**Лабораторная работа № 3. Подключение по SSH**

**Что такое SSH-ключ**

к сведению

Аббревиатура SSH означает Secure Shell, что дословно переводится как «безопасная оболочка». Если говорить точнее, SSH — это защищенный сетевой протокол с проверкой подлинности и шифрованием. Он используется для передачи файлов, управления сетью и удаленного доступа к операционной системе. Аббревиатура SSH также используется для описания набора инструментов, используемых для взаимодействия с одноименным протоколом.

Подключение через SSH можно просто охарактеризовать как подключение к командной строке удаленного узла. То есть любые команды, которые будут вводиться в терминал на основной машине будут работать так же, будто вы вводите их напрямую на удаленном узле, сидя с клавиатурой около него.

Так как SSH работает по системе «клиент-сервер», обязательное условие для работы этого протокола — наличие на удаленном узле демона SSH. Демон SSH — это ПО, которое прослушивает определенный сетевой порт и при прохождении аутентификации другим узлом создает необходимую среду для работы с ним.

В свою очередь на локальной машине должно быть установлено соответствующее ПО — SSH-клиент. Он взаимодействует с удаленным хостом и передает ему необходимые данные, которые нужны для прохождения аутентификации.

**Структура ключа**[**​**](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/#%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D0%BA%D0%BB%D1%8E%D1%87%D0%B0)

Можно сказать, что определение "SSH-ключ" составное, так как на самом деле это два ключа — **открытый** и **закрытый**. На узле A создаются и хранятся оба ключа, а на узел B передается только копия публичного SSH-ключа, что позволяет подключаться к узлу B с узла A.

Ключи могут быть сгенерированы с помощью различных алгоритмов, которые поддерживает текущая версия протокола SSH. Например, если использован тип шифрования RSA, то файлы будут именоваться следующим образом:

* id\_rsa — закрытый ключ,
* id\_rsa.pub — публичный (открытый) ключ.

В чем же разница открытого и закрытого ключа?

**Открытые и закрытые SSH-ключи**[**​**](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/#%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B5-%D0%B8-%D0%B7%D0%B0%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B5-ssh-%D0%BA%D0%BB%D1%8E%D1%87%D0%B8)

**Открытый** (он же **публичный**) ключ используется для шифрования данных при обращении к удаленному узлу. Проще говоря, это набор символов, при помощи которых мы шифруем информацию. Он доступен всем. Не стоит бояться того, что открытый ключ может попасть в чужие руки, так как наличие одного лишь публичного SSH-ключа не дает злоумышленнику никаких преимуществ. Открытый SSH-ключ хранится на удаленном узле.

**Закрытый** (**приватный**) SSH-ключ — это ключ к данным. Он расшифровывает сами сообщения. Хранится он на устройстве, которое будет подключаться к удаленному узлу (на котором находится открытый ключ). Приватный ключ ни в коем случае нельзя передавать в чужие руки, в том числе через мессенджеры или файлообменники, во избежание утечки информации и персональных данных. Также рекомендуем сразу установить пароль на закрытый ключ, чтобы обеспечить ему дополнительную защиту.

**Как работает SSH-авторизация?**

Давайте представим, что вы хотите подключиться к серверу с использованием SSH-ключа. Предварительно вы уже создали пару ключей и передали публичный ключ нам. Алгоритм взаимодействия будет следующим:

1. Вы должны изъявить свое желание подключиться к нам, то есть отправить запрос на подключение по TCP-порту.
2. В случае установки TCP-соединения мы обмениваемся информацией о версиях наших SSH-протоколов. С помощью этой информации можно понять, какую именно конфигурацию (версию протоколов и алгоритмы работы) использовать. Самостоятельно узнать версию OpenSSH можно с помощью команды ssh -V.
3. После согласования мы (сервер) направляем вам (клиенту) открытый ключ. Теперь уже вы решаете, доверять такому ключу или нет. В случае положительного ответа мы с вами генерируем сеансовый ключ, который будет использоваться для симметричного шифрования канала. Этот ключ существует, только пока существует канал (текущая сессия).
4. Теперь следует аутентифицировать вас. Для этого вы отсылаете нам свой открытый ключ. Мы в свою очередь проверяем его со своим списком открытых SSH-ключей. Если совпадение найдено, мы генерируем случайное число, шифруем его открытым ключом и отправляем обратно. Вы как клиент расшифровываете сообщение закрытым ключом и отправляете полученные данные нам. В случае совпадения присланного числа с первоначальным аутентификация признается успешной.



Поздравляем! Теперь вам открыт доступ на сервер.

**Создание SSH-ключа**

**Параметры утилиты ssh-keygen[​](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/" \l "%D0%BF%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D1%8B-%D1%83%D1%82%D0%B8%D0%BB%D0%B8%D1%82%D1%8B-ssh-keygen" \o "Прямая ссылка на Параметры утилиты ssh-keygen)**

В Linux для создания SSH-ключей используется команда ssh-keygen. Далее приведены наиболее популярные параметры для этой команды и их описание.

По умолчанию ssh-keygen использует в качестве параметра -t именно RSA, так как этот алгоритм обеспечивает наилучшую совместимость из всех, но требует большего размера ключа для обеспечения достаточной безопасности. Длина ключа по умолчанию составляет 3072 бит, но вы можете самостоятельно задать его размер от 1024 до 16384 бит с помощью опции -b команды ssh-keygen.

**-C comment[​](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/" \l "-c-comment" \o "Прямая ссылка на -c-comment)**

Создание нового комментария. Например, одной из самых известных команд является добавление комментария с информацией о том, кто, на какой машине и когда создал ключ:

ssh-keygen -C "$(whoami)@$(uname -n)-$(date -I)"

**-p**

Если указать данный параметр при вводе ssh-keygen, то вам будет предложено изменить старую секретную фразу на новую. Ввод команды будет выглядеть следующим образом:

ssh-keygen -p

Также вы можете задать все параметры для изменения секретной фразы сразу:

ssh-keygen -p [-P old\_passphrase] [-N new\_passphrase] [-f keyfile]

**-t type[​](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/" \l "-t-type" \o "Прямая ссылка на -t-type)**

Задает тип создаваемого ключа. Чтобы узнать какие типы доступны для вашей версии, введите:

ssh-keygen -t ?

**-v**

Подробный режим. ssh-keygen будет печатать отладочные сообщения о ходе выполнения. Несколько опций -v увеличивают степень подробности информации (максимум 3).

