



Vault Kubernetes Integration

February 2023

Copyright © 2021 HashiCorp



Agenda

1. Helm Chart for Vault
2. Pod Secret Access
3. Vault Agent Injector
4. Container Storage Interface
5. Resources

01

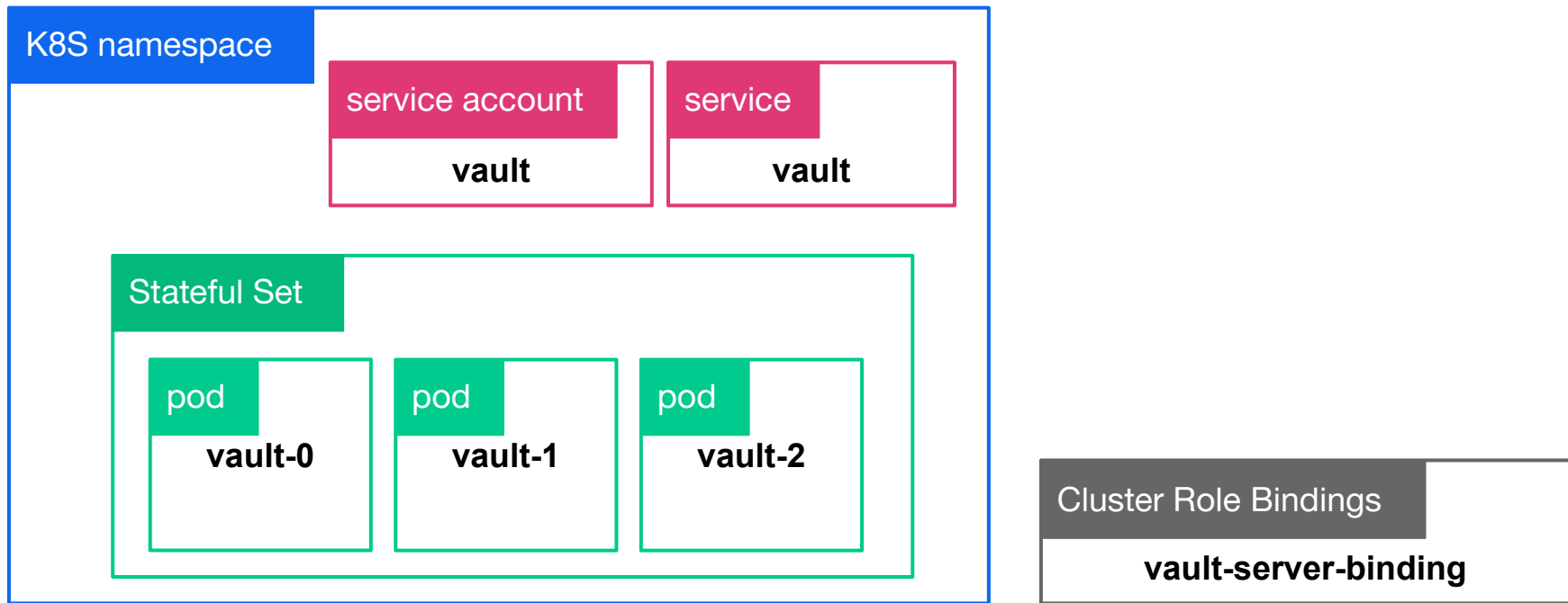
Helm Chart for Vault

Helm Chart for Vault



- Deployment via Helm is the recommended installation and configuration method for Vault on Kubernetes
- The Helm chart can be used to install a Vault server cluster and/or the Agent Injector
- Managing your Vault deployment using Helm can also simplify lifecycle management of your Vault Servers
- Vault Helm chart is compatible with Helm 3.6+ and Kubernetes 1.16+

Vault in Kubernetes



Vault Helm Chart

[hashicorp/vault-helm](https://github.com/hashicorp/vault-helm)



The screenshot shows the GitHub repository page for `hashicorp/vault-helm`. The repository is public and has 53 watches, 645 stars, and 544 forks. The main branch is `main`, with 6 branches and 28 tags. The repository contains a file tree with the following files and their commit history:

