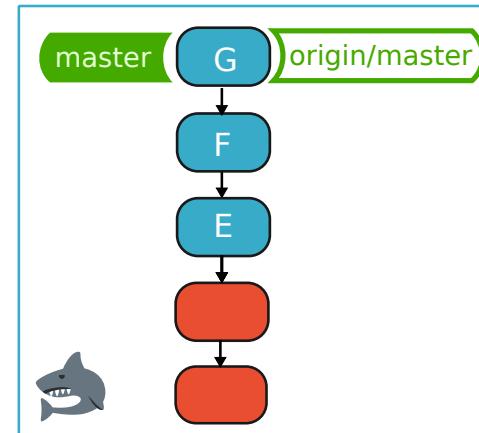
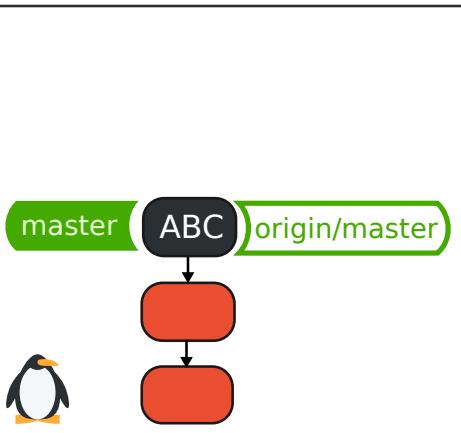
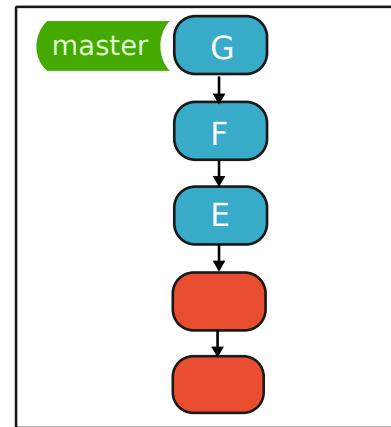


