Loading SSH keys automatically on Windows (OpenSSH)

If you use Windows and its built-in OpenSSH client, follow these steps to configure the OpenSSH client so you can use your passphrase-protected SSH key without prompts.

Prerequisites

Windows 10 or later.

Before you begin

Check if you have any existing SSH keys. Refer to the GitHub Docs article, Checking for existing SSH keys .

If you don't have an SSH key:

1. Create one by following the GitHub Docs article, Generating a new SSH key .
2. Add the new SSH key to your GitHub account, by following the GitHub Docs article, Adding a new SSH key to your GitHub account .

Procedure

1. Check if `C:\Users\<YourUserName>\.ssh\config` exists. **Note**: In some applications, such as Git Bash, `.ssh` is a hidden subdirectory.
2. Create `C:\Users\<YourUserName>\.ssh\config` if it doesn't exist.
3. Add the following content to `C:\Users\<YourUserName>\.ssh\config` to set the key to load in the authentication agent and specify its use by the target server. Make sure you add this entry before any global settings marked as `Host *`.

   ```
   Host SERVER_NAME
       IdentitiesOnly yes
       IdentityFile FILE_PATH
   ```

   `SERVER_NAME` is the server that uses the file specified by `IdentityFile`. A sample value is `github.com`. `FILE_PATH` is the fully qualified path to the SSH file you created. A sample value is `C:\Users\<YourUserName>\.ssh\<FILE>`, where `<FILE>` might be `id_rsa`, `id_ecdsa`, `id_ed25519`, or a custom name.

   Example:

   ```
   Host github.com
       IdentitiesOnly yes
       IdentityFile C:/Users/user1/.ssh/id_ed25519
   ```

4. Open the Windows PowerShell, making sure you open it by selecting **Run as Administrator**.
5. Configure the SSH Authentication Agent service so it starts each time you reboot your computer, by running the following command:

```
Get-Service ssh-agent | Set-Service -StartupType Automatic
```

6. Start the service, by running the following command:

   ```
   Start-Service ssh-agent
   ```

7. Check that the service is running, by running the following command and confirming that the `Status` value is `Running`:

   ```
   Get-Service ssh-agent
   ```

8. Load your key file into the `ssh-agent`, replacing `<FILE>` with the actual file name of your key, then type your passphrase, if prompted.

   ```
   ssh-add $env:USERPROFILE\.ssh\<FILE>
   ```

   Example:

   ```
   ssh-add $env:USERPROFILE\.ssh\id_ed25519
   ```

9. Make sure Git uses the Windows OpenSSH client instead of the SSH client included with Git, by using either of the following methods:

   For system-wide configuration, create an environment variable named `GIT_SSH_COMMAND` with a value of `C:/Windows/System32/OpenSSH/ssh.exe`. **Important**: Make sure you use forward slashes in the path.

   To set the configuration for a specific scope, run the following `git config` command in a terminal. Refer to the git config documentation for details. For example:

   ```
   git config --global core.sshCommand C:/Windows/System32/OpenSSH/ssh.exe
   ```

   Important: Make sure you use forward slashes in the path.

You can now use the Unity Package Manager to fetch packages from that Git repository over SSH using your passphrase-protected SSH key.

Additional resources

Loading SSH keys and passphrases automatically on Windows (PuTTY)

Using passphrase-protected SSH keys with SSH Git URLs