

Дорожная карта

- 0. Приступаем
- 1. Введение в Git
- 2. Начало работы с Git
- 3. Просмотр истории
- 4. Ветвление
- 5. Слияние
- 6. Отмена изменений
- 7. Рабочий процесс

Дорожная карта (продолжение)

8. Работа в команде ➡

9. Метки

10. Последние штрихи

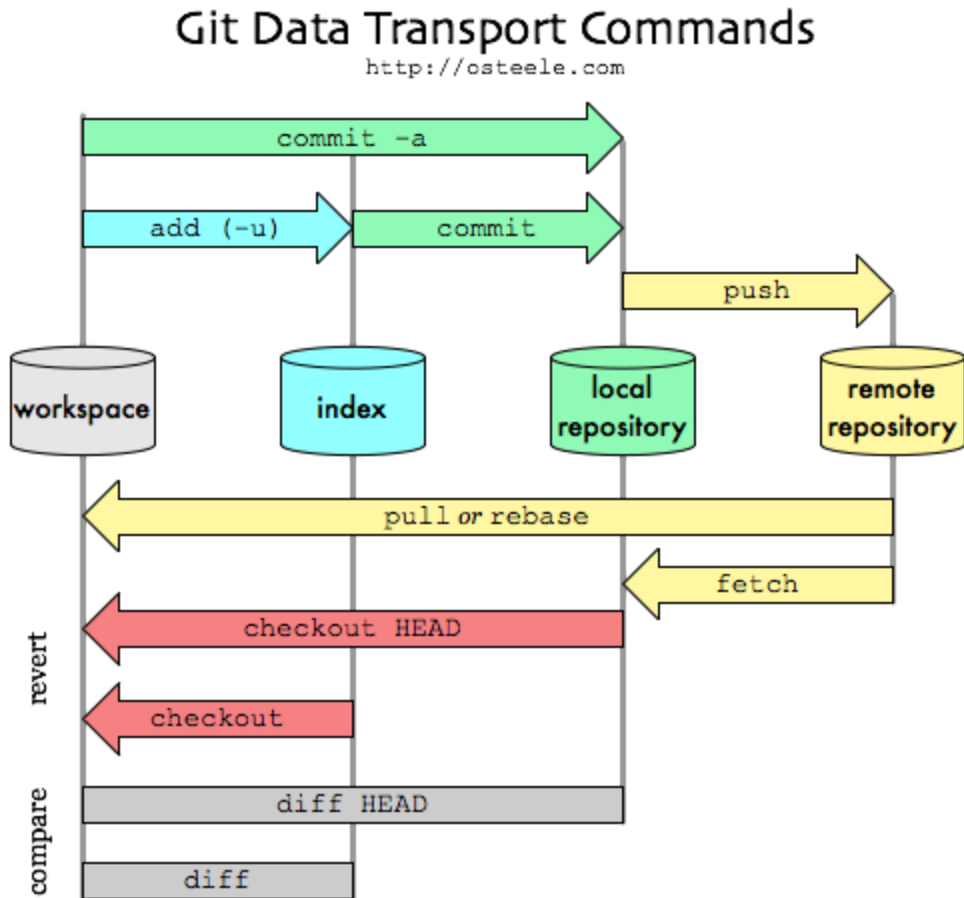
11. Завершаем

Кто это написал?!

```
$ git blame
```

Workflow

Команды передачи данных



Удалённые репозитории

Удалённые репозитории представляют собой версии вашего (или чужого) проекта, сохранённые в интернете или ещё где-то в сети.

У вас может быть несколько удалённых репозиториев, каждый из которых может быть доступен для чтения или для чтения-записи.

Взаимодействие с другими пользователями предполагает управление удалёнными репозиториями, а также отправку и получение данных из них.

Управление репозиториями включает в себя как умение добавлять новые, так и умение удалять устаревшие репозитории, а также умение управлять различными удалёнными ветками, объявлять их отслеживаемыми или нет и так далее.

