



Creating a pre-receive hook environment

In this article

Creating a pre-receive hook environment using Docker

Creating a pre-receive hook environment using chroot

Uploading a pre-receive hook environment on GitHub Enterprise Server

Uploading a pre-receive hook environment via the administrative shell

To execute pre-receive hooks, use either the default pre-receive environment, or create a custom environment.

A pre-receive environment for GitHub Enterprise Server is a Linux chroot environment. Because pre-receive hooks execute on every push event, they should be fast and lightweight. The environment needed for such checks will typically be minimal.

GitHub Enterprise Server provides a default environment which includes these packages: awk, bash, coreutils, curl, find, gnupg, grep, jq, sed.

If you have a specific requirement that isn't met by this environment, such as support for a particular language, you can create and upload your own 64-bit Linux chroot environment.

Creating a pre-receive hook environment using Docker ∂

You can use a Linux container management tool to build a pre-receive hook environment. This example uses <u>Alpine Linux</u> and <u>Docker</u>.

- 1 Ensure Docker is installed locally.
- 2 Create the file Dockerfile.alpine that contains this information:

```
FROM alpine:latest
RUN apk add --no-cache git bash
```

3 From the working directory that contains Dockerfile.alpine, build an image:

```
$ docker build -f Dockerfile.alpine -t pre-receive.alpine .
> Sending build context to Docker daemon 12.29 kB
> Step 1 : FROM alpine:latest
> ---> 8944964f99f4
> Step 2 : RUN apk add --no-cache git bash
> ---> Using cache
> ---> 0250ab3be9c5
> Successfully built 0250ab3be9c5
```

4 Create a container:

5 Export the Docker container to a gzip compressed tar file:

docker export pre-receive.alpine | gzip > alpine.tar.gz

This file alpine.tar.gz is ready to be uploaded to the GitHub Enterprise Server appliance.

Creating a pre-receive hook environment using chroot *₽*

- 1 Create a Linux chroot environment.
- 2 Create a gzip compressed tar file of the chroot directory.

cd /path/to/chroot
tar -czf /path/to/pre-receive-environment.tar.gz .

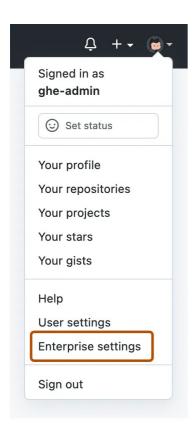
Notes:

- Do not include leading directory paths of files within the tar archive, such as
 /path/to/chroot .
- /bin/sh must exist and be executable, as the entry point into the chroot environment.
- Unlike traditional chroots, the dev directory is not required by the chroot environment for pre-receive hooks.

For more information about creating a chroot environment see "Chroot" from the *Debian Wiki*, "BasicChroot" from the *Ubuntu Community Help Wiki*, or "Installing Alpine Linux in a chroot" from the *Alpine Linux Wiki*.

Uploading a pre-receive hook environment on GitHub Enterprise Server *ℯ*

1 In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



- 2 In the enterprise account sidebar, click & Settings.
- 3 Under "袋 Settings", click Hooks.
- 4 Click Manage environments.
- 6 Click Add environment.
- 6 In the "Environment name" field, enter the desired name.
- In the "Upload environment from a URL" field, enter the URL of the *.tar.gz file that contains your environment.
- 8 Click Add environment.

Uploading a pre-receive hook environment via the administrative shell *₽*

- 1 Upload a readable *.tar.gz file that contains your environment to a web host and copy the URL or transfer the file to the GitHub Enterprise Server appliance via scp. When using scp, you may need to adjust the *.tar.gz file permissions so that the file is world readable.
- 2 Connect to the administrative shell.
- 3 Use the ghe-hook-env-create command and type the name you want for the environment as the first argument and the full local path or URL of a *.tar.gz file that contains your environment as the second argument.

```
admin@ghe-host:~$ ghe-hook-env-create AlpineTestEnv
/home/admin/alpine.tar.gz
> Pre-receive hook environment 'AlpineTestEnv' (2) has been created.
```

Legal

© 2023 GitHub, Inc. <u>Terms</u> <u>Privacy</u> <u>Status</u> <u>Pricing</u> <u>Expert services</u> <u>Blog</u>