



ПЕРВОИСТОЧНИКИ. ОБНОВЛЕНИЕ. ТЕГИ

Сессия 3

МАРТ 21, 2016

ПЕРВОИСТОЧНИКИ (UPSTREAMS)

Первоисточник (upstream) — термин Гита для обозначения удаленного связанного объекта.

Для репозитория это репозиторий-источник, с которого он был клонирован, а для локальной ветки — связанная ветвь в репозитории-источнике.

У вас может быть несколько репозиториев.

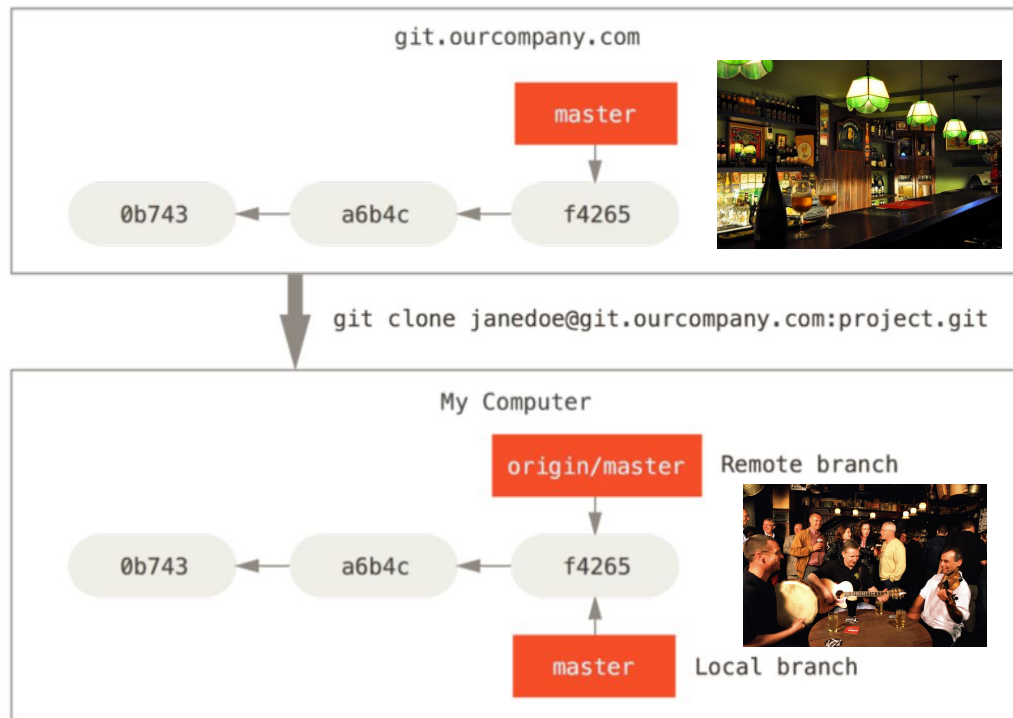
“ORIGIN” — ЭТО ПСЕВДОНИМ

После того как мы выполнили “git clone”, мы получаем полную локальную копию удаленного репозитория.