Настройка SSH

Проверка ключей:

```
$ ls -al ~/.ssh  
# Lists the files in your .ssh directory, if they exist
```

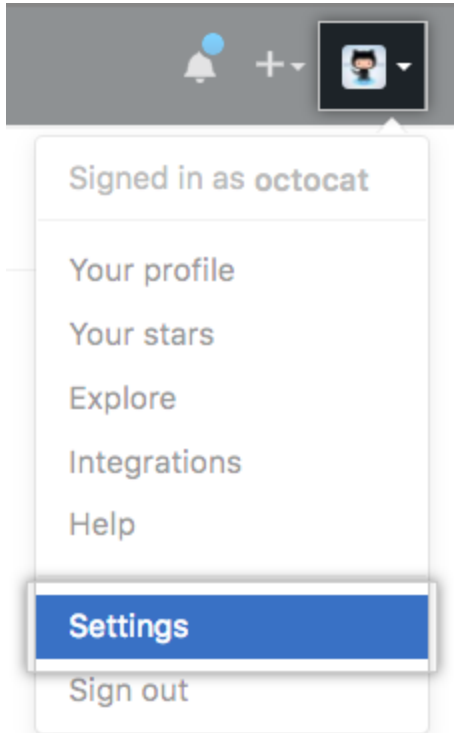
- Ищите пару файлов с именами `что-нибудь` и `что-нибудь.pub`, где `что-нибудь` - обычно `id_dsa` или `id_rsa`.
- Файл с расширением `.pub` - это ваш открытый ключ, а второй файл - ваш секретный ключ.
- Если у вас нет этих файлов (или даже нет каталога `.ssh`), вы можете создать их, запустив программу `ssh-keygen`:

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

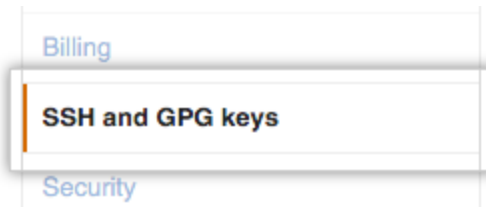
Содержимое файла `~/.ssh/id_rsa.pub` добавляется в список ключей на сервере.

Доступ к GitHub

В правом верхнем углу нажать на иконку пользователя и выбрать **Settings** :

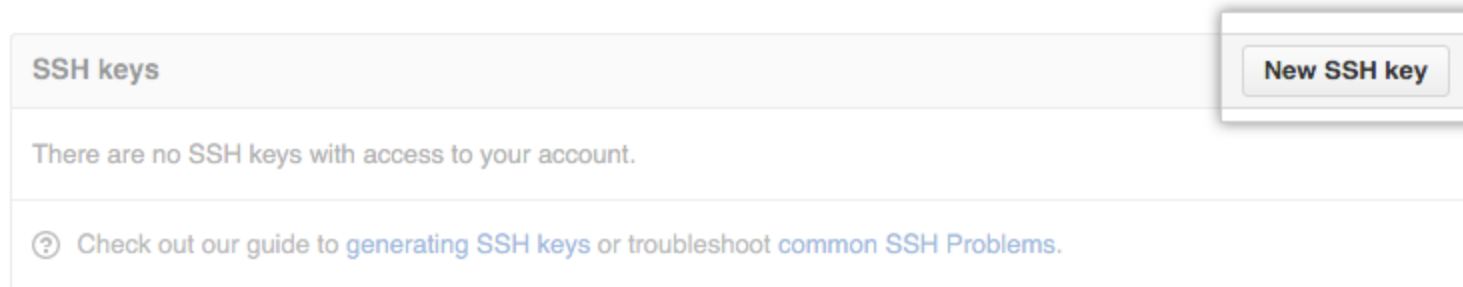


Слева на панели настроек выбрать **SSH and GPG keys** :



Доступ к GitHub (продолжение)

Нажать **New SSH key** или **Add SSH key** :



Доступ к GitHub (продолжение)

В поле **Title** добавьте краткое описание ключа. Например, для доступа с работы **Work**. Вставьте содержимое открытого (**что-нибудь.pub**) ключа в поле **Key**:

SSH keys

New SSH key

There are no SSH keys with access to your account.

Title

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

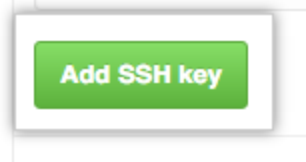
Add SSH key

?

 Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

Доступ к GitHub (продолжение)

Нажмите Add SSH key :



Укажите пароль от ключа при необходимости:

Confirm password to continue

Password [Forgot password?](#)

Confirm password

Доступ к GitHub (продолжение)

Проверка подключения:

```
$ ssh -T git@github.com
Hi maxx27! You've successfully authenticated, but GitHub does not provide shell access.
```

В некоторых случаях нужно подтвердить подключение набрав `yes` в ответ:

The authenticity of host 'github.com (IP ADDRESS)' can't be established. RSA key fingerprint is 16:27:ac:....df:a6:48. Are you sure you want to continue connecting (yes/no)?

В случае ошибки прочитать <https://help.github.com/en/articles/error-permission-denied-publickey> или спросить знающих людей.