File	Description	Commit	Time
<code>.circleci</code>	fix chart publish job (#620)	9fa25e9	2 months ago
<code>.github</code>	Update jira action (#644)		16 days ago
<code>templates</code>	remove support for the leader-elector container (#649)		15 days ago
<code>test</code>	vault-helm 0.18.0 release (#650)		15 days ago
<code>.gitignore</code>	feature: Support configuring various properties as YAML directly...		5 months ago
<code>.helmignore</code>	Ignore bin dirs		3 years ago
<code>CHANGELOG.md</code>	vault-helm 0.18.0 release (#650)		15 days ago
<code>CONTRIBUTING.md</code>	vault-helm default branch is now <code>main</code> (#618)		2 months ago
<code>Chart.yaml</code>	vault-helm 0.18.0 release (#650)		15 days ago
<code>LICENSE.md</code>	Add license		3 years ago

The repository also includes a sidebar with the following sections:

- About**: Helm chart to install Vault and other associated components.
- Releases**: 27 releases, with the latest being `v0.18.0` (15 days ago).
- Packages**: No packages published.



Helm Repository

```
> helm repo add hashicorp \
https://helm.releases.hashicorp.com
"Hashicorp" has been added to your repositories
```

```
> helm search repo
hashicorp/consul ...
hashicorp/vault ...
```

```
> helm install vault hashicorp/vault
NAME: vault
...
```

Default Values



```
# ...
```

```
server:
```

```
# Run Vault in "dev" mode. This requires no further setup, no ...
```

```
# and no initialization. This is useful for experimenting with ...
```

```
# needing to unseal, store keys, et. al. All data is lost on ...
```

```
# use dev mode for anything other than experimenting.
```

```
# See https://www.vaultproject.io/docs/concepts/dev-server.html ...
```

```
dev:
```



```
enabled: false
```

```
--set "server.dev.enabled=true"
```


Override Default Values in a File



```
server:
  affinity: ""
  ha:
    enabled: true
```



Licensing

```

> secret=$(cat licensefile.hclic)

> kubectl create secret generic vault-ent-license
--from-literal="license=${secret}"

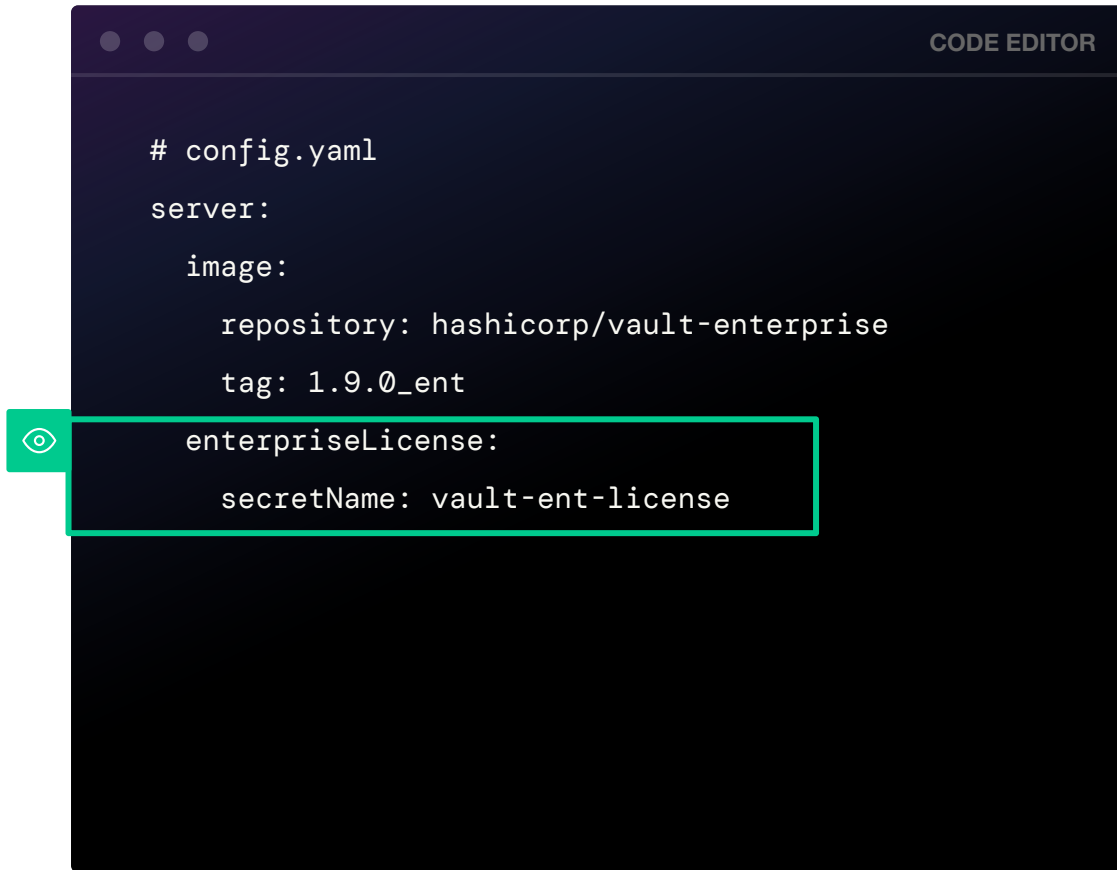
> helm install hashicorp hashicorp/vault -f config.yaml

> kubectl exec -ti vault-0 -- vault license get

