PASSWORD MANAGER > ENTWICKLERTOOLS

SSH Agent

SSH Agent

Bitwarden Password Manager desktop app can act as an SSH agent to securely encrypt and store your SSH (Secure Shell) keys for use with:

- Authenticating to servers

- Signing Git commits

- Interacting with SSH based services

The Bitwarden SSH agent will organize and protect your keys in one secure location. SSH keys can be accessed with the desktop app, web app, browser extension, and mobile app. SSH keys can be generated using the desktop app, web app, and browser extension.

> ⓘ **Note**
>
> The SSH Agent requires release version 2025.1.2 or newer.
>
> **macOS**:
> The macOS store builds do not support the SSH agent at this time, the .dmg download can be used for SSH agent support.
>
> **Linux**:
> The Flatpak builds do not support the SSH agent at this time, the Snap download can be used for SSH agent support.

Storing an SSH key

New SSH keys can be created and saved in the Bitwarden desktop app. Bitwarden SSH keys will store:

Field	Description
Key name	The name for your SSH key.
Private key	The private key is sensitive data that will be used by the server to facilitate secure connection. Private key data should be treated with care and kept secure. Users may use Bitwarden to generate a secure, unique private key.
Public key	Portion of the key shared with the server that you will be connecting to.
Fingerprint	A short unique string generated from the public key for identification. For example, SSH-signed git commits can be verified using the fingerprint.

SSH keys stored in the Bitwarden Password Manager will have access to Bitwarden features such as folders, favorites, master password re-prompt, notes, cloning items, attachments, and custom fields.

bitwarden

Create new SSH key

Create a new SSH key using the Bitwarden desktop app, web app, or browser extension. Once created, SSH keys stored in Bitwarden can be accessed from the desktop app, web app, browser extension, and mobile apps.

1. Select the **New** button and choose **SSH key** as the item type.

There are no items to list.

Add item

ADD ITEM

Type

SSH key

Name

GitHub

Private key

-----BEGIN OPENSSH PRIVATE KEY-----
••••••••••••••••••••••••••••••••••••••
••••••••••••••••••••••••••••••••••••••
••••••••••••••••••••••••••••••••••••••
••••••••••••••••••••••••••••••••••••••
••••••••••••••••••••••••••••••••••••••
-----END OPENSSH PRIVATE KEY-----

Public key

ssh-

Fingerprint

SHA256:

Import key from clipboard

Folder

No folder

Favorite

Master password re-prompt

+

Cancel

Create new SSH key on desktop client

> **ⓘ Note**
>
> At this time, Bitwarden can only generate ED25519 type SSH keys.

