

# Знакомство с GitHub

Артамонов Ю.Н.

Университет "Дубна"  
филиал Котельники

1 февраля 2017 г.

# Общая информация о GitHub из Википедии

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome).

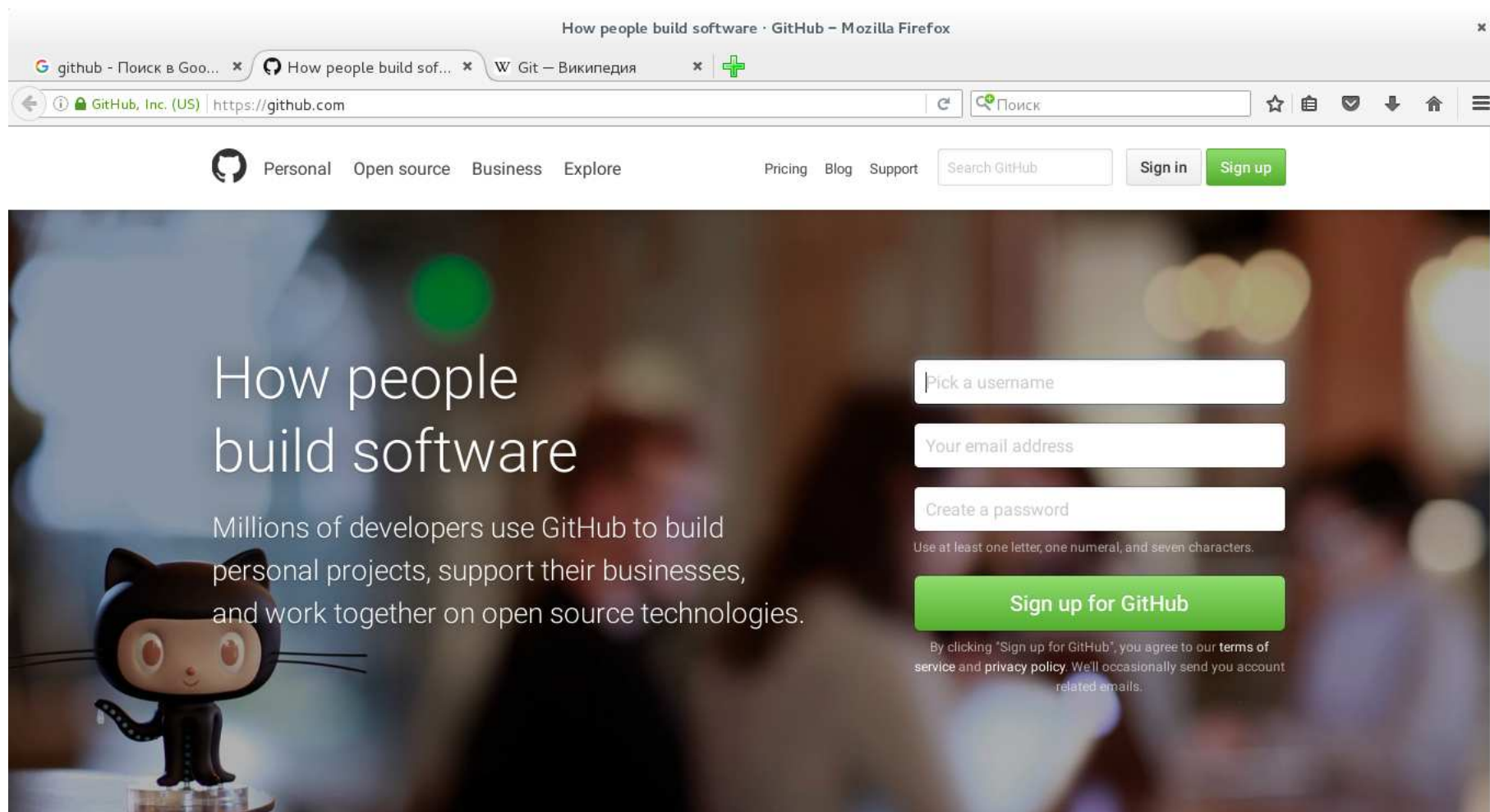
Сервис абсолютно бесплатен для проектов с открытым исходным кодом и предоставляет им все возможности (включая SSL), а для частных проектов предлагаются различные платные тарифные планы.

Слоган сервиса — «Social Coding» — на русский можно перевести как «Пишем код вместе». На футболках же печатают совсем другую фразу: «Fork you!» («Ответвись!»). С одной стороны, она созвучна с англоязычным ругательством и намекает на неформальную атмосферу. С другой, эти слова напоминают, что создавать новые форки с Git можно легко и безболезненно.

