

?????????? VPN ?????? ????????

? VPS

?????????? ?? ??????????.

- Имеется VPS с выделенным IP где-то в Европе. Там стоит Linux, в моем случае это Ubuntu. Нужно иметь на сервер права администратора. Сервер должен принимать соединения по ssh.
- Машина с Windows 10 или 11.
- VPS должен быть виден через ваше интернет-соединение. В моем случае у меня было два интернет-подключения: через мобильное соединение VPS виден не был, через проводное соединение видимость была.
- Не должно быть работающих VPN-соединений.

?????????? ????????

Нужно открыть windows PowerShell от лица администратора.

В windows PowerShell нужно подключиться к вашему VPS:

```
ssh -D 0.0.0.0:1080 -N root@[IP_your_vps]
```

Windows может попросить подтвердить использование туннеля.

Если у вас нет ключа доступа к VPS, то потребуется ввести пароль для root.

Сессия откроется, это окно PowerShell нужно оставить открытым, оно "зависнет", то есть через туннель пойдет обмен.

Откройте другое окно PowerShell с обычными правами. Проверка соединения:

```
curl.exe --proxy socks5h://127.0.0.1:1080 https://api.ipify.org
```

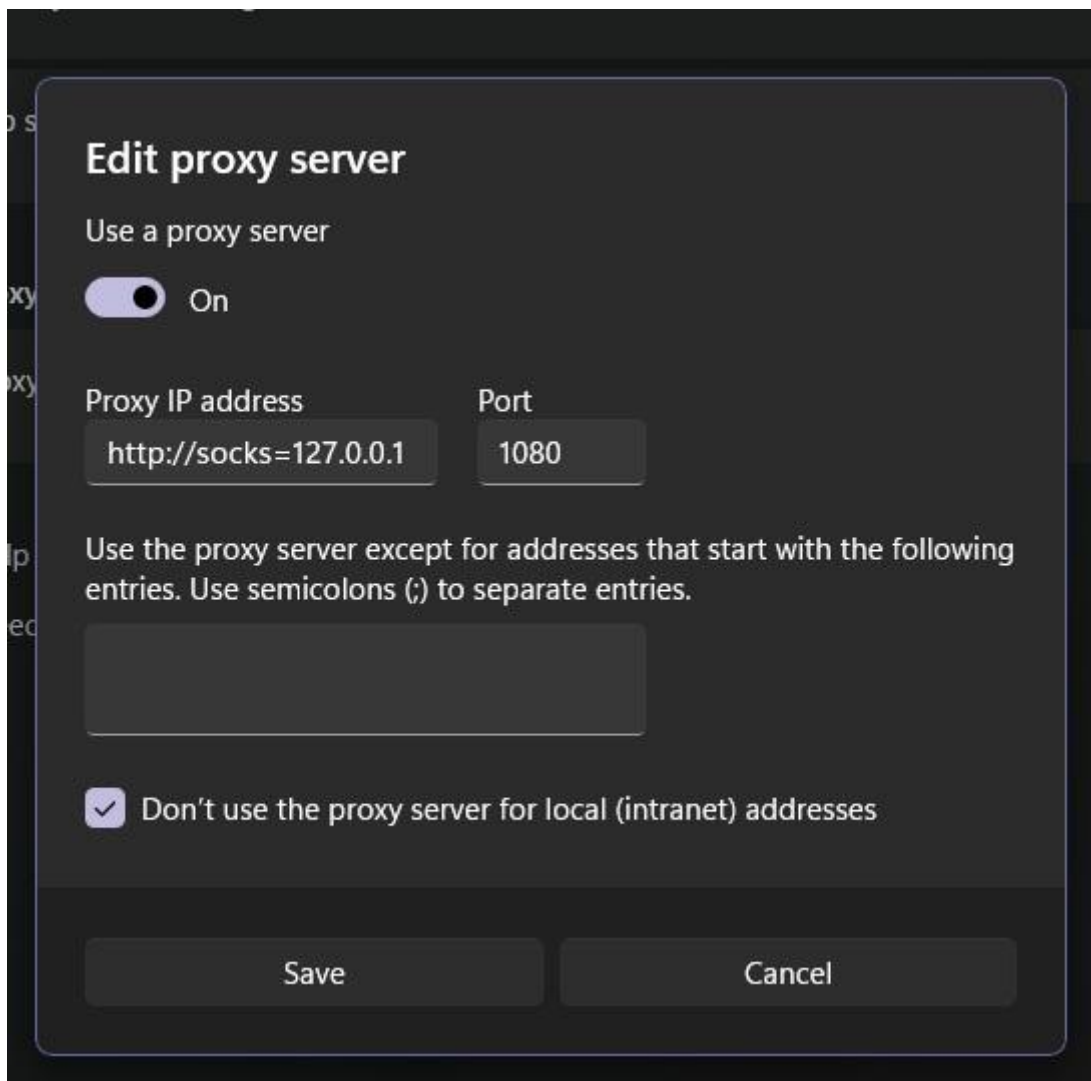
Команда должна показать IP вашего VPS.

