

Push protection for repositories and organizations

In this article

- About push protection for repositories and organizations
- Enabling secret scanning as a push protection
- Enabling push protection for a custom pattern
- Using secret scanning as a push protection from the command line
- Using secret scanning as a push protection from the web UI

You can use secret scanning to prevent supported secrets from being pushed into your enterprise, organization, or repository by enabling push protection.

Secret scanning alerts for partners runs automatically on public repositories and public npm packages to notify service providers about leaked secrets on GitHub.com. Secret scanning alerts for users are available for free on all public repositories. Organizations using GitHub Enterprise Cloud with a license for GitHub Advanced Security can also enable secret scanning alerts for users on their private and internal repositories. For more information, see "[About secret scanning](#)" and "[About GitHub Advanced Security](#)." For information about how you can try GitHub Advanced Security for free, see "[Setting up a trial of GitHub Advanced Security](#)."

About push protection for repositories and organizations

Up to now, secret scanning checks for secrets *after* a push and alerts users to exposed secrets. When you enable push protection for your organization or repository, secret scanning also checks pushes for high-confidence secrets (those identified with a low false positive rate). Secret scanning lists any secrets it detects so the author can review the secrets and remove them or, if needed, allow those secrets to be pushed. Secret scanning can also check pushes for custom patterns. For more information, see "[Defining custom patterns for secret scanning](#)."

If a contributor bypasses a push protection block for a secret, GitHub:

- creates an alert in the **Security** tab of the repository in the state described in the table below.
- adds the bypass event to the audit log.
- sends an email alert to organization or personal account owners, security managers, and repository administrators who are watching the repository, with a link to the secret and the reason why it was allowed.

Note: The github.dev web-based editor doesn't support push protection. For more information about the editor, see "[The github.dev web-based editor](#)."

You can monitor security alerts to discover when users are bypassing push protections

and creating alerts. For more information, see "[Auditing security alerts](#)."

If you are an organization owner or security manager, you can view metrics on how push protection is performing across your organization. For more information, see "[Viewing metrics for secret scanning push protection in your organization](#)."

This table shows the behavior of alerts for each way a user can bypass a push protection block.

Bypass reason	Alert behavior
It's used in tests	GitHub creates a closed alert, resolved as "used in tests"
It's a false positive	GitHub creates a closed alert, resolved as "false positive"
I'll fix it later	GitHub creates an open alert

For information on the secrets and service providers supported for push protection, see "[Secret scanning patterns](#)."

Additionally, you can enable push protection for yourself, so that no matter which public repository you push to, you will be protected. For more information, see "[Push protection for users](#)."

Enabling secret scanning as a push protection


For you to use secret scanning as a push protection in public repositories, the enterprise, organization, or repository needs to have secret scanning enabled. To use secret scanning as a push protection in private or internal repositories, the enterprise or organization also needs to have GitHub Advanced Security enabled. For more information, see "[Managing GitHub Advanced Security features for your enterprise](#)," "[Managing security and analysis settings for your organization](#)," "[Managing security and analysis settings for your repository](#)," and "[About GitHub Advanced Security](#)."

Organization owners, security managers, and repository administrators can also enable push protection for secret scanning via the API. For more information, see "[Repositories](#)" and expand the "Properties of the `security_and_analysis` object" section in the REST API documentation.

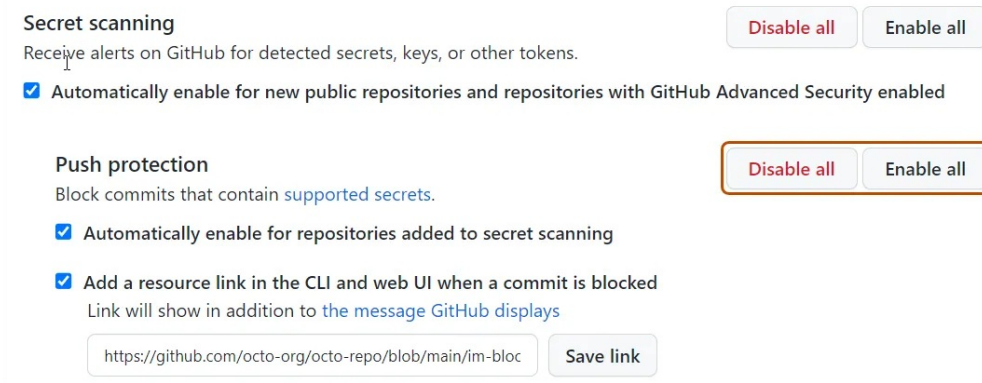
Enterprise administrators can also enable or disable secret scanning as a push protection for the enterprise via the API. For more information, see "[Code security and analysis](#)" in the REST API documentation.