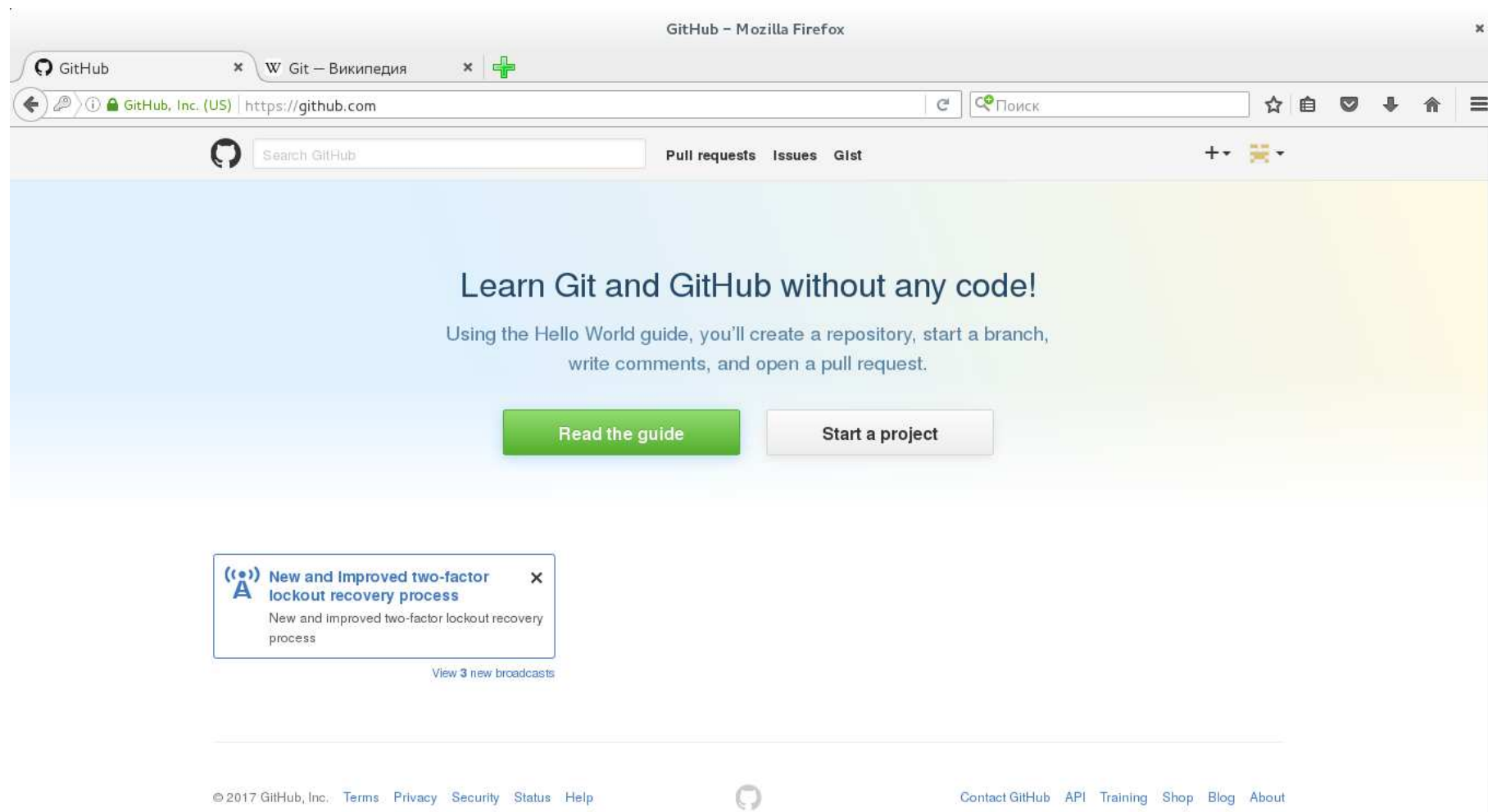
# Общая информация о GitHub из Википедии (продолжение)

Создатели сайта называют GitHub «социальной сетью для разработчиков». Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. С помощью широких возможностей Git программисты могут объединять свои репозитории — GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева. Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок. Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования. На платных тарифных планах можно создавать приватные репозитории, доступные ограниченному кругу пользователей. Кроме Git, сервис поддерживает получение и редактирование кода через SVN и Mercurial.

# Скриншот официального сайта, [ссылка на сайт](https://github.com)



# Начальная страница



# Создание нового репозитория

Create a New Repository – Mozilla Firefox

Create a New Reposi... x W Git – Википедия x +

GitHub, Inc. (US) | https://github.com/new Поиск

## Create a new repository

A repository contains all the files for your project, including the revision history.

**Owner** **Repository name**

junaart / Security ✓

Great repository names are short and memorable. Need inspiration? How about **urban-palm-tree**.

**Description** (optional)

Основы информационной безопасности

☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** Add a license: **GNU General Public License v3.0** ⓘ

**Create repository**

# Страница созданного проекта

The screenshot shows a web browser window with the title 'junaart/Security: Основы информационной безопасности – Mozilla Firefox'. The address bar displays 'https://github.com/junaart/Security'. The page content includes the repository name 'junaart / Security', navigation tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. Below these, the repository description 'Основы информационной безопасности' is shown with an 'Edit' button. A summary bar indicates '1 commit', '1 branch', '0 releases', '1 contributor', and 'GPL-3.0' license. Action buttons include 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A file list shows 'LICENSE' and 'README.md' as initial commits. The 'README.md' content is visible, featuring the title 'Security' and the subtitle 'Основы информационной безопасности'.

junaart/Security: Основы информационной безопасности – Mozilla Firefox

junaart/Security: Оchn... x W Git – Википедия x +

GitHub, Inc. (US) https://github.com/junaart/Security Поиск

This repository Search Pull requests Issues Gist + -

junaart / Security Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Основы информационной безопасности Edit

