

# Using a staging environment

## In this article

About staging environments for GitHub Enterprise Server

Using a staging environment with GitHub Actions

Learn about using GitHub Actions with GitHub Enterprise Server staging instances.

## About staging environments for GitHub Enterprise Server [↗](#)

It can be useful to have a staging or testing environment for your GitHub Enterprise Server instance, so that you can test updates or new features before implementing them in your production environment. For more information, see "[Setting up a staging instance](#)."

## Using a staging environment with GitHub Actions [↗](#)

A common way to create the staging environment is to restore a backup of your production GitHub Enterprise Server instance to a new virtual machine in the staging environment. If you use a staging instance and plan to test GitHub Actions functionality, you should review your storage configuration in the staging environment.

After you restore a GitHub Enterprise Server backup to the staging instance, if you try to view logs or artifacts from existing GitHub Actions workflow runs on your staging instance, you will see `404` errors, because this data will be missing from your staging storage location. To work around the `404` errors, you can copy data from production to use in your staging environment.

## Configuring storage [↗](#)

When you set up a staging environment that includes a GitHub Enterprise Server instance with GitHub Actions enabled, you must use a different external storage configuration for GitHub Actions storage than your production environment.

**Warning:** If you don't change the storage configuration, your staging instance may be able to write to the same external storage that you use for production, which could result in loss of data.

For more information about storage configuration for GitHub Actions, see "[Getting started with GitHub Actions for GitHub Enterprise Server](#)."

## Copying files from production to staging [↗](#)

To more accurately mirror your production environment, you can optionally copy files from your production storage location for GitHub Actions to the staging storage location.

- For an Azure storage account, you can use [azcopy](#). For example:

```
azcopy copy 'https://SOURCE-STORAGE-ACCOUNT-NAME.blob.core.windows.net/SAS-TOKEN' 'https://DESTINATION-STORAGE-ACCOUNT-NAME.blob.core.windows.net/' --recursive
```

- For Amazon S3 buckets, you can use [aws s3 sync](#) . For example:

```
aws s3 sync s3://SOURCE-BUCKET s3://DESTINATION-BUCKET
```

## Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)