Note: When you fork a repository with secret scanning as a push protection enabled, this is not enabled by default on the fork. You can enable it on the fork the same way you enable it on a standalone repository.

Enabling secret scanning as a push protection for your enterprise

- 1 In the top-right corner of GitHub.com, click your profile photo, then click **Your enterprises**.
- 2 In the list of enterprises, click the enterprise you want to view.
- 3 In the enterprise account sidebar, click  **Settings**.

- 4 In the left sidebar, click **Code security and analysis**.
- 5 Under "Secret scanning", under "Push protection", click **Enable all**.



Secret scanning
Receive alerts on GitHub for detected secrets, keys, or other tokens.

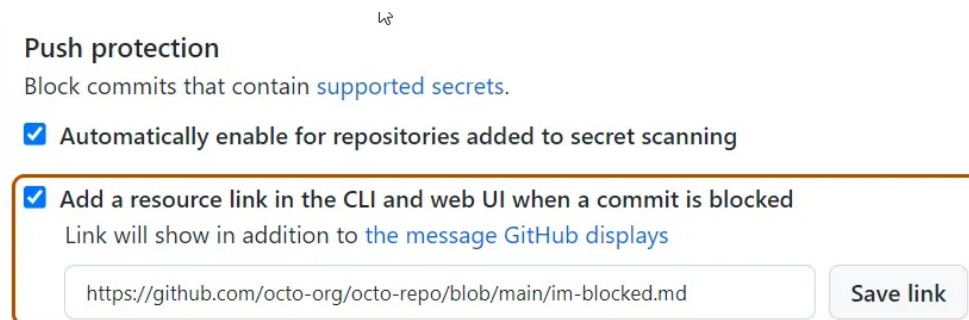
☒ Automatically enable for new public repositories and repositories with GitHub Advanced Security enabled

Push protection
Block commits that contain [supported secrets](#).

☒ Automatically enable for repositories added to secret scanning

☒ Add a resource link in the CLI and web UI when a commit is blocked
Link will show in addition to [the message GitHub displays](#)

- 6 Optionally, click "Automatically enable for repositories added to secret scanning."
- 7 Optionally, to include a custom link in the message that members will see when they attempt to push a secret, click **Add a resource link in the CLI and web UI when a commit is blocked**, then type a URL, and click **Save link**.



Push protection
Block commits that contain [supported secrets](#).


☒ Automatically enable for repositories added to secret scanning

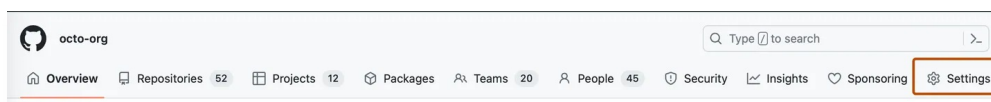
☒ Add a resource link in the CLI and web UI when a commit is blocked
Link will show in addition to [the message GitHub displays](#)


Enabling secret scanning as a push protection for an organization [🔗](#)

You can use security overview to find a set of repositories and enable or disable secret scanning as a push protection for them all at the same time. For more information, see "[Enabling security features for multiple repositories](#)."

You can also use the organization settings page for "Code security and analysis" to enable or disable secret scanning as a push protection for all existing repositories in an organization.

- 1 On GitHub.com, 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, click  **Code security and analysis**.
- 4 Under "Code security and analysis", find "GitHub Advanced Security."

- 5 Under "Secret scanning", under "Push protection", click **Enable all**.
- 6 Optionally, click "Automatically enable for repositories added to secret scanning."
- 7 Optionally, to include a custom link in the message that members will see when they attempt to push a secret, select **Add a resource link in the CLI and web UI when a commit is blocked**, then type a URL, and click **Save link**.

Push protection
Block commits that contain [supported secrets](#).

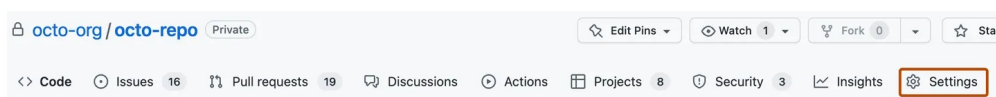
☒ Automatically enable for repositories added to secret scanning

☒ Add a resource link in the CLI and web UI when a commit is blocked
Link will show in addition to [the message GitHub displays](#)

For more information about enabling security features across an organization, see "[Securing your organization](#)."

Enabling secret scanning as a push protection for a repository [🔗](#)

- 1 On GitHub.com, 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 "Push protection", click **Enable**.

Secret scanning
Receive alerts on GitHub for detected secrets, keys, or other tokens.

Push protection
Block commits that contain [supported secrets](#).

Enabling push protection for a custom pattern [🔗](#)




You can enable secret scanning as a push protection for custom patterns stored at the enterprise, organization, or repository level.

