Defining custom patterns for secret scanning

In this article

You can extend secret scanning to detect secrets beyond the default patterns.

Secret scanning is available for organization-owned repositories in GitHub Enterprise Server if your enterprise has a license for GitHub Advanced Security. For more information, see "About secret scanning" and "About GitHub Advanced Security."

About custom patterns for secret scanning

You can define custom patterns to identify secrets that are not detected by the default patterns supported by secret scanning. For example, you might have a secret pattern that is internal to your organization. For details of the supported secrets and service providers, see "Secret scanning patterns."

You can define custom patterns for your enterprise, organization, or repository. Secret scanning supports up to 500 custom patterns for each organization or enterprise account, and up to 100 custom patterns per repository.

You can also enable push protection for custom patterns. For more information about push protection, see "Push protection for repositories and organizations."

Regular expression syntax for custom patterns

You can specify custom patterns for secret scanning as one or more regular expressions.

- **Secret format:** an expression that describes the format of the secret itself.
- **Before secret:** an expression that describes the characters that come before the secret. By default, this is set to `\A|[^0-9A-Za-z]` which means that the secret must be at the start of a line or be preceded by a non-alphanumeric character.
- **After secret:** an expression that describes the characters that come after the secret. By default, this is set to `\z|[^0-9A-Za-z]` which means that the secret must be followed by a new line or a non-alphanumeric character.
- **Additional match requirements:** one or more optional expressions that the secret

itself must or must not match.

For simple tokens you will usually only need to specify a secret format. The other fields provide flexibility so that you can specify more complex secrets without creating complex regular expressions. For an example of a custom pattern, see "Example of a custom pattern specified using additional requirements" below.

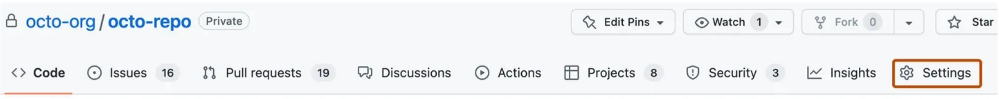
Secret scanning uses the Hyperscan library and only supports Hyperscan regex constructs, which are a subset of PCRE syntax. Hyperscan option modifiers are not supported. For more information on Hyperscan pattern constructs, see "Pattern support" in the Hyperscan documentation.

Defining a custom pattern for a repository

Before defining a custom pattern, you must ensure that secret scanning is enabled on your repository. For more information, see "Configuring secret scanning for your repositories."

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, click **Code security and analysis**.

4. Under "Code security and analysis", find "GitHub Advanced Security."

5. Under "Secret scanning", under "Custom patterns", click **New pattern**.

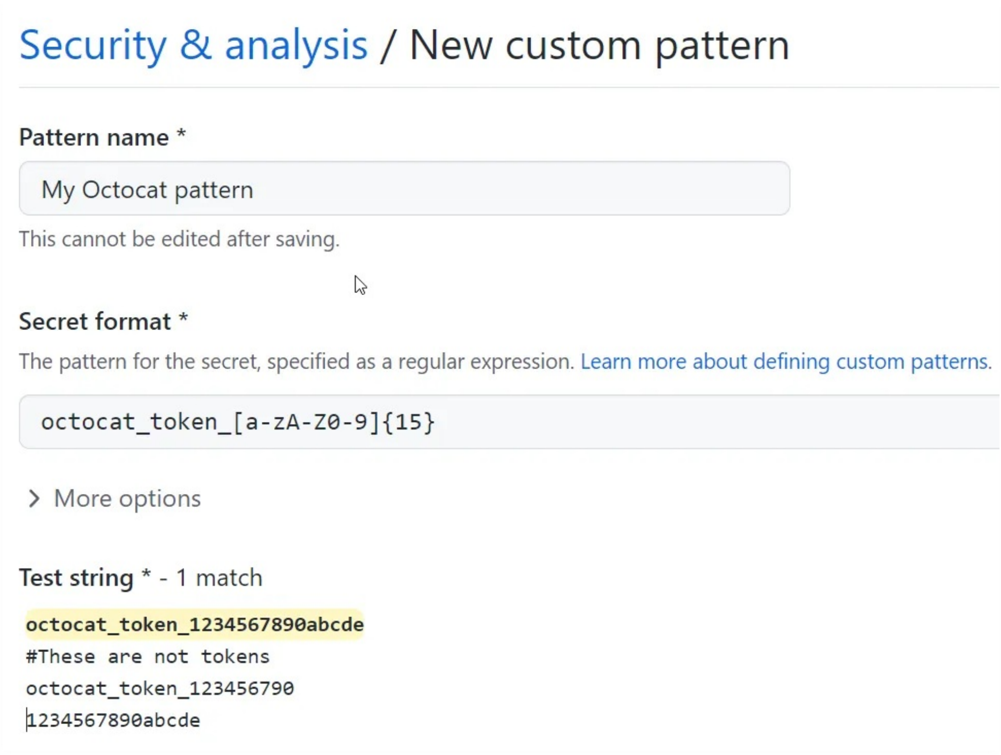
6. Enter the details for your new custom pattern. You must at least provide the name for your pattern, and a regular expression for the format of your secret pattern.

