



1

WEEK2 - Feedbacks

Mathias COUSTÉ
06.02.2020

3

teams did not compete...
(because of wrong Jar name)

7

teams had runtime errors... (*GenericClientError*)

4 NPE

2 Index Out Of Bounds

1 Wrong Format Exception

7

teams had runtime errors

4 ML

What about testing?

↳ index Out Of Bounds

1

Wrong Format Exception

10

teams did not found any checkpoints

1

(out of 21)

team finished the race

\o/

1

(out of 21)



GG Flibustiers

team finished the race

\o/

```
<!-- Serialization -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>${jackson.databind.version}</version>
    <scope>provided</scope>
</dependency>
```

ADDITIONAL FACULTATIVE DELIVERY

Saturday the 8th at 6PM

TAG: WEEK2B

Same regatta as Tuesday



Git for teams

Mathias COUSTÉ
06.02.2020

1



Git Basics

Git basics

➤ Clone

➤ Pull

➤ Add

➤ Commit

➤ Push

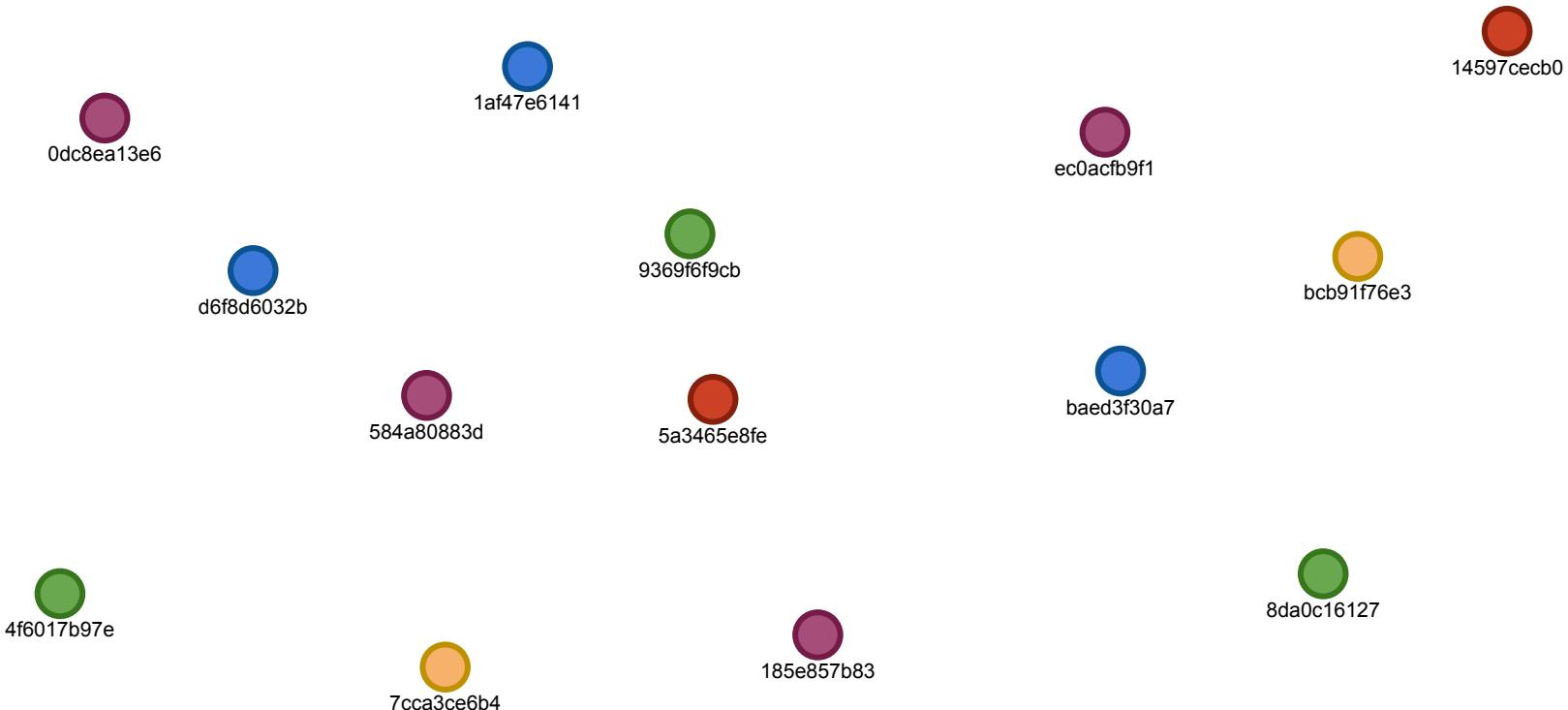
➤ Tag

2

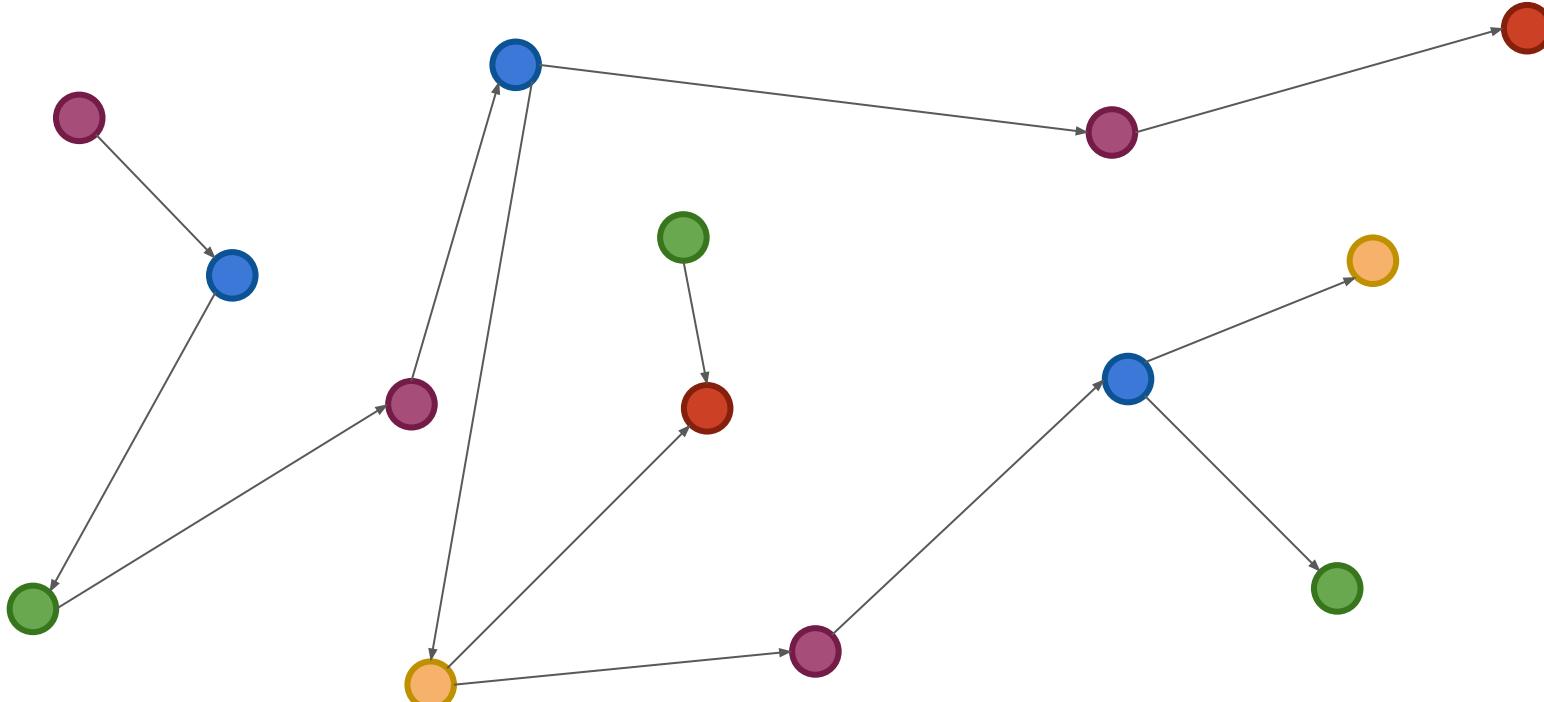


Understanding Git Branches

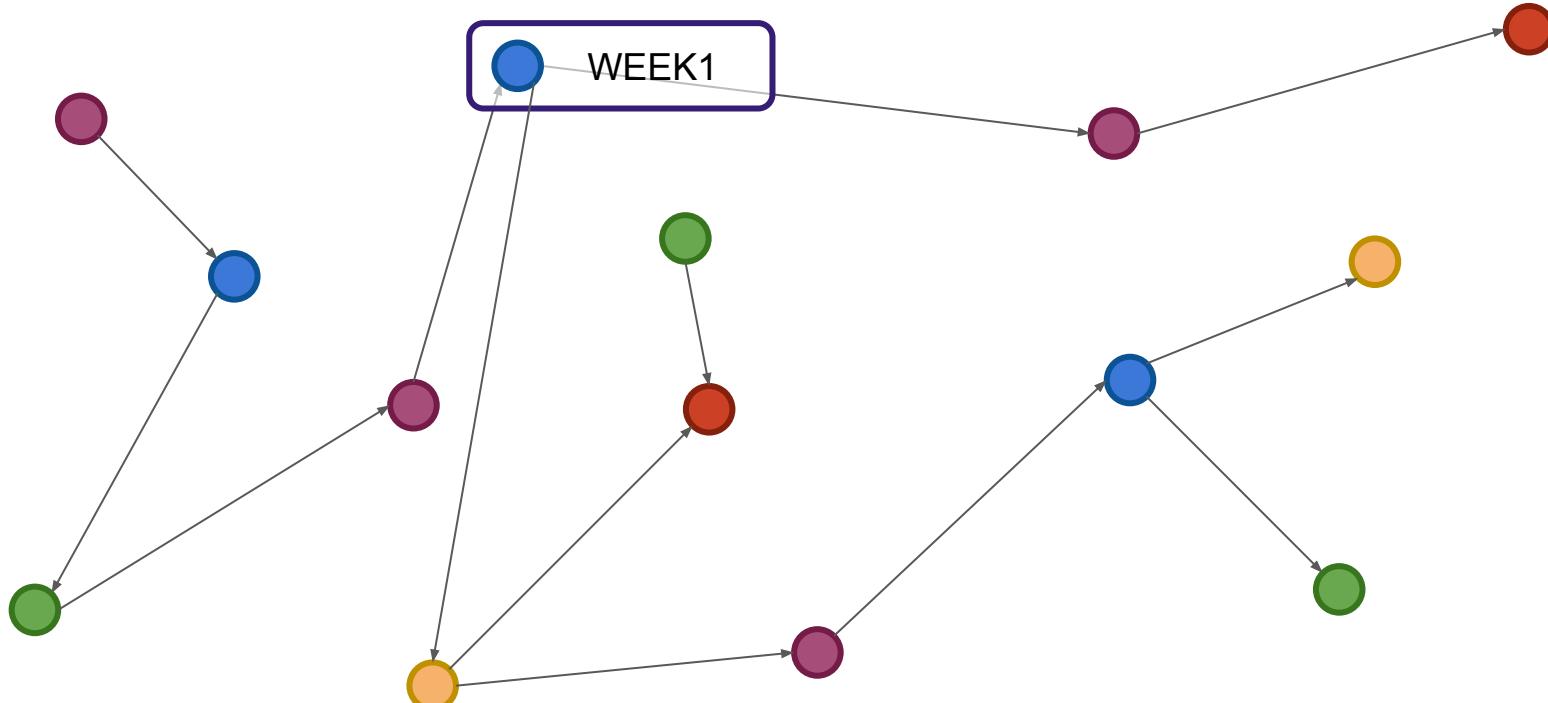
Lot of commit



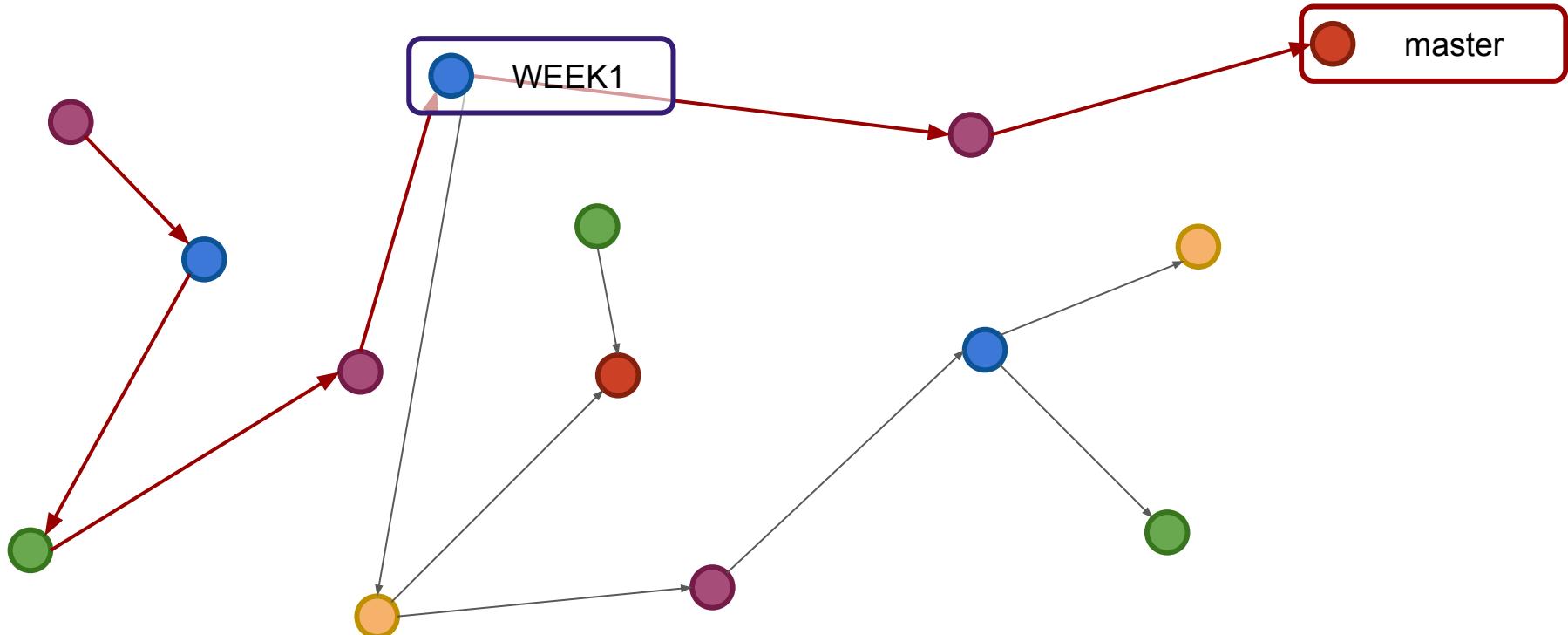
Links between commits



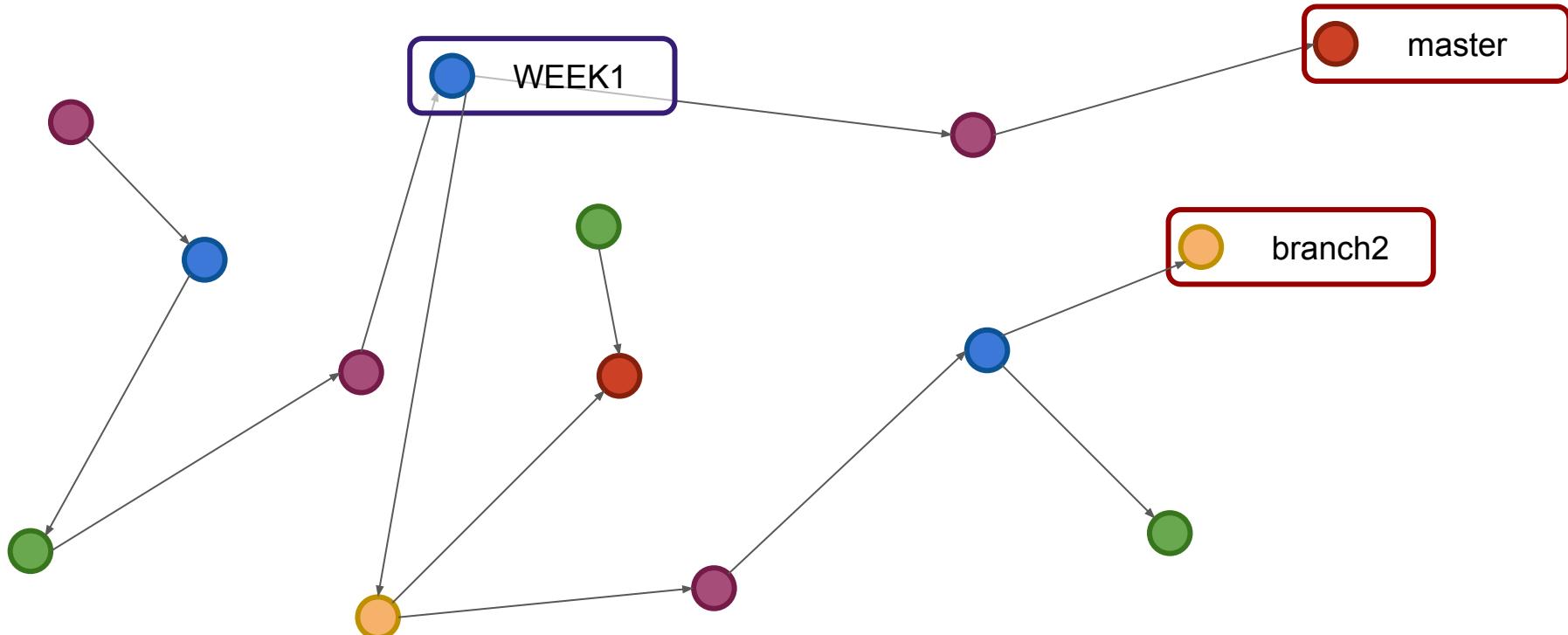
What is a tag?