```



Licensing

A code editor window with a dark theme. The title bar on the right says "CODE EDITOR". The code is in a light blue font. The configuration is for Vault Enterprise. The "enterpriseLicense" section is highlighted with a red box, and a red eye icon is next to it.

```
# config.yaml
server:
  image:
    repository: hashicorp/vault-enterprise
    tag: 1.9.0_ent
enterpriseLicense:
  secretName: vault-ent-license
```

Primary HA Vault ENT Cluster Deployment



```

> secret=$(cat licensefile.hcllic)

> > kubectl create secret generic vault-ent-license
--from-literal="license=${secret}"

> helm install vault hashicorp/vault \
  --set='server.image.repository=hashicorp/vault-enterprise' \
  --set='server.image.tag=1.9.0_ent' \
  --set='server.ha.enabled=true' \
  --set='server.ha.raft.enabled=true' \
  --set='server.enterpriseLicense.secretName=vault-ent-license'
```

Primary HA Vault ENT Cluster Deployment



Initialize cluster and unseal first node

```
> kubectl exec -ti vault-primary-0 -- vault operator init  
> kubectl exec -ti vault-primary-0 -- vault operator unseal
```

Join second pod to raft cluster and unseal

```
> kubectl exec -ti vault-primary-1 -- vault operator raft join \  
http://vault-primary-0.vault-primary-internal:8200  
> kubectl exec -ti vault-primary-1 -- vault operator unseal
```

Join third pod to raft cluster and unseal

```
> kubectl exec -ti vault-primary-2 -- vault operator raft join \  
http://vault-primary-0.vault-primary-internal:8200  
> kubectl exec -ti vault-primary-2 -- vault operator unseal
```

DR HA Vault ENT Cluster Deployment



```
● ● ● TERMINAL

> secret=$(cat licensefile.hclic)

> > kubectl create secret generic vault-ent-license
--from-literal="license=${secret}"

> helm install vault hashicorp/vault \
  --set='server.image.repository=hashicorp/vault-enterprise' \
  --set='server.image.tag=1.9.0_ent' \
  --set='server.ha.enabled=true' \
  --set='server.ha.raft.enabled=true' \
  --set='server.enterpriseLicense.secretName=vault-ent-license'
```

DR HA Vault ENT Cluster Deployment



Initialize cluster and unseal first node

```
> kubectl exec -ti vault-primary-0 -- vault operator init  
> kubectl exec -ti vault-primary-0 -- vault operator unseal
```

