

Vault Access for K8S Service Accounts (SA)

Downloaded from Epic Games Confluence

Date: 2025-07-12 04:07:47

Original URL: <https://confluence-epicgames.atlassian.net/wiki/spaces/CDE/pages/81068332>

Document Level Classification

200

- [Known Issues](#)
 - [Secret Path Not Shared With SA](#)
 - [Missing SA Annotations](#)
 - [Can't Revoke Vault Access For SA](#)
 - [Can't Find Secret Path in SSSM](#)

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.

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

Known Issues

Secret Path Not Shared With SA

After creating your custom Service Account you will be faced with the fact that vault injector cannot get secrets and you will see about the following errors in the logs:

```
2022-12-20T12:10:58.601Z [INFO] auth.handler: authenticating
2022-12-20T12:10:58.605Z [ERROR] auth.handler: error authenticating:
  error=
  | Error making API request.
  |
  | URL: PUT https://vault.substrate.on.epicgames.com/v1/auth/kubernetes/dead-dev/dead-dev-eks124-upgrade/login
  | Code: 400. Errors:
  |
  | * invalid role name "dead-dev.dead-dev-eks124-upgrade.eks-components-tests.sct-vaultinjector-validator-1"
    backoff=43.03s
2022-12-20T12:11:41.641Z [INFO] auth.handler: authenticating
2022-12-20T12:11:41.644Z [ERROR] auth.handler: error authenticating:
  error=
  | Error making API request.
  |
  | URL: PUT https://vault.substrate.on.epicgames.com/v1/auth/kubernetes/dead-dev/dead-dev-eks124-upgrade/login
  | Code: 400. Errors:
  |
  | * invalid role name "dead-dev.dead-dev-eks124-upgrade.eks-components-tests.sct-vaultinjector-validator-1"
```

This happens because by default the new service account doesn't have access rights in vault for some secrets. To solve this you need to use [SSSM](#). You can use it to give access to the vault secrets both to k8s SA, teams or even IAM roles.

In order for our service account to have access to vault secrets we need share secrets.

Important

Only the owner of the path can share access to vault secrets. If you do not see the necessary secrets, then you do not own them.

To share secrets with Kubernetes, at first expand the namespace by clicking its row, and then select the Kubernetes tab. Next, click the **SHARE WITH SERVICE ACCOUNT** button.

- You can use the filter box to the right of **Kube Namespaces** to search for a specific AWS account, cluster, or namespace.
- You can share multiple service accounts by entering them in the field in the **Kube Service Accounts** column.
- There are many service accounts in Kubernetes. 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 service account from your selection.

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.

Kubernetes 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 Kubernetes paths as specific as possible.

After these actions in the logs will be about the following and the service will work as it should:

```
2022-12-20T12:12:55.122Z [INFO] auth.handler: authenticating
2022-12-20T12:12:55.174Z [INFO] auth.handler: authentication successful, sending token to sinks
2022-12-20T12:12:55.174Z [INFO] auth.handler: starting renewal process
2022-12-20T12:12:55.174Z [INFO] template.server: template server received new token
2022-12-20T12:12:55.174Z [INFO] (runner) stopping
2022-12-20T12:12:55.174Z [INFO] (runner) creating new runner (dry: false, once: false)
2022-12-20T12:12:55.174Z [INFO] sink.file: token written: path=/home/vault/.vault-token
2022-12-20T12:12:55.174Z [INFO] sink.server: sink server stopped
2022-12-20T12:12:55.174Z [INFO] sinks finished, exiting
2022-12-20T12:12:55.174Z [INFO] (runner) creating watcher
2022-12-20T12:12:55.175Z [INFO] (runner) starting
2022-12-20T12:12:55.199Z [INFO] (runner) rendered "(dynamic)" => "/tmp/vault-validator.txt"
2022-12-20T12:12:55.199Z [INFO] (runner) stopping
2022-12-20T12:12:55.199Z [INFO] template.server: template server stopped
2022-12-20T12:12:55.199Z [INFO] auth.handler: shutdown triggered, stopping lifetime watcher
2022-12-20T12:12:55.199Z [INFO] auth.handler: auth handler stopped
2022-12-20T12:12:55.199Z [INFO] (runner) received finish
```

Missing SA Annotations

A minimum [epic-app](#) configuration template for working with secrets is shown below.

In this example we:

1. Create an application `example-app` that needs to use secrets.
2. For this application we will create an SA `custom-sa` which will be located in the same `NS` as the application (the `NS` is specified during the installation of the helm chart, in this example it is `eks-components-tests`).