Squash

Server repository

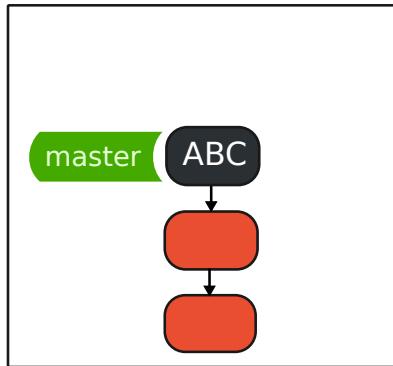


Server repository

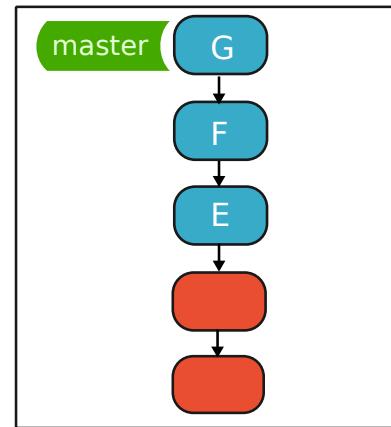


Squash

Server repository



Server repository



master (ABC) origin/master



G origin/master



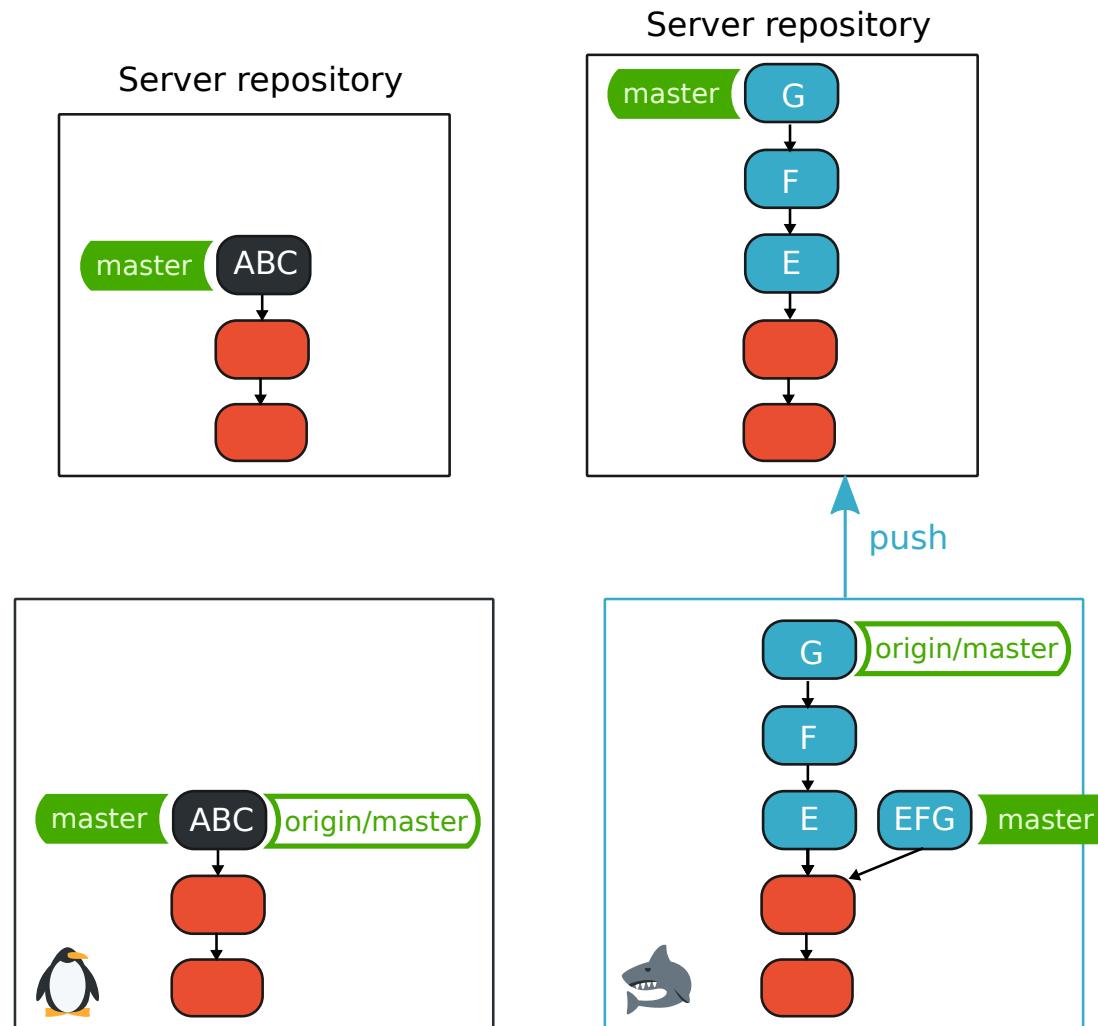
master

G origin/master

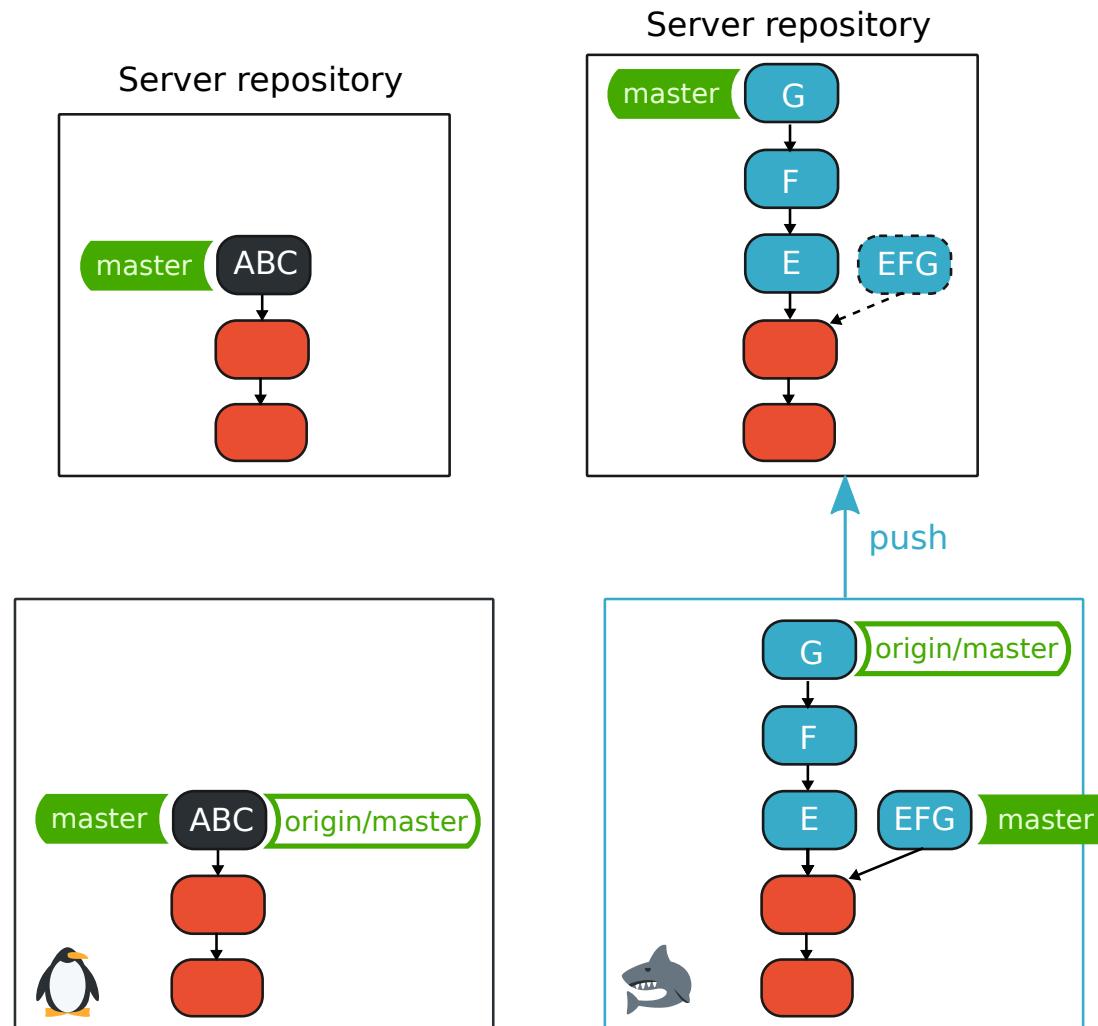


master

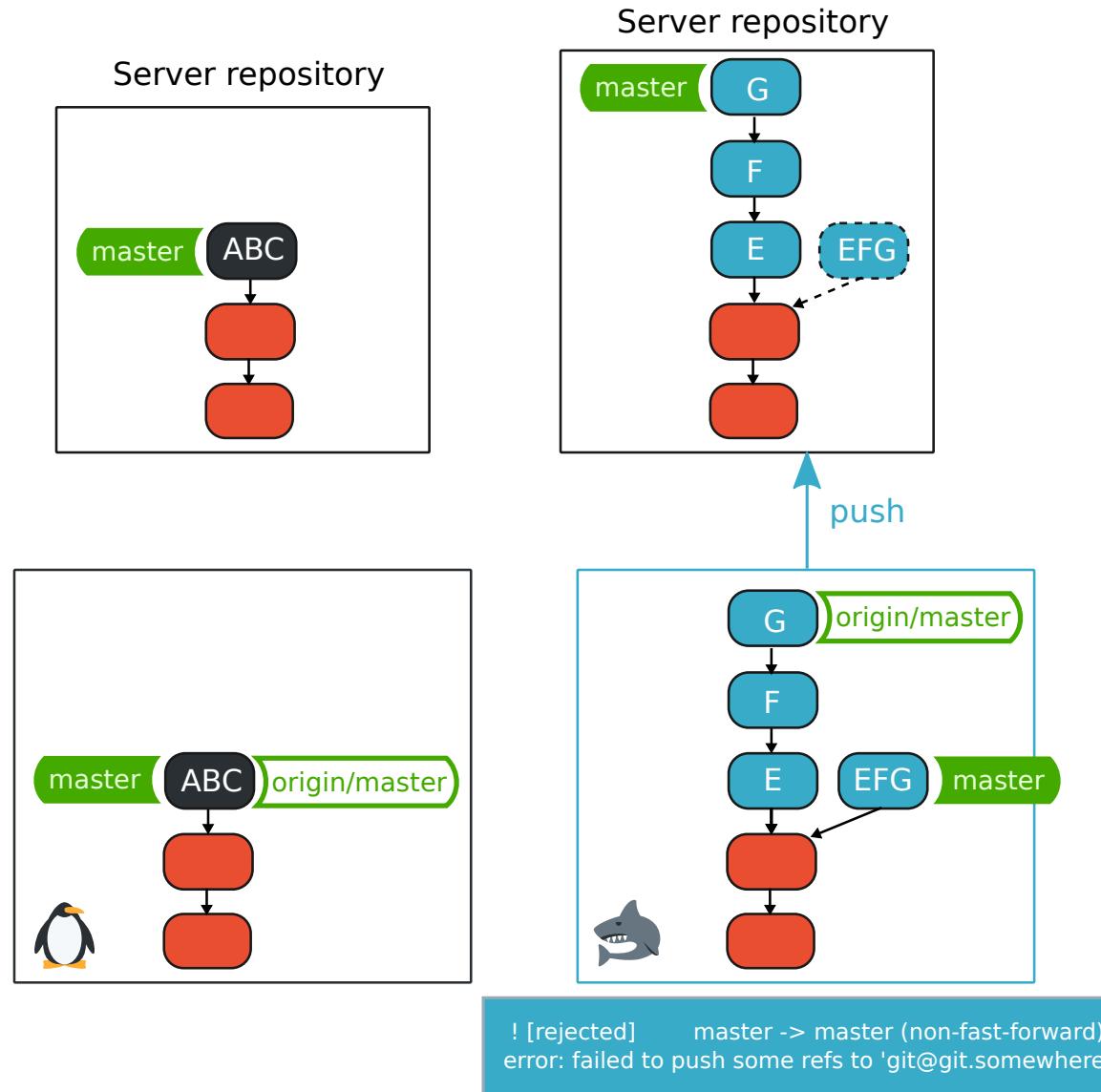
Squash



Squash

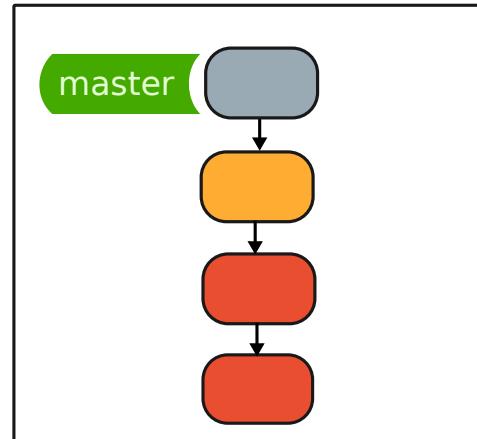


Squash

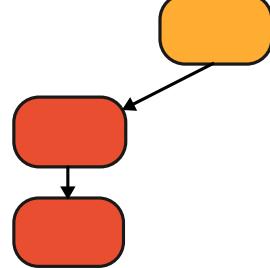


Another branch

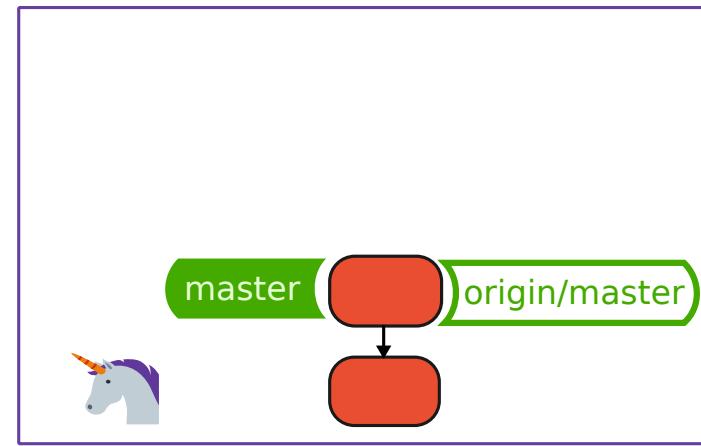
Server repository



master () origin/master

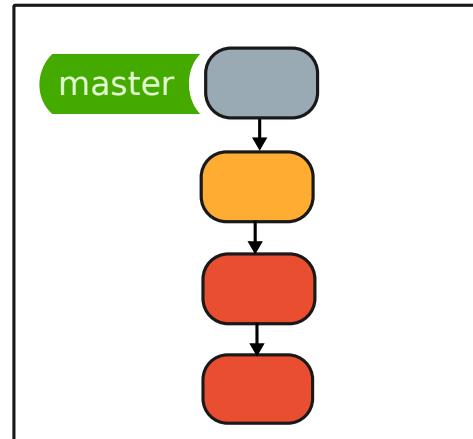


master () origin/master

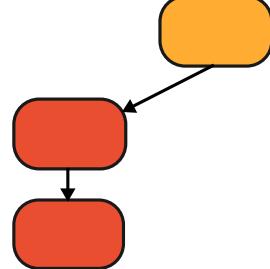


Another branch

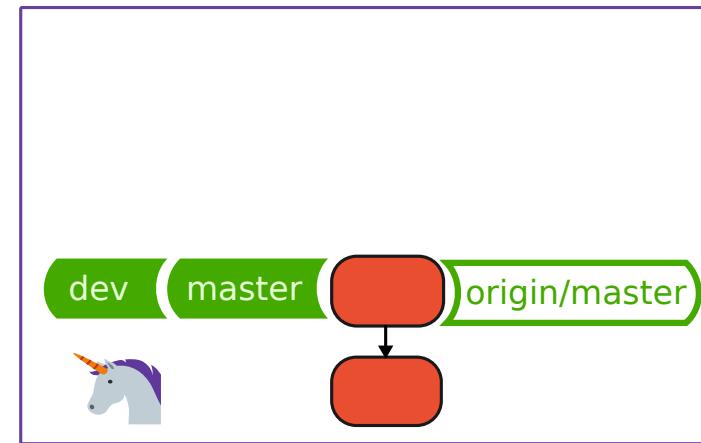
Server repository



master (grey oval) origin/master (green arrow)

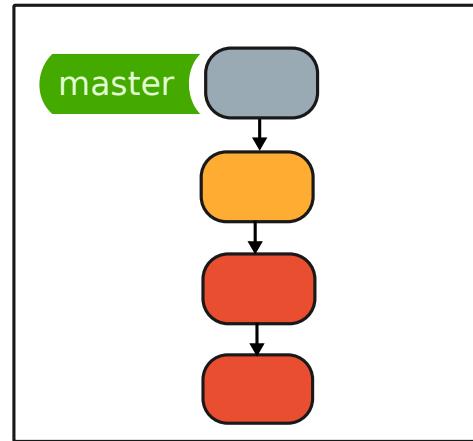


dev (master (red oval) origin/master (green arrow)

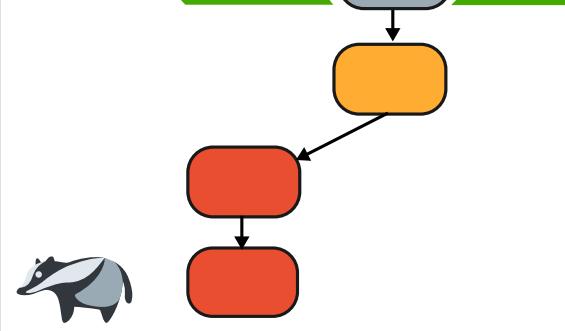


Another branch

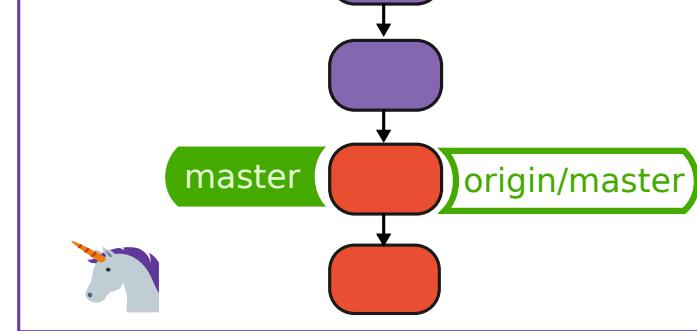
Server repository



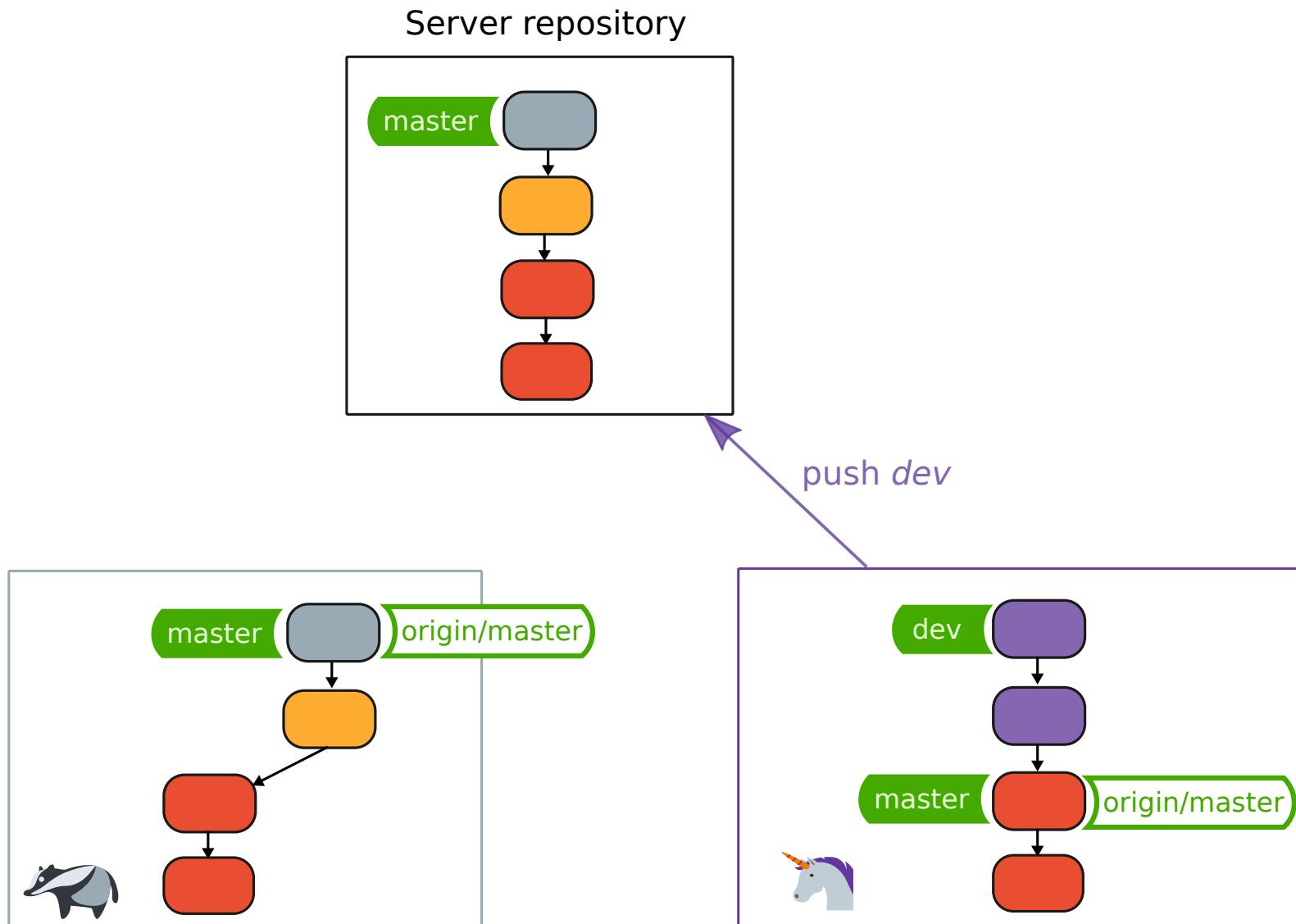
master (grey oval) origin/master (green oval)



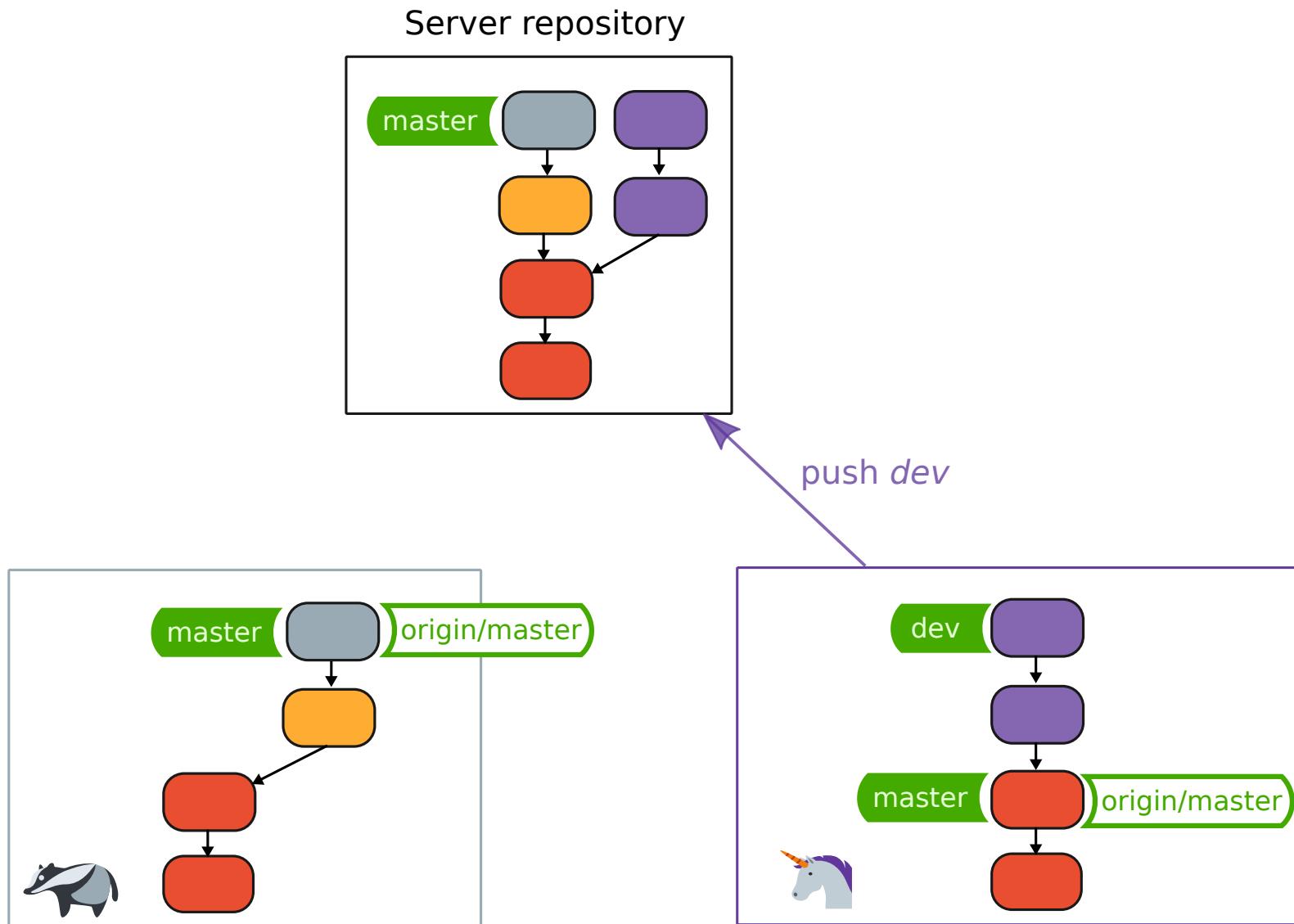
dev (purple oval)
master (orange oval) origin/master (green oval)