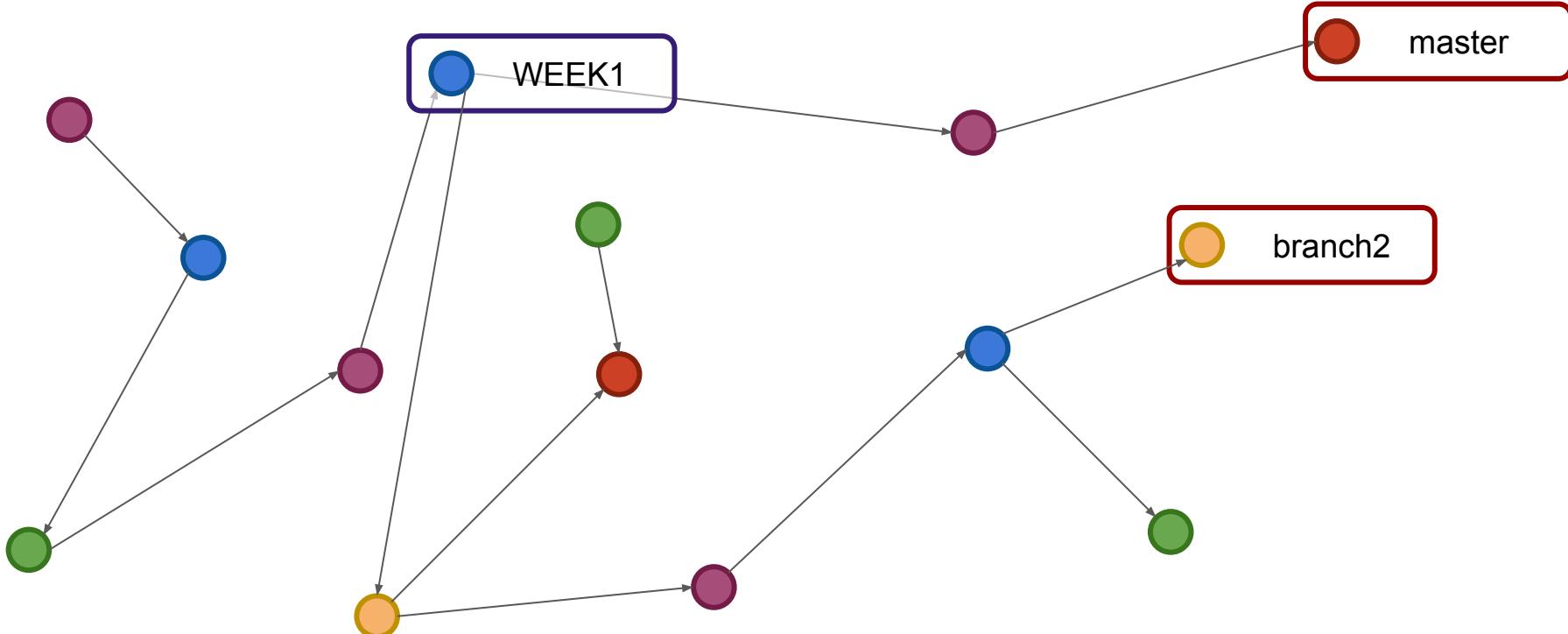
What is a branch?



What is a branch?

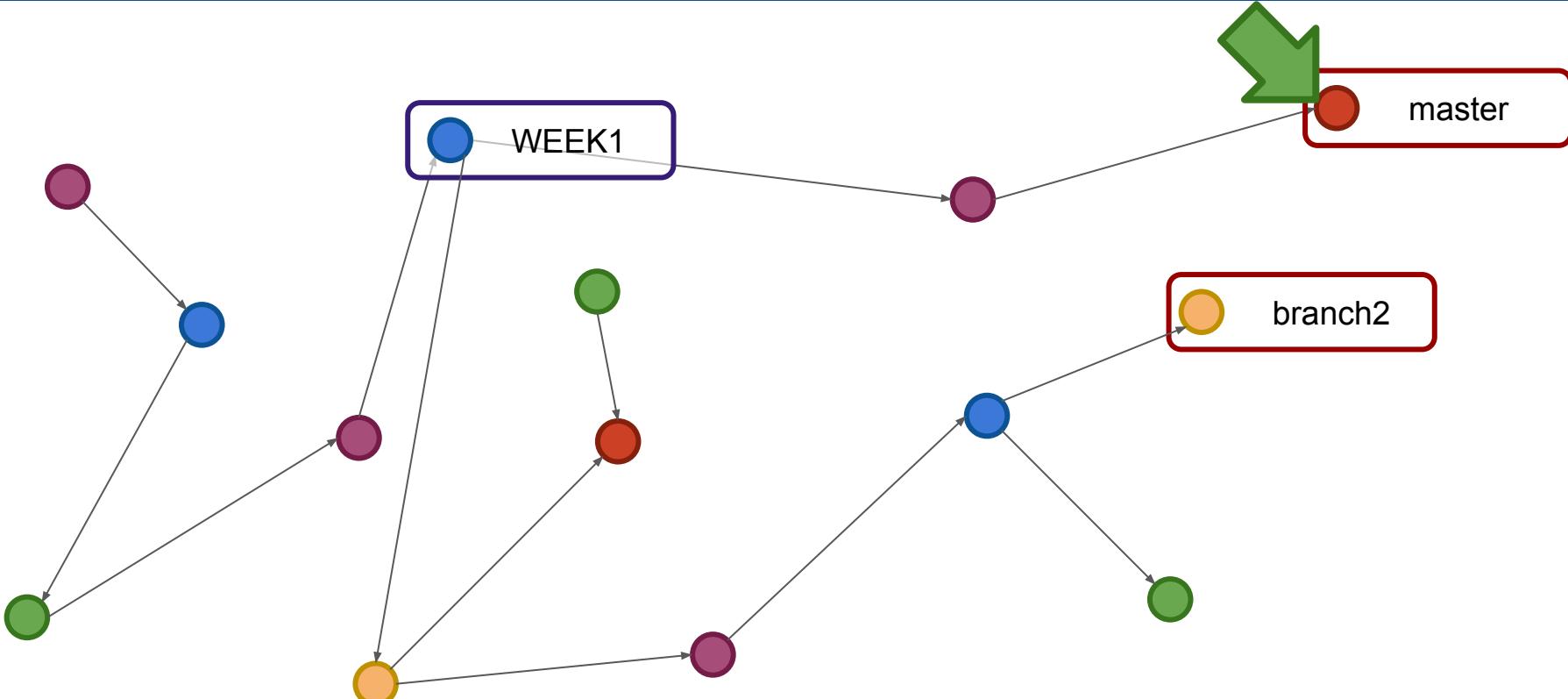


What is the “HEAD”?



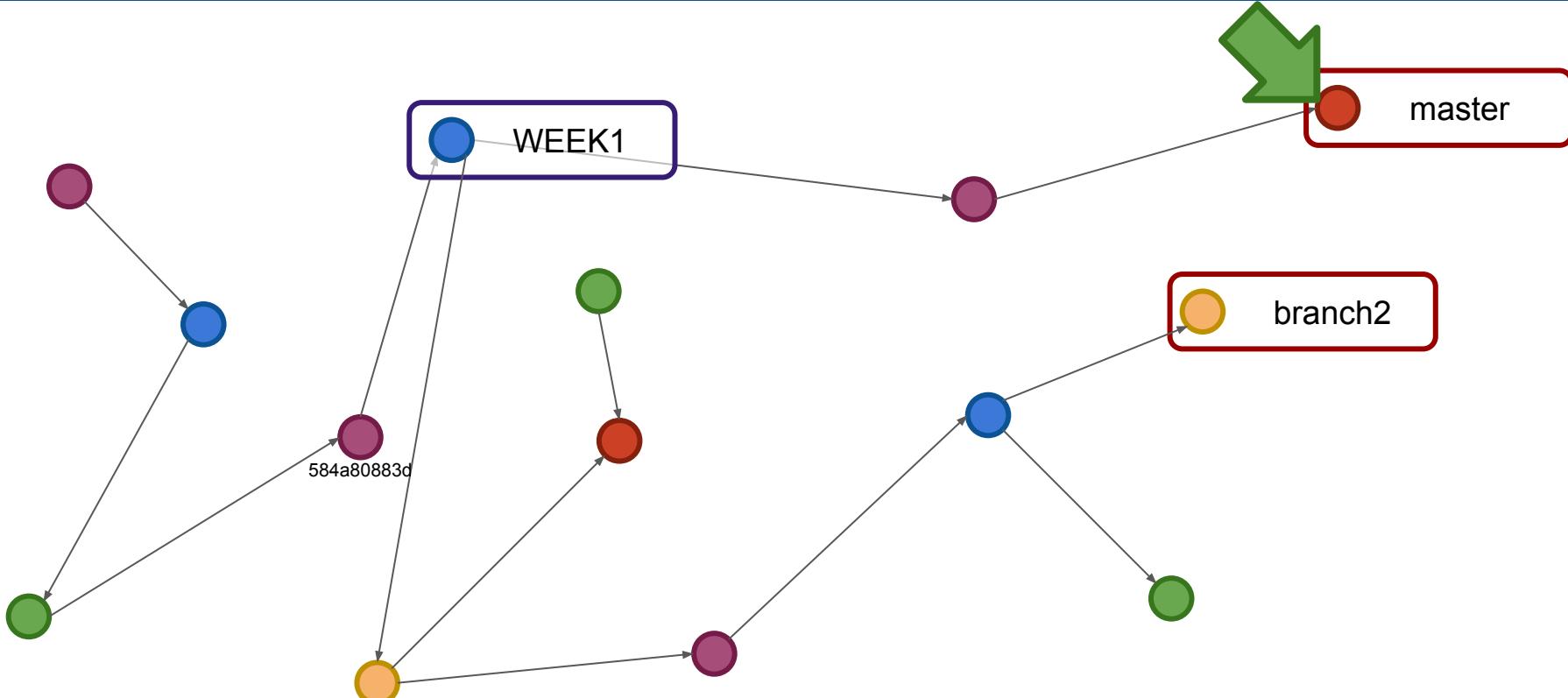
```
$ > GIT CHECKOUT master
```

What is the “HEAD”?



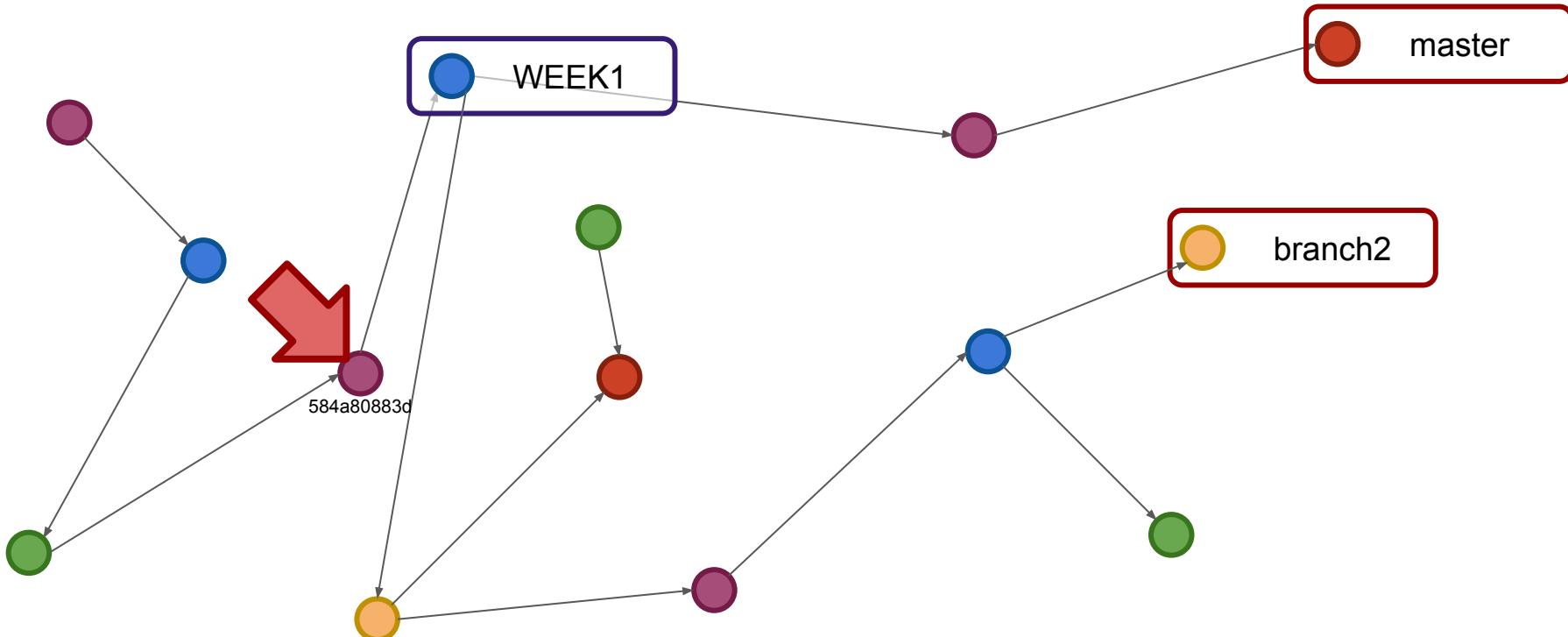
```
$ > GIT CHECKOUT master
```

What about “detached HEAD”?



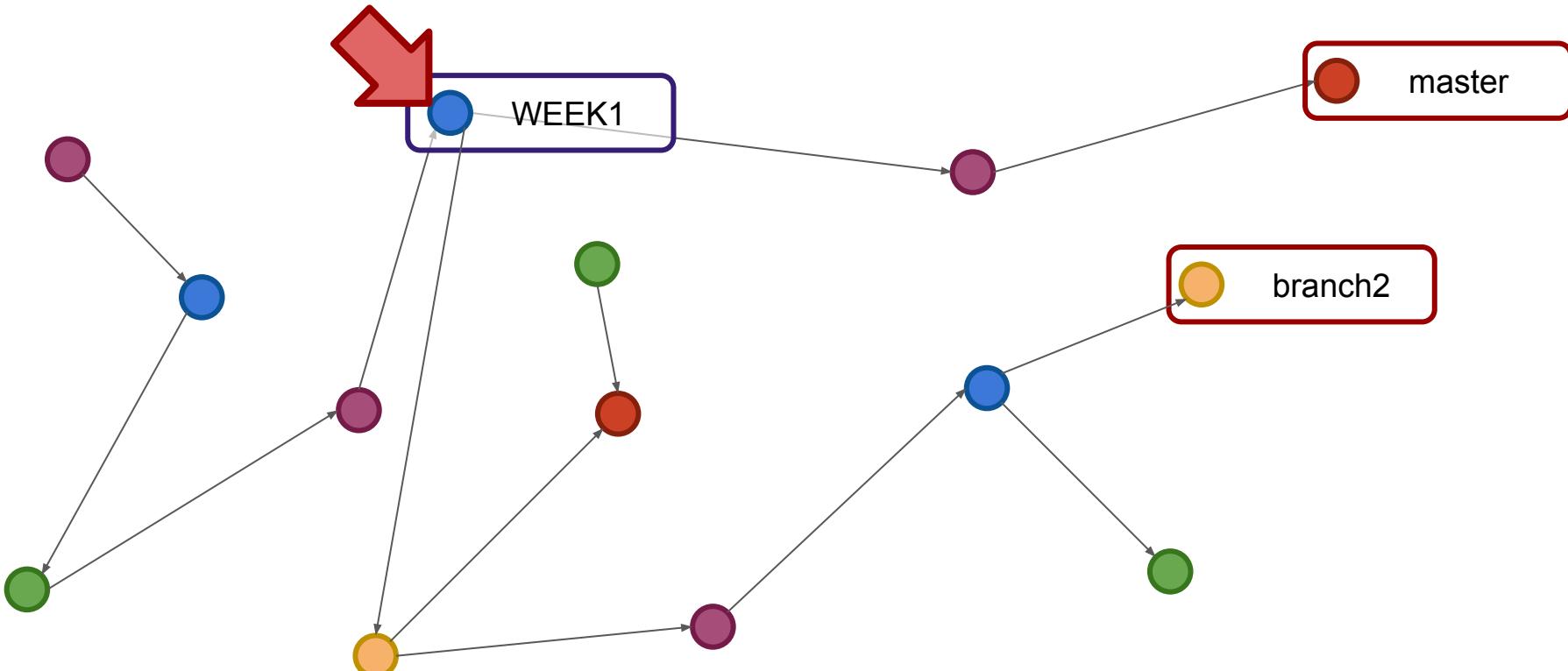
\$ > GIT CHECKOUT 84a80883D

What about “detached HEAD”?



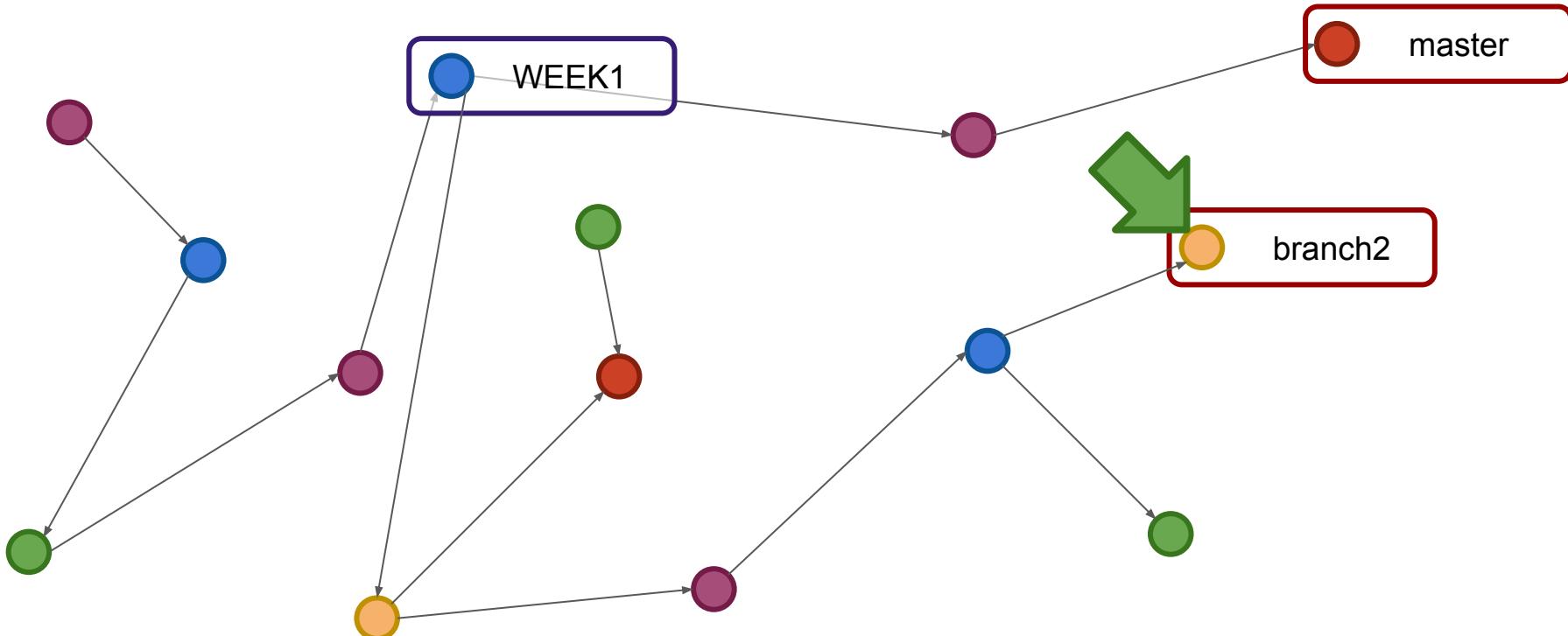
\$ > GIT CHECKOUT 84a80883D

What about “detached HEAD”?