For security reasons, it is recommended not to use a default service account and instead create and use a separate SA for each application

If you are redeploying the application with a new service account or a service account name changed, please make sure that all vault secrets paths are shared with this new service account in SSSM. Otherwise your application won't be able to receive secrets from Vault

3. Through the pod annotations we will specify SA which will receive secrets, for this, in `vault.hashicorp.com/role` we will specify `cluster_account.cluster_name.namespace.sa_name` (in this case `dead-dev.dead-dev-dontwork.eks-components-tests.custom-sa`).

After creating a custom SA, you need to wait for it to grow into the SSSM in order to give it access to the secrets

4. Specify the file where the secrets will be planted and in which folder they will be stored (`/tmp/newrelic-key`).
5. Point the way to the secret in the vault (`secret/infrastructure/example-app/use1a/dev/runtime/newrelic`).
6. And also in the template we will specify which fields from this secret and in what form they will be substituted in the file.

vaules.yaml

```
epic-app:
  loadbalancers:
```

```

public:
  hosts:
    - host: example-app.dead.dev.us1a.on.epicgames.com
  port:
    number: 8000
  annotations:
    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:us-east-
podAnnotations:
  vault.hashicorp.com/agent-inject: "true"
  vault.hashicorp.com/role: "dead-dev.dead-dev-dontwork.eks-component
  vault.hashicorp.com/agent-inject-secret-newrelic-key: "secret/infra
  vault.hashicorp.com/secret-volume-path-newrelic-key: /tmp
  vault.hashicorp.com/agent-inject-template-newrelic-key: |
    {{ with secret "secret/infrastructure/example-app/us1a/dev/runt
      export NEW_RELIC_API_KEY="{{ .Data.data.apiKey }}"
    {{- end }}
serviceAccount:
  create: true
  name: "custom-sa"
containers:
  example-app:
    image:
      name: hub.01.epicgames.net/substrate/example-app
    environmentValueFrom:
      HOSTNAME_FQDN:
        fieldRef: status.hostIP
  resources:
    limits:
      cpu: 200m
      memory: 128Mi
    requests:
      cpu: 200m
      memory: 128Mi

```

In order to get PodAnnotations you can use the automatic generator from SSSM. To do this, click on the **HELM** icon next to the name of the desired account.

The screenshot shows the AWS IAM console interface. At the top, there are tabs for 'Teams', 'Kubernetes', and 'AWS'. Below the tabs, there is a button labeled '+ SHARE WITH SERVICE ACCOUNT'. A dark tooltip box is overlaid on the interface, containing the text 'Share individual Vault secret path access with Kubernetes Cluster(s)'. A red arrow points from this tooltip to a small icon (a document with a key) next to the first service account in the list. The service accounts are listed under the heading 'Service Accounts:'. The list includes various service accounts with names like 'dead-dev.dead-dev-bad.eks-components-tests.default', 'dead-dev.dead-dev-bad.eks-components-tests.sct-nri-bundle-validator', etc. At the bottom of the list, there is a link 'EDIT SERVICE ACCOUNTS'. The status 'Last Updated: 2 hours ago' is shown at the bottom left, and 'Stat' is at the bottom right.

This will open a generator that allows you to specify the name of the secret, vault path, as well as the necessary fields and on this basis will generate annotations.

When using `epic-app`, replace "`annotations:`" in the generated output with "`podAnnotations:`".

Pod Annotations Generator

1. Set the secret name

database-creds

The secret name will be the filename of the rendered secret, `database-creds` written to the `/vault/secrets` directory, the name must consist of alphanumeric characters, `-`, `_` or `.`

2. Specify the Vault path

secret/ops/components-testing/global/live/runtime/<child-path>

Specify the vault path for the secret you want to retrieve from [Substrate vault](#)

3. Add fields in the secret templates


PGSQL_USERNAMEusername

PGSQL_PASSWORDpassword

+ Add field

Specify fields in the secret template to export environment variables, get your secrets from your specified [Vault path](#)

4. Inject secrets into the pod

Copy the annotations to the [helm chart](#) or Kubernetes configuration files by clicking the icon  below

```
annotations:
  vault.hashicorp.com/agent-inject: "true"
  vault.hashicorp.com/agent-pre-populate-only: "true"
  vault.hashicorp.com/role: "dead-dev.dead-dev-bad.eks-components-tests.default"
  vault.hashicorp.com/agent-inject-secret-database-creds: "secret/ops/components-testing/global/live/runtime/<child-path>"
  vault.hashicorp.com/agent-inject-template-database-creds: |
    {{ with secret "secret/ops/components-testing/global/live/runtime/<child-path>" -}}
      export PGSQL_USERNAME='{{ .Data.data.username }}'
      export PGSQL_PASSWORD='{{ .Data.data.password }}'
```

Close

Can't Revoke Vault Access For SA

To remove 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.