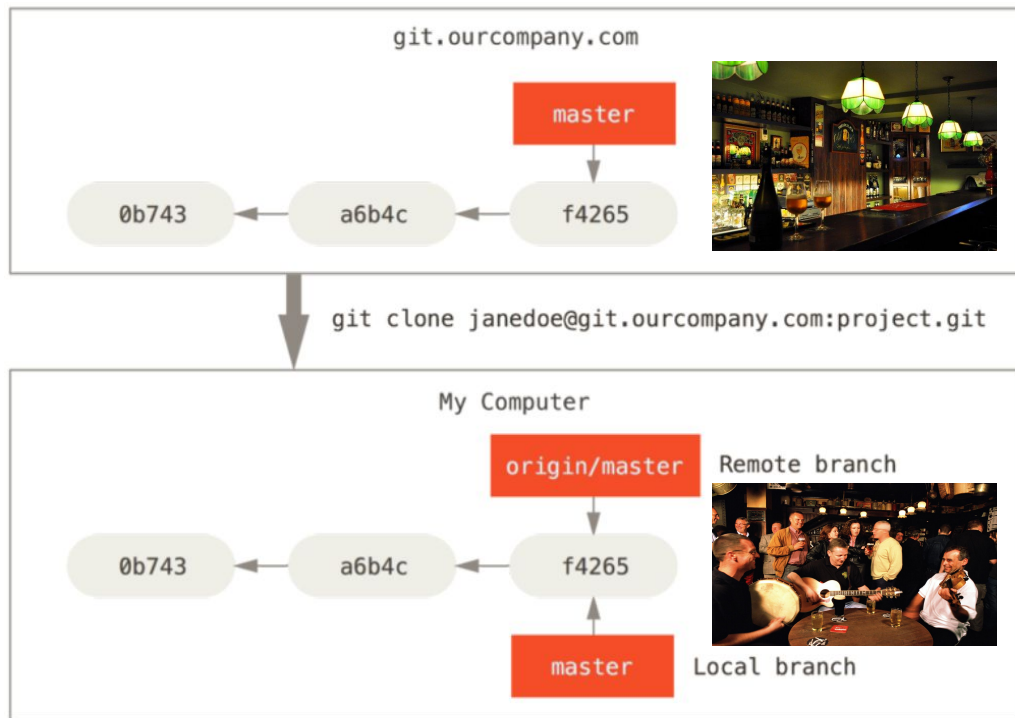
Поскольку в Гите репозиторий может быть несколько, то для каждого из них выбирается псевдоним, по которому мы можем к нему обращаться.

“Origin” — псевдоним по-умолчанию, для первого удаленного репозитория, если никакой другой не был задан.

Точно так же как и “master” — имя выбираемое по-умолчанию для первой ветви в только что созданном новом репозитории.



“ORIGIN” – ЭТО ПСЕВДОНИМ



Все `origin/*` ветви не достижимы напрямую, так что вы не можете модифицировать их указатели.

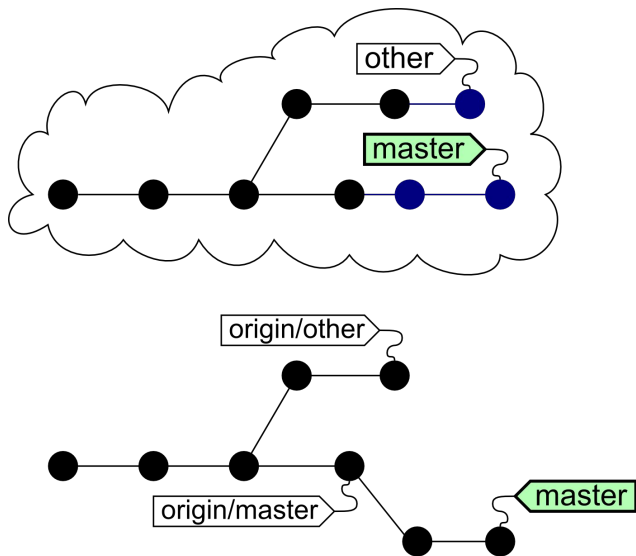
Вместо этого вы создаете локальные ветки, указывающие на одну из `origin/*` ветвей, которые на самом деле не являются ветвями на удаленном репозитории.

На самом деле, они являются копиями удаленных ветвей, которые существуют в вашем клоне.

Этот клон, в свою очередь, хранит ссылку на репозиторий-источник.