New Add topics

1 commit 1 branch 0 releases 1 contributor GPL-3.0

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

junaart Initial commit Latest commit b0eb818 20 seconds ago

LICENSE Initial commit 20 seconds ago

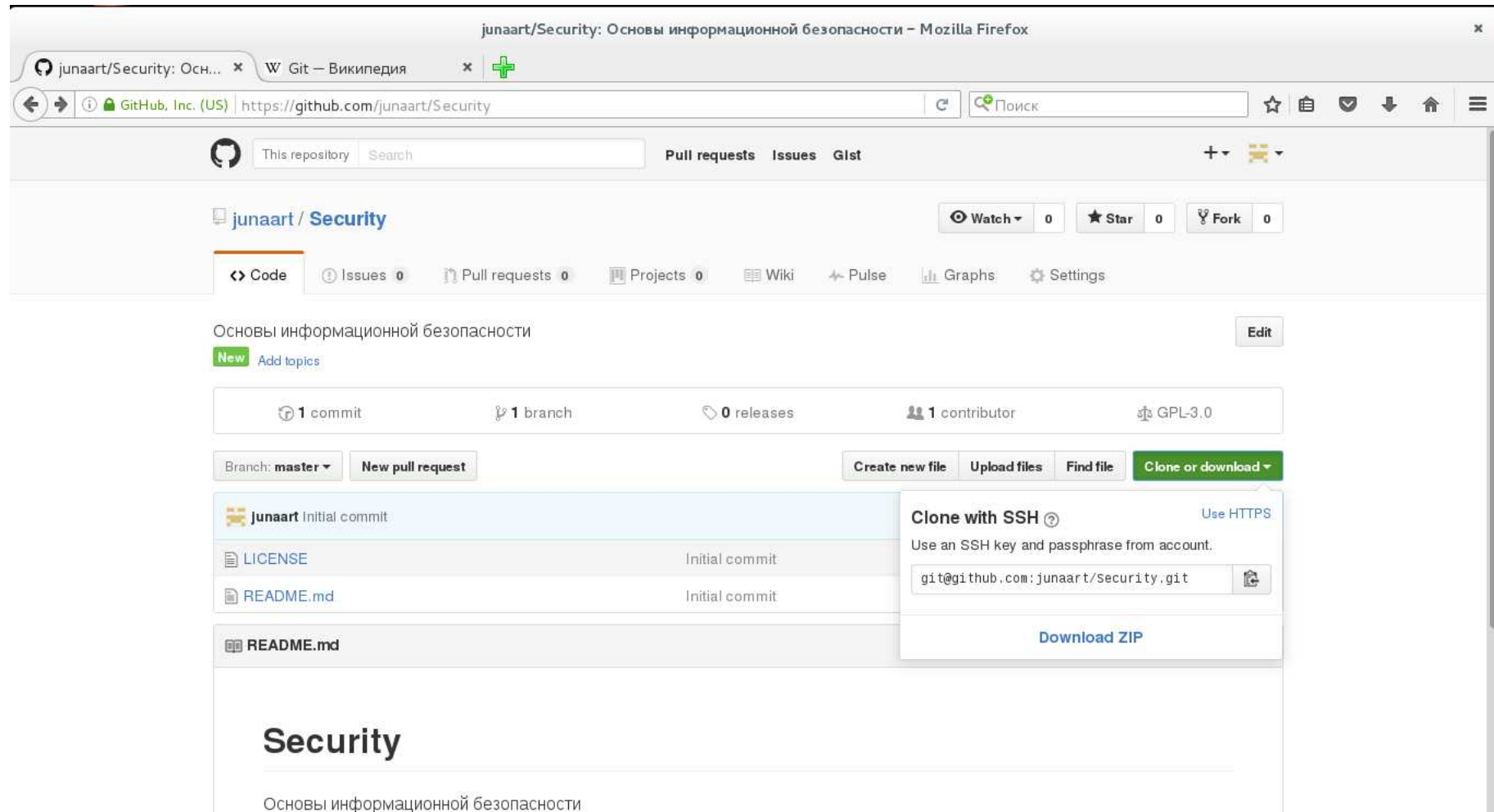
README.md Initial commit 20 seconds ago

README.md

## Security

Основы информационной безопасности

# Скачивание (клонирование) проекта по протоколам SSH, https, git, архив





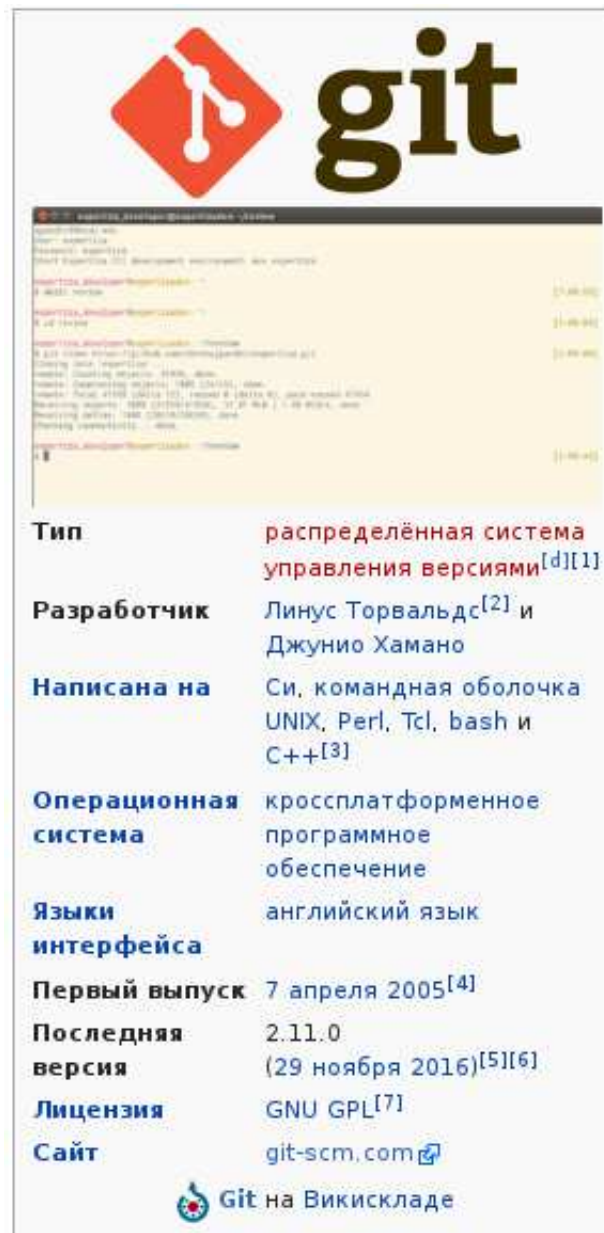
# Система контроля версий Git

Git (произн. «гит») — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.

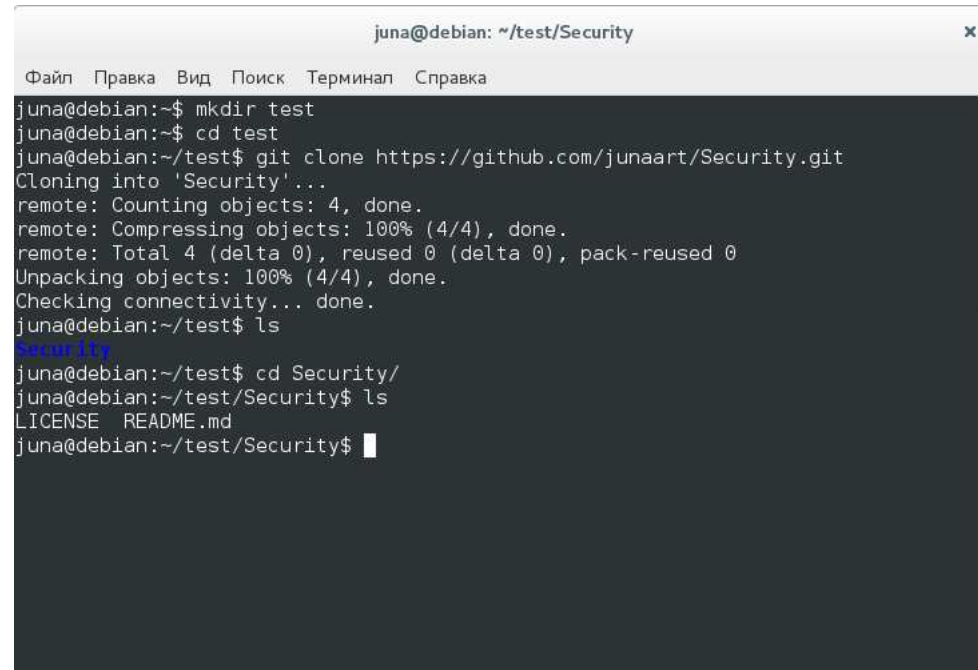
Примерами проектов, использующих Git, являются Ядро Linux, Swift, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, Qt и некоторые дистрибутивы Linux.

