

Using Substrate Vault with vault-injector to Inject Secrets

Downloaded from Epic Games Confluence

Date: 2025-07-12 04:07:27

Original URL: <https://confluence-epicgames.atlassian.net/wiki/spaces/CDE/pages/81068744>

Document Level Classification

[200](#)

Starting from August 2024, it is mandatory that all new Kubernetes clusters use External Secrets Operator (ESO). We strongly recommend using ESO, but the Vault Injector sidecar will still remain available with limited support, bug fixes will be provided if needed to maintain core functionality.

The following documentation describes using secrets from Substrate Vault with HashiCorp Vault (AKA Substrate Vault), SSSM, and vault injector in epic-app. If you are looking to use secrets from Substrate Vault by using native Kubernetes Secrets in your application, then it is recommended that you use ESO (External Secrets Operator) for injecting secrets.

Reference [Using External Secrets Operator \(ESO\) in epic-app to inject secrets](#) for documentation on using ESO.

- [Introduction](#)
 - [Substrate Vault](#)

- [SSSM](#)
- [How do I get Access To Vault?](#)
- [Granting Secrets Access to Your Pod](#)
- [How do I Authenticate with Substrate Vault?](#)
 - [Vault UI](#)
 - [Vault CLI \(Using HashiCorp CLI\)](#)
 - [Reading Secrets using the Vault CLI](#)
 - [Castle CLI](#)
- [How do I Publish and Share Secrets with Substrate Vault and SSSM?](#)
 - [Secrets Layout](#)
 - [Vault Access Policies](#)
 - [Substrate Vault Secrets Across Environments](#)
 - [Creating a New Namespaces in Substrate Vault](#)
 - [Creating New Secrets in Substrate Vault \(Vault UI\):](#)
 - [Use SSSM to Share Vault Secrets with Your Substrate Application](#)
 - [Use SSSM to Revoke Access from Your Substrate Application](#)
 - [Use SSSM to Share Vault Secrets with Other Teams](#)
 - [Use SSSM to Revoke Access from Teams](#)
 - [Use SSSM to Share Secrets with AWS Services via IAM Roles](#)
 - [Use SSSM to Revoke Access from AWS Services via IAM Roles](#)
 - [Use SSSM to Delete a Secret Path](#)
 - [Actions Requiring Administrative Access in SSSM](#)
- [How do I Access and Use Vault Secrets in My Application?](#)
 - [Configuring your Pod](#)
 - [Using Secrets in your Application](#)
 - [Validating Secrets in your Pod](#)
- [How do I Get Support for Substrate Vault and SSSM?](#)

Introduction

Applications use sensitive configurations, or secrets, when accessing other backend or downstream resources. For example, a database password or an API key. For a Substrate application, secrets are stored and managed in [Substrate Vault](#). Substrate Vault is an instance of [HashiCorp Vault](#) hosted within Epic Games' infrastructure and is supplemented by [Self-](#)

[Service Secrets Management \(SSSM\)](#) for self-service namespace provisioning and permissions management.

Substrate Vault

[Substrate Vault](#) is a deployment of HashiCorp Vault that teams at Epic games use to share secrets with their teams and with their applications running in Substrate. Substrate Vault is integrated with Kubernetes, AWS, and other Substrate components such as Codefresh and SSSM. Substrate Vault provides secrets management for all aspects of Substrate lifecycle, including local development, builds in Codefresh and TeamCity, and deployment via Kubernetes.

SSSM

[Self-service Secrets Management](#) (SSSM) is a web UI and API that allows Substrate customers to configure access to secrets in Substrate Vault. SSSM serves two key functions. First, SSSM allows customers to create project namespaces to store their secrets. After creating a project namespace, teams can store secrets in an isolated location in Substrate Vault. Second, SSSM allows customers to share secrets with other systems, including Kubernetes, AWS Lambda, and AWS EC2. Additionally,

secrets owned by specific teams are automatically made available to their Codefresh pipelines.

SSSM manages a convergence of policy and access permissions across Vault, Codefresh, AWS, Kubernetes, and other systems by way of metadata tracked in Starmap API. SSSM itself does not store any data. Configuration is stored in [Starmap API](#) and secret material is stored in [Substrate Vault](#). Secret material does not pass through the SSSM API or UI.

How do I get Access To Vault?

In order to create or browse secrets in Vault you must be part of a TONUAM group and your team must have a Vault policy. If you are not in a TONUAM group, [head to Sailpoint to get started](#). Once you're there, select the **Team Online UAM (TONUAM) Access** application, and then choose a TONUAM group from the list of **Access Profiles**.

Request

You are requesting access to Team Online UAM (TONUAM) Access

Access Profiles
310 Access Profiles

2

1-25 of 310 Access Profiles

- ☐ **3lateral-core-tech**
Grants access to TONUAM-3lateral-core-tech
Available for me and others
- ☐ **TONUAM-team-ecosec**
Grants access to TONUAM-team-ecosec
Available for me and others
- ☐ **WWT Emporium Contractors**
Grants access to TONUAM-WWT Emporium Contractors
Available for me and others
- ☐ **online-social-clock-raf**
Grants access to TONUAM-online-social-clock-raf
Available for me and others

Requesting For

3 ☒ Myself
☐ Others


Add Comments

4

Set Expiration Date
Optional. Select a date when this access should be automatically removed.

5

Once you are in a group you will be able to login to Substrate Vault and the Self-Service Secrets Management app (see below).

If you do not yet have a TONUAM group, reach out to us in  #cloud-ops-support-ext for help.

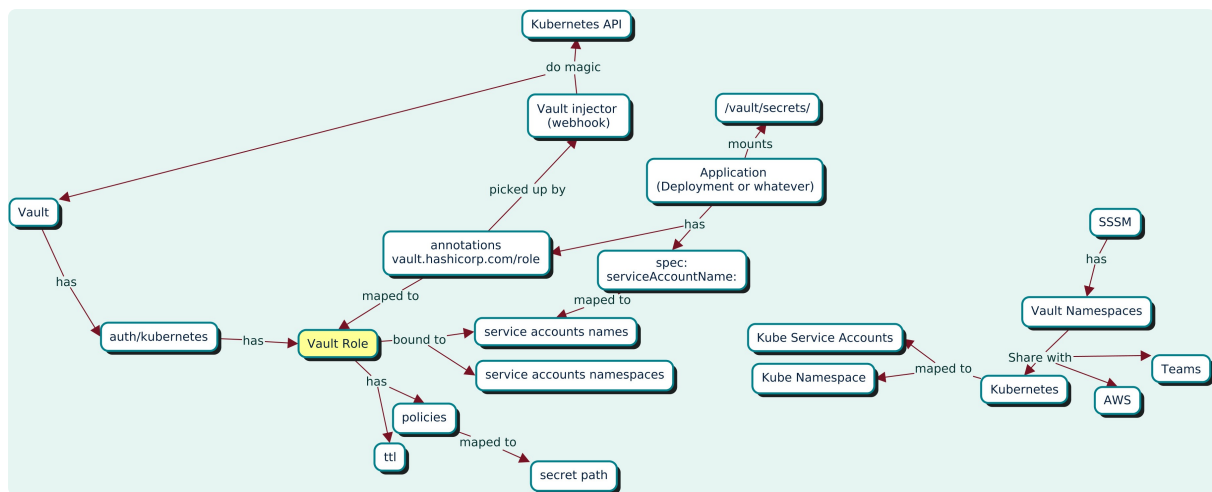
Several example secrets are automatically added to Vault to help you browse to the right "folder" so if you can see these already, you're good to go!

