Configuring TLS

In this article

You can configure Transport Layer Security (TLS) on your GitHub Enterprise Server instance so that you can use a certificate that is signed by a trusted certificate authority.

About Transport Layer Security

TLS, which replaced SSL, is enabled and configured with a self-signed certificate when GitHub Enterprise Server is started for the first time. As self-signed certificates are not trusted by web browsers and Git clients, these clients will report certificate warnings until you disable TLS or upload a certificate signed by a trusted authority, such as Let's Encrypt.

The GitHub Enterprise Server appliance will send HTTP Strict Transport Security headers when SSL is enabled. Disabling TLS will cause users to lose access to the appliance, because their browsers will not allow a protocol downgrade to HTTP. For more information, see "HTTP Strict Transport Security (HSTS)" on Wikipedia.

Warning: When terminating HTTPS connections on a load balancer, the requests from the load balancer to GitHub Enterprise Server also need to use HTTPS. Downgrading the connection to HTTP is not supported.

To allow users to use FIDO U2F for two-factor authentication, you must enable TLS for your instance. For more information, see "Configuring two-factor authentication."

Prerequisites

To use TLS in production, you must have a certificate in an unencrypted PEM format signed by a trusted certificate authority. To use a certificate signed by an internal certificate authority, you must install the root certificate and any intermediate certificates. For more information, see "Troubleshooting TLS errors."

Your certificate will also need Subject Alternative Names configured for the subdomains listed in "Enabling subdomain isolation" and will need to include the full certificate chain if it has been signed by an intermediate certificate authority. For more information, see "Subject Alternative Name" on Wikipedia.

You can generate a certificate signing request (CSR) for your instance using the `ghe-ssl-generate-csr` command. For more information, see "Command-line utilities."

Your key must be an RSA key and must not have a passphrase. For more information, see "Troubleshooting TLS errors".

Uploading a custom TLS certificate

Warning: Configuring TLS causes a small amount of downtime for your GitHub Enterprise Server instance.

1. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.
2. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
3. In the "🚀 Site admin" sidebar, click **Management Console**.
4. In the "Settings" sidebar, click **Privacy**.
5. Select **TLS only (recommended)**.
6. Under "TLS Protocol support", select the protocols you want to allow.
7. Under "Certificate", click **Choose File**, then choose a TLS certificate or certificate chain (in PEM format) to install. This file will usually have a *.pem*, *.crt*, or *.cer* extension.
8. Under "Unencrypted key", click **Choose File**, then choose an RSA key (in PEM format) to install. This file will usually have a *.key* extension.
9. Under the "Settings" sidebar, click **Save settings**.

   Note: Saving settings in the Management Console restarts system services, which could result in user-visible downtime.

10. Wait for the configuration run to complete.

About Let's Encrypt support

Let's Encrypt is a public certificate authority that issues free, automated TLS certificates that are trusted by browsers using the ACME protocol. You can automatically obtain and renew Let's Encrypt certificates on your appliance without any required manual maintenance.

To use Let's Encrypt automation, your appliance must be configured with a hostname that is publicly accessible over HTTP. The appliance must also be allowed to make outbound HTTPS connections.

When you enable automation of TLS certificate management using Let's Encrypt, your GitHub Enterprise Server instance will contact the Let's Encrypt servers to obtain a certificate. To renew a certificate, Let's Encrypt servers must validate control of the configured domain name with inbound HTTP requests.

You can also use the `ghe-ssl-acme` command line utility on your GitHub Enterprise Server instance to automatically generate a Let's Encrypt certificate. For more information, see "Command-line utilities."

Configuring TLS using Let's Encrypt

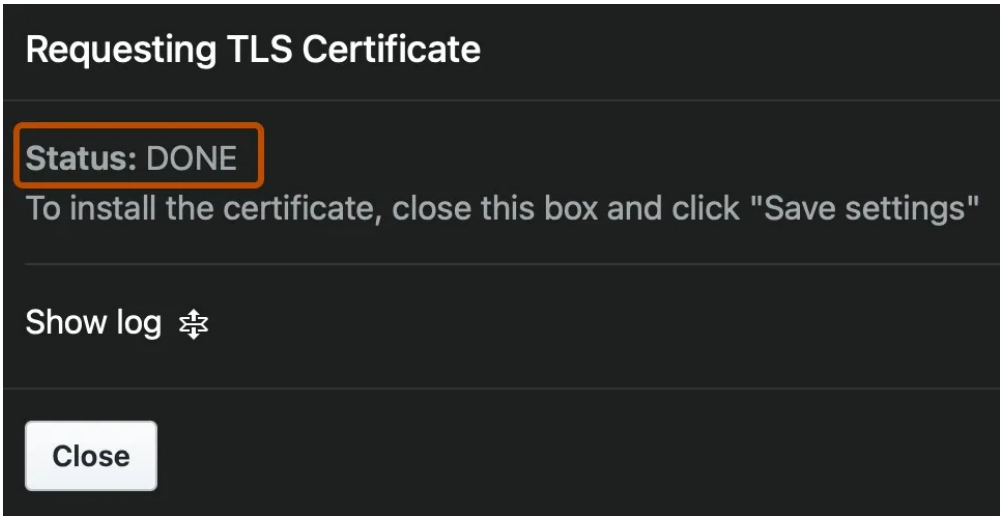
To use Let's Encrypt automation, your appliance must be configured with a hostname that is publicly accessible over HTTP. The appliance must also be allowed to make outbound HTTPS connections.

Warning: Configuring TLS causes a small amount of downtime for your GitHub Enterprise Server instance.

1. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.
2. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
3. In the "🚀 Site admin" sidebar, click **Management Console**.
4. In the "Settings" sidebar, click **Privacy**.
5. Select **TLS only (recommended)**.
6. Select **Enable automation of TLS certificate management using Let's Encrypt**.
7. Under the "Settings" sidebar, click **Save settings**.

   Note: Saving settings in the Management Console restarts system services, which could result in user-visible downtime.

8. Wait for the configuration run to complete.
9. In the "Settings" sidebar, click **Privacy**.
10. Click **Request TLS certificate**.
11. Wait for the "Status" to change from "STARTED" to "DONE".



12. Click **Save configuration**.