

# **ASTRA LINUX GUIDE**

**SSH**

## SSH

Одним из важнейших инструментов в работе системного администратора является SSH.

SSH, или Secure Shell, — это протокол, используемый для безопасного входа на удаленные

### Базовый синтаксис

Чтобы подключиться к удаленной системе с помощью SSH, мы будем использовать команду ssh. В самом базовом виде команда имеет следующую форму:

```
ssh remote_host
```

remote\_host в этом примере является IP-адресом или доменным именем узла, к которому вы пытаетесь подключиться.

Эта команда предполагает, что ваше имя пользователя на удаленной системе совпадает с именем пользователя в локальной системе.

Если ваше локальное имя пользователя отличается от имени пользователя в удаленной системе, вы можете задать его, используя следующий синтаксис:

```
ssh remote_username@remote_host
```

После подключения к серверу вам, возможно, потребуется подтвердить вашу личность с помощью пароля. Позже мы рассмотрим, как сгенерировать ключи, которые можно использовать вместо паролей.

Чтобы завершить сеанс ssh и вернуться в сеанс локальной оболочки, введите следующую команду:

```
exit
```

### Как работает SSH?

SSH выполняет подключение клиентской программы к серверу ssh с именем sshd.

В предыдущем разделе команда ssh использовалась для вызова клиентской программы. Сервер ssh уже запущен на удаленном хосте remote\_host, который мы указали.

На вашем сервере должен быть запущен сервер sshd. Если это не так, вам может потребоваться подключение к серверу через веб-консоль или локальную последовательную консоль.

Процесс запуска сервера ssh зависит от дистрибутива Linux, который вы используете.

В Ubuntu вы можете запустить сервер ssh с помощью следующей команды:

```
sudo systemctl start ssh
```

Эта команда должна запускать сервер sshd, после чего вы сможете выполнять удаленный вход.

## Настройка SSH

При изменении конфигурации SSH вы меняете настройки сервера sshd.

В Ubuntu основной файл конфигурации sshd находится в каталоге /etc/ssh/sshd\_config.

Выполните резервное копирование текущей версии этого файла перед началом редактирования:

```
sudo cp /etc/ssh/sshd_config{,.bak}
```

Откройте файл в текстовом редакторе:

```
sudo nano /etc/ssh/sshd_config
```

Скорее всего, вы захотите оставить большинство опций в этом файле без изменений. Однако существует несколько настроек, на которые вам стоит обратить особое внимание:

```
/etc/ssh/sshd_config
```

```
Port 22
```

Объявление порта указывает, подключения к какому порту будет отслеживать сервер sshd. По умолчанию используется порт 22. Вам, скорее всего, не придется изменять данную настройку, если только у вас нет конкретных причин для иного решения. Если вы решите изменить порт, позже мы покажем, как подключиться к новому порту.

```
/etc/ssh/sshd_config
```

```
HostKey /etc/ssh/ssh_host_rsa_key  
HostKey /etc/ssh/ssh_host_dsa_key  
HostKey /etc/ssh/ssh_host_ecdsa_key
```

В объявлениях ключей хоста указывается, где нужно искать глобальные ключи хоста. Мы обсудим, что такое ключ хоста, позже.

```
/etc/ssh/sshd_config
```

```
SyslogFacility AUTH  
LogLevel INFO
```

Эти две строки указывают на уровень логирования, который необходимо использовать.

Если вы сталкиваетесь с проблемами при работе с SSH, увеличение объема логируемых данных может быть хорошим решением, которое поможет понять, в чем заключается проблема.

```
/etc/ssh/sshd_config
```

```
LoginGraceTime 120  
PermitRootLogin yes  
StrictModes yes
```

Эти параметры определяют некоторые данные для входа в систему.

Опция `LoginGraceTime` определяет количество секунд, в течение которых следует сохранять подключение при отсутствии успешных попыток входа в систему.

Возможно, вам может быть полезным задать для этого параметра чуть большее количество времени, чем то, которое вам обычно требуется для входа.

`PermitRootLogin` определяет, разрешен ли вход с помощью пользователя с правами `root`.

В большинстве случаев необходимо изменить значение на `no`, если вы создали учетную запись пользователя, которая имеет доступ к высокому уровню привилегий (через `su` или `sudo`) и может использоваться для входа в систему через `ssh`.

`strictModes` — это защитник, который будет препятствовать попыткам входа, если файлы аутентификации доступны для чтения всем.

Он позволяет предотвратить попытки входа в систему, когда файлы конфигурации не находятся в безопасном состоянии.

```
/etc/ssh/sshd_config
```

```
X11Forwarding yes  
X11DisplayOffset 10
```

Эти параметры используются для настройки такой возможности, как X11 Forwarding. X11 Forwarding позволяет просматривать графический пользовательский интерфейс (GUI) удаленной системы на локальной системе.

Эта функция должна быть активирована на сервере и передана клиенту SSH во время подключения с помощью опции `-X`.

После внесения изменений сохраните и закройте файл, введя `CTRL+X, Y`, а затем нажмите `ENTER`.

Если вы внесли изменения в какие-либо настройки в файле `/etc/ssh/sshd_config`, необходимо перезапустить ваш сервер `sshd`, чтобы изменения вступили в силу:

```
sudo systemctl reload ssh
```

Вы должны тщательно протестировать ваши изменения, чтобы убедиться, что все работает так, как вы ожидаете.

