

Linux and more. Git advanced.

Artem Trunov for Ozon Masters



На прошлых занятиях

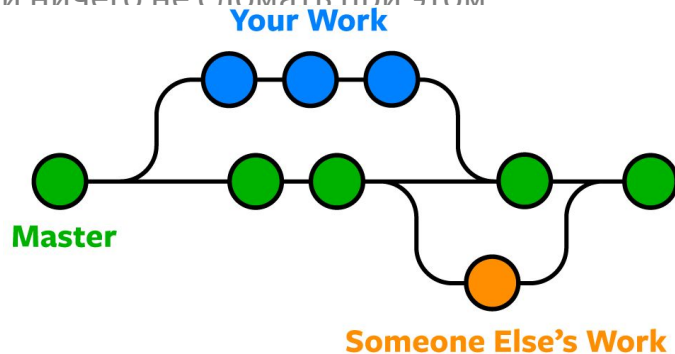
- Познакомились с системами контроля версий (CVS)
- Начали работать с git и Github
 - Создали репо, добавили файл, commit, push
 - Clone на другой машине (в другой директории) и работа с двумя репо.
 - Разрешение конфликтов при слиянии
 - Работа с коллабораторами
 - Работа с deploy-key

На этом занятии

- Как вносить изменения в “чужой” репо (сделать pull-request)
 - Как это принято в проектах open source (и не только)
- Практика: Pull request
- hub - утилита для работы с Github
 - Реализует пул-реквесты через командную строку!

О ветках

- Ветка - копия репо по состоянию на определенный коммит
- Удобный способ управлять разработкой
- Master (main) - “главная” ветка, в которой собирается код
 - Одно правило - в main всегда рабочий код, готовый к выпуску в прод.
- Разработка нового и починка старого кода ведется в отдельных ветках.
 - Типа песочницы, где можно редактировать, коммитить, и ничего не сломать при этом
 - Которые потом сливаются в главную.
 - В идеале - на каждую задачу или баг создается ветка
 - После слияния ветка больше не нужна и удаляется
 - Таким образом девелоперы не мешают друг другу



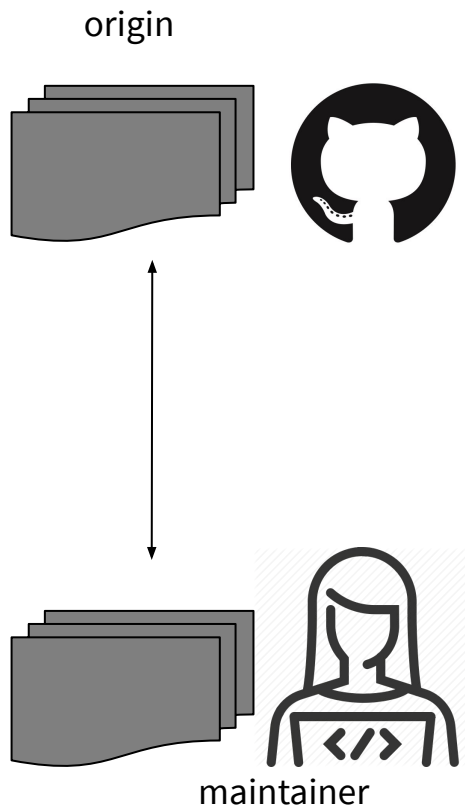
Git-flow and github-flow

- Git-flow - схема наименования веток и принципы работы
 - Main (master)
 - Develop
 - Feature branches
 - Release branches
 - Hotfix
- Github flow
 - Main (master)
 - Feature branches
- Хорошие обзоры
 - Git flow: <https://nvie.com/posts/a-successful-git-branching-model>
 - Github-flow: <https://guides.github.com/introduction/flow>

Способы совместной работы

- Shared repository
 - У всех разработчиков есть доступ на запись
 - Но никто не комитит прямо с ветку мастер!
 - Разработка ведется в ветках
 - пулл-запросы на слияния своей ветки в мастер используются для code review
 - Хорошая модель для небольших и дисциплинированных команд
- Fork & Pull-request
 - Только у ответственных (maintainers) есть доступ на запись.
 - Разработка ведется в форках главного репозитория
 - Модель для open-source проектов