2. Fill in remaining details such as **Name** and select the 🖫 Save icon once complete.

Organization SSH keys

Organization owned SSH keys are not able to be used in the SSH agent. Individual organization users may create and store SSH keys in their individual vault for authentication. Sharing SSH credentials is not a recommended practice.
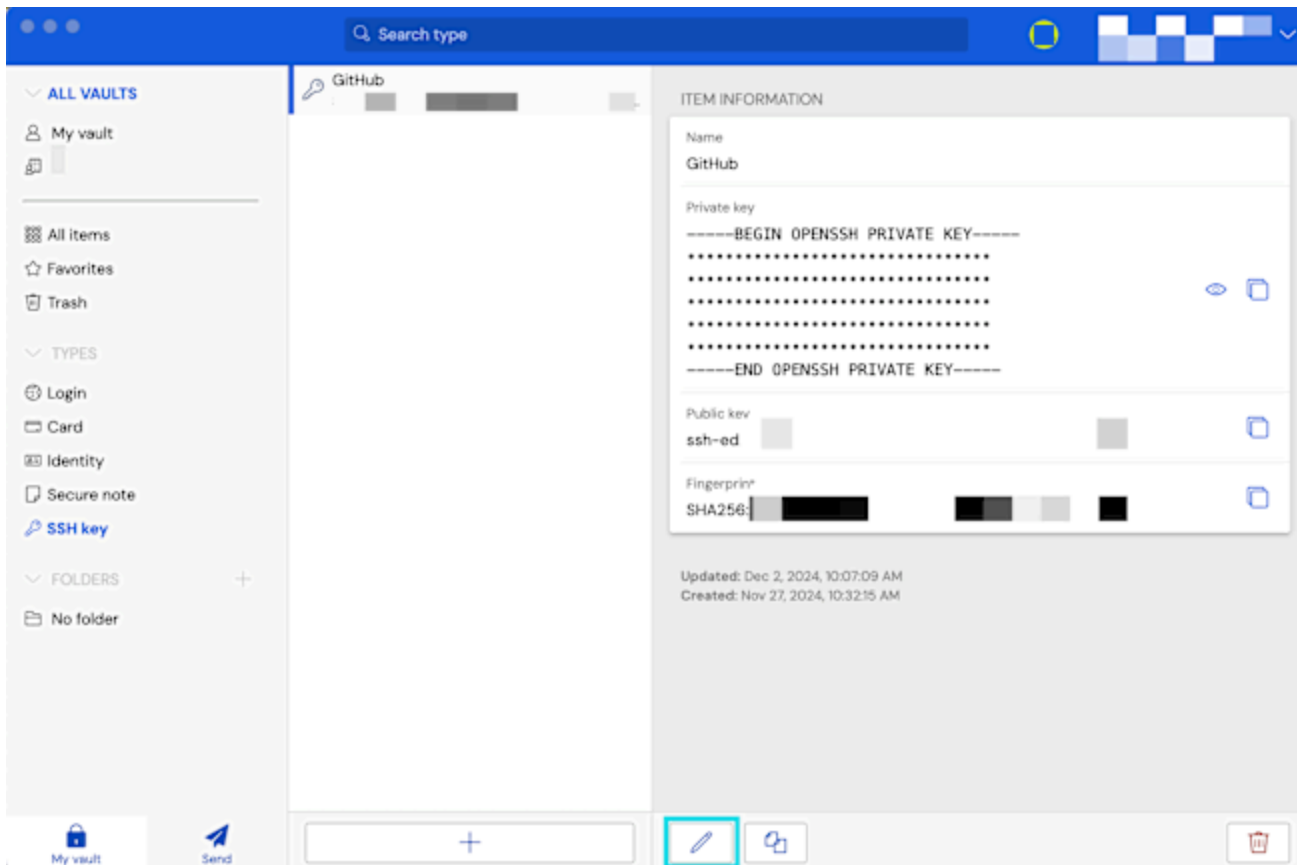
Edit existing keys

Once an SSH key has be saved in your Bitwarden vault, you may edit the key:

⇒Desktop

To edit SSH keys on the Bitwarden desktop app:

1. Open the Bitwarden desktop app and navigate to **SSH keys**.

2. Locate the SSH key you wish to edit and then select ✏ **Edit**.

Edit desktop SSH item

3. Once you have completed the desired changes, select 🖫 **Save**.

⇒Web app

To edit SSH keys on the Bitwarden web app:

1. Open the Bitwarden web app and navigate to **SSH keys**.

2. Locate and select the SSH key you wish to edit. A dialogue will appear on screen, then select **Edit**.

Edit SSH item web app

3. Once you have completed the desired changes, select **Save**.

⇒Mobile

To edit SSH keys on the Bitwarden mobile app:

1. Open the Bitwarden mobile app and navigate to **SSH keys**.

Mobile SSH key vault

2. Locate the SSH key you wish to edit and then select **Edit**.

Select edit SSH key iOS

3. Once you have completed the desired changes, select **Save**.

⇒Browser extension

To edit SSH keys on the Bitwarden browser extension:

1. Open the Bitwarden browser extension and navigate to **SSH keys**.

2. Locate and select the SSH key you wish to edit. A dialogue will appear on screen, then select **Edit**.
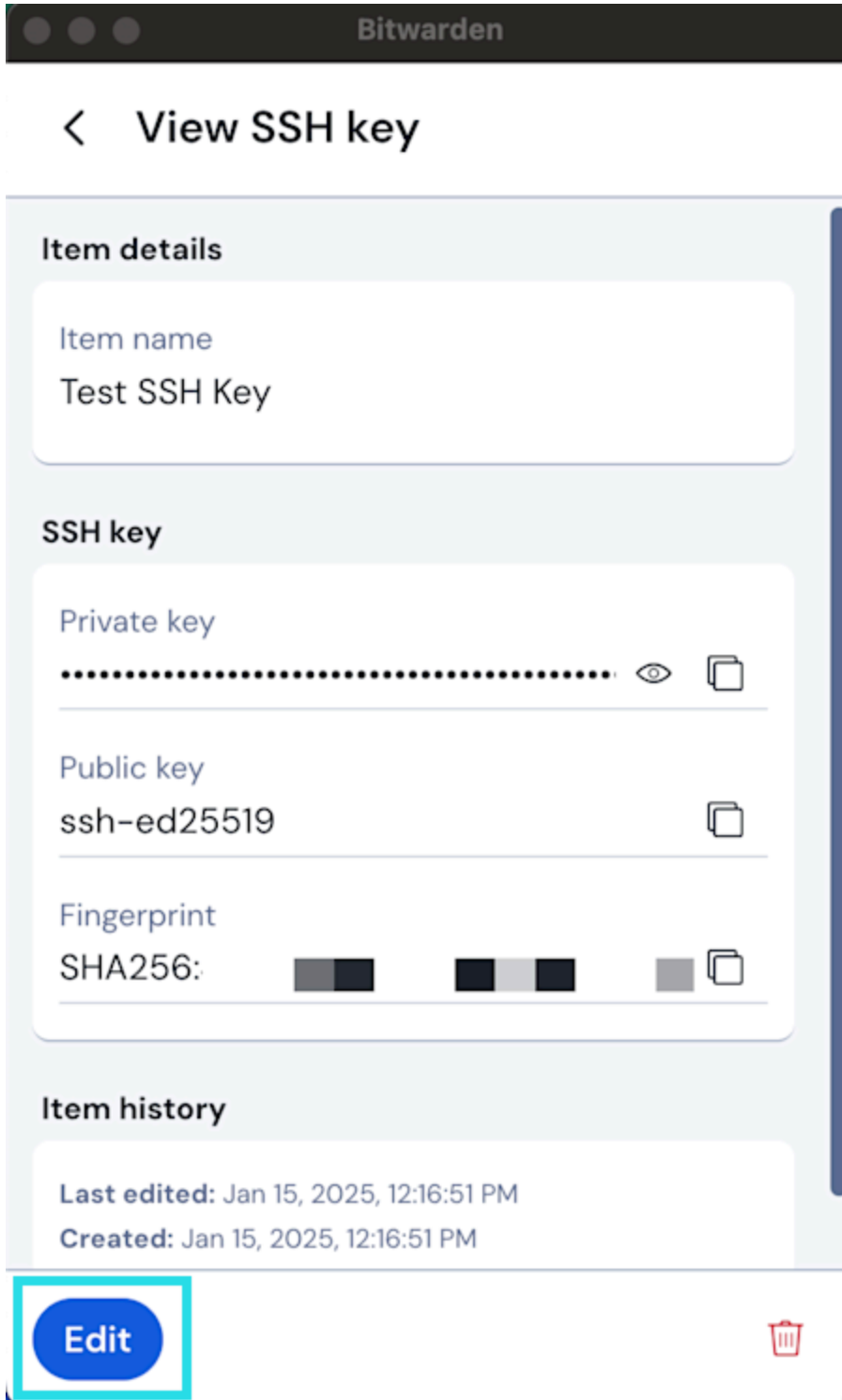
Edit SSH Browser

3. Once you have completed the desired changes, select **Save**.

Import key to Bitwarden

Existing SSH keys can be imported into Bitwarden.

1. Select 🔑 **SSH key** from the navigation menu.

2. Copy the existing SSH key you wish to import into Bitwarden. Use the **Import key from clipboard** option. This will automatically paste the SSH key into Bitwarden.

ADD ITEM

Type

SSH key ⌄

Name

GitHub

Private key

-----BEGIN OPENSSH PRIVATE KEY-----

•••••••••••••••••••••••••••••••••••

•••••••••••••••••••••••••••••••••••

•••••••••••••••••••••••••••••••••••

•••••••••••••••••••••••••••••••••••

•••••••••••••••••••••••••••••••••••

-----END OPENSSH PRIVATE KEY-----

Public key

ssh-

Fingerprint

SHA256:

Import key from clipboard

Import SSH key on desktop client

> ⓘ **Note**
>
> Imported keys must be in **OpenSSH** or **PKCS#8** format
>
> Additionally, at this time, imported SSH keys from Putty are not compatible.