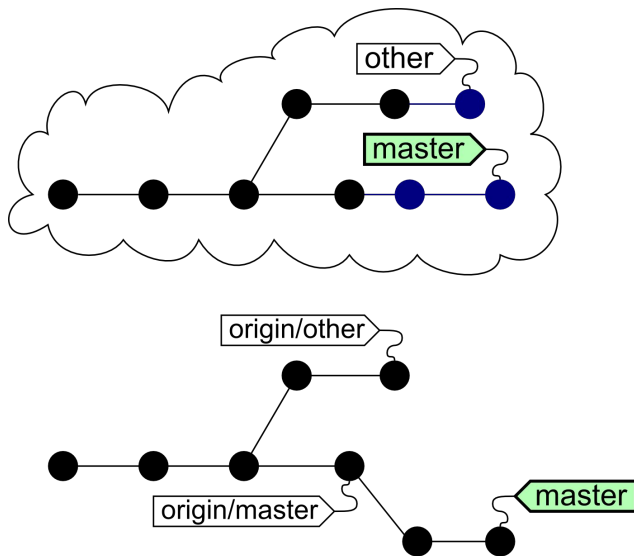
СИНХРОНИЗАЦИЯ ЛОКАЛЬНОГО РЕПОЗИТОРИЯ (FETCH)

“Fetch” — это команда Гиту обновить указанный локальный репозиторий последними изменения из репозитория-источника.



СИНХРОНИЗАЦИЯ ЛОКАЛЬНОГО РЕПОЗИТОРИЯ (FETCH)

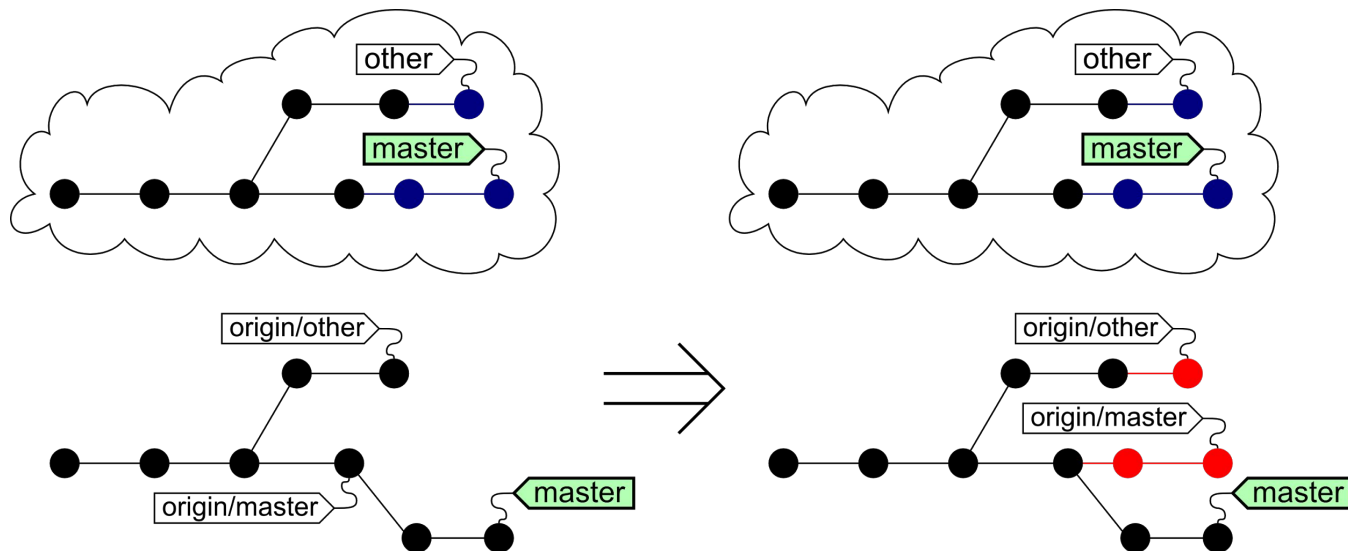
“Fetch” — это команда Гиту обновить указанный локальный репозиторий последними изменения из репозитория-источника.



```
git fetch origin
```

СИНХРОНИЗАЦИЯ ЛОКАЛЬНОГО РЕПОЗИТОРИЯ (FETCH)