Join second pod to raft cluster and unseal

```
> kubectl exec -ti vault-primary-1 -- vault operator raft join \  
http://vault-primary-0.vault-primary-internal:8200  
> kubectl exec -ti vault-primary-1 -- vault operator unseal
```

Join third pod to raft cluster and unseal

```
> kubectl exec -ti vault-primary-2 -- vault operator raft join \  
http://vault-primary-0.vault-primary-internal:8200  
> kubectl exec -ti vault-primary-2 -- vault operator unseal
```



Enable Disaster Recovery Replication

Primary Cluster

TERMINAL

```
> kubectl exec -ti vault-primary-0 -- vault write -f  
sys/replication/dr/primary/enable  
primary_cluster_addr=https://vault-primary-active:8201  
  
> kubectl exec -ti vault-primary-0 -- vault write  
sys/replication/dr/primary/secondary-token id=secondary
```




Enable Disaster Recovery Replication

Secondary Cluster

```
TERMINAL

> kubectl exec -ti vault-secondary-0 -- vault write
sys/replication/dr/secondary/enable token=<TOKEN FROM
PRIMARY>

> kubectl delete pod vault-secondary-1
> kubectl exec -ti vault-secondary-1 -- vault operator
unseal <PRIMARY UNSEAL TOKEN>

> kubectl delete pod vault-secondary-2
> kubectl exec -ti vault-secondary-2 -- vault operator
unseal <PRIMARY UNSEAL TOKEN>
```

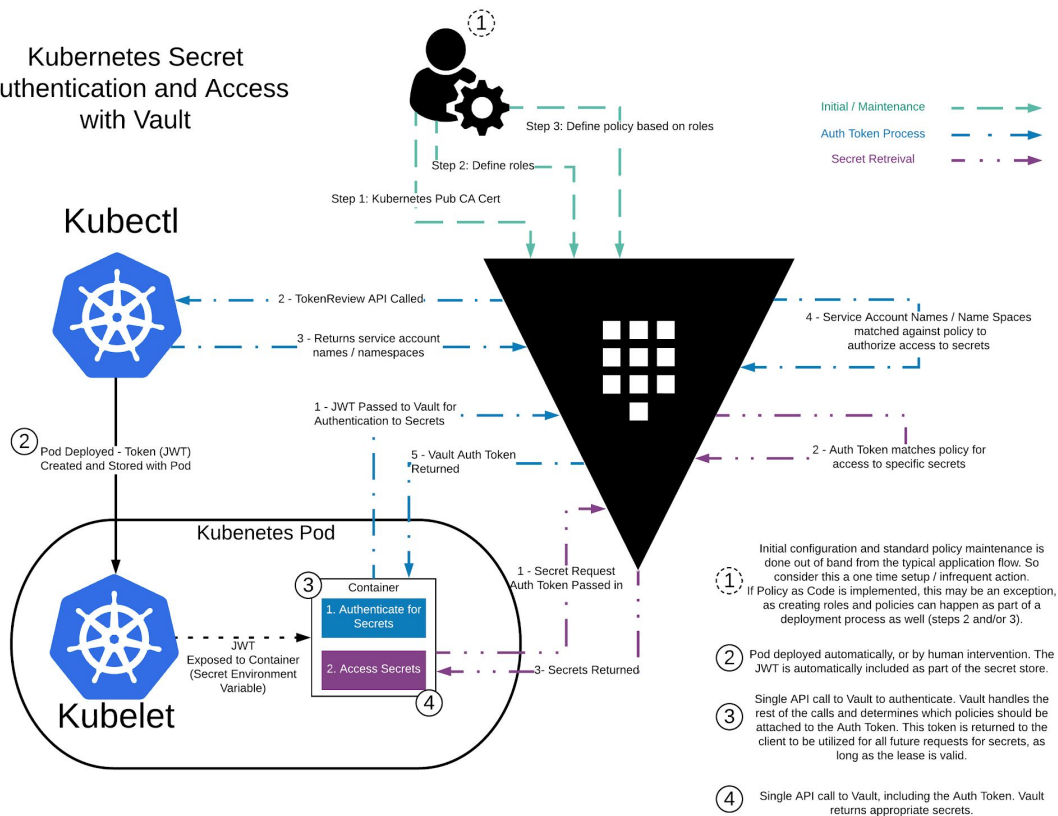
02

Pod Secret Access



Kubernetes Auth Flow

Kubernetes Secret Authentication and Access with Vault





Application Pod Definition

```
apiVersion: v1
kind: Pod
...
spec:
  serviceAccountName: k8s-service-acct
  containers:
    - name: app
      image: burtlo/exampleapp-ruby:k8s
      env:
        - name: VAULT_ADDR
          value:
            "http://vault.default.svc.cluster.local:8200"
        - name: VAULT_ROLE
          value: "internal-app"
```



Example App Code Changes

```
CODE EDITOR

response =
  HTTP.put("#{vault_url}/v1/auth/kubernetes/login") do
    |req|
      req.headers['Content-Type'] = 'application/json'
      req.body = { "role" => vault_role, "jwt" => jwt
    }.to_json
  end

vault_token =
  JSON.parse(response.body)["auth"]["client_token"]

logger.info "Received Vault Token: [#{vault_token}]"
```

03

Vault Agent Injector

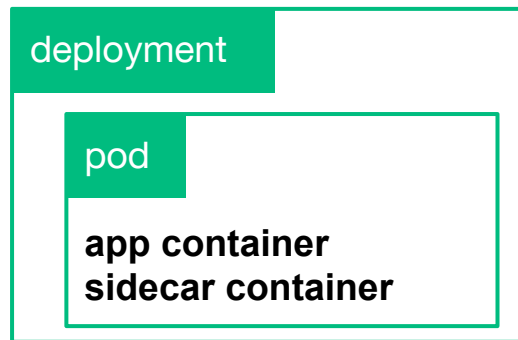


Sidecar Pattern

Vault unaware pods would offload the authentication and secret retrieval to a dedicated container appended to every deployment/pod

Sidecar container needs:

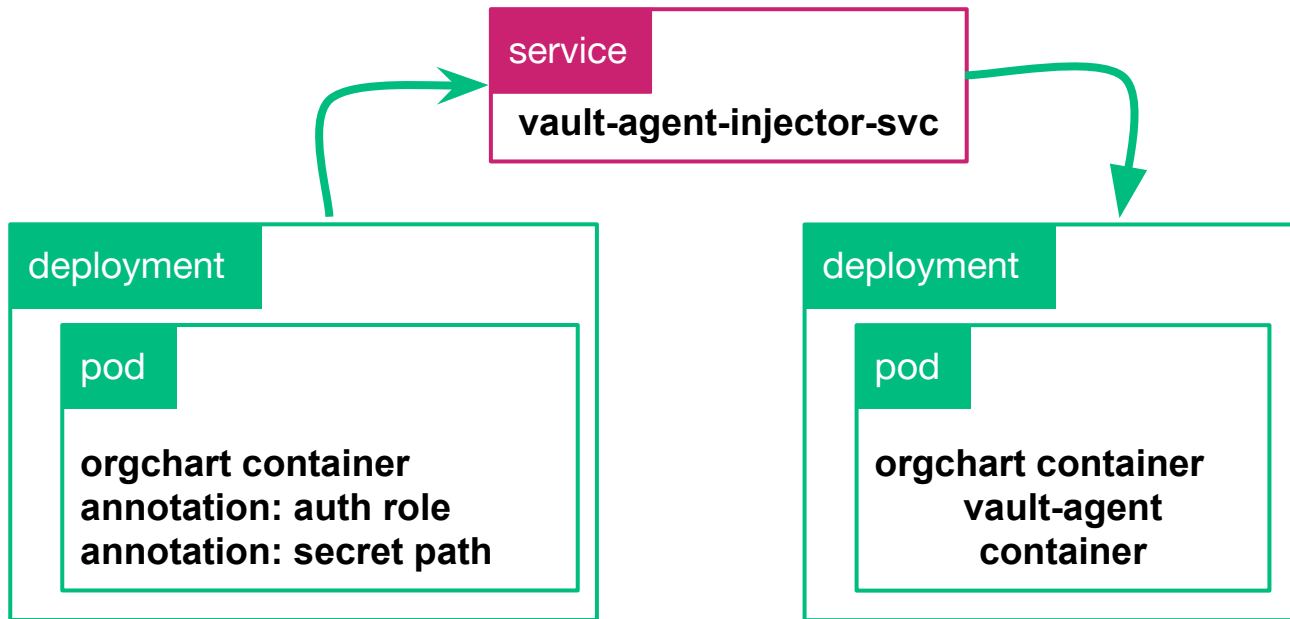
- Vault address
- Vault authentication role
- Vault secret path





