

?????????? SSH ??? VPS (Windows && Linux-WPS)

?????????? ??? Windows.

Для настройки соединения у вас обязательно должен быть файл с приватным ключом и желательно с публичным тоже. В моем примере я использую ключи в формате ED25519, которые приходят сейчас на смену ключам RSA. На VPS вы сгенерировали файлы с ключами, в моем случае они назывались `id_ed25519` и `id_ed25519.pub`, лучше их переименовать в виде `id_ed25519_myproxy` и `id_ed25519_myproxy.pub`. Если у вас имеется несколько серверов с различными ключами, то названия файлов будут различаться и подключение будет управляться через файл `config`.

Положите файлы в папку **C:\Users\[your_user_name]\.ssh**, в той же папке создайте файл **config** без расширения примерно с таким содержанием:

```
Host myproxy
    HostName 11.22.33.44          # IP-адрес или домен вашего VPS
    User root                    # Логин для подключения
    IdentityFile ~/.ssh/id_ed25519_myproxy # Путь к файлу ключа
    DynamicForward 0.0.0.0:1080
    IdentitiesOnly yes
    SessionType none
    Port 22                      # Порт (если меняли стандартный)
```

Строка **DynamicForward 0.0.0.0:1080** это прямой аналог флага `-D 0.0.0.0:1080` в файле конфигурации. Он включает локальный SOCKS5-прокси на порту `1080` для всех сетевых интерфейсов (`0.0.0.0`), строка `SessionType none` это директива `RequestTTY no` которая аналогична флагу `-N` и позволяет не выполнять команды на удаленном сервере, а только держать туннель.

Запуск туннеля на ваш сервер будет теперь выполняться одной командой:

```
ssh myproxy
```

Проверить, открыт ли порт **1080** и слушаются ли на нем подключения, можно прямо из консоли Windows (Command Prompt / PowerShell) или из среды WSL.

Ниже приведено несколько простых способов проверки работает ли соединение.

?????? 1. ?????????? ?????? PowerShell (Windows) — ?????? ?????????

Откройте **PowerShell** от имени администратора или обычного пользователя и выполните:

PowerShell

■ ■

```
Get-NetTCPConnection -LocalPort 1080 -ErrorAction SilentlyContinue
```

- **Если порт открыт и прослушивается:** Команда выведет таблицу с состоянием подключения (State: Listen), а в колонке OwningProcess будет указан PID (идентификатор процесса, например, вашего ssh.exe).
- **Если порт закрыт:** Команда ничего не выведет (вернется пустая строка).

?????? 2. ?????????????????? ?????????? netstat (Windows CMD / PowerShell)

Откройте командную строку или PowerShell и выполните:

DOS

■ ■

```
netstat -ano | findstr :1080
```

- **Ожидаемый результат (порт открыт):**

Plaintext

■ ■

TCP	0.0.0.0:1080	0.0.0.0:0	LISTENING	1234
-----	--------------	-----------	-----------	------

(Значение LISTENING означает, что SOCKS-туннель успешно поднят и ждет подключений, а 1234 в конце — это PID процесса SSH).

- **Если порт закрыт:** Вывод будет абсолютно пустым.

?????? 3. ??????????? ?????? ??????????? WSL (Linux)

Если туннель или проверяемое приложение запущены внутри WSL, используйте стандартные утилиты Linux:

Bash



```
# Через ss (современный аналог netstat)
ss -tulpn | grep 1080

# Или через классический netstat
netstat -tulpn | grep 1080
```

?????? 4. ?????????-????????? ????????????? (Test-NetConnection)

Если вы хотите убедиться, что порт не просто открыт, а к нему действительно можно успешно подключиться локально (например, со стороны браузера или приложения):

PowerShell



```
Test-NetConnection -ComputerName 127.0.0.1 -Port 1080
```

В ответ вы получите отчет. Обратите внимание на нижнюю строку:

- `TcpTestSucceeded : True` — порт открыт и принимает соединения.
- `TcpTestSucceeded : False` — порт закрыт (SSH-туннель не запущен или упал).

“ Узнать, какая программа заняла порт:

Если порт 1080 занят, но вы не уверены, чем именно, узнайте PID из команды `netstat` (последнее число в строке) и выполните в PowerShell:

```
Get-Process -Id <PID>
```

Revision #3

Created 2026-08-07 10:21:04 UTC by Nikolay

Updated 2026-08-12 10:17:36 UTC by Nikolay