Configuring Dependabot to work with limited internet access

In this article

You can configure Dependabot to generate pull requests for version and security updates using private registries when your GitHub Enterprise Server instance has limited, or no, internet access.

About Dependabot updates

You can use Dependabot updates to fix vulnerabilities and keep dependencies updated to the latest version in your GitHub Enterprise Server instance. Dependabot updates require GitHub Actions with self-hosted runners set up for Dependabot to use. Dependabot alerts and security updates use information from the GitHub Advisory Database accessed using GitHub Connect. For more information, see "Managing self-hosted runners for Dependabot updates on your enterprise" and "Enabling Dependabot for your enterprise."

Dependabot can access public registries by default, and you can configure Dependabot to also access private registries. Alternatively, if your GitHub Enterprise Server instance has limited or no internet access, you can configure Dependabot to use only private registries as a source for security and version updates. For information on which ecosystems are supported as private registries, see "Removing Dependabot access to public registries."

The instructions below assume that you need to set up Dependabot runners with the following limitations.

- No internet access.
- Access to limited internal resources, such as private registries for Dependabot.

Restricting internet access for Dependabot runners

Before configuring Dependabot, install Docker on your self-hosted runner. For more information, see "Managing self-hosted runners for Dependabot updates on your enterprise."

1. On your GitHub Enterprise Server instance, navigate to the `github/dependabot-action` repository and retrieve information about the `dependabot-updater` and `dependabot-proxy` container images from the `containers.json` file.

Each release of GitHub Enterprise Server includes an updated `containers.json` file at: `https://HOSTNAME/github/dependabot-action/blob/ghes-VERSION/docker/containers.json`. You can see the GitHub.com version of the file at: containers.json.

2. Preload all the container images from the GitHub Container registry onto the Dependabot runner using the `docker pull` command. Alternatively, preload the `dependabot-proxy` image and then preload only the container images for the ecosystems you require.

   For example, to support npm and GitHub Actions you could use the following commands, copying details of the images to load from the `containers.json` file to ensure that you have the correct version and SHA for each image.

   ```
   docker pull ghcr.io/github/dependabot-update-job-proxy/dependabot-update-job-proxy:VERSION@SHA
   docker pull ghcr.io/dependabot/dependabot-updater-github-actions:VERSION@SHA
   docker pull ghcr.io/dependabot/dependabot-updater-npm:VERSION@SHA
   ```

   Note: You will need to repeat this step when you upgrade to a new minor version of GitHub Enterprise Server, or if you manually update the Dependabot action from GitHub.com. For more information, see "Manually syncing actions from GitHub.com."

3. When you have finished adding these images to the runner, you are ready to restrict internet access to the Dependabot runner, ensuring that it can still access your private registries for the required ecosystems and for your GitHub Enterprise Server instance.

   You must add the images first because Dependabot runners pull `dependabot-updater` and `dependabot-proxy` from the GitHub Container registry when Dependabot jobs start running.

Verifying the configuration of Dependabot runners

1. For a test repository, configure Dependabot to access private registries and remove access to public registries. For more information, see "Configuring access to private registries for Dependabot" and "Removing Dependabot access to public registries."
2. In the **Insights** tab for the repository, click **Dependency graph** to display details of the dependencies.
3. Click **Dependabot** to display the ecosystems configured for version updates.
4. For ecosystems that you want to test, click **Last checked TIME ago** to display the "Update logs" view.
5. Click **Check for updates** to check for new updates to dependencies for that ecosystem.

When the check for updates is complete, you should check the "Update logs" view to verify that Dependabot accessed the configured private registries on your GitHub Enterprise Server instance to check for version updates.

After you have verified that the configuration is correct, ask repository administrators to update their Dependabot configurations to use private registries only. For more information, see "Removing Dependabot access to public registries."