Configure Bitwarden SSH agent

In order to use Bitwarden as your primary SSH agent, you will be required to configure your SSH client to communicate with Bitwarden for authentication.
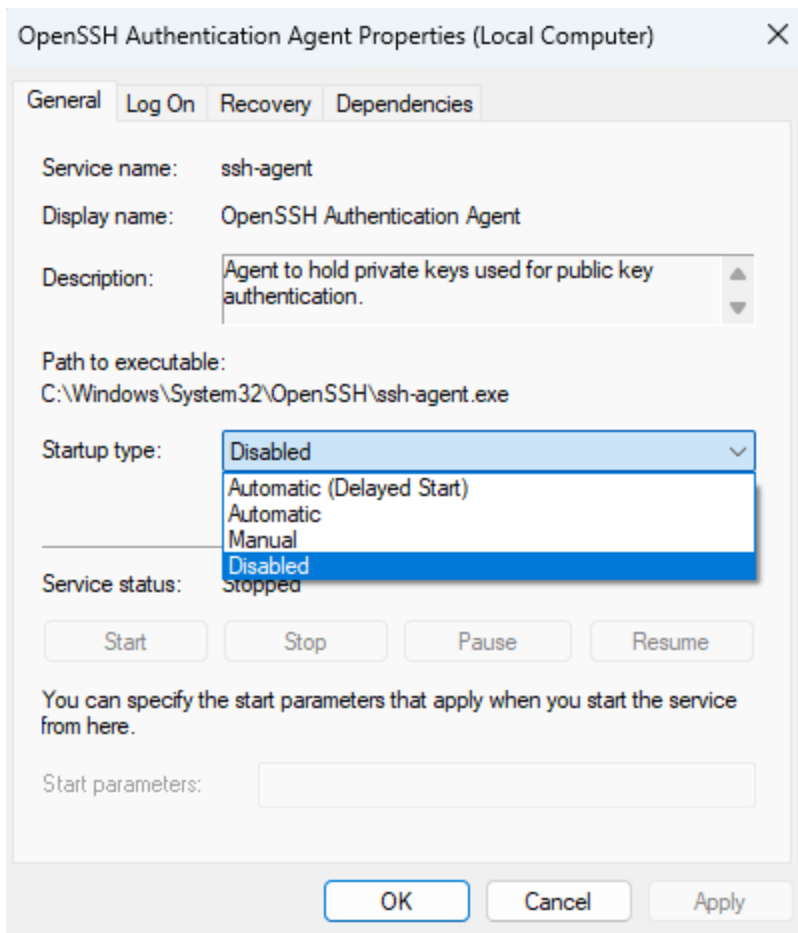
⇒Windows

To enable the Bitwarden SSH agent on Windows, you must disable the OpenSSH service on your Windows machine. To disable OpenSSH:

1. On your Windows machine, navigate to **Services → OpenSSH Authentication Agent**. Services can be located with the Windows search bar.
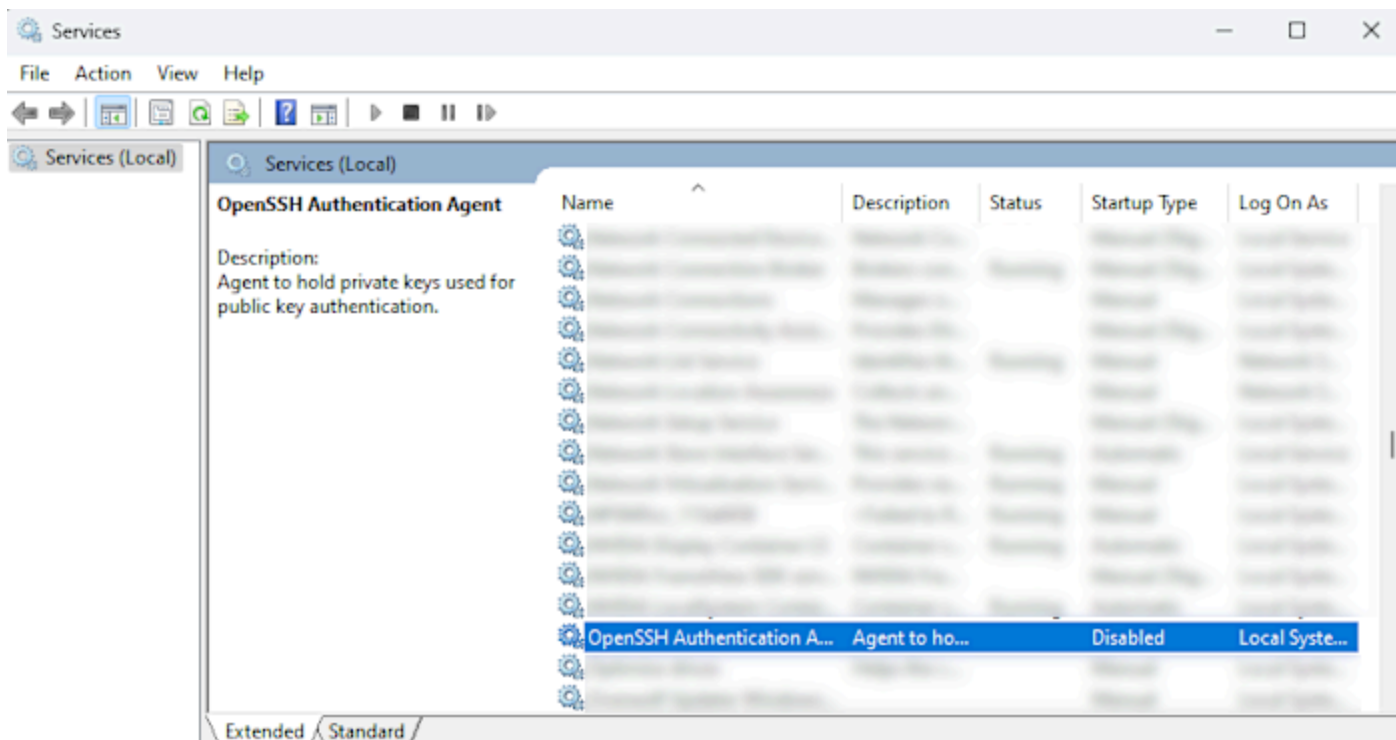
Windows Services panel