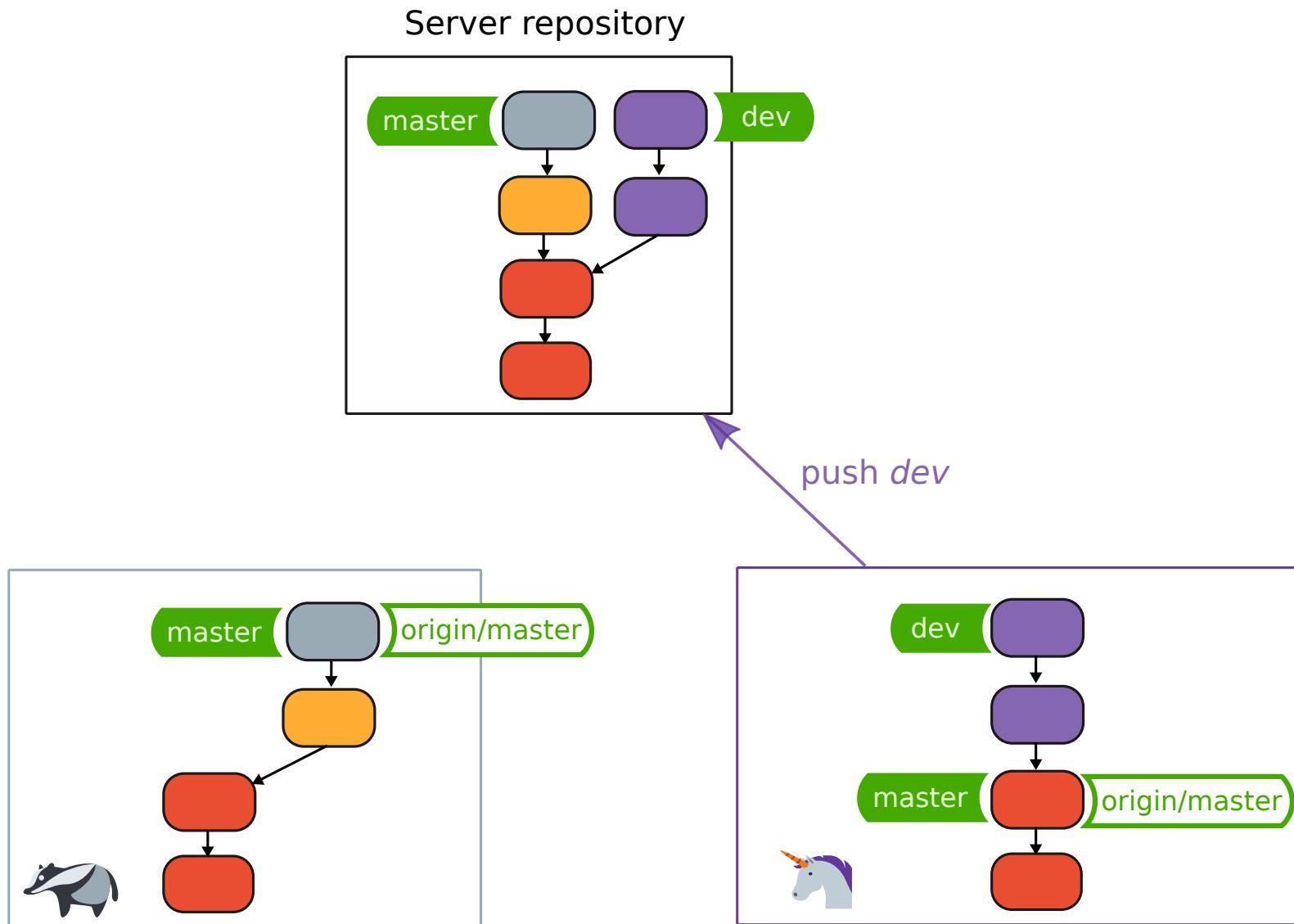
Another branch



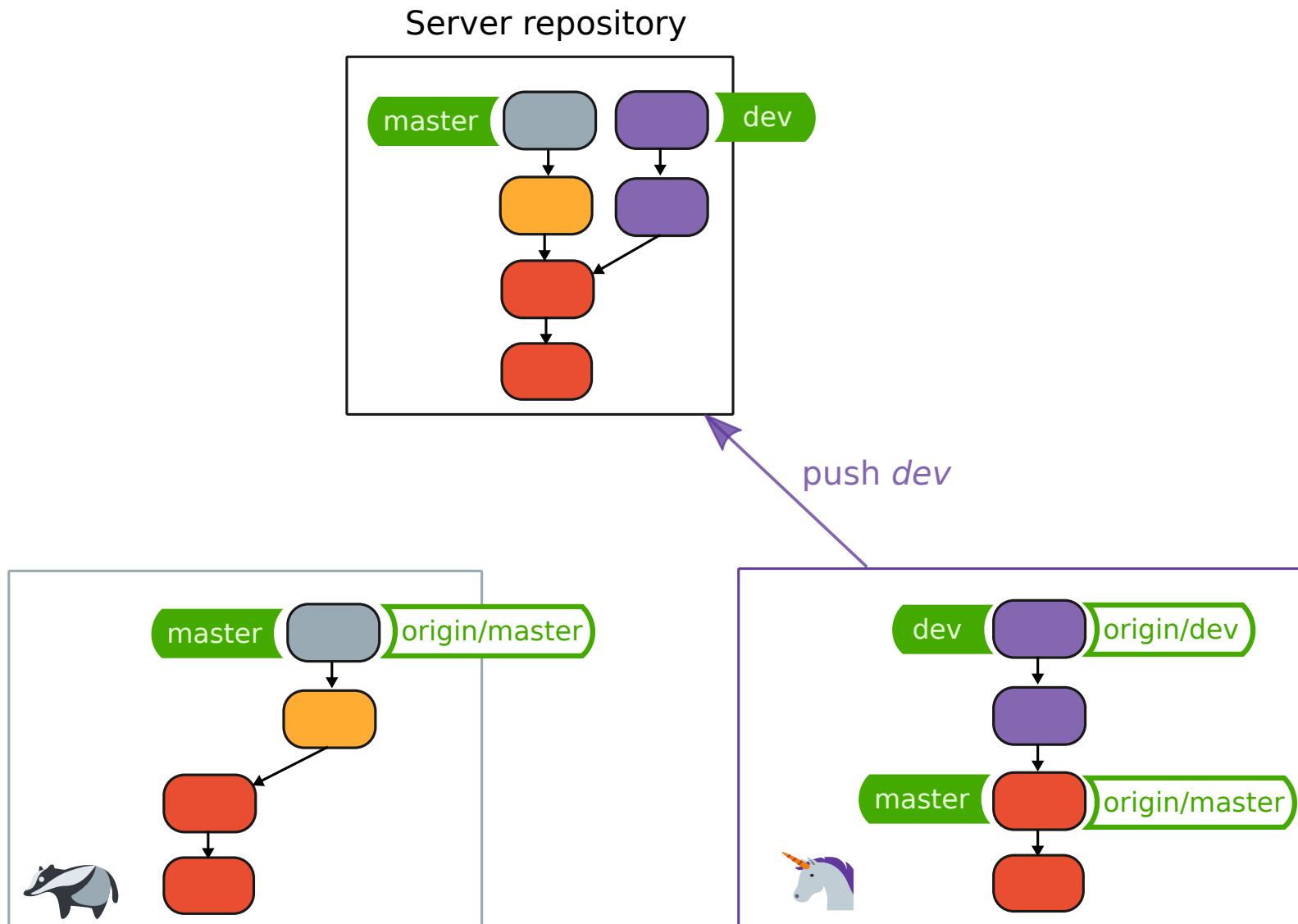
Another branch



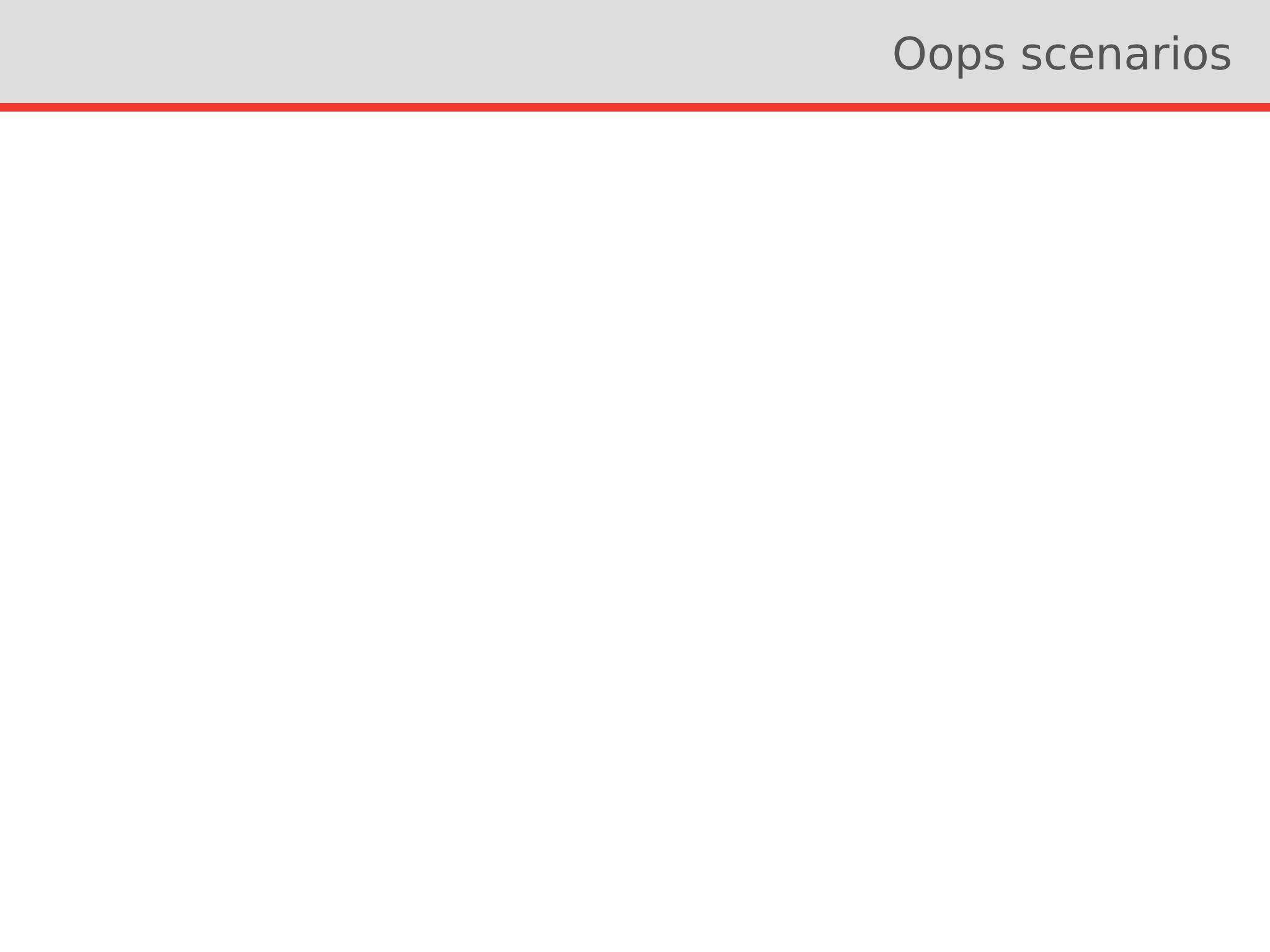
Another branch



Another branch



Oops scenarios



- Oops! (before commit)

```
$ git checkout -- file.txt
```

- Oops! (before commit)

```
$ git checkout -- file.txt
```

- Oops! (after commit & before push)

"uncommit"

```
$ git reset HEAD~1
```

"recommit"

```
$ git commit --amend
```

- Oops! (before commit)

```
$ git checkout -- file.txt
```

- Oops! (after commit & before push)

"uncommit"

```
$ git reset HEAD~1
```

"recommit"

```
$ git commit --amend
```

- Bigger Oops! (after commit & before push)

```
$ git reflog
```

- Oops! (before commit)

```
$ git checkout -- file.txt
```

- Oops! (after commit & before push)

"uncommit"

```
$ git reset HEAD~1
```

"recommit"

```
$ git commit --amend
```

- Bigger Oops! (after commit & before push)

```
$ git reflog
```

- Public Oops! (after push)

- Oops! (before commit)

```
$ git checkout -- file.txt
```

- Oops! (after commit & before push)
"uncommit"

```
$ git reset HEAD~1
```

"recommit"

```
$ git commit --amend
```

- Bigger Oops! (after commit & before push)

```
$ git reflog
```

- Public Oops! (after push)
- Mega Oops!

