Configuring access to private registries for Dependabot

In this article

You can configure Dependabot to access dependencies stored in private registries. You can store authentication information, like passwords and access tokens, as encrypted secrets and then reference these in the Dependabot configuration file.

About private registries

Dependabot version updates keeps your dependencies up-to-date. Dependabot can access public registries. In addition, you can give Dependabot version updates access to private package registries and private GitHub repositories so that you can keep your private and innersource dependencies as up-to-date as your public dependencies.

In most ecosystems, private dependencies are usually published to private package registries. These private registries are similar to their public equivalents, but they require authentication.

For specific ecosystems, you can configure Dependabot to access *only* private registries by removing calls to public registries. For more information, see "Removing Dependabot access to public registries."

Configuring private registries

You configure Dependabot's access to private registries in the `dependabot.yml` file. The top-level `registries` key is optional and specifies authentication details. You can allow all of the defined registries to be used by setting `registries` to `"*"`. Alternatively, you can list the registries that the update can use. To do this, use the name of the registry as defined in the top-level `registries` section of the `dependabot.yml` file.

You use the following options to specify access settings. Registry settings must contain a `type` and a `url`, and typically either a `username` and `password` combination or a `token`.

Option	Description
`type`	Identifies the type of registry. See the full list of types below.
`url`	The URL to use to access the dependencies in this registry. The protocol is optional. If not specified, `https://` is assumed. Dependabot adds or ignores trailing slashes as required

`username`	The username that Dependabot uses to access the registry. `username` is the username or email address for the account.
`password`	A reference to a Dependabot secret containing the password for the specified user. For more information, see "Configuring access to private registries for Dependabot." `password` is the password for the account specified by the username. If the account is a GitHub account, you can use a GitHub personal access token in place of the password.
`key`	A reference to a Dependabot secret containing an access key for this registry. For more information, see "Configuring access to private registries for Dependabot."
`token`	A reference to a Dependabot secret containing an access token for this registry. For more information, see "Configuring access to private registries for Dependabot." `token` is used to provide an access token for an external system and should not be used to provide a GitHub personal access token. If you want to use a GitHub personal access token, you should supply it as a password.
`replaces-base`	For registries, if the boolean value is `true`, Dependabot will resolve dependencies by using the specified URL rather than the base URL of that specific ecosystem. For example, for registries with `type: python-index`, if the boolean value is `true`, pip resolves dependencies by using the specified URL rather than the base URL of the Python Package Index (by default `https://pypi.org/simple`).

For more information about the configuration options that are available, how to use them, and about the supported types, see "Configuration options for the dependabot.yml file."

Storing credentials for Dependabot to use

To give Dependabot access to the private registries supported by GitHub, you store the registry's access token or secret in the secret store for your repository or organization.

About encrypted secrets for Dependabot

Dependabot secrets are encrypted credentials that you create at either the organization level or the repository level. When you add a secret at the organization level, you can specify which repositories can access the secret. You can use secrets to allow Dependabot to update dependencies located in private package registries. When you add a secret, it's encrypted before it reaches GitHub and it remains encrypted until it's used by Dependabot to access a private package registry.

Dependabot secrets also include secrets that are used by GitHub Actions workflows triggered by Dependabot pull requests. Dependabot itself may not use these secrets, but the workflows require them. For more information, see "Automating Dependabot with GitHub Actions."

After you add a Dependabot secret, you can reference it in the `dependabot.yml` configuration file like this: `${{secrets.NAME}}`, where "NAME" is the name you chose for the secret. For example:

```
password: ${{secrets.MY_ARTIFACTORY_PASSWORD}}
```

For more information, see "Configuration options for the dependabot.yml file."

Naming your secrets

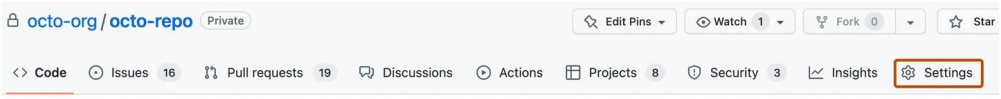
The name of a Dependabot secret:

- Can only contain alphanumeric characters (`[A-Z]`, `[0-9]`) or underscores (`_`). Spaces are not allowed. If you enter lowercase letters these are changed to uppercase.
- Must not start with the `GITHUB_` prefix.
- Must not start with a number.

Adding a repository secret for Dependabot

To create secrets for a personal account repository, you must be the repository owner. To create secrets for an organization repository, you must have `admin` access.

1. On your GitHub Enterprise Server instance, navigate to the main page of the repository.
2. Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the ••• dropdown menu, then click **Settings**.



3. In the "Security" section of the sidebar, select **Secrets and variables**, then click **Dependabot**.
4. Click **New repository secret**.
5. Type a name for your secret in the **Name** input box.
6. Enter the value for your secret.
7. Click **Add secret**.

   The name of the secret is listed on the Dependabot secrets page. You can click **Update** to change the secret value. You can click **Remove** to delete the secret.

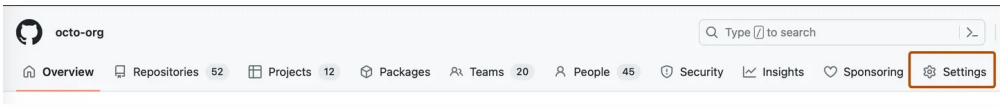
Adding an organization secret for Dependabot

When creating a secret in an organization, you can use a policy to limit which repositories can access that secret. For example, you can grant access to all repositories, or limit access to only private repositories or a specified list of repositories.

To create secrets at the organization level, you must have `admin` access.

1. On your GitHub Enterprise Server instance, navigate to the main page of the organization.

2. Under your organization name, click ⚙ **Settings**. If you cannot see the "Settings" tab, select the ••• dropdown menu, then click **Settings**.



3. In the "Security" section of the sidebar, select ✳ **Secrets and variables**, then click **Dependabot**.
4. Click **New organization secret**.
5. Type a name for your secret in the **Name** input box.
6. Enter the **Value** for your secret.
7. From the **Repository access** dropdown list, choose an access policy.
8. If you chose **Selected repositories**:
   - Click ⚙.
   - In the dialog box, select the repositories that can access this secret.
   - Click **Update selection**.
9. Click **Add secret**.

   The name of the secret is listed on the Dependabot secrets page. You can click **Update** to change the secret value or its access policy. You can click **Remove** to delete the secret.