Sidecar Pattern

Registers a Mutating Webhook Configuration that takes action when pod/deployment annotations are defined





Install Agent Injector



TERMINAL

```
> helm repo add hashicorp
https://helm.releases.hashicorp.com
"hashicorp" has been added to your repositories

> helm search repo hashicorp/vault
NAME                CHART VERSION   APP VERSION DESCRIPTION
hashicorp/vault 0.18.0          1.9.0           Official
HashiCorp Vault Chart

> helm install vault hashicorp/vault \
--set="injector.enabled=true"
```

Agent Annotations



```
spec:
  template:
    metadata:
      annotations:
        vault.hashicorp.com/agent-inject: "true"
        vault.hashicorp.com/role: "internal-app"
        vault.hashicorp.com/agent-inject-secret-database-config.txt:
"internal/data/database/config"
```



View the Secret

```
> kubectl exec orgchart --container orgchart \
  -- cat /vault/secrets/database-config.txt
```

```
data: map[password:db-secret-password
username:db-readonly-user]
metadata:
map[created_time:2019-12-20T18:17:50.930264759Z
deletion_time: destroyed:false version:2]
```

04

Container Storage Interface

Overview



- Secrets Store CSI driver for Kubernetes secrets - Integrates secrets stores with Kubernetes via a Container Storage Interface (CSI) volume
- The Secrets Store CSI driver allows Kubernetes to mount multiple secrets, keys, and certs stored in enterprise-grade external secrets stores into their pods as a volume
- Once the Volume is attached, the data is mounted into the container's file system

Secrets Store CSI Driver

CSI Driver



The screenshot shows the GitHub repository page for `kubernetes-sigs/secrets-store-csi-driver`. The repository is public and has 17 watchers, 569 stars, and 136 forks. It has 48 issues, 4 pull requests, 1 project, and 1 wiki page. The main branch is selected, showing 6 branches and 28 tags. A recent pull request #814 by `k8s-ci-robot` is highlighted, merged 2 days ago. The commit history shows recent updates to workflows, deprecated flags, and manifests. The right sidebar provides an overview of the project, including its description, links to related projects, and license information.

Repository: `kubernetes-sigs/secrets-store-csi-driver` (Public)

Stats: 17 Watchers, 569 Stars, 136 Forks

Navigation: Code, Issues (48), Pull requests (4), Actions, Projects (1), Wiki, Security, Insights

Branches: main (6 branches), 28 tags

Recent Activity:

- `k8s-ci-robot` Merge pull request #814 from spiffxp/use-k8s-infra-for-... 2 days ago (859 commits)
- `.github` ci: add markdown-link-check workflow 22 days ago
- `.local` chore: remove deprecated `--filtered-watch-secret` flag 23 days ago
- `apis` feat: add SecretProviderClass and SecretProviderClassPodStatu... 2 months ago
- `charts` release: update manifest and helm charts for v1.0.0 2 months ago
- `cmd/secrets-store-csi-driver` chore: remove deprecated `--filtered-watch-secret` flag 23 days ago
- `config` feat: add SecretProviderClass and SecretProviderClassPodStatu... 2 months ago
- `controllers` Issue#636 - add last lint check items 2 months ago
- `deploy` release: update manifest and helm charts for v1.0.0 2 months ago
- `docker` images: use k8s-staging-test-infra/gcb-docker-gcloud 2 days ago
- `docs` docs: fix dead links based on errors 22 days ago

About: Secrets Store CSI driver for Kubernetes secrets - Integrates secrets stores with Kubernetes via a CSI volume.

