Configuring web commit signing

In this article

You can enable auto-signing of commits made in the web interface of GitHub Enterprise Server.

> **Who can use this feature**
> Site administrators can configure web commit signing for your GitHub Enterprise Server instance.

About web commit signing

If you enable web commit signing, GitHub Enterprise Server will automatically use GPG to sign commits users make on the web interface of your GitHub Enterprise Server instance. Commits signed by GitHub Enterprise Server will have a verified status. For more information, see "About commit signature verification."

You can enable web commit signing, rotate the private key used for web commit signing, and disable web commit signing.

Enabling web commit signing

1. In the administrative shell, create a PGP key. Make note of the email address and key ID.

   Bash

   ```
   gpg --full-generate-key --pinentry-mode=loopback
   ```

   - Use the default key type and at least `4096` bits with no expiry.
   - Use `web-flow` as the username. If `web-flow` is unavailable or unusable, use any new unique username. Use this username throughout the following steps in this article.
   - If you have a no-reply email address defined in the Management Console, use that email address. If not, use any email address, such as `web-flow@my-company.com`. The email address does not need to be valid.
   - The PGP key **cannot** be protected by a passphrase.

2. Define the key as an environment variable for GitHub Enterprise Server, replacing `<YOUR-KEY-ID>` with the GPG key ID.

   Bash

```
ghe-config "secrets.gpgverify.web-signing-key" "$(gpg --export-secret-keys -
a <YOUR-KEY-ID> | awk '{printf "%s\\n", $0}')"
```

3. Update the settings for GitHub Enterprise Server's commit signing service.

```
Bash
sudo consul-template -once -template /etc/consul-templates/etc/nomad-
jobs/gpgverify/gpgverify.hcl.ctmpl:/etc/nomad-jobs/gpgverify/gpgverify.hcl

nomad job run /etc/nomad-jobs/gpgverify/gpgverify.hcl
```

4. Enable web commit signing.

```
Bash
ghe-config app.github.web-commit-signing-enabled true
```

5. Create a new user on your GitHub Enterprise Server instance via built-in authentication or external authentication. For more information, see "About authentication for your enterprise."
   - The user's username must be the same username you used when creating the PGP key in step 1 above, for example, `web-flow`.
   - The user's email address must be the same address you used when creating the PGP key.

6. Run the following command, replacing KEY-ID with your PGP key ID.

```
Bash
gpg --armor --export KEY-ID
```

7. Copy your PGP key, beginning with `-----BEGIN PGP PUBLIC KEY BLOCK-----` and ending with `-----END PGP PUBLIC KEY BLOCK-----`.
8. Sign into GitHub Enterprise Server as the user created for web commit signing, for example, `web-flow`.
9. Add the public PGP key to the user's profile. For more information, see "Adding a GPG key to your GitHub account."

> **Note:** Do not remove other public keys from the list of GPG keys. If a public key is deleted, any commits signed with the corresponding private key will no longer be marked as verified.

10. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.
11. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.

12. In the "🚀 Site admin" sidebar, click **Management Console**.
13. In the "Settings" sidebar, click **Email**.
14. Under "No-reply email address", type the same email address you used when creating the PGP key.

    > **Note:** The "No-reply email address" field will only be displayed if you've enabled email for your GitHub Enterprise Server instance. For more information, see "Configuring email for notifications."

15. Under the "Settings" sidebar, click **Save settings**.

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

16. Wait for the configuration run to complete.

Rotating the private key used for web commit signing

1. In the administrative shell, create a PGP key. Make note of the email address and key ID.

   ```Bash
   gpg --full-generate-key --pinentry-mode=loopback
   ```

   - Use the default key type and at least `4096` bits with no expiry.
   - Use the web commit signing user's username, for example, `web-flow`.
   - Use the no-reply email address defined in the Management Console, which should be the same as the email address of the web commit signing user, for example, `web-flow`.
   - The PGP key **cannot** be protected by a passphrase.

2. Define the key as an environment variable for GitHub Enterprise Server, replacing `<YOUR-KEY-ID>` with the GPG key ID.

   ```Bash
   ghe-config "secrets.gpgverify.web-signing-key" "$(gpg --export-secret-keys -a <YOUR-KEY-ID> | awk '{printf "%s\\n", $0}')"
   ```

3. Update the settings for GitHub Enterprise Server's commit signing service.

   ```Bash
   sudo consul-template -once -template /etc/consul-templates/etc/nomad-jobs/gpgverify/gpgverify.hcl.ctmpl:/etc/nomad-jobs/gpgverify/gpgverify.hcl

   nomad job run /etc/nomad-jobs/gpgverify/gpgverify.hcl
   ```

4. Run the following command, replacing KEY-ID with your PGP key ID.

```bash
gpg --armor --export KEY-ID
```

5. Copy your PGP key, beginning with `-----BEGIN PGP PUBLIC KEY BLOCK-----` and ending with `-----END PGP PUBLIC KEY BLOCK-----`.
6. Sign into GitHub Enterprise Server as the user created for web commit signing, for example, `web-flow`.
7. Add the public PGP key to the user's profile. For more information, see "Adding a GPG key to your GitHub account."

> **Note:** Do not remove other public keys from the list of GPG keys. If a public key is deleted, any commits signed with the corresponding private key will no longer be marked as verified.

Disabling web commit signing

You can disable web commit signing for your GitHub Enterprise Server instance.

1. In the administrative shell, run the following command.

```bash
ghe-config app.github.web-commit-signing-enabled false
```

2. Apply the configuration.

```bash
ghe-config-apply
```