Программа является свободной и выпущена под лицензией GNU GPL версии 2.

# Логотип Git с Wiki



# Работаем с Git в консоле



```
juna@debian: ~/test/Security
Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~$ mkdir test
juna@debian:~$ cd test
juna@debian:~/test$ git clone https://github.com/junaart/Security.git
Cloning into 'Security'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
Checking connectivity... done.
juna@debian:~/test$ ls
Security
juna@debian:~/test$ cd Security/
juna@debian:~/test/Security$ ls
LICENSE  README.md
juna@debian:~/test/Security$
```

- Создаем тестовую папку (mkdir test)
- Переходим в тестовую папку (cd test)
- Клонировем удаленный проект с помощью git по протоколу https (git clone https://github.com/junaart/Security.git)
- Смотрим содержимое тестовой папки (ls)
- Переходим в папку склонированного проекта Security (cd Security)
- Смотрим содержимое этой папки (ls)

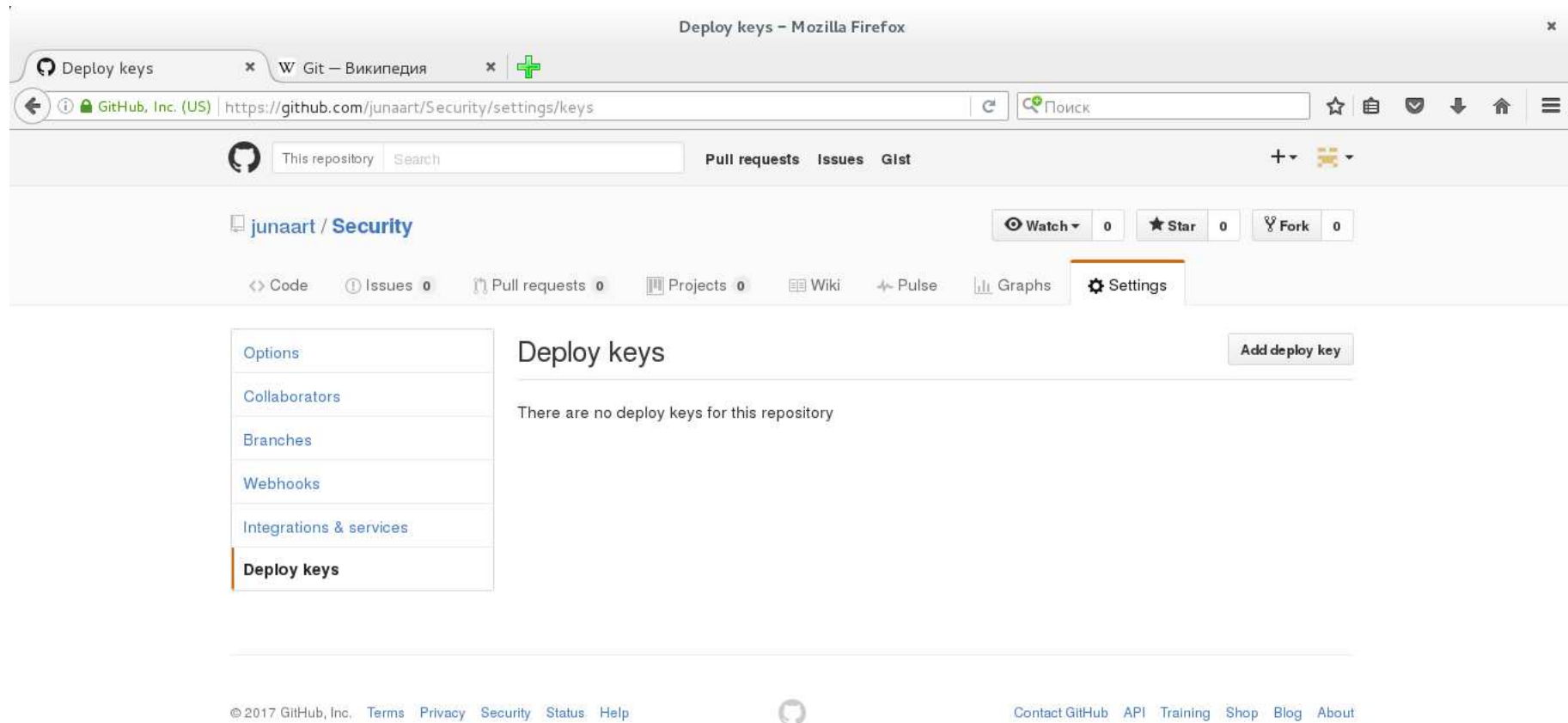
# Попытка клонирования по протоколу ssh

```
juna@debian: ~/test
Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~/test$ rm -R ./Security/
rm: удалить защищенный от записи обычный файл «./Security/.git/objects/9c/eccl4d4669ee8af2ca727a5d8cde10cd8b2d7cc»? y
rm: удалить защищенный от записи обычный файл «./Security/.git/objects/d0/3b10453aeac81a9a55cc3afd0c8f86ce9a0074»? y
rm: удалить защищенный от записи обычный файл «./Security/.git/objects/3d/ae3d3b84a5de974080c91653a8b0134bd8273a»? y
rm: удалить защищенный от записи обычный файл «./Security/.git/objects/b0/eb8189b225c9c25e57f97c98bd47665179a3f1»? y
juna@debian:~/test$ ls
juna@debian:~/test$ git clone git@github.com:junaart/Security.git
Cloning into 'Security'...
The authenticity of host 'github.com (192.30.253.113)' can't be established.
RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.113' (RSA) to the list of known hosts.
Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
juna@debian:~/test$
```

- Удаляем папку Security (`rm -R ./Security/`)
- Проверяем, что удалили (`ls`)
- Пытаемся клонировать (`git clone git@github.com:junaart/Security.git`)
- Получаем сообщение об ошибке - у нас нет ключа для доступа на сайт!
- Итак, нужно сгенерить ключ по асимметричному протоколу - приватный ключ хранить у себя, публичный положить на сайт github.