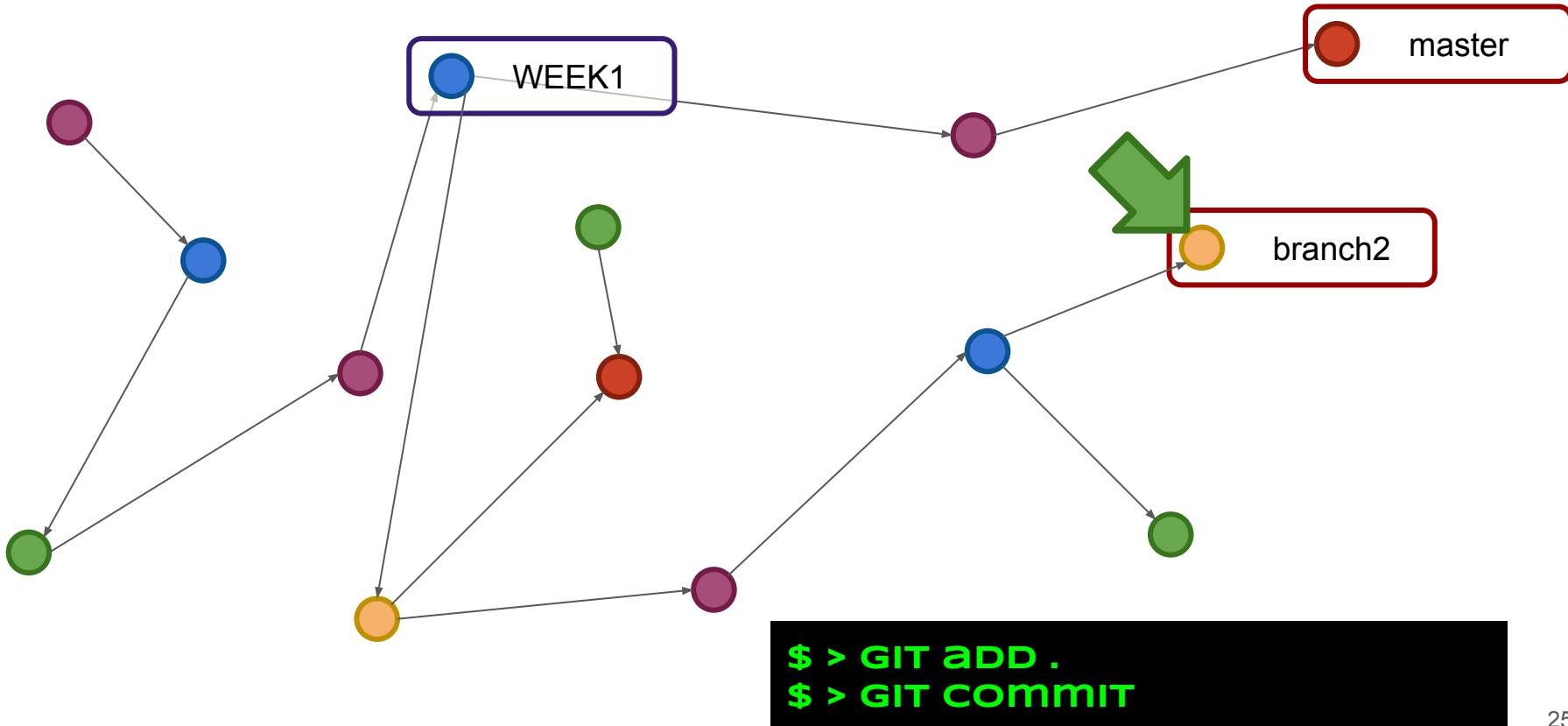
\$ > GIT CHECKOUT week1

Updating a branch

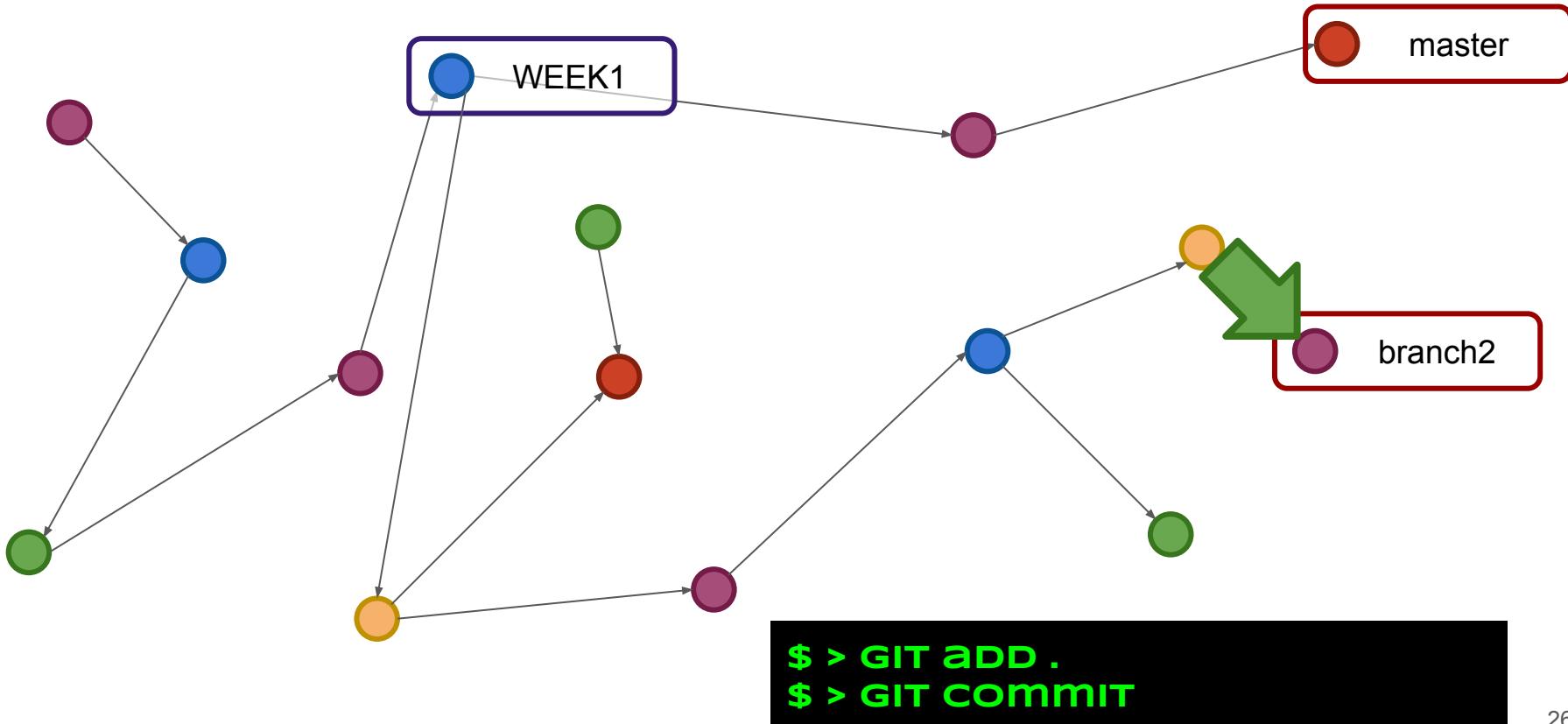


\$ > GIT CHECKOUT BRANCH2

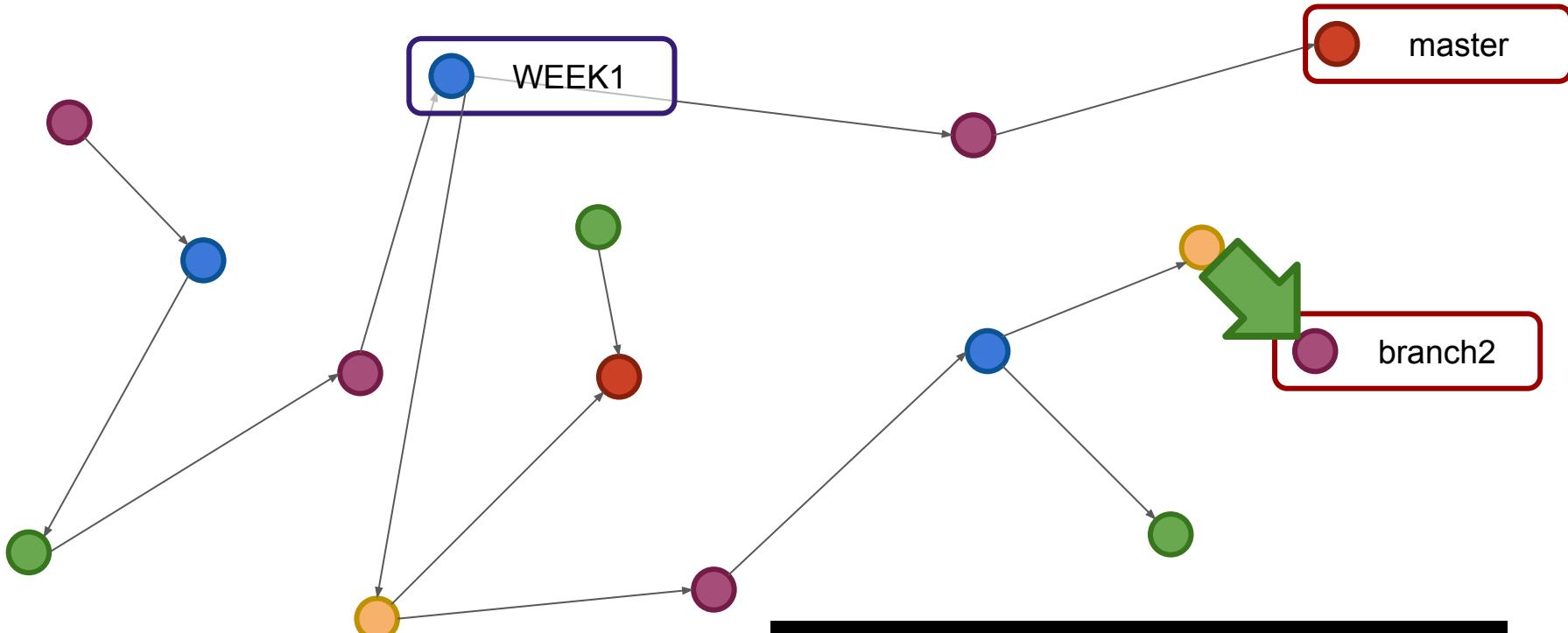
Updating a branch



Updating a branch

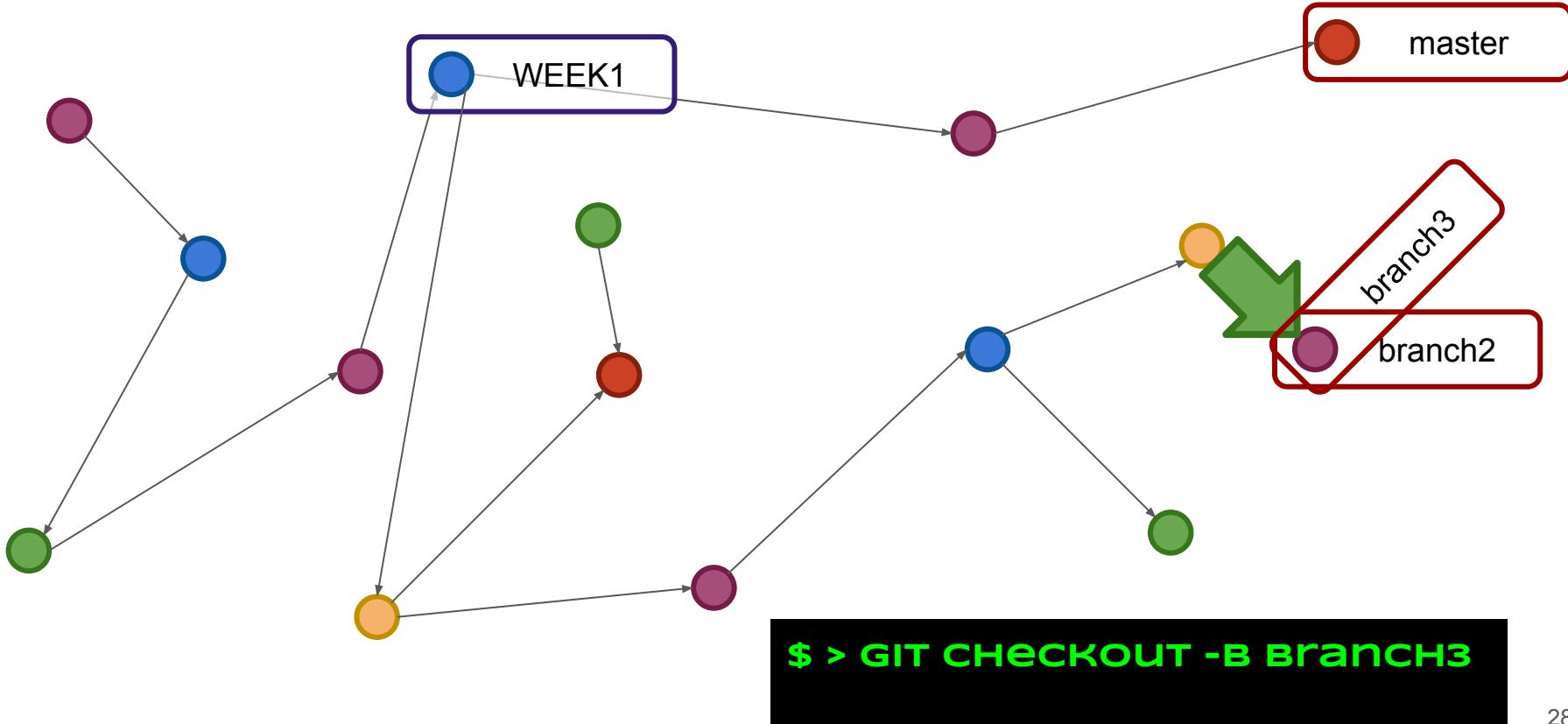


Creating a branch

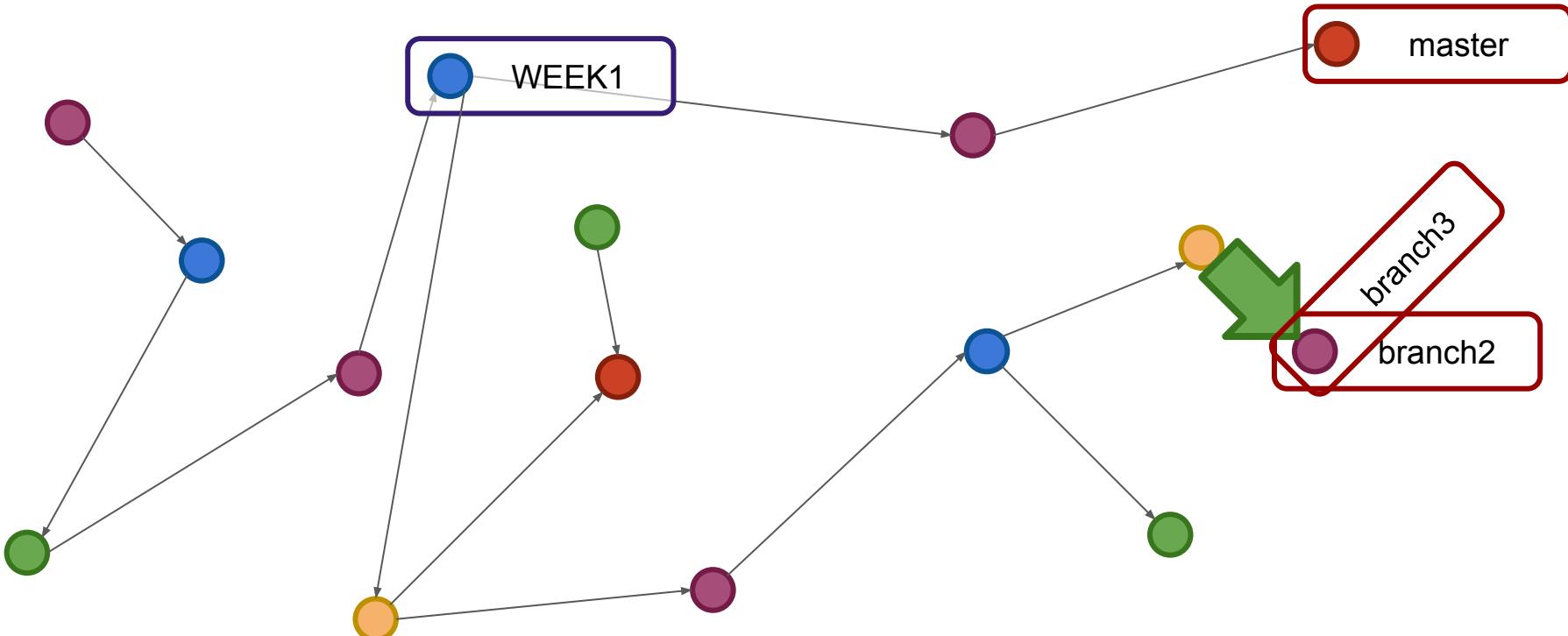


\$ > git checkout -b branch3

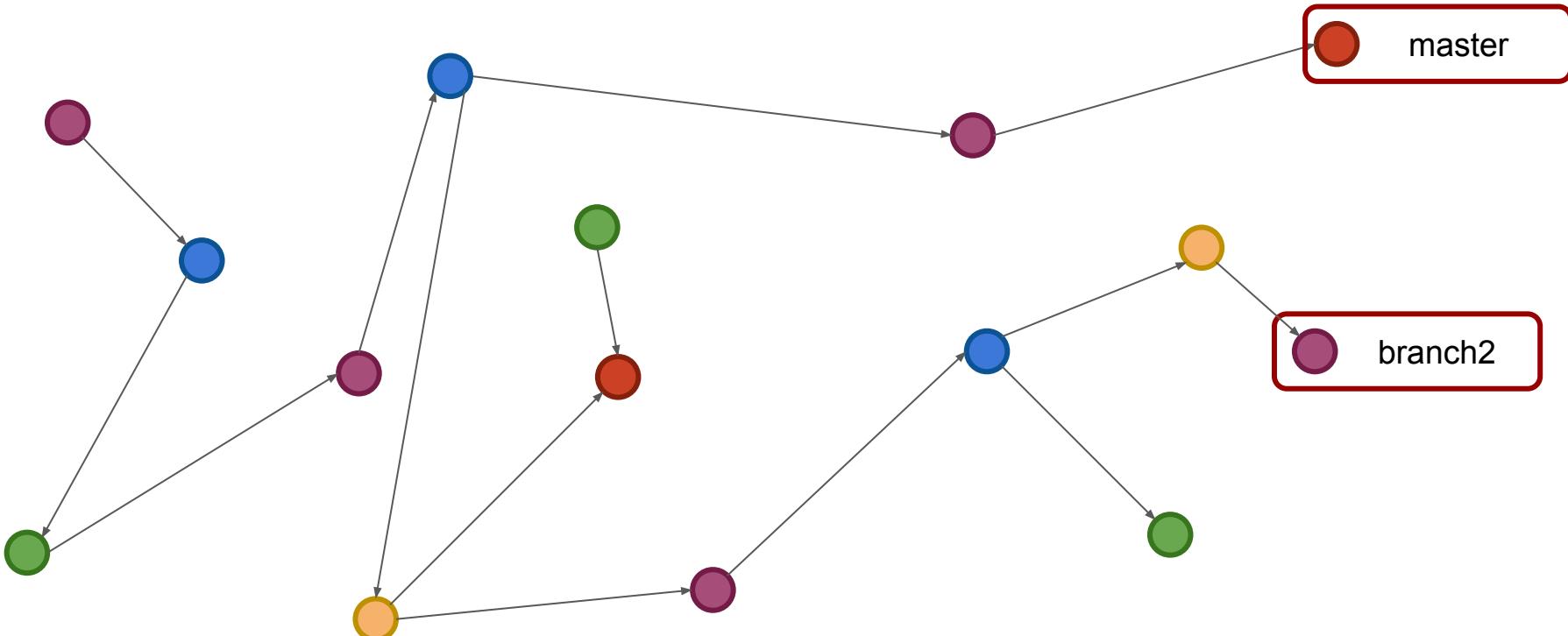
Creating a branch



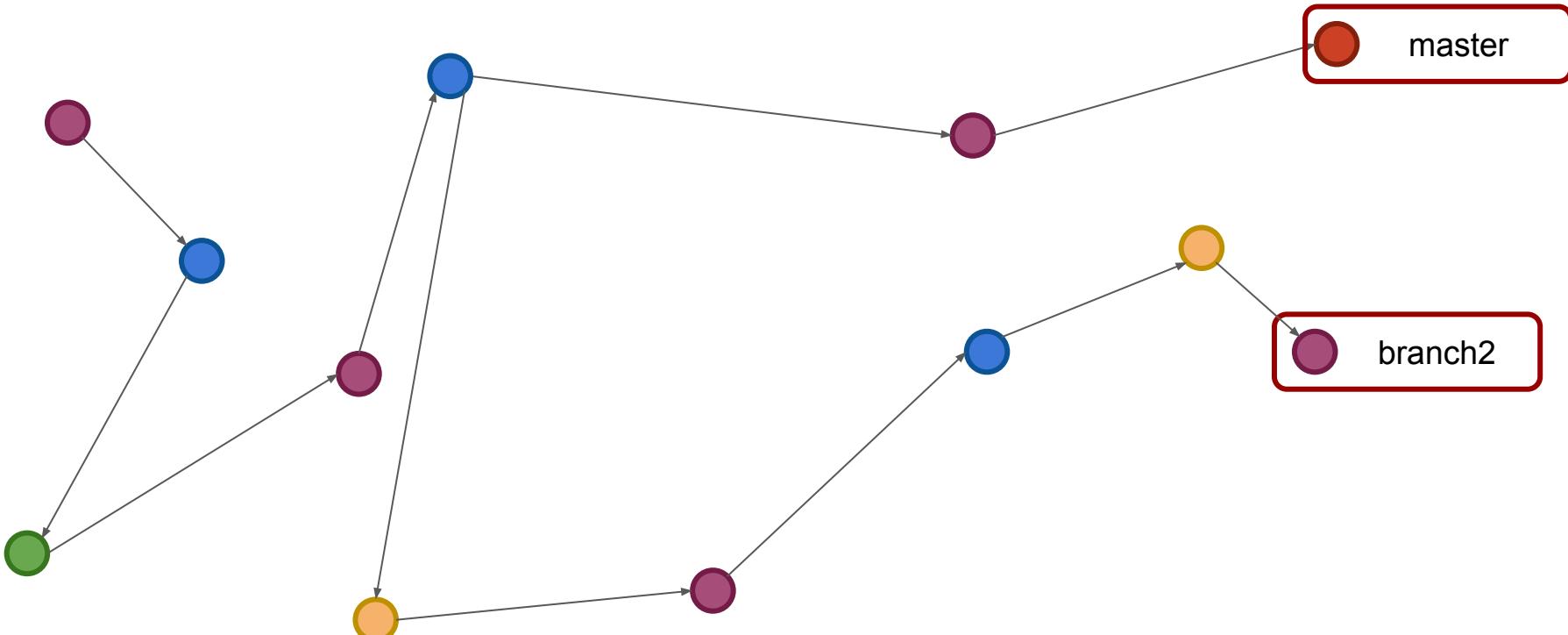
Representing a branch



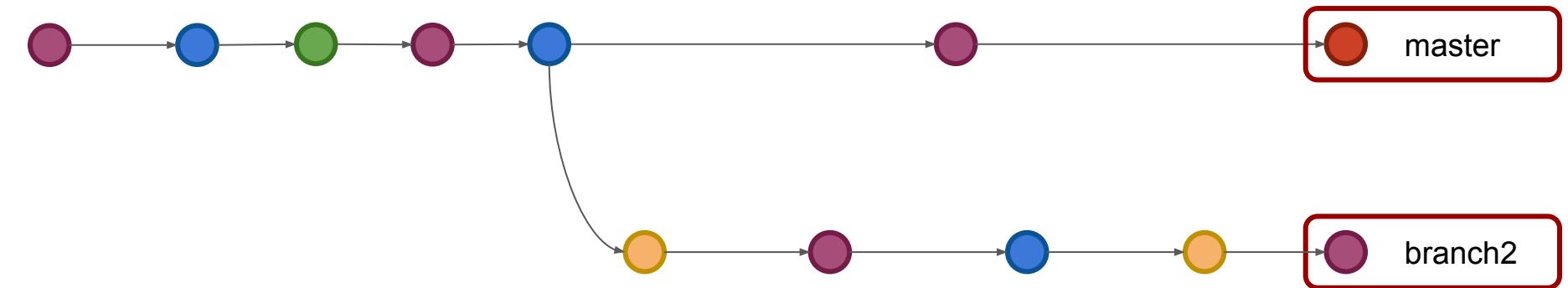
Representing a branch



Representing a branch



Representing a branch

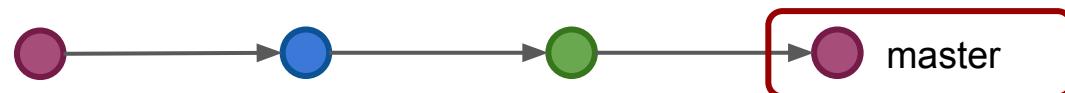


3

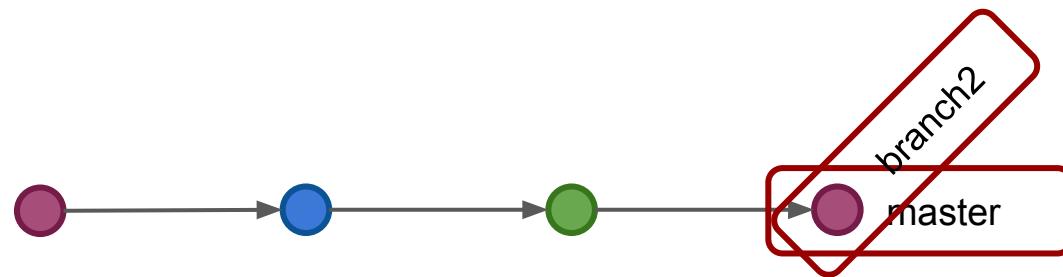


Why using branches?