**Утилиты для создания SSH-ключа**

Далее мы создадим RSA ключ, добавим публичную часть на ВМ и подключимся с помощь него по SSH к ВМ. Сделаем это тремя способами:

* С помощью терминала и утилит Linux.
* С помощью утилит, предоставляемых PuTTY;
* С помощью SSH-агента PowerShell;

**Генерация SSH-ключей в Linux[​](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/" \l "%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F-ssh-%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%B9-%D0%B2-linux" \o "Прямая ссылка на Генерация SSH-ключей в Linux)**

Процесс создания SSH-ключей на базе Linux предельно прост. Вам необходимо лишь указать некоторые параметры, которые мы опишем далее.

Мы будем создавать ключ RSA. При вводе команды ssh-keygen опцию -t RSA можно не указывать явно, так как RSA является алгоритмом по умолчанию.

ssh-keygen

Generating public/private RSA key pair.  
Enter file *in* which to save the key (/root/.ssh/id\_rsa):

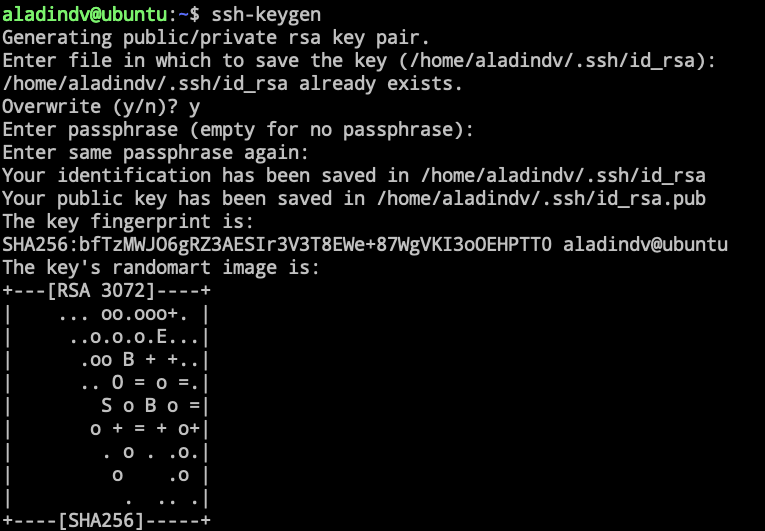
В последней строке вам предлагается выбрать путь, куда будут сохранены ключи. Если оставить поле пустым, будут созданы файлы с именами id\_rsa.pub и id\_rsa.

Enter passphrase (empty *for* no passphrase):

В данной строке предлагается создать кодовую фразу. Оставив строку пустой, дополнительной защиты не будет.

С одной стороны, не вводить кодовую фразу опасно. Если закрытый ключ попадет в руки злоумышленников, то у них может появиться доступ к серверу. В то же время, если вы введете кодовую фразу, в будущем это может доставить вам неудобства. Пароль придется вводить каждый раз при использовании SSH-подключения с этим ключом. Поэтому использование кодовой фразы остается на ваше усмотрение.

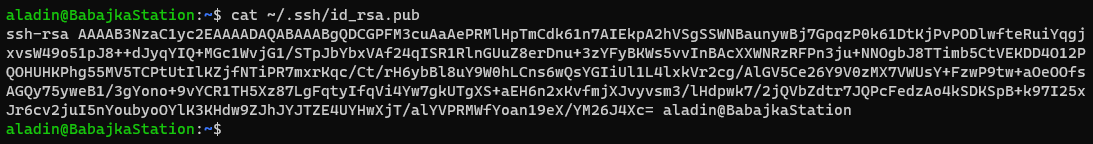
Как мы упоминали ранее, кодовую фразу можно будет установить (или заменить) самостоятельно, введя ssh-keygen -p, когда ключ уже будет создан:



Созданные файлы хранятся в директории /home/<имя пользователя>/.ssh/.

Посмотреть публичный ключ текущего пользователя можно с помощью команды:

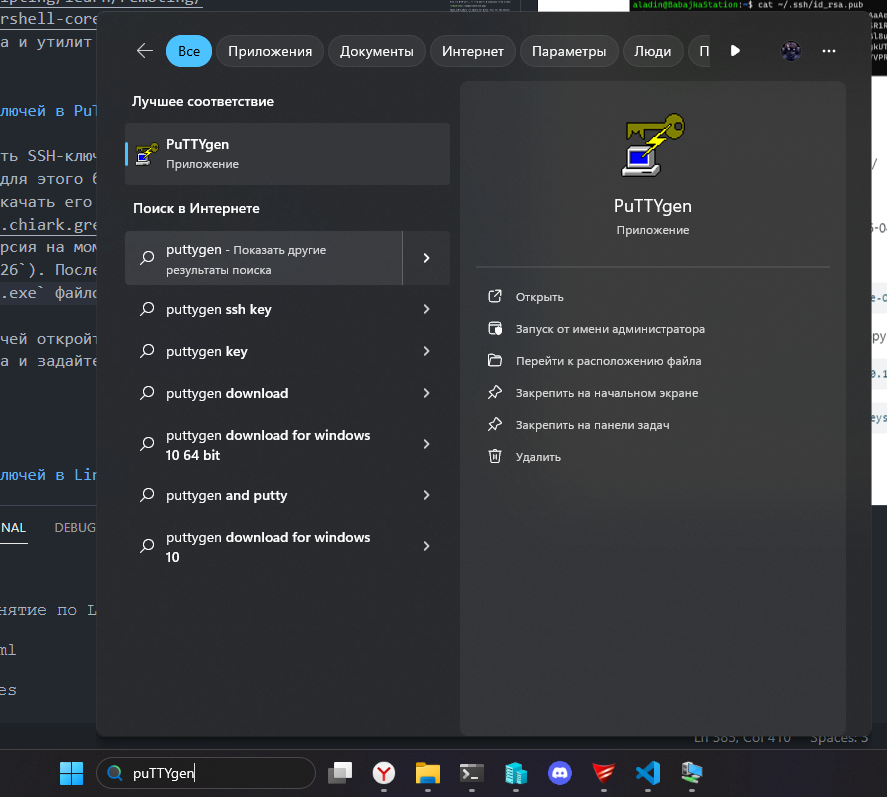
cat ~/.ssh/id\_rsa.pub



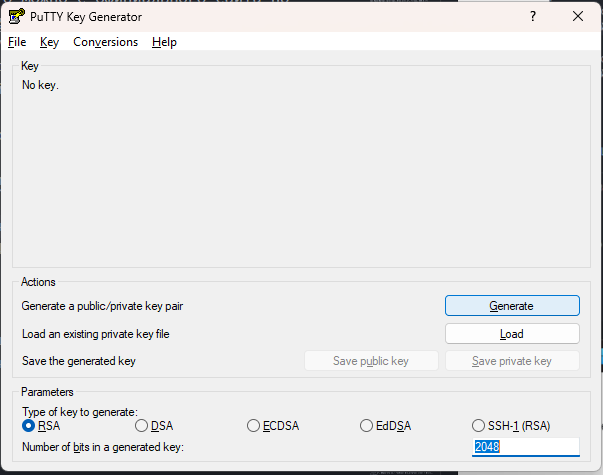
**Генерация SSH-ключей в PuTTY**[**​**](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/#%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F-ssh-%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%B9-%D0%B2-putty)

Если вы хотите создать SSH-ключи на базе ОС Windows, самым популярным решением для этого будет использование программного обеспечения Putty. Скачать его можно с официального сайта по [ссылке](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html). После установки программы вам будет доступно несколько .exe файлов.