- Git Extensions
 - General
 - Appearance
 - Revision links
 - Build server integration
 - Scripts
 - Hotkeys
 - Shell extension
 - Advanced
 - Detailed
 - SSH
- Git
- Plugins

Settings source: ☒ Global for all repositories

Specify which ssh client to use

☐ PuTTY

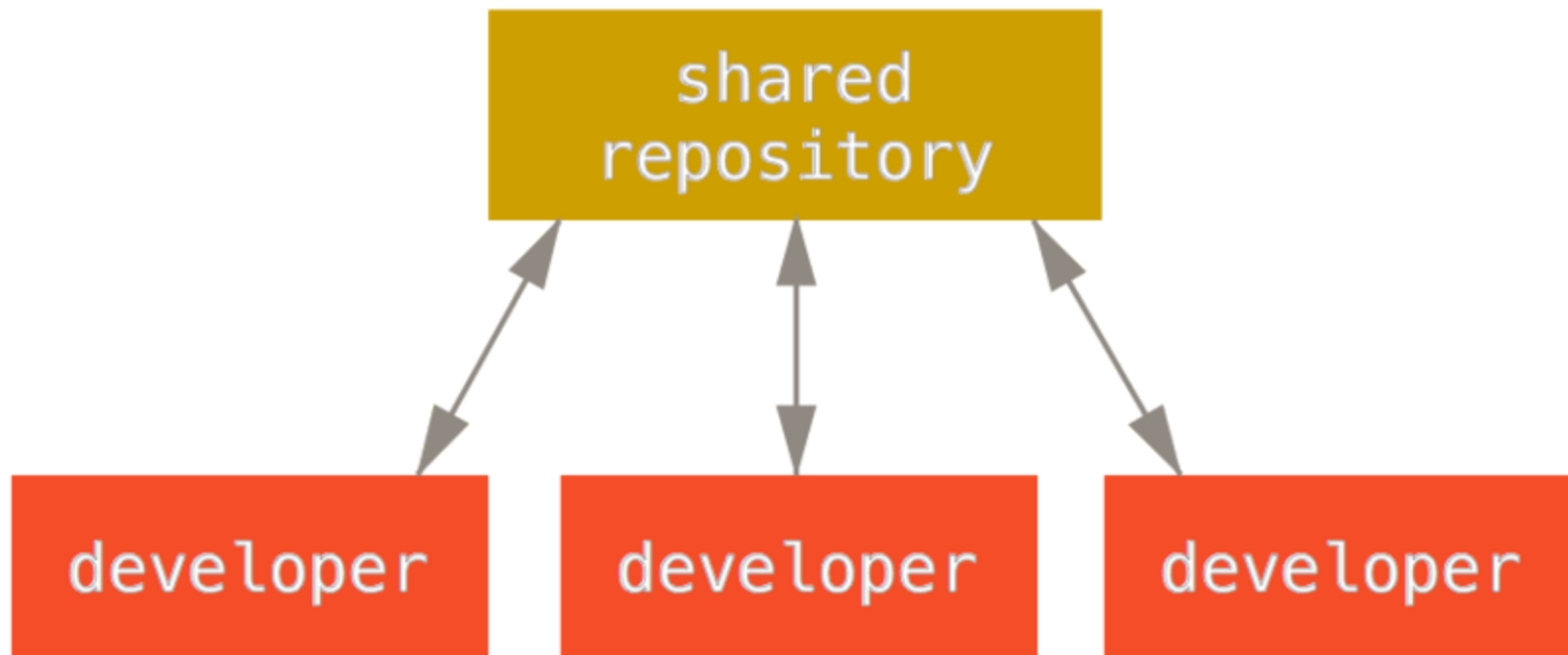
OpenSSH is a commandline tool. PuTTY is more userfriendly to use for windows users, but requires the PuTTY authentication client to run in the background.
When OpenSSH is used, command line dialogs are shown!

