Migrating your enterprise to the Container registry from the Docker registry

In this article

You can migrate Docker images previously stored in the Docker registry on your GitHub Enterprise Server instance to the Container registry.

> **Who can use this feature**
> Enterprise owners can migrate Docker images to the Container registry.

Note: Container registry is currently in beta for GitHub Enterprise Server and subject to change.

Both GitHub Packages and subdomain isolation must be enabled to use Container registry. For more information, see "Working with the Container registry."

About the Container registry

The Container registry stores container images within your organization or personal account, and allows you to associate an image with a repository. You can choose whether to inherit permissions from a repository, or set granular permissions independently of a repository. You can also access public container images anonymously. For more information, see "Working with the Container registry."

For more information about configuring GitHub Packages for your GitHub Enterprise Server instance, see "Getting started with GitHub Packages for your enterprise."

About migration from the Docker registry

The Container registry replaces GitHub's Docker registry. If the Docker registry on your GitHub Enterprise Server instance contains images, you must manually migrate the images to the Container registry.

Note: GitHub Enterprise Server 3.6 supports migration for Docker images stored in organizations. A future release will support migration of user-owned images.

Any scripts or GitHub Actions workflows that use the namespace for the Docker registry, `docker.HOSTNAME`, will continue to work after migration to the Container registry at `containers.HOSTNAME`. For more information about the impact of migration to the

Container registry, see "Migrating to the Container registry from the Docker registry."

Migrating organizations to the Container registry

You can start a migration of all your organizations' Docker images to the Container registry. The duration of the migration operation depends on the total number of images to migrate, and the overall load on your instance. After a successful migration, GitHub Enterprise Server will display a summary, and all future uploads of Docker images will use the Container registry.

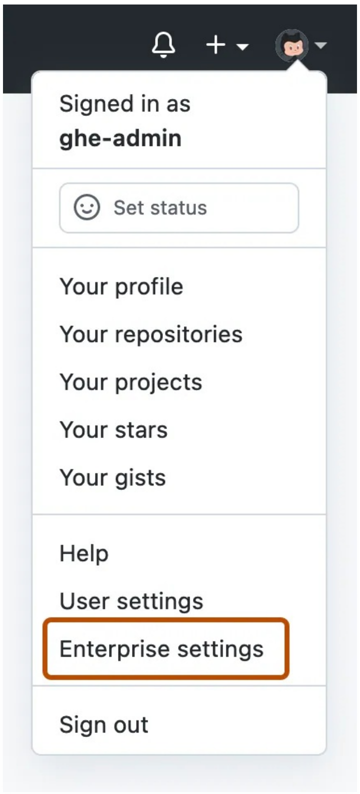
If a site administrator has configured email notifications for your GitHub Enterprise Server instance, you will receive an email after the migration is complete. For more information, see "Configuring email for notifications."

Notes:

- During the migration, the CPU and memory usage for your instance will increase. To ensure the performance of the instance for your users, GitHub recommends that you begin a migration during a period of reduced activity.
- During the migration, do not modify settings for your enterprise or run `ghe-config-apply` from an administrative SSH session. These actions will trigger a configuration run, which can restart services and may interrupt the migration.
- After the migration, storage pressure on your instance will increase due to the duplication of image files in the Docker registry and the Container registry. A future release of GitHub Enterprise Server will remove the duplicated files when all migrations are complete.

For more information about monitoring the performance and storage of your GitHub Enterprise Server instance, see "Accessing the monitor dashboard."

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. In the left sidebar, click **Packages**.

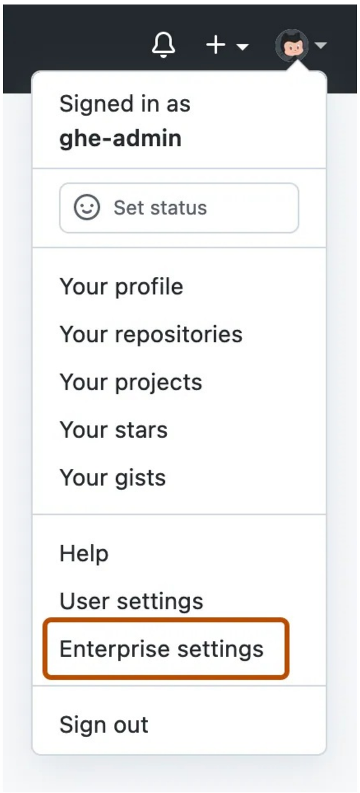
4. To the right of the number of packages to migrate, click **Start migration**. During the migration, GitHub Enterprise Server will display progress on this page.

After the migration completes, the page will display the results. If a migration fails, the page will show the organizations that own the package that caused the failure.

Re-running a failed organization migration

Prior to migration, if a user has created a package in the Container registry that has an identical name to an existing package in the Docker registry, the migration will fail.

1. Delete the affected container in the Container registry. For more information, see "Deleting and restoring a package."
2. In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



Signed in as
ghe-admin

3. In the enterprise account sidebar, click ⚙ **Settings**.
4. In the left sidebar, click **Packages**.
5. To the right of the number of packages to migrate, click **Re-run migration**. During the migration, GitHub Enterprise Server will display progress on this page.
6. If the migration fails again, start from step 1 and re-run the migration.

Legal

Terms Privacy Status Pricing Expert services Blog