ssh_server on ubuntu_vm, and ssh_client on windows_host:

(установка программы (точнее) сервиса (службы): “SSH Server” на виртуальную машину на которой установлена операционная система Ubuntu)

on vm:

```
$ sudo apt install openssh-server
```

p.s. по фефолту эта программа (точнее этот сервис) запускается сразу

p.s. чтобы проверить статус этого сервиса - выполнить команду:

```
$ sudo systemctl status ssh
```

про (обычное) подключение по ssh

сначала просто проверить что сервис "ssh server" запущен и хоть как-то работает
для начала просто на Локалхосте - то есть отправить запрос на "loopback интерфейс"
то есть: on vm - выполнить команду:

```
$ ssh localhost
```

p.s. по дефолту будет разрешено подлючаться по ssh к любому из сетевых интерфейсов (то есть и к “lo”, и к “eth0”, и к “eth1”),
а чтобы задать только конкретные разрешенные интерфейсы
это нужно задать в ssh конфиге:
ДОПИСАТЬ ПРО ЭТО

p.s. для того чтобы вручную запустить (стартануть) сервис "ssh server" - выполнить команду:

```
$ sudo systemctl enable --now ssh
```

p.s. по умолчанию - брандмауэр Ничего Не блокирует (ну то есть не блокируется порт 22 и т.д.)
но *если что* - чтобы настроить на фаерволе “правило”: “разрешить подключение по ssh (то есть разрешить получать запросы на порт который слушается службой

“ssh server” (порт задается в ssh конфиге, а дефолтное значение равно 22))” - нужно выполнить команду

```
$ sudo ufw allow ssh
```

p.s. чтобы подключиться по ssh из (from) Dell Windows на (к) Ubuntu VM:

- нужно на Dell Windows установить какой-нибудь ssh-клиент (то есть схема такая: сейчас на виртуалке установлен ssh server, чтобы подключиться (по ssh) к этому ssh серверу из (from) хоста - нужно на хост (Windows) установить ssh client (который будет подключаться к ssh серверу))
  (p.s. в Windows 11 уже есть встроенный ssh client, так что: ничего Не нужно устанавливать (то есть никаких сторонних ssh клиентов Не нужно устанавливать, но *если что* - например на *более старых* версиях винды - самый простой сторонний ssh client это putty утилита (это прямо типа отдельное приложение - выполнено в стиле отдельного shell терминала - (то есть это Не командная утилита (которую можно запустить из повершела) а именно отдельное приложение в стиле отдельного shell терминала (наподобие как отдельный терминал mingw), putty утилита скачивается из официального сайта - там есть опция с инсталятором .msi, а ниже опция просто скачать “portable” exe-шник (тогда прямо просто этот exe-шник запускается))
- далее:

выполняем на windows в powershell-e:
ssh <vm_user>@<vm_any_adapter_ip>
p.s. example:

```
PS C:\Users\WindowsUser> ssh u1@192.168.125.223
u1@192.168.125.223's password:
```

означает: что нужно ввести пароль of пользователя u1 (который на виртуалке):
p.s. p@ssw0rd

```
u1@vm1:~$
```

означает что: подключились; теперь строка приглашения - это “pwd” на виртуалке - по дефолту это “**~**” (то есть домашняя директория пользователя (в данном случае пользователя “u1”, то есть это “/home/u1”),
так вот с этого момента можно вводить “удаленные” команды - то есть эти команды будут по факту выполняться на виртуалке (то есть на удаленном (remote) ssh сервере):
например введем команду “pwd”:

```
u1@vm1:~$ pwd
/home/u1
```

p.s. чтобы закрыть ssh-сессию:
нужно ввести команду:

```
u1@vm1:~$ exit
PS C:\Users\WindowsUser>
```

дефолтный конфиг для ssh сервера: /etc/ssh/sshd_config
по умолчанию “PasswordAuthentication”: true (то есть можно залогиниться просто по логину и паролю а Не по сертификату и ключу)

чтобы настроить аутентификацию по приватному ключу:

- нужно сгенерировать пару ключей

например чтобы сгенерировать новую пару 2048-битных ключей RSA - выполнить команду:
(на Ubuntu-е VM:)

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

p.s. во время выполнения команды - утилита спрашивает путь куда будут сохраняться ключи, а дефолтные пути: для приватного ключа: ~/.ssh/id_rsa, для публичного ключа: ~/.ssh/id_rsa.pub
а также:
p.s. также во время выполнения команды утилита просит ввести парольную фразу (эта фраза является опциональной, эта фраза для таких случаев когда разрешен анонимный вход по ssh - тогда эта фраза будет хоть каким-то паролем), так вот: здесь можно либо вводить фразу (это типа текст - примерно как пароль - например p@ssw0rd) либо можно Не задавать фразу - для это нужно нажать Enter вместо парольной фразы (это будет означать что парольная фраза Не задана)
а также:
p.s. если нужно несколько раз выполнить команду например для генерации нескольких пар ключей (например для логического разделения: например одна пара ключей для nginx сервиса, другая для postgresql сервиса, т.д.) то нужно задавать конкретные названия файлов (например для тех ключей которые будут использоваться для nginx-а указать название “nginx_rsa”, а для postgresql указать название “postgresql_rsa” и т.д. Но директорию рекомендуется оставлять дефолтной: ~/.ssh (это типа best practices))

SSH Port Forwarding

(проброс (перенаправление) портов)
пример:

```
localhost:~$ ssh  -L 9999:127.0.0.1:80 user@remoteserver
```

Команда устанавливает прослушку на порт 9999. Порт 80 используется для проброса.

про обратный port forwarding:
если на хосте крутится сервис на порту 3000 (например на хосте установлен git tea server по адресу: localhost:3000) и мы хотим на него заходить из виртуалки например по порту 3001
то нужно сделать "обратный" “port forwarding” - (перенаправление порта) - то есть перенаправление хостового порта на порт который на виртуалке:
выполняем на хосте такую команду:

```
ssh u1@192.168.125.223
```

затем:

```
ssh -R 3001:127.0.0.1:3000 u1@192.168.125.223
```

тогда на виртуалке можно открыть в браузере: *localhost:3001* и тогда там будет отображаться Тоже Самое что отображается на хосте (тоже в браузере) по адресу: *localhost:3000*)