   a. In the "Pattern name" field, type a name for your pattern.

   b. In the "Secret format" field, type a regular expression for the format of your secret pattern.

   c. You can click **More options** ⌄ to provide other surrounding content or additional match requirements for the secret format.

   d. Provide a sample test string to make sure your configuration is matching the patterns you expect.



Security & analysis / New custom pattern

Pattern name *

My Octocat pattern

This cannot be edited after saving.

Secret format *

The pattern for the secret, specified as a regular expression. Learn more about defining custom patterns.

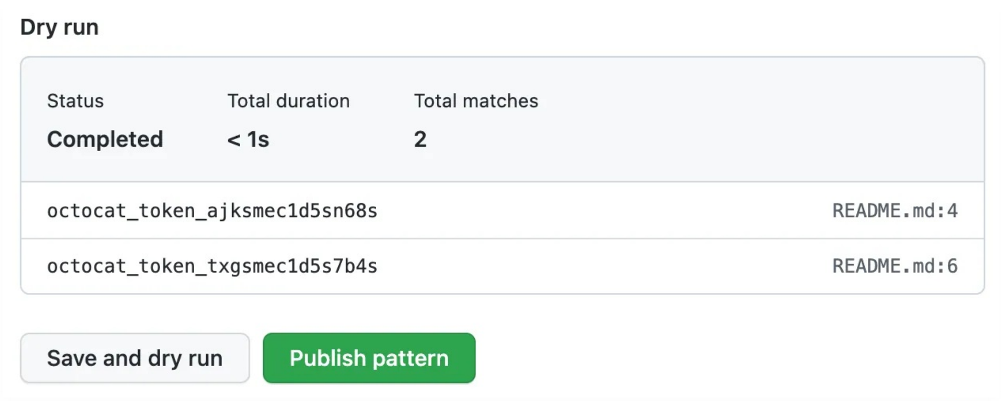
```
octocat_token_[a-zA-Z0-9]{15}
```

Test string * - 1 match

```
octocat_token_1234567890abcde
#These are not tokens
octocat_token_123456790
1234567890abcde
```

7. When you're ready to test your new custom pattern, to identify matches in the repository without creating alerts, click **Save and dry run**.

8. When the dry run finishes, you'll see a sample of results (up to 1000). Review the results and identify any false positive results.



Dry run

Status	Total duration	Total matches
Completed	< 1s	2

octocat_token_ajksmec1d5sn68s	README.md:4
octocat_token_txgsmec1d5s7b4s	README.md:6

9. Edit the new custom pattern to fix any problems with the results, then, to test your changes, click **Save and dry run**.

10. When you're satisfied with your new custom pattern, click **Publish pattern**.

11. Optionally, to enable push protection for your custom pattern, click **Enable**.

> **Note**: The "Enable" button isn't available until after the dry run succeeds and you publish the pattern.

For more information about push protection, see "Push protection for repositories and organizations."

After your pattern is created, secret scanning scans for any secrets in your entire Git history on all branches present in your GitHub repository. For more information on viewing secret scanning alerts, see "Managing alerts from secret scanning."