Granting Secrets Access to Your Pod

Before you continue to configuring pod annotations, use Self-service Secrets Management to share access with your Kubernetes service account.

Substrate Vault Guidance

See <https://confluence-epicgames.atlassian.net/wiki/spaces/CE/pages/93487474> for details.



How do I Authenticate with Substrate Vault?

Vault UI

1. Make sure you are connected to the network in-office or using [GlobalProtect VPN](#) if you are working remotely.
2. Go to the Substrate Vault UI using this link <https://vault.substrate.on.epicgames.com/ui/vault/auth?with=oidc>
 1. If you have already logged into your Okta dashboard, the link above will automatically log you in using Okta credentials.
 2. However, if you see the login screen instead, select **OIDC** for the Method and enter the role which you will be assuming. Leave it empty for a default role usage and click **Sign in with OIDC Provider**.

Sign in to Vault

Method

OIDC

Role

Default

Leave blank to sign in with the default role if one is configured

More options

Sign in with OIDC Provider

Okta Login

Some users may not receive a push notification on their phone (due to phone notification system glitches) on login via Okta. It is recommended to open Okta Verify app and see if you actually received a confirmation request.

Vault CLI (Using HashiCorp CLI)

To use the vault CLI you must first download the latest open source Vault release from <https://releases.hashicorp.com/vault/>. This will be the highest number release with no suffix at the end. For example, the latest version is currently **vault_1.14.1**. Do not download versions with a suffix of ent, ent.hsm, ent.hsm.fips1402, or any variation with a rc* suffix.

../

vault_1.14.1+ent.hsm.fips1402 ❌

vault_1.14.1+ent.hsm ❌

vault_1.14.1 ✅

vault_1.14.1+ent.fips1402 ❌

vault_1.14.1+ent ❌

Configure and Run Vault

```
export VAULT_ADDR=https://vault.substrate.on.epicgames.com
vault login -method=oidc
vault read /auth/token/lookup-self
```

Reading Secrets using the Vault CLI

You can browse Vault's KV space using the `vault kv list` and `vault kv get` commands. Note that you will always need to supply at least `"secret/"` as the path (see below) in order to get any results.

```
$ vault kv list secret/
Keys
----
btools/
substrate/
terraform/
uam/
uas/
...

$ vault kv list secret/substrate/k8s/global/live/runtime
Keys
```



```
----  
example-secret
```

Secrets in Vault are stored as a map / hash of key-value pairs. When you read a secret you will see all of the keys.

```
$ vault kv get secret/substrate/vault/use1a/live/runtime/vault-healthch  
===== Metadata =====  
Key                Value  
---              -  
created_time       2021-04-29T17:48:05.443500438Z  
deletion_time      n/a  
destroyed          false  
version            3  
  
===== Data =====  
Key                Value  
---              -  
opsgenie-api-key   432ca231-ce98-43c1-98be-cecf7892cf99  
token              s.40dfs0IUERwer9dsfZ432dsV  
token_accessor     Mt4723REc7SD9s94cnFE371
```

You can specify an individual key using `-field`. This is useful to use a secret value in an environment variable or in an input to another command line tool.

```
$ vault kv get -field token secret/substrate/vault/use1a/live/runtime/v  
s.40dfs0IUERwer9dsfZ432dsV
```

Gotcha

When referencing secrets on the command-line, you will use `secret/path/to/secret`. However, when referencing those secrets in the Vault API, you must use `secret/data/path/to/secret`. Take a look at the table below under "Secrets Paths Across Systems" for more details.

Read more about the [vault login command](#) and the [vault kv command](#).

Castle CLI

For instructions using the Castle CLI, refer to the sub-page called [Managing Substrate Vault Secrets with Castle \(CLI\)](#).

How do I Publish and Share Secrets with Substrate Vault and SSSM?

When a Substrate environment is provisioned, your team is also provided permissions to a *Project Namespace* in [SSSM](#), which relates to a path-prefix in Vault: `secret/<brand>/<project>/<region>/<environment>/<category>/<name>` . This allows you to sign-in to [Vault \(using Okta credentials\)](#) and create or update secrets.

Secrets in Vault have a *Path* (for example, `secret/my-brand/my-project/use1a/live/runtime/database`), a *Key* (for example, `db_password`) and its associated (*secret*) *Value*. You can create any meaningful paths within your team's SSSM-namespace to hold secrets.

Secrets Layout

Secrets layout in Substrate Vault follows a defined pattern. Since many systems access Vault, we use these access patterns to automate policy generation and simplify access across systems. On first glance this appears complicated. It is complicated! Don't hesitate to ask for help if you feel confused or stuck. It is important to note: Vault does not enforce any particular path when you write a secret, but automated tooling will not work correctly unless you follow the pattern. Here's the composition of


a secret path: `secret/<brand>/<project>/<region>/<environment>/<category>/<name>`

- **brand** (or product) refers to a major initiative like eos, fortnite, or unrealengine. Any brand may contain numerous projects. Important: this should not be a team name. Teams change over time, and we don't want to rewrite policies and migrate secrets when this happens.
- **project** refers to a complete system like leaderboards. It may be composed of many individual services, span cluster or continents, and have pre-production and production environments.
- **region** refers to a specific infrastructure region. use1a, currently the most common region identifier, refers to "us-east-1 amazon". But this could also be euw1a, usw2a, or global, for example.
- **environment** typically refers to dev or live. All Substrate environments come with dev and live accounts, and secrets are segmented in the same way. Secrets that are region-agnostic may live under global. Other environments like test exist but they are not common.
- **category** is used by automated tools that write policies. The two most important ones you will see today are build and runtime. Build secrets are globally available to TeamCity. Runtime secrets are selectively available to Kubernetes clusters.
- **name** refers to the specific name of your secret.

Vault Access Policies

Substrate Vault is designed to have restrictive access controls by default. Vault applies access controls based on the secret's path. This means two things:

1. A policy must exist before you can read or write to Vault. These policies are managed by [UAM](#).
2. Secrets must be placed in the proper path in Vault in order to work correctly. The specification for paths is detailed in <https://confluence-epicgames.atlassian.net/wiki/spaces/CE/pages/93488039>.

