**Лабораторная работа №1: настройка удаленного доступа к серверу с использованием SSH.**

**Цель работы: ННаучиться подключаться по протоколу ssh к удаленному компьютеру. Уметь изменять конфигурацию Open SSH сервера под себя. Научиться выполнять дистанционно команды по ssh на удаленных компьютерах. Научиться устанавливать удаленное соединение по ключу без ввода логина и пароля. Ознакомиться с пользовательскими файлами конфигурации ssh. Использовать SFTP.**

**Теоретическая часть**

SSH — это Secure Shell (безопасная оболочка); это специальный протокол для передачи данных в безопасном режиме. Он очень часто применяется для удаленного управления компьютерами и устройствами по сети.

Настройки sshd находятся в файле /etc/ssh/sshd\_config.  
Порт по умолчанию используется tcp:22

**Как устанавливается сессия SSH?**  
Есть несколько шагов, которые нужно пройти, чтобы начать SSH-сеанс между компьютерами.

* Сначала нужно обеспечить безопасный способ обмена сообщениями между компьютерами, то есть настроить зашифрованный канал.
* Далее нужно проверить целостность данных, отправляемых клиентом.
* После этого проверяется подлинность клиента.
* После этих трёх шагов мы можем безопасно общаться с удалённым компьютером, делиться секретными данными, а также проверить, есть ли у клиента разрешение на доступ к хосту. Каждый из разделов ниже будет более подробно описывать эти действия.

**Настройка зашифрованного канала**

Вся информация, отправляемая с использованием SSH, зашифрована. Обе стороны должны знать и понимать способ шифрования. Для шифрования передаваемых данных используется симметричное шифрование. Суть данного подхода заключается в том, что оба компьютера имеют одинаковый ключ шифрования, который называется «симметричный ключ». Симметричное шифрование работает очень хорошо, но только до тех пор, пока сторонние не имеют доступа к ключу.

Один компьютер может создать ключ и отправить в виде сообщения через интернет. Но сообщение ещё не будет зашифровано, поэтому любой, кто перехватит его, сразу же сможет расшифровать все следующие сообщения.  
Решение этой проблемы состоит в использовании протокола обмена ключами Диффи-Хеллмана. Оба компьютера создают свой закрытый и открытый ключ. Вместе они образуют пару ключей. Компьютеры делятся своими открытыми ключами друг с другом через интернет. Используя свой закрытый и чужой открытый ключ, стороны могут независимо сгенерировать одинаковый симметричный ключ.

**Верификация**  
Следующий этап процесса установки сеанса SSH заключается в проверке того, что данные не были подделаны во время их передачи и что другой компьютер действительно является тем, за кого себя выдаёт.

Для верификации используют хеш-функцию. Это математическая функция, которая принимает входные данные и создаёт строку фиксированного размера.

Важной особенностью этой функции является то, что практически невозможно определить входные данные, зная лишь результат её работы.

После того как клиент и хост сгенерировали свои симметричные ключи, клиент использует хеш-функцию для генерации HMAC, что означает «код аутентификации сообщений, использующий хеширование». Клиент отправит этот HMAC на сервер для верификации.  
Функция хеширования использует:

* симметричный ключ клиента,
* порядковый номер пакета,
* содержимое сообщения (зашифрованное).

Когда хост получает HMAC, он может использовать ту же самую хеш-функцию с этими тремя компонентами:

* собственный (идентичный клиентскому) симметричный ключ;
* порядковый номер пакета;
* зашифрованное сообщение.

Если сформированный хеш совпадает с HMAC, полученным от клиента, то мы можем быть уверены, что подключаемый компьютер — это компьютер с симметричным ключом, потому что только хост и клиент знают симметричный ключ, а другие компьютеры — нет.

Прелесть этого подхода в том, что мы не просто проверили личность клиента и убедились, что данные не были подделаны, но мы сделали это без передачи какой-либо секретной информации.

**Аутентификация**

Даже если мы используем симметричные ключи для безопасного общения, мы не знаем, имеет ли подключающийся компьютер разрешение на доступ к содержимому хоста. Для того чтобы проверить это, необходимо произвести аутентификацию.

Многие используют аутентификацию по паролю. Клиент отправляет хосту зашифрованное сообщение, содержащее пароль. Хост его расшифровывает и ищет пароль в базе данных, чтобы удостовериться, есть ли у клиента разрешение на доступ. Использование пароля для аутентификации допустимо, но имеет свои недостатки, так как необходимо хранить все пароли на сервере.

