# **Что такое SSH. Генерирация SSH-ключа**

# **Что такое SSH**

Когда компьютеры обмениваются данными в сети, они следуют **сетевым протоколам** (англ. *network protocols*) — правилам обмена данными между компьютерами.

Один из наиболее распространённых сетевых протоколов — **SSH** (от англ. ***S****ecure* ***Sh****ell Protocol*). Он обеспечивает безопасный обмен данными в сети. С помощью этого протокола можно получать данные с удалённого компьютера или отправлять их на него. Трафик шифруется, поэтому протокол безопасен.

SSH использует пару ключей для обеспечения безопасности — публичный и приватный:

* **Приватный ключ** (англ. *private key*) хранится только на вашем компьютере и не должен передаваться кому-либо ещё. Он используется для расшифровки данных.
* **Публичный ключ** (англ. *public key*) доступен всем и используется для шифрования данных. Они могут быть расшифрованы парным приватным ключом.

Только вы можете расшифровать данные с помощью приватного ключа, но любой владелец публичного ключа может их для вас зашифровать. Эти два ключа связаны и образуют **SSH-пару**.

### **Проверка наличия SSH-ключа**

Прежде чем генерировать SSH-ключи, убедитесь, что у вас их ещё нет. По умолчанию директория с SSH-ключами находится в домашней директории пользователя. Перейдите в неё.



Обычно SSH-ключи находятся в директории .ssh/. Проверить наличие этой директории и файлов в ней можно с помощью следующей команды.



Если папка пустая или её нет, всё в порядке.

Если есть файлы с похожими названиями, SSH-ключи уже создавались:

* id\_dsa.pub;
* id\_ecdsa.pub;
* id\_ed25519.pub;
* id\_rsa.pub.

Если вы не создавали эти файлы, удалите их все.

### **Инструкция по генерации SSH-ключа**

Для генерации SSH-пары можно использовать программу ssh-keygen. Откройте терминал и введите следующую команду.



Используйте электронную почту, к которой привязан ваш GitHub-аккаунт.

Если вы видите сообщение об ошибке, то, скорее всего, ваша система не поддерживает алгоритм шифрования ed25519. Ничего страшного: используйте другой алгоритм.



После ввода отобразится такое сообщение.



Укажите место хранения ключей. Простой вариант — сделать домашний каталог пользователя путём по умолчанию. Для этого нажмите Enter.

**MacOS**



**Windows**



Теперь в указанной директории появится пара ключей.

Программа запросит **кодовую фразу** (англ. *passphrase*) для доступа к SSH-ключу. Вы можете оставить поле пустым. Для этого нажмите Enter, а затем ещё раз Enter для подтверждения.



Готово! Теперь осталось проверить, что ключи действительно сгенерировались. Для этого вызовите эту команду.

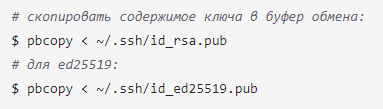


На экране должны появиться два файла — один с расширением .pub, другой — без. Файл в .pub — публичный, им можно делиться с веб-сайтами или коллегами. Файл без расширения .pub — приватный. Ни в коем случае не передавайте его никому!

Вся последовательность действий в консоли показана на скриншоте ниже.



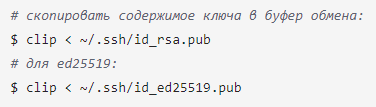
**MacOS**



Здесь используется команда pbcopy — она копирует поток данных в буфер обмена. Запись pbcopy < ~/.ssh/id\_rsa.pub означает: «Скопируй в буфер обмена всё содержимое файла ~/.ssh/id\_rsa.pub».

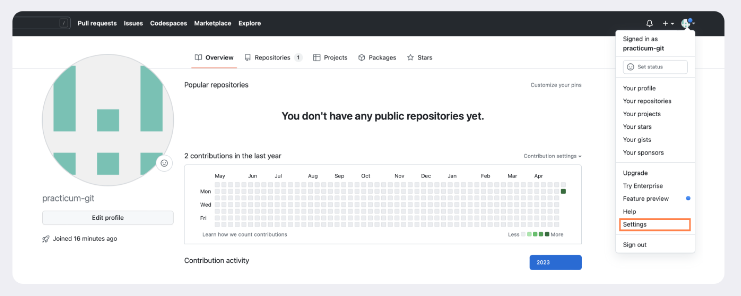
В качестве альтернативы вы можете распечатать файл на экран с помощью cat ~/.ssh/id\_rsa.pub и скопировать его вручную.