Related Projects: `kubernetes`, `hashicorp-vault`, `csi`, `azure-keyvault`, `aws-secrets-manager`, `k8s-sig-auth`, `gcp-secret-manager`, `csi-secrets-store`, `mount-multiple-secrets`

License: Apache-2.0 License

Code of conduct

Releases: 20



Install Container Storage Interface



TERMINAL

```
> helm repo add hashicorp
https://helm.releases.hashicorp.com
"hashicorp" has been added to your repositories

> helm search repo hashicorp/vault

NAME                CHART VERSION    APP VERSION DESCRIPTION
hashicorp/vault     0.18.0           1.9.0           Official
HashiCorp Vault Chart

> helm install vault hashicorp/vault \
  --set "injector.enabled=false" \
  --set "csi.enabled=true" \
  --set "injector.externalVaultAddr=http://addr:8200"
```



Install Secrets Store CSI Driver

```

> helm repo add secrets-store-csi-driver \
https://raw.githubusercontent.com/kubernetes-sigs/secrets-store-csi-driver/master/charts
...

> helm install csi
secrets-store-csi-driver/secrets-store-csi-driver
...
```


Define SecretProviderClass



```
apiVersion: secrets-store.csi.x-k8s.io/v1alpha1
kind: SecretProviderClass
metadata:
  name: vault-database
spec:
  provider: vault
  parameters:
    vaultAddress: "http://vault.default.svc.cluster.local:8200"
    roleName: "internal-app"
    objects: |
      - objectName: "db-password"
        secretPath: "internal/data/database/config"
        secretKey: "password"
```

CODE EDITOR

Define a Pod with a Volume



```
spec:
  containers:
    - image: nginx
      name: webapp
      volumeMounts:
        - name: secrets-store-inline
          mountPath: "/mnt/secrets-store"
          readOnly: true
  volumes:
    - name: secrets-store-inline
      csi:
        driver: secrets-store.csi.k8s.io
        readOnly: true
        volumeAttributes:
          secretProviderClass: "vault-database"
```

CODE EDITOR



Pattern Comparison

[Kubernetes Vault Integration via Sidecar Agent Injector vs. CSI Provider](#)

	Agent Sidecar	CSI
Secret projection	Shared Memory Volume Environment Variable	Ephemeral Disk Environment Variables Kubernetes Secrets
Secret scope	Global	Global
Secret types	All Secret Engines (Static & Dynamic)	All Secret Engines (Static & Dynamic)
Secret templating	Yes	No
Secret size limit	No Limit (both storage types)	No Limit (both storage types)
Secret definitions	CLI / API / UI	CLI / API / UI
Encryption	Yes (at rest & in-transit)	Yes (at rest & in-transit)
Secret rotation	Yes	No
Secret caching	Yes	No
Auditability	Yes	Yes
Deployment method	1 Shared K8s Cluster Service + 1 Sidecar Container Per Application Pod	Daemonset
Vault agent support	Yes	No
Helm support	Yes	Yes

05

Resources



Resources

- [Vault on Kubernetes Security Considerations](#)
- [Vault on Kubernetes Reference Architecture](#)
- [Vault Helm Chart](#)
- [Vault Enterprise License Management - Kubernetes](#)
- [Helm Chart Examples](#)
- [Running Vault - OpenShift](#)
- Tutorials - Vault Installation to Managed Kubernetes Services
 - [Google GKE](#)
 - [Azure AKS](#)
 - [Amazon EKS](#)
- [Injecting Secrets into Kubernetes Pods via Vault Agent Containers](#)
- [Mount Vault Secrets through Container Storage Interface \(CSI\) Volume](#)
- [Integrate a Kubernetes Cluster with an External Vault](#)

Q & A



Thank You

customer.success@hashicorp.com

www.hashicorp.com