



# 10 examples to generate SSH key in Linux (ssh-keygen)

WRITTEN BY - ADMIN

Updated On November 11, 2023

## Topics we will cover [ [hide](#) ]

### [Overview on ssh-keygen](#)

- [1. Generate ssh key without any arguments](#)
  - [2. Define Key Type](#)
  - [3. Define Bit size](#)
  - [4. Assign Passphrase](#)
  - [5. Измените кодовую фразу закрытого ключа](#)
  - [6. Создайте ключи с пользовательским именем файла](#)
  - [7. Добавьте пользовательский комментарий к ключу.](#)
  - [8. Измените комментарий к ключу.](#)
  - [9. Хеширование содержимого файла known hosts](#)
  - [10. Удалите ключи для имени хоста из файла known hosts](#)
- [Заключение](#)

## [Ссылки](#)

Мы используем инструмент `ssh-keygen` для генерации SSH-ключей, которые используются для аутентификации на основе открытого ключа с помощью SSH. На момент написания этой статьи с помощью SSH возможно [6 различных типов методов аутентификации](#). Но [аутентификация с открытым ключом](#) является одним из наиболее часто используемых методов аутентификации, используемых в производственной среде.

Для использования аутентификации на основе открытого ключа вам потребуется пара открытого и закрытого ключей.

- Содержимое открытого ключа должно быть добавлено в `authorized_keys` файл сервера
- Закрытый ключ будет храниться на клиенте

Таким образом, когда клиент пытается установить безопасное соединение, он будет использовать эту комбинацию пары закрытого и открытого ключей для установления соединения

## Обзор `ssh-keygen`

- `ssh-keygen` это утилита, предоставляемая `openssh` грм, которая должна быть установлена по умолчанию во всех дистрибутивах Linux.

- `ssh-keygen` генерирует, управляет и преобразует ключи аутентификации для ssh версии 2.0 и выше
- Этот инструмент поддерживает различные аргументы, которые могут быть использованы для создания ключей в соответствии с требованиями
- Если вы хотите использовать SSH с аутентификацией по открытому ключу, используйте это один раз для создания ключа аутентификации в `~/.ssh/id_dsa` , `~/.ssh/id_ecdsa` , `~/.ssh/id_ed25519` или `~/.ssh/id_rsa`
- Если вы забыли кодовую фразу, то сбросить ее невозможно, и вы должны воссоздать новую кодовую фразу и поместить их пары ключей в соответствующие места, чтобы повторно активировать аутентификацию с открытым ключом

**ТАКЖЕ ЧИТАЙТЕ**

Добавьте временную метку в подробные журналы SSH  
[Стандартный вывод и файл журнала]

Давайте рассмотрим `ssh-keygen` инструмент для генерации различных типов пар ключей в Linux

## 1. Сгенерируйте ssh-ключ без каких-либо аргументов

- Вы можете выполнить `ssh-keygen` без каких-либо аргументов, который сгенерирует пары ключей по умолчанию, используя алгоритм **RSA**
- Инструмент запросит местоположение для хранения пар ключей RSA.
- Расположение по умолчанию будет находиться внутри домашней папки пользователя в разделе `.ssh` т.е. `~/.ssh`
- Инструмент создаст `~/.ssh` , если каталог еще не существует
- Синтаксис именования по умолчанию, используемый для закрытого ключа RSA, будет `id_rsa` , а открытый ключ будет `id_rsa.pub`
- Далее указана ключевая фраза, вы можете просто нажать **ENTER** , чтобы создать пару ключей без ключевой фразы

bash

```
# ssh-keygen
```

Фрагмент из моего терминала

```
[root@rhel-8 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): 
Enter passphrase (empty for no passphrase): 
Enter same passphrase again: 
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:MnL5GQB8r4sfVNo8Q7+zgAJkzivBPhizwvD/eUcz8GY root@rhel-8.example.com
The key's randomart image is:
+---[RSA 3072]-----+
|  ..                      |
| ...                     |
|  o  ...o                |
| .=      Oo.             |
| =.+ . B.So.             |
| =* o +. =E.             |
| =+o ..o.+oo            |
| .....o...o             |
|  .o+.  ..               |
+---[SHA256]-----+
[root@rhel-8 ~]# ls -l ~/.ssh/
total 16
-rw----- 1 root root 405 Nov 21  2019 authorized_keys
-rw----- 1 root root 2610 May 23 18:27 id_rsa
-rw-r--r-- 1 root root 577 May 23 18:27 id_rsa.pub
-rw-r--r-- 1 root root 568 Mar 27 13:09 known_hosts
```