# Смотрим, где хранятся публичные ключи на github



<https://github.com/junaart/Security/subscription>

# Немного об ssh

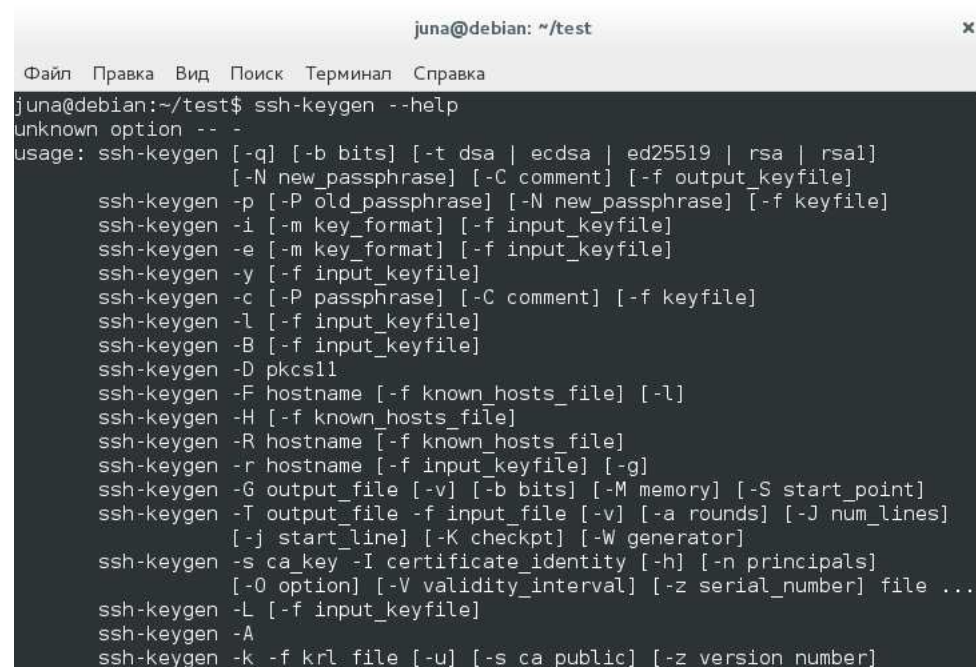
SSH (англ. Secure Shell — «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

# Немного об ssh (продолжение)

SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео (например, с веб-камеры). Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Window System.

Большинство хостинг-провайдеров за определённую плату предоставляют клиентам доступ к их домашнему каталогу по SSH. Это может быть удобно как для работы в командной строке, так и для удалённого запуска программ (в том числе графических приложений).

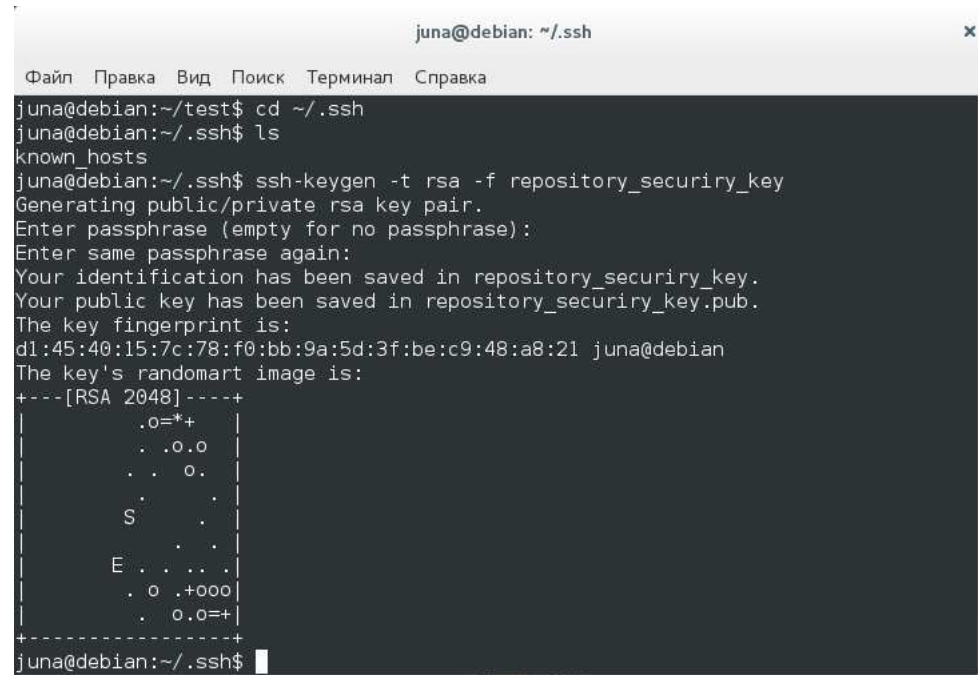
# Смотрим параметры ssh-keygen



```
juna@debian: ~/test
Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~/test$ ssh-keygen --help
unknown option -- -
usage: ssh-keygen [-q] [-b bits] [-t dsa | ecdsa | ed25519 | rsa | rsal]
                  [-N new_passphrase] [-C comment] [-f output_keyfile]
ssh-keygen -p [-P old_passphrase] [-N new_passphrase] [-f keyfile]
ssh-keygen -i [-m key_format] [-f input_keyfile]
ssh-keygen -e [-m key_format] [-f input_keyfile]
ssh-keygen -y [-f input_keyfile]
ssh-keygen -c [-P passphrase] [-C comment] [-f keyfile]
ssh-keygen -l [-f input_keyfile]
ssh-keygen -B [-f input_keyfile]
ssh-keygen -D pkcs11
ssh-keygen -F hostname [-f known_hosts_file] [-l]
ssh-keygen -H [-f known_hosts_file]
ssh-keygen -R hostname [-f known_hosts_file]
ssh-keygen -r hostname [-f input_keyfile] [-g]
ssh-keygen -G output_file [-v] [-b bits] [-M memory] [-S start_point]
ssh-keygen -T output_file -f input_file [-v] [-a rounds] [-J num_lines]
                  [-j start_line] [-K chkpt] [-W generator]
ssh-keygen -s ca_key -I certificate_identity [-h] [-n principals]
                  [-O option] [-V validity_interval] [-z serial_number] file ...
ssh-keygen -L [-f input_keyfile]
ssh-keygen -A
ssh-keygen -k -f krl_file [-u] [-s ca_public] [-z version_number]
```



