Changing the hostname for your instance

In this article

About changes to the hostname for GitHub Enterprise Server

Migrating to an instance with a new hostname

If you want to change the hostname for an existing GitHub Enterprise Server instance, you must restore the settings and data to a new instance.

About changes to the hostname for GitHub Enterprise Server

If you need to use a new hostname for your GitHub Enterprise Server instance, you must back up the existing instance's settings and data, configure a new instance, restore the backup to the new instance, and then adjust your DNS configuration to send traffic to the new instance.

Migration to a new instance requires downtime. The amount of downtime required depends on how much data you need to back up, as well as the speed of the network connection between the backup host and the instances.

In this article, the term "source instance" refers to the instance with the old hostname, and "destination instance" refers to the instance with the new hostname.

> **Warning**: Do not change the hostname for GitHub Enterprise Server after initial setup. Changing the hostname will cause unexpected behavior, up to and including instance outages. If you have changed the hostname for your instance and are experiencing problems, contact GitHub Enterprise Support or GitHub Premium Support.

Migrating to an instance with a new hostname

1. Configure a destination instance of GitHub Enterprise Server with the new hostname you'd like to use. For more information, see the following documentation.
   - "Setting up a GitHub Enterprise Server instance"
   - "Configuring the hostname for your instance"

2. Inform the instance's users of the scheduled downtime. Optionally, you can create a mandatory message that will appear for all users who sign in. For more information, see "Customizing user messages for your enterprise."

3. On the source instance, enable maintenance mode. For more information, see "Enabling and scheduling maintenance mode."

4. Back up the source instance's data and settings using GitHub Enterprise Server Backup Utilities. For more information, see "Configuring backups on your instance."

5. Restore the backup to the destination instance with the desired hostname. When you run the `ghe-restore` utility, use the `-c` option to overwrite the destination instance's configuration. For more information, see "Configuring backups on your instance."

6. Finalize configuration of the destination instance. For more information, see "Configuring GitHub Enterprise."

7. On the destination instance, enable maintenance mode.

8. Optionally, while the destination instance is in maintenance mode, validate the instance's configuration and verify that user data is intact. For more information, see "Enabling and scheduling maintenance mode."

9. To direct traffic to the destination instance, update the DNS `CNAME` record with the source instance's hostname to resolve to the IP address of the destination instance.

   > **Note**: Restored user-generated content in the instance's web application will likely contain URLs that reference the source instance's old hostname. Optionally, to ensure that these links continue to resolve to the destination instance, you can configure a redirect using DNS. In addition to the `CNAME` record that resolves to the new instance's hostname, configure a second DNS `CNAME` record that directs traffic from the original hostname to the new hostname. For more information, see the documentation for your DNS provider.

10. On the destination instance, disable maintenance mode.