

Using SSH over the HTTPS port - GitHub Docs

2-3 minutes

Sometimes, firewalls refuse to allow SSH connections entirely. If using [HTTPS cloning with credential caching](#) is not an option, you can attempt to clone using an SSH connection made over the HTTPS port. Most firewall rules should allow this, but proxy servers may interfere.

GitHub Enterprise Server users: Accessing GitHub Enterprise Server via SSH over the HTTPS port is currently not supported.

To test if SSH over the HTTPS port is possible, run this SSH command:

```
$ ssh -T -p 443 git@ssh.github.com
# Hi USERNAME! You've successfully authenticated, but GitHub does not
# provide shell access.
```

Note: The hostname for port 443 is `ssh.github.com`, not `github.com`.

If that worked, great! If not, you may need to [follow our troubleshooting guide](#).

Now, to clone the repository, you can run the following command:

```
git clone ssh://git@ssh.github.com:443/YOUR-USERNAME/YOUR-REPOSITORY.git
```

Enabling SSH connections over HTTPS

If you are able to SSH into `git@ssh.github.com` over port 443, you can override your SSH settings to force any connection to `GitHub.com` to run through that server and port.

To set this in your SSH configuration file, edit the file at `~/.ssh/config`, and add this section:

```
Host github.com
  Hostname ssh.github.com
  Port 443
  User git
```

You can test that this works by connecting once more to `GitHub.com`:

```
$ ssh -T git@github.com
# Hi USERNAME! You've successfully authenticated, but GitHub does not
# provide shell access.
```

Updating known hosts

The first time you interact with GitHub after switching to port 443, you may get a warning message that the host wasn't found in `known_hosts`, or that it was found by another name.

```
# The authenticity of host '[ssh.github.com]:443 ([140.82.112.36]:443)' can't be
established.
# ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
# This host key is known by the following other names/addresses:
#   ~/.ssh/known_hosts:32: github.com
# Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

It is safe to answer "yes" to this question, assuming that the SSH fingerprint matches one of GitHub's published fingerprints. For the list of fingerprints, see "[GitHub's SSH key fingerprints](#)."