Other Scenarios



- Non-versioned files

- Non-versioned files
- Release

- Non-versioned files
- Release
- Partial commit

- Non-versioned files
- Release
- Partial commit
- Bisect

- Non-versioned files
- Release
- Partial commit
- Bisect
- New repository

1. Introduction

2. Hands-on:

- Basic operations
- Branching / Merge / Rebase
- Remote repository

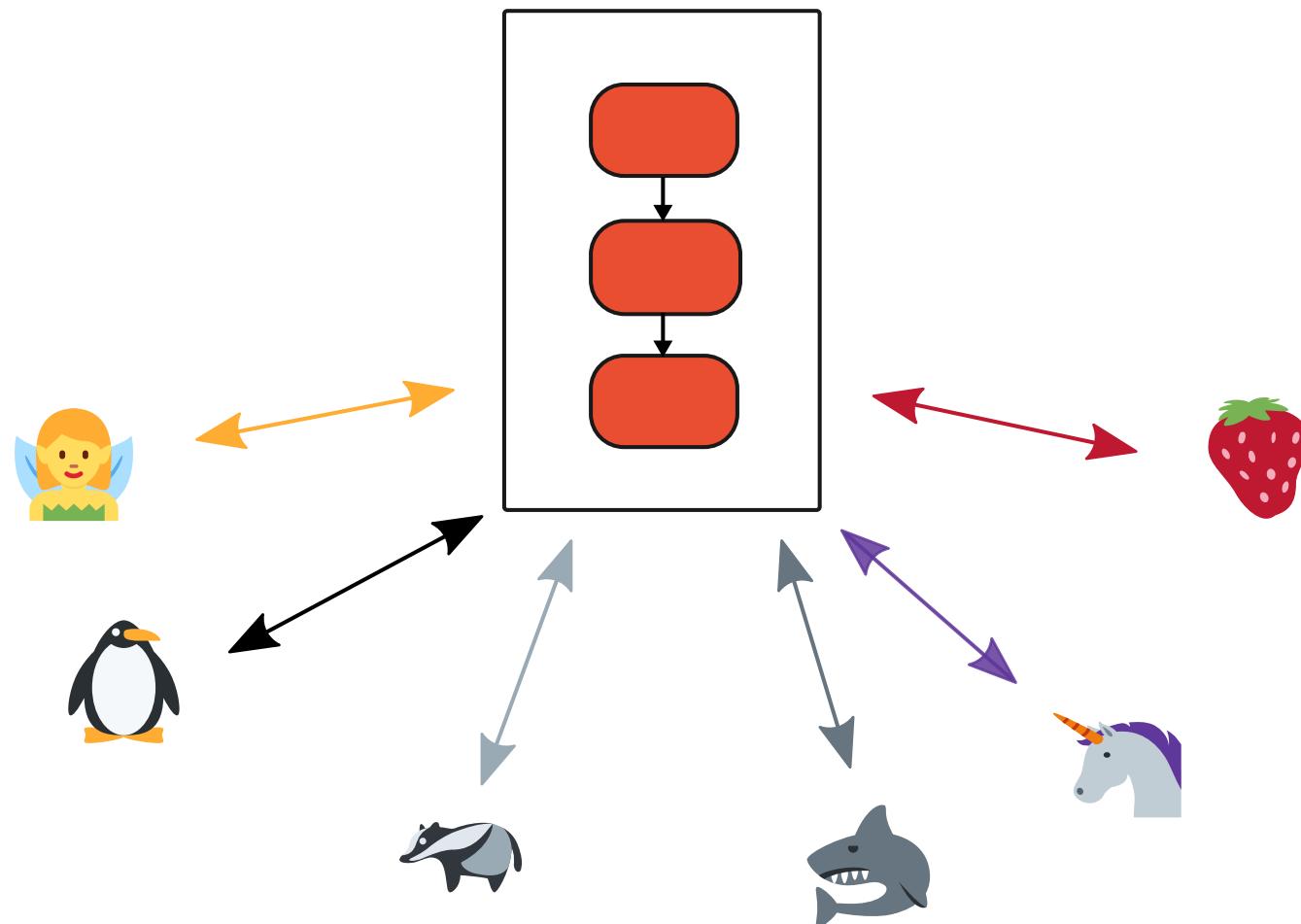
3. Common scenarios

4. Repository management

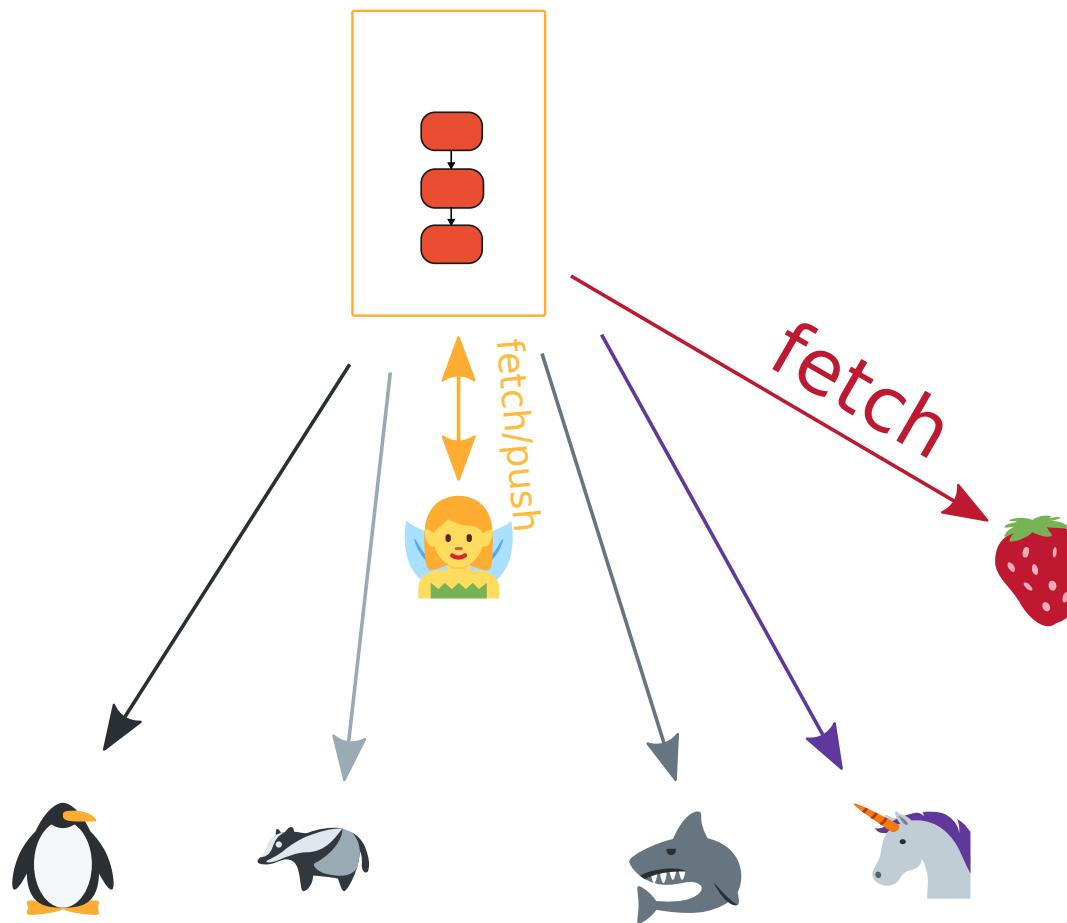
5. *Bonus topics*

Repository Access

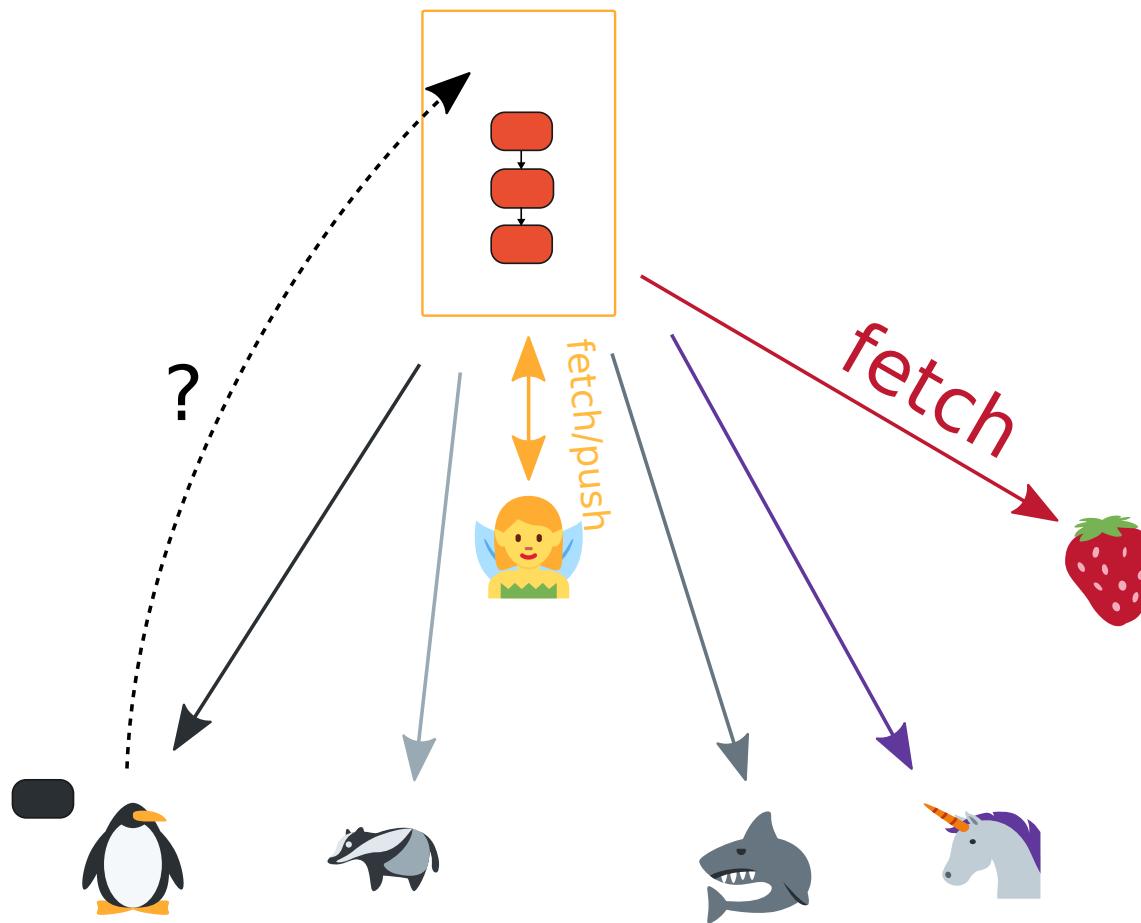
Repository



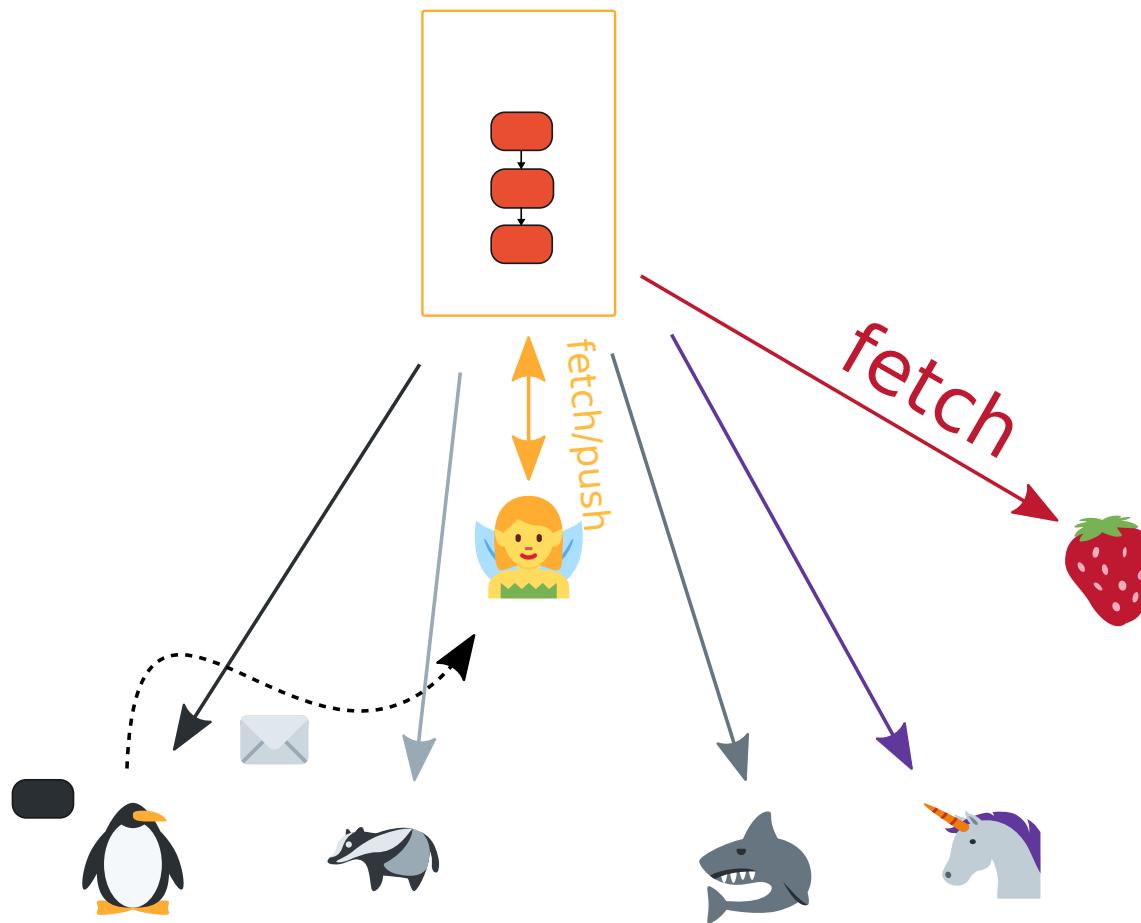
Pull Requests



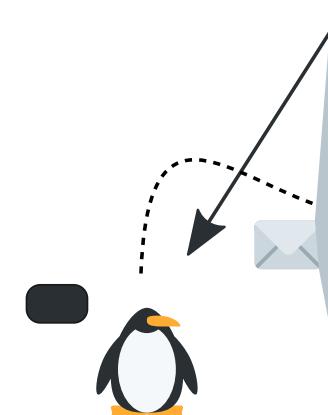
Pull Requests



Pull Requests



Pull Requests



```
From      Gabriel Krisman Bertazi <>
Subject   [PATCH RFC 2/2] futex: Implement mechanism
Date      Tue, 30 Jul 2019 18:06:02 -0400

This is a new futex operation, called FUTEX_WAIT_MULTIPLE, which allows
a thread to wait on several futexes at the same time, and be awoken by
any of them. In a sense, it implements one of the features that was
supported by pooling on the old FUTEX_FD interface.

My use case for this operation lies in Wine, where we want to implement
a similar interface available in Windows, used mainly for event
handling. The wine folks have an implementation that uses eventfd, but
it suffers from FD exhaustion (I was told they have application that go
to the order of multi-million FDs), and higher CPU utilization.

    ...

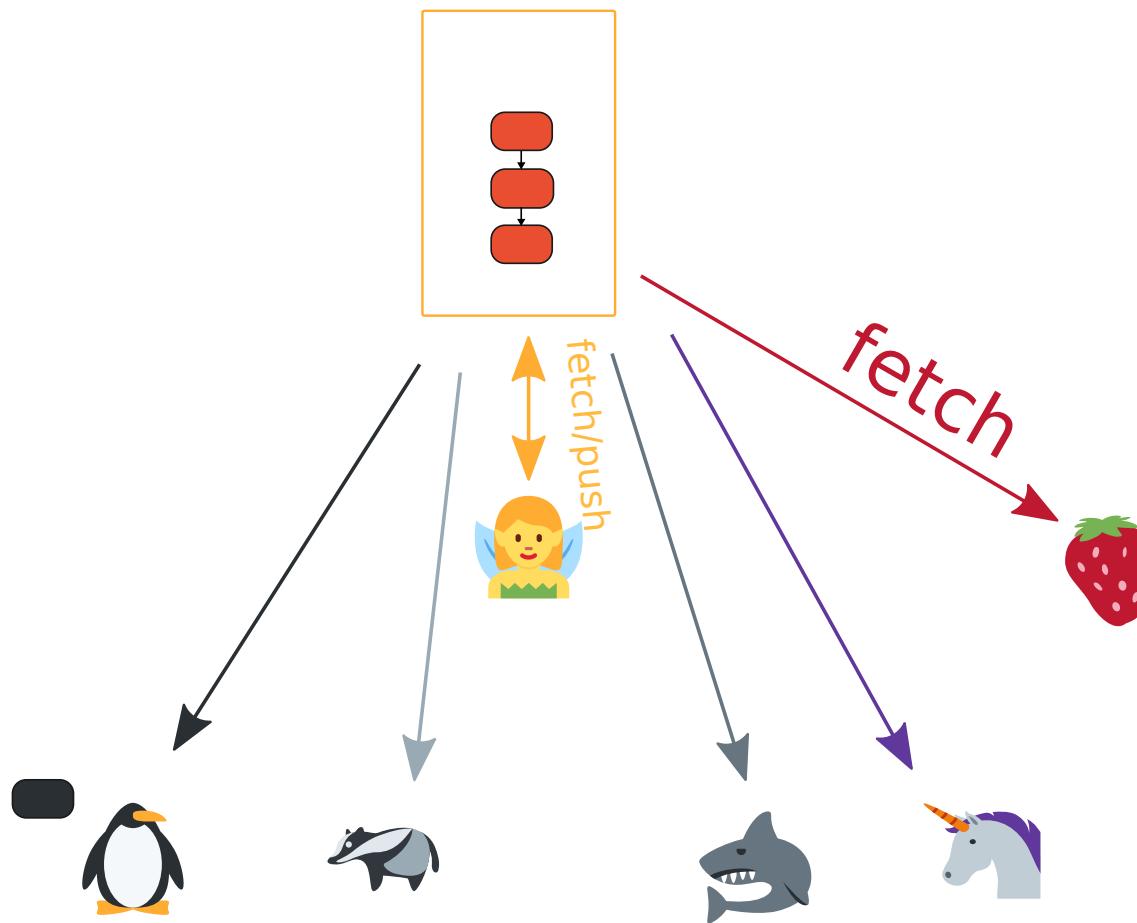
Signed-off-by: Zebediah Figura <z.figural2@gmail.com>
Signed-off-by: Steven Noonan <steven@valvesoftware.com>
Signed-off-by: Pierre-Loup A. Griffais <pgriffais@valvesoftware.com>
Signed-off-by: Gabriel Krisman Bertazi <krisman@collabora.com>
---
diff --git a/include/uapi/linux/futex.h b/include/uapi/linux/futex.h
index a89eb0accd5e..2401c4cf5095 100644
--- a/include/uapi/linux/futex.h
+++ b/include/uapi/linux/futex.h
@@ -21,6 +21,7 @@
#define FUTEX_WAKE_BITSET      10
#define FUTEX_WAIT_REQUEUE_PI 11
#define FUTEX_CMP_REQUEUE_PI 12
+#define FUTEX_WAIT_MULTIPLE   13

#define FUTEX_PRIVATE_FLAG     128
#define FUTEX_CLOCK_REALTIME   256
@@ -150,4 +151,10 @@ struct robust_list_head {
    ((op & 0xf) << 28) | ((cmp & 0xf) << 24)
    | ((oparg & 0xffff) << 12) | (cmparg & 0xffff)      \

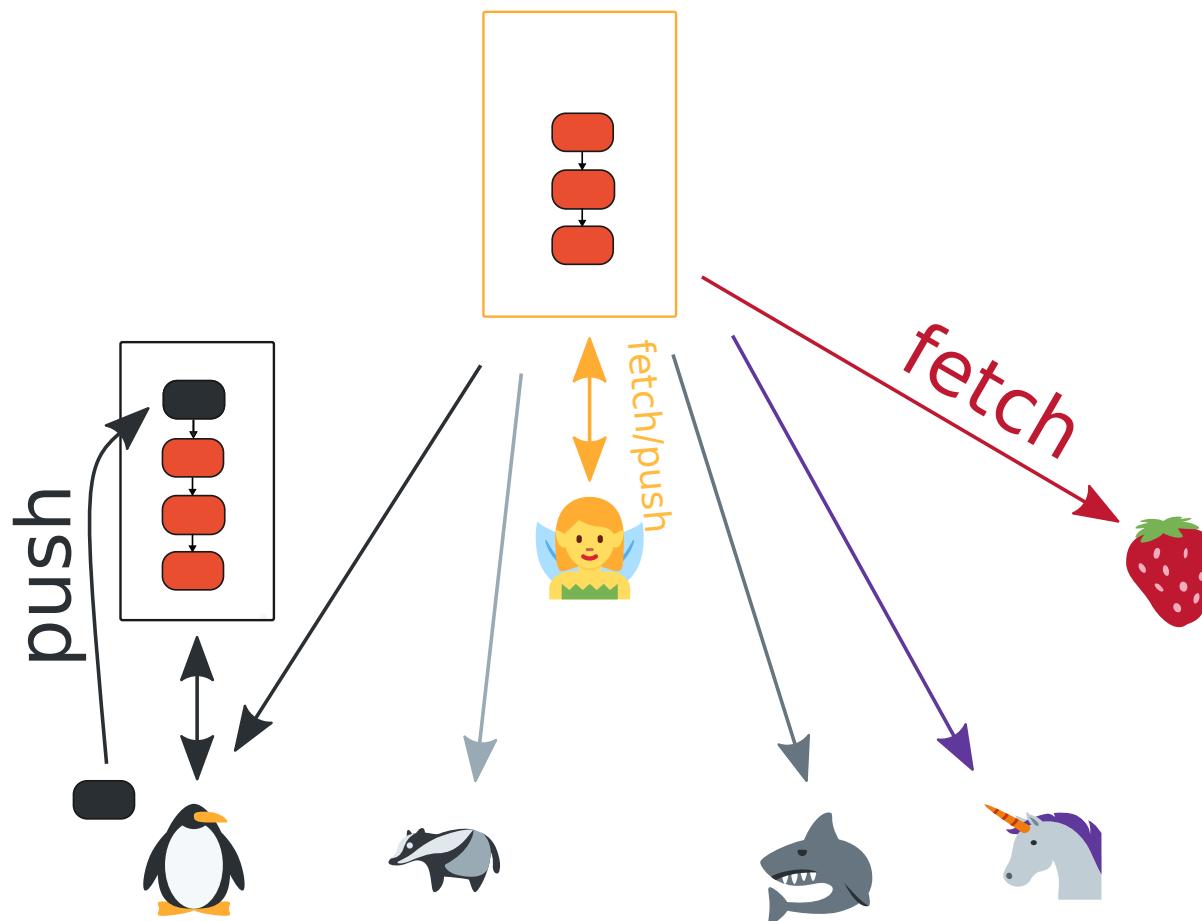

+struct futex_wait_block {
+    __u32 __user *uaddr;
+    __u32 val;
+    __u32 bitset;
+};
```



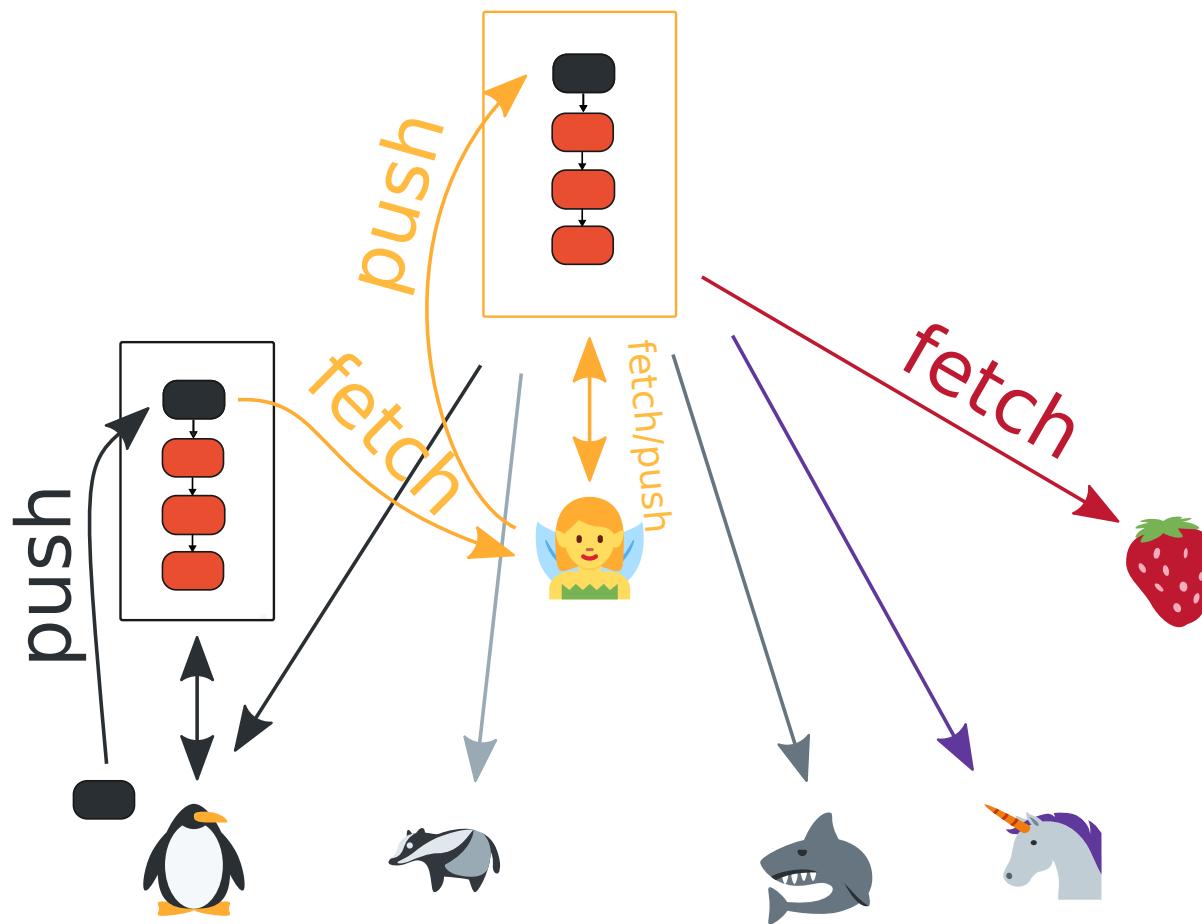
Pull Requests



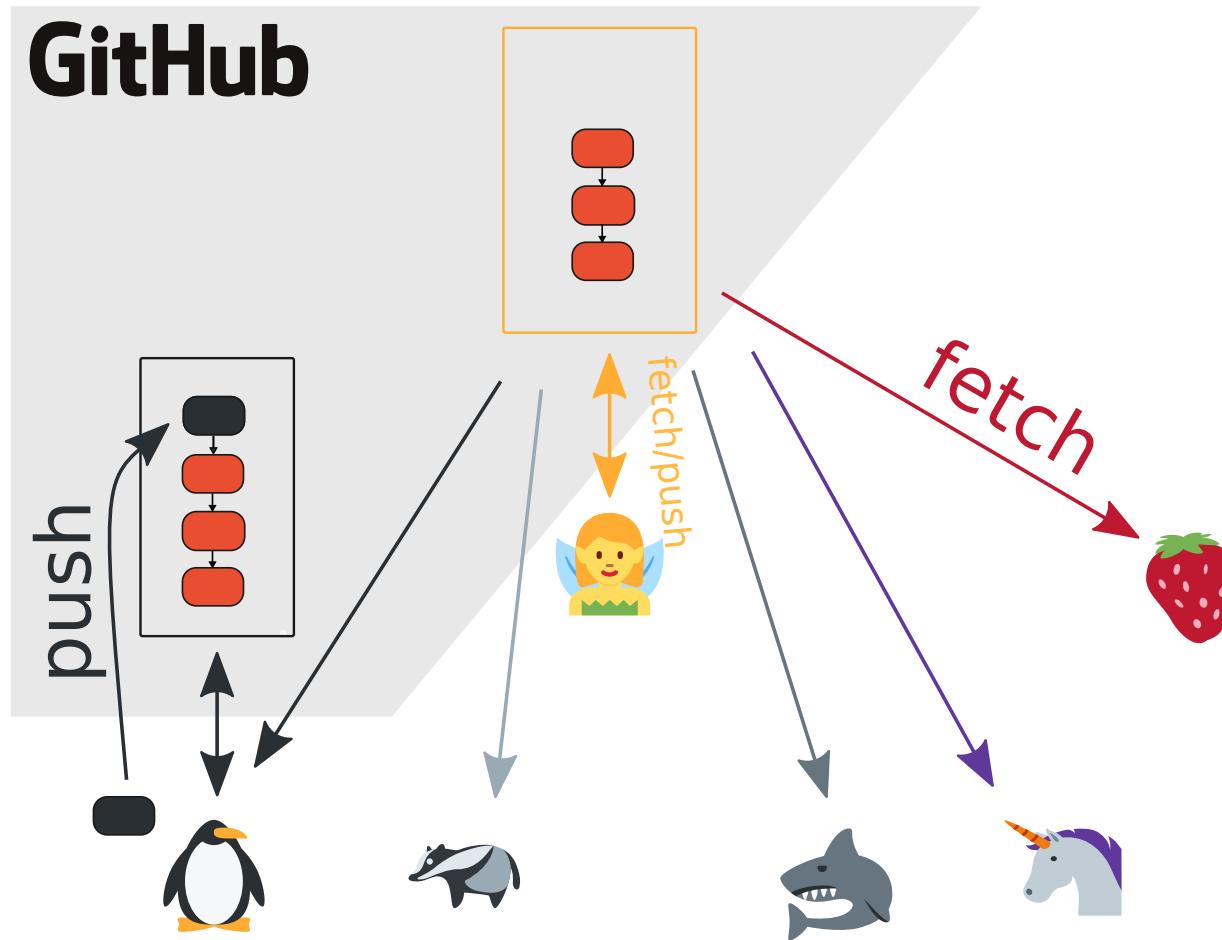
Pull Requests



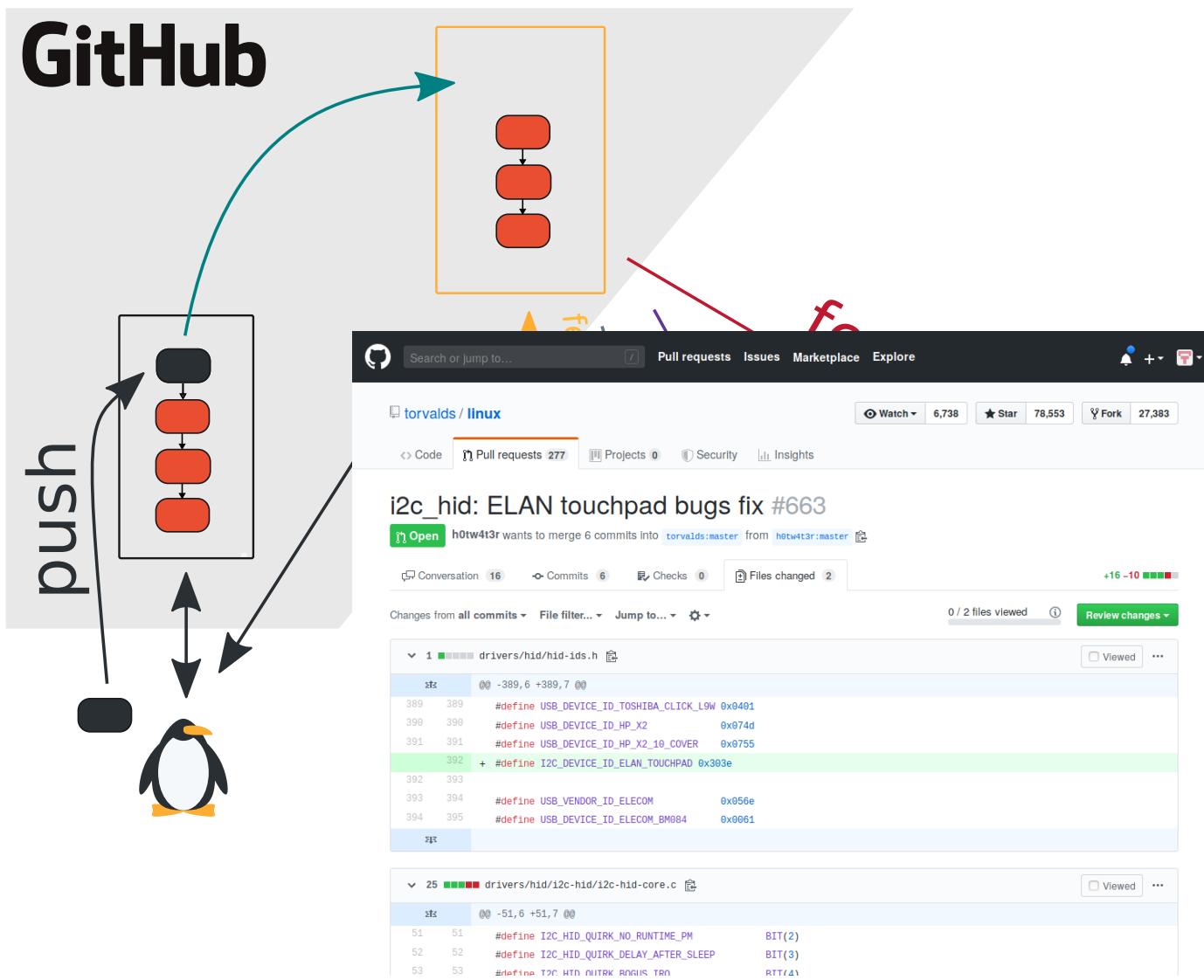
Pull Requests



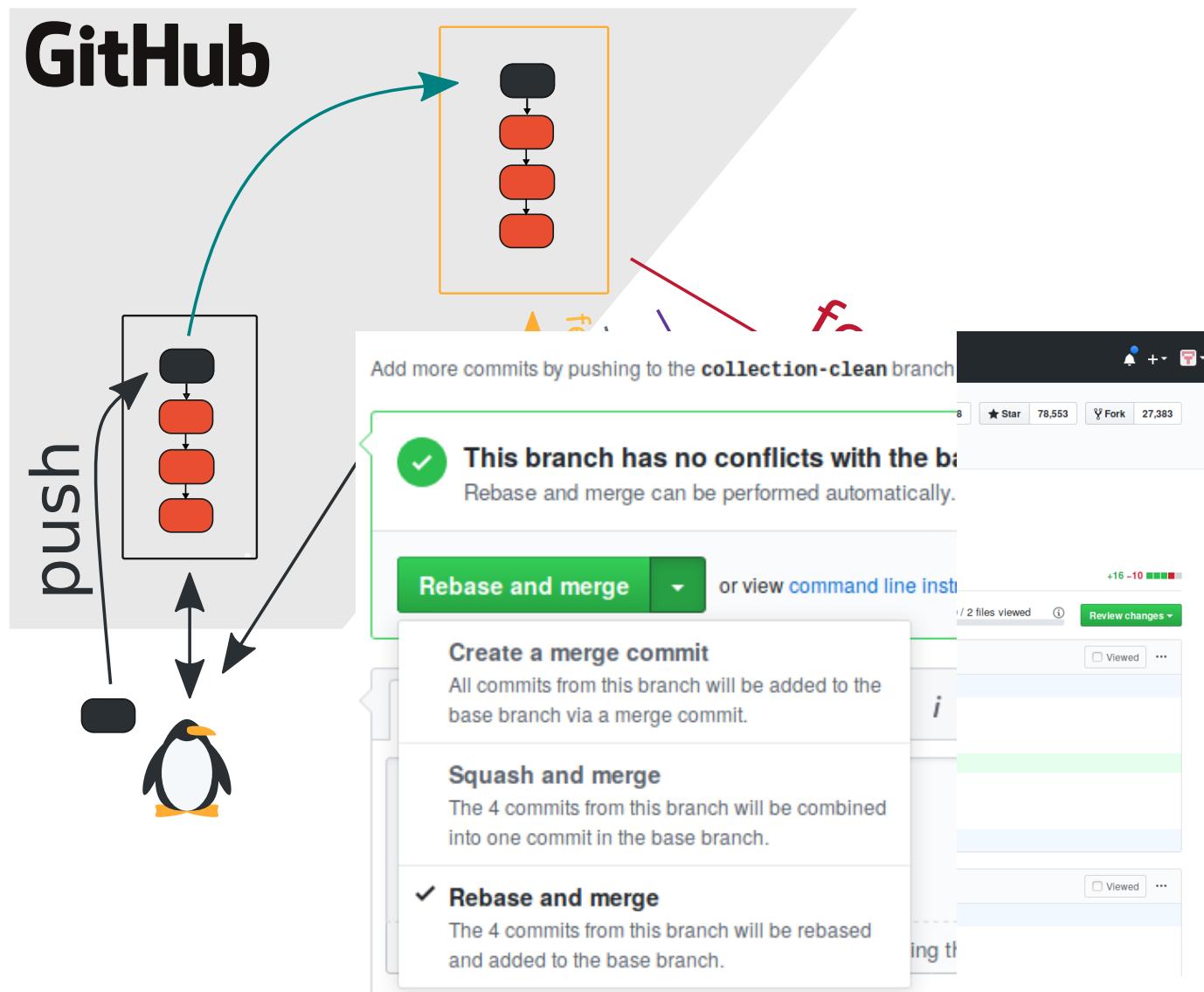
Pull Requests



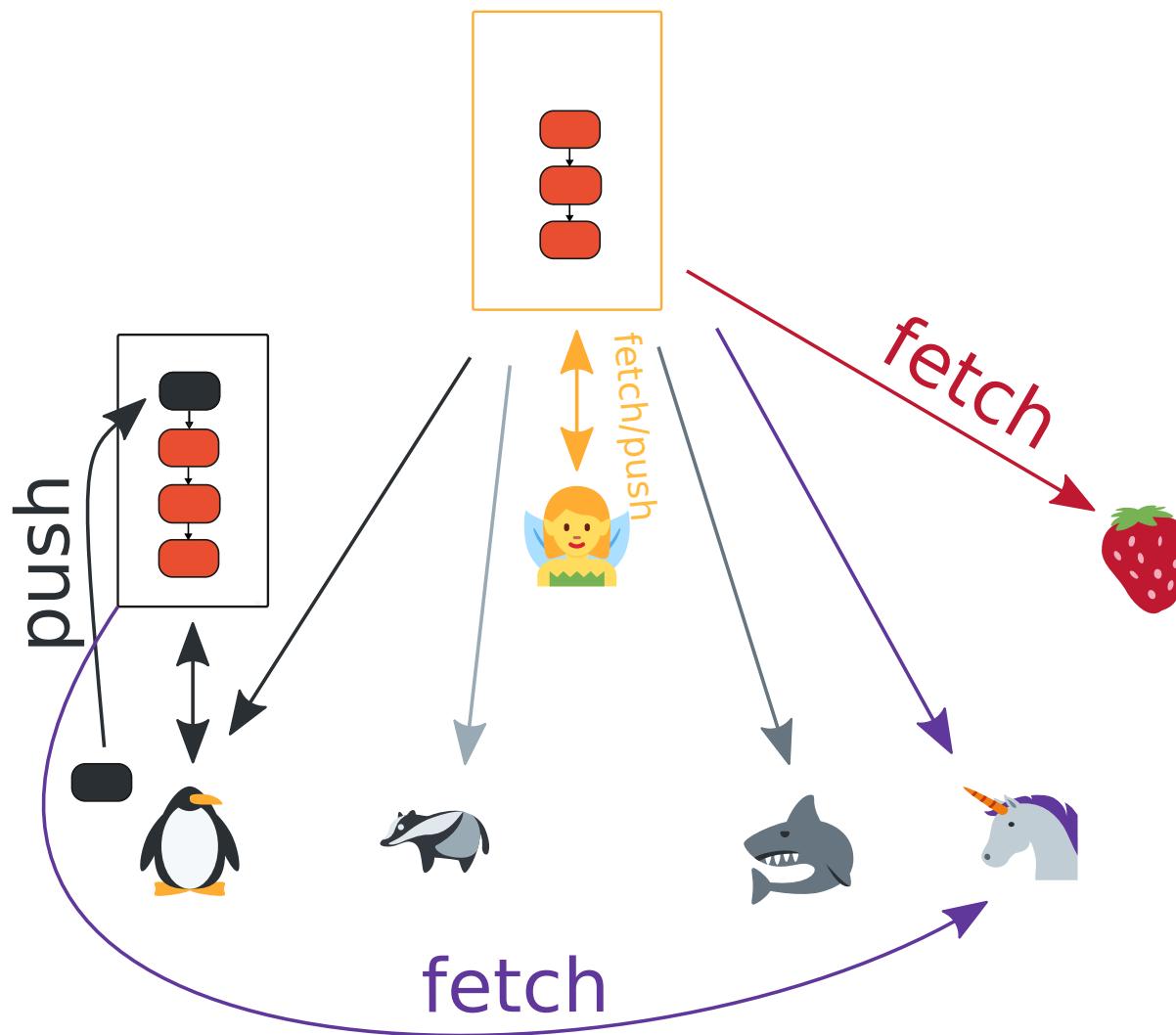
Pull Requests



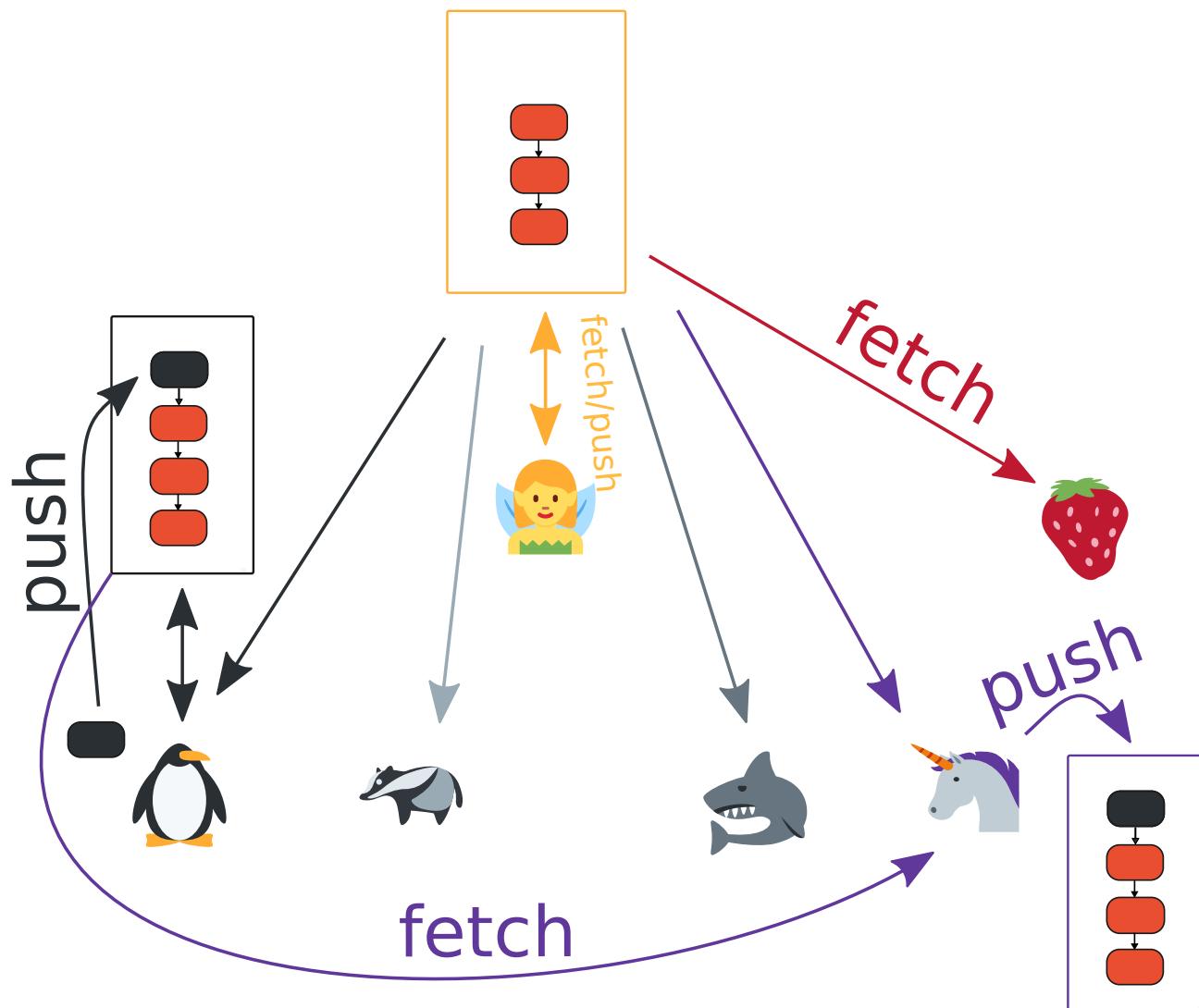
Pull Requests