Генерировать SSH-ключ без каких-либо аргументов

## 2. Определите тип ключа

- По умолчанию `ssh-keygen` будет создан ключ типа RSA
- Вы можете создать ключ с помощью `dsa` , `ecdsa` , `ed25519` или `rsa` типа
- Используйте `-t <key>` аргумент для определения типа ключа
- В этом примере я создаю пару ключей типа `ED25519`

bash



```
# ssh-keygen -t ed25519
```

Фрагмент из моего терминала

```
[root@rhel-8 ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): 
Enter passphrase (empty for no passphrase): 
Enter same passphrase again: 
Your identification has been saved in /root/.ssh/id_ed25519.
Your public key has been saved in /root/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:IzF2S4bAY8KdjJ5pTG2TQSSqUiRCZzDJcX35yM0zixs root@rhel-8.example.com
The key's randomart image is:
+--[ED25519 256]--+
|==+o.o .
|.==. + o
| o + =
| o . * * *
|o = # S +
|. + * E *
| * + +
| . . o .
| .
+-----[SHA256]-----+
[root@rhel-8 ~]#
[root@rhel-8 ~]# ls -l ~/.ssh/
total 20
-rw----- 1 root root 405 Nov 21  2019 authorized_keys
-rw----- 1 root root 419 May 23 18:40 id_ed25519
-rw-r--r-- 1 root root 105 May 23 18:40 id_ed25519.pub
-rw----- 1 root root 664 May 23 18:37 known_hosts
-rw-r--r-- 1 root root 568 Mar 27 13:09 known_hosts.old
```

Определение типа ключа

### 3. Определите размер бита

По умолчанию ssh-keygen генерирует SSH-ключ с **2048** размером бита. Вы также можете указать количество битов, которые будут использоваться для ключей, используя

**-b <bit\_size>**

В этом примере я сгенерирую ключи размером 4096 бит

bash



```
# ssh-keygen -b 4096
```

Фрагмент из моего терминала

```
[root@rhel-8 ~]#  
[root@rhel-8 ~]# ssh-keygen -b 4096  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /root/.ssh/id_rsa.  
Your public key has been saved in /root/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:oKVSaxbDFihHwJ34AyzIY3Y2E/9ku2moE0IrU132iX8 root@rhel-8.example.com  
The key's randomart image is:  
+---[RSA 4096]---+  
|*o+o+  
|+XoO..o  
|+o*.O+o= .  
| ..=.B=.+  
|..o B +S  
|oo = . + E  
|o . . . + .  
| . . .  
| ..  
+---[SHA256]-----+  
[root@rhel-8 ~]#
```

Определение размера бита

## 4. Назначьте кодовую фразу

По умолчанию `ssh-keygen` будет запрашиваться кодовая фраза перед созданием пар ключей. Но мы также можем назначить кодовую фразу с помощью `-P <your_password>`

bash



```
# ssh-keygen -P "MyPassw0rd"
```

Фрагмент из моего терминала



```
[root@rhel-8 ~]#  
[root@rhel-8 ~]# ssh-keygen -P "MyPassw0rd"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
Your identification has been saved in /root/.ssh/id_rsa.  
Your public key has been saved in /root/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:UCRbtXl9DNzIJsnXDvRfPSlGc1H143bU1Ho/qpzoVNQ root@rhel-8.example.com  
The key's randomart image is:  
+---[RSA 3072]---+  
|    .+. . . Bo*O|  
|    =   o*.X=O|  
|    o   o..E.BX|  
|    .   . . . +oB|  
|    S    .   ++|  
|    .   .   .o|  
|    .   .   .|  
|    . o  .   |  
|    .o +.   |  
+-----[SHA256]-----+
```

Assign passphrase

ТАКЖЕ ЧИТАЙТЕ

5 команд для копирования файла с одного сервера на другой в Linux или Unix

## 5. Измените кодовую фразу закрытого ключа

- Вы также можете изменить существующую кодовую фразу вашего закрытого ключа
- Используйте `ssh-keygen` with `-p`, который запросит у вас местоположение вашего файла закрытого ключа
- Затем укажите существующую кодовую фразу вашего закрытого ключа
- Если предоставленная кодовая фраза верна, вы получите приглашение назначить новую кодовую фразу вашему существующему закрытому ключу

bash



```
# ssh-keygen -p
```

Введите файл, в котором находится ключ (/root/.ssh/id\_rsa):

Введите старую кодовую фразу:

Ключ имеет комментарий 'root@rhel-8.example.com'

Введите новую кодовую фразу (пустую, поскольку кодовой фразы нет):

Введите ту же кодовую фразу еще раз:

Ваша идентификация была сохранена с новой ключевой фразой.