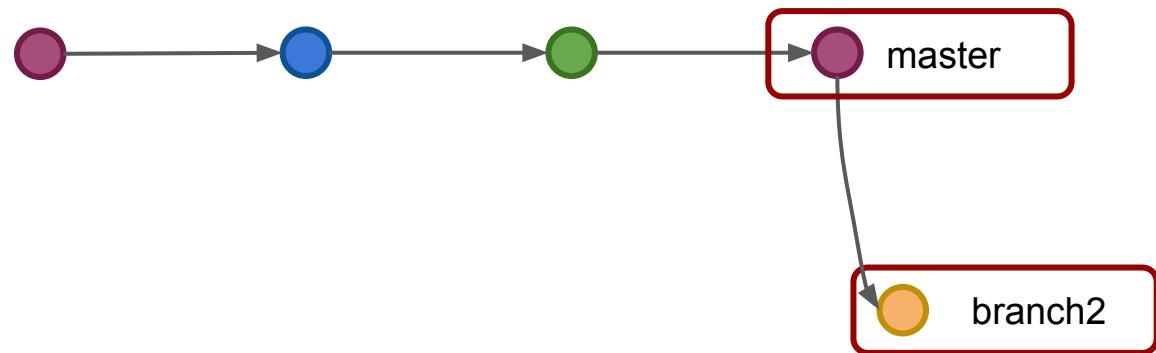
Isolating changes



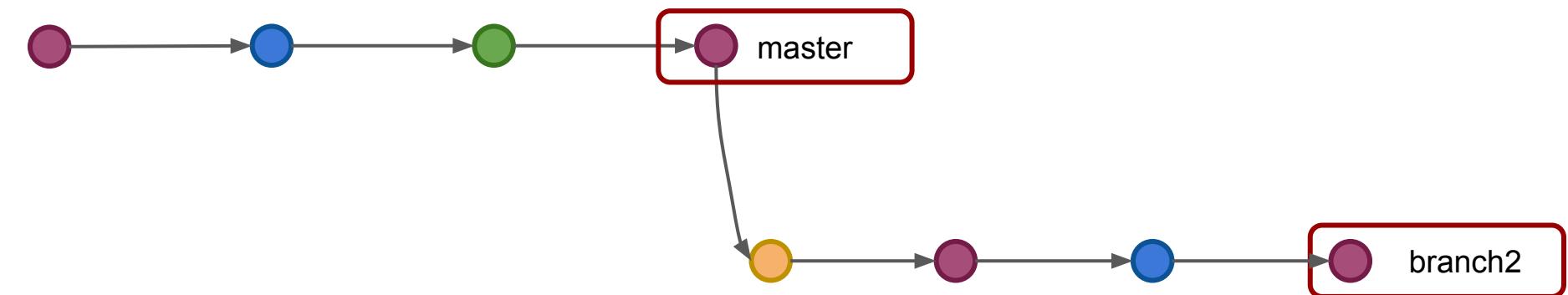
Isolating changes



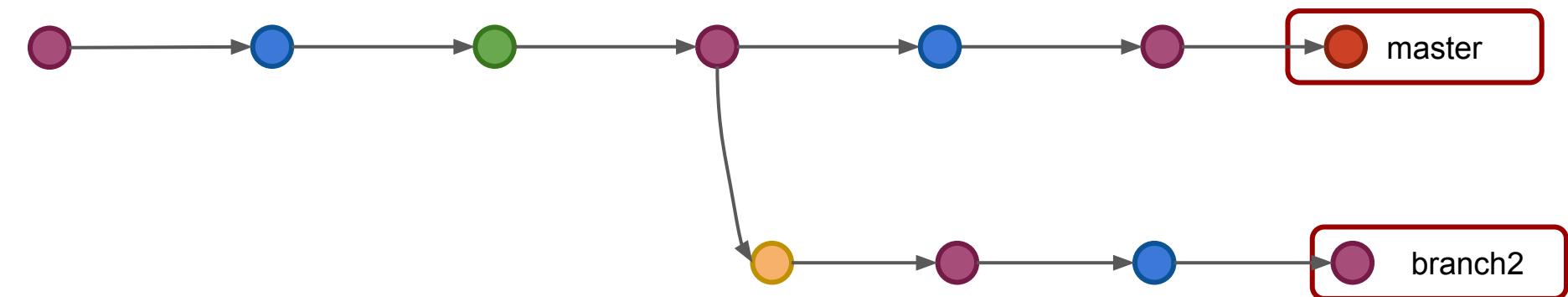
Isolating changes



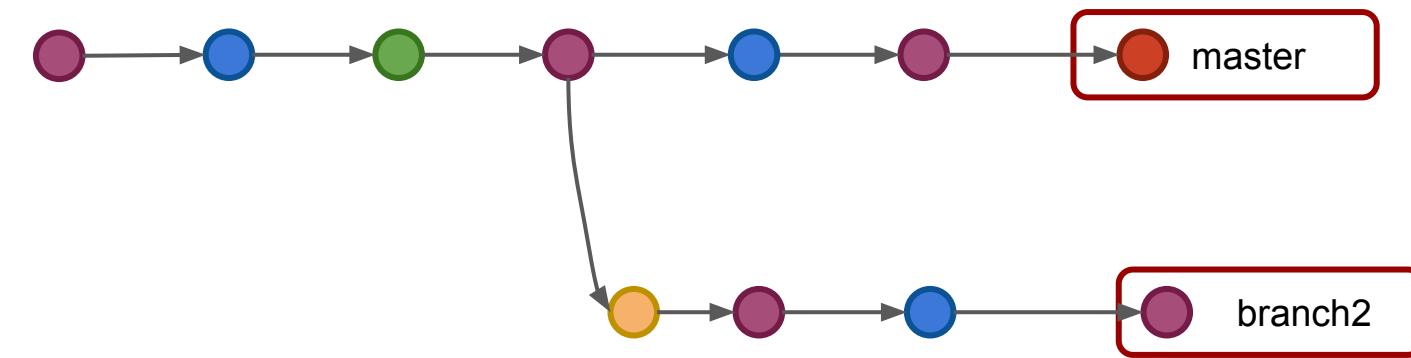
Isolating changes



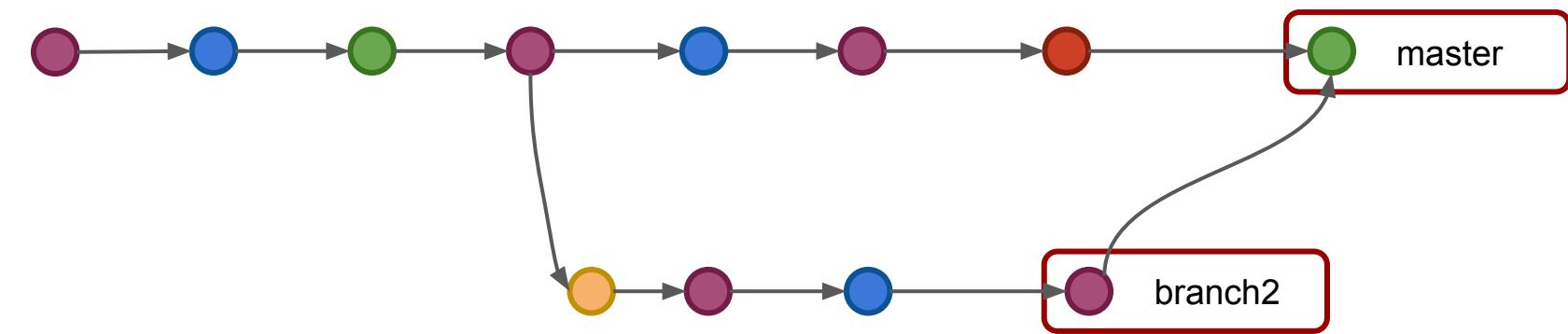
Isolating changes



Isolating changes



Isolating changes

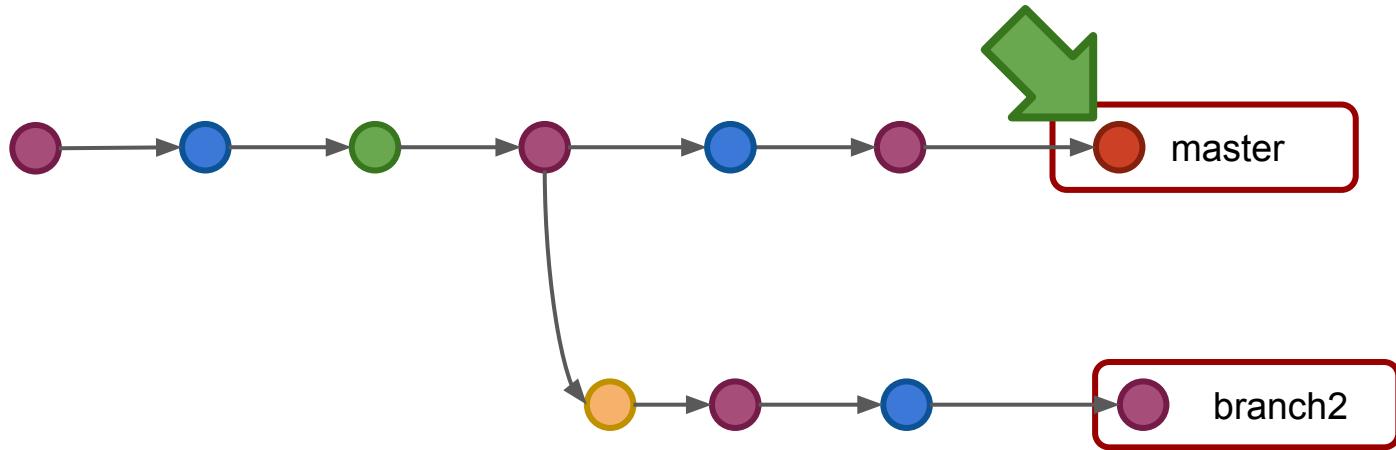


4

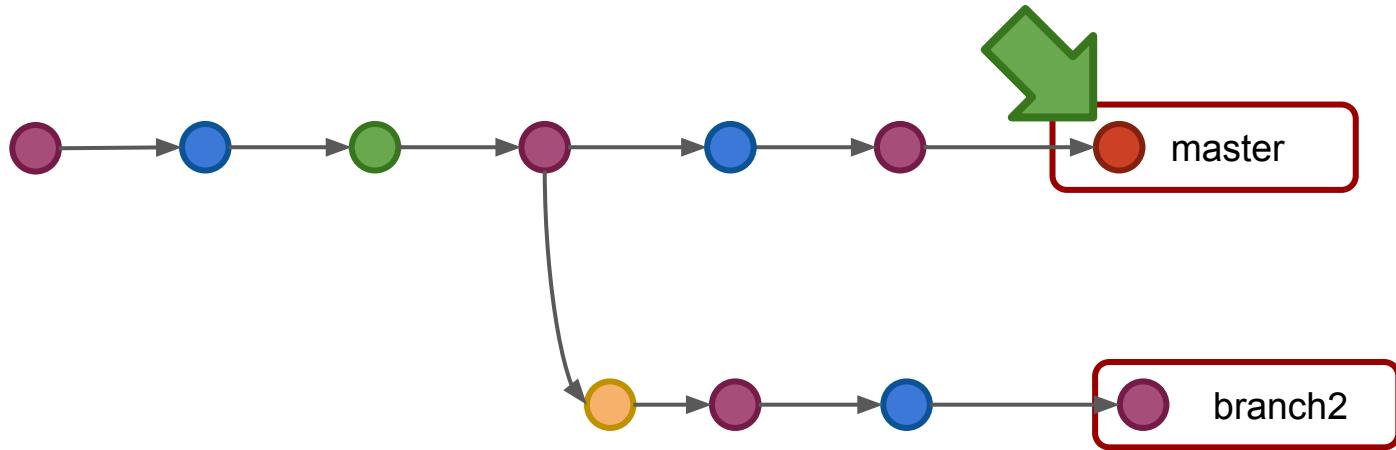


Best to know commands

Merge

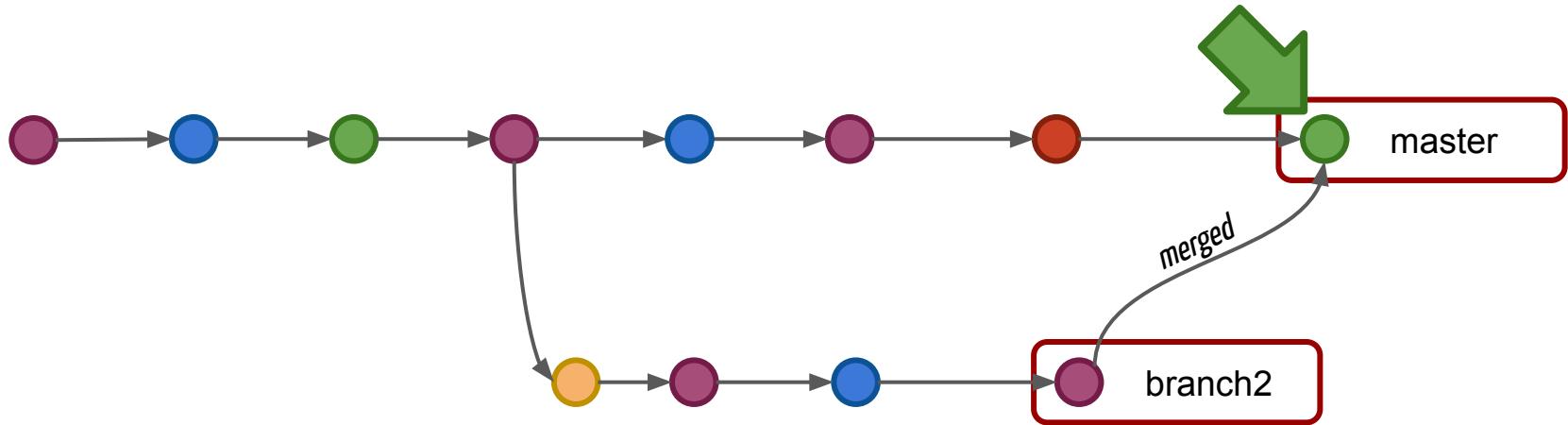


Merge

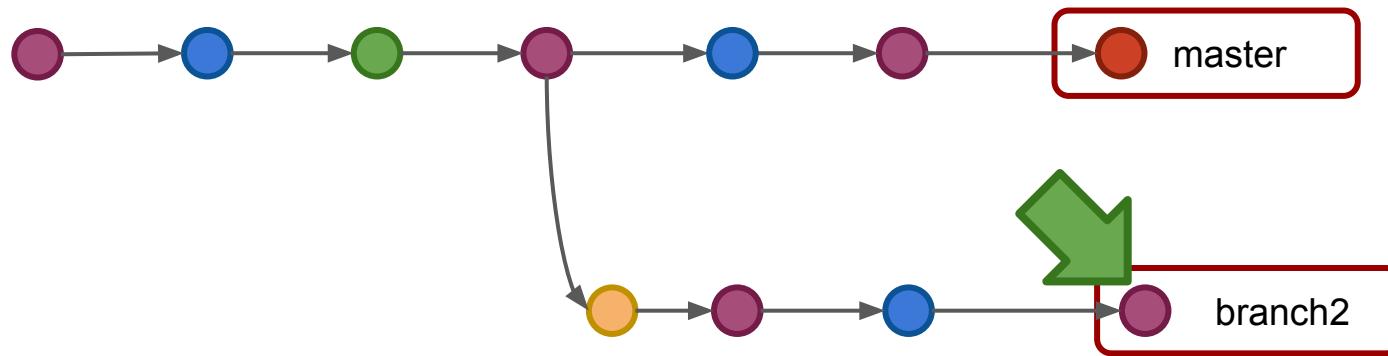


```
$ > GIT merge branch2
```

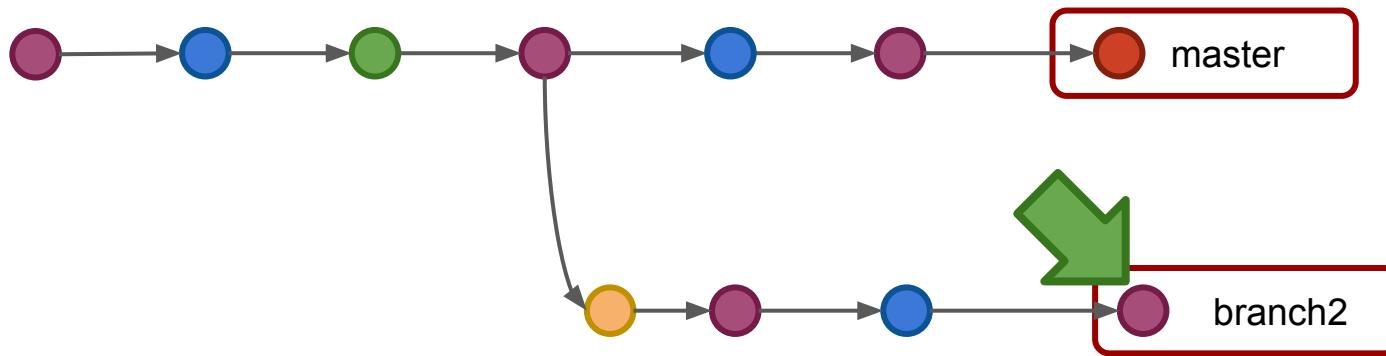
Merge



Rebase

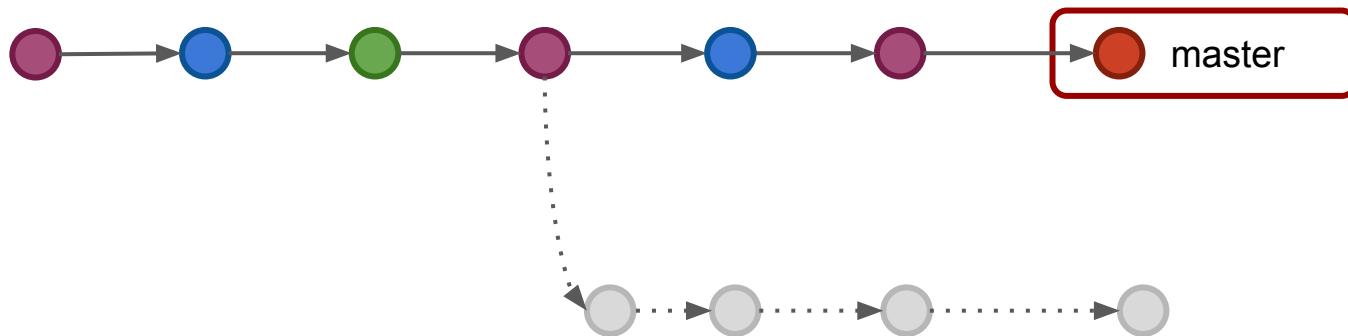


Rebase



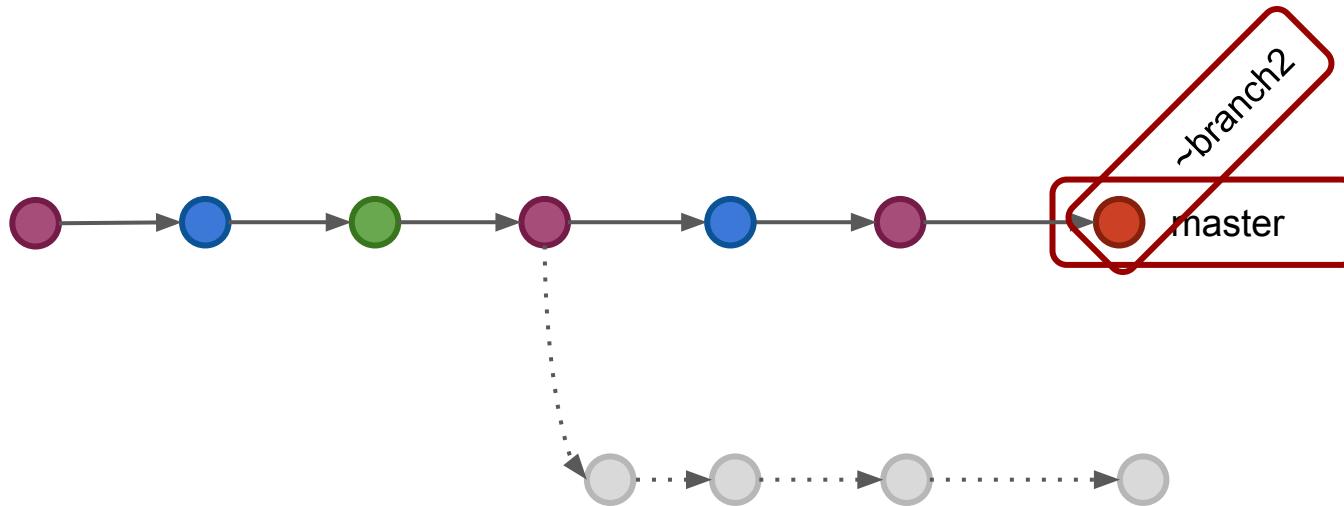
```
$ > GIT rebase master
```

Rebase



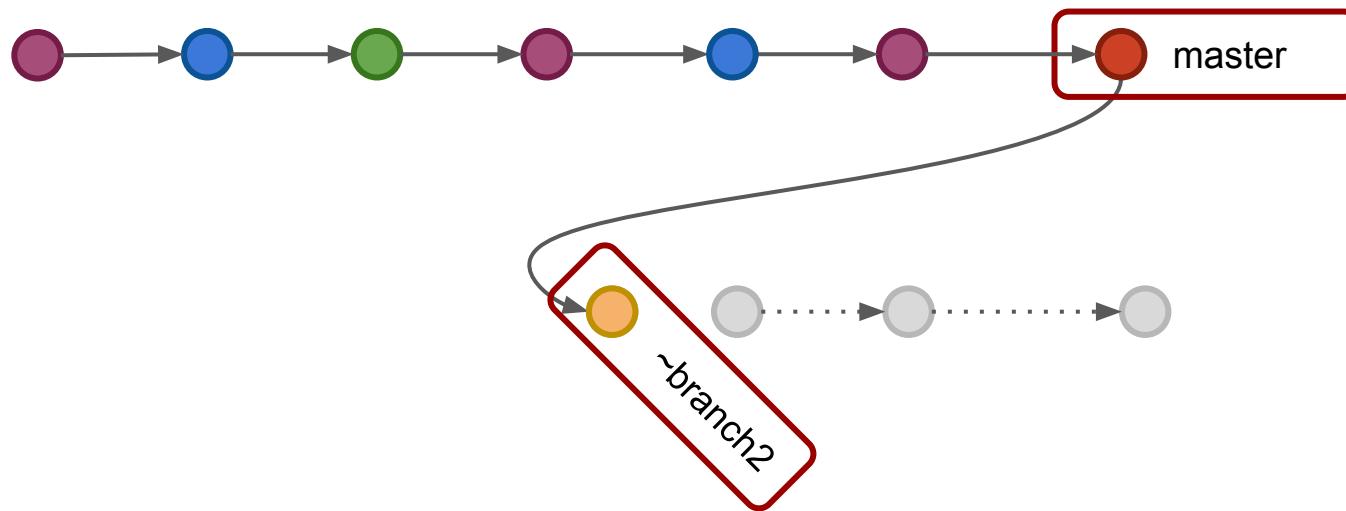
```
$ > GIT rebase master
```

Rebase



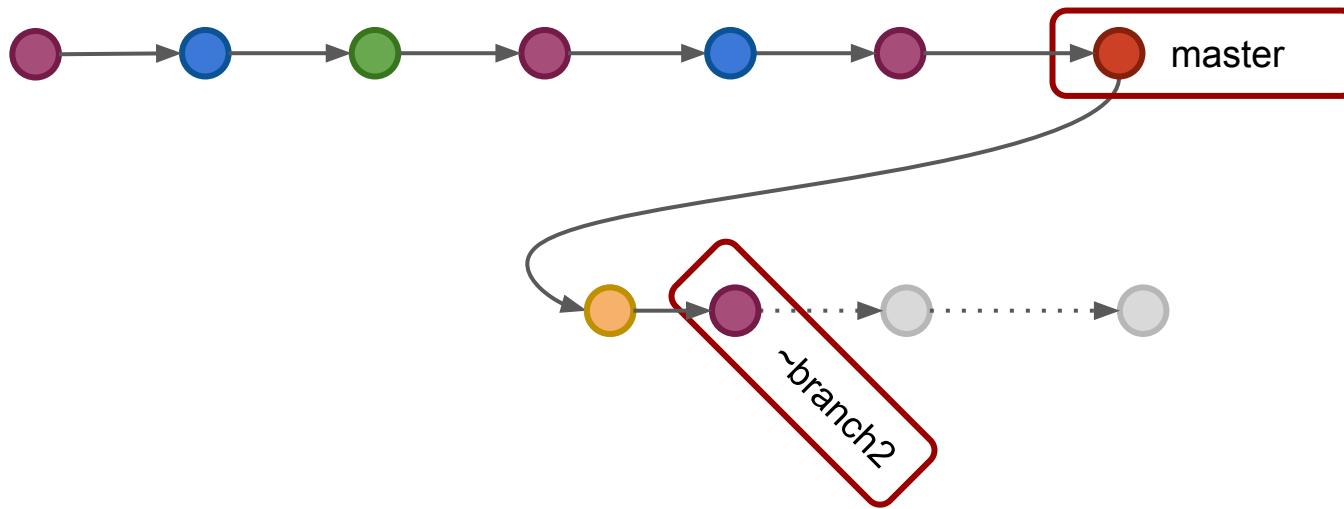
```
$ > GIT rebase master
```

Rebase



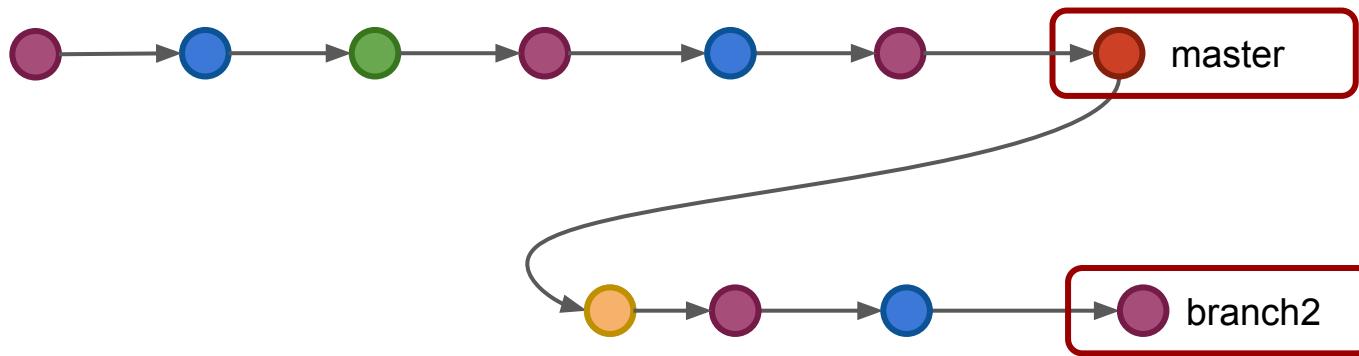
```
$ > GIT rebase master
```

Rebase



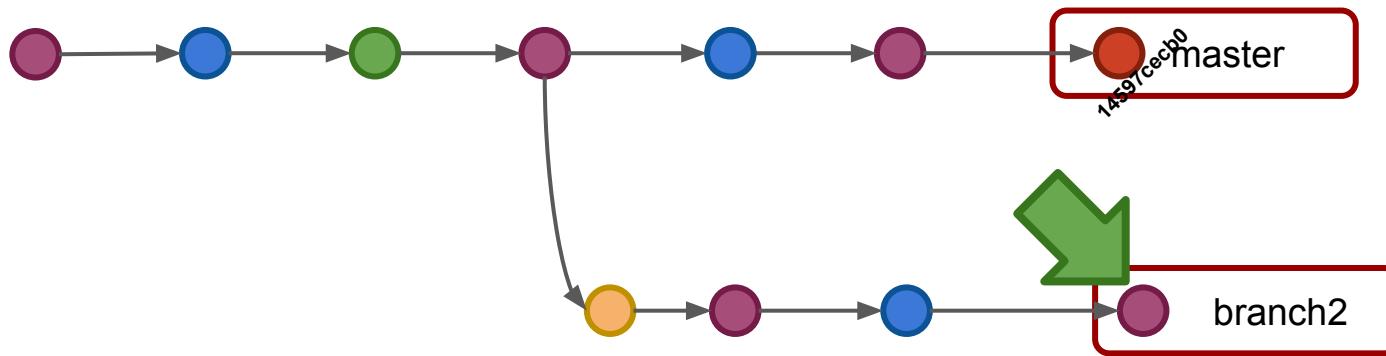
```
$ > GIT rebase master
```

Rebase

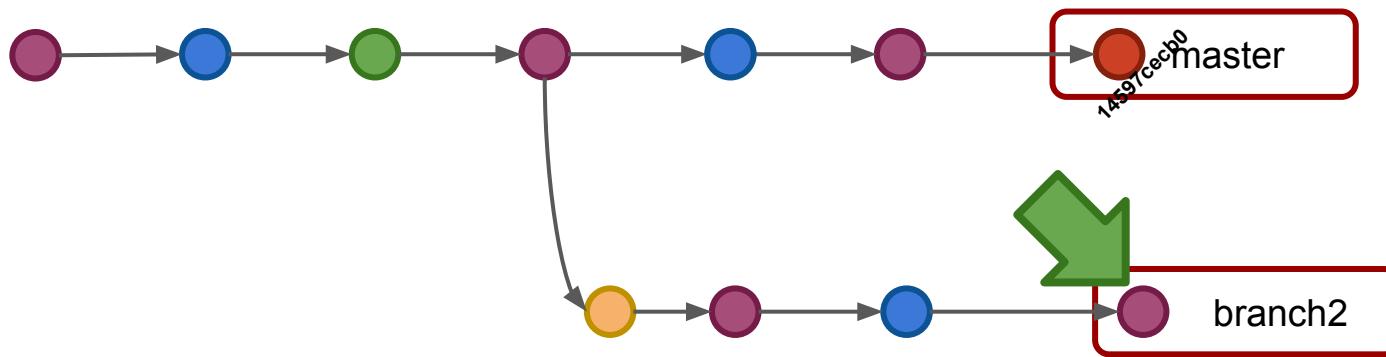


```
$ > GIT rebase master
```

Cherry-pick

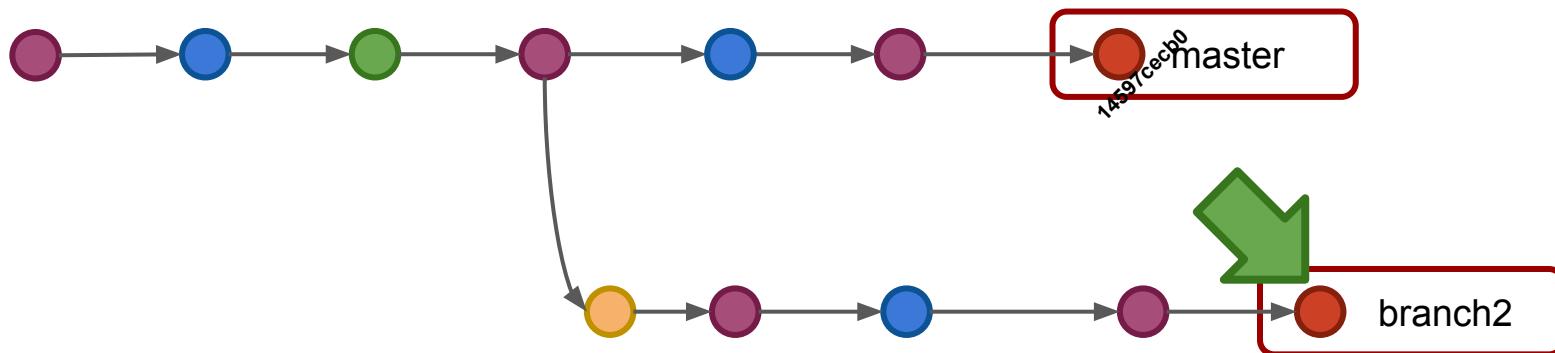


Cherry-pick



```
$ > GIT CHERRY-PICK 14597cecbo
```

Cherry-pick



```
$ > GIT CHERRY-PICK 14597cecbo
```

5



Pull requests

What is a pull request?

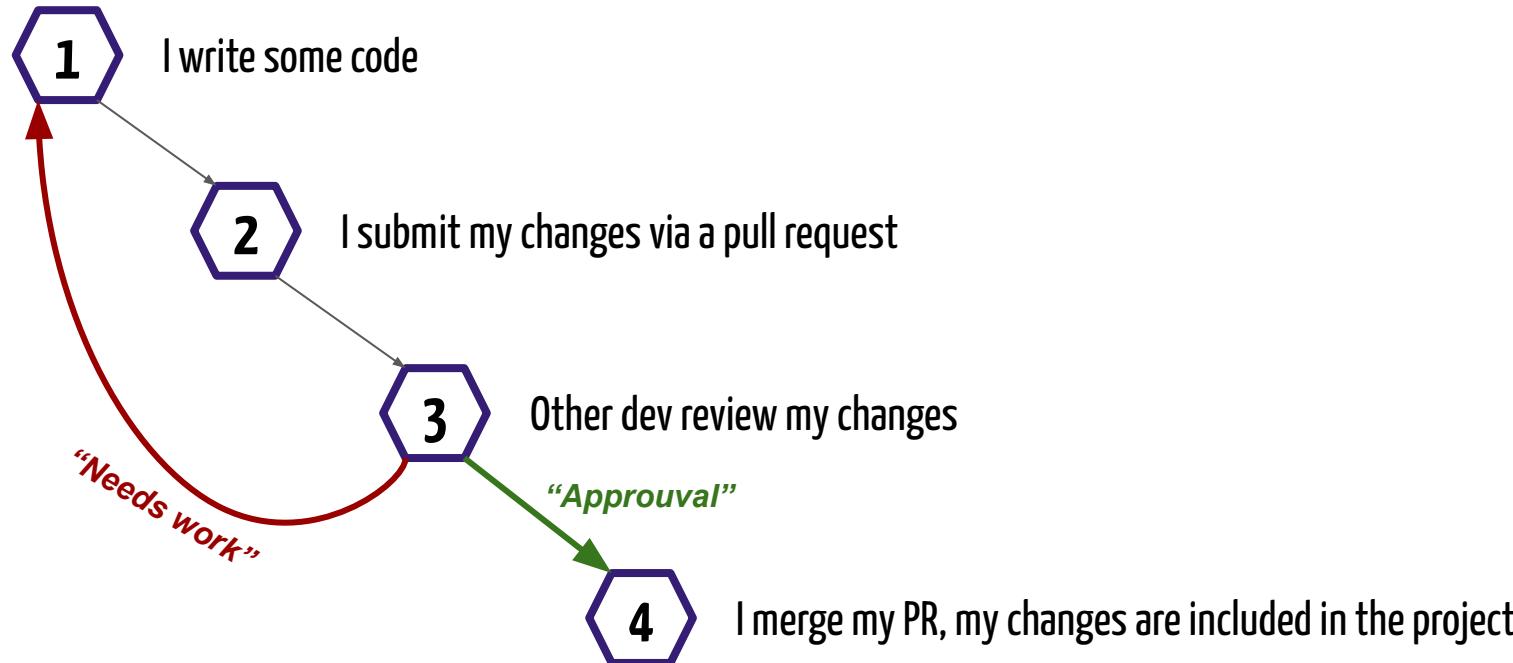
“A **pull request** (PR) is a method of submitting contributions to an *open* development project. It occurs when a developer asks for changes committed to an *external repository* to be considered for inclusion in a project's *main repository* after the peer review.”

- kenorb, on StackExchanges

What is a pull request for us?

A **pull request** (PR) is a method of submitting contributions to an development project. It occurs when a developer asks for changes committed on *a branch* to be considered for inclusion in a project's *other branch* after the peer review.

Why using pull requests?



How to create a pull request?

mathiascouste / qgl-1920