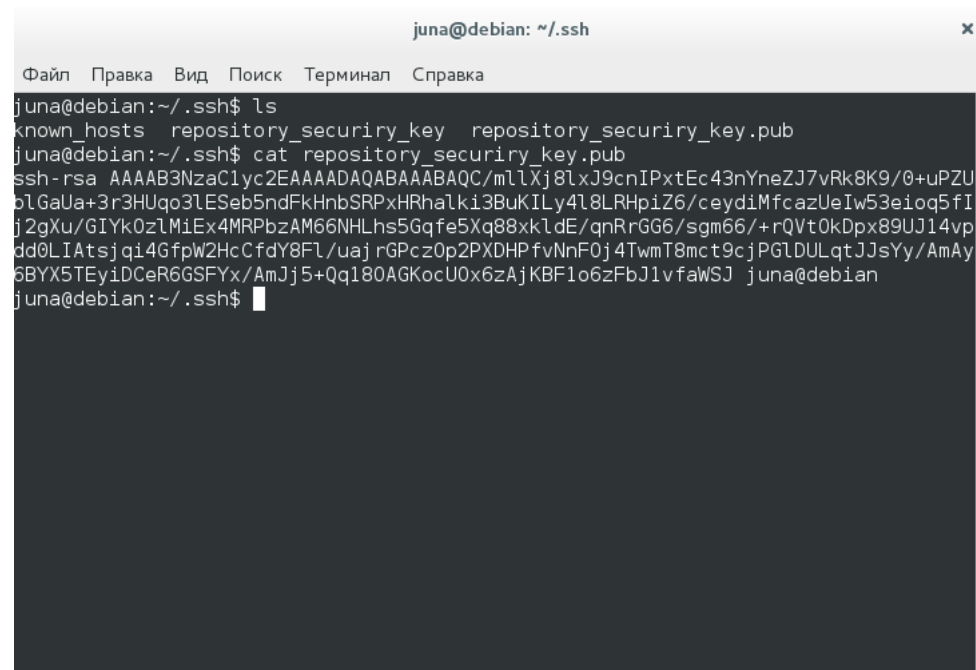
# Генерим ключи



```
juna@debian: ~/.ssh
Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~/test$ cd ~/.ssh
juna@debian:~/.ssh$ ls
known_hosts
juna@debian:~/.ssh$ ssh-keygen -t rsa -f repository_securiry_key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in repository_securiry_key.
Your public key has been saved in repository_securiry_key.pub.
The key fingerprint is:
d1:45:40:15:7c:78:f0:bb:9a:5d:3f:be:c9:48:a8:21 juna@debian
The key's randomart image is:
+---[RSA 2048]---+
|      .o=*+      |
|      .o.o       |
|      . . o.     |
|      S         |
|      . .       |
|      E . . . .  |
|      . o .+ooo  |
|      . o.o=+    |
+-----+
juna@debian:~/.ssh$
```

- Переходим в скрытую папку домашнего каталога .ssh
- Смотрим ее содержимое (ls)
- Генерим ключи по протоколу RSA (ssh-keygen -t rsa -f repository\_securiry\_key)

