Using Remote Tunnels for VSCode Remote Connection

Helen Wang - 2025-08-26 - Research Systems

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET)

However, since we forbid user running interactive processes on head node, to avoid multiple processes sitting on head node, users can use remote tunnels to remote machine, in this case, the compute node, for data transit from one network to another. This can eliminate the need for source code to be on your VS Code client machine since the extension runs commands and other extensions directly on the remote machine.

https://code.visualstudio.com/docs/remote/tunnels

two paths to work with tunnels:

- Run the `tunnel` command of the `code` command-line interface (CLI).
- Enable tunneling through the VS Code Desktop UI.

You may create and use tunnels through the `code` CLI.

1. Install the `code` CLI on a remote machine you'd like to develop against from a VS Code client. The CLI establishes a tunnel between a VS Code client and your remote machine.

Download vscode_cli.tar.gz and unpack it.

2. Create a secure tunnel with the `tunnel` command:

code tunnel

This command downloads and starts the VS Code Server on this machine and then creates a tunnel to it.

3. This CLI will output a vscode.dev URL tied to this remote machine, such as `https://vscode.dev/tunnel/<machine_name>/<folder_name>`. You can open this URL on a client of your choosing.

4. When opening a vscode.dev URL for the first time on this client, you'll be prompted to log into your GitHub account at a `https://github.com/login/oauth/authorize...` URL. This authenticates you to the tunneling service to ensure you have access to the right set of remote machines.