Unwatch 1 Star 2 Fork 9

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

33 commits 4 branches 0 packages 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

 mathiascouste	Update README.md	Latest commit 3bb935a 13 days ago
 championship	Update README.md	13 days ago
 lessons	lesson 2 on metrics and sonar	13 days ago
 project	Hints for week2	13 days ago
 README.md	Update README.md	last month

How to create a pull request?

mathiascouste / qgl-1920

Unwatch 1 Star 2 Fork 9

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

33 commits 4 branches 0 packages 0 releases 3 contributors

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

 mathiascouste	Update README.md	Latest commit 3bb935a 13 days ago
 championship	Update README.md	13 days ago
 lessons	lesson 2 on metrics and sonar	13 days ago
 project	Hints for week2	13 days ago
 README.md	Update README.md	last month

How to create a pull request?

The screenshot shows a GitHub repository page for `mathiascouste / qgl-1920`. The top navigation bar includes options for `Code`, `Issues 0`, `Pull requests 0`, `Actions`, `Projects 0`, `Wiki`, `Security`, `Insights`, and `Settings`. The repository has 1 unwatched star, 2 forks, and 9 forks.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: `master` compare: `week2_preview` ✓ Able to merge. These branches can be automatically merged.

Week2 results

Write Preview AA B i “ < > @ [] ↵

Let them know...

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

How to create a pull request?

[mathiascouste / qgl-1920](#)

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master ▾ compare: week2_preview ▾

✓ Able to merge. These branches can be automatically merged.

Week2 results

Write Preview

Let them know...

Attach files by dragging & dropping, selecting or pasting them.

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Create pull request ▾

[Code](#)[Issues 0](#)[Pull requests 1](#)[Actions](#)[Projects 0](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

Week2 results #3