В настройках Windows есть раздел "проxy" его можно использовать для включения обмена, но с оговорками. Нужно иметь в виду, что туннель открывается только для протокола socks, для протоколов http и https он не работает. В поле "Proxy IP address" нужно прописать

```
socks=127.0.0.1:1080
```

в поле для значения порта ничего быть не должно.

После того, как вы запишите эти настройки, они корректно пропишутся в реестр windows, но при редактировании оттуда будут взяты неверные значения. при повторной записи работа прокси сломается, ниже пример того, как интерпретирует эти настройки интерфейс windows:



для корректной настройки лучше пользоваться PowerShell, чтобы посмотреть настройки используйте команду

```
Get-ItemProperty 'HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings'
```

для установки протокола socks:

```
Set-ItemProperty 'HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings'  
ProxyServer 'socks=127.0.0.1:1080'
```

Нужно иметь в виду, что с таким проху в вашем случае будут работать только те приложения, которые используют протокол socks.

Не все приложения Windows используют системный SOCKS-прокси.

Например:

- браузеры обычно умеют
- некоторые программы используют WinINET/WinHTTP
- некоторые (включая часть Electron/Node-приложений) игнорируют системный SOCKS и требуют отдельной настройки

Для включения проху используйте команду

```
Set-ItemProperty 'HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings'  
ProxyEnable 1
```

В обычном `cmd.exe` (или FAR например) аналог делается через `reg add`.

Включить прокси:

```
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v ProxyEnable /t  
REG_DWORD /d 1 /f
```

Отключить прокси:

```
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v ProxyEnable /t  
REG_DWORD /d 0 /f
```

Установить SOCKS-прокси:

```
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v ProxyServer /t  
REG_SZ /d "socks=127.0.0.1:1080" /f
```

Проверить текущие значения:

```
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v ProxyEnable  
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v ProxyServer
```

После изменения реестра некоторые программы сразу подхватывают настройки, а некоторые — только после перезапуска. Чтобы уведомить Windows о смене настроек прокси без выхода из системы, можно выполнить:

```
RUNDLL32.EXE inetctl.cpl,LaunchConnectionDialog
```

или просто перезапустить приложение (браузер, VS Code и т.п.).

?????????? ?? ?????????? ?????????? ?? ChatGPT.

Вопрос:

что конкретно делает эта команда `ssh -D 0.0.0.0:1080 -N root@[IP_your_vps]`

На вашем компьютере запускается **локальный SOCKS5-прокси** на порту `1080`.

Все программы, которые используют этот SOCKS-прокси, будут выходить в интернет через VPS.

Схема:

Приложение

↓

SOCKS5 127.0.0.1:1080

↓

SSH-соединение

↓

VPS

↓

Интернет

??????? ??????????????

```
ssh
```

Запускает SSH-клиент.

```
root@IP_VPS
```

Подключается к VPS:

Пользователь: root

Сервер: IP_VPS

Например:

```
ssh root@38.242.220.114
```

```
-D
```

Означает **Dynamic Port Forwarding**.

То есть SSH создаёт SOCKS-прокси.

Например:

```
-D 1080
```

создаёт:

```
SOCKS5 localhost:1080
```

Любая программа, умеющая SOCKS5, может его использовать.

```
0.0.0.0:1080
```

Очень важный момент.

Если написать:

```
-D 1080
```

или

```
-D 127.0.0.1:1080
```

то SOCKS будет доступен только на локальном интерфейсе Windows.

WSL обычно не сможет к нему подключиться напрямую.

Если написать:

```
-D 0.0.0.0:1080
```

то SSH слушает на всех интерфейсах:

```
127.0.0.1  
LAN IP  
WSL gateway  
и т.д.
```

-N

Означает:

```
Не запускать shell на VPS
```

То есть:

□ не открывать консоль Linux

□ только держать туннель

Окно просто висит:

```
ssh -D ...
```

и прокси работает.

??? ?? ??????? ?? ???????

Она НЕ создаёт полноценный VPN.

Не происходит автоматического перенаправления всего трафика.

Например в WSL:

```
curl https://api.ipify.org
```

по умолчанию всё ещё идёт напрямую.

А вот:

```
curl --proxy socks5h://172.31.32.1:1080 https://api.ipify.org
```

идёт через VPS.

??? ??????????? ?? VPN

VPN:

Весь трафик

↓

VPN

↓

VPS

SSH SOCKS:

Только программы,
которые настроены на SOCKS
↓
VPS

?????? ?? ??????

Для связки:

Windows
└─ WSL
 └─ VS Code
 └─ Continue

достаточно настроить Continue, VS Code или переменные окружения на использование SOCKS.

Тогда:

- Continue
- OpenAI API
- Gemini API
- GitHub Copilot

будут работать через VPS,

а остальной интернет останется обычным.

Для дополнительной безопасности я бы рекомендовал использовать не `root`, а отдельного пользователя на VPS:

```
ssh -D 0.0.0.0:1080 -N devadmin@IP_VPS
```

Это не влияет на работу туннеля, но считается хорошей практикой администрирования.

Revision #7

Created 2026-06-12 09:18:31 UTC by Nikolay

Updated 2026-06-13 06:41:20 UTC by Nikolay