2. Once you have opened the OpenSSH Authentication Agent Properties window, set the **Startup type** setting to **Disabled**.

Disable OpenSSH Windows

3. Once the settings have been adjusted, select **Apply** and then **OK**.

> ⓘ **Note**
>
> If OpenSSH Authentication Agent is not an option in the Services list, there is no need to disable the service.

4. To use the SSH agent with Git, configure the `core.sshCommand` variable in your Git config to use Microsoft OpenSSH:

```
Plain Text

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

5. This can also be set using your `gitconfig` file:

```
Plain Text

[core]
   sshCommand = C:/Windows/System32/OpenSSH/ssh.exe
```

⇒macOS

Enable the Bitwarden SSH agent on macOS:

1. Configure the SSH_AUTH_SOCK variable to point to the Bitwarden SSH Agent socket. The following example demonstrates how to do this for macOS after replacing <user> with your username:

```Bash
export SSH_AUTH_SOCK=/Users/<user>/.bitwarden-ssh-agent.sock
```

⇒Linux

Enable the Bitwarden SSH agent on Linux:

1. Configure the SSH_AUTH_SOCK variable to point to the Bitwarden SSH Agent socket. The following example demonstrates how to do this for Linux after replacing <user> with your username:

```Plain Text
export SSH_AUTH_SOCK=/home/<user>/.bitwarden-ssh-agent.sock
```

⇒Snap

Enable the Bitwarden SSH agent on snap installations:

1. Configure the SSH_AUTH_SOCK variable to point to the Bitwarden SSH Agent socket. The following example demonstrates how to do this for snap after replacing <user> with your username:

```Plain Text
export SSH_AUTH_SOCK=/home/<user>/snap/bitwarden/current/.bitwarden-ssh-agent.sock
```

Enable SSH agent

To enable the SSH agent on your Bitwarden desktop app, navigate to **Settings** and **Enable SSH agent**.

✅ Enable SSH agent
Enable the SSH agent to sign SSH requests right from your Bitwarden vault.

Enable SSH storage on desktop client

Testing SSH keys

Once the SSH agent has been configured for Bitwarden, we can test the setup by requesting an SSH list:

```
Plain Text

ssh-add -L
```

This will return a list of SSH keys saved in your Bitwarden desktop client.
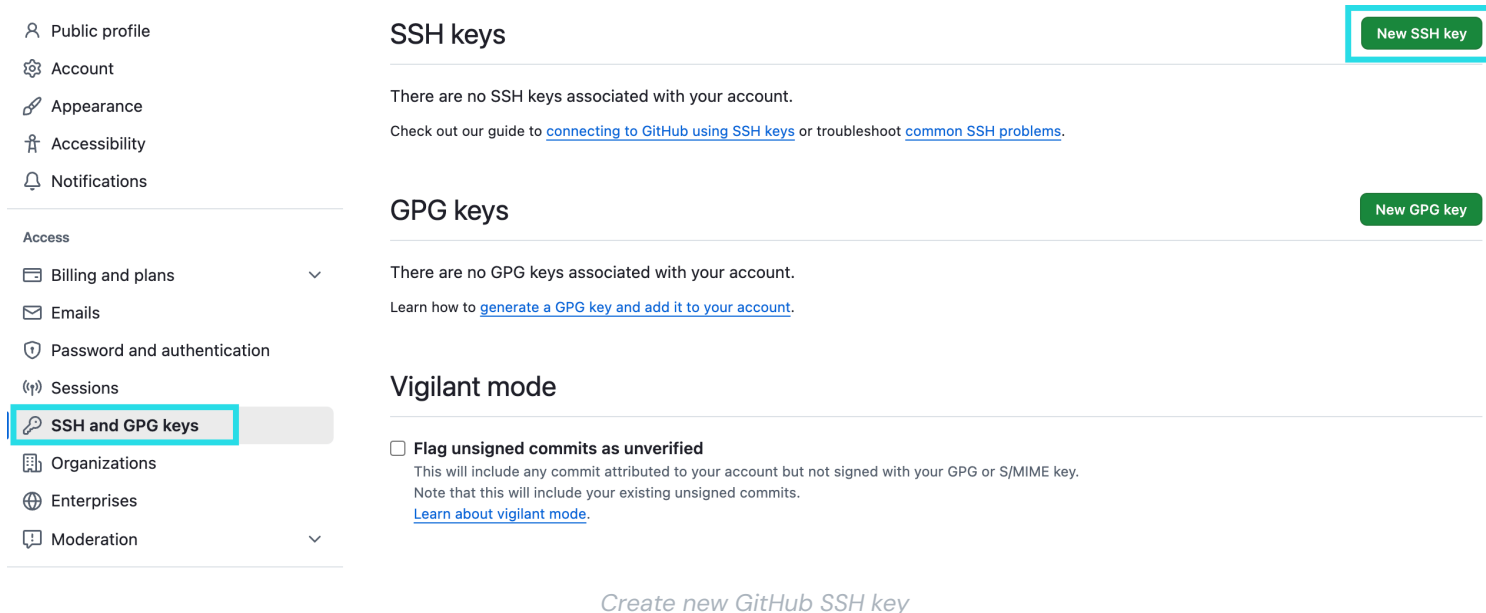
> ⓘ **Note**
>
> When accessing an SSH key, the behavior of Bitwarden will differ depending on the locked or unlocked status of the client.
>
> - **Locked vault**: If your Bitwarden vault is locked, Bitwarden will automatically prompt you to unlock your vault in order to gain access to the SSH key.
>
> - **Unlocked vault**: If the desktop vault is unlocked, you will be prompted to confirm the SSH key usage.

Use SSH key to authenticate with Git

SSH can be used to authenticate with Git. The Bitwarden SSH agent can add security and ease of use to your Git workflows. In this example, the Bitwarden SSH agent will authenticate to GitHub.

1. On your GitHub account, setup an SSH key by navigating to **Settings**, **SSH and GPG keys**, then select **New SSH Key**.

👤 Public profile	**SSH keys** `New SSH key`
⚙ Account	There are no SSH keys associated with your account.
🖌 Appearance	Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.
🧍 Accessibility	
🔔 Notifications	**GPG keys** `New GPG key`
Access	There are no GPG keys associated with your account.
🗔 Billing and plans ⌄	Learn how to generate a GPG key and add it to your account.
✉ Emails	
🛡 Password and authentication	**Vigilant mode**
(ɰ) Sessions	
🔑 **SSH and GPG keys**	☐ Flag unsigned commits as unverified
🏢 Organizations	This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
🌐 Enterprises	Note that this will include your existing unsigned commits.
💬 Moderation ⌄	Learn about vigilant mode.

Create new GitHub SSH key

2. On the add new SSH key screen, add a **Name**, select a **Key type.** Choose `Authentication Key`. Copy & paste the **Public key** from your Bitwarden vault into the **Key** field on GitHub.

Add new SSH Key

Title

GitHub

Key type

Authentication Key ⇅

Key

ssh- ▉ ▉ ▉▉ ▉▉ ▉ ▉▉▉ ▉▉ ▉▉ ▉▉ ▉▉▉ ▉▉ ▉▉ ▉▉ ▉▉ ▉ ▉ ▉▉

Add SSH key

Create new GitHub key

3. Once you have completed all of the fields, select **Add SSH key** to save the key. GitHub will request you verify your GitHub account before the key is saved.

4. Test the GitHub SSH key in your terminal, for example if you are using macOS:

```
Plain Text


ssh git@github.com
```

5. If successful, Bitwarden will prompt you to verify the access request. Select **Authorize** to confirm. If successful, you will receive a message verifying the authentication attempt:

```
Plain Text


Hi <USER>! You've successfully authenticated, but GitHub does not provide shell access.
```

Authenticate with git repositories

Use the Bitwarden SSH agent to sign SSH Git commits. Before using the Bitwarden SSH agent to sign Git commits, your system will require:

- Git version 2.34 or newer. Check your Git version with:

```
Plain Text


git --version
```

- OpenSSH version 8.8 or newer. Check version with:

```
Plain Text

ssh -V
```

- Bitwarden desktop client with SSH agent enabled.

Configure Git for SSH signing

Configure your Git environment to point to your SSH key for signing. To complete this you may set global variables or establish the instructions in your `.gitconfig` file.

Set global variables

To configure Git settings using `--global` variables:

1. Set Git to use SSH for signing:

```
Plain Text

git config --global gpg.format ssh
```

2. Specify the SSH key to use as the singing key. To use the Bitwarden SSH agent, replace `<YOUR_PUBLIC_KEY>` with the public key copied from the SSH key saved in your Bitwarden vault.

```
Plain Text

git config --global user.signingkey "<YOUR_PUBLIC_KEY>"
```

Set `.gitconfig` file

To configure Git using a `.gitconfig` file:

1. Access `.gitconfig` with your preferred text editor:

```
Plain Text

nano ~/.gitconfig
```

2. Add the following configurations:

```bash
[gpg]

        format = ssh
[user]

        signingkey = "<YOUR_PUBLIC_KEY>"

        name = <USER_NAME>

        email = <USER_EMAIL>
[commit]

