Configuring network settings for GitHub Copilot - GitHub Docs

5-6 minutes

You can connect to Copilot through an HTTP proxy and use custom certificates.

Who can use this feature?

GitHub Copilot Individual or GitHub Copilot Business.

- JetBrains IDEs
- Visual Studio
- Visual Studio Code

Introduction

By default, GitHub Copilot connects to GitHub's server directly from your environment via a secure HTTPS connection. You don't necessarily need to configure any additional network settings to use Copilot.

Some networks use an HTTP proxy server to intercept Internet traffic before sending it to its intended location. Companies often use an HTTP proxy to detect suspicious traffic or restrict the content entering their networks. If you're working on a corporate network, you may need to configure Copilot to connect via an HTTP proxy.

Configuring proxy settings for GitHub Copilot

GitHub Copilot supports basic HTTP proxy setups. If you need to authenticate to a proxy, GitHub Copilot supports basic authentication or authentication with Kerberos. If the proxy URL starts `https://`, the proxy is not currently supported.

You can configure an HTTP proxy for GitHub Copilot in your chosen editor. To view instructions for your editor, use the tabs at the top of this article.

If you don't configure a proxy directly in your editor, GitHub Copilot checks if a proxy URL is set in any of the following environment variables, listed from highest to lowest priority.

- `HTTPS_PROXY`
- `https_proxy`
- `HTTP_PROXY`
- `http_proxy`

Note

You can use any of these variables to store the URL of a standard HTTP proxy. In standard usage, the `http` and `https` portions of these variables refer to the type of request being made, not the URL of the proxy itself. GitHub Copilot does not follow this convention and uses the URL stored in the variable with the highest priority as the proxy for both HTTP and HTTPS requests.

If you have configured a proxy but are still encountering connection errors, see "Troubleshooting network errors for GitHub Copilot."

Authentication with Kerberos

Kerberos is an authentication protocol that allows users and services to prove their identity to each other. When a user successfully authenticates, an authentication service grants the user a ticket that gives them access to a service for a period of time. Network administrators may prefer Kerberos to basic authentication because it is more secure and doesn't require sending unencrypted credentials.

GitHub Copilot supports authentication to a proxy with Kerberos. To use Kerberos, you must have the appropriate krb5 library for your operating system installed on your machine and an active ticket for the proxy service (either created manually with the `kinit` command or by another application). You can use the `klist` command to check if you have a ticket for the proxy service.

Kerberos uses a service principal name (SPN) to uniquely identify a service instance. By default, the SPN is derived from the proxy URL. For example, if the proxy URL is `http://proxy.example.com:3128`, the SPN is `HTTP/proxy.example.com`.

If the default SPN isn't correct for your proxy, you can override the SPN in VS Code and in JetBrains IDEs. You cannot currently override the default SPN in Visual Studio. However, you can use the environment variable `AGENT_KERBEROS_SERVICE_PRINCIPAL` to override the SPN for Visual Studio and JetBrains IDEs.

Allowing GitHub Copilot to use custom certificates

Copilot can read custom SSL certificates installed on a user's machine. This allows a proxy server to be identified as the intended recipient of Copilot's secure connection, so network traffic can be inspected. Without a custom certificate, an HTTP proxy can be used to monitor, route, and terminate Copilot's connection, but not to inspect the contents of the traffic.

Copilot reads certificates from the operating system's trust store. It also reads extra certificates from the file specified by the standard Node.js environment variable `NODE_EXTRA_CA_CERTS`. For more information, see the Node.js documentation.

Copilot can read certificates regardless of whether a proxy is configured directly on a user's machine. This allows Copilot to support setups such as transparent proxies or Zscaler.

Installing custom certificates

Generally, if you're using company equipment, your company's IT department should have already installed any required certificates on your machine. If you need to install a certificate, see the following instructions.

Warning

Installing a custom certificate is an instruction for your computer to trust the creator of the certificate, potentially allowing the creator to intercept all Internet traffic from your machine. You should be very careful to verify that you are installing the correct certificate.

- For Windows, see Installing the trusted root certificate in the Microsoft documentation.
- For macOS, see Add certificates to a keychain using Keychain Access on Mac in the Keychain Access User Guide.
- For Linux, see Installing a root CA certificate in the trust store in the Ubuntu documentation. Similar instructions should apply to most Linux distributions.

If you have installed a certificate but Copilot isn't detecting it, see "Troubleshooting network errors for GitHub Copilot."