Example of a custom pattern specified using additional requirements

A company has an internal token with five characteristics. They use the different fields to specify how to identify tokens as follows:

Characteristic	Field and regular expression
Length between 5 and 10 characters	Secret format: `[$#%@AA-Za-z0-9]{5,10}`
Does not end in a `.`	After secret: `[^\.]`
Contains numbers and uppercase letters	Additional requirements: secret must match `[A-Z]` and `[0-9]`
Does not include more than one lowercase letter in a row	Additional requirements: secret must not match `[a-z]{2,}`
Contains one of `$%@!`	Additional requirements: secret must match `[$%@!]`

These tokens would match the custom pattern described above:

```
a9@AAfT!         # Secret string match: a9@AAfT
ee95GG@ZA942@aa  # Secret string match: @ZA942@a
a9@AA!ee9        # Secret string match: a9@AA
```

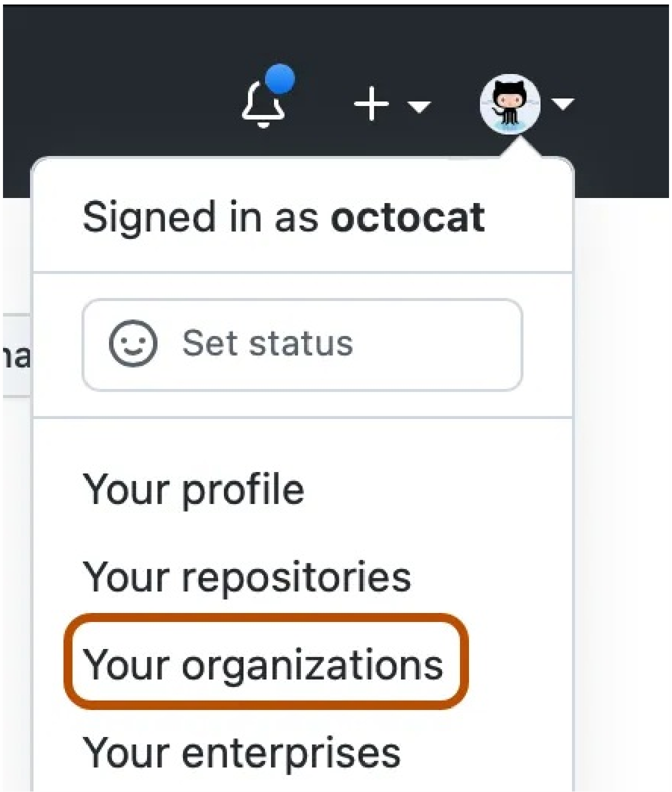
These strings would not match the custom pattern described above:

```
a9@AA.!
a@AAAAA
aa9@AA!ee9
aAAAe9
```

Defining a custom pattern for an organization

Before defining a custom pattern, you must ensure that you enable secret scanning for the repositories that you want to scan in your organization. To enable secret scanning on all repositories in your organization, see "Managing security and analysis settings for your organization."

1. In the top right corner of GitHub Enterprise Server, click your profile photo, then click **Your organizations**.



Signed in as **octocat**

Set status

Your profile

Your repositories

Your organizations

Your enterprises

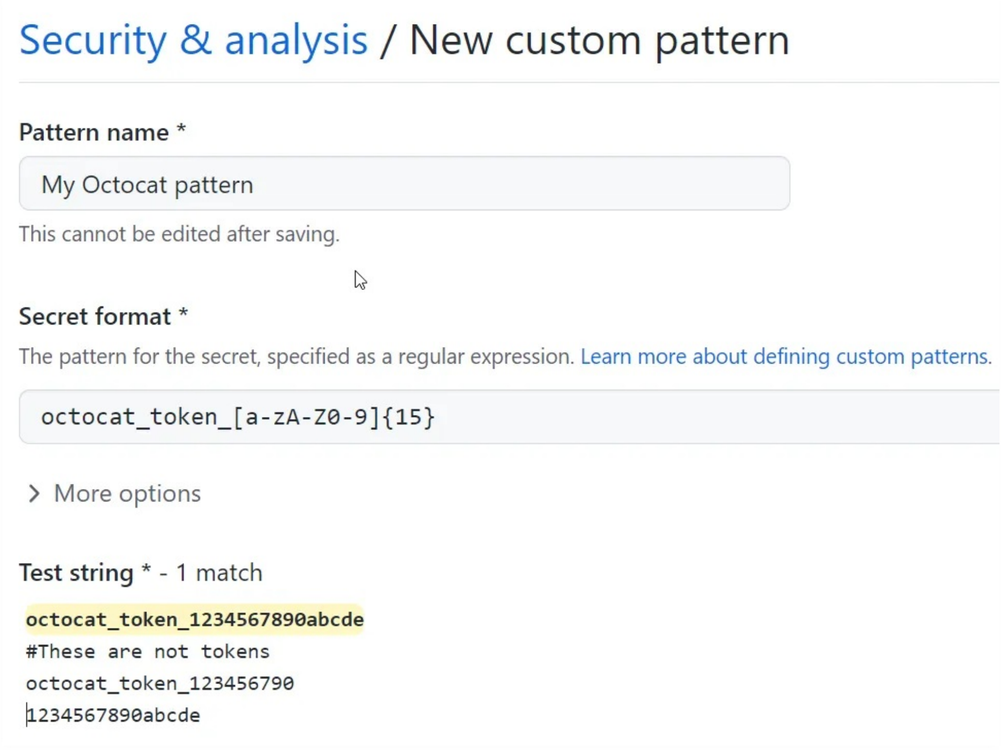
2. Next to the organization, click **Settings**.
3. In the "Security" section of the sidebar, click **Code security and analysis**.
4. Under "Code security and analysis", find "GitHub Advanced Security."
5. Under "Secret scanning", under "Custom patterns", click **New pattern**.
6. Enter the details for your new custom pattern. You must at least provide the name for your pattern, and a regular expression for the format of your secret pattern.

   a. In the "Pattern name" field, type a name for your pattern.

   b. In the "Secret format" field, type a regular expression for the format of your secret pattern.

   c. You can click **More options** ⌄ to provide other surrounding content or additional match requirements for the secret format.

   d. Provide a sample test string to make sure your configuration is matching the patterns you expect.



Security & analysis / New custom pattern

Pattern name *

My Octocat pattern

This cannot be edited after saving.

Secret format *

The pattern for the secret, specified as a regular expression. Learn more about defining custom patterns.

```
octocat_token_[a-zA-Z0-9]{15}
```

Test string * - 1 match

```
octocat_token_1234567890abcde
#These are not tokens
octocat_token_123456790
1234567890abcde
```

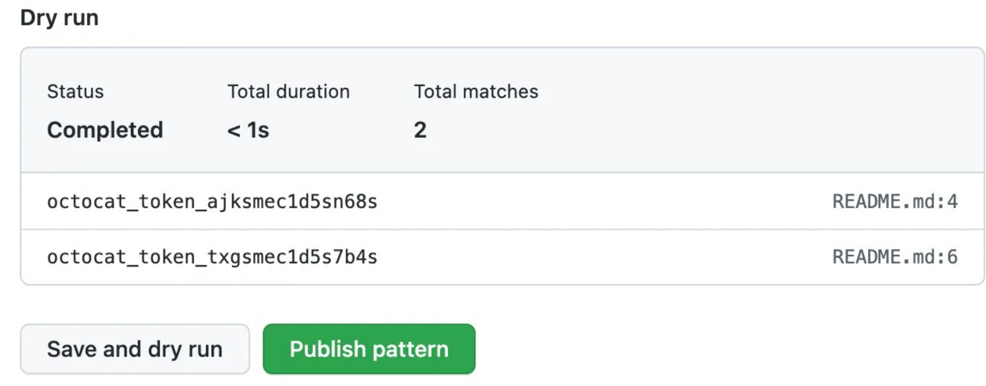
7. When you're ready to test your new custom pattern, to identify matches in select repositories without creating alerts, click **Save and dry run**.

8. Select the repositories where you want to perform the dry run.

   - To perform the dry run across the entire organization, select **All repositories in the organization**.
   - To specify the repositories where you want to perform the dry run, select **Selected repositories**, then search for and select up to 10 repositories.

9. When you're ready to test your new custom pattern, click **Run**.

10. When the dry run finishes, you'll see a sample of results (up to 1000). Review the results and identify any false positive results.



Dry run

Status	Total duration	Total matches
Completed	< 1s	2

```
octocat_token_ajksmec1d5sn68s      README.md:4
octocat_token_txgsmec1d5s7b4s      README.md:6
```

11. Edit the new custom pattern to fix any problems with the results, then, to test your changes, click **Save and dry run**.

12. When you're satisfied with your new custom pattern, click **Publish pattern**.

13. Optionally, to enable push protection for your custom pattern, click **Enable**. For more information, see "[Push protection for repositories and organizations](#)."

Notes:

  - Push protection for custom patterns will only apply to repositories in your organization that have secret scanning as push protection enabled. For more information, see "Push protection for repositories and organizations."
  - Enabling push protection for commonly found custom patterns can be disruptive to contributors.

After your pattern is created, secret scanning scans for any secrets in repositories in your organization, including their entire Git history on all branches. Organization owners and repository administrators will be alerted to any secrets found and can review the alert in the repository where the secret is found. For more information on viewing secret scanning alerts, see "Managing alerts from secret scanning."

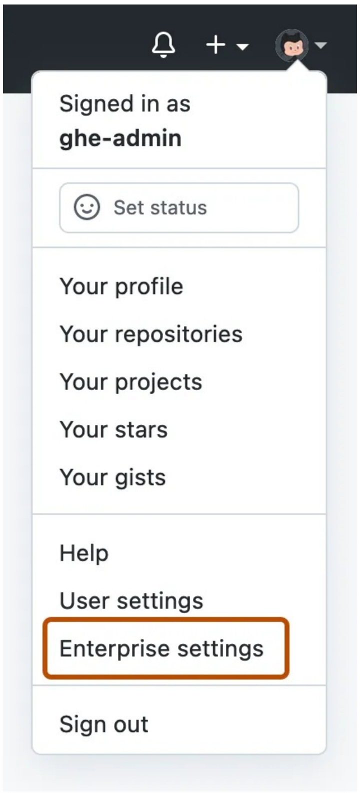
Defining a custom pattern for an enterprise account

Before defining a custom pattern, you must ensure that you enable secret scanning for your enterprise account. For more information, see "Enabling GitHub Advanced Security for your enterprise."

Notes:

- At the enterprise level, only the creator of a custom pattern can edit the pattern, and use it in a dry run.
- You can only perform a dry run on repositories that you have administration access to. If an enterprise owner wants access to perform dry runs on any repository in an organization, they must be assigned the organization owner role. For more information, see "Managing your role in an organization owned by your enterprise."

1. In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



2. In the enterprise account sidebar, click ⚖ **Policies**.

3. Under ⚖ "Policies", click **Code security and analysis**.

4. Under "Code security and analysis", click **Security features**.

5. Under "Secret scanning custom patterns", click **New pattern**.

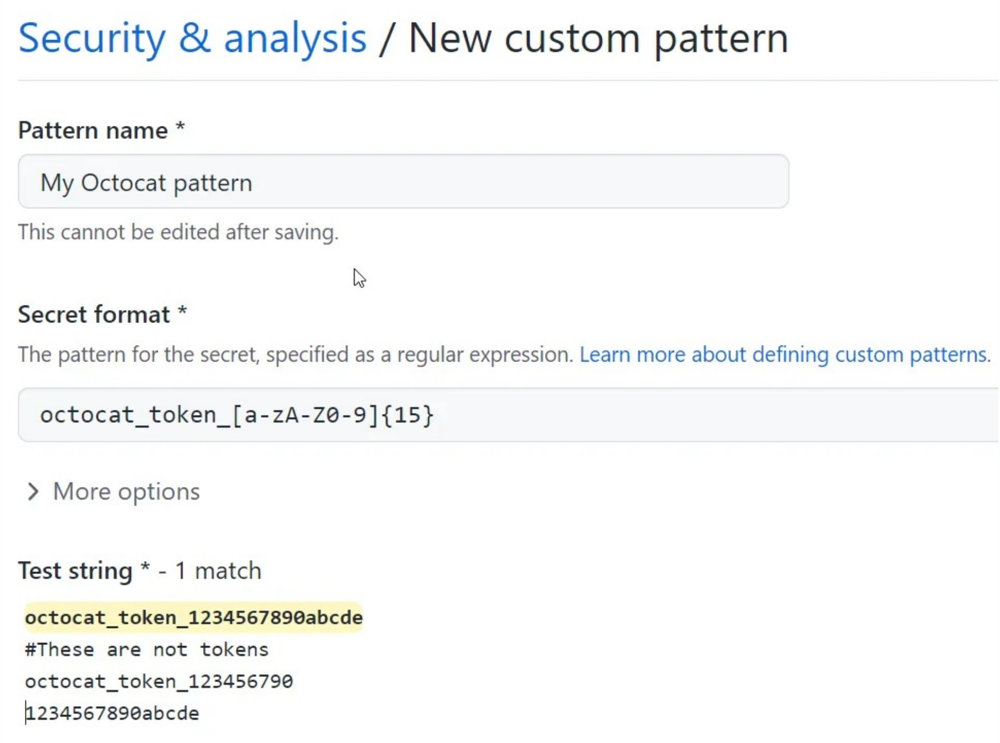
6. Enter the details for your new custom pattern. You must at least provide the name for your pattern, and a regular expression for the format of your secret pattern.