☒ OpenSSH☐ Other ssh client
Browse

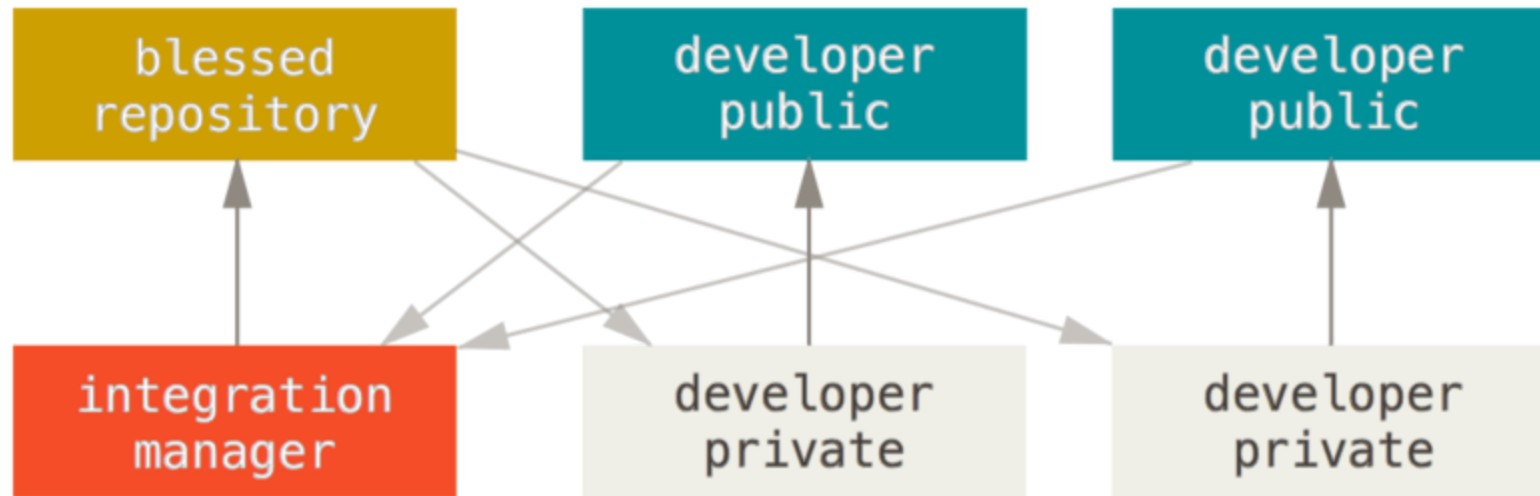
Распределенный рабочий процесс

- Распределенная природа Git позволяет более гибко взаимодействовать разработчикам в рамках проекта
- Каждый разработчик это и узел и хаб, то есть каждый разработчик может как *отправлять код* в другие репозитории, так и *поддерживать публичный репозиторий*, в который другие разработчики смогут отправлять свой код, взяв его за основу.
- Это предоставляет огромное количество вариаций для организации рабочего процесса над проектом и/или для команды.