**Windows**

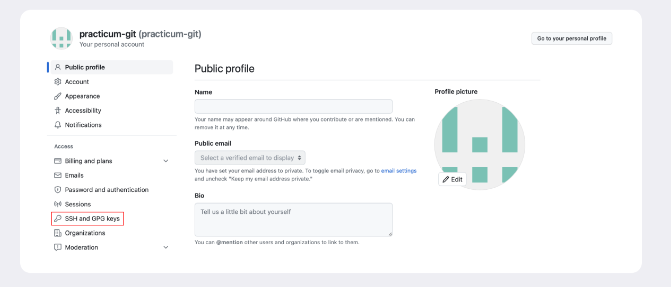


Если clip не сработает, выведите содержимое файла с помощью cat ~/.ssh/id\_rsa.pub или cat ~/.ssh/id\_ed25519.pub и скопируйте вывод в буфер обмена из консоли.

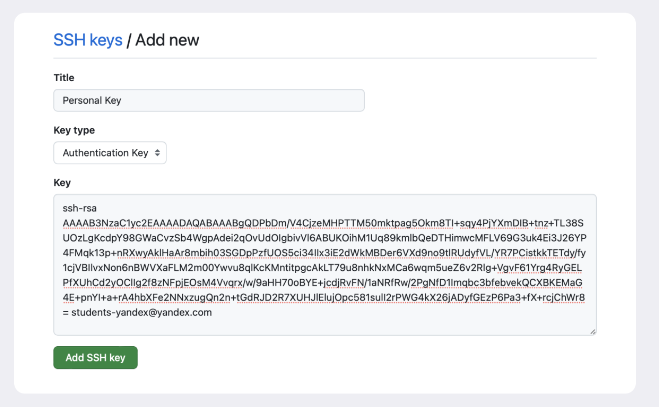
* Перейдите на GitHub и выберите пункт **Settings** (англ. «настройки») в меню аккаунта.



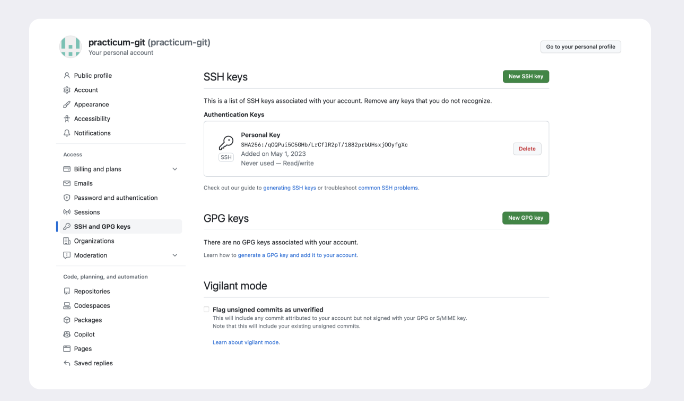
* В меню слева нажмите на пункт **SSH and GPG keys**.



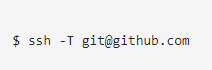
* В открывшейся вкладке выберите **New SSH key** (англ. «новый SSH-ключ»).
* В поле **Title** (англ. «заголовок») напишите название ключа. Например, **Personal key** (англ. «личный ключ»).
* В поле **Key type** (англ. «тип ключа») должно быть **Authentication Key** (англ. «ключ аутентификации»).
* В поле **Key** скопируйте ваш ключ из буфера обмена.



* Нажмите на кнопку **Add SSH key** (англ. «добавить SSH-ключ»).



* Проверьте правильность ключа с помощью следующей команды.



Если это первый раз, когда вы используете Git, чтобы поделиться проектом на GitHub, появится похожее предупреждение.

“The authenticity of host 'github.com (140.82.121.4)' can't be established. ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU. This key is not known by any other names. Are you sure you want to continue connecting (yes/no/[fingerprint])?”

Это предупреждение сообщает, что вы никогда не соединялись с сервером GitHub. Поэтому Git не может гарантировать, что сервер является тем, за кого он себя выдаёт.

Для подтверждения подлинности сервер генерирует и публикует ключи SHA256. Вы можете проверить ключи GitHub [по этой ссылке](https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/githubs-ssh-key-fingerprints). Если ключ в предупреждении совпадает с тем, что вы видите на сайте, значит, сервер является действительным. Введите yes, чтобы продолжить. Вы увидите приветствие на экране.

“Hi %ВАШ\_АККАУНТ%! You've successfully authenticated, but GitHub does not provide shell access.”