bitwarden | Help Center Article

SECRETS MANAGER > INTEGRATIONS

GitLab CI/CD

View in the help center:
https://bitwarden.com/help/gitlab-integration/

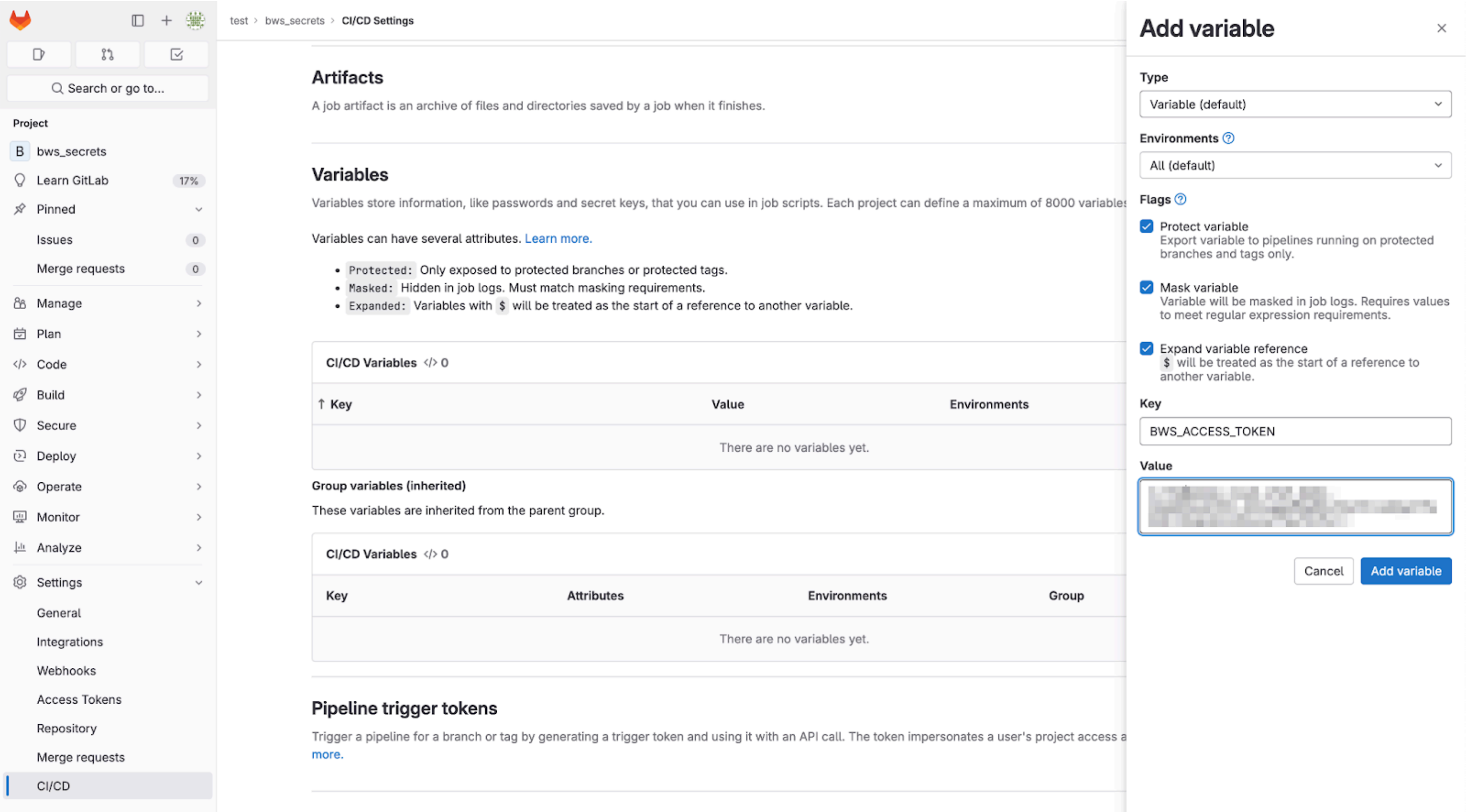
GitLab CI/CD

Bitwarden provides a way to inject secrets into your GitLab CI/CD pipelines using the Bitwarden Secrets Manager CLI. This allows your to securely store and use secrets in your CI/CD workflows. To get started:

Save an access token

In this step, we're going to save an access token as a GitLab CI/CD variable. This token will be used to authenticate with the Bitwarden Secrets Manager API and retrieve secrets.

1. In GitLab, navigate to your project's **Settings** > **CI/CD** page.
2. Select **Expand** in the **Variables** section.
3. Select **Add variable**.
4. Check the **Mask variable** flag.
5. Name the key `BWS_ACCESS_TOKEN`. This is the variable that the Secrets Manager CLI looks for to authenticate. Alternatively, if you need to name the key something else, specify `--access-token NAME_OF_VAR` on the `bws secret get` line later.
6. In another tab, open the Secrets Manager web app and create an access token.
7. Back in GitLab, paste the newly-created access token into the **Value** field.
8. Select **Add variable** to save.



Artifacts

A job artifact is an archive of files and directories saved by a job when it finishes.

Variables

Variables store information, like passwords and secret keys, that you can use in job scripts. Each project can define a maximum of 8000 variables

Variables can have several attributes. Learn more.

- `Protected:` Only exposed to protected branches or protected tags.
- `Masked:` Hidden in job logs. Must match masking requirements.
- `Expanded:` Variables with `$` will be treated as the start of a reference to another variable.

CI/CD Variables </> 0

↑ Key	Value	Environments
There are no variables yet.		

Group variables (inherited)

These variables are inherited from the parent group.

CI/CD Variables </> 0

Key	Attributes	Environments	Group
There are no variables yet.			

Pipeline trigger tokens

Trigger a pipeline for a branch or tag by generating a trigger token and using it with an API call. The token impersonates a user's project access a more.

Add variable ×

Type: Variable (default)

Environments: All (default)

Flags

☑ Protect variable
Export variable to pipelines running on protected branches and tags only.

☑ Mask variable
Variable will be masked in job logs. Requires values to meet regular expression requirements.

☑ Expand variable reference
`$` will be treated as the start of a reference to another variable.

Key: BWS_ACCESS_TOKEN

Value: [redacted]

Cancel | Add variable

Add a variable in GitLab

Add to your workflow file

Next, we're going to write a rudimentary GitLab CI/CD workflow. Create a file called `.gitlab-ci.yml` in the root of your repository with the following contents:

```bash
stages:
- default_runner

image: ubuntu
build:
  stage: default_runner
  script:
  - |
    # install bws
    apt-get update && apt-get install -y curl git jq unzip
    export BWS_VER="1.0.0"
    curl -LO \
      "https://github.com/bitwarden/sdk/releases/download/bws-v$BWS_VER/bws-x86_64-unknown-linux-gnu-$BWS_VER.zip"
    unzip -o bws-x86_64-unknown-linux-gnu-$BWS_VER.zip -d /usr/local/bin

  # use the `bws run` command to inject secrets into your commands
  - bws run -- 'npm run start'
```

Where:

- `BWS_VER` is the version of the Bitwarden Secrets Manager CLI to install. You can pin the version being installed by changing this to a specific version, for example `BWS_VER="0.3.1"`.

> ⚠ **Warning**
>
> Secrets are stored as environment variables. It is important to avoid running commands that would output these secrets to the logs.

Run the CI/CD pipeline

On the left, select **Build** > **Pipelines** and select **Run pipeline** at the top-right of the pace. Select **Run pipeline** on the page to run the newly-created pipeline.