Enterprise administrators / Packages / Configure package ecosystems

Configuring package ecosystem support for your enterprise

In this article

You can configure GitHub Packages for your enterprise by globally enabling or disabling individual package ecosystems on your enterprise, including Container registry, Docker, and npm. Learn about other configuration requirements to support specific package ecosystems.

> **Who can use this feature**
> Site administrators can enable GitHub Packages and configure enterprise settings.

About package ecosystem support

You can 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.

Prerequisites

Before you can configure individual package ecosystems, GitHub Packages must be enabled and configured for your instance. For more information, see "Getting started with GitHub Packages for your enterprise."

Enabling or disabling individual package ecosystems

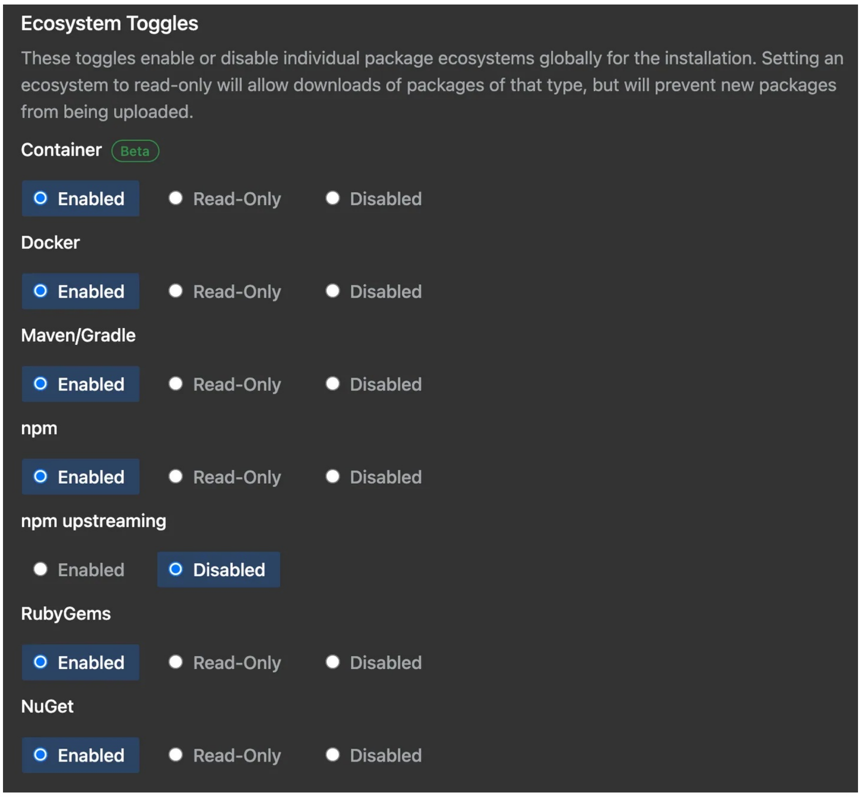
To prevent new packages from being uploaded, you can set an ecosystem you previously enabled to **Read-Only**, while still allowing existing packages to be downloaded.

1. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.
2. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
3. In the "🚀 Site admin" sidebar, click **Management Console**.

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

5. Under "Ecosystem Toggles", for each package type, select **Enabled**, **Read-Only**, or **Disabled**.

> **Note**: Subdomain isolation must be enabled to toggle the Container registry options.



Ecosystem Toggles

These toggles enable or disable individual package ecosystems globally for the installation. Setting an ecosystem to read-only will allow downloads of packages of that type, but will prevent new packages from being uploaded.

Container Beta

[✓ Enabled] Read-Only Disabled

Docker

[✓ Enabled] Read-Only Disabled

Maven/Gradle

[✓ Enabled] Read-Only Disabled

npm

[✓ Enabled] Read-Only Disabled

npm upstreaming

Enabled [✓ Disabled]

RubyGems

[✓ Enabled] Read-Only Disabled

NuGet

[✓ Enabled] Read-Only Disabled

6. Under the "Settings" sidebar, click **Save settings**.

> **Note:** Saving settings in the Management Console restarts system services, which could result in user-visible downtime.

7. Wait for the configuration run to complete.

Connecting to the official npm registry

If you've enabled npm packages on your enterprise and want to allow access to the official npm registry as well as the GitHub Packages npm registry, then you must perform some additional configuration.

GitHub Packages uses a transparent proxy for network traffic that connects to the official npm registry at `registry.npmjs.com`. The proxy is enabled by default and cannot be disabled.

To allow network connections to the npm registry, you will need to configure network ACLs that allow GitHub Enterprise Server to send HTTPS traffic to `registry.npmjs.com` over port 443/TCP.

Note that connections to `registry.npmjs.com` traverse through the Cloudflare network, and subsequently do not connect to a single static IP address; instead, a connection is made to an IP address within the CIDR ranges listed here: https://www.cloudflare.com/ips/.

If you wish to enable npm upstream sources, select `Enabled` for `npm upstreaming`.

Next steps

As a next step, we recommend you check if you need to update or upload a TLS certificate for your packages host URL. For more information, see "Getting started with GitHub Packages for your enterprise."