If your service or use-case is not already covered by existing policy, you can ask for a policy in  #cloud-ops-support-ext.

Here is an example of a policy that gives users in the `team-ogs` group in Okta access to certain secrets in Vault:

```
# Vault Policy for TONUAM-team-ogs
path "secret/*" {
    capabilities = ["list"]
}
path "secret+/eos/leaderboards/*" {
    capabilities = ["read", "list", "update", "create", "delete"]
}
path "secret+/eos/statsachievements/*" {
    capabilities = ["read", "list", "update", "create", "delete"]
}
path "secret+/eos/sdkconfig/*" {
    capabilities = ["read", "list", "update", "create", "delete"]
}
```

Read more about Vault policies in HashiCorp's [Vault policy documentation](#).

Substrate Vault Secrets Across Environments

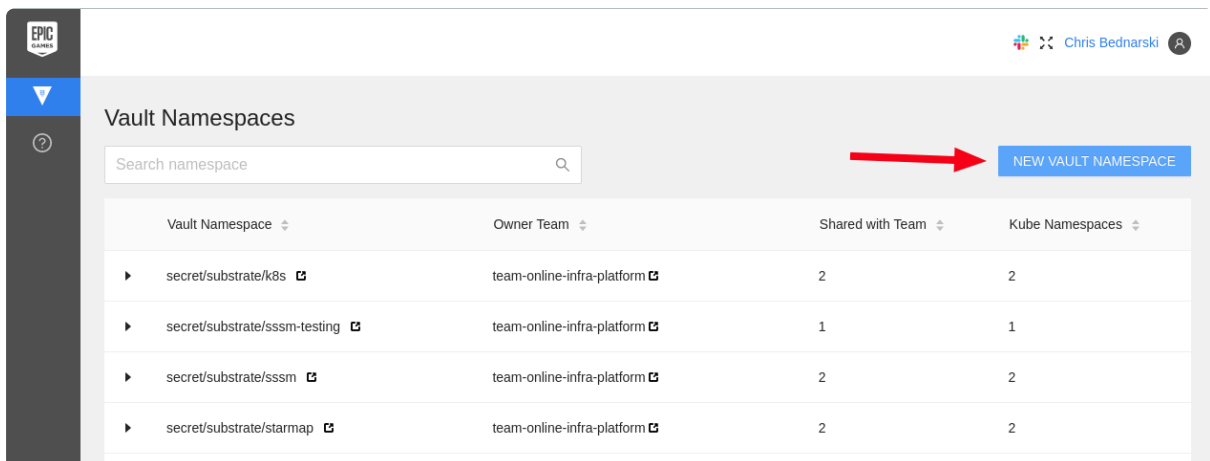
In order to access secrets across environments you must specify the correct path for your secret as this is the differentiator between them. For example, if you want to create and access a secret for the live environment, the `environment` part of the path must be live. If you want to create and access a secret for the dev environment, the environment part of the path must be dev.

Examples:

Live Environment Secret	<code>secret/my-brand/my-project/use1a/live/runtime/database</code>
Dev Environment Secret	<code>secret/my-brand/my-project/use1a/dev/runtime/database</code>

Creating a New Namespaces in Substrate Vault

As explained above under [Secrets Layout](#), Substrate Vault organizes secrets into projects. You can create a new project namespace in the [Secrets Management UI](#) under any brand you already have access to by clicking **New Vault Namespace**.



In the pop-up window, select the team, brand, and project name. If you are on more than one team, members of the "owners" team will be allowed to share this secret with other teams.

Create Vault Namespace

X

Owners : Search team

Secret Path : secret / brand


▼

/ Input your project name

Cancel

Request

After a few moments you will be able to create secrets here in the Vault UI.

If you do not yet have access to any brands, please reach out to  #cloud-ops-support-ext for help.

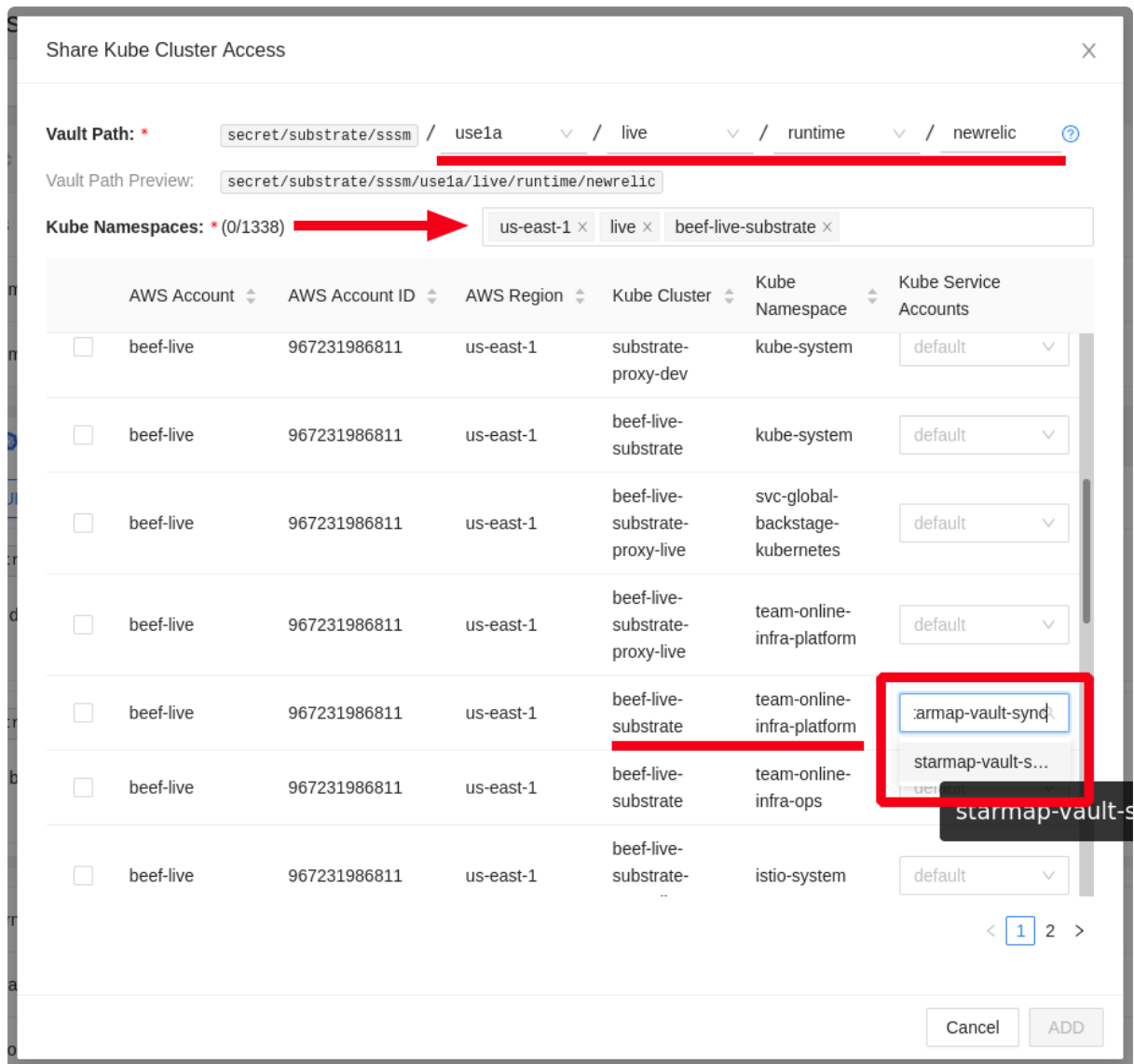
Creating New Secrets in Substrate Vault (Vault UI):

1. Sign-in to [Vault \(using Okta credentials\)](#)
2. Navigate to the secrets location for your environment. For example `secret/my-brand/my-project/use1a/live/runtime` .
3. Click **Create secret +**.
4. Enter a name for the secret in the **Key** field and the secret data in the value field next to it.
5. Click the **Save** button to save the secrete.

Note: Secrets can also be created using the [Vault CLI](#).

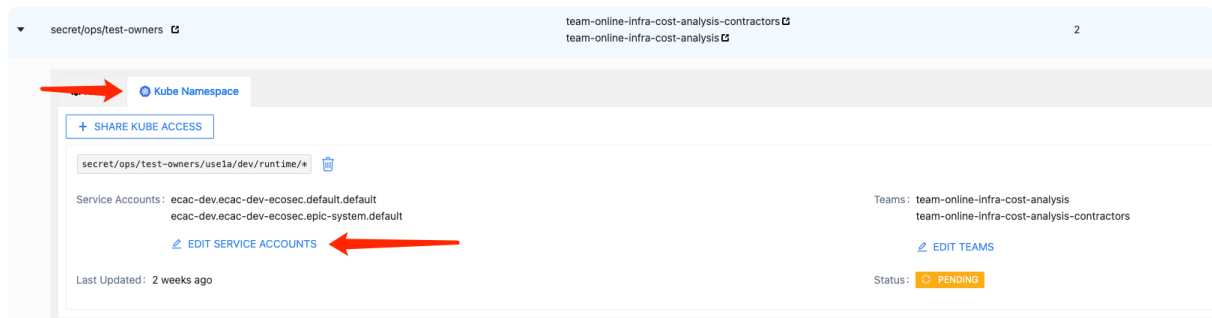
Use SSSM to Share Vault Secrets with Your Substrate Application

1. Login to [SSSM](#).
2. Navigate to the namespace that matches the secret created in the vault.
3. In the **Kubernetes** tab, choose **Share Kube Access**.
4. Fill in the details for the **Vault path**, **Substrate cluster** and **Substrate namespace**.
5. Use the drop-down to choose the [Service Account associated with your application](#).

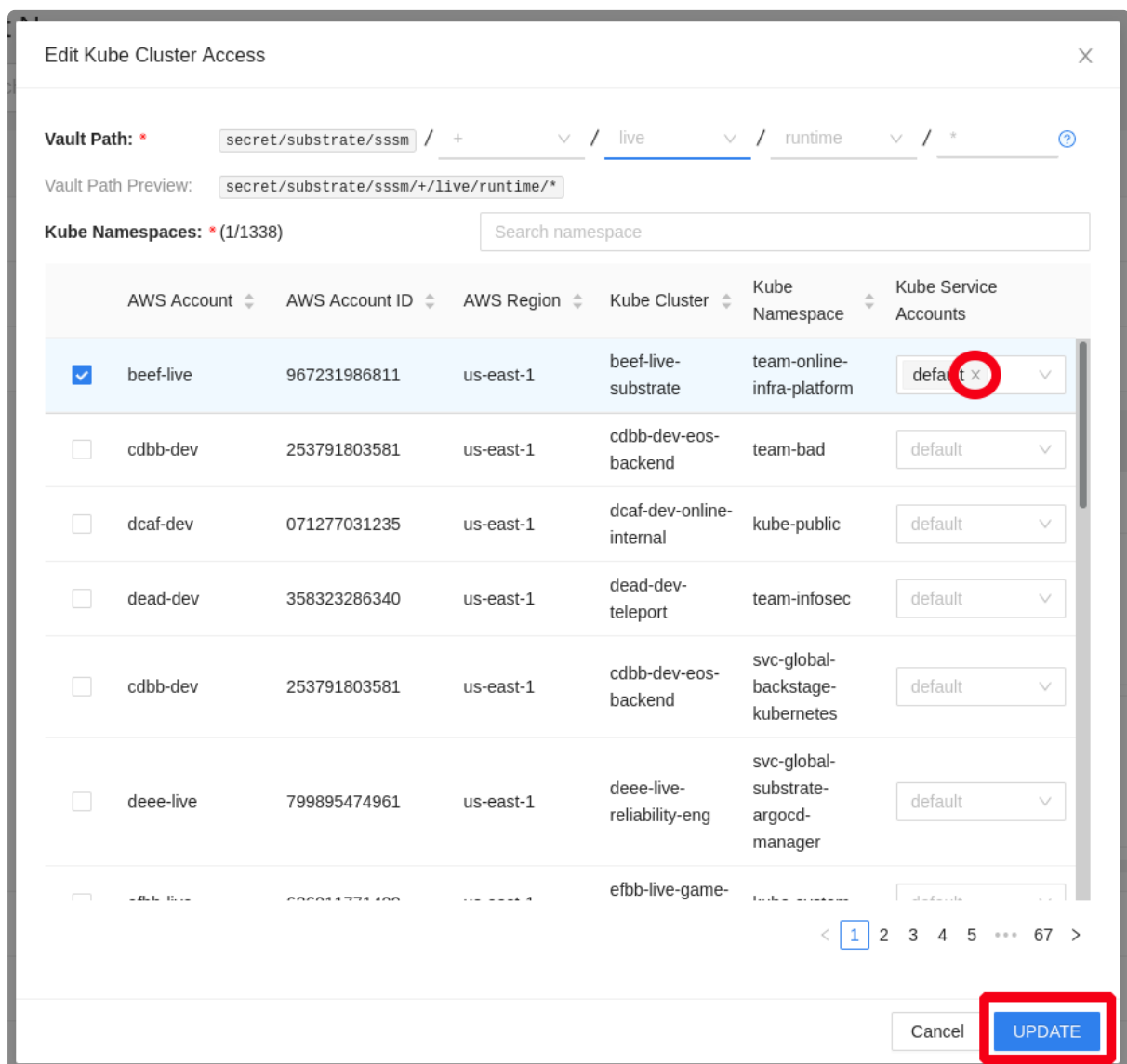


Use SSSM to Revoke Access from Your Substrate Application

To revoke access from Kubernetes, first expand the namespace in the list of namespace. Select the **Kubernetes** tab, and then click the **Pencil icon** below the Kubernetes service account list you want to edit.



In the pop-up window, click the X next to the service account you which to remove, and then click the Update button.



Revoking access from a Kubernetes service account will not have an immediate effect for two reasons. First, removing the policy is not immediate. Second, pods that were already deployed and already had access to Kubernetes will not be affected. The updated policies will only apply to new pods that launch after the new configuration is applied.

Use SSSM to Share Vault Secrets with Other Teams

To share secrets with another team, first expand the namespace by clicking its row (you can click the arrow or the name of the namespace).

The screenshot shows the Vault SSSM interface. At the top, a table lists namespaces. The 'secret/ops/test-owner' namespace is expanded, showing a list of teams. A red arrow points to the 'Teams' section, and another red arrow points to the '+ SHARE WITH TEAM' button. Below this, a pop-up window shows the path 'secret/ops/test-owner/*' and a list of teams: 'team-online-infra-cost-analysis-contractors', 'team-online-infra-ops', and 'team-online-infra-cost-analysis'. The 'team-online-infra-cost-analysis-contractors' team is selected. Below the list, there is an 'EDIT TEAMS' link and a 'Last Updated: 2 days ago' timestamp. The status is 'PENDING'.

Namespace	Team	Count	Count
secret/ops/test-owner	team-online-infra-cost-analysis-contractors	3	0
secret/ops/test-owner	team-online-infra-ops	3	0
secret/ops/test-owner	team-online-infra-cost-analysis	3	0
secret/ops/test-owners	team-online-infra-cost-analysis-contractors	2	2
secret/ops/test-owners	team-online-infra-cost-analysis	2	2
secret/ops/test-owners	team-online-infra-ops	3	4
secret/ops/test-owners	team-online-infra-platform-contractors	3	1
secret/ops/test-owners	team-online-infra-platform	3	1
secret/ops/test-owners	team-online-infra-cost-analysis-contractors	2	0
secret/ops/test-owners	team-online-infra-cost-analysis	2	0
secret/ops/test-owners	team-online-infra-ops	3	4
secret/ops/test-owners	team-online-infra-cost-analysis-contractors	4	4
secret/ops/test-owners	team-online-infra-cost-analysis	2	0

Click the **Share With Team** button. In the pop-up window, fill in the path and select a team from the list on the left. Check the box and click the right-facing arrow to add the team.

Vault uses ***** (asterisk) as the wildcard at the end of a path and uses **+** (plus sign) as the wildcard in the middle of the path.

Edit Team Access

The selected Vault path is already shared with the team below, please be careful to make changes!

Vault Path: / * / / / / name

Vault Path Preview:

Teams: * (2/255)

1/2 items Unshared teams

☐ team-online-infra-ops-tier1

☒ team-online-infra-ops

2 items Shared teams

☐ team-online-infra-platform-contractors

☐ team-online-infra-platform

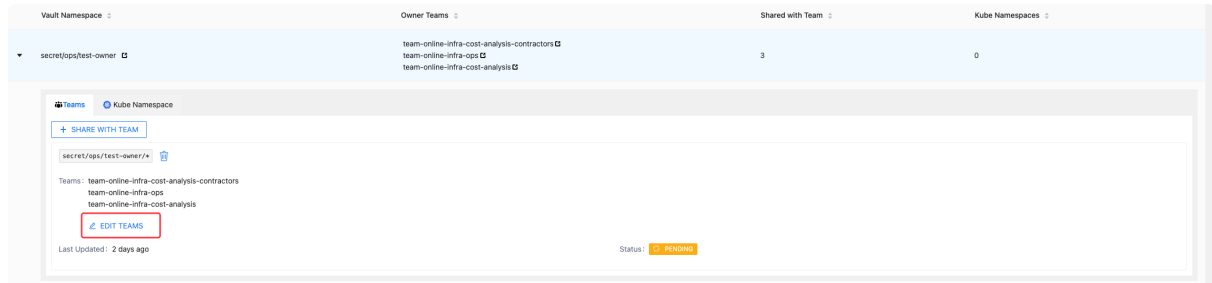
Cancel UPDATE

Click Update when you are done making changes. The status will change from "Live" to "Pending" and will change back to "Live" after the policies have been updated. Usually this takes about a minute.

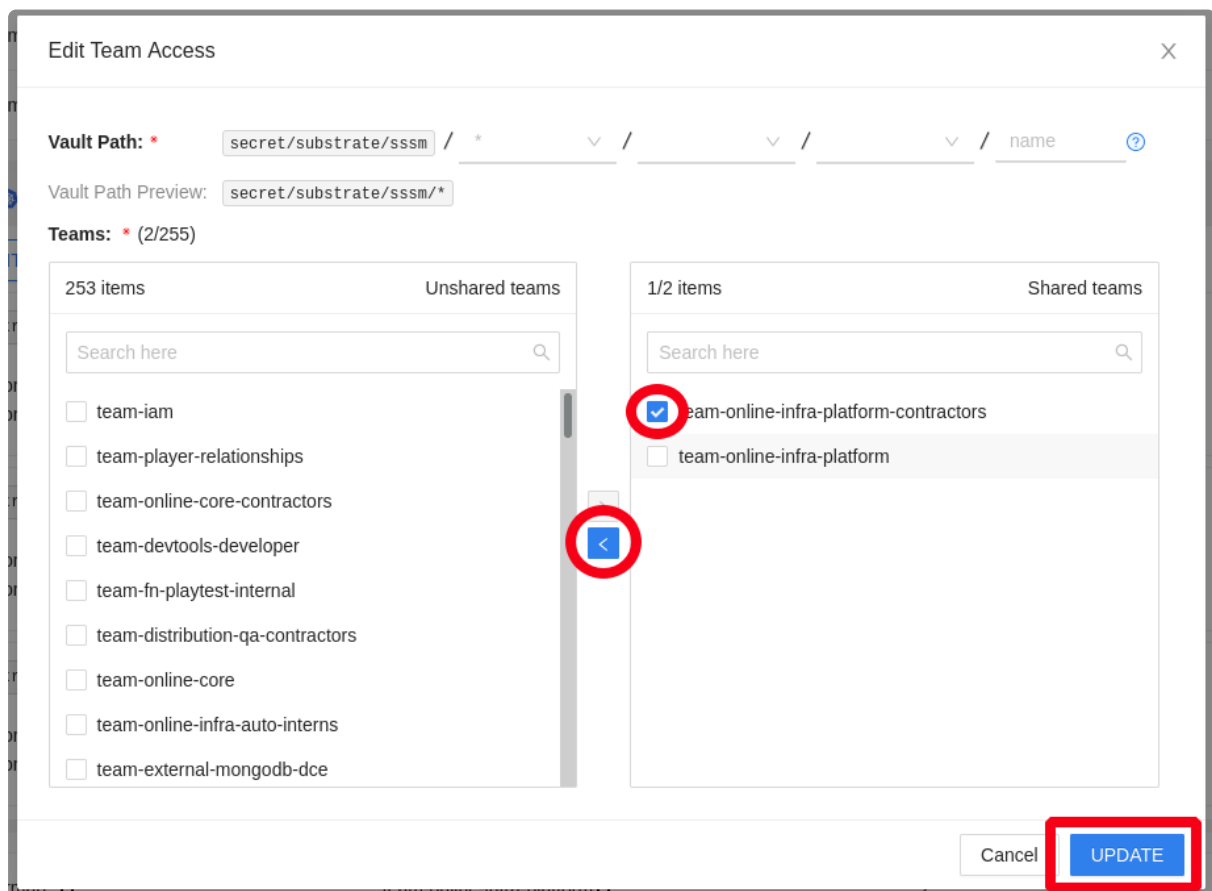
Namespaces shared with other teams will grant read/write/delete access to Vault under the specified namespace, but other teams will not be able to share access using the Secrets Management UI. The team which created the namespace is responsible for sharing access.

Use SSSM to Revoke Access from Teams


To revoke access from another team, first expand the namespace by clicking its row and then click the **Pencil icon** below the team list you want to edit.



In the pop-up window, check the team on the right side and use the left arrow to remove them. Click **Update** when you are done making changes.



Note: You cannot remove access for the team which created the namespace.

Revoking access takes longer to apply compared to other operations. Typically access will be removed within 12 hours. If you have a security issue and need immediate support, please reach out via  #cloud-ops-support-ext or Shit Happens for faster response.

Use SSSM to Share Secrets with AWS Services via IAM Roles

1. Login to [SSSM](#).
2. Navigate to the namespace that matches the secret created in the vault.
3. Expand the namespace by clicking its row and then select the AWS tab.
4. Click the **Share with IAM Role** button.



Vault Namespace ▾

Owner Teams ▾

Teams ▾

Kubernetes ▾

AWS ▾


▼ secret/substrate/sssm-testing	 New	team-online-infra-platform-contractors team-online-infra-platform 	2	1	1
---------------------------------	---	--	---	---	---

Teams

Kubernetes

aws AWS

+ SHARE WITH IAM ROLE

secret/substrate/sssm-testing/use1a/live/runtime/* 


Roles: sssm.dead-dev.aws-elasticbeanstalk-ec2-role

Teams: team-online-infra-platform
team-online-infra-platform-contractors

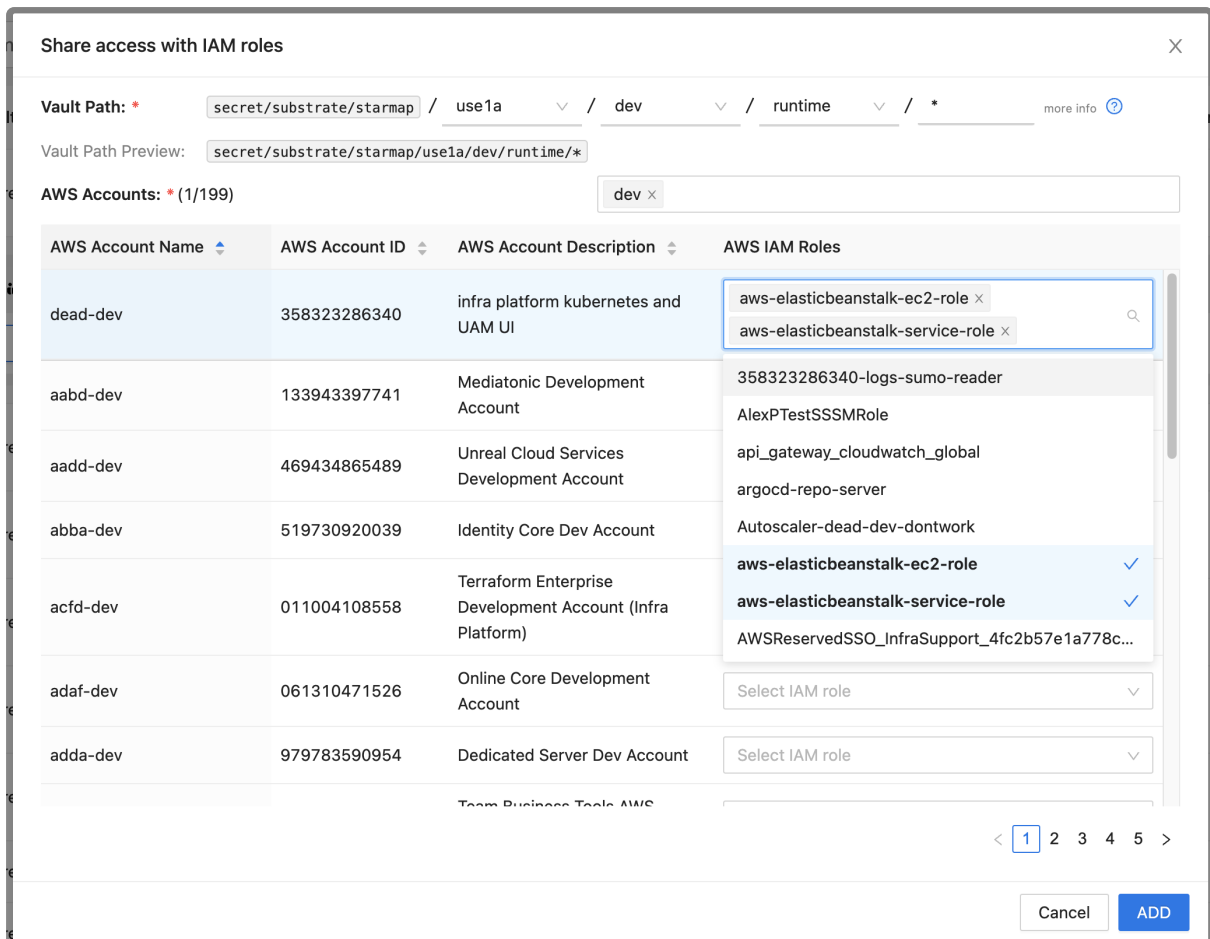
[EDIT IAM ROLES](#)

[EDIT TEAMS](#)

Last Updated: yesterday

Status:  LIVE



Access is shared with AWS IAM Roles by specifying the Vault path alongside with account, and the roles in that account.



This pop-up window is fairly detailed so here are a few tips:

- While entering the **Vault Path**, you can click the **Vault Path Preview** to view that path in Vault and select a specific secret to share with Kubernetes. (This will open Vault UI in a new tab)
- You can use the filter box to the right of **AWS Accounts** to search for a particular AWS account or accounts tier (like dev or live). The filter will automatically update if you select a region and environment in the Vault Path.
- You can share with more than one AWS IAM Role by entering them into the box in the **AWS IAM Roles** column.
- There are many AWS Accounts. The one you want may appear on page 2 (or later).
- Click the **X** next to any item to remove it from the filter, or to remove the IAM Role from your selection.

5. Click **Add** when you have made your selections.

Vault uses  (asterisk) as the wildcard at the end of a path and uses  (plus sign) as the wildcard in the middle of the path.

Request to Vault signed by specified AWS IAM Role credentials will receive read-only access to Vault, and will not be able to list secrets using the Vault API. In spite of this, for least-privileged access, we recommend making your AWS paths as specific as possible.

After share path with AWS IAM Role, SSSM will generate corresponding Vault policy and Vault role. The name of the role should be used to get access to the specified path. It can be copied to a clipboard by hovering over a role name and then pressing appearing **Copy Vault role** button.

This role name then can be used in AWS Lambda, EC2 or Elastic Beanstalk services to authorize request to specified path.

Use SSSM to Revoke Access from AWS Services via IAM Roles

To revoke access from AWS, first expand the namespace in the list of namespace. Select the **AWS** tab, and then click the **Edit IAM Roles** button with Pencil icon below the AWS IAM Roles list you want to edit.

Vault Namespace
Owner Teams
Teams
Kubernetes
AWS

secret/substrate/sssm-testing

New

team-online-infra-platform-contractors
team-online-infra-platform

211

Teams
Kubernetes
AWS

+ SHARE WITH IAM ROLE

secret/substrate/sssm-testing/use1a/live/runtime/*

Roles: sssm.dead-dev.aws-elasticbeanstalk-ec2-role

Teams: team-online-infra-platform
team-online-infra-platform-contractors

EDIT IAM ROLES

EDIT TEAMS

Last Updated: yesterday

Status: LIVE

In the pop-up window click the **X** next to the IAM Role you wish to remove, and then click **Update** button.

Edit IAM Roles

Vault Path: secret/substrate/sssm-testing/use1a/live/runtime/*

AWS Accounts: (1/209)


Search account

AWS Account Name	AWS Account ID	AWS Account Description	AWS IAM Roles
dead-dev	358323286340	infra platform kubernetes and UAM UI	<div>aws-elasticbeanstalk-ec2-role</div> <div>aws-elasticbeanstalk-service-role</div>
dedf-dev	515281371940	Team Online Infra Cost Analysis Development Account	Select IAM role
team-infra-platform-tomato	264799539613	team-infra-platform-tomato account	Select IAM role
team-infra-platform-live	188014929444	team-infra-platform-live account	Select IAM role
team-analytics	542851537354	team-analytics account	Select IAM role
ecac-dev	401767072812	Ecosystem Security Team Dev Account	Select IAM role
beea-live	269018228832	Identity Core Live Account	Select IAM role
bebe-dev	688916340032	Substrate development account for team-dev-portal	Select IAM role

< 1 2 3 4 5 ... 11 >

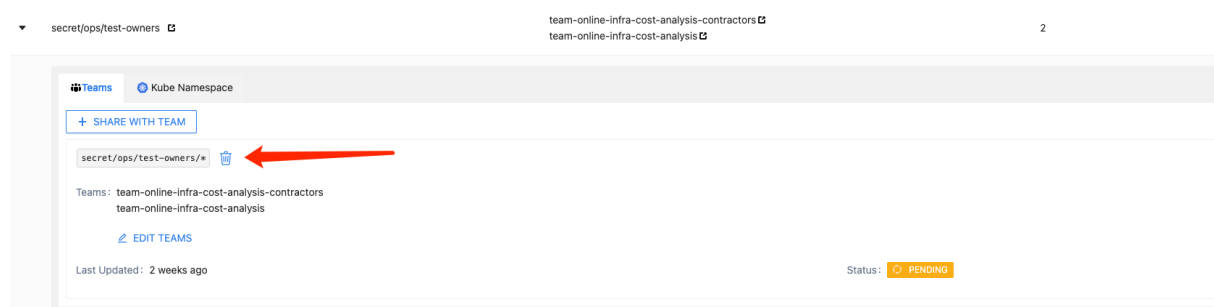
Cancel
UPDATE

Revoking access from IAM role will not have an immediate effect. First removing the Vault policy and role is not immediate. Second AWS services that are running already have access to created token at least for an hour. Newly created services will have an updated policy.

Revoking access takes longer to apply compared to other operations. Typically access will be removed within 12 hours. If you have a security issue and need immediate support, please reach out via  #cloud-ops-support-ext or Shit Happens for faster response.

Use SSSM to Delete a Secret Path

To delete a specific secret path, first, expand the Vault namespace, and click the **delete** icon next to the path you want to delete. Please note that the shared teams and Kubernetes namespaces will lose access to the path after deleting it, and only the Vault namespace owner can delete the secret path.



In the confirmation field, type delete and Click the delete button

Delete Access

X

Delete team access permanently? This action cannot be undone.

The shared teams will lose access to the secret path below

secret/damon-test/new-service/use1a/dev/runtime/*

To confirm deletion, type **delete** in the field.

delete

Cancel

Delete

There is a notification (the message will disappear automatically after 10 seconds) on the top-right to show which secret path has been deleted, and the deleted secret path disappears from the UI.

Vault Namespaces

Search namespace

NEW VAULT NAMESPACE

Vault Namespace	Owner Teams	Shared with Team	Kube Namespaces
secret(ops/test-owner)	team-online-infra-cost-analysis-contractors team-online-infra-ops team-online-infra-cost-analysis	3	0

Teams

Kube Namespace

+ SHARE WITH TEAM

secret(ops/test-owner)/*

Teams: team-online-infra-cost-analysis-contractors
team-online-infra-ops
team-online-infra-cost-analysis

EDIT TEAMS


Last Updated: 2 days ago

Status: PENDING

Notification

secret(ops/test-owner/use1a) has been deleted

Actions Requiring Administrative Access in SSSM

Some actions are not supported in SSSM UI. In these cases, you will need to reach out to  #cloud-ops-support-ext for help. Here are some actions that are not currently supported in the UI, and will require assistance from an admin:

- Deleting a project

- Changing ownership of a project to a different team
- Granting write access for Kubernetes pods (this is uncommon)

How do I Access and Use Vault Secrets in My Application?

For applications to access secrets, you can use the [HashiCorp Vault Agent Injector](#). The Vault Agent Injector is already installed in a Substrate cluster, you need to use annotations to enable it for your application. When the application is deployed, the Vault Agent Injector will fetch the secrets configured via [annotations](#) and make them available as files to your application.

Configuring your Pod

Once the sidecar injector is configured in your cluster, its fairly easy to pull vault secrets into your container.

1. Add annotations to your helm chart to read the secrets. These go in your Kubernetes yaml for your container at ***spec.template.metadata.annotations***
2. Update your application to read the secrets from a file or from [environment variables](#)

The annotation example below fetches a secret from Vault at path ***secret/eos/datarights/use1a/dev/runtime/postgresql***, and then uses a template to write the secret into a file so we can use it in our startup script. Our example secret is named ***database-creds*** in the config. The secret name is sometimes concatenated with an annotation.

```
annotations:  
  vault.hashicorp.com/agent-inject: "true"  
  vault.hashicorp.com/agent-pre-populate-only: "true"  
  vault.hashicorp.com/role: "cdbb-dev.eos-backend.eos-backend"  
  vault.hashicorp.com/agent-inject-secret-database-creds: "secret/eos/d  
  vault.hashicorp.com/agent-inject-template-database-creds: |
```

```
{{ with secret "secret/eos/datarights/use1a/dev/runtime/postgresql"
  export PGSQL_USERNAME='{{ .Data.data.username }}'
  export PGSQL_PASSWORD='{{ .Data.data.password }}'
{{- end }}
```

Let's walk through this line-by-line.

1. `vault.hashicorp.com/agent-inject: "true"`

This annotation activates the vault sidecar for your pod.

2. `vault.hashicorp.com/agent-pre-populate-only: "true"`

By default the Vault sidecar will continuously run beside your pod and update secrets on disk as they are changed in Vault. However, if you are only using Vault to configure environment variables, this means Vault is running all the time and your application never reads the updated secrets from disk. When you add this annotation, the Vault sidecar will start alongside your pod, retrieve secrets from Vault, and then exit.


If you are running a Kubernetes job instead of a service this will make sure the Vault sidecar pod exits properly.

3. `vault.hashicorp.com/role: "cdbb-dev.eos-backend.eos-backend.default"`

The `role` annotation controls how your pod authenticates to Vault. The format is `<account>.<cluster>.<namespace>.<service-account>`, as in `cdbb-dev.eos-backend.eos-backend.default`, above. If the role is incorrect the sidecar will be unable to authenticate and will fail to start.

The role corresponds to the Kubernetes service account associated with your pod. You can copy the role name from Self-Service Secrets

Management:

`secret/substrate/starmap/*/live/runtime/*` 

Service Accounts: `beef-live.beef-live-substrate.team-online-infra-platform.starmap`
`beef-live.beef-live-substrate.team-online-infra-platform.starmap-aws-iam-roles-collector`
`beef-live.beef-live-substrate.team-online-infra-platform.starmap-codefresh-accounts-collector`
`beef-live.beef-live-substrate.team-online-infra-platform.starmap-keda-vault-access-granter`
`beef-live.beef-live-substrate.team-online-infra-platform.starmap-vault-sync`

[EDIT SERVICE ACCOUNTS](#)

Last Updated: 4 days ago

4. `vault.hashicorp.com/agent-inject-secret-database-creds: "secret/eos/datarights/use1a/dev/runtime/postgresql"`

The `agent-inject-secret` annotation specifies the vault path for the secret you want to retrieve from Vault. It looks like

`vault.hashicorp.com/agent-inject-secret-<secret_name>`. In our example, we named our secret ***database-creds***. This will become important in the next step when we render the secret in a template. The sidecar uses a unique policy per service account and can typically only read `/runtime/` secrets for the corresponding environment. For example, the Vault policy for this service account will look something like: `secret/+eos/datarights/use1a/live/runtime/*`.

Substrate Vault Guidance

See the section title "Secrets Paths Across Systems" in <https://confluence-epicgames.atlassian.net/wiki/spaces/CE/pages/93487474> for details.

5. `{{vault.hashicorp.com/agent-inject-template-database-creds: |`
`{{ with secret "secret/eos/datarights/use1a/dev/runtime/postgresql" -}}`
`export PGSQL_USERNAME='{{.Data.data.username}}'`
`export PGSQL_PASSWORD='{{.Data.data.password}}'`

```
{{- end }}
```

The `agent-inject-template` annotation configures the [template Vault Agent uses](#) to render a secret. Each template must have a unique name which must match the secret name used in `agent-inject-secret`. The format is `vault.hashicorp.com/agent-inject-template-<secret_name>`. Remember that secrets in Vault are maps. The template syntax will help you select the specific field(s) you want to use. In our example, `username` and `password` specify fields in the secret `.../postgresql`. You may specify multiple fields in a single template to export multiple environment variables at once.

When retrieving multiple secrets from Vault you will need to repeat the `agent-inject-secret` and `agent-inject-template` annotations (steps 3 and 4 above) for each new secret.

You can read about the [complete list of Vault Sidecar annotations](#) in the [Vault Agent Sidecar Injector documentation](#). For annotations in general, see the documentation for [Kubernetes Annotations](#).

Using Secrets in your Application

Vault Agent [templates are rendered](#) to the pod's filesystem at `/vault/secrets/<secret_name>`. It is up to you to decide how to consume these secrets in your application. In the examples above we rendered a file that exports environment variables, but you will need to source this in your application's startup scripts to work.

From our previous example:

```
{{ with secret "secret/eos/datarights/usel1a/dev/runtime/postgresql"
-}}
    export PGSQL_USERNAME='{{ .Data.data.username }}'
    export PGSQL_PASSWORD='{{ .Data.data.password }}'
{{- end }}
```

becomes:

```
export PGSQL_USERNAME='database_user'
```

```
export PGSQL_PASSWORD='database_password'
```

The examples above generate bash code. If your password contains special characters like quotes `'`, `"`, `$`, or other characters with special meaning in bash you will need to make sure it is properly escaped using single quotes or a HEREDOC. For example:

```
export PGSQL_PASSWORD=`cat << 'ESCAPE'
{{ .Data.data.password }}
ESCAPE`
```

You can also write specially formatted secrets such as certificates, base64-encoded or binary-encoded data directly to files instead.

But you will still need to `source` this file. For example:

startup.sh

```
#!/bin/bash

# populate the environment with our secrets
source /vault/secrets/database-creds

/my-application -server -addr 0.0.0.0:8080
```


Validating Secrets in your Pod

Once your annotations are correctly configured you can test that the secret was successfully imported in two ways:

1. If the vault sidecar was able to communicate with vault and find your secret the vault init container will complete successfully, if not, it will block - if your pod isn't starting and you find it blocking on the vault-agent-init init container, check the init container logs
2. You can run the following command to check that your secret was successfully placed in the expected file:


```
$ kubectl -n <namespace> exec <your-pod> --container <your-app-container>
export SECRET_TEST="secret"`
```

How do I Get Support for Substrate Vault and SSSM?

If help is needed with Substrate vault, you can reach out to the  [#cloud-ops-support-ext](#) slack channel

Page Information:

Page ID: 81068744

Space: Cloud Developer Platform

Downloaded: 2025-07-12 04:07:27