[Edit](#)[Open](#)mathiascouste wants to merge 1 commit into [master](#) from [week2_preview](#)[Conversation 0](#)[Commits 1](#)[Checks 0](#)[Files changed 38](#)[+538,414 -0](#)

mathiascouste commented 32 seconds ago

[Owner](#) + ...

Let them know...

[Week2 results](#)

1c27432

[mathiascouste requested a review from collet now](#)[mathiascouste added the wontfix label now](#)Add more commits by pushing to the [week2_preview](#) branch on [mathiascouste/qgl-1920](#).**Review requested**[Show all reviewers](#)Review has been requested on this pull request. It is not required to merge. [Learn more](#).**1 pending reviewer****This branch has no conflicts with the base branch**

Merging can be performed automatically.

[Merge pull request](#)or view [command line instructions](#).

Reviewers

collet



Assignees

No one—assign yourself



Labels

wontfix



Projects

None yet



Milestone

No milestone



Linked issues

Successfully merging this pull request may close these issues.

None yet



Notifications

[Customize](#)[Unsubscribe](#)

You're receiving notifications because

Q&A



Branching strategies

Mathias COUSTÉ
06.02.2020

1

You said branching strategy?

the decision
was or
one of
played
t risks.
ut into

risky,

[Latin: related to dictator]
dictatorial /diktə'tɔrɪkl/
like a dictator. 2 overbearing.
orily adv. [Latin: re-
TATOR]

diction /'dɪkʃ(ə)n/ n. man-
ciation in speaking or si-
dictio from *dico* dict- say]
dictionary /'dɪkʃənərɪ/
book listing (usu. alpha-
explaining the words of

Branching strategies

Branching strategy



PROCESS

Branching strategies

A team agreement to
standardize the usage
of GIT tools

Branching strategies

A strategy covers many aspect of Git:

- branch naming convention
- merging strategy
- branch lifecycle
- rebase VS merge
- cherry-picking fixes
- merge strategy
- etc...

2



Common branching strategies

Git flow

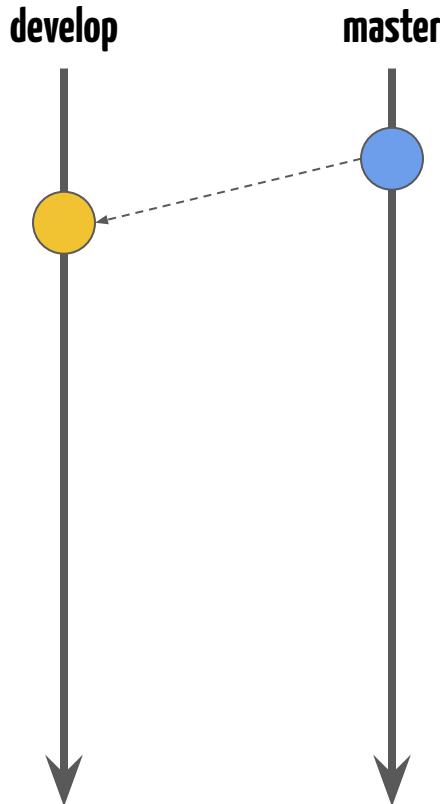
master



Master:

- infinite lifetime
- very stable
- production ready

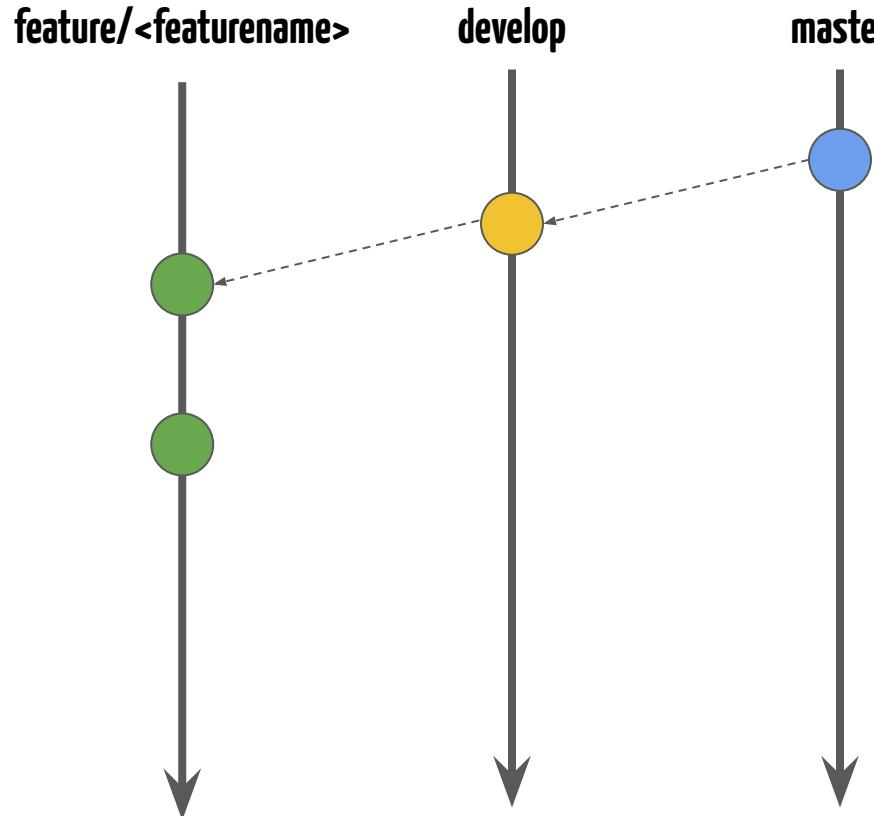
Git flow



Develop:

- infinite lifetime
- unstable

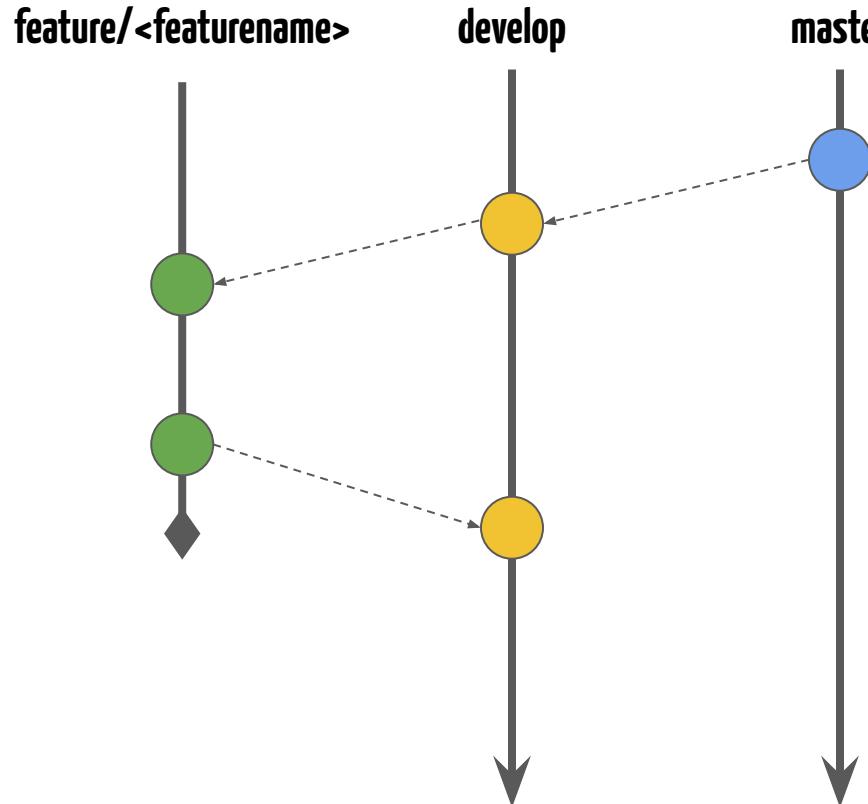
Git flow



Feature/<featurename>:

