**SSH для новичков**



Программистов, сисадминов, devops'ов, тестировщиков часто объединяет использование некоторых общих технологий. Например, возможность иметь удалённый доступ к серверу пригодилось бы любому из указанной выше анекдотичной компании. Поэтому в этой статье мы рассмотрим основы работы с протоколом SSH на базовом уровня (без генерации ключей, подписей, мам, пап и кредитов). "Продвинутое" использование посмотрим уже отдельно, в следующий раз. Материал подготовил Макс, автор YouTube-канала [PyLounge](https://www.youtube.com/channel/UCru5FZQN_Xa0tKfrBqUIcng). Поехали!

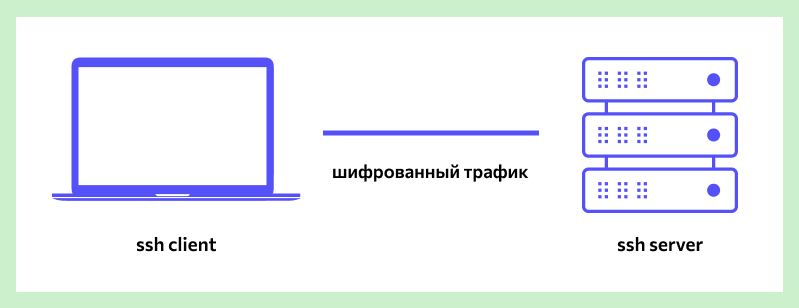
**Введение**

**SSH (Secure SHell - защищенная оболочка)** — сетевой **протокол** прикладного уровня, предназначенный для безопасного удаленного доступа к **различным системам (Linux, Windows, Mac)**.

Данный протокол шифрует всю передаваемую информацию по сети. По умолчанию, используется **22-й порт**. В основном он нужен для **удаленного управления** данными пользователя на сервере, запуска служебных команд, работы в консольном режиме с базами данных.

Эта служба была создана в качестве замены не зашифрованному [Telnet](https://ru.wikipedia.org/wiki/Telnet" \t "_blank) и использует криптографические техники, чтобы обеспечить, что всё сообщение между сервером и пользователем было зашифровано.

Чтобы установить SSH-соединение, необходимы два компонента: *SSH-сервер* и *SSH-клиент*. Сервер прослушивает определенный порт (по умолчанию это порт 22) и при успешной аутентификации дает доступ пользователю. Все команды, которые используются на SSH-клиенте, отправляются через защищенный канал связи на SSH-сервер, на котором они выполняются и откуда отправляют результат работы обратно клиенту.



**SSH-сервер**

Устанавливается на управляемой операционной системе и принимает входящие подключения от клиентских машин, проверяя соединение одним из способов:

* по IP-адресу клиента, что не слишком безопасно из-за риска подмены;
* по публичному ключу клиента и имени пользователя. Нужно создать приватный (закрытый) и публичный (открытый) ключ. Зашифровав информацию одним ключом, можно расшифровать ее только другим;
* по паролю клиента, который передается в зашифрованном виде. Это один из наиболее распространенных вариантов. Вводить его нужно при каждом подключении.

Платные и бесплатные SSH-серверы есть для всех распространенных ОС:

* BSD — OpenSSH;
* Linux — dropbear, lsh-server, openssh-server;
* Windows — freeSSHd, copssh, WinSSHD, OpenSSH и т. д.

**SSH-клиент**

Используется для входа на удаленный сервер и выполнения команд. Через клиент выполняется управление удаленным компьютером.

SSH-клиенты разработаны для всех ОС. Имеют платные и бесплатные версии:

* Linux/BSD — openssh-client, putty, ssh, Vinagre;
* Windows — PuTTY, SecureCRT, ShellGuard;
* Android — connectBot.

Встроенный SSH-клиент Windows, основанный на OpenSSH, предустановлен в ОС, начиная с Windows 10 1809.

**Установка OpenSSH на Ubuntu**

Установка с помощью менеджера пакетов *apt-get*:

sudo apt-get install openssh-server

**Проверка SELinux**

Security Enhanced Linux, или SELinux – это усовершенствованный механизм контроля доступа, встроенный в большинство современных дистрибутивов Linux. Первоначально он был разработан Агентством национальной безопасности США **для защиты компьютерных систем от вторжения злоумышленников и взлома**. SELinux также может помешать работе с ssh. Поэтому необходимо убедиться, что он отключен, либо [отключить его](https://support.kaspersky.com/KWTS/6.1/ru-RU/165993.htm).