Enabling push protection for a custom pattern stored in an enterprise [🔗](#)

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.

Before enabling push protection for a custom pattern at enterprise level, you must also test your custom patterns using dry runs. 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.com, click your profile photo, then click **Your enterprises**.
- 2 In the list of enterprises, click the enterprise you want to view.
- 3 In the enterprise account sidebar, click  **Policies**.
- 4 Under  "Policies", click **Code security and analysis**.
- 5 Under "Code security and analysis", click **Security features**.
- 6 Under "Secret scanning", under "Custom patterns", click  for the pattern of interest.

Note: At the enterprise level, you can only edit and enable push protection for custom patterns that you created.

- 7 To enable push protection for your custom pattern, scroll down to "Push Protection", and click **Enable**.

Push Protection

Block commits containing this custom pattern. [Learn more about push protection](#).

Enable

Save and dry run


Publish changes

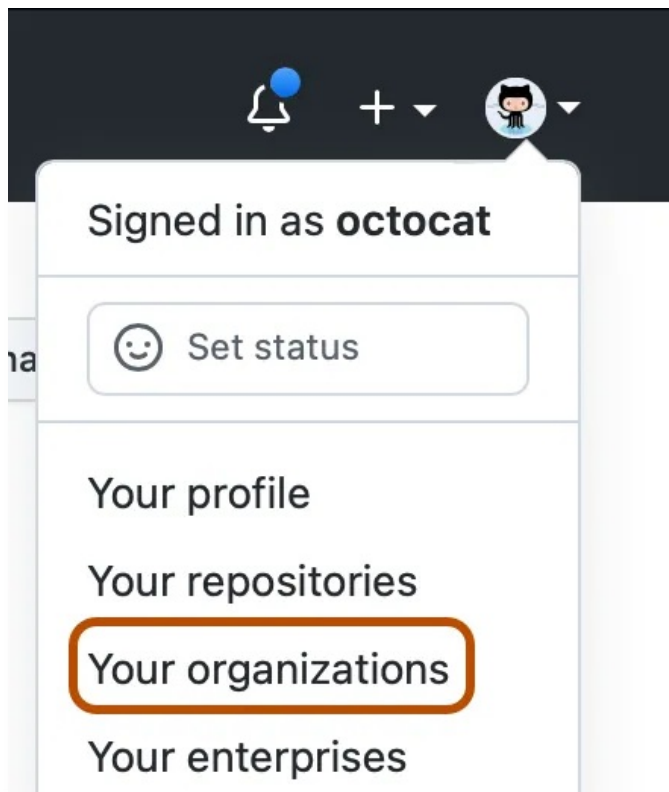




Published. Any alerts no longer matching the updated pattern will be closed.

Enabling secret scanning as a push protection in an organization for a custom pattern

Before enabling push protection for a custom pattern at organization level, 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.com, click your profile photo, then click  **Your organizations**.



- 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  for the pattern of interest.
- 6 To enable push protection for your custom pattern, scroll down to "Push Protection", and click **Enable**.

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.

Push Protection

Block commits containing this custom pattern. [Learn more about push protection](#).

Enable

Save and dry run

Publish changes



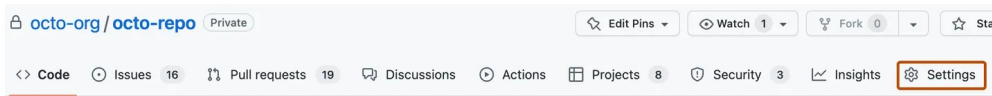
Published. Any alerts no longer matching the updated pattern will be closed.

Enabling secret scanning as a push protection in a repository for a custom pattern

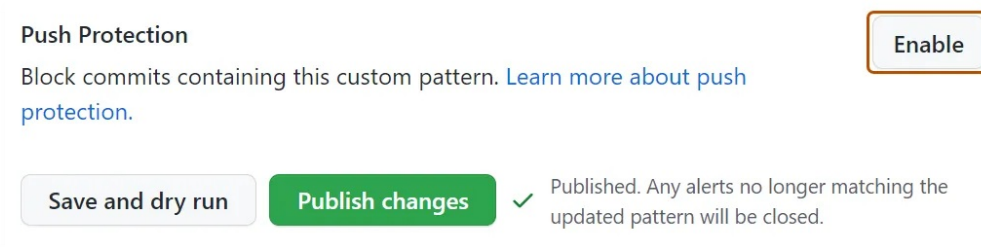
Before enabling push protection for a custom pattern at repository level, you must define the custom pattern for the repository, and test it in the repository. For more information,

see ["Defining custom patterns for secret scanning."](#)

- 1 On GitHub.com, 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 ✎️ for the pattern of interest.
- 6 To enable push protection for your custom pattern, scroll down to "Push Protection", and click **Enable**.



Using secret scanning as a push protection from the command line 🔗

When you attempt to push a supported secret to a repository or organization with secret scanning as a push protection enabled, GitHub will block the push. You can remove the secret from your branch or follow a provided URL to allow the push.

Up to five detected secrets will be displayed at a time on the command line. If a particular secret has already been detected in the repository and an alert already exists, GitHub will not block that secret.

Organization owners can provide a custom link that will be displayed when a push is blocked. This custom link can contain organization-specific resources and advice, such as directions on using a recommended secrets vault or who to contact for questions relating to the blocked secret.

If you confirm a secret is real, you need to remove the secret from your branch, *from all the commits it appears in*, before pushing again. For more information about remediating blocked secrets, see ["Pushing a branch blocked by push protection."](#)

If you confirm a secret is real and that you intend to fix it later, you should aim to remediate the secret as soon as possible. For example, you might revoke the secret and remove the secret from the repository's commit history. Real secrets that have been exposed must be revoked to avoid unauthorized access. You might consider first rotating the secret before revoking it. For more information, see ["Removing sensitive data from a repository."](#)

Notes:

- If your git configuration supports pushes to multiple branches, and not only to the current branch, your push may be blocked due to additional and unintended refs being pushed. For more information, see the [push.default options](#) in the Git documentation.
- If secret scanning upon a push times out, GitHub will still scan your commits for secrets after the push.

Allowing a blocked secret to be pushed [↗](#)

If GitHub blocks a secret that you believe is safe to push, you can allow the secret and specify the reason why it should be allowed.

When you allow a secret to be pushed, an alert is created in the **Security** tab. GitHub closes the alert and doesn't send a notification if you specify that the secret is a false positive or used only in tests. If you specify that the secret is real and that you will fix it later, GitHub keeps the security alert open and sends notifications to the author of the commit, as well as to repository administrators. For more information, see "[Managing alerts from secret scanning](#)."

When a contributor bypasses a push protection block for a secret, GitHub also sends an email alert to the organization owners, security managers, and repository administrators who have opted in for email notifications.

- 1 Visit the URL returned by GitHub when your push was blocked.
- 2 Choose the option that best describes why you should be able to push the secret.
 - If the secret is only used in tests and poses no threat, click **It's used in tests**.
 - If the detected string is not a secret, click **It's a false positive**.
 - If the secret is real but you intend to fix it later, click **I'll fix it later**.
- 3 Click **Allow me to push this secret**.
- 4 Reattempt the push on the command line within three hours. If you have not pushed within three hours, you will need to repeat this process.

Using secret scanning as a push protection from the web UI [↗](#)

When you use the web UI to attempt to commit a supported secret to a repository or organization with secret scanning as a push protection enabled, GitHub will block the commit.

You will see a dialog box with information about the secret's location, as well as options allowing you to push the secret. The secret will also be underlined in the file so you can easily find it.

GitHub will only display one detected secret at a time in the web UI. If a particular secret has already been detected in the repository and an alert already exists, GitHub will not block that secret.

Organization owners can provide a custom link that will be displayed when a push is blocked. This custom link can contain resources and advice specific to your organization. For example, the custom link can point to a README file with information about the organization's secret vault, which teams and individuals to escalate questions to, or the organization's approved policy for working with secrets and rewriting commit history.

You can remove the secret from the file using the web UI. Once you remove the secret, you will be able to commit your changes.

Bypassing push protection for a secret

If you confirm a secret is real, you need to remove the secret from your branch, *from all the commits it appears in*, before pushing again. For more information about remediating blocked secrets, see "[Pushing a branch blocked by push protection](#)."

If you confirm a secret is real and that you intend to fix it later, you should aim to remediate the secret as soon as possible. For more information, see "[Removing sensitive data from a repository](#)."

If GitHub blocks a secret that you believe is safe to push, you can allow the secret and specify the reason why it should be allowed.

When you allow a secret to be pushed, an alert is created in the **Security** tab. GitHub closes the alert and doesn't send a notification if you specify that the secret is a false positive or used only in tests. If you specify that the secret is real and that you will fix it later, GitHub keeps the security alert open and sends notifications to the author of the commit, as well as to repository administrators. For more information, see "[Managing alerts from secret scanning](#)."

When a contributor bypasses a push protection block for a secret, GitHub also sends an email alert to the organization owners, security managers, and repository administrators who have opted in for email notifications.

- 1 In dialog box that appeared when GitHub blocked your commit, review the name and location of the secret.
- 2 Choose the option that best describes why you should be able to push the secret.
 - If the secret is only used in tests and poses no threat, click **It's used in tests**.
 - If the detected string is not a secret, click **It's a false positive**.
 - If the secret is real but you intend to fix it later, click **I'll fix it later**.
- 3 Click **Allow secret**.

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)