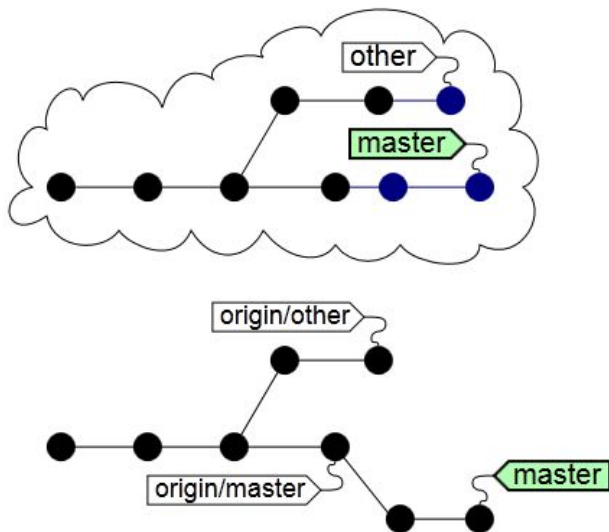
“Fetch” — это команда Гиту обновить указанный локальный репозиторий последними изменения из репозитория-источника.



`git fetch origin`

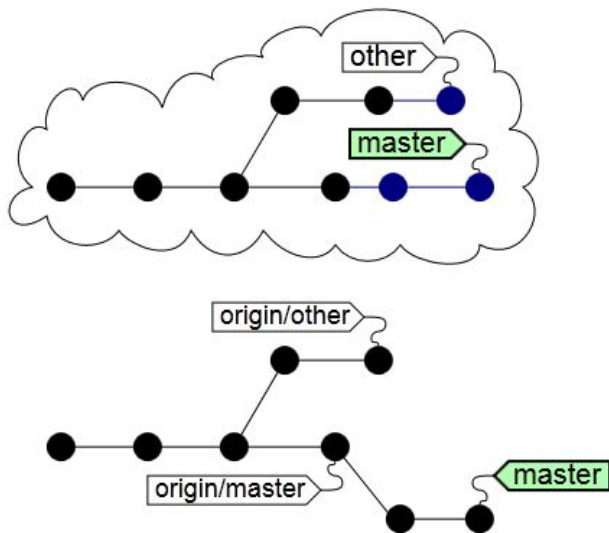
СИНХРОНИЗАЦИЯ + СЛИЯНИЕ (PULL)

“Pull” — на самом деле, всего лишь комбинация команд “fetch” и “merge”. Сначала ваш репозиторий синхронизирует данные с остальными и затем выполняется слияние текущей ветки и ее первоисточника.



СИНХРОНИЗАЦИЯ + СЛИЯНИЕ (PULL)

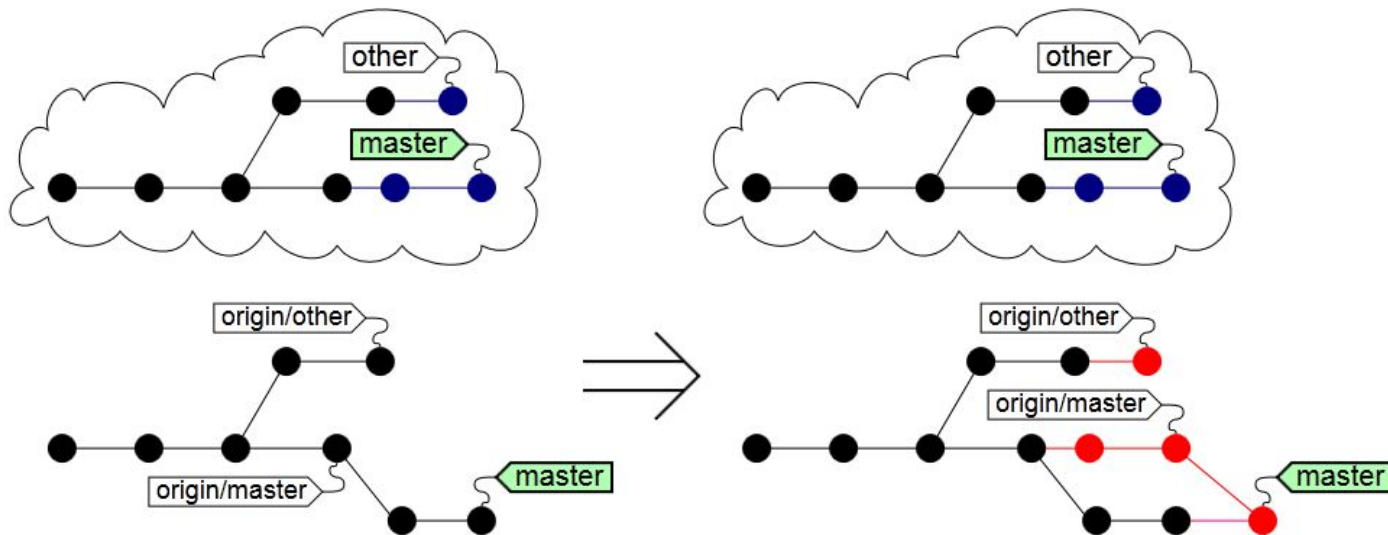
“Pull” — на самом деле, всего лишь комбинация команд “fetch” и “merge”. Сначала ваш репозиторий синхронизирует данные с остальными и затем выполняется слияние текущей ветки и ее первоисточника.



```
git pull origin
```