Устанавливаем доп. инструменты для работы с сетью и настройками безопасности:

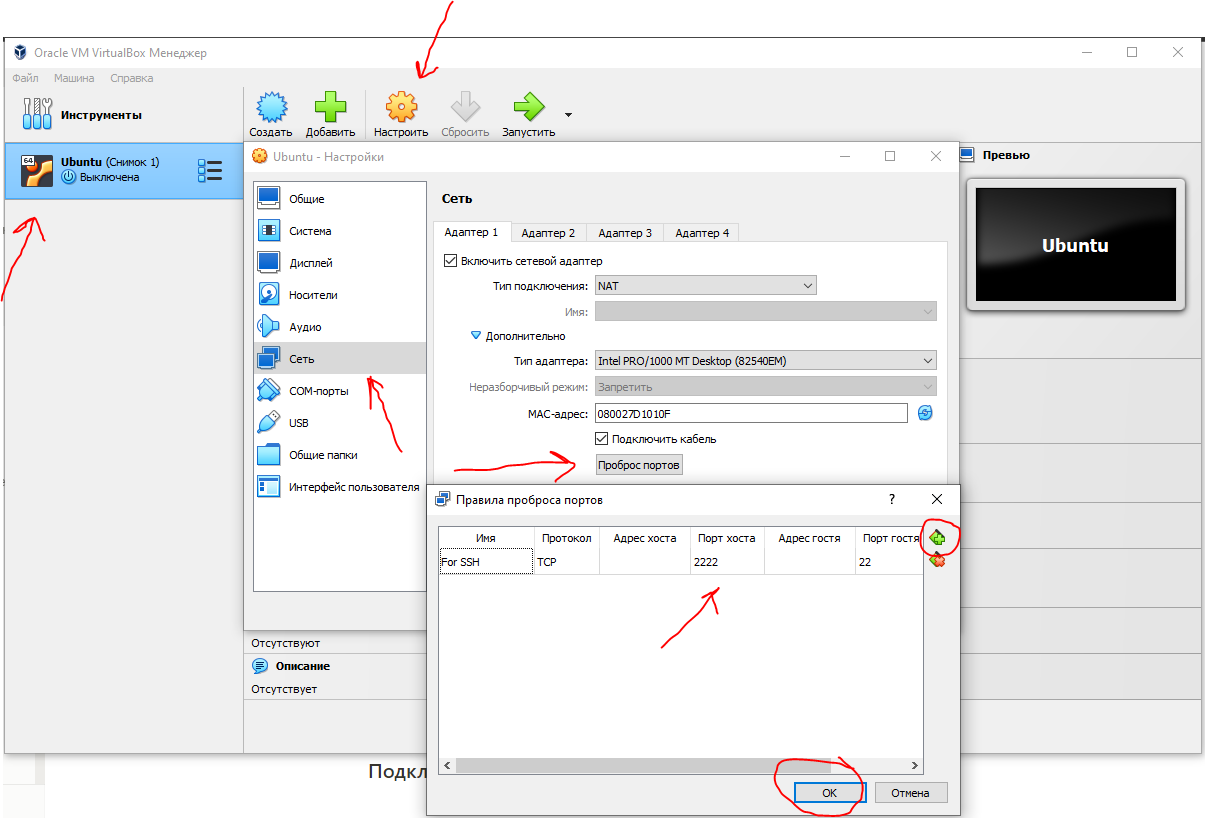
sudo apt-get install policycoreutils  
sudo apt-get install net-tools

Проверяем статус SELinux. Состояние должно быть disable.

sestatus

**Подключение к виртуальной машине с типом подключения NAT**

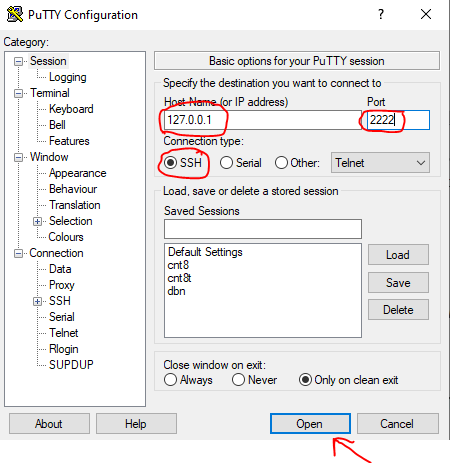
Если виртуальная машина использует тип сетевого подключения [NAT](https://wiki.merionet.ru/seti/13/nat-na-palcax-chto-eto/), то первоначально необходимо настроить проброс портов. Пробросим порт хоста 2222 (или любой свободный) на гостевой порт виртуальной машины номер 22 (порт ssh по умолчанию).



После этого мы можем подключиться из Windows к нашей виртуальной машине.

**Подключение по ssh через PuTTY**

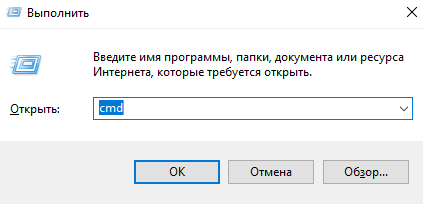
Т.к. в качестве сетевого подключения у виртуальной машины используется NAT, при подключении мы будем указывать петлевой адрес [localhost](http://localhost) (127.0.0.1) c портом, на который был назначен проброс.



После этого вводим имя пользователя **(пользователь должен существовать в linux!!!)** и пароль от этого пользователя.

**Подключение по ssh через Командную строку Windows (cmd)**

Запускаем командную строку сочетанием клавиш **Win + R .** В открывшемся окне указываем **cmd.** Нажимаем **OK.**



В общем видео подключение по ssh имеет вид:

ssh имя\_пользователя@ip\_адрес\_компьютера\_к\_которому\_подключаемся -p порт

Если для подключения используется стандартный порт ssh - 22, то порт можно не указывать.