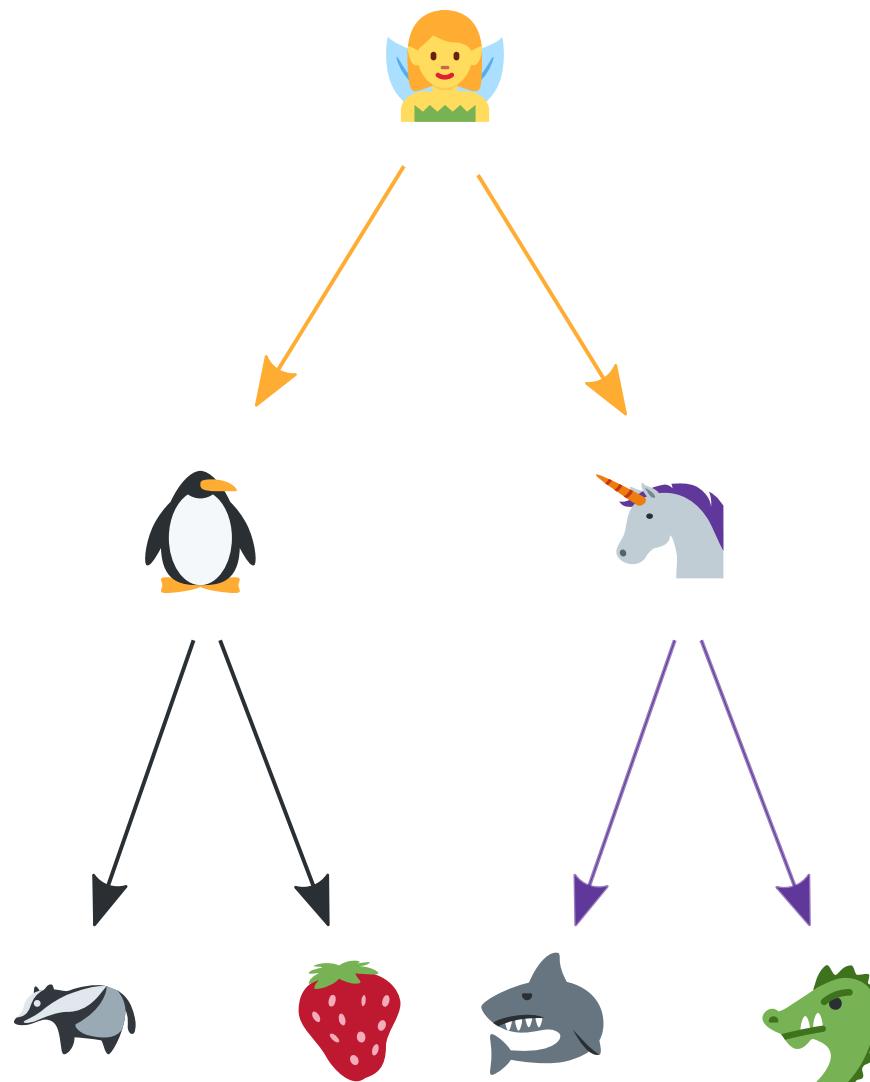
Pull Requests



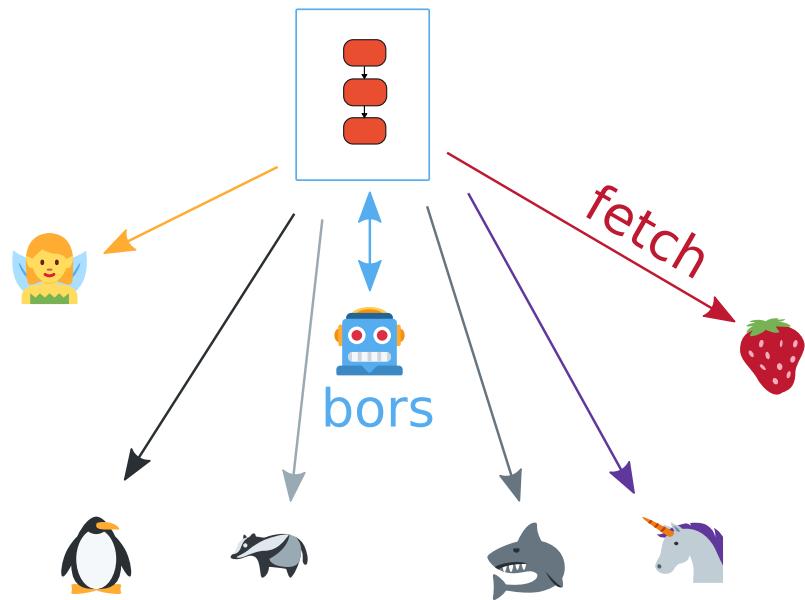
Pull Requests



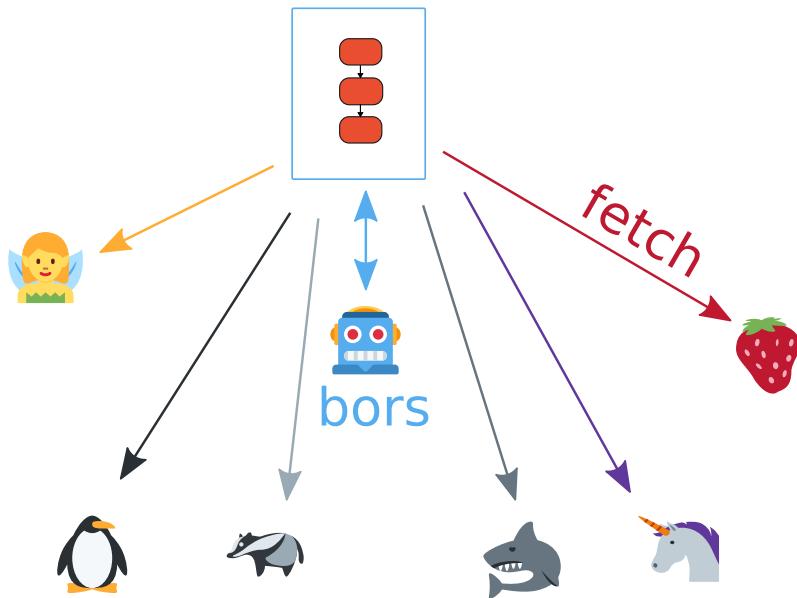
Pull Requests



Pull Requests & BORS



Pull Requests & BORS



Most tools just use `try_run`.

Reply...

alexcrichton commented yesterday

@bors: r+

Sounds reasonable to me!