СИНХРОНИЗАЦИЯ + СЛИЯНИЕ (PULL)

“Pull” — на самом деле, всего лишь комбинация команд “fetch” и “merge”. Сначала ваш репозиторий синхронизирует данные с остальными и затем выполняется слияние текущей ветки и ее первоисточника.



`git pull origin`

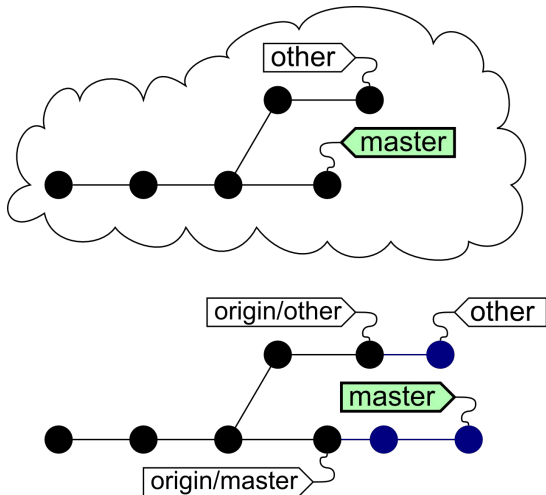
СИНХРОНИЗАЦИЯ + СЛИЯНИЕ (PULL)

Стоит заметить, что “pull”
выполняет слияние (обновление)
только для текущей ветки.

Поэтому важно не забывать обновить другие ветки
после переключения на них!

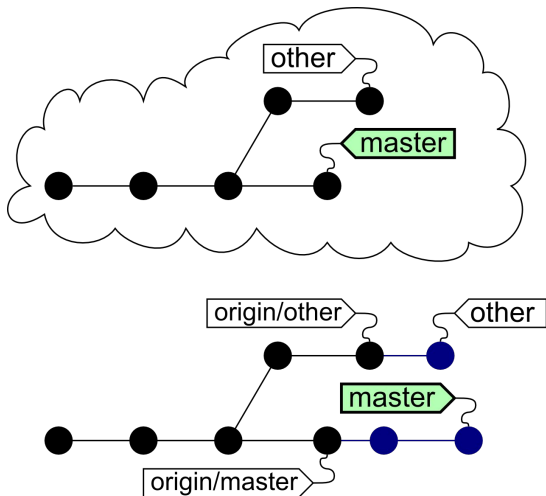
ПУБЛИКАЦИЯ ИЗМЕНЕНИЙ (PUSH)

“Push” — это команда Гиту опубликовать локальные коммиты. Публикация изменений влечет за собой неявное fast-forward слияние ветвей на репозиториях получателях.