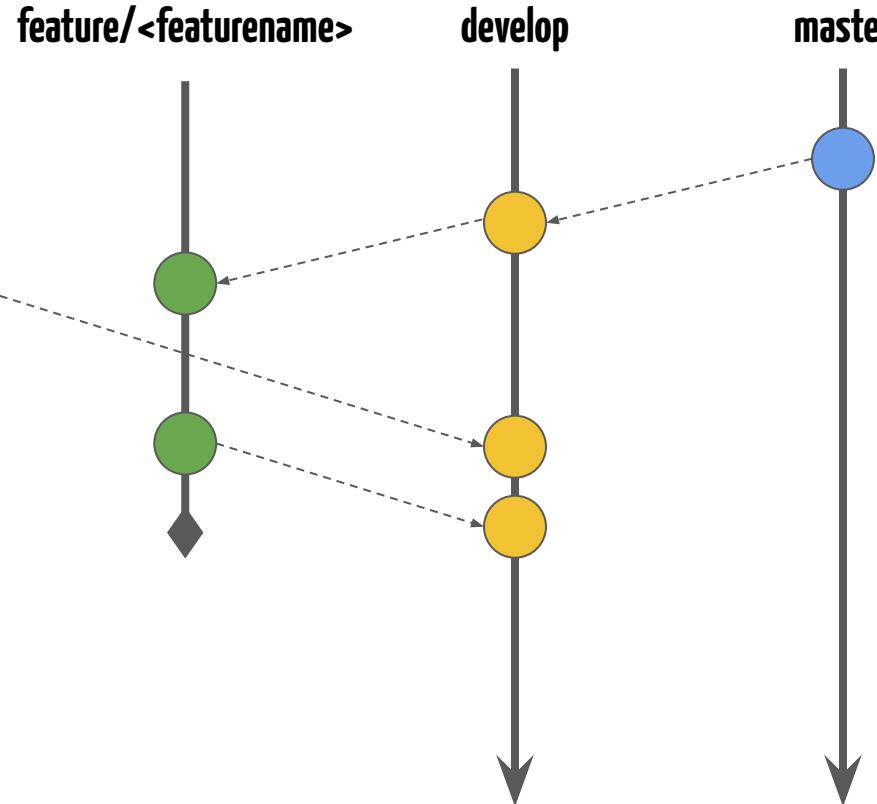
- destroyed after feature development
- unstable
- work in progress

Git flow



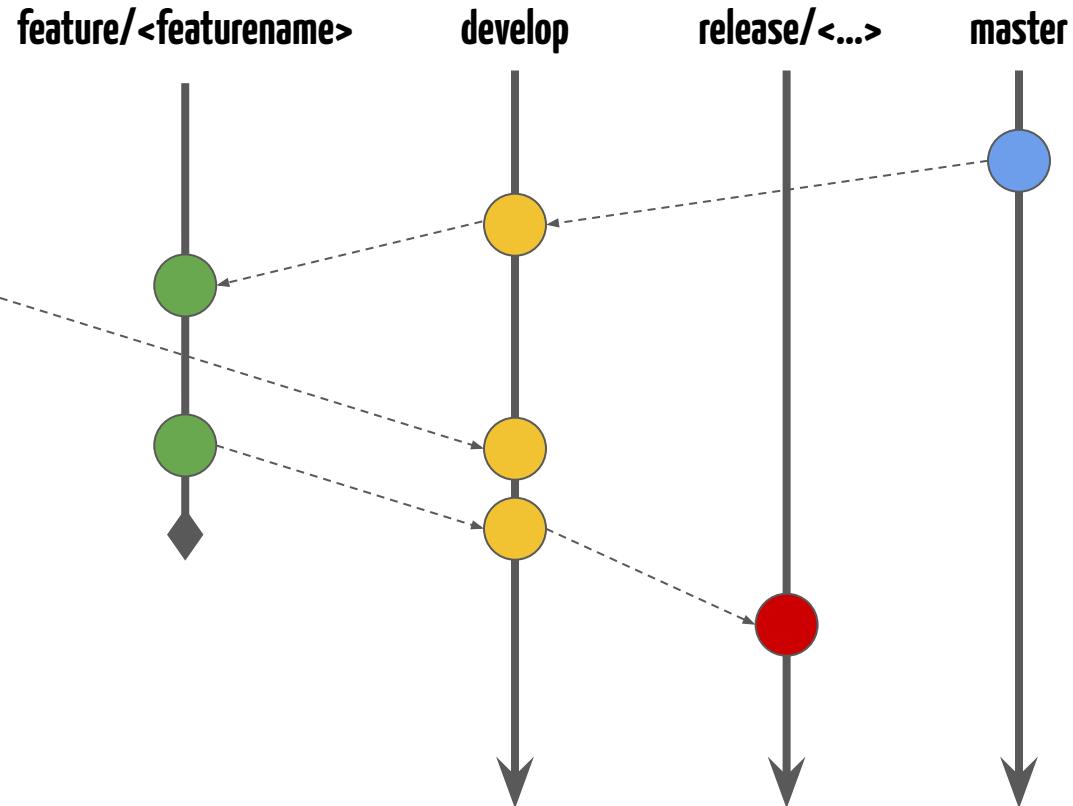
1. Feature is merged into develop
2. feature / <featurename> can be destroyed

Git flow



Many feature can be
merged into develop...

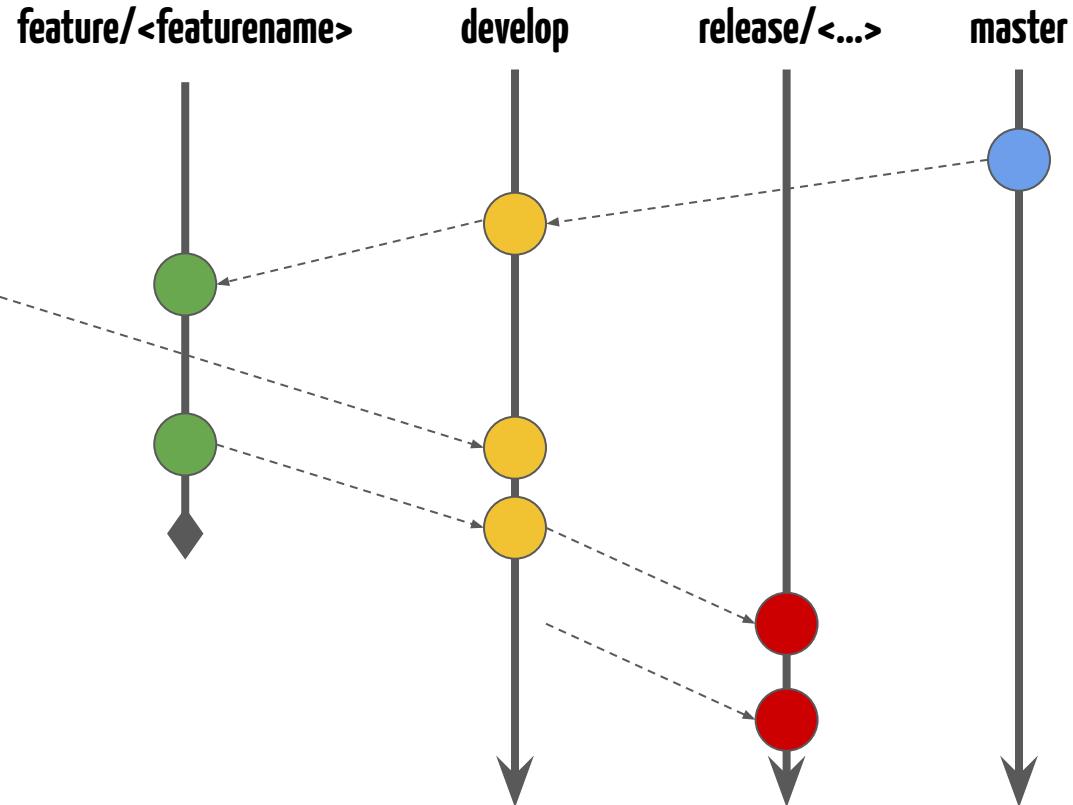
Git flow



Release / <releasename>:

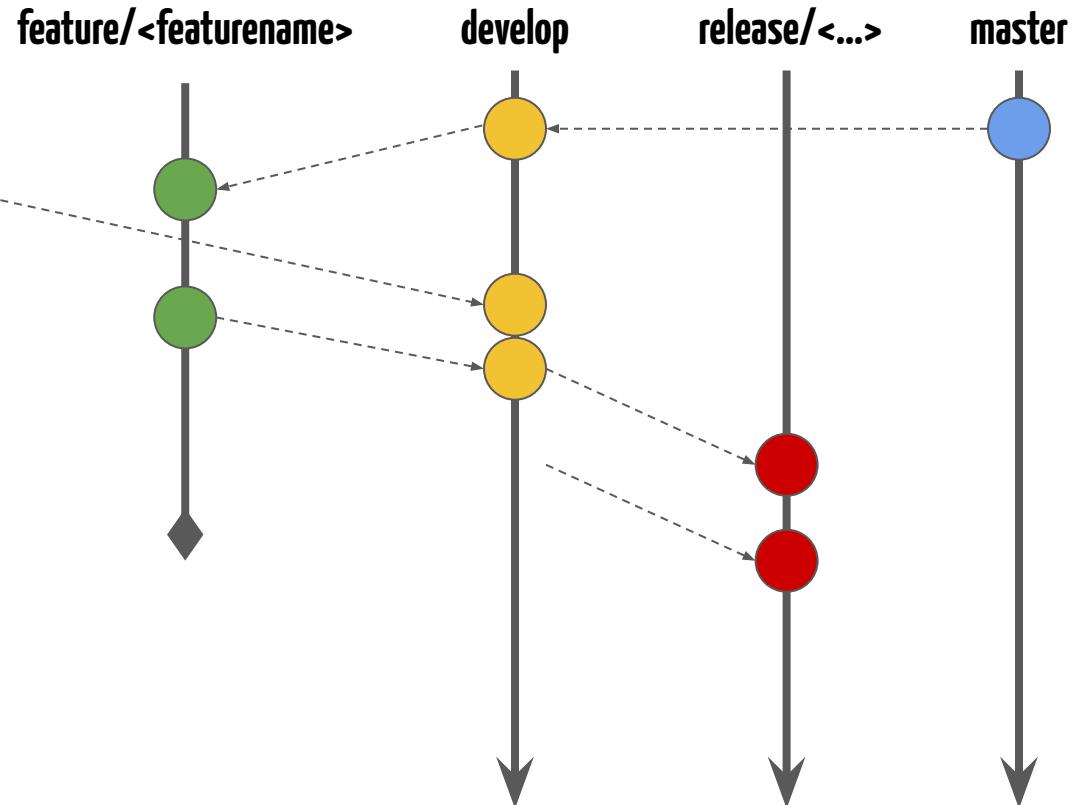
- destroyed after release is not supported
- tends to be stable

Git flow



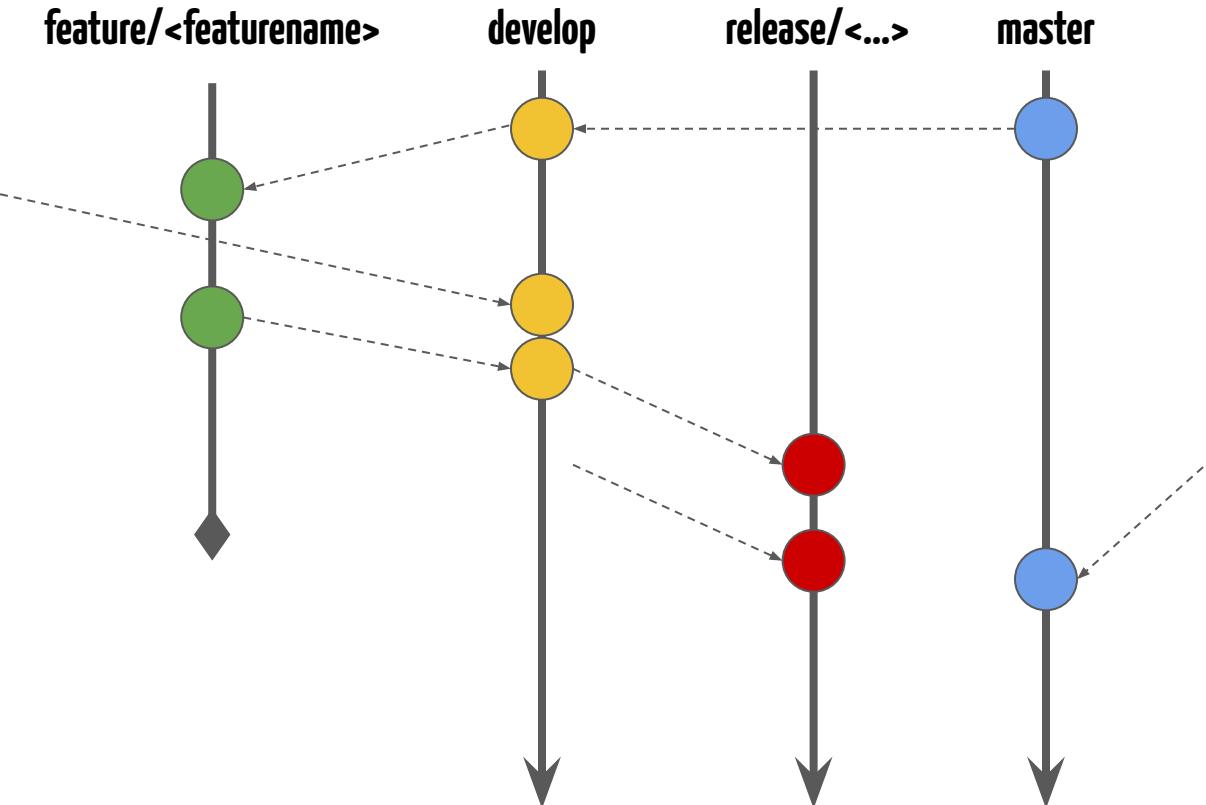
Add fixes to the release
branch to increase its
stability

Git flow



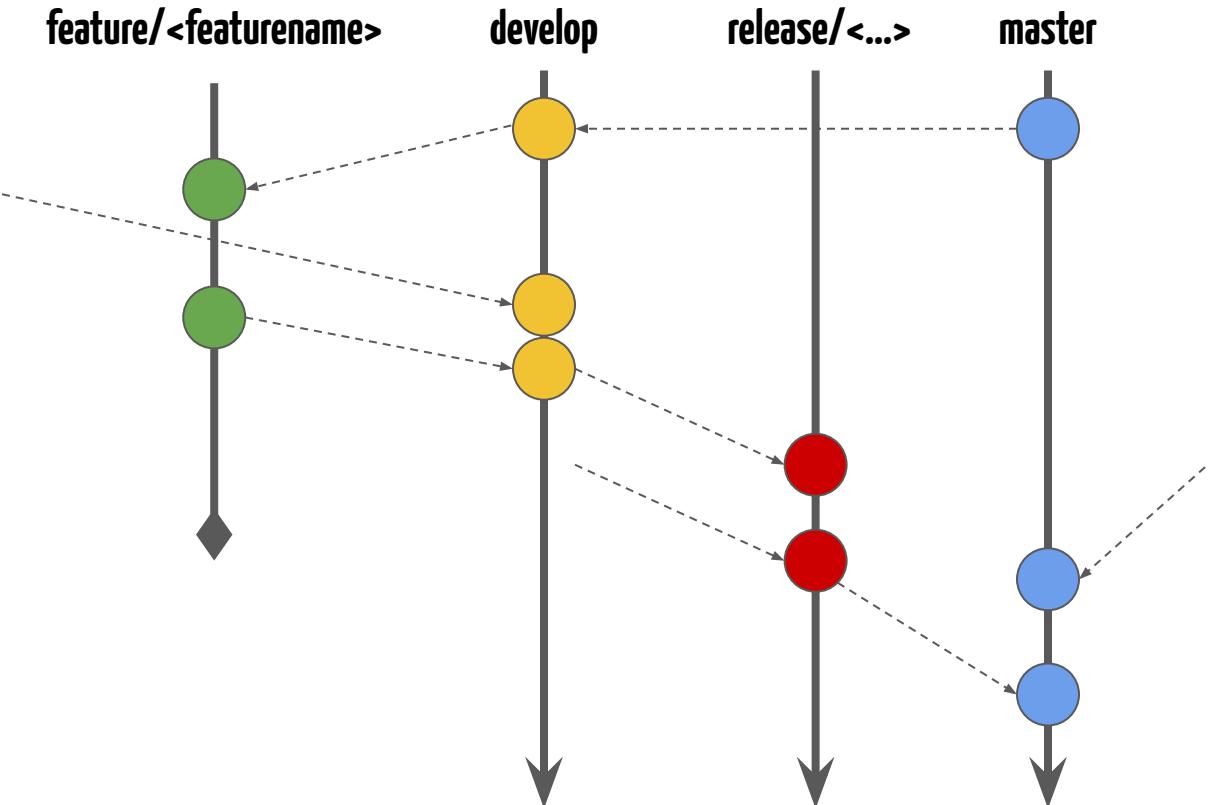
Add fixes to the release
branch to increase its
stability

Git flow



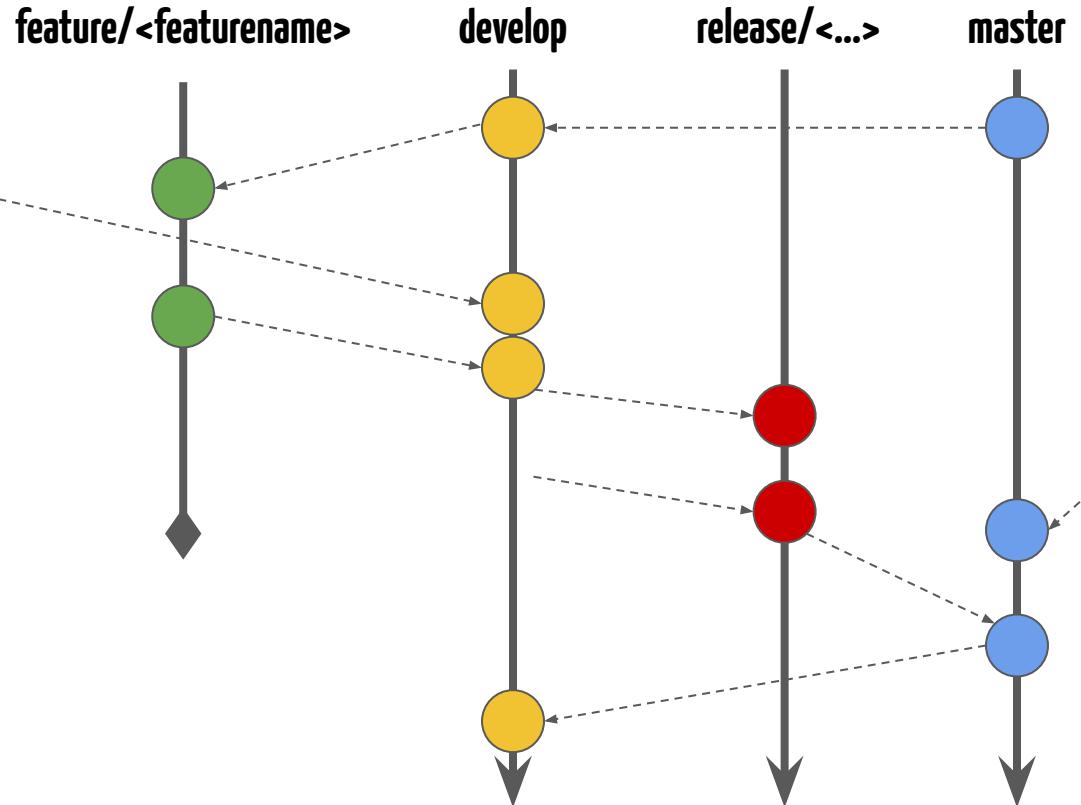
Hotfixes can be added to
the master branch

Git flow



When a release is ready
for production, we merge
it into master

Git flow



Don't forget to keep
develop up to date with
master...

