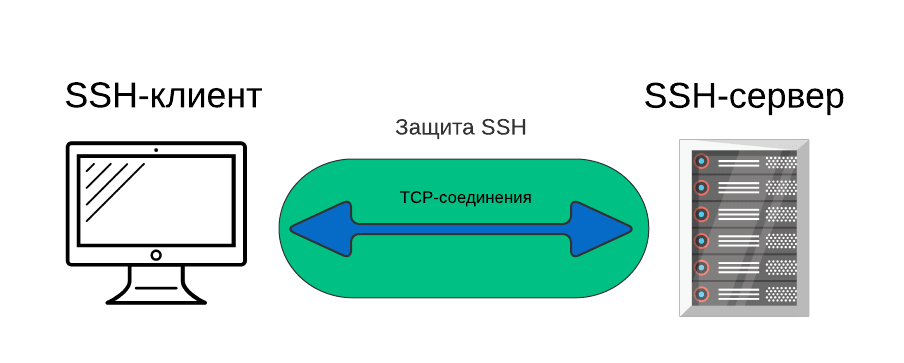
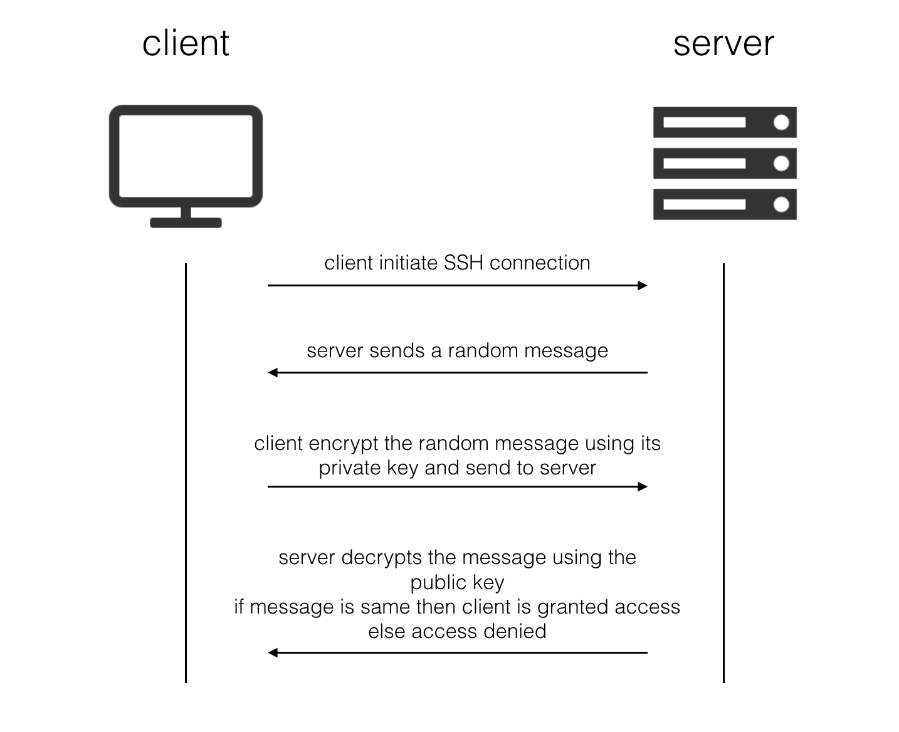
**=============== SSH ===============**





SSH — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений. Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли

Парой ключей называются файлы открытого и закрытого ключей, которые используются в некоторых протоколах аутентификации.

При аутентификации SSH на основе открытого ключа используются асимметричные алгоритмы шифрования для создания двух файлов ключей, один из которых считается закрытым, а второй открытым. Файлы закрытых ключей выполняют функцию паролей, а значит, должны быть постоянно защищены. Если кто-то получает закрытый ключ, он может войти как вы на любой сервер SSH, к который у вас есть доступ. Открытый ключ размещается на сервере SSH. Его можно свободно распространять, не компрометируя закрытый ключ.

