Migrating to the Container registry from the Docker registry

In this article

About the Container registry

About migration from the Docker registry

Further reading

An enterprise owner can migrate Docker images previously stored in the Docker registry on your GitHub Enterprise Server instance 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."

About migration from the Docker registry 🔗

The Container registry replaces GitHub's Docker registry. If you've stored Docker images in the Docker registry, an enterprise owner will gradually migrate the images to the Container registry. No action is required on your part.

> **Note**: GitHub Enterprise Server 3.6 supports migration for Docker images stored in organizations. A future release will support migration of user-owned images. For more information about finding the version of GitHub Enterprise Server that you use, see "About versions of GitHub Docs."

After a Docker image has been migrated to the Container registry, you'll see the following changes to the details for the package.

- The package icon will be the Container registry logo (a 📦 icon) instead of the Docker logo.
- The domain in the pull URL will be `containers.HOSTNAME` instead of `docker.HOSTNAME`.

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`.

After migration, you'll no longer be able to use the GraphQL API to query for packages with a `PackageType` of "DOCKER". Instead, you can use the REST API to query for packages with a `package_type` of "container". For more information, see "[Packages](#)" in the REST API documentation.

Further reading 🔗

- "[Migrating your enterprise to the Container registry from the Docker registry](#)"