

???????? VS Code ??? ??????? ? WSL

WSL - это среда Linux под Windows [подробнее о WSL](#).

Для стека **Ubuntu-хостинг + .NET 10 + React + SSH-туннели** VS Code через WSL обычно удобнее, чем запуск проекта напрямую в Windows.

1. ????? ????????????? ?????????? ? ??????????

Если у вас продакшен на Ubuntu.

Когда VS Code открыт как:

WSL: Ubuntu

команды выполняются в Linux:

```
dotnet build  
npm install  
npm run dev
```

То есть почти в той же среде, что и сервер.

Меньше сюрпризов вида:

“На Windows работает, а на Ubuntu нет”.

2. ????? ?????????? Node.js ? React

Многие инструменты фронтенда изначально ориентированы на Linux/macOS.

Например:

- Vite
- Webpack
- ESLint
- npm
- pnpm

обычно чувствуют себя в WSL стабильнее, чем при работе с проектом на диске Windows.

Особенно это заметно на больших проектах.

3. ????????? Linux-????????????

Сразу доступны:

```
grep
find
sed
awk
ssh
rsync
curl
```

Не нужно искать Windows-аналоги.

4. ???????? ????????? ? ?????????

Если вы постоянно используете SSH к VPS.

В WSL можно:

```
ssh server
scp file
rsync project
```

без дополнительных утилит.

5. AI-???????????? ????????? ????? ? ?????

Когда VS Code открыт в WSL:

```
VS Code
↓
VS Code Server (в WSL)
↓
Continue / Copilot
↓
Файлы проекта
```

AI видит проект напрямую в Linux-среде.

6. .NET ????? ? ?????????? ???????????

Если сервер Ubuntu, то:

```
dotnet publish
```

в WSL собирает проект в Linux-окружении.

Легче ловить проблемы:

- путей (/ против \)
- регистра файлов
- прав доступа
- конфигурации Linux

????? ?????? ??????????? ?????? Windows

Есть случаи, когда удобнее работать напрямую из Windows:

- WinForms
- WPF
- MAUI под Windows
- COM-компоненты
- специфичные Windows SDK

Но если у вас:

```
React  
ASP.NET Core  
Ubuntu
```

то WSL выглядит более естественным вариантом.

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

Если вы используете WSL, держите проект внутри Linux-файловой системы:

```
/home/devadmin/projects/myapp
```

а не в:

```
/mnt/c/Users/...
```

На небольших проектах разница может быть незаметна, но на React-проектах с большим количеством файлов производительность и скорость работы инструментов обычно лучше внутри файловой системы WSL.

Для работы в WSL вам нужно выполнить сначала установку VS Code под Windows, скачайте его с официального сайта по [ссылке](#).

После запуска установленного VS Code вам нужно установить расширение WSL. Перейдите на вкладку расширений и в строке поиска наберите:

```
wsl
```

В результатах поиска найдите расширение с названием WSL и примерно таким описанием:

Open any folder in the Windows Subsystem for Linux (WSL) and take advantage of Visual Studio Code's full feature set.

Установите это расширение. После установки вы сможете работать с WSL. Для этого нужно сделать к нему подключение.

Нажмите Ctrl->Alt->O или кликните в нижнем левом углу по иконке "><", откроется Remote Menu, в нем выберите "Connect to WSL", после запуска будет загружен и запущен сервер VS Code в WSL, затем вы сможете работать в проводнике с папками в вашем WSL-хранилище.

????????? ???????? Continue ? ?????????? ???
??? AI-?????????.

-- Что ниже пока неизвестно, нужно или нет --

Откройте:

```
nano ~/.bashrc
```

добавьте в конец:

```
export ALL_PROXY=socks5://172.31.32.1:1080
export HTTP_PROXY=socks5://172.31.32.1:1080
export HTTPS_PROXY=socks5://172.31.32.1:1080
```

Сохраните:

Ctrl+O

Enter

Ctrl+X

Применить:

```
source ~/.bashrc
```

??? VS Code Remote WSL

В `settings.json`:

```
{
  "http.proxy": "socks5://172.31.32.1:1080",
  "http.proxySupport": "override",
  "http.proxyStrictSSL": false
}
```

После этого:

- Continue
- Copilot
- npm
- dotnet restore
- git clone
- curl

из WSL будут идти через VPS.

Ваша итоговая схема теперь:

Windows

|

| SSH -D 0.0.0.0:1080

|

v

VPS (38.242.221.111)

|

v

Internet

WSL

|

v

172.31.32.1:1080

Revision #5

Created 2026-06-16 07:16:12 UTC by Nikolay

Updated 2026-08-12 15:21:56 UTC by Nikolay