   a. In the "Pattern name" field, type a name for your pattern.

   b. In the "Secret format" field, type a regular expression for the format of your secret pattern.

   c. You can click **More options** ⌄ to provide other surrounding content or additional match requirements for the secret format.

   d. Provide a sample test string to make sure your configuration is matching the patterns you expect.



Security & analysis / New custom pattern

Pattern name *

My Octocat pattern

This cannot be edited after saving.

Secret format *

The pattern for the secret, specified as a regular expression. Learn more about defining custom patterns.

```
octocat_token_[a-zA-Z0-9]{15}
```

Test string * - 1 match

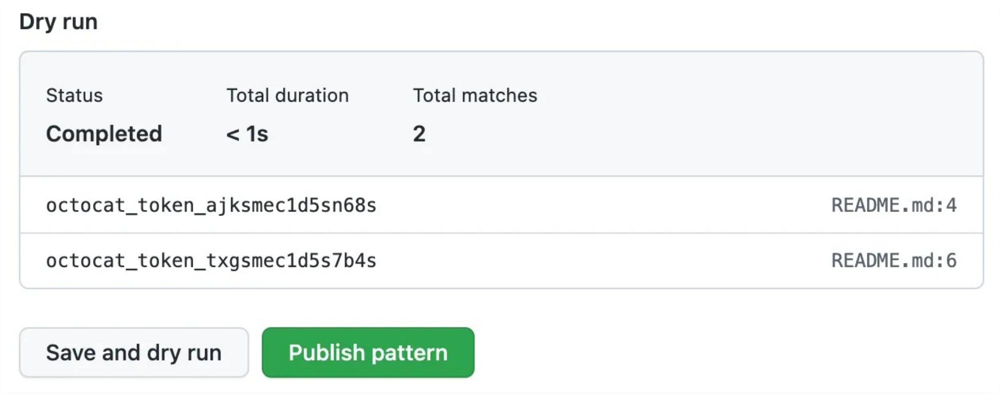
```
octocat_token_1234567890abcde
#These are not tokens
octocat_token_123456790
1234567890abcde
```

7. When you're ready to test your new custom pattern, to identify matches in the enterprise without creating alerts, click **Save and dry run**.

8. Search for and select up to 10 repositories where you want to perform the dry run.

9. When you're ready to test your new custom pattern, click **Run**.

10. When the dry run finishes, you'll see a sample of results (up to 1000). Review the results and identify any false positive results.



Dry run

Status	Total duration	Total matches
Completed	< 1s	2

```
octocat_token_ajksmec1d5sn68s        README.md:4
octocat_token_txgsmec1d5s7b4s        README.md:6
```

11. Edit the new custom pattern to fix any problems with the results, then, to test your changes, click **Save and dry run**.

12. When you're satisfied with your new custom pattern, click **Publish pattern**.

13. Optionally, to enable push protection for your custom pattern, click **Enable**. For more information, see "Push protection for repositories and organizations."

    Notes:

    - To enable push protection for custom patterns, secret scanning as push protection needs to be enabled at the enterprise level. For more information, see "Push protection for repositories and organizations."
    - Enabling push protection for commonly found custom patterns can be disruptive to contributors.

After your pattern is created, secret scanning scans for any secrets in repositories within your enterprise's organizations with GitHub Advanced Security enabled, including their entire Git history on all branches. Organization owners and repository administrators will be alerted to any secrets found, and can review the alert in the repository where the secret is found. For more information on viewing secret scanning alerts, see "Managing alerts from secret scanning."

Editing a custom pattern

When you save a change to a custom pattern, this closes all the secret scanning alerts that were created using the previous version of the pattern.

1. Navigate to where the custom pattern was created. A custom pattern can be created in a repository, organization, or enterprise account.

    - For a repository or organization, display the "Security & analysis" settings for the repository or organization where the custom pattern was created. For more information, see "Defining a custom pattern for a repository" or "Defining a custom pattern for an organization".
    - For an enterprise, under "Policies" display the "Advanced Security" area, and then click **Security features**. For more information, see "Defining a custom pattern for an enterprise account" above.