ПУБЛИКАЦИЯ ИЗМЕНЕНИЙ (PUSH)

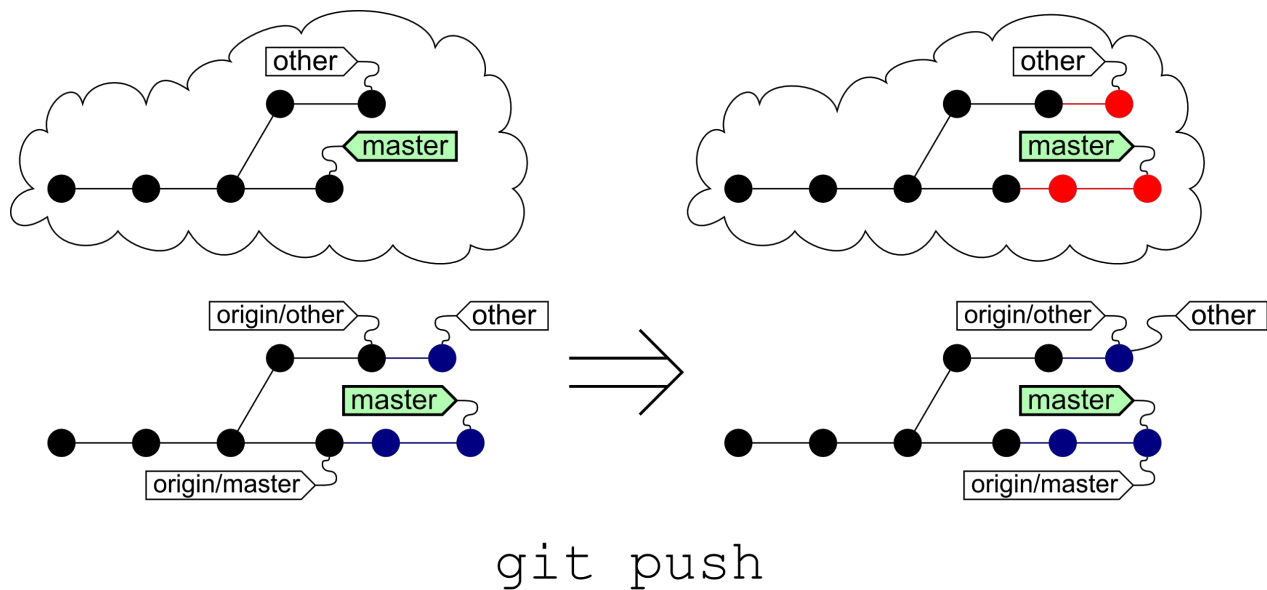
“Push” — это команда Гиту опубликовать локальные коммиты. Публикация изменений влечет за собой неявное fast-forward слияние ветвей на репозиториях получателях.



git push

ПУБЛИКАЦИЯ ИЗМЕНЕНИЙ (PUSH)

“Push” — это команда Гиту опубликовать локальные коммиты. Публикация изменений влечет за собой неявное fast-forward слияние ветвей на репозиториях получателях.

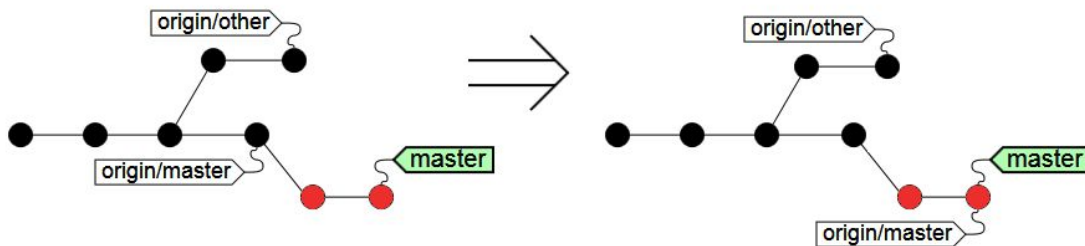
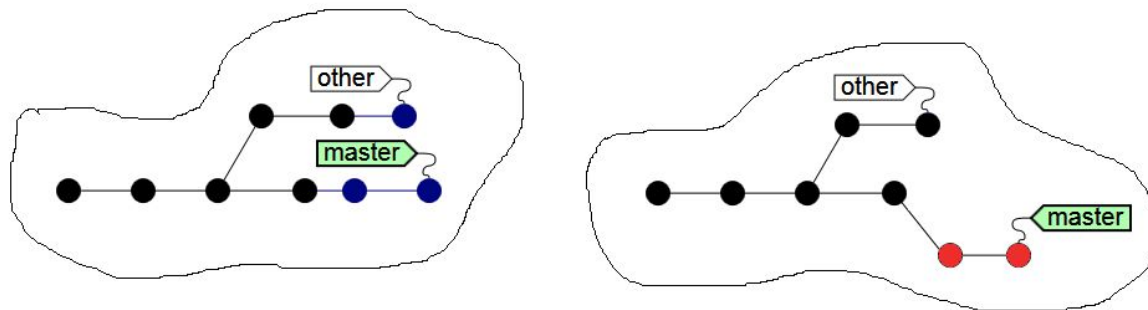