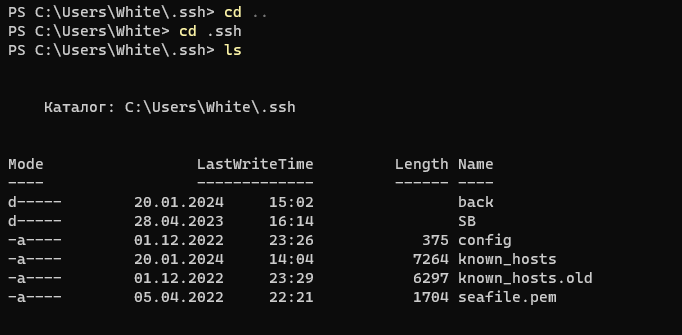
**Генерация пары ключей**

В консоли

Переходим в папку ssh

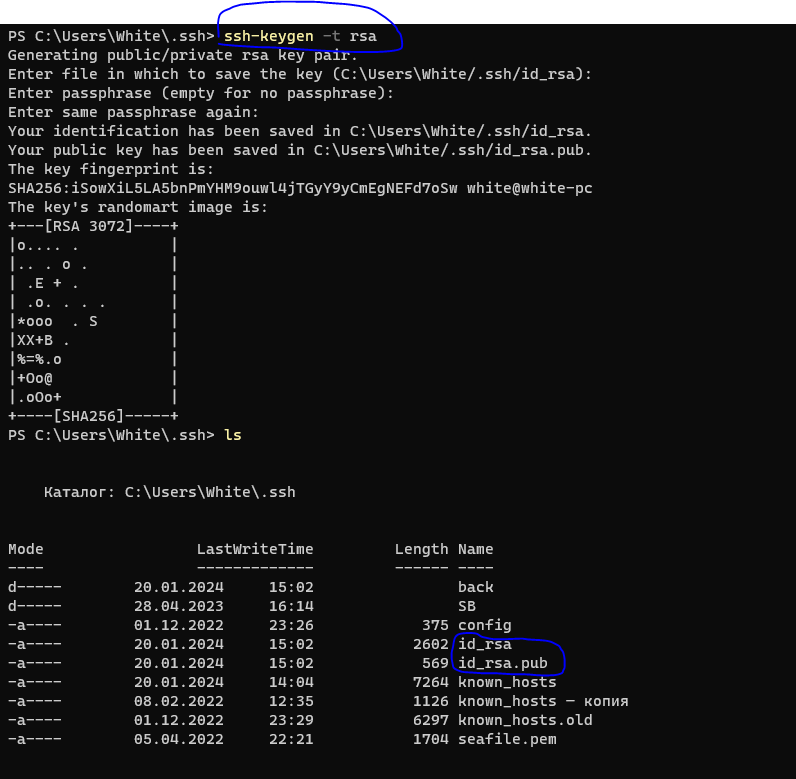
cd .ssh

Смотрим её содержимое

ls  


Создаём пару (приватный и публичные ключи)

ssh-keygen -t rsa



id\_rsa – Приватный ключ

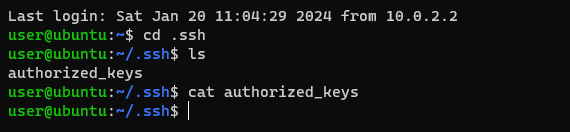
id\_rsa.pub - Публичный ключ

Сохраняем эти файлы!

**Копирование ключа на сервер**

Для подключения по SSH используя ключи, нужно публичный ключ скопировать на сервер к которому подключаемся. Ключ размещается в файле /home/user/.ssh/ authorized\_keys

Проверим что файл пустой



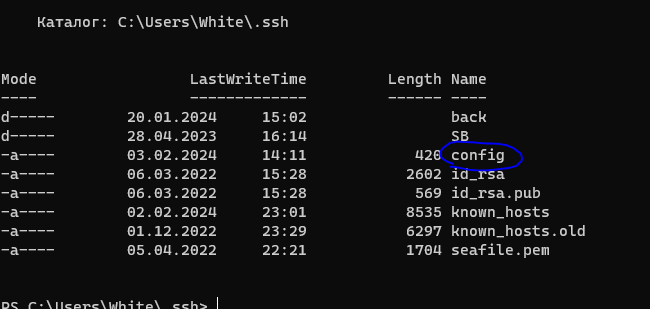
**ssh-copy-id** — небольшая утилита для копирования публичных ключей на удалённый ПК. Пара ключей (приватный и публичный) генерируются с помощью [ssh-keygen](http://xgu.ru/w/index.php?title=ssh-keygen&action=edit&redlink=1" \o "ssh-keygen (такой страницы не существует))

<http://xgu.ru/wiki/ssh-copy-id>

ssh-copy-id -i id\_rsa.pub user@host

**Настройка SSH клиента (**config file**)**

Настройка подключения осуществляется правкой конфигурационного файла config



Пример  
Host server  
 HostName server  
 User white  
 IdentitiesOnly=yes  
 IdentityFile ~/.ssh/id\_rsa

Где:

***Host***

Ограничивает множество хостов, к которым применяются последующие (до ближайшей новой директивы Host директивы, по указанным шаблонам (хост должен соответствовать хотя бы одному шаблону). Шаблон состоящий из одного символа `\*' соответствует любому хосту. Под хостом в данном контексте понимается аргумент имя-хоста передаваемый в командной строке (т.е. никаких преобразований перед сравнением не выполняется).

***HostName***

Устанавливает соответствие между псевдонимами, сокращениями и настоящими именами хостов. По умолчанию используется имя передаваемое в командной строке. Допустимо непосредственное указание IP-адресов (как в командной строке, так и в директивах HostName ).