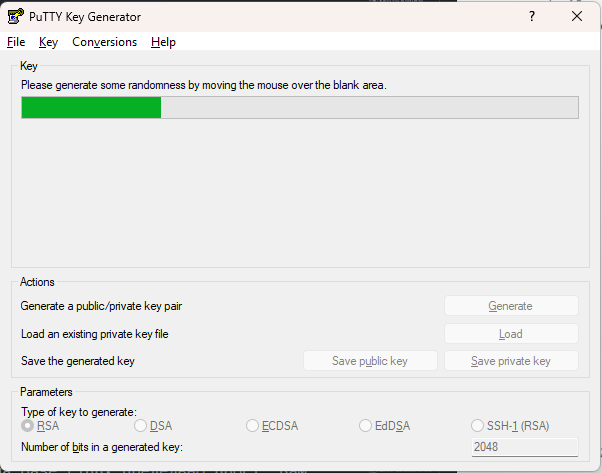
Для создания SSH-ключей откройте puttygen.exe:



Выберите необходимый тип ключа и задайте его размер. Мы будем использовать RSA с длиной 2048:

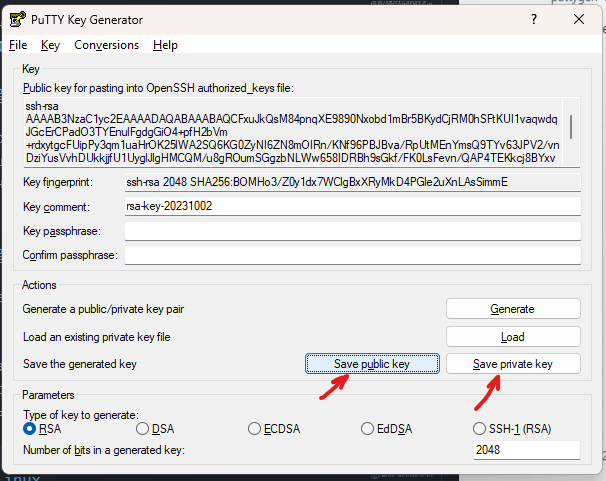


Далее нажмите Generate и водите мышкой по экрану в свободной зоне до тех пор, пока генерация SSH-ключей не будет завершен. Это необходимо делать для того, чтобы задать псевдослучайность при создании ключей:



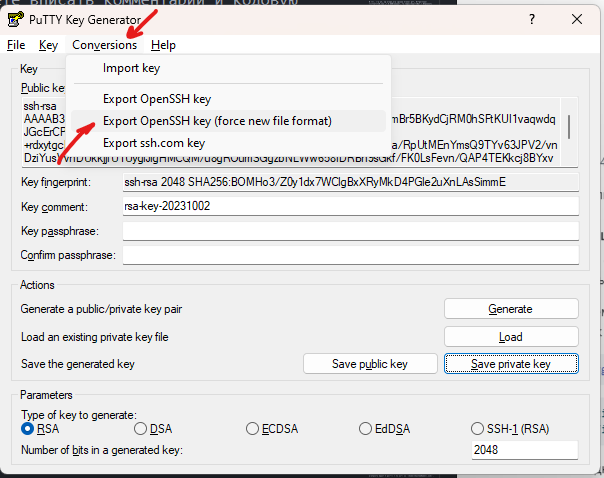
После генерации ключа вы можете вписать комментарий и кодовую фразу.

Далее необходимо сохранить на диск открытый и закрытый SSH-ключи, для этого нажмите на соответствующие кнопки в окне программы. Вы можете самостоятельно выбрать путь, куда сохранять данные ключи:

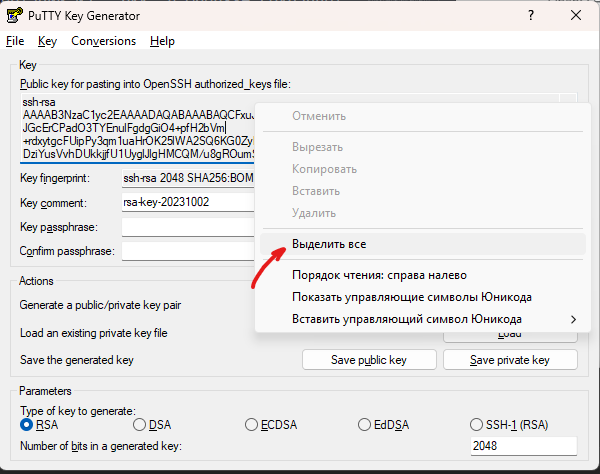


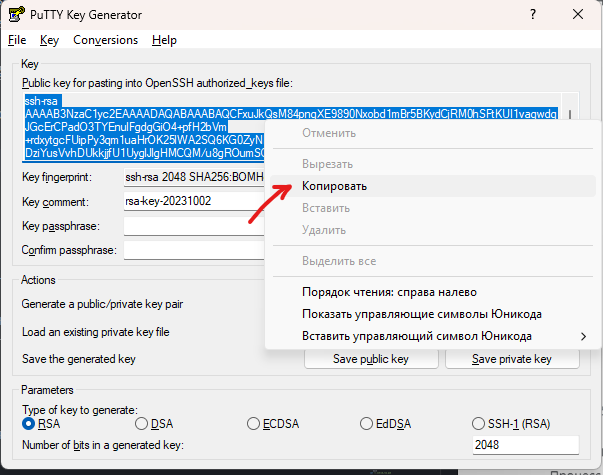
Для публичного используйте имя id\_rsa.pub. Для приватного id\_rsa.ppk.