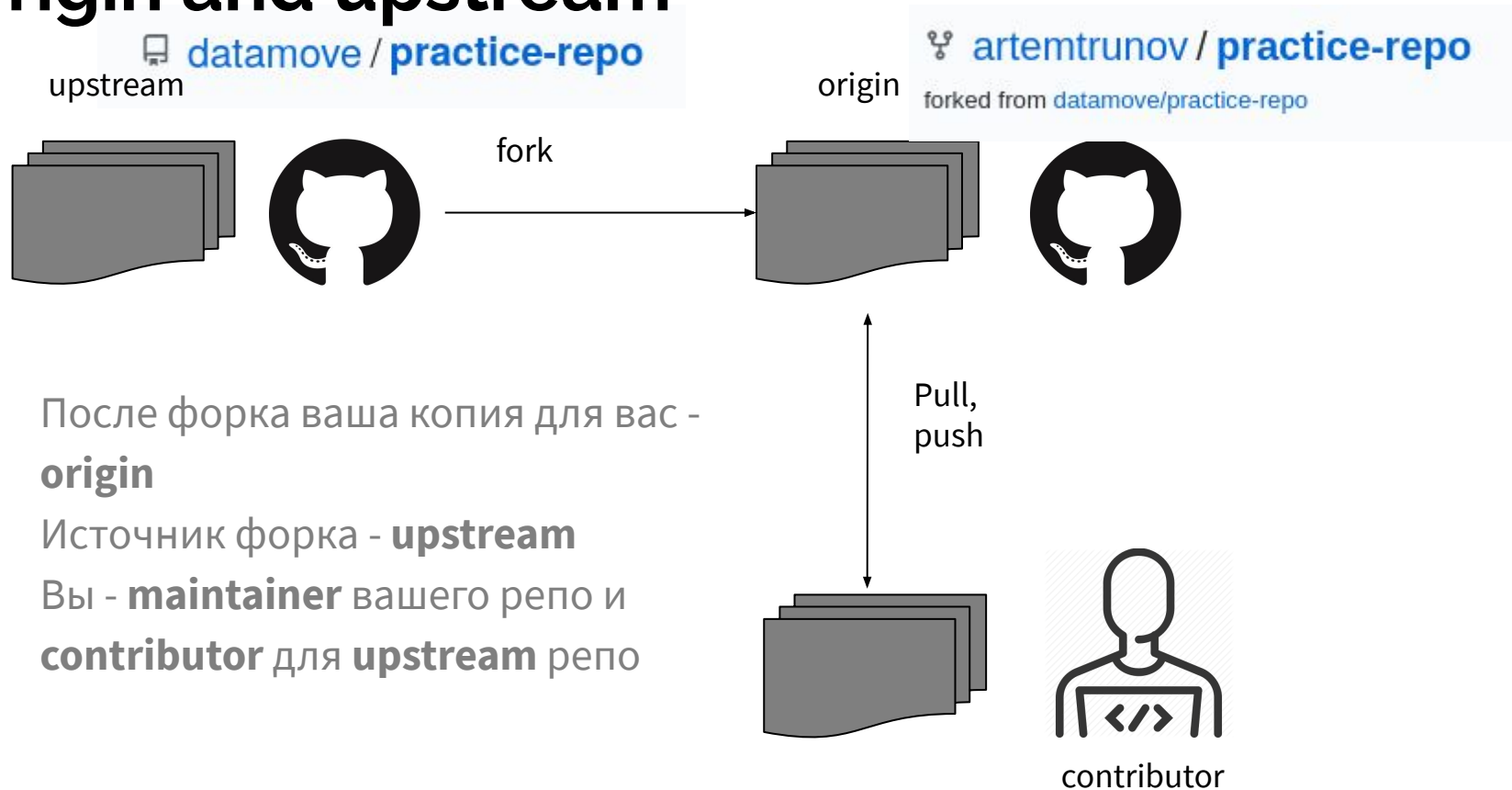
remote



- Remote - другая копии репозитория
 - Т.е. название + URL
- Ваша копия репозитория на гитхабе называется **origin**
- Вы - **maintainer** вашего репо
- Вы синхронизируете вашу локальную копию с **origin**
 - `git push [origin master]`

```
datamove@linux1:~/practice-repo$ git remote
origin
datamove@linux1:~/practice-repo$ git remote -v
origin  https://github.com/datamove/practice-repo.git (fetch)
origin  https://github.com/datamove/practice-repo.git (push)
datamove@linux1:~/practice-repo$
```

Origin and upstream



Fork, pull-request

- Это не функции гита
- Это функции Github.com (и других)
 - Которые облегчают разработчикам жизнь.
 - Хотя можно все сделать используя только команды гит

Практика. С высоты птичьего полета

- Форкаем чужой репозиторий
- Клонировем наш форк на сервер
- Создаем в нем новую ветку
- В этой ветке делаем изменения
- Фиксируем наши изменения (commit)
- Синхронизируем изменения в нашем форке на гитхабе (push)
- Делаем запрос на слияние ветки в нашем форке с мастер-веткой в оригинале (pull request).

Практика

<https://github.com/datamove/ozonmasters-linux/blob/master/tutorials/Git-pull-request.md>

- На сайте Github.com
 - Fork datamove/practice-repo
- На нашем сервере под персональным аккаунтом
 - `git clone https://github.com/your_nick/practice-repo`
 - `cd practice-repo`
 - `git checkout -b ваш_гх_ник-hello`
 - Создайте программу hello world на питоне в файле с названием ваш_гх_ник.py
 - `git add, git commit, git push`
- На сайте Github.com
 - В вашем форке Pull request
- На нашем сервере под персональным аккаунтом
 - `git checkout master`
 - `git pull upstream`
 - `git push origin`

Утилита hub

- От создателей Github
- Повторяет некоторые команды git
 - Может быть использована как замена команде git
 - Создатели даже рекомендуют сделать алиас, т.е. использовать hub как git.
- Реализует новые команды с помощью Github API
 - hub fork, hub pull-request etc
- Скачать
 - <https://github.com/github/hub/releases/download/v2.14.2/hub-linux-amd64-2.14.2.tgz>
 - `sudo apt install hub`
- Как использовать
 - `https://hub.github.com`

Практика с утилитой hub (самостоятельно)

- Удалите ваш форк practice-repo на сайте Github.
 - Settings->Danger Zone в конце страницы, Delete this repository
- Удалите папку practice-repo на сервере
 - `cd ~; rm -rf practice-repo`

```
git clone https://github.com/datamove/practice-repo.git
hub fork --remote-name origin
git remote -v
git checkout -b ваш_гх_ник-1
Git branch
nano ваш_гх_ник.py
git add ваш_гх_ник.py && git commit -a -m "added hello"
git push -u origin ваш_гх_ник-1
hub pull-request
hub pr list
```