### ПОДСКАЗКА:

Для автоматизации этого шага вы можете использовать `ssh-keygen` с `-f` для предоставления файла закрытого ключа, `-P` для определения вашей старой ключевой фразы и `-N` для определения новой ключевой фразы

bash



```
# ssh-keygen -p -f ~/.ssh/id_rsa -P "old_password" -N "new_password"
```

Ключ имеет комментарий 'root@rhel-8.example.com'

Ваша идентификация была сохранена с новой ключевой фразой.

## 6. Создайте ключи с пользовательским именем файла

- По умолчанию `ssh-keygen` создает закрытый ключ с именем `id_rsa` и открытым ключом в виде `id_rsa.pub`
- Мы также можем создавать ключи с пользовательским именем файла, используя `-f <file_name>`
- Это позволит создать и сохранить сертификаты в текущем местоположении, из которого вы запускаете инструмент `ssh-keygen`
- В этом примере моим закрытым ключом будет `my-own-rsa-key`, а открытым ключом будет `my-own-rsa-key.pub`

bash



```
# ssh-keygen -f my-own-rsa-key
```

*Фрагмент из моего терминала*



```
[root@rhel-8 ~]#  
[root@rhel-8 ~]# ssh-keygen -f my-own-rsa-key  
Generating public/private rsa key pair.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in my-own-rsa-key.  
Your public key has been saved in my-own-rsa-key.pub.  
The key fingerprint is:  
SHA256:m0Abnwxq0r46LxMe+5lbmib0mVekzqI3Iy4SItLgybk root@rhel-8.example.com  
The key's randomart image is:  
+---[RSA 3072]-----+  
|  
| . . o * ..  
|oo+ + o So  
|==++ . .o.  
|+o.=.. =+.  
|.EB ooX=+  
| ..X****  
+---[SHA256]-----+  
[root@rhel-8 ~]#  
[root@rhel-8 ~]# ls -l my-own-rsa-key*  
-rw----- 1 root root 2610 May 23 19:05 my-own-rsa-key  
-rw-r--r-- 1 root root 577 May 23 19:05 my-own-rsa-key.pub  
[root@rhel-8 ~]#
```

Сгенерируйте SSH-ключ и назначьте имя файла

## 7. Добавьте пользовательский комментарий к ключу

Вы также можете добавить пользовательский комментарий к своему закрытому ключу для дополнительной идентификации. Используйте `-C <comment>` для генерации ключей с вашим пользовательским комментарием

bash



```
# ssh-keygen -C "Это для server1.example.com"
```

Мы можем использовать `-l` для печати отпечатка пальца и комментария к закрытому ключу

bash



```
# ssh-keygen -l  
Введите файл, в котором находится ключ (/root/.ssh/id_rsa):  
3072 SHA256: JxBpArCDsIVME0HDtQG7FqFQefaS9ommeohVoEmg39g Это для server1.example.com
```

ТАКЖЕ ЧИТАЙТЕ    Расширенный пакет Mgmt с командой dpkg [Шпаргалка]

## 8. Измените комментарий к ключу

You can also change the existing comment of your private key using `-c` argument

bash



```
# ssh-keygen -c
Введите файл, в котором находится ключ (/root/.ssh/id_rsa):
Теперь ключ имеет комментарий "Это для server1.example.com"
Введите новый комментарий: Это для rhel-8.example.com
Комментарий в вашем ключевом файле был изменен.
```

Проверьте новый комментарий к вашему закрытому ключу

bash



```
# ssh-keygen -l
Введите файл, в котором находится ключ (/root/.ssh/id_rsa):
3072 SHA256: JxBpArCDsIVME0HdtQG7FqFQefaS9ommeohVoEmg39g Это для rhel-8.example.com (F
```

## 9. Хэширование содержимого файла `known_hosts`

- Каждый раз, когда вы выполняете подключение по SSH к другому серверу, отпечаток SSH для безопасного подключения добавляется в файл клиента `~/.ssh/known_hosts`
- Используется для проверки подлинности SSH-соединения
- Содержимое `known_hosts` файла будет в таком формате

bash



```
# cat .ssh/known_hosts
10.10.10.10 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDGIXmWjH3Ly6ty9O3hYeg8p/ld7Isl65Da
192.168.43.22 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABl
```