ПУБЛИКАЦИЯ ИЗМЕНЕНИЙ (PUSH)

По-умолчанию, Гит публикует изменения со всех ветвей.

Поэтому всегда указывайте ветвь, изменения которой должны быть опубликованы.

ПРИНУДИТЕЛЬНАЯ ПУБЛИКАЦИЯ (FORCE PUSH)



`git push -f`

ПРИНУДИТЕЛЬНАЯ ПУБЛИКАЦИЯ (FORCE PUSH)



ПРИНУДИТЕЛЬНАЯ ПУБЛИКАЦИЯ (FORCE PUSH)



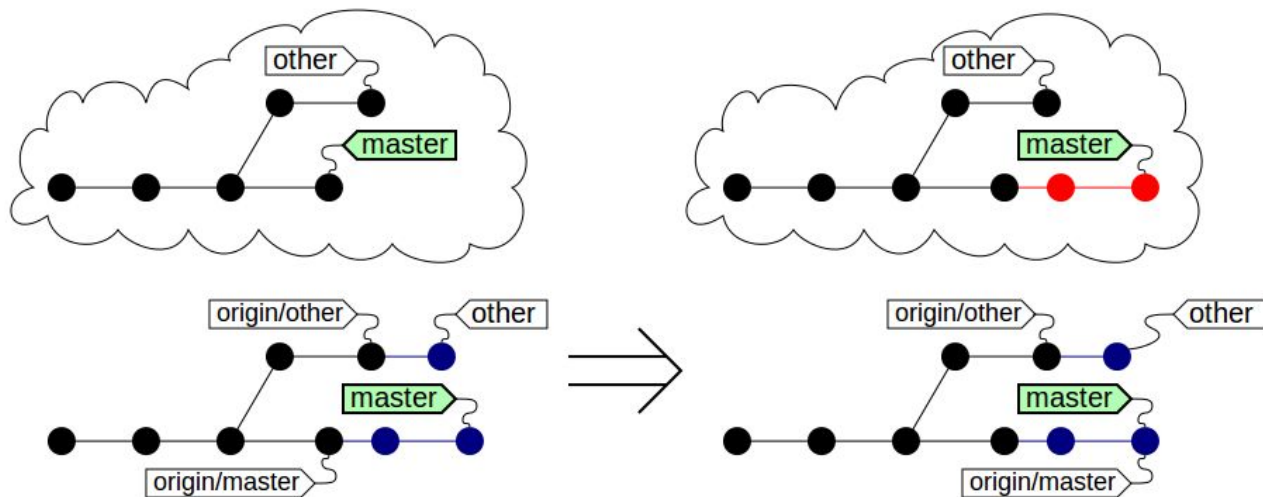
Никогда не используйте флаг --force (-f) без указания имени ветки!
НИКОГДА! Так как это основная причина потери данных в Гите!

ПРИНУДИТЕЛЬНАЯ ПУБЛИКАЦИЯ (FORCE PUSH)

Всякий раз, когда вы делаете “force push”
убедитесь, что
вы громко сообщили об этом всей команде.