        gpgsign = true
```
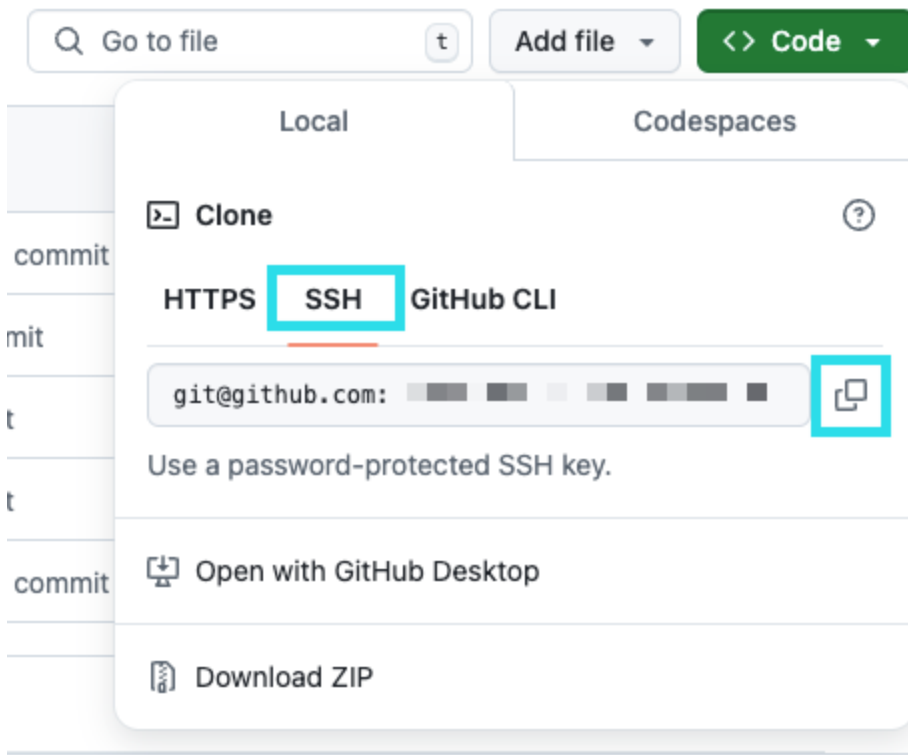
Sign Git commits

Using SSH to authenticate with Git can add security and ease of use to your workflow. Similarly, SSH keys stored in Bitwarden can be used to sign and verify Git commits using SSH protocol. In this example, the Bitwarden SSH agent will be used to sign Git commits to GitHub.

1. On your GitHub account, setup an SSH signing key by navigating to **Settings**, **SSH and GPG keys**, then select **New SSH Key**.

2. On the add new SSH key screen, add a **Name** and select a **Key type**, Choose `Signing Key`. Copy & paste the **Public key** from your Bitwarden vault into the **Key** field on GitHub.

3. Use the SSH key to clone your repository with SSH method:
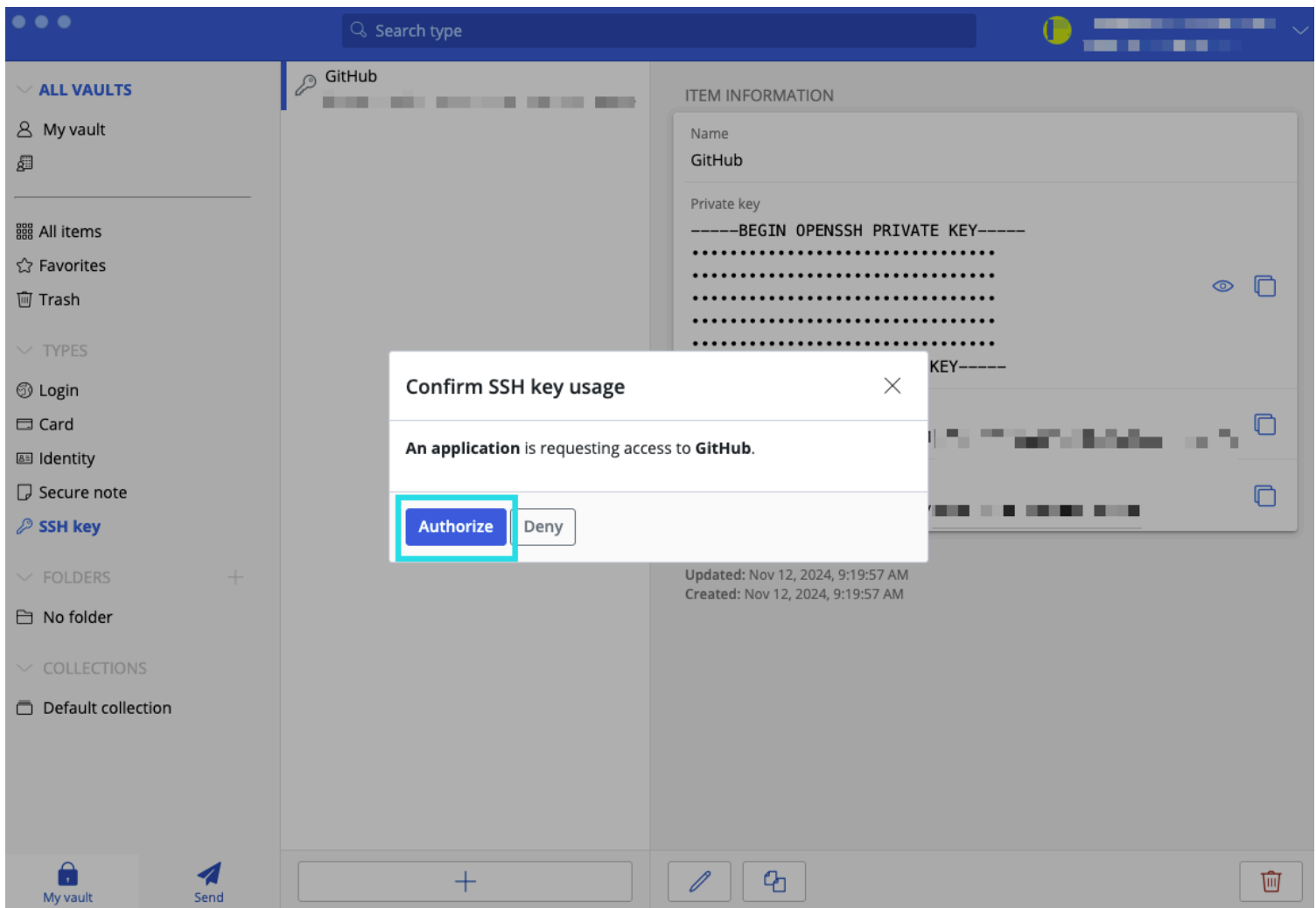

SSH clone

```
Plain Text

git clone git@github.com:<USER>/<repository>.git
```

4. Create the Git commit using terminal or your preferred text editor:

```
Plain Text

git commit -m "This commit is signed using SSH"
```

5. Bitwarden will prompt you to authorize the key usage:

Authorize SSH with client

6. Once authorized, the SSH key will be initiated to approve the commit. You may now push the commit:

```
Plain Text

git push
```

7. You may verify your commit on Github by navigating to GitHub commits: