Epic Games - Cloud Developer Platform

Using Substrate Vault Secrets with Github Actions

Downloaded from Epic Games Confluence
Date: 2025-07-12 04:07:26
Original URL: https://confluence-epicgames.atlassian.net/wiki/spaces/CDE/pages/81068349

Document Level Classification

200

- Interacting with Kubernetes from GitHub Actions

Introduction

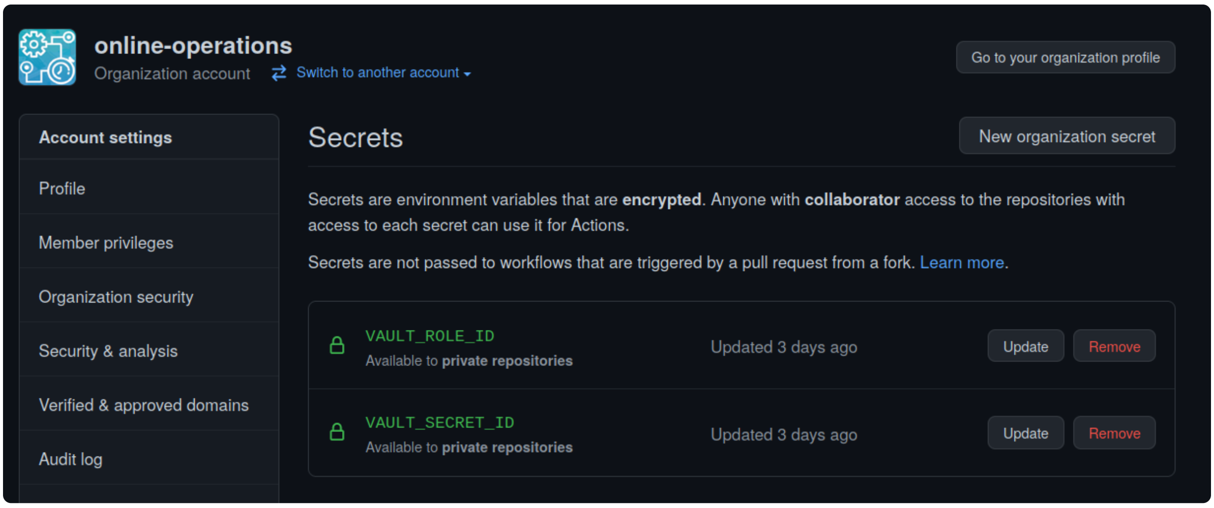
Copied from GitHub Actions User Guide

Operations will, upon request in #cloud-ops-support-ext, generate a GitHub org-specific Vault AppRole ID and secret for a GitHub org owner.

Once generated, the org owner should add the secrets at the org level under the following keys:

- VAULT_ROLE_ID
- VAULT_SECRET_ID

Secrets added at the GitHub org level are available to all repos under the org.



Secrets

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to the repositories with access to each secret can use it for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. Learn more.

VAULT_ROLE_ID — Available to **private repositories** — Updated 3 days ago

VAULT_SECRET_ID — Available to **private repositories** — Updated 3 days ago

Once the AppRole secrets are in place, a workflow may utilize the Vault action like so:

```
jobs:
  ci:
    steps:
    - name: Retrieve Vault secrets
      uses: actions/hashicorp_vault-action@v2.4.0
      with:
        url: https://vault.substrate.on.epicgames.com
        method: approle
        roleId: ${{ secrets.VAULT_ROLE_ID }}
        secretId: ${{ secrets.VAULT_SECRET_ID }}
        secrets: |
          secret/data/brand/project/region/environment/category/name ap
          secret/data/brand/project/region/environment/category/name ur
    - name: Use Vault secrets
      run: echo "${API_TOKEN}" >keys
```

In this example, "api_token" and "url" are retrieved and set as the environment variables API_TOKEN and API_URL, respectively (if a destination is not explicitly listed, a normalized version is created). The action registers these environment variables as masking, so their values should not be printed in run output. For more information about using the action, please see the action's documentation.

Terraform mappings from GHE org to Vault AppRoles are available here.

Interacting with Kubernetes from GitHub Actions

To interact with Kubernetes from a GitHub Action you can use this GitHub Action step, courtesy Andy Sammalmaa of the Online Web team.

```
yaml
steps:
- name: configure
  uses: online-web/configure-kubernetes-action@v1
  with:
    vaultRoleId: ${{ secrets.VAULT_ROLE_ID}}
    vaultSecretId: ${{ secrets.VAULT_SECRET_ID }}
    clusterName: bebe-dev-eos-dev-portal
    namespace: team-dev-portal
```

Your GitHub org's Vault token must have `read` and `list` permissions to the following Vault path:

```
secret/+/substrate/k8s/use1a/{dev|live}/build/{ACCOUNT CODE}/+/deployer
                         ^       ^                 ^         ^
                         |       |                 |         |
                         |       |                 |         The + will
                         |       |                 |         You can ch
                         |       |                 |         Github's a
                         |       |                 |
                         |       |                Account code is your
                         |       |
                         |      Use "dev" for dev clusters, "live" for
                         |
                        Today most clusters are in use1a (us-east-1 AWS
                        the future you could use "+" here for multiple
```

Your vault policy can include more than one kuberentes credential path to provide github access to more than one account, environment, or cluster.

Reach out to #cloud-github-ext for help.

Page Information:
Page ID: 81068349
Space: Cloud Developer Platform
Downloaded: 2025-07-12 04:07:26