Git flow

PROS

- Ensure a clean state of branches
- Branches' name follow an easy to understand systematic pattern
- Support multiple production version
- High level of composition thanks to feature branches

CONS

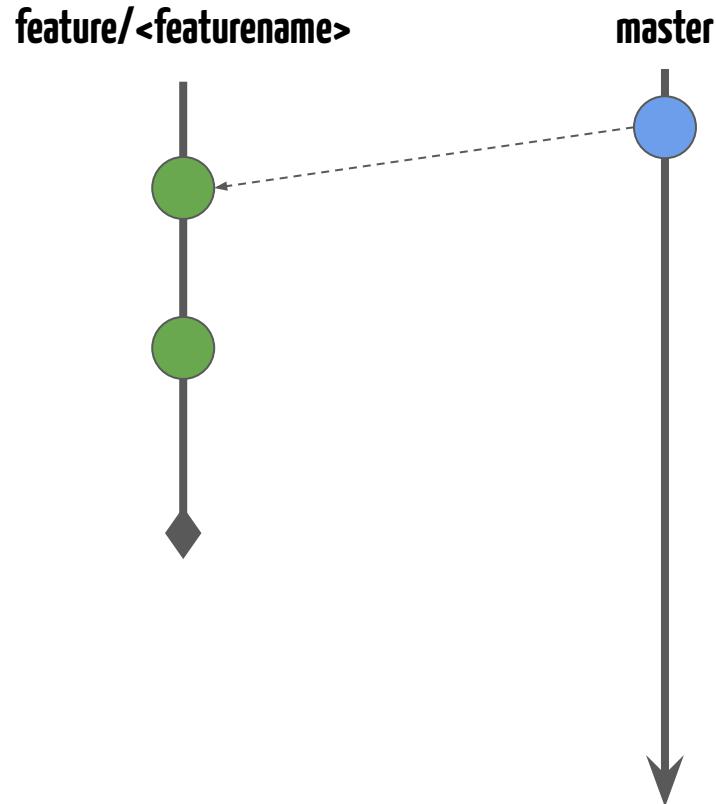
- Not recommended to support only 1 production version
- Merging old features costs a lot
- Boilerplate

Github flow

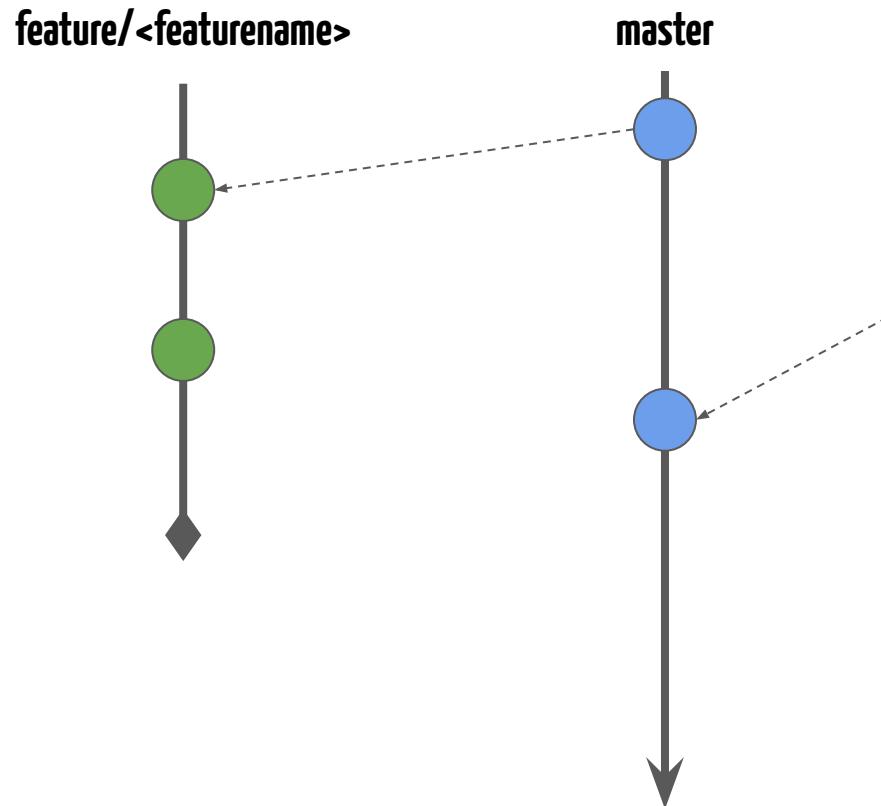
master



Github flow



Github flow



Github flow

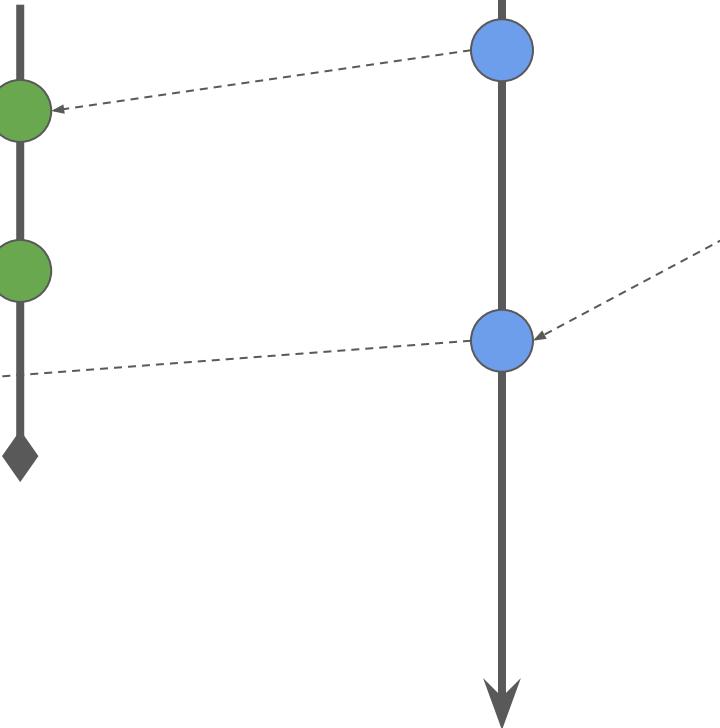
feature/feature2



feature/<featurename>



master



Github flow

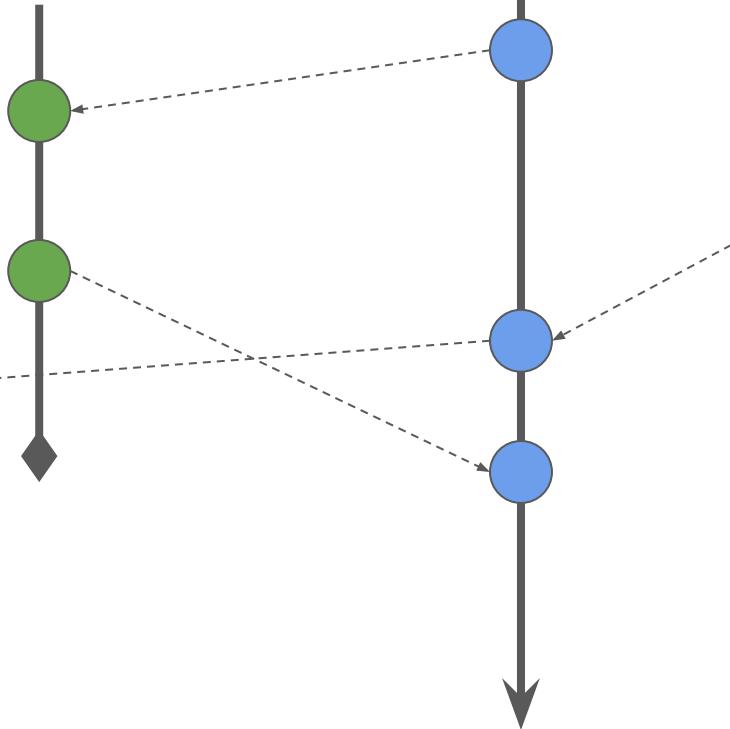
feature/feature2



feature/<featurename>



master



Github flow

feature/feature2



feature/<featurename>



master



Github flow

feature/feature2



feature/<featurename>



master



Github flow

PROS

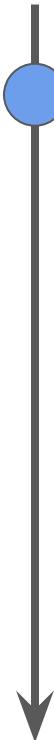
- Continuous delivery friendly
- Simpler than git flow
- Ideal for single production version

CONS

- master can be unstable
- Hard to follow a release plan
- May need code freeze on master

Gitlab flow

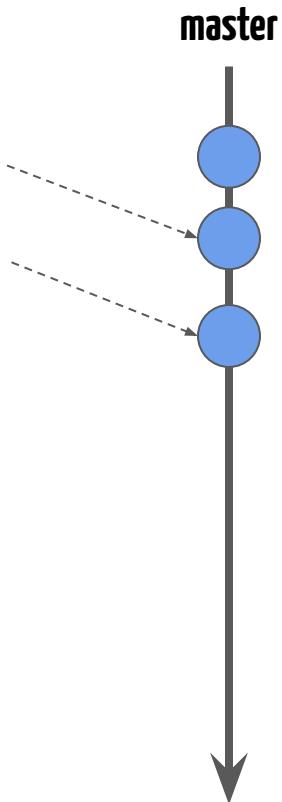
master



Master:

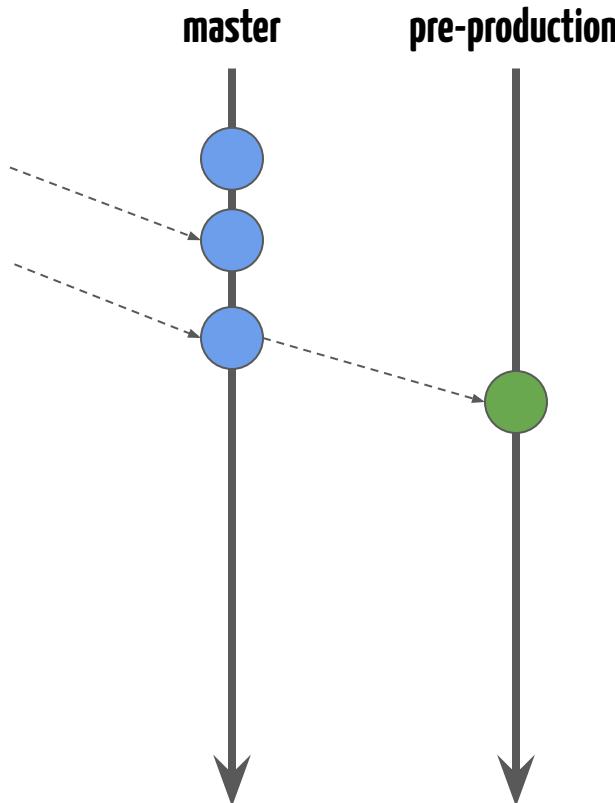
- infinite lifetime
- unstable

Gitlab flow



Add new code as you
want...

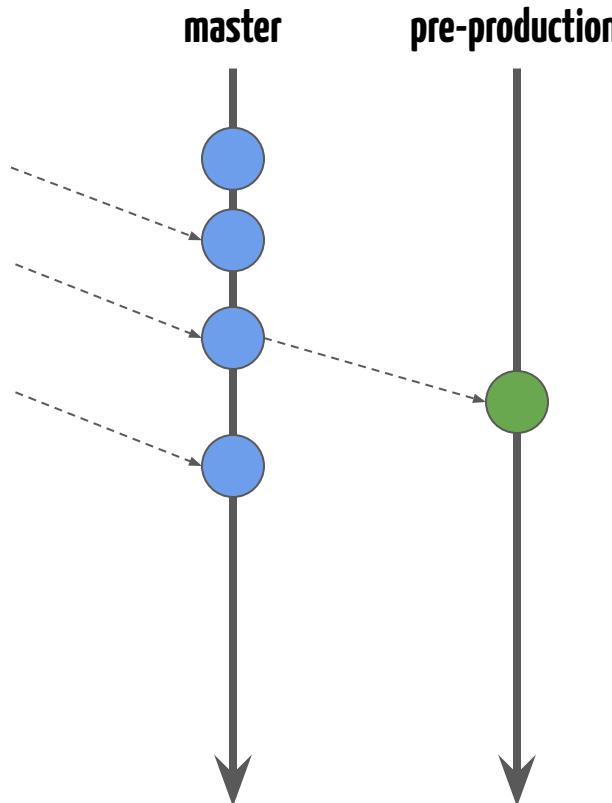
Gitlab flow



Pre-production:

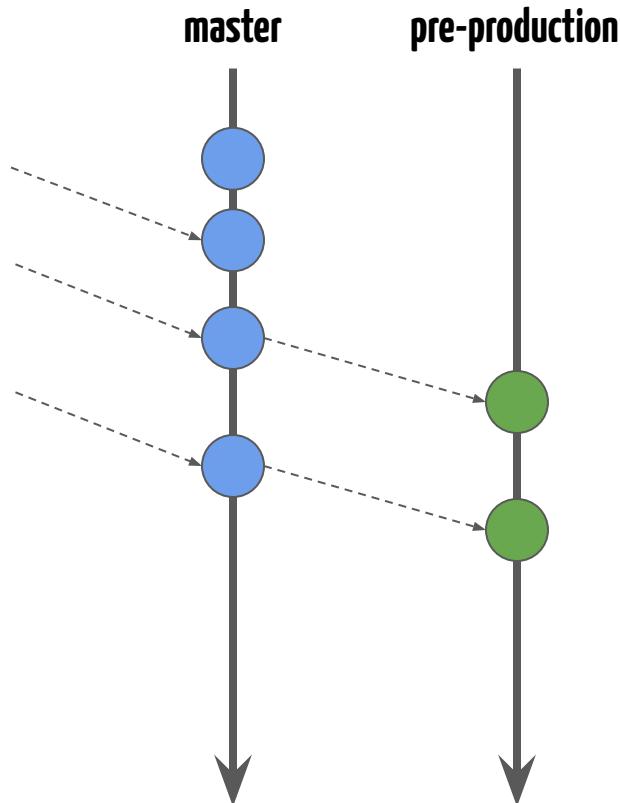
- infinite lifetime
- stable
- pre-production ready

Gitlab flow



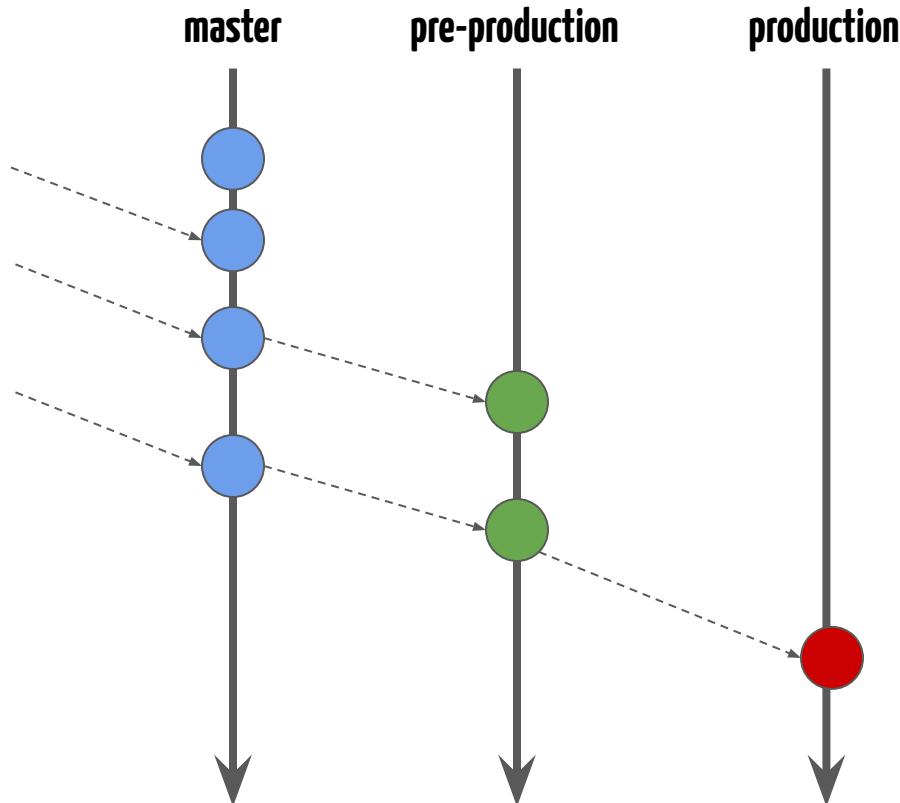
Fixing pre-production
bugs...

Gitlab flow



Fixing pre-production
bugs...

Gitlab flow



Production:

- infinite lifetime
- stable
- production ready

Gitlab flow

PROS

- Continuous delivery graal
- Simpler than git flow
- Ideal for single production version

CONS

- High regression risks - needs lot of automated tests
- Hard to follow a release plan
- May need code freeze on master

3



Your branching strategy !

Define your own branching strategy !



Define a branching strategy for your team that matches the project specificity



Apply your strategy during the project development



Defend it in front of a jury

CONTRIBUTING.md

Add a CONTRIBUTING.md file in your project that explains your branching strategy

Contributing to the Spring Framework

First off, thank you for taking the time to contribute! 👍🎉

Table of Contents

- [Code of Conduct](#)
- [How to Contribute](#)
 - [Discuss](#)
 - [Create an Issue](#)
 - [Issue Lifecycle](#)
 - [Submit a Pull Request](#)
- [Build from Source](#)
- [Source Code Style](#)
- [Reference Docs](#)

Code of Conduct

This project is governed by the [Spring Code of Conduct](#). By participating you are expected to uphold this code. Please report unacceptable behavior to spring-code-of-conduct@pivotal.io.

How to Contribute

Example here: <https://github.com/spring-projects/spring-framework/blob/master/CONTRIBUTING.md>

Q&A