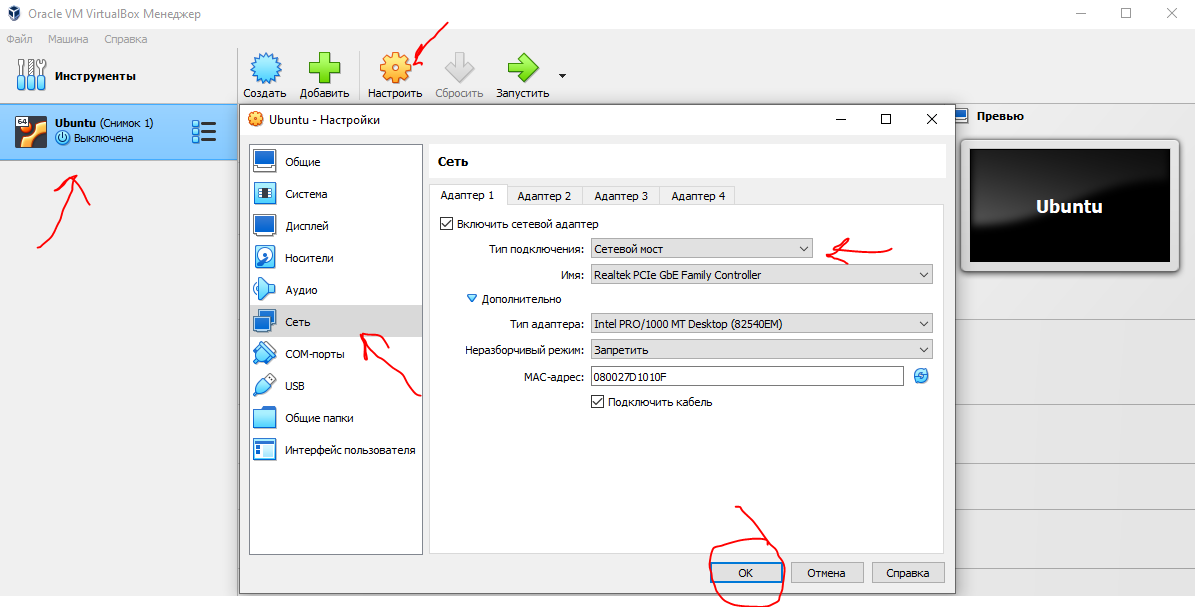
В открывшемся окне вводим команду:

ssh student@127.0.0.1 -p 2222

Вводим пароль пользователя. Всё. Вы великолепны!

**Подключение к виртуальной машине с типом подключение Сетевой мост**

С типом подключения Сетевой мост необходимо выдать статический ip для Linux системы. Дальнейшее подключение будет уже по этому, выданному ip, а не через localhost.



**Установка статического ip**

Начиная с выпуска Ubuntu 17.10 **NetPlan** теперь это инструмент настройки сети по умолчанию для управления настройками сети, заменяющий файл конфигурации /etc/network/interfaces, который использовался в предыдущих версиях.

Файлы конфигурации для Netplan находятся в каталоге /etc/netplan и написаны на языке *YAML*. В этом каталоге вы найдете файлы конфигурации YAML.

Откроем конфигурационный файл:

sudo nano /etc/netplan/\*.yaml

Отредактируем конфигурационный файл:

network:   
 version: 2  
 renderer: NetworkManager  
 ethernets:  
 enp0s3:  
 dhcp4: false  
 addresses: [10.40.0.224/24]  
 gateway4: 10.40.0.1  
 nameservers:  
 addresses: [10.2.0.8, 10.2.0.3]

Здесь:

* ethernets - указывает на проводное Ethernet подключение;
* enp0s3 - имя сетевого интерфейса (можно узнать командой **ifconfig**);
* dhcp4 - автоматическая выдача ipv4-адреса с помощью службы dhcp (для ipv6 указывать dhcp6);
* addresses - ip-адрес и маска сети, которые назначаются компьютеру;
* gateway4 - шлюз по умолчанию;
* nameservers - список dns-серверов.

Посмотреть настройки можно с хостовой машины на Windows с помощью команды **ipconfig /all**.

Чтобы применить изменения и перезагрузить сеть, вводим команду:

sudo netplan apply

**Подключение по ssh**

Запускаем командную строку сочетанием клавиш **Win + R .** В открывшемся окне указываем **cmd.** Нажимаем **OK.**

В открывшемся окне вводим команду:

ssh student@10.40.0.224

Вводим пароль пользователя. Всё. Вы великолепны (2)!

**Заключение**

Несмотря на высокую защиту протокола, использование стандартного порта значительно повышает риск взлома сервера. Например, это можно сделать при помощи **brute-force** (атака полным перебором). Порт 22 — открытый, поэтому злоумышленнику нужно только подобрать логин и пароль к серверу. Это сложно, но возможно. Чтобы максимально ограничить доступ к серверу для злоумышленников (настроить брандмауэр, iptables), можно сменить SSH-порты.