Putty сохраняет закрытые ключи с разрешением .ppk, что означает, что такой ключ можно будет использовать только с Putty. Чтобы сохранить секретный ключ в формате, пригодном для OpenSSH, нужно во вкладке Конвертация выбрать пункт "Экспортировать ключ в формате OpenSSH (новый формат)" и указать расположение нового файла (старый формат при конвертации из .pkk в OpenSSH сохраняет не все поля — например, комментарий):



Посмотреть публичный ключ можно в окне PuTTY Key Generator. Скопировать можно через правую кнопку мыши, вызывав тем самым контекстное меню:



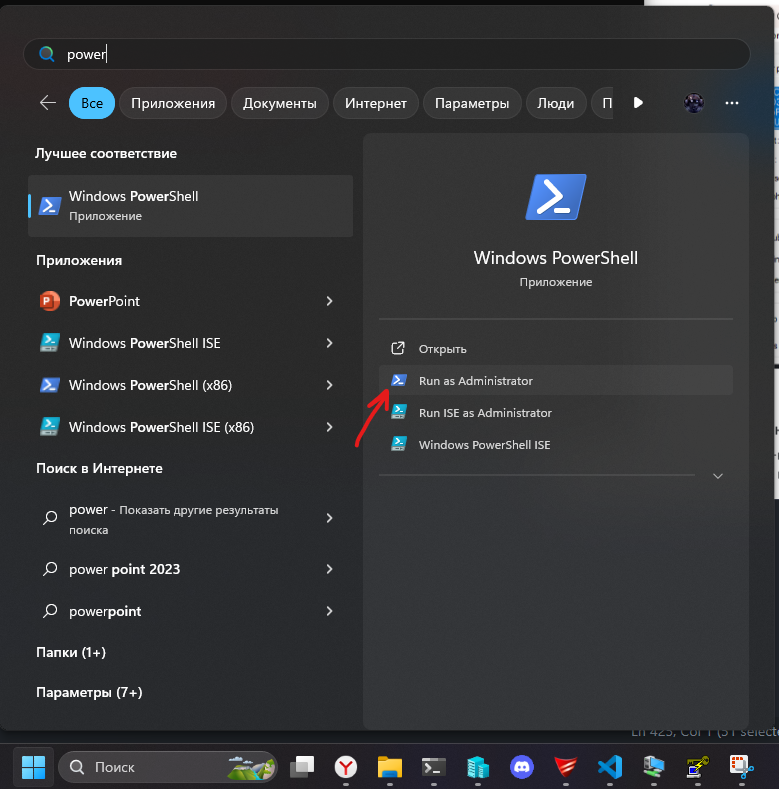


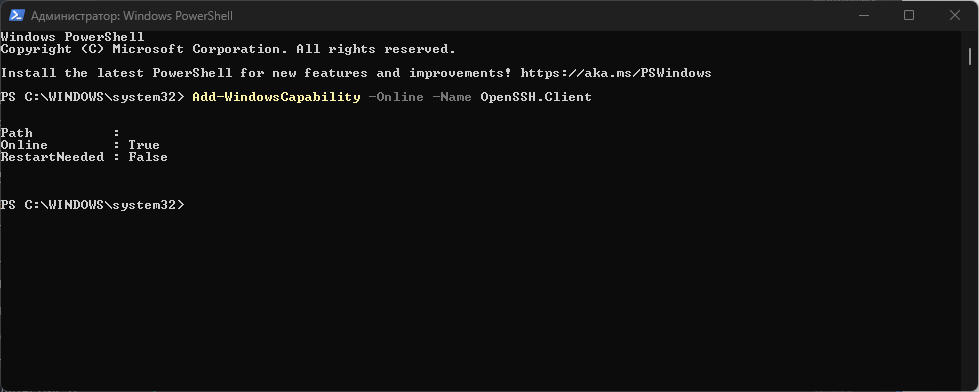
**Генерация SSH-ключей в Windows с помощью OpenSSH[​](https://iu5edu.ru/wiki/devops/docs/labs/linux/ssh/" \l "%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F-ssh-%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%B9-%D0%B2-windows-%D1%81-%D0%BF%D0%BE%D0%BC%D0%BE%D1%89%D1%8C%D1%8E-openssh" \o "Прямая ссылка на Генерация SSH-ключей в Windows с помощью OpenSSH)**

Не так давно в ОС Windows добавили возможность использования инструментов OpenSSH. Пакет добавлен в ОС, начиная с Windows 10 и Windows Server 2019.

Установить клиент OpenSSH можно с помощью следующей команды в PowerShell (PowerShell нужно запустить от имени администратора):

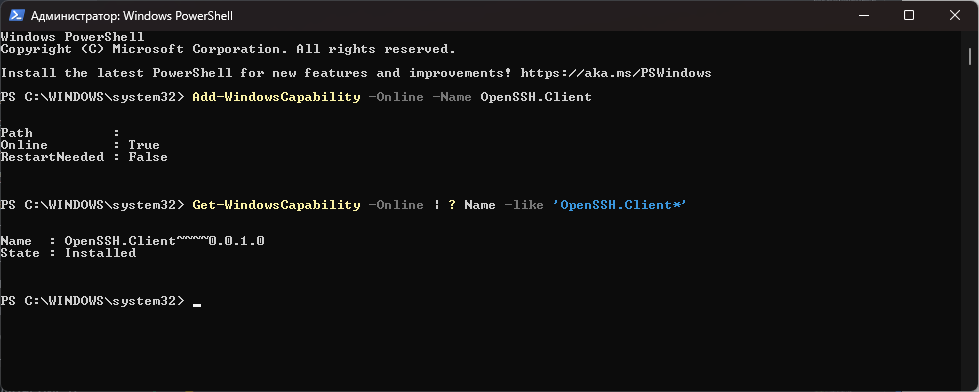
Add-WindowsCapability -Online -Name OpenSSH.Client





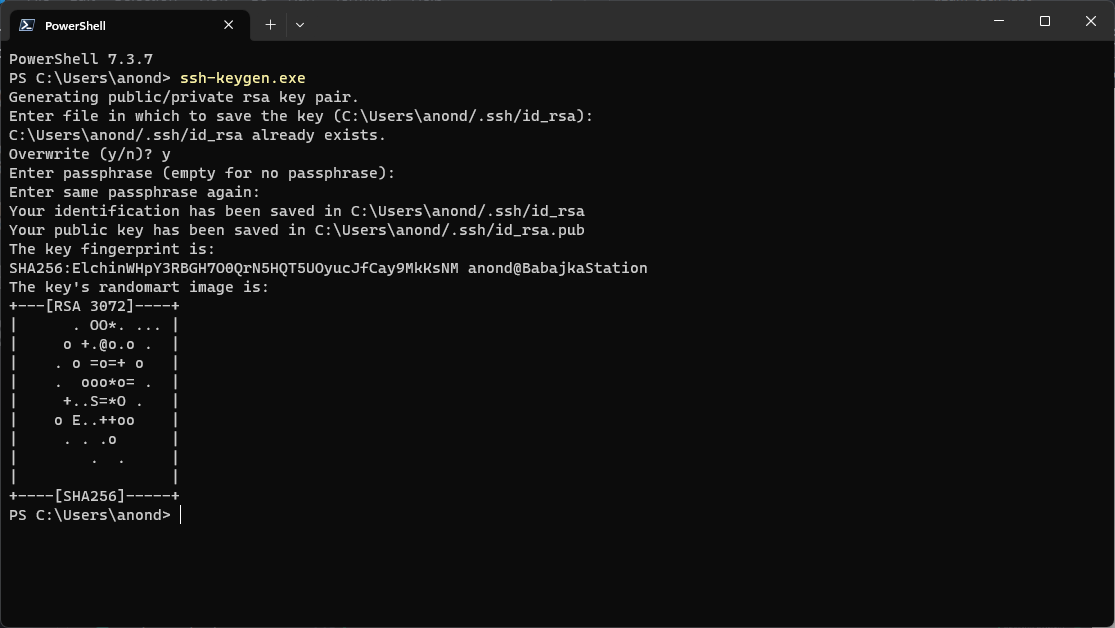
Проверить статус компонента можно командой (также от администратора):

Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client\*'

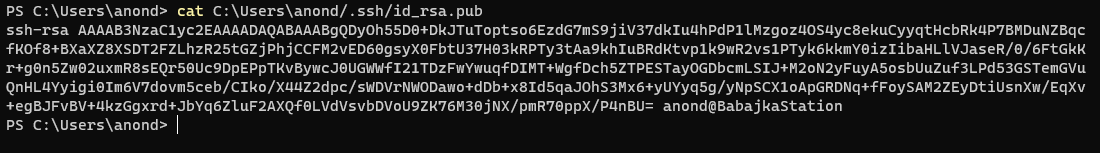


Если в ответе команды статус Installed, то загрузка выполнена успешно.

Генерация SSH-ключей происходит таким же образом, как и в ОС Linux, с помощью ssh-keygen (команды нужно выполнять от имени пользователя):



Созданные ключи можно найти в папке C:\Users\<имя пользователя>\.ssh, если она не была изменена при генерации. Посмотреть ключ можно с помощью команды cat:



**Копирование открытого ключа на сервер**

**Linux**

Данный метод подойдет тем, чья ОС поддерживает команду SSH-Copy-ID, и удаленный сервер имеет доступ по SSH без ключа. Если это не так, попробуйте использовать второй или третий способ.

Синтаксис команды выглядит следующим образом:

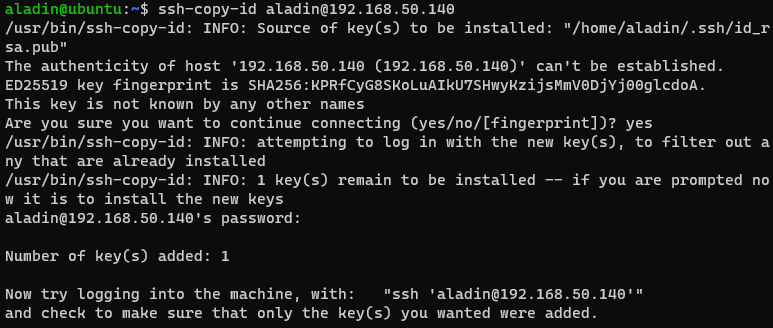
ssh-copy-id {USERNAME}@{IP-ADDRESS-OR-FQDN} -p {PORT}

Замените {IP-ADDRESS-OR-FQDN} на IP-адрес или полное доменное имя (полное доменное имя) сервера, на которое вы хотите скопировать открытый ключ, {USERNAME} - имя пользователя, к которому хотите подключиться, {PORT} - порт SSH (по умолчанию 22).

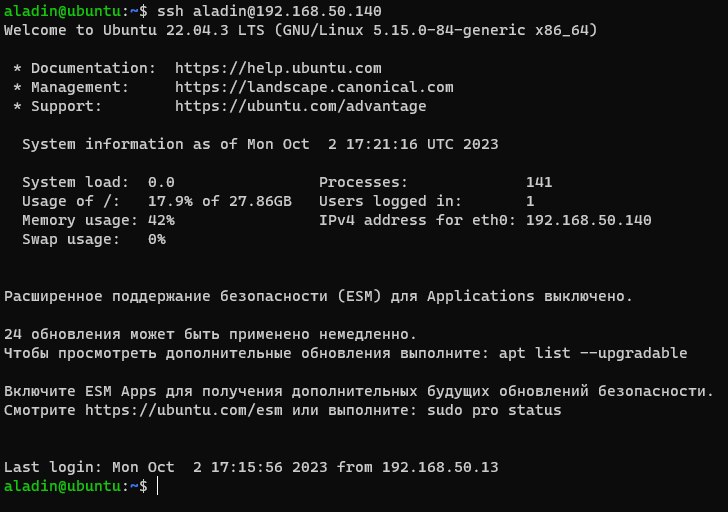
предупреждение

Далее используйте имя пользователя ({USERNAME}), которого вы ранее сделали суперпользователем (ansible).

Обратите внимание на последние строки вывода:

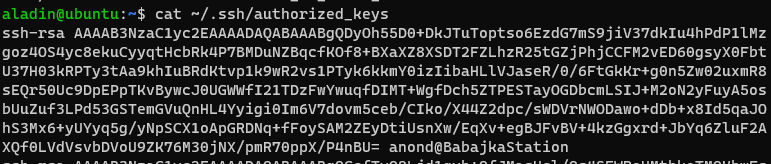


Наш клиент запрашивает разрешение на продолжение подключения к удаленному узлу, так как встречает его впервые. Необходимо ввести yes. После успешного подключения ключи будут добавлены и мы увидим приглашение к вводу пароля. Вводим пароль и получаем подтверждение о том, что теперь мы можем подключиться к удаленному узлу Linux с помощью команды ssh:



Чтобы проверить, действительно ли скопировался открытый ключ, найдем искомый файл authorized\_keys на удаленном узле и посмотрим его содержимое, так как именно в него добавляются открытые SSH-ключи. Содержится он в домашней директории в скрытой папке .ssh. Посмотреть его можно так:

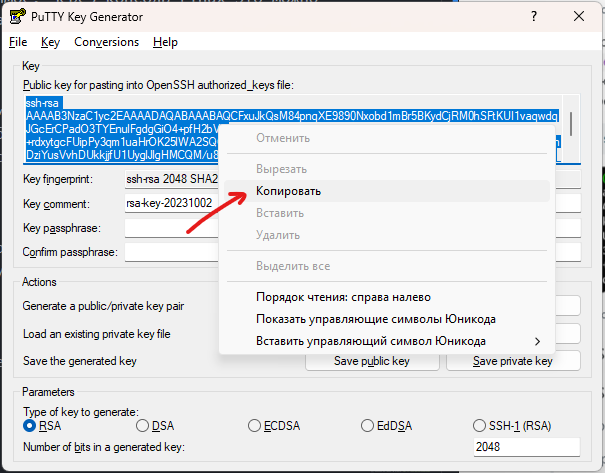
cat ~/.ssh/authorized\_keys



**PuTTY**

Альтернативный способ показан в [статье](https://selectel.ru/blog/ssh-authentication/). В нем используется только putty.exe без Pageant.