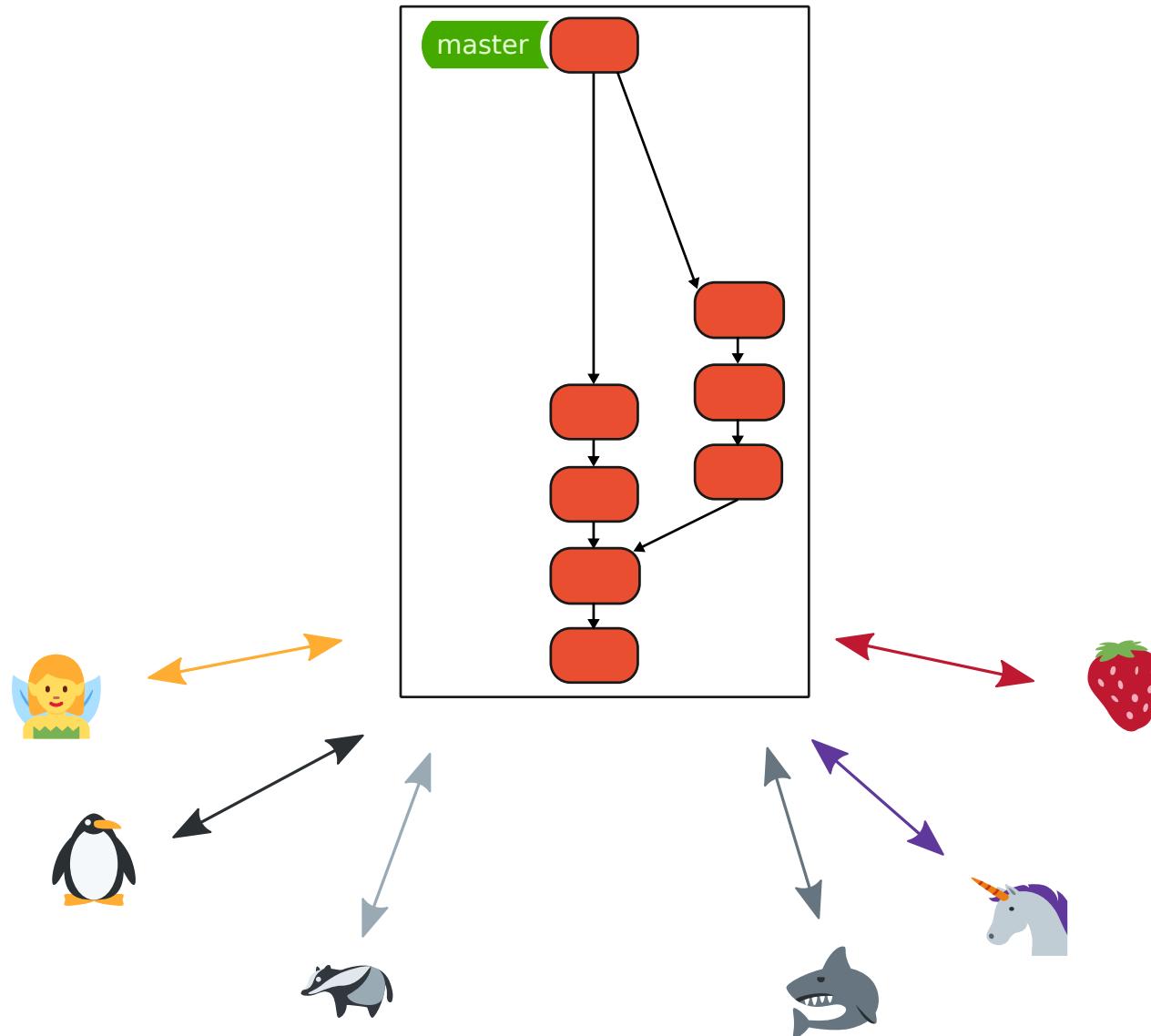
bors commented yesterday

Commit [30f61de](#) has been approved by alexcrichton

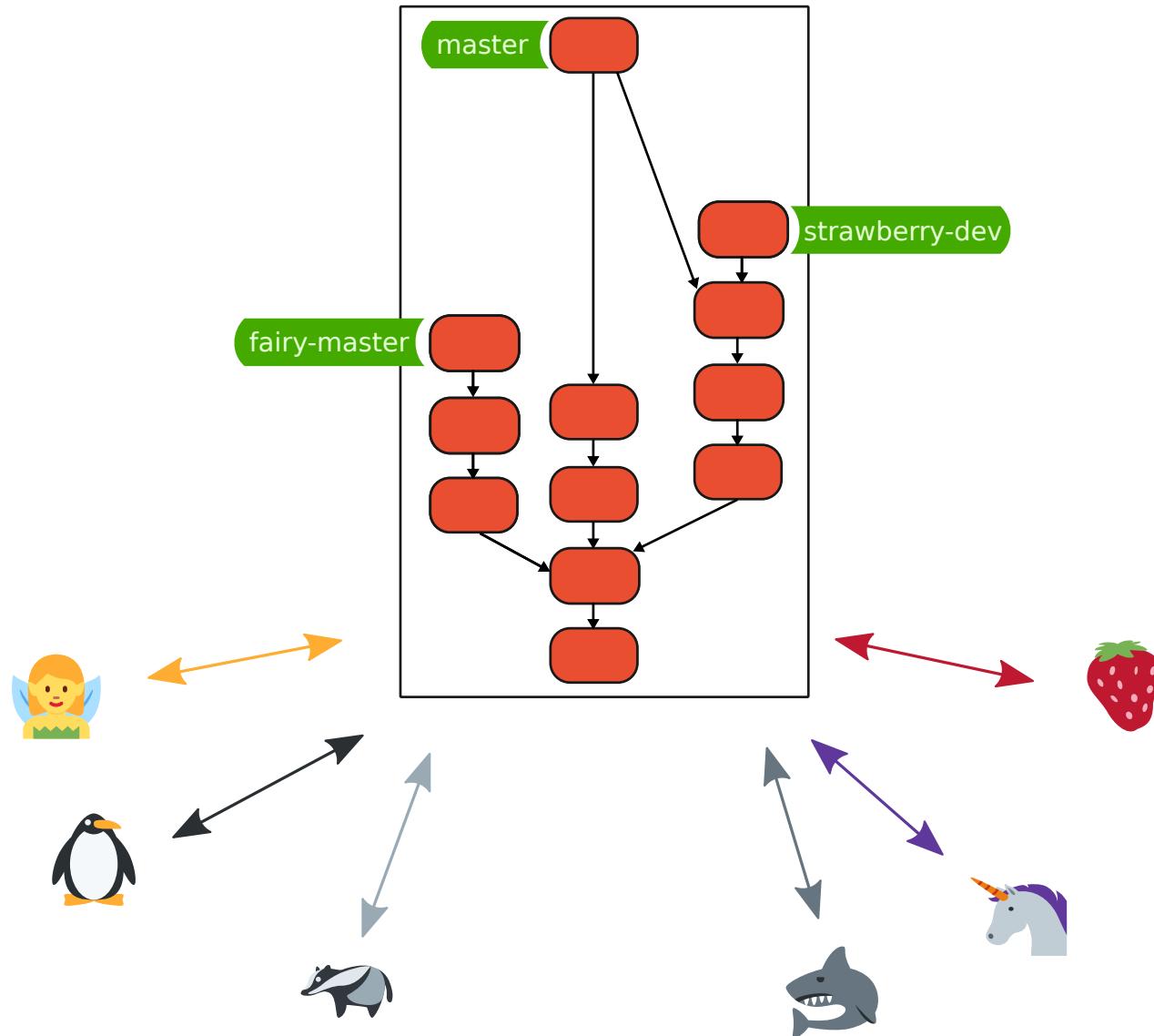
bors added `S-waiting-on-bors` and removed `S-waiting-on-review` labels yesterday

Branching Policy

Simple Branch Policy



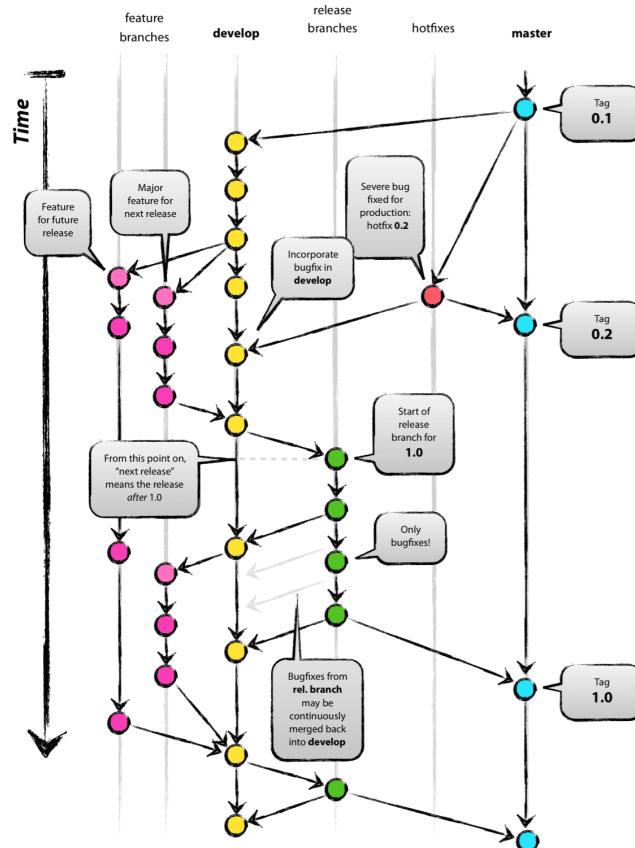
Simple Branch Policy



"Successful" Branching model

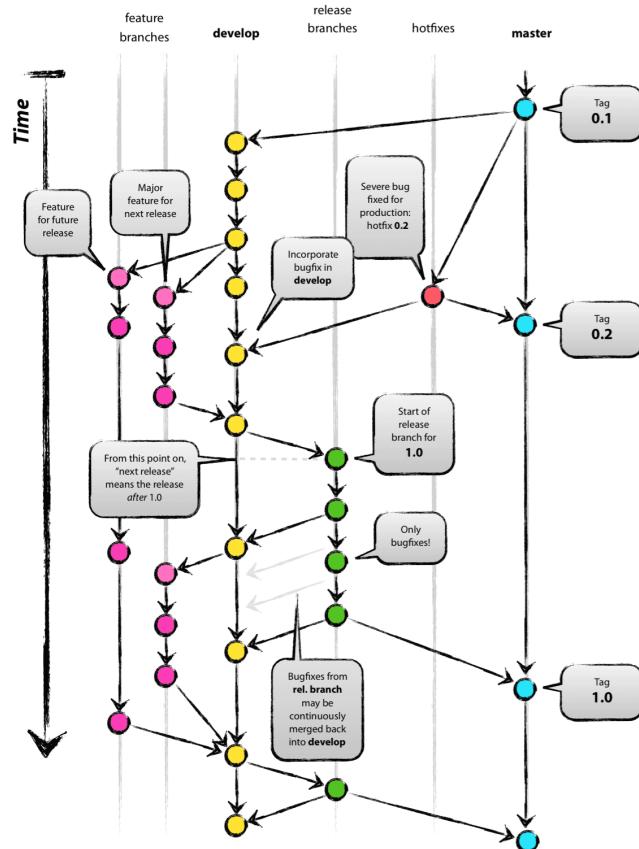
"Successful" Branching model

<https://nvie.com/posts/a-successful-git-branching-model/>



"Successful" Branching model

<https://nvie.com/posts/a-successful-git-branching-model/>



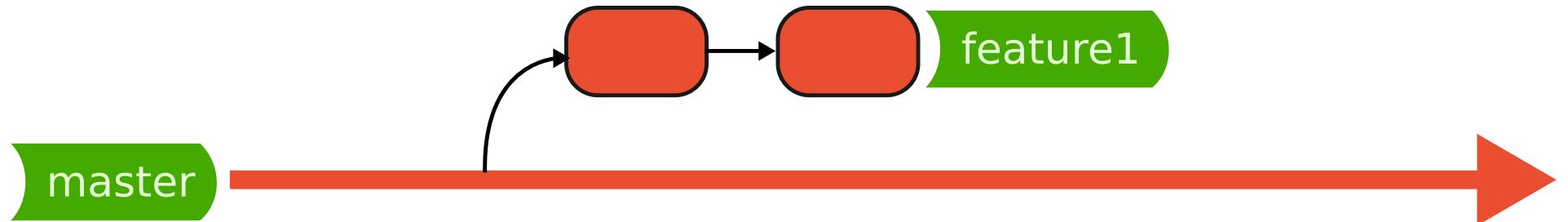
<https://barro.github.io/2016/02/a-successful-git-branching-model-considered-harmful/>