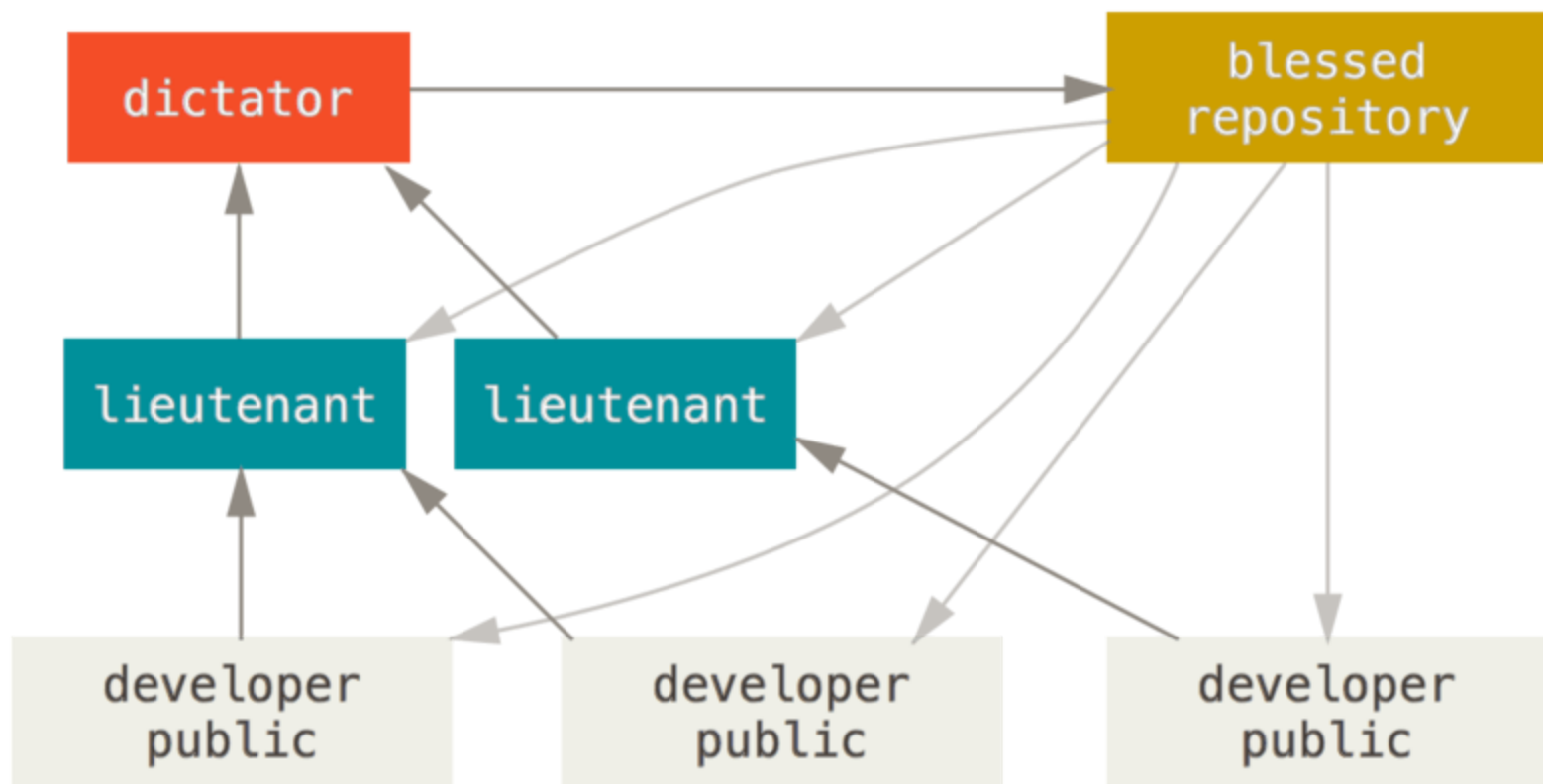
Централизованный рабочий процесс



Рабочий процесс с менеджером по интеграции



Рабочий процесс с Диктатором и Лейтенантами



Просмотр удалённых репозиториев

```
$ git clone https://github.com/schacon/ticgit
...
$ cd ticgit
$ git remote
origin
$ git remote -v
origin  https://github.com/schacon/ticgit (fetch)
origin  https://github.com/schacon/ticgit (push)
cho45   https://github.com/cho45/ticgit (fetch)
cho45   https://github.com/cho45/ticgit (push)
```

Добавление удалённых репозиториев

```
$ git remote
origin

$ git remote add pb https://github.com/paulboone/ticgit

$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
pb https://github.com/paulboone/ticgit (fetch)
pb https://github.com/paulboone/ticgit (push)

$ git fetch pb
remote: Counting objects: 43, done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 43 (delta 10), reused 31 (delta 5)
Unpacking objects: 100% (43/43), done.
From https://github.com/paulboone/ticgit
* [new branch]      master      -> pb/master
* [new branch]      ticgit      -> pb/ticgit
```

Получение изменений из удалённого репозитория - Fetch и Pull

```
$ git fetch [remote-branch]
$ git merge <remote-branch>

# pull = fetch + merge
$ git pull <remote-branch>

# pull = fetch + rebase
$ git pull --rebase <remote-branch>
```

Отправка изменений в удаленный репозиторий (Push)

```
$ git push origin master
```

Просмотр удаленного репозитория

```
$ git remote show origin
* remote origin
Fetch URL: https://github.com/schacon/ticgit
Push URL: https://github.com/schacon/ticgit
HEAD branch: master
Remote branches:
  master                                tracked
  dev-branch                            tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```

Удаление и переименование удалённых репозиториев

```
$ git remote rename pb paul  
$ git remote  
origin  
paul
```

```
$ git remote rm paul  
$ git remote  
origin
```

Удалённые ветки

Ветки на сервере имеют префикс `remotes/<server>/` :

```
$ git branch -a
* main
  oldmain
  remotes/origin/HEAD -> origin/main
  remotes/origin/bindings/libgit2sharp/020_2
  remotes/origin/bindings/libgit2sharp/022_1
  remotes/origin/brianmario/attr-from-tree
```


Удалить ветку на сервере

```
git push origin --delete my-branch  
git push origin :my-branch
```

Отслеживание веток

```
$ git remote show origin
* remote origin
Fetch URL: git@github.com:maxx27/training-git.git
Push URL: git@github.com:maxx27/training-git.git
HEAD branch: master
Remote branch:
  master tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (fast-forwardable)
```

Установка отслеживания изменений в удалённых ветках

```
$ git branch -u origin/master master  
$ git branch --set-upstream-to=origin/master master
```

Заключение