Более безопасной является аутентификация по сертификату. Сформировав сертификат, клиент единожды вводит пароль для доступа к серверу и отправляет ему открытую часть сертификата. В дальнейшем ввод пароля не требуется. Этот подход считается более безопасным, чем просто использование пароля, поскольку не подразумевает хранение секрета пользователя на хосте.

**SFTP.**

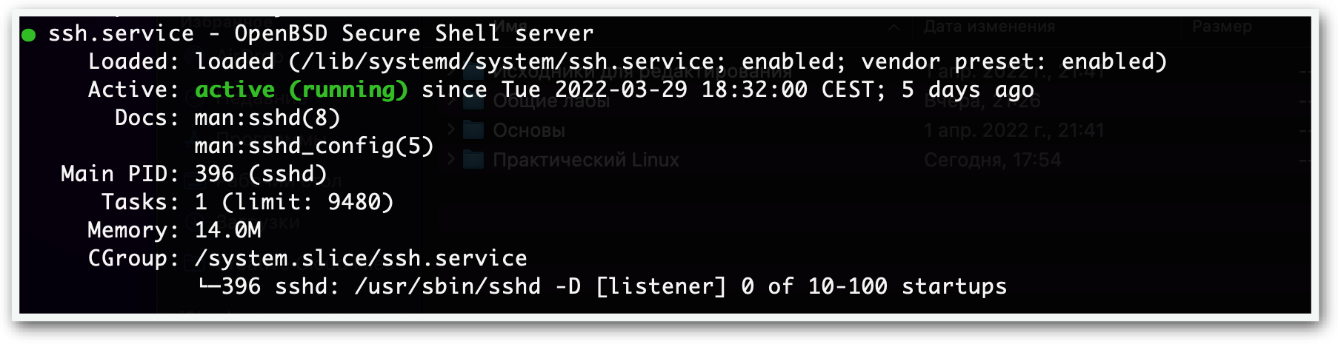
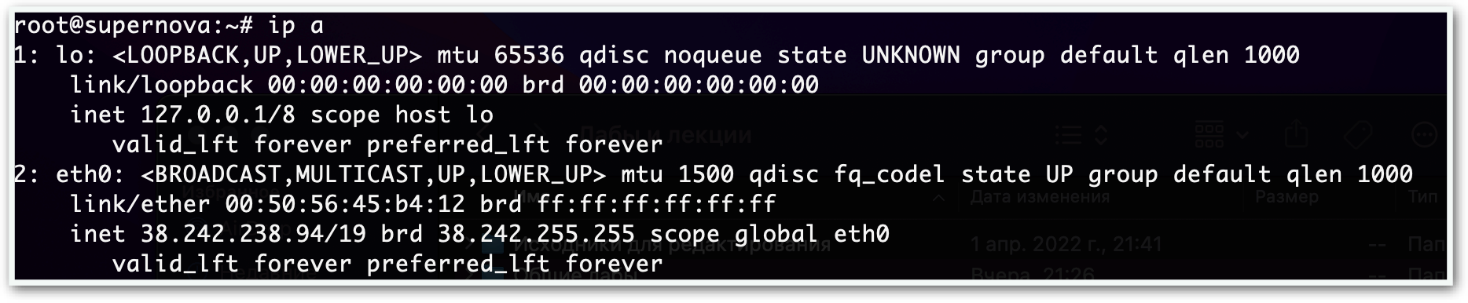
SFTP доступен по умолчанию без дополнительной настройки на всех серверах, имеющих доступ по SSH. Как следует из названия протокола, это безопасный способ передачи файлов на сервер с использованием зашифрованного SSH-соединения.

В стандартной конфигурации сервер SSH предоставляет доступ к передаче файлов и к оболочке терминала всем системным пользователям с учетной записью.

SFTP это «надстройка над ssh»

**Практическая часть**

**Ознакомление с ssh сервером.**

1. Перейдите в командную строку виртуальной машины. Обновите информацию о пакетах выполнив команду apt update.
2. Установите Open SSH если он не установлен.
3. Используйте команду systemctl status sshd чтобы посмотреть состояние службы ssh. 
4. **Если пароль для root не задан.** Перейдите в root пользователя sudo su и используйте команду passwd чтобы задать пароль.
5. Смотрим IP адрес командой ip a.
6. C основного компьютера подключаемся по ssh используя команду **ssh root@ip\_удаленной\_машины**. Нажимаем Y. Вводим пароль.
7. Почитайте файл конфигурации. Измените порт по умолчанию. Смена порта связана с тем, что многочисленные сетевые сканеры постоянно пытаются соединиться с 22-м портом и как минимум получить доступ путем перебора логинов/паролей из своей базы. Даже если у вас и отключена парольная аутентификация — эти попытки сильно засоряют журналы и (в большом количестве) могут негативно повлиять на скорость работы ssh сервера. Для лучшей защиты настраивают fail2ban для защиты ssh от перебора паролей. Сделайте так что бы было нельзя подключиться по root пользователю.
8. Перезагрузите **sshd**. Для этого воспользуйтесь **systemctl**. Также проверьте состояние службы.
9. Отобразите в отчете результат изменений. Чтобы узнать как подключиться с другим портом почитайте man по команде ssh или воспользуйтесь поисковиком.
10. Верните порт по умолчанию на стандартный порт **ssh**. Перезагрузите службу.