- Любой злоумышленник может использовать эту информацию для получения данных отпечатков пальцев человека `hostname` .
- Вы можете использовать `ssh-keygen` для хэширования `hostname` записей в `known_hosts` файле, используя `-H` аргумент

- Эта опция не изменит существующий хэш `hostname` и, следовательно, безопасна для использования с файлами, в которых сочетаются хэшированные и нехэшированные имена.
- Это создаст резервную копию файла с `.old` расширением в том же месте

bash



```
# ssh-keygen -H
```

Далее проверьте содержимое файла `known_hosts`

bash



```
# cat .ssh/known_hosts
|1|DnQfHwXX0E78Kqd9sM+jhKICLhM=|A7gki0vPIUajFLR0xD1jIxE6rGM= ssh-rsa AAAAB3NzaC1yc2EAAAADAQ
|1|RK+RdFcebk+2EK81Rs16e9Im6Hk=|b7QKZly3lm6mBEzIvsLDps4x44I= ecdsa-sha2-nistp256 AAAAE2VjZHNh
```

Как вы видите, теперь имя хоста невозможно понять, поскольку оно хэшировано. В том же месте также создается файл резервной копии

bash



```
# ls -l ~/.ssh/known_hosts*
-rw----- 1 root root 664 May 23 18:37 /root/.ssh/known_hosts
-rw-r--r-- 1 root root 568 Mar 27 13:09 /root/.ssh/known_hosts.old
```

ТАКЖЕ ЧИТАЙТЕ    Запись логов ssh и sshd strace [Шаг за шагом]

## 10. Удалите ключи для имени хоста из файла `known_hosts`

- Каждый раз, когда вы выполняете SSH, ключ RSA для SSH-соединения для соответствующего `hostname` сохраняется внутри `~/.ssh/known_hosts` файла
- Но если вы переустановите целевой сервер и попытаетесь выполнить SSH, возможно, SSH может завершиться сбоем из-за неправильного совпадения отпечатка пальца
- Таким образом, вы можете либо вручную выполнить поиск и удалить отпечаток RSA вашего сервера из `known_hosts` файла, либо использовать `ssh-keygen` для

выполнения этой работы

- Используется `-R <hostname>` для автоматического поиска и удаления всех записей отпечатков пальцев и ключей RSA для предоставленного файла `hostname` from `known_hosts`
- Например, для удаления всех ключей, связанных с `192.168.43.22` хостом, из `known_hosts` файла

bash



```
Найден# Хост 192.168.43.22: обновлена строка 2 /root/.ssh/known_hosts.  
# ssh-keygen -R 192.168.43.22 Исходное содержимое сохранено как /root/.ssh/known_hosts.old
```

Что дальше?


[Настройте аутентификацию по открытому ключу для использования SSH с парольной фразой или без нее в Linux](#)

## Заключение

`ssh-keygen` это очень обширный инструмент, который может делать гораздо больше, чем генерация SSH-ключей. Он также поддерживает подписание ключей для создания сертификатов, которые могут использоваться для аутентификации пользователя или хоста. В этой статье мы узнали о различных аргументах, которые можно использовать для генерации SSH-ключей для аутентификации по открытому ключу с помощью SSH

Вы также можете объединить все аргументы из этого руководства, чтобы автоматизировать процесс. Наконец, я надеюсь, что шаги из статьи для более подробного понимания инструмента ssh-keygen с различными примерами в Linux были полезны. Итак, дайте мне знать о ваших предложениях и отзывах, используя раздел комментариев.

## Ссылки

Я использовал приведенные ниже внешние ссылки для этого учебного руководства [справочная страница для ssh-keygen](#) 

### ТАКЖЕ ЧИТАЙТЕ

Как ограничить доступ пользователя root к файлу и каталогу или изменить их в Linux

📁 [Категории](#), [SSH](#)

Не можете найти то, что ищете? Позвольте нам помочь вам.

Введите свой запрос ниже, и мы мгновенно предоставим результаты с учетом ваших потребностей.

Если мои статьи о **GoLinuxCloud** помогли вам, пожалуйста, подумайте о том, чтобы угостить меня кофе в знак признательности.



Для любых других отзывов или вопросов вы можете отправить письмо по адресу [admin@www.golinuxcloud.com](mailto:admin@www.golinuxcloud.com)

Спасибо за вашу поддержку!!

[Sitemap](#) [Privacy Policy](#) [Disclaimer](#) [Contact](#)

Copyright © 2024 | Hosted On Rocket.net