2. Under "Secret scanning", to the right of the custom pattern you want to edit, click ✎ .

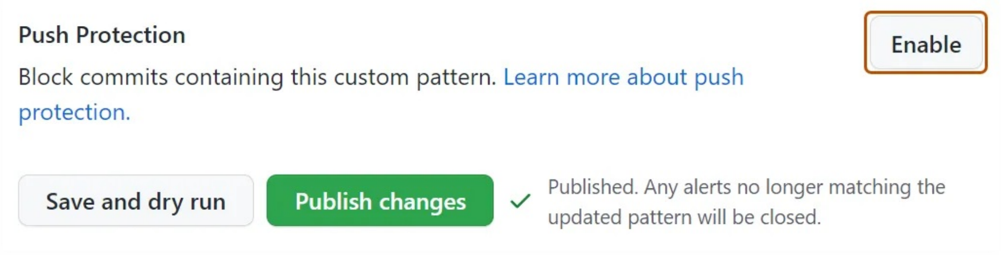
3. When you're ready to test your edited custom pattern, to identify matches without creating alerts, click **Save and dry run**.

4. When you have reviewed and tested your changes, click **Publish changes**.

5. Optionally, to enable push protection for your custom pattern, click **Enable**.

   Note:

   - Push protection for custom patterns will only apply to repositories that have secret scanning as push protection enabled. For more information about enabling push protection, see "Push protection for repositories and organizations."
   - Enabling push protection for commonly found custom patterns can be disruptive to contributors.

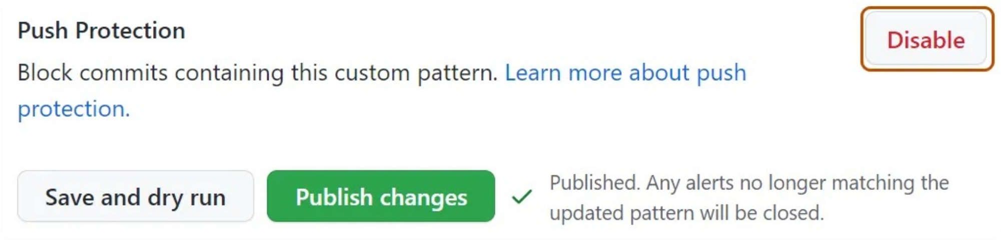


Push Protection

Block commits containing this custom pattern. Learn more about push protection.

✓ Published. Any alerts no longer matching the updated pattern will be closed.

6. Optionally, to disable push protection for your custom pattern, click **Disable**.



Push Protection

Block commits containing this custom pattern. Learn more about push protection.

✓ Published. Any alerts no longer matching the updated pattern will be closed.

Removing a custom pattern

1. Navigate to where the custom pattern was created. A custom pattern can be created in a repository, organization, or enterprise account.

   - For a repository or organization, display the "Security & analysis" settings for the repository or organization where the custom pattern was created. For more information, see "Defining a custom pattern for a repository" or "Defining a custom pattern for an organization".
   - For an enterprise, under "Policies" display the "Advanced Security" area, and then click **Security features**. For more information, see "Defining a custom pattern for an enterprise account" above.

2. To the right of the custom pattern you want to remove, click 🗑.

3. Review the confirmation, and select a method for dealing with any open alerts relating to the custom pattern.

4. Click **Yes, delete this pattern**.

Metrics for custom patterns

Organization owners and people with admin permissions can see an overview of the activity for custom patterns. The overview includes alert and push protection activity for the custom pattern during the last 30 days.

Note: Metrics for custom patterns are in public beta and subject to change.

Viewing metrics for custom patterns

1. Navigate to where the custom pattern was created. A custom pattern can be created in a repository, organization, or enterprise account.

    - For a repository or organization, display the "Security & analysis" settings for the repository or organization where the custom pattern was created. For more information, see "Defining a custom pattern for a repository" or "Defining a custom pattern for an organization".
    - For an enterprise, under "Policies" display the "Advanced Security" area, and then click **Security features**. For more information, see "Defining a custom pattern for an enterprise account" above.

2. Under "Secret scanning", click the custom pattern you want to view.

The metrics are displayed under the custom pattern's name.