Getting started with GitHub Packages for your enterprise

In this article

You can start using GitHub Packages on your GitHub Enterprise Server instance by enabling the feature, configuring third-party storage, configuring the ecosystems you want to support, and updating your TLS certificate.

Who can use this feature
Site administrators can enable and configure GitHub Packages.

> **Note:** GitHub Packages on GitHub Enterprise Server does not currently support clustering.

Step 1: Check whether GitHub Packages is available for your enterprise 🔗

GitHub Packages is available in GitHub Enterprise Server 3.0 or higher. If you're using an earlier version of GitHub Enterprise Server, you'll have to upgrade to use GitHub Packages. For more information about upgrading your GitHub Enterprise Server instance, see "About upgrades to new releases."

Step 2: Review hardware requirements 🔗

If you plan to enable Container registry for the users of your instance, at least 10% more CPU resources are required.

We recommend reviewing the levels of activity for users and automations on the instance to ensure that you have provisioned adequate CPU for your users. For more information, see "Monitoring your instance."

For more information about minimum hardware requirements for your GitHub Enterprise Server instance, see the hardware considerations for your instance's platform.

- AWS
- Azure
- Google Cloud Platform
- Hyper-V

- [OpenStack KVM](#)
- [VMware](#)

For more information about adjusting resources for an existing instance, see "[Increasing CPU or memory resources](#)."

Step 3: Enable GitHub Packages and configure external storage 🔗

GitHub Packages on GitHub Enterprise Server uses external blob storage to store your packages.

After enabling GitHub Packages for your GitHub Enterprise Server instance, you'll need to prepare your third-party storage bucket. The amount of storage required depends on your usage of GitHub Packages, and the setup guidelines can vary by storage provider.

Supported external storage providers

- Amazon Web Services (AWS) S3
- Azure Blob Storage
- MinIO

To enable GitHub Packages and configure third-party storage, see:

- "[Enabling GitHub Packages with AWS](#)"
- "[Enabling GitHub Packages with Azure Blob Storage](#)"
- "[Enabling GitHub Packages with MinIO](#)"

Step 4: Specify the package ecosystems to support on your instance 🔗

Choose which package ecosystems you'd like to enable, disable, or set to read-only on your GitHub Enterprise Server instance. Available options are Container registry, Docker, RubyGems, npm, Apache Maven, Gradle, or NuGet. For more information, see "[Configuring package ecosystem support for your enterprise](#)."

Step 5: Ensure you have a TLS certificate for your package host URL, if needed 🔗

If subdomain isolation is enabled for your GitHub Enterprise Server instance, you will need to create and upload a TLS certificate that allows the package host URL for each ecosystem you want to use, such as `containers.HOSTNAME`. Make sure each package host URL includes `https://`.

You can create the certificate manually, or you can use *Let's Encrypt*. If you already use *Let's Encrypt*, you must request a new TLS certificate after enabling GitHub Packages. For more information about package host URLs, see "[Enabling subdomain isolation](#)." For more information about uploading TLS certificates to GitHub Enterprise Server, see "[Configuring TLS](#)."

Step 6: Check for and rename reserved names 🔗

If you want to use the Docker ecosystem with subdomain isolation disabled, you **must** first rename any user or organization named `v2` on your GitHub Enterprise Server instance, prior to enabling Docker ecosystem support in the Management Console. Docker uses a `v2` account name to manage path conflicts with the Docker API, and once

Docker registry support is enabled, you won't be able to use this name anymore.

You can view a full list of logins reserved for internal use by navigating to the "Reserved logins" page in the Site admin dashboard. For more information, see "[Site admin dashboard](#)."