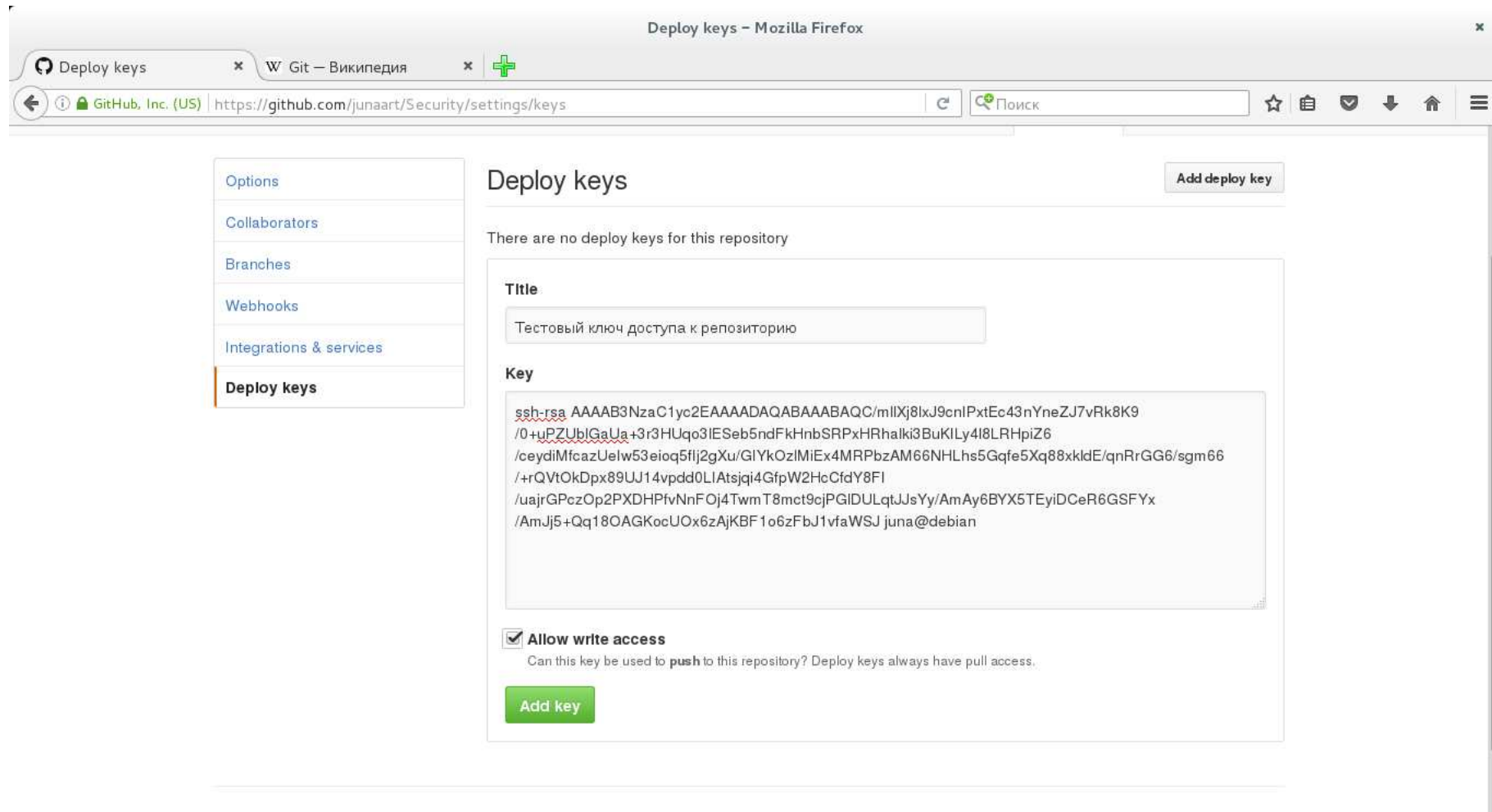
# Получаем содержимое публичного ключа



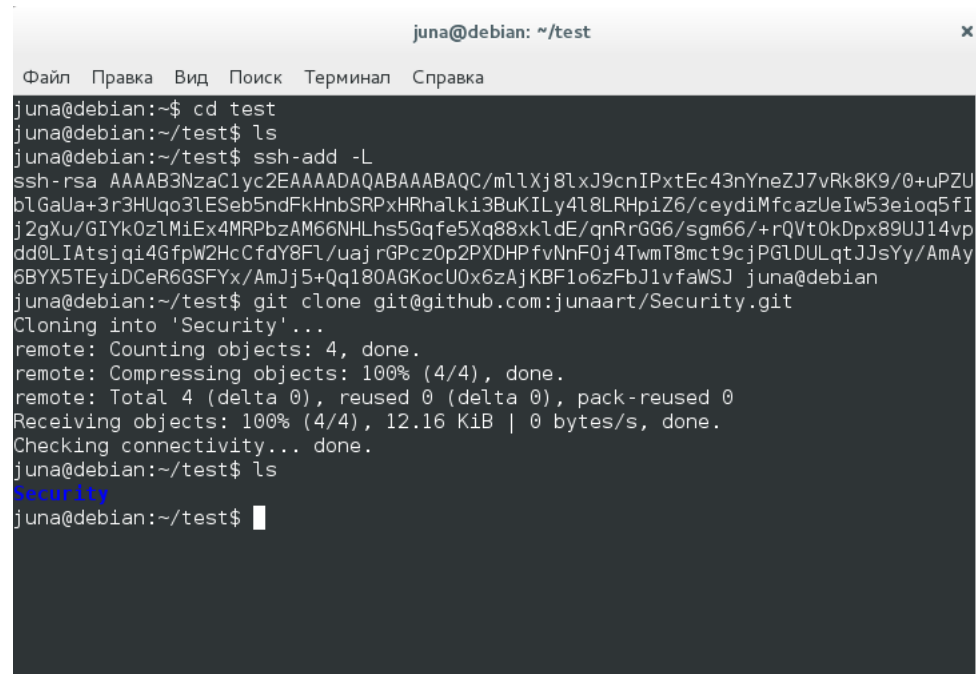
```
juna@debian: ~/.ssh
Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~/.ssh$ ls
known_hosts  repository_securiry_key  repository_securiry_key.pub
juna@debian:~/.ssh$ cat repository_securiry_key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCMl1Xj8lxJ9cnIPxtEc43nYneZJ7vRk8K9/0+uPZU
blGaUa+3r3HUqo3LESeb5ndFkHnbSRPxHRhalki3BuKILy4l8LRHpiZ6/ceydiMfcazUeIw53eioq5fI
j2gXu/GIYk0zlMiEx4MRPbzAM66NHLhs5Gqfe5Xq88xklDE/qnRrGG6/sgm66/+rQVt0kDpx89UJ14vp
dd0LIAtsjqi4GfpW2HcCfdY8FL/uajrGPcz0p2PXDHPfvNnF0j4TwmT8mct9cjPGLDULqtJJsYy/AmAy
6BYX5TEyiDCeR6GSFYx/AmJj5+Qq180AGKocU0x6zAjKBF1o6zFbJlvfaWSJ juna@debian
juna@debian:~/.ssh$
```

- Смотрим содержимое каталога .ssh (ls)
- Смотрим ее содержимое (cat repository\_securiry\_key.pub)