Скопируйте публичный ключ, который отображается в puttygen.exe:

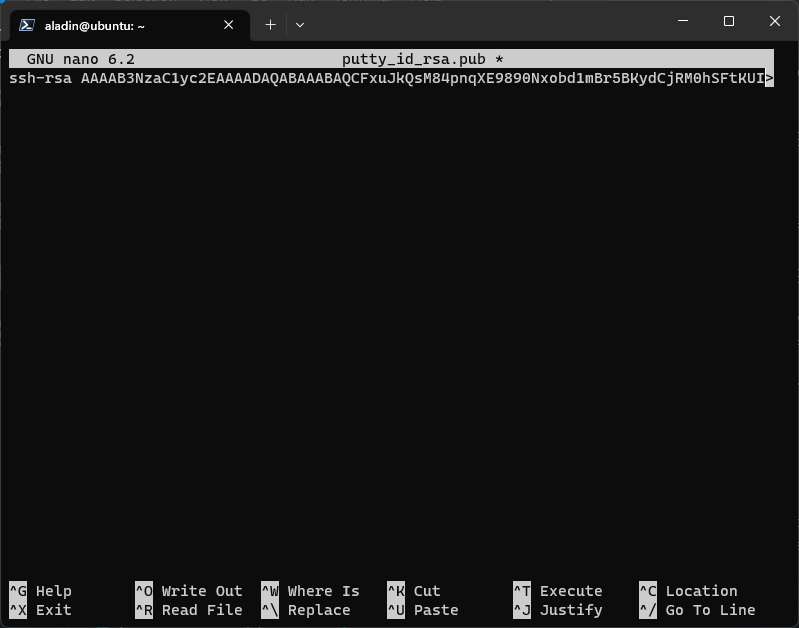


Затем следует любым доступным вам способом подключиться к удаленному серверу, куда необходимо скопировать открытый SSH-ключ.

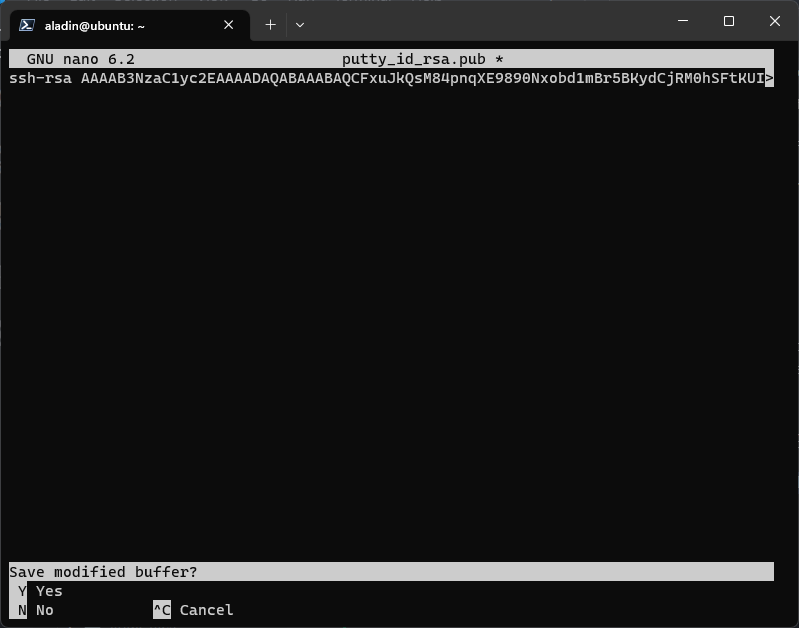
В домашней директории создайте файл putty\_id\_rsa.pub. Сделать это можно с помощью редактора Nano:

nano putty\_id\_rsa.pub

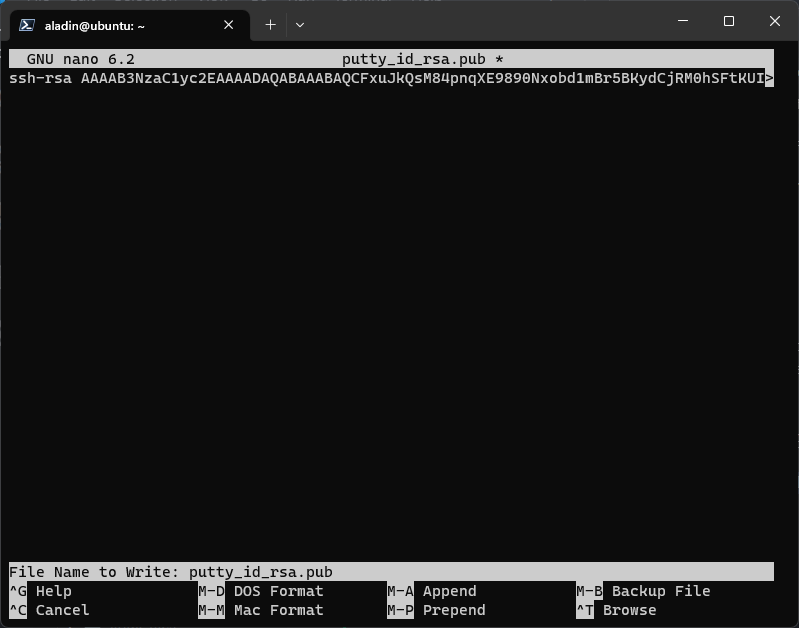
Далее вставьте содержимое из буфера обмена с помощью нажатия правой клавишы мыши в окне или с помощь комбинации клавиш shift+insert:



Сохраните содержимое с помощью комбинаций ctrl+x. Подтвердите, что вы хотите сохранить содержимое из буфера на диск нажав на клавишу y:



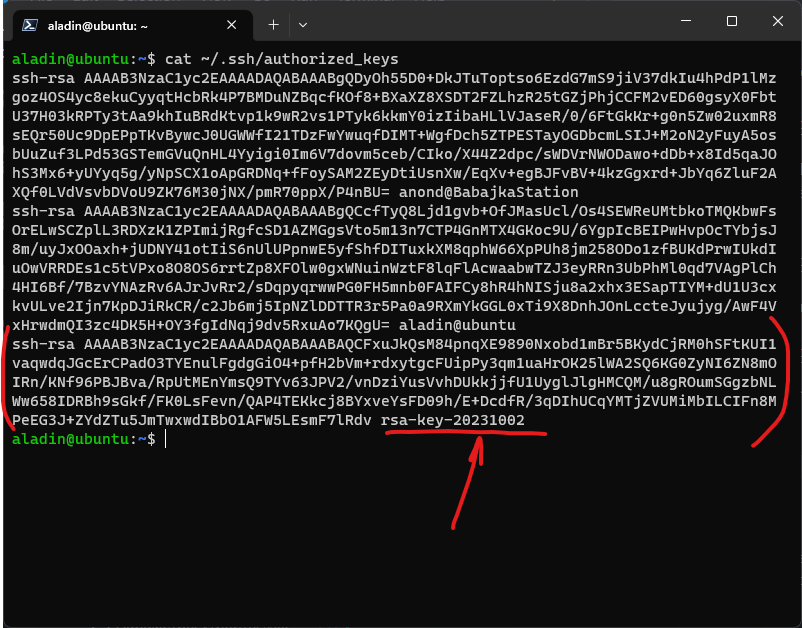
Далее подтвердите с помощью клавиши enter путь сохранения:



Затем требуется добавить наш публичный ключ в authorized\_keys.

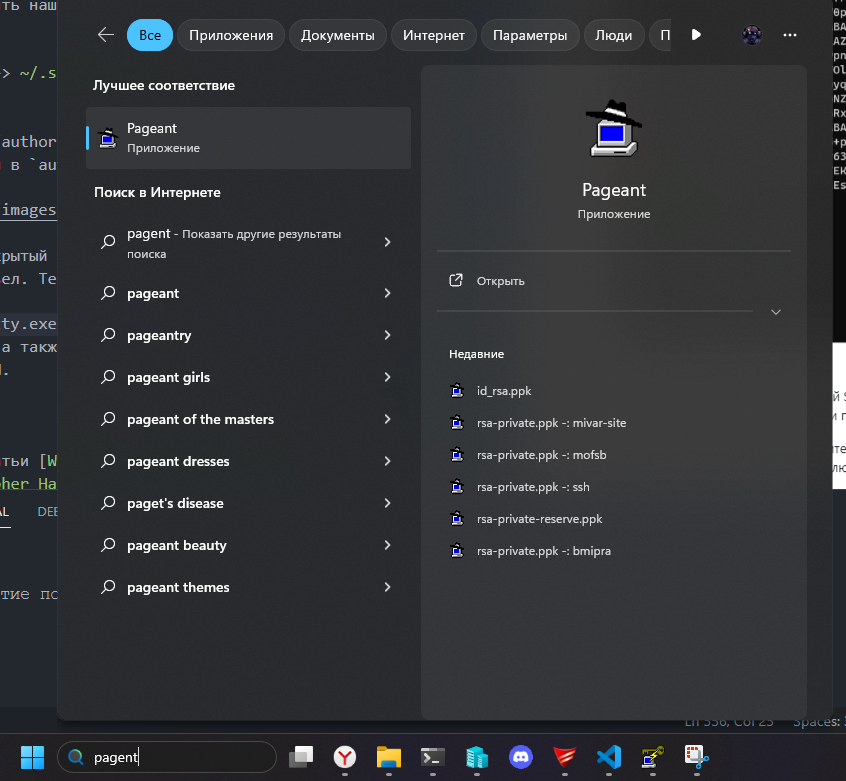
cat putty\_id\_rsa.pub >> ~/.ssh/authorized\_keys

С помощью cat ~/.ssh/authorized\_keys проверьте, что ключ был действительно добавлен в authorized\_keys:

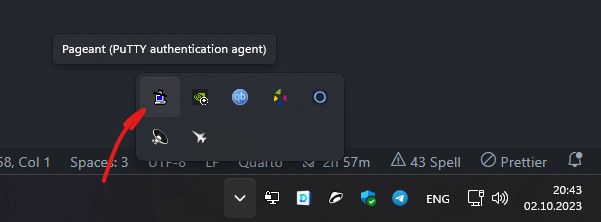


Мы уже скопировали открытый SSH-ключ, сгенерированный с помощью Putty, на удаленный узел. Теперь необходимо произвести подключение.

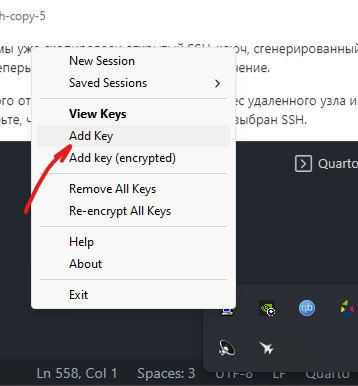
Для этого откройте Pageant и добавьте туда приватный ключ, сгенерированный с помощью puttygen.exe:



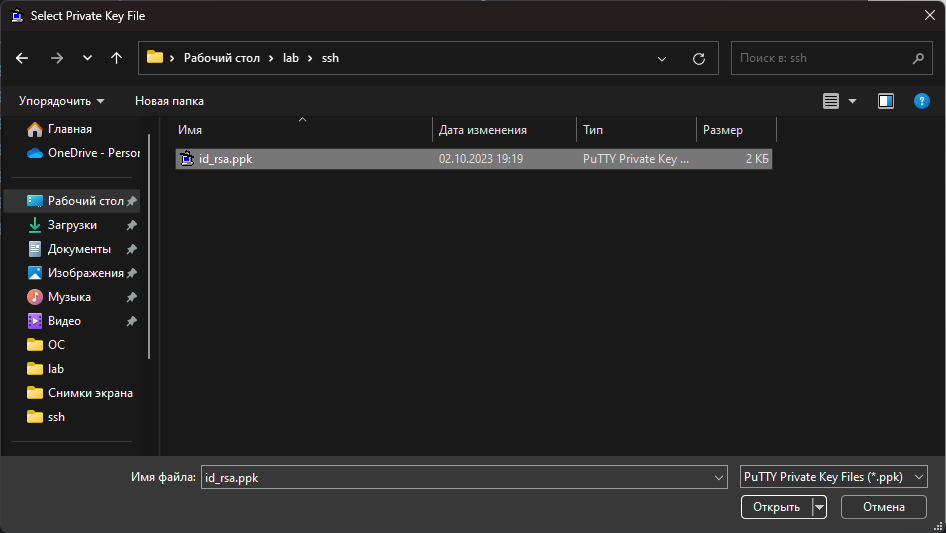
Pageant по умолчанию после запуска прячется в значках панели управления рядом с часами:



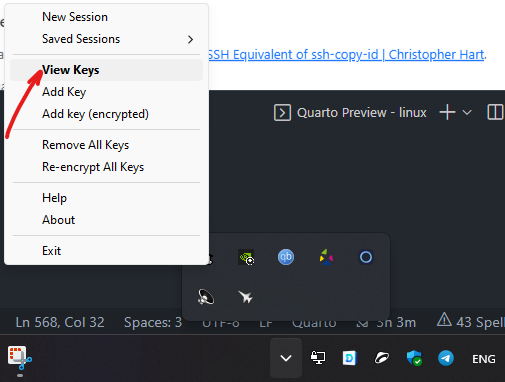
Вызовите контекстное меню с помощью правой кнопки мышки и выберите пункт "Add key":



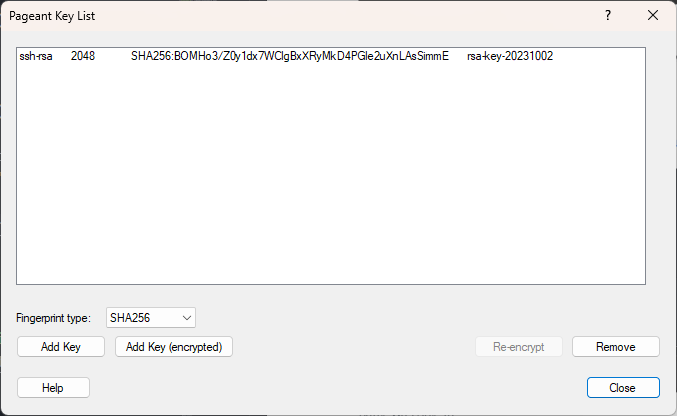
Откройте ключ в формате .ppk:



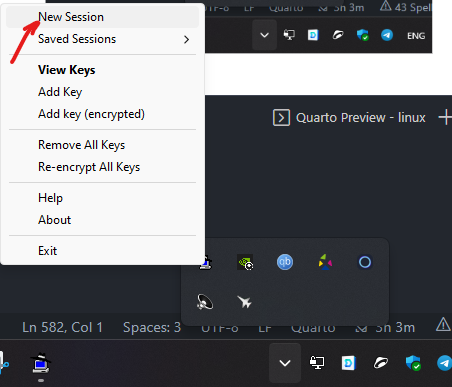
Чтобы убедиться, что ключ был добавлен, необходимо в том же контекстном меню Pageant выбрать пункт "View Keys":



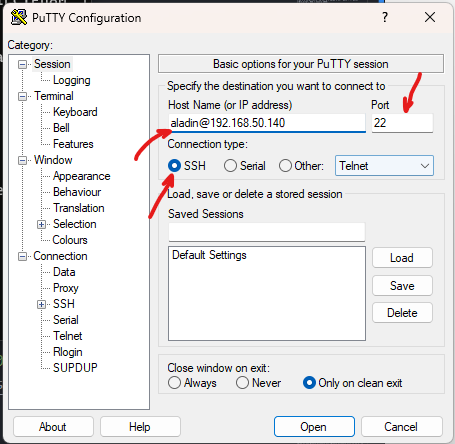
Откроется окно с активным ключами:



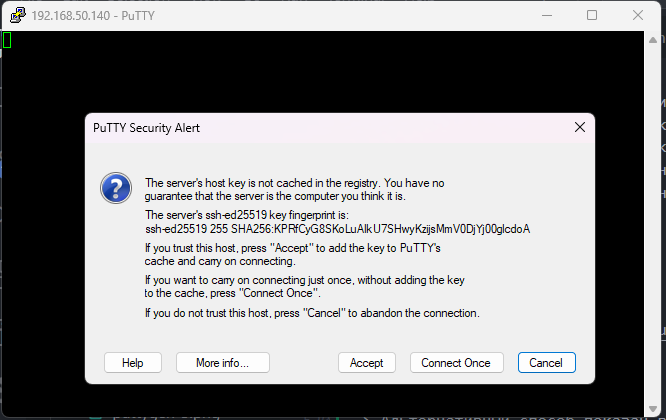
Теперь можно установить сессию, которая будет для подключения использовать наш приватный ключ. В контекстном Pageant меню выберите "New Session":



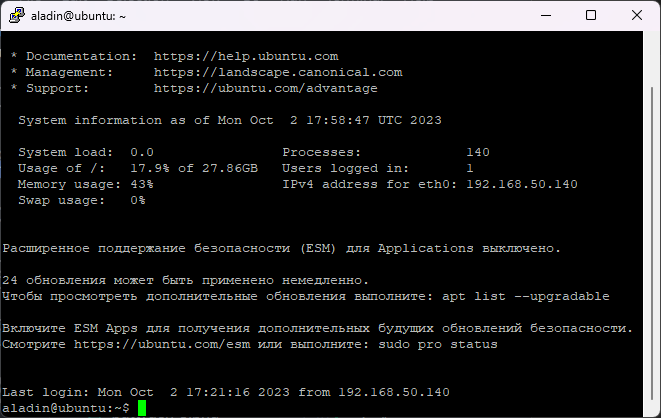
Введите IP-адрес удаленного узла, имя пользователя и порт для подключения, а также проверьте, что в качестве типа подключения выбран SSH:



Нажмите "Open". При первом подключении вам потребуется подтвердить, что удаленный узел вам известен и его можно добавить в известные:



При успешном подключении появится приглашение к вводу команд:

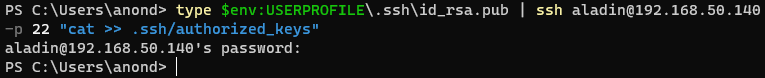


**PowerShell**

Далее мы используем приведенную ниже онлайн-команду PowerShell, чтобы скопировать содержимое открытого ключа id\_rsa.pub на удаленное устройство Linux. Замените {IP-ADDRESS-OR-FQDN} на IP-адрес или полное доменное имя (полное доменное имя) сервера, на которое вы хотите скопировать открытый ключ, {USERNAME} - имя пользователя, к которому хотите подключиться, {PORT} - порт SSH (по умолчанию 22):

type $env:USERPROFILE\.ssh\id\_rsa.pub | ssh {USERNAME}@{IP-ADDRESS-OR-FQDN} -p {PORT} "cat >> .ssh/authorized\_keys"

Далее у вас спросят пароль от пользователя. Пример того, как выполняется копирование ключа:



Убедитесь, что вы можете подключиться по SSH к удаленному устройству Linux с помощью команды ssh. Ниже показан пример подключения к удаленному устройству Linux по IP-адресу. Обратите внимание, что нам не нужно было вводить пароль, чтобы установить SSH-подключение к удаленному устройству Linux:

