HashiCorp
Vault

Lunch & Learn: Kubernetes Integration

January 2022

Agenda

- Helm Chart for Vault
- Pod Secret Access
- Vault Agent Injector
- Container Storage Interface
- Resources
- Q & A

01

Helm Chart for Vault

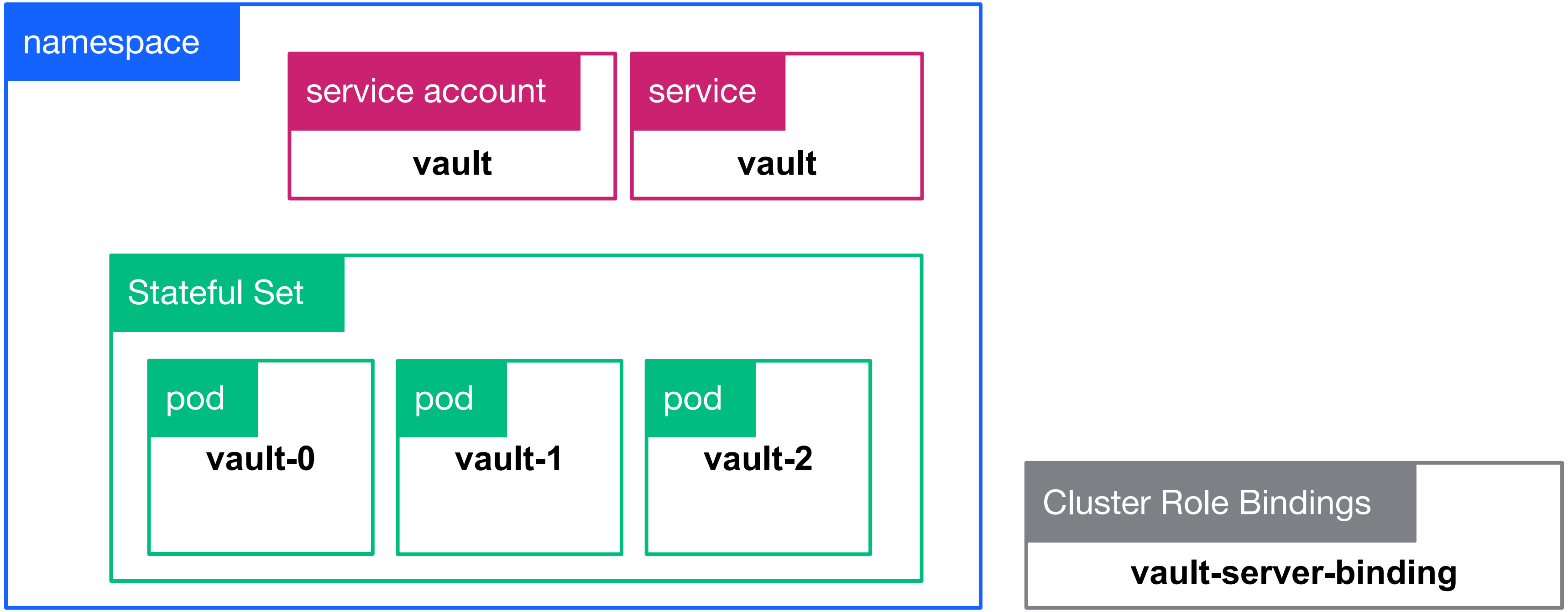
Overview

Recommended way to install and configure Vault on Kubernetes

The Vault helm chart enables operators to easily install and configure 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.

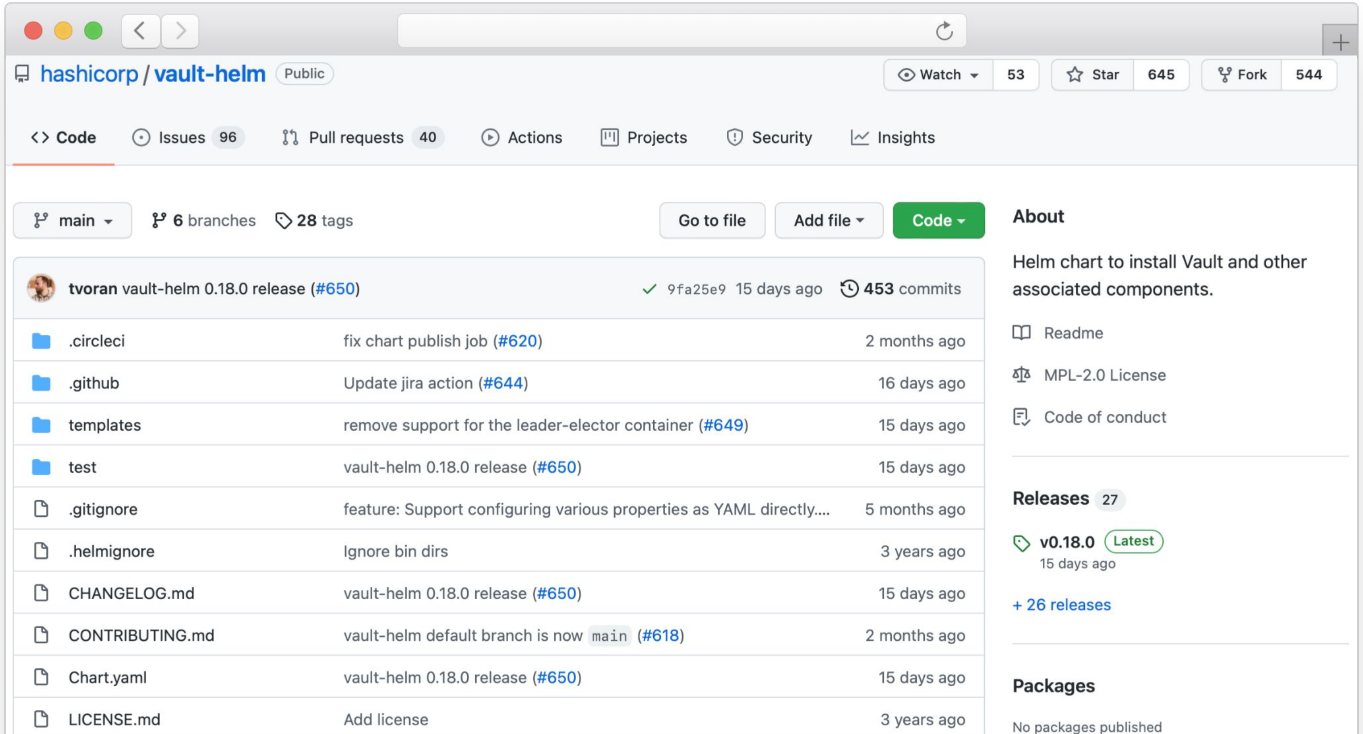
Vault Helm chart is compatible with Helm 3.

Vault in Kubernetes



Vault Helm Chart

hashicorp/vault-helm



tvoran vault-helm 0.18.0 release (#650) ✓ 9fa25e9 15 days ago 453 commits

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

About

Helm chart to install Vault and other associated components.

Readme

MPL-2.0 License

Code of conduct

Releases 27