Feature branches

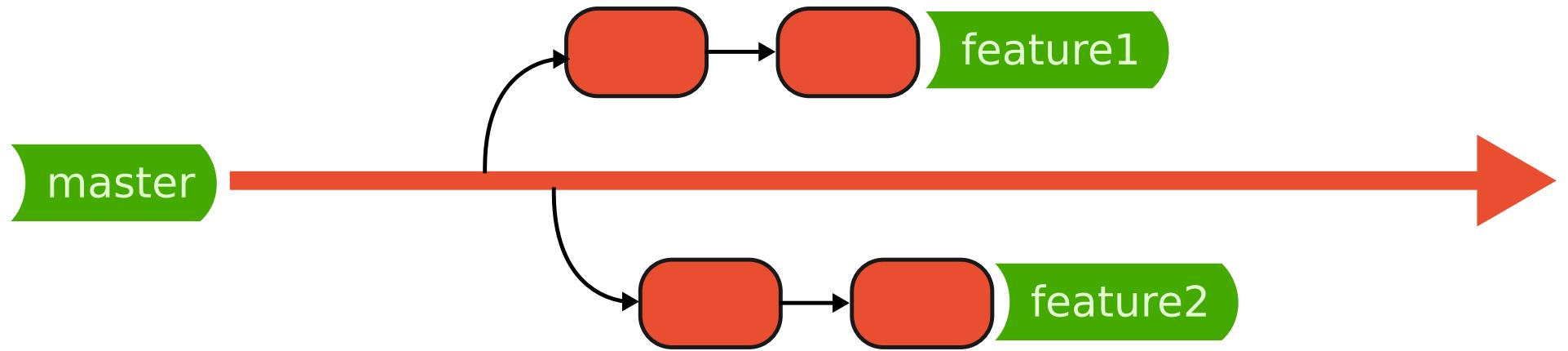
master



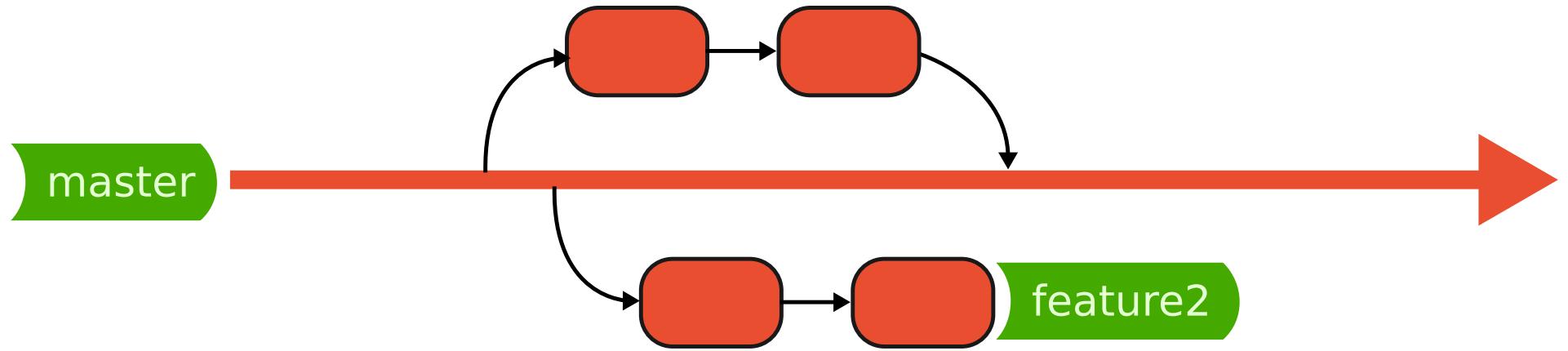
Feature branches



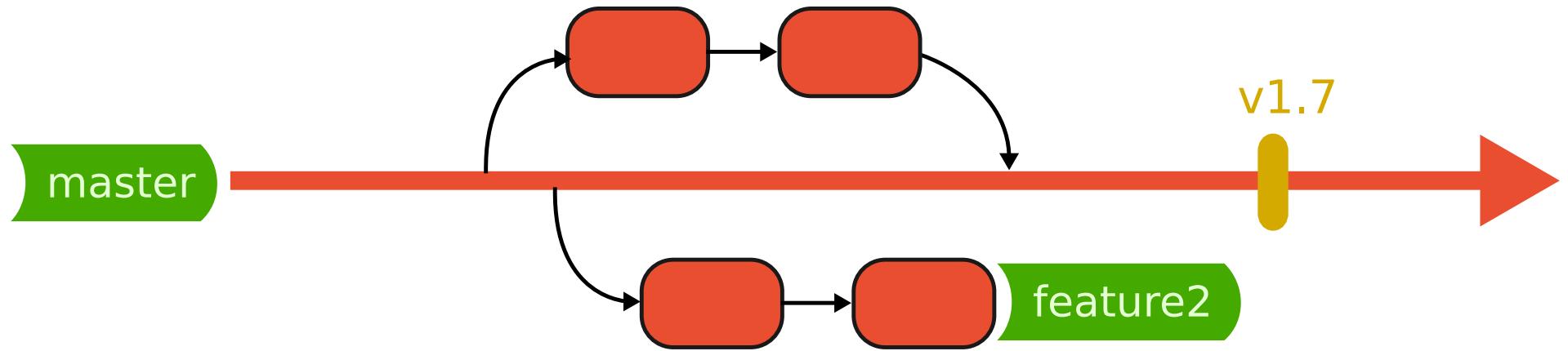
Feature branches



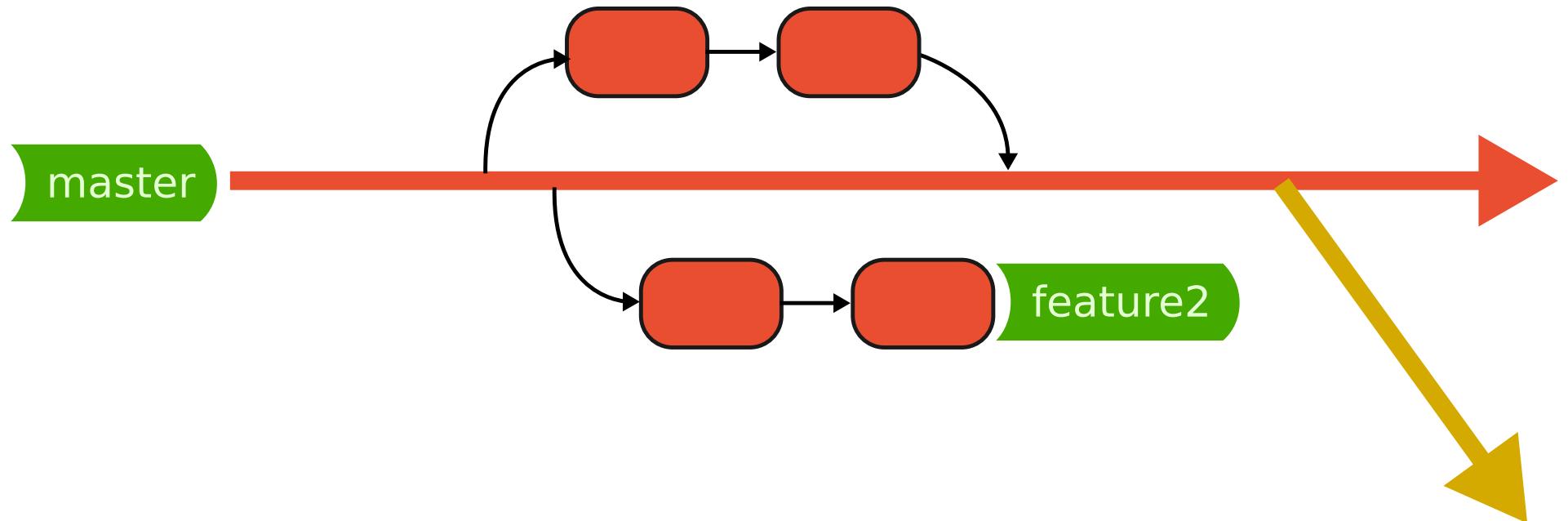
Feature branches



Feature branches

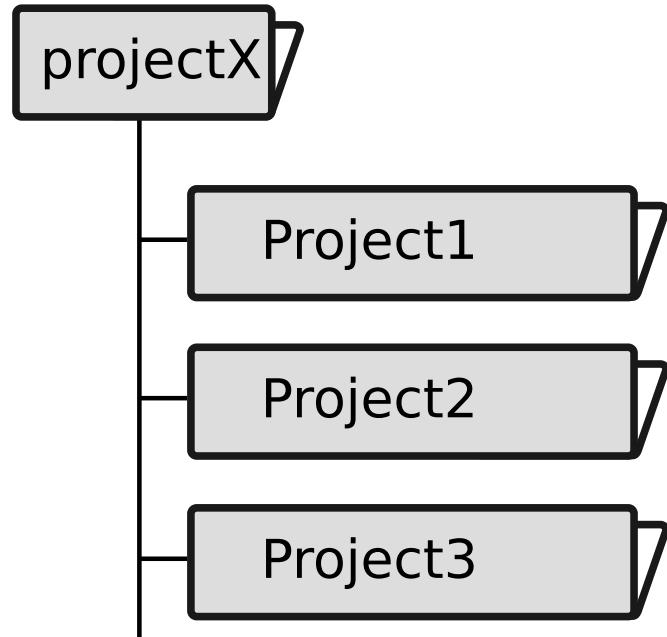


Feature branches

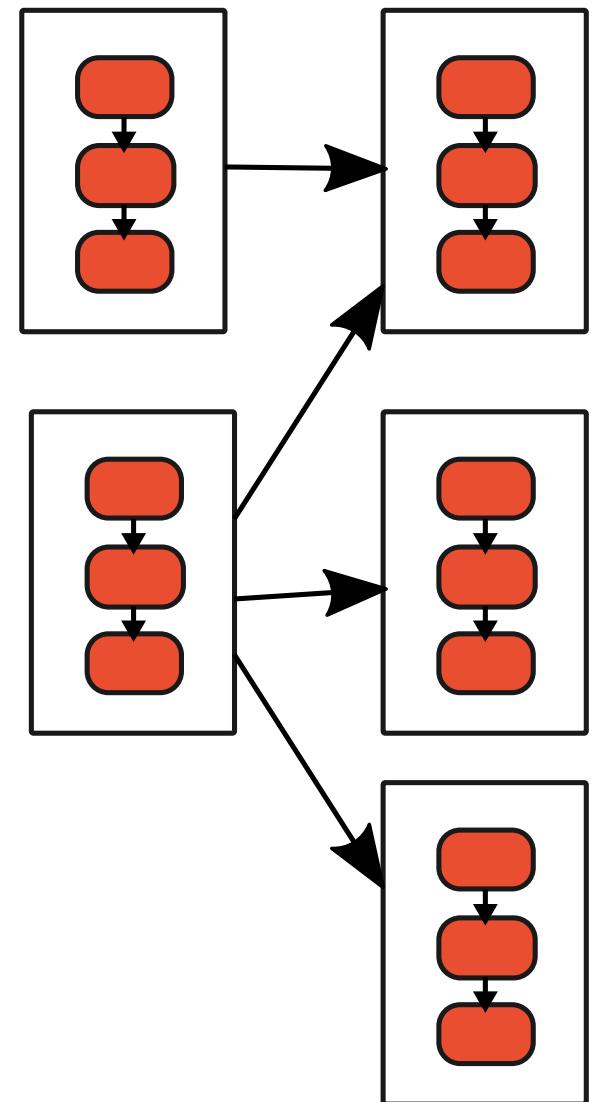
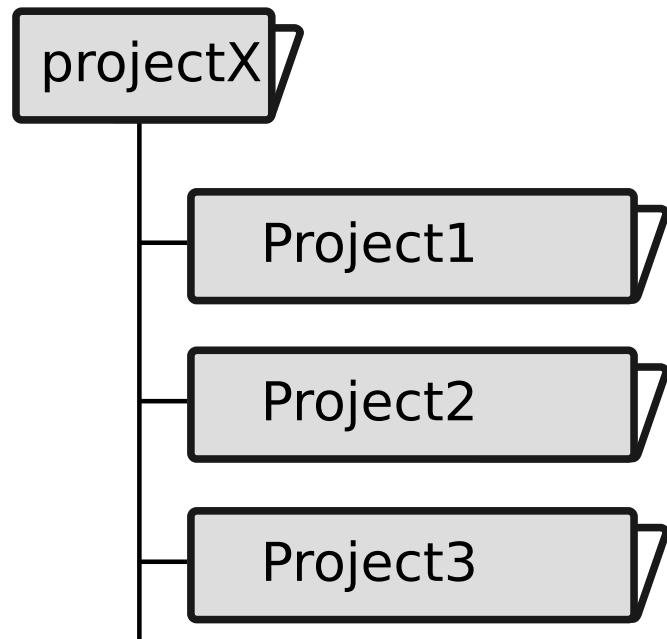


Mono-repo vs. *Multi-repo*

Mono-repo vs. *Multi-repo*



Mono-repo vs. *Multi-repo*



Commit Messages

git commit -m "changes"



Writing

Useless Git Commit Messages

O RLY?

@ThePracticalDev

Commit Messages

Commit messages are the text descriptions that accompany code commits in version control systems like Git. They serve as documentation for the changes made in each commit.

Good commit messages are descriptive and follow a standard format:

- Start with a verb (e.g., "Add", "Fix", "Remove")

- Describe the change (e.g., "Add support for Python 3.8", "Fix bug in calculate function")

- End with a period (e.g., ".")

For example, a good commit message might look like this:

Add support for Python 3.8. (.)

Bad commit messages, on the other hand, might look like this:

"fixes a bug in calculate function" (.)

or

"fixes a bug" (.)

These messages are less descriptive and don't provide clear context for the changes made.

Commit messages are important because they help other developers understand the purpose of each commit and how it fits into the overall project history.

They also make it easier to search for specific changes and track down bugs or issues.

By following best practices for commit messages, you can ensure that your code is well-documented and easy to maintain over time.

Remember, a good commit message is descriptive, follows a standard format, and ends with a period.

With these tips in mind, you'll be able to write effective commit messages that help you and your team stay organized and productive.

Rule 1 first line matters

Commit Messages

Rule 1 first line matters

Rule 2 if more lines, leave second one empty

Rule 1 first line matters

Rule 2 if more lines, leave second one empty

Rule 3 consistency

Commit Messages

Project: Visual Studio Code

Commits on Aug 5, 2019

Don't include File schema in web for file picker	 alexr00 committed 21 hours ago ✓	 5535ee4 
fix compilo	 jrieken committed 21 hours ago ✓	 b4b5e5a 
consistent dto namings	 jrieken committed 21 hours ago	 32da48e 
debt - test all optional modules	 bpasero committed 21 hours ago ✓	 41df3bc 
strict prop init terminal	 Tyriar committed 21 hours ago	 046c3f5 
fixes #78505	 joaomoreno committed 21 hours ago ✓	 15a523f 
Fix nav mode and windows shell helper strict prop init	 Tyriar committed 21 hours ago	 873fdf8 
	 jrieken committed 22 hours ago ✓	 051b669 

Commit Messages

Project: Linux

mm/memory_hotplug.c: remove unneeded return for void function ...	 Weitao Hou authored and torvalds committed 3 days ago	 aa4996b 
mm/migrate.c: initialize pud_entry in migrate_vma() ...	 Ralph Campbell authored and torvalds committed 3 days ago	 7b358c6 
coredump: split pipe command whitespace before expanding template ...	 pabs3 authored and torvalds committed 3 days ago	 315c692 
page flags: prioritize kasan bits over last-cpuid ...	 arndb authored and torvalds committed 3 days ago	 ee38d94 
ubsan: build ubsan.c more conservatively ...	 arndb authored and torvalds committed 3 days ago	 af700ea 
kasan: remove clang version check for KASAN_STACK ...	 arndb authored and torvalds committed 3 days ago	 ebb6d35 
mm: compaction: avoid 100% CPU usage during compaction when a task is... ...	 gormanm authored and torvalds committed 3 days ago	 670105a 
mm: migrate: fix reference check race between __find_get_block() and	 jankara authored and torvalds committed 3 days ago	 ebdf4de 

Commit Messages

Project: Linux

Diff Old version New version Lines of context: 3 ▾ Ignore space change Line diff ▾

Author: Jernej Skrabec <jernej.skrabec@siol.net> 2019-07-13 11:07:16
Committer: Mark Brown <broonie@kernel.org> 2019-07-16 20:13:57
Parent: [3c7577d442a76c2015dd765497395fb394b78051](#) (regulator: max77650: use vsel_step)
Child: [8f46e22b5ac692b48d04bb722547ca17b66dda02](#) (regulator: axp20x: fix DCDC5 and DCDC6 for AXP803)
Branches: [master](#), [remotes/origin/master](#)
Follows: [v5.2-rc4](#)
Precedes:

regulator: axp20x: fix DCDC5 and DCDC6 for AXP803

Refactoring of the driver introduced bugs in AXP803's DCDC5 and DCDC6 regulator definitions.

In DCDC5 case, AXP803_DCDC5_1120mV_STEPS was obtained by subtracting 0x47 and 0x33. This should be 0x14 (hex) and not 14 (dec).

In DCDC6 case, axp803_dcdcd_ranges[] contains two ranges with same start and end macros, which is clearly wrong. Second range starts at 1.6V so it should use AXP803_DCDC6_1600mV_[START|END] macros. They are already defined but unused.

Fixes: [db4a555f7c4c](#) ("regulator: axp20x: use defines for masks")
Signed-off-by: Jernej Skrabec <jernej.skrabec@siol.net>
Link: <https://lore.kernel.org/r/20190713090717.347-2-jernej.skrabec@siol.net>
Signed-off-by: Mark Brown <broonie@kernel.org>

----- drivers/regulator/axp20x-regulator.c -----

```
index 15205336186..c951568994a 100644
@@ -240,7 +240,7 @@
#define AXP803_DCDC5_600mV_END \
    (AXP803_DCDC5_600mV_START + AXP803_DCDC5_600mV_STEPS)
#define AXP803_DCDC5_1120mV_START 0x33
-#define AXP803_DCDC5_1120mV_STEPS 14
+#define AXP803_DCDC5_1120mV_STEPS 20
#define AXP803_DCDC5_1120mV_END \
    (AXP803_DCDC5_1120mV_START + AXP803_DCDC5_1120mV_STEPS)
#define AXP803_DCDC5_NUM_VOLTAGES 72
@@ -774,8 +774,8 @@ static const struct regulator_linear_range axp803_dcdcd_ranges[] = {
    AXP803_DCDC5_600mV_END,
    20000),
    REGULATOR_LINEAR_RANGE(1600000,
-        AXP803_DCDC5_600mV_START,
-        AXP803_DCDC5_600mV_END,
-        AXP803_DCDC5_1600mV_START
+
```

Commit Messages

<https://xkcd.com/1296/>

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

Commit Messages

<https://xkcd.com/1296/>

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Merge branch 'asdfasjkfdlas/alkdjf' into sdkjfls-f

1. Introduction

2. Hands-on:

- Basic operations
- Branching / Merge / Rebase
- Remote repository

3. Common scenarios

4. Repository management

5. *Bonus topics*

- .git content & hooks
- Git as graph DB
- How teach newcomers