# Добавляем публичный ключик на github



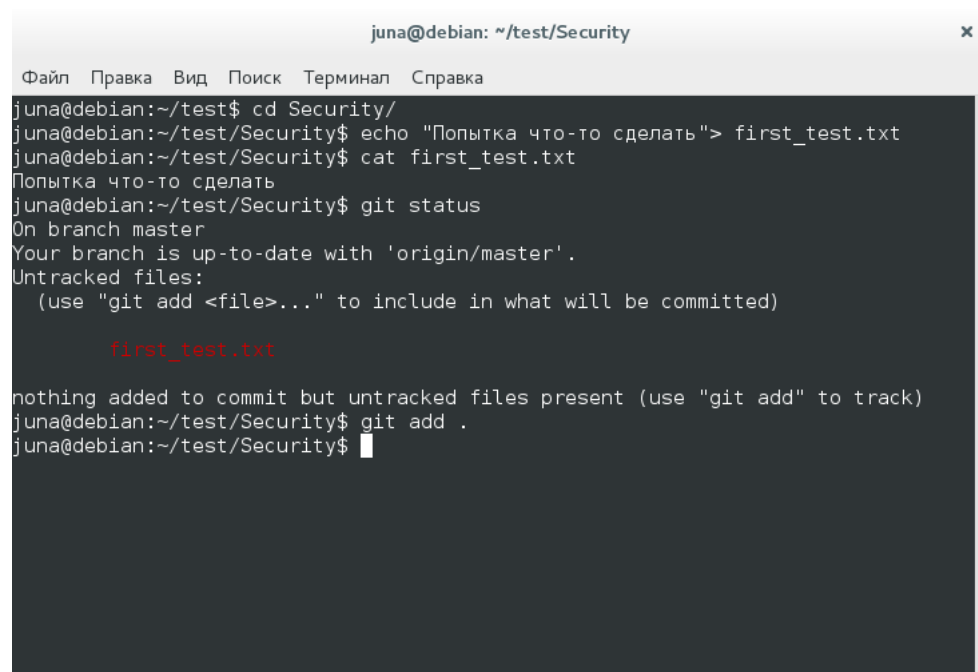
# Клонируем с github по ssh с помощью созданного ключа



```
juna@debian: ~/test
Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~$ cd test
juna@debian:~/test$ ls
juna@debian:~/test$ ssh-add -L
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAmlXj8lxJ9cnIPxtEc43nYneZJ7vRk8K9/0+uPZU
b1GaUa+3r3HUqo3lESeb5ndFkHnbSRPxHRhalki3BuKILy4l8LRHpiZ6/ceydiMfczUeIw53eioq5fI
j2gXu/GIYk0zlMiEx4MRPbzAM66NHLhs5Gqfe5Xq88xklde/qnRrGG6/sgm66/+rQVt0kDpx89UJ14vp
dd0LIAtsjqi4Gfpw2HcCfdY8FL/uajrGPczOp2PXDHPfvNnF0j4TwmT8mct9cjPGLDULqtJJsYy/AmAy
6BYX5TEyiDCeR6GSFYx/AmJj5+Qq180AGKocU0x6zAjKBFlo6zFbJ1vfawSJ juna@debian
juna@debian:~/test$ git clone git@github.com:junaart/Security.git
Cloning into 'Security'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), 12.16 KiB | 0 bytes/s, done.
Checking connectivity... done.
juna@debian:~/test$ ls
Security
juna@debian:~/test$
```

- Переходим в папку test (cd test)
- Смотрим ее содержимое (ls)
- Смотрим, объявлен ли публичный ключ к использованию (ssh-add -L)
- Клонировем репозиторию (git clone git@github.com:junaart/Security.git)
- Смотрим, как изменилось содержимое директории test (ls)

# Создаем в локальной копии репозитория новый файл



```
juna@debian: ~/test/Security
Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~/test$ cd Security/
juna@debian:~/test/Security$ echo "Попытка что-то сделать"> first_test.txt
juna@debian:~/test/Security$ cat first_test.txt
Попытка что-то сделать
juna@debian:~/test/Security$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        first_test.txt

nothing added to commit but untracked files present (use "git add" to track)
juna@debian:~/test/Security$ git add .
juna@debian:~/test/Security$
```

- Переходим в папку Security (cd Security/)
- Печатаем в файл first\_test.txt строку (echo Попытка что-то сделать> first\_test.txt)
- Смотрим содержимое созданного файла (cat first\_test.txt)
- Смотрим статус git (git status)
- Фиксируем изменения в git (git add .)

# Добавляем commit

```
juna@debian:~/test/Security$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   first_test.txt

juna@debian:~/test/Security$ git commit -m "Add file first_test.txt"
[master 36aabdb] Add file first_test.txt
Committer: juna <juna@debian>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 first_test.txt
```

# Пушим изменения на github

```
juna@debian:~/test/Security$ git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 352 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:junaart/Security.git
   b0eb818..36aabdb master -> master
juna@debian:~/test/Security$
```

# Смотрим изменения на github

The screenshot shows a web browser window with the title "junaart/Security: Основы информационной безопасности - Mozilla Firefox". The address bar displays "https://github.com/junaart/Security". The repository page for "junaart / Security" is shown, with tabs for "Code", "Issues", "Pull requests", "Projects", "Wiki", "Pulse", "Graphs", and "Settings". The repository has 2 commits, 1 branch, 0 releases, 1 contributor, and is licensed under GPL-3.0. The commit history table is as follows:

Commit	Message	Time
juna	Add file first_test.txt	Latest commit 36aabdb 6 minutes ago
LICENSE	Initial commit	2 hours ago
README.md	Initial commit	2 hours ago
first_test.txt	Add file first_test.txt	6 minutes ago

The README file is titled "Security". A link "https://github.com/pulls" is visible in the bottom left corner of the screenshot.



# Удаляем файл, добавляем новый

```
juna@debian: ~/test/Security

Файл  Правка  Вид  Поиск  Терминал  Справка
juna@debian:~/test/Security$ rm -f first_test.txt
juna@debian:~/test/Security$ ls
LICENSE  README.md
juna@debian:~/test/Security$ cp ~/Vigener.jpg ./
juna@debian:~/test/Security$ ls
LICENSE  README.md  Vigener.jpg
juna@debian:~/test/Security$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    first_test.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Vigener.jpg

no changes added to commit (use "git add" and/or "git commit -a")
juna@debian:~/test/Security$ git add .
juna@debian:~/test/Security$ git commit -m "Add table Vigener and delete first_test.txt"
[master f27c2ff] Add table Vigener and delete first_test.txt
Committer: juna <juna@debian>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 1 deletion(-)
```

# Наблюдаем изменения на github

junaart/Security: Основы информационной безопасности – Mozilla Firefox

GitHub, Inc. (US) <https://github.com/junaart/Security> Поиск

This repository Search Pull requests Issues Gist

junaart / Security Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Основы информационной безопасности Edit

New Add topics

3 commits 1 branch 0 releases 1 contributor GPL-3.0

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

juna	Add table Vigenere and delete first_test.txt	Latest commit f27c2ff 4 minutes ago
LICENSE	Initial commit	2 hours ago
README.md	Initial commit	2 hours ago
Vigenere.jpg	Add table Vigenere and delete first_test.txt	4 minutes ago

README.md

Security

<https://github.com/junaart/Security/projects>

# Удаляем репозиторий на github