Вы можете использовать несколько активных сеансов во время внесения изменений. Это позволит вам вернуться к первоначальной конфигурации, если это потребуется.

## Выполнение входа через SSH с использованием ключей

Хотя возможность входа в удаленную систему с помощью паролей может быть полезна, гораздо лучшей идеей будет настройка аутентификации с помощью ключей.

### Как работает аутентификация с помощью ключей?

Аутентификация с помощью ключей реализуется путем создания пары ключей: приватного ключа и публичного ключа.

Приватный ключ располагается на клиентском компьютере, этот ключ защищен и хранится в секрете.

Публичный ключ может передаваться любому лицу или размещаться на сервере, доступ к которому вы хотите получить.

При попытке подключения с использованием пары ключей сервер будет использовать публичный ключ для создания сообщения для клиентского компьютера, которое может быть прочитано только с помощью приватного ключа.

Затем клиентский компьютер отправляет соответствующий ответ обратно серверу, после чего сервер будет знать, что клиент не является поддельным.

Весь этот процесс выполняется в автоматическом режиме после того, как вы настроите ключи.

## Создание ключей SSH

Ключи SSH необходимо генерировать на компьютере, откуда вы хотите войти в систему. Как правило, это ваш локальный компьютер.

Введите следующую команду в командной строке:

```
ssh-keygen -t rsa
```

Нажмите ENTER, чтобы принять используемые по умолчанию значения. Ваши ключи будут сгенерированы в файлах `~/.ssh/id_rsa.pub` и `~/.ssh/id_rsa`.

Перейдите в каталог `.ssh` с помощью следующей команды:

```
cd ~/.ssh
```

Просмотрите данные о разрешениях для файлов:

```
ls -l
```

### Output

```
-rw-r--r-- 1 demo demo 807 Sep 9 22:15 authorized_keys
-rw----- 1 demo demo 1679 Sep 9 23:13 id_rsa
-rw-r--r-- 1 demo demo 396 Sep 9 23:13 id_rsa.pub
```

Как вы можете видеть, файл `id_rsa` доступен для чтения и записи только владельцу. Именно такие разрешения позволяют сохранить его в секрете.

В то же время файл `id_rsa.pub` может использоваться совместно и имеет соответствующие разрешения для данной деятельности.

## Как передать ваш публичный ключ на сервер

Если в настоящее время вы используете доступ к серверу с помощью пароля, вы можете скопировать ваш публичный ключ на сервер, воспользовавшись данной командой:

```
ssh-copy-id remote_host
```

В результате будет создан сеанс SSH. Когда вы введете пароль, ваш публичный ключ будет скопирован в файл авторизованных ключей сервера, что позволит не использовать пароль при входе в следующий раз.

## Опции для клиентской стороны

Существует ряд опциональных флагов, которые вы можете использовать при подключении через SSH.

Некоторые из них могут быть необходимы при наличии определенных настроек конфигурации sshd на удаленном хосте.

Например, если вы изменили номер порта в конфигурации sshd, вам потребуется указать этот порт на клиентской стороне с помощью следующей команды:

```
ssh -p port_number remote_host
```

Если вы хотите выполнить отдельную команду на удаленной системе, вы можете указать ее после имени хоста следующим образом:

```
ssh remote_host command_to_run
```

В результате будет установлено подключение к удаленному компьютеру, а после успешной аутентификации команда будет выполнена.

Как уже отмечалось ранее, если функция X11 forwarding активирована на обоих компьютерах, вы можете получить доступ к данному функционалу, воспользовавшись следующей командой:

```
ssh -X remote_host
```

При наличии соответствующих инструментов на вашем компьютере программы GUI, которые вы используете на удаленной системе, теперь будут открываться в отдельном окне на локальной системе.

## Отключение аутентификации по паролю

Если вы создали ключи SSH, вы можете повысить уровень безопасности вашего сервера, отключив аутентификацию только по паролю. Помимо консоли единственным способом входа на ваш сервер будет использование приватного ключа, который используется в паре с публичным ключом, установленным на сервере.

Предупреждение: перед выполнением этих действий необходимо убедиться, что публичный ключ установлен на сервере. В противном случае вы заблокируете доступ к серверу!

Откройте файл конфигурации sshd, воспользовавшись пользователем root или пользователем с привилегиями sudo:

```
sudo nano /etc/ssh/sshd_config
```

Найдите строку Password Authentication и раскомментируйте ее, удалив символ # в начале строки. Теперь вы можете указать значение no:

```
/etc/ssh/sshd_config
```

```
PasswordAuthentication no
```

Вы должны также изменить значения двух других настроек (если вы не вносили изменения в этот файл ранее) — PubkeyAuthentication и ChallengeResponseAuthentication. Значения устанавливаются по умолчанию и выглядят следующим образом:

```
/etc/ssh/sshd_config
```

```
PubkeyAuthentication yes  
ChallengeResponseAuthentication no
```

После внесения изменений сохраните и закройте файл.

Теперь нужно перезапустить демон SSH:

```
sudo systemctl reload ssh
```

Теперь аутентификация по паролю должна быть отключена, а ваш сервер должен быть доступен только с помощью аутентификации по ключу SSH.