ИЗМЕНЕНИЕ ПОЛИТИКИ ПУБЛИКАЦИИ

Вы можете изменить поведение Гита по-умолчанию при отправке коммитов.



git push

(config: push.default=simple)

ИЗМЕНЕНИЕ ПОЛИТИКИ ПУБЛИКАЦИИ

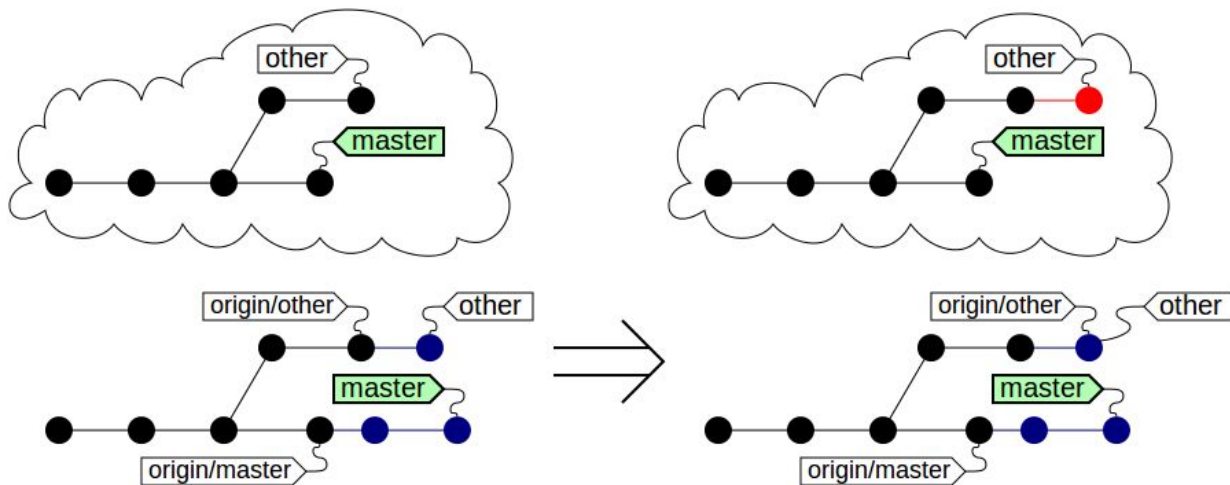
Есть несколько возможных вариантов для задания “push.default”:

- `nothing` — не отправлять ничего, пока имена репозитория и ветвей не будут заданы явно.
- `current` — отправить текущую ветку, которая обновит ветвь с точно таким же именем на получателе.
- `upstream` — отправить текущую ветку обновить связанную с ней ветвь (@{upstream}).
- `simple` — работает как и предыдущая, но не отправляет ветвь, если связанная ветка имеет другое имя.
- `matching` — отправляет все локальные ветки обновить ветви с совпадающими именами на получателе.

“Matching” была политикой по-умолчанию до выхода Гита версии 2.0.

ИЗМЕНЕНИЕ ПОЛИТИКИ ПУБЛИКАЦИИ

Но я все же строго рекомендую всегда явно указывать имя ветки.



`git push origin other`

ИСПОЛЬЗУЙТЕ ТЕГИ НА ВАЖНЫХ ВЕТКАХ

Тег — неизменяемый указатель.

Самый простой способ создать тег на текущий коммит это:

```
git tag v1.4
```

или так, с комментарием

```
git tag -a v1.4 -m 'my version 1.4'
```

Как уже было сказано, в случае потери данных или истории, вы можете просто загрузить коммит из вашего тега.

Q&A

ПОЛЕЗНЫЕ ССЫЛКИ И РЕСУРСЫ

Основные ресурсы:

1. [Книга "Pro Git"](#)
2. [Официальная документация](#)
3. [Git magic](#)

Самоучители:

1. [Неплохой пошаговый самоучитель](#)
2. [Become a Git guru](#)

Дополнительно:

1. [Git visual reference](#)

Ссылки по теме:

1. [Git best practices](#)
2. [Upstream term discussion](#)
3. [Fast-forward проблема](#)