**User** – имя пользователя под которым подключаться

***IdentitiesOnly***

Разрешить использование программой ssh(1) только файлов идентификации указанных в файлах конфигурации , даже если ssh-agent1 предоставляет и другие субъекты. Допустимые значения - ``yes'' и ``no'' Полезно если ssh-agent предоставляет много различных субъектов. Значение по умолчанию - ``no''

***IdentityFile***

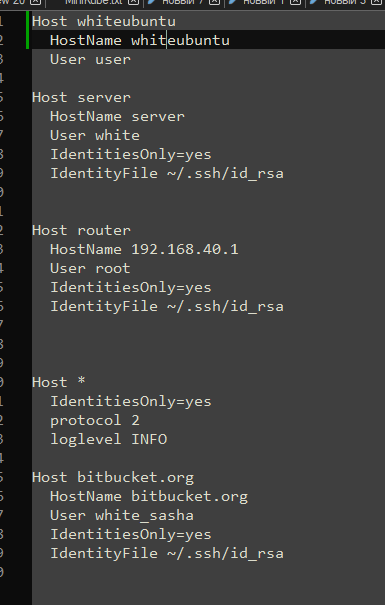
Файл с идентификационными данными (субъектом) RSA или DSA. Значение по умолчанию - ~/.ssh/identity для протокола версии 1, ~/.ssh/id\_rsa и ~/.ssh/id\_dsa для протокола версии 2.

В имени файла может присутствовать тильда для указания каталог пользователя, а также следующие последовательности: `%d' (домашний каталог локального пользователя). `%u' (имя локального пользователя), `%l' (имя локального хоста), `%h' (имя удалённого хоста), `%r' (имя удалённого пользователя).

Допустимо указание нескольких файлов с идентификационными данными в файлах конфигурации; все они будут пробоваться по очереди.

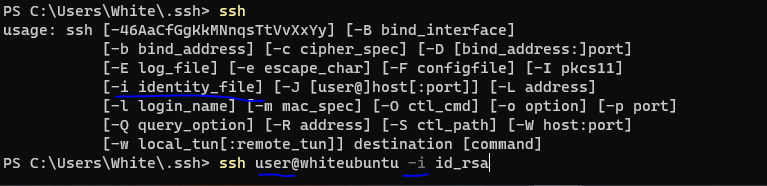
[подробней](https://www.opennet.ru/man.shtml?topic=ssh_config&category=5&russian=0)

пример:



После этих настроек ожно подкючаться по SSH без указания имени пользователя и указания какой ключ использовать, если их несколько для нескольких серверов.





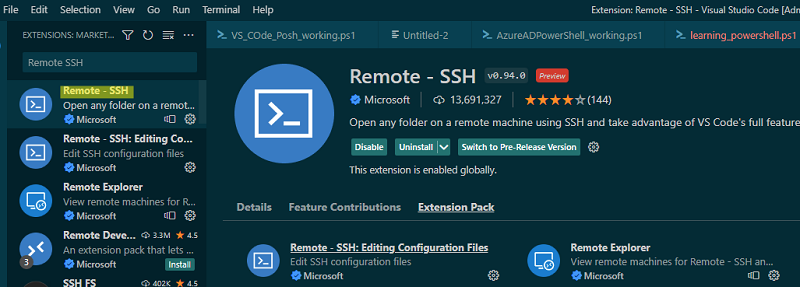
**Настройка SSH подключений в Visual Studio Code**



[Полная статья](https://winitpro.ru/index.php/2022/12/22/nastrojka-ssh-vscode/)

Visual Studio Code – это очень функциональный и универсальный текстовый редактор, который могут использовать не только разработчкики, но и системные администраторы. Вы можете использовать VS Code для редактирования скриптов, конфигурационных файлов или даже командной оболочки консоли. С помощью расширения Remote SSH вы можете подключаться к удаленным компьютерам по SSH и редактировать любые файлы, скрипты, конфиги или выполнять команды на удаленных Windows или Linux хостах прямо из VS Code, установленного на вашем компьютере.

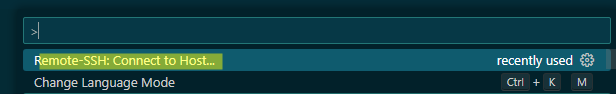
Запустите VS Code, перейдите в раздел Extensions ( Ctrl + Shift + X ), найдите и установите расширение Remote SSH



Выберите Remote Explorer в левой панели VS Code и создайте новое SSH подключение. Мастер подключения запросит:

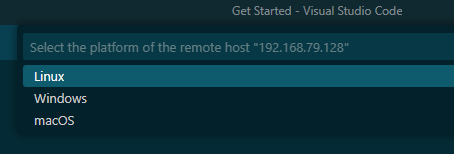
строку SSH подключения: root@192.168.31.4

путь к конфигурационному файлу, в котором будут храниться настройки этого подключения (по умолчанию C:\Users\%username%\.ssh\config .



Теперь вы можете подключиться к удаленному хосту по SSH. Выберите нужный хост в списке и выберите Connect.

При первом подключении VSCode запрашивает у вас тип платформы на удаленном SSH хосте (Linux/Windows/MacOS) и пароль для подключения:



После подключение к удаленном SSH хосту вы можете прямо из VSCoder редактировать любые текстовые файлы и выполнять команды в Shell как будто вы подключились с помощью ssh-клиента

(чтобы отобразить окно терминала, выберите Terminal -> New Terminal),