TeamsKubernetesAWS

+ SHARE WITH SERVICE ACCOUNT

secret/ops/components-testing/global/live/runtime/*

Service Accounts:

dead-dev.dead-dev-bad.eks-components-tests.default

dead-dev.dead-dev-bad.eks-components-tests.sct-nri-bundle-validator

dead-dev.dead-dev-dontwork.eks-components-tests.default

dead-dev.dead-dev-dontwork.eks-components-tests.sct-nri-bundle-validator

dead-dev.dead-dev-eks124-upgrade.eks-components-tests.sct-nri-bundle-validator

dead-dev.dead-dev-eks124-upgrade.eks-components-tests.sct-vaultinjector-validator

dead-dev.dead-dev-infraops.eks-components-tests.default

dead-dev.dead-dev-substrate.eks-components-tests.default

dead-dev.dead-dev-substrate.eks-components-tests.sct-nri-bundle-validator

EDIT SERVICE ACCOUNTS

Teams: team-online-infra-ops

EDIT TEAMS

Last Updated: yesterday

Status: LIVE

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

<input checked="" type="checkbox"/>	dead-dev	358323286340	us-east-1	dead-dev-eks124-upgrade	eks-components-tests	<div> <div>sct-nri... x</div> <div>sct-va... x ✓</div> </div>
<input type="checkbox"/>	dead-dev	358323286340	us-east-1	dead-dev-ui	kube-node-lease	default
<input type="checkbox"/>	cdbb-dev	253791803581	us-east-1	cdbb-dev-eos-backend	team-bad	default
	dcaf-dev-					


< 1 2 3 4 5 ... 223 >

Cancel

UPDATE

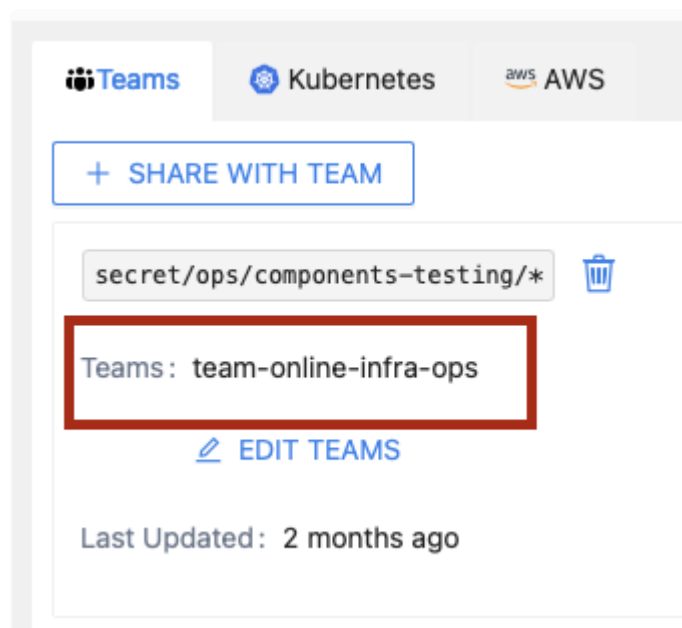
Revoking access from a Kubernetes service account will not have an immediate effect for two reasons. First, removing the policy is not immediate. Second, pods that were already deployed and already had

access to Vault will not be affected. The updated policies will only apply to new pods that launch after the new configuration is applied.

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-support-ext](#) for faster response.


Can't Find Secret Path in SSSM

If during the search for a secret you notice that you do not see the desired Vault Path in SSSM, then you are not the owner of this path. To gain access to this secret, its owner must share the secret with you. To do this, owner can use SSSM and the tab **Teams**. Here you can see all the commands that have access to secrets in the desired path.



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). Click the **Share With Team** button.

 Teams

 Kubernetes

 AWS

[+ SHARE WITH TEAM](#)

secret/ops/components-testing/*



Teams: team-online-infra-ops

[EDIT TEAMS](#)

Last Updated: 2 months ago

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.

Share With Team

Vault Path: / / / / [Help](#)

Vault Path Preview:

Teams: * (1/299)

13 items Unshared teams

[x](#)

☐ team-online-infra-auto-interns
☐ team-online-infra-producers
☐ team-online-infra-cost-analysis
☐ team-online-infra-platform-interns
☐ team-online-infra-platform-contractors
☐ team-analytics-infra
☒ team-online-infra-auto
☐ team-online-infra-cost-analysis-contractors
☐ team-online-infra-btools

1 item Shared teams

[Q](#)

☐ team-online-infra-ops

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.

Page Information:

Page ID: 81068332

Space: Cloud Developer Platform

Downloaded: 2025-07-12 04:07:47