**Подключение по ключу.**

1. Выполните команду **ssh-keygen** для создания ключа на основной машине. Утилита спросит вас, куда сохранить создаваемый закрытый ключ. По умолчанию (если нажмете ENTER), ключ будет сохранен в папку **.ssh** вашего домашнего каталога. Следующим шагом вам будет предложено установить пароль на закрытый ключ. Пароль будет запрашиваться при каждом использовании этого закрытого ключа. Установка пароля необязательна (можно просто нажать ENTER), но необходимо учесть, что любой человек, к которому попадет не защищенный паролем закрытый ключ, сможет получить доступ к вашему серверу. Установив (или не установив) пароль, вы завершите процедуру генерации SSH-ключей. Если в процессе не указывали иного, создадутся следующие файлы:

* Закрытый ключ: ~/.ssh/id\_rsa
* Открытый ключ: ~/.ssh/id\_rsa.pub

1. Теперь вам необходимо скопировать открытый ключ на сервер. Для этого выполните следующую команду: **ssh-copy-id username@ipserver**. Вам понадобится ввести пароль этого пользователя. После этого содержимое локального файла ~/.ssh/id\_rsa.pub будет добавлено в файл ~/.ssh/authorized\_keys домашнего каталога соответствующего пользователя на сервере.
2. Теперь вы можете авторизоваться на сервере с помощью SSH-ключей с помощью команды: ssh username@ipserver. Без ввода пароля.
3. После того как проверили авторизацию по ssh ключу отключите авторизацию по логину и паролю. Для этого вам понадобится отредактировать файл **/etc/ssh/sshd\_config** на вашей виртуальной машине. В файле необходимо найти директиву **PasswordAuthentication**, раскомментировать ее, если она закомментирована, и установить для нее значение «**no**»

**Удаленный запуск команд.**

1. Для удаленного запуска команд используют следующий синтаксис **ssh USER@HOST ‘command’**. Выполните удаленную установку какого либо пакета к примеру mc. Подобный способ выполнения удалённых команд может быть использоваться для создание скриптов.

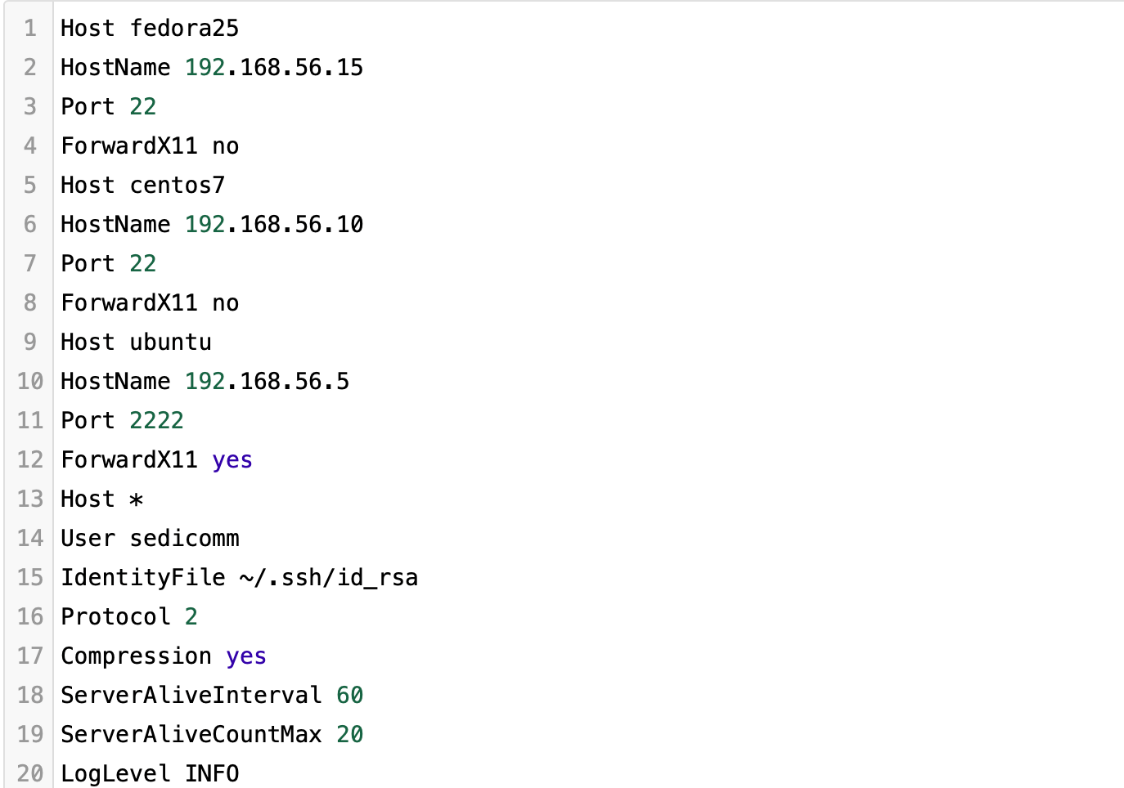
**Пользовательские файлы конфигурации ssh.**

1. Необходима 2-я машина с Linux. Сгенерируйте ssh ключ и передайте на 2 машину. Чтобы реализовать подключение по ключу.
2. Пользовательский файл обычно не создается по умолчанию, поэтому вам нужно создать его с разрешениями на чтение и запись только для пользователя. Воспользуйтесь следующими командами:

touch ~/.ssh/config

chmod 0700 ~/.ssh/config

1. Вышеупомянутый файл содержит разделы, определенные спецификациями хостов, и раздел применяется только к хостам, которые соответствуют одному из шаблонов, заданных в спецификации. Пример такого файла:

  
Подробное объяснение приведенных выше параметров конфигурации ssh:

* + **HostName** — определяет имя реального хоста для входа в систему, альтернативно, вы можете использовать числовые IP-адреса, это также разрешено (как в командной строке, так и в спецификациях HostName).
  + **User** — указывает, что пользователь должен войти в систему.
  + **Port** — устанавливает номер порта для подключения на удаленном хосте, по умолчанию – 22. Используйте номер порта, настроенный в файле конфигурации sshd удаленного хоста.
  + **Protocol** — этот параметр определяет версии протокола, которые ssh должен поддерживать в порядке предпочтения. Обычными значениями являются «1» и «2», несколько версий должны быть разделены запятыми.
  + **IdentityFile** — определяет файл, с которого считывается идентификатор аутентификации пользователя DSA, Ed25519, RSA или ECDSA.
  + **ForwardX11** — определяет, будут ли соединения X11 автоматически перенаправляться по защищенному каналу и устанавливать DISPLAY. Он имеет два возможных значения «да» или «нет».
  + **Compression** — оно используется для установки сжатия во время удаленного соединения со значением «да». По умолчанию «нет».
  + **ServerAliveInterval** — устанавливает интервал таймаута в секундах, после которого, если никакой ответ (или данные) не был получен с сервера, ssh отправит сообщение через зашифрованный канал, чтобы запросить ответ от сервера. Значение по умолчанию равно 0, то есть сообщения не будут отправляться на сервер, если была определена опция BatchMode.
  + **ServerAliveCountMax** — устанавливает количество живых сообщений сервера, которые могут быть отправлены без получения ответа ssh с сервера.
  + **LogLevel** — определяет уровень детализации, который используется при регистрации сообщений из ssh. Допустимые значения включают: QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2 и DEBUG3. И значение по умолчанию – INFO.

**Настройте файл конфигурации для подключения к виртуальной машине 2. Хостнейм ваша фамилия**

1. Теперь вы сможете подключаться к серверу просто вписав hostname к примеру так **ssh centos7**. В отчете отобразите что у вас удалось подключиться по hostname. Такой способ удобен когда вам необходимо администрировать большое количество серверов. Вам не приходиться запоминать логины пароли порт для конкретного сервера а также IP адрес.

**Работа с SFTP**

1. Выполните подключение по sftp. sftp username@ipserver.
2. Оказавшись в командной строке sftp можно получить список доступных команд с помощью команды help.
3. Посмотрите текущей рабочий каталог на своем и удаленном пк.
4. Попробуйте изменить рабочею директорию на своем и удаленном пк.
5. Для того чтобы передать файл по sftp используйте команду put. Для передачи папки используйте ключ -r.
6. Для того чтобы скачать что то с удаленного сервера используйте команду get. Для загрузки директории используйте ключ -r.
7. Создайте и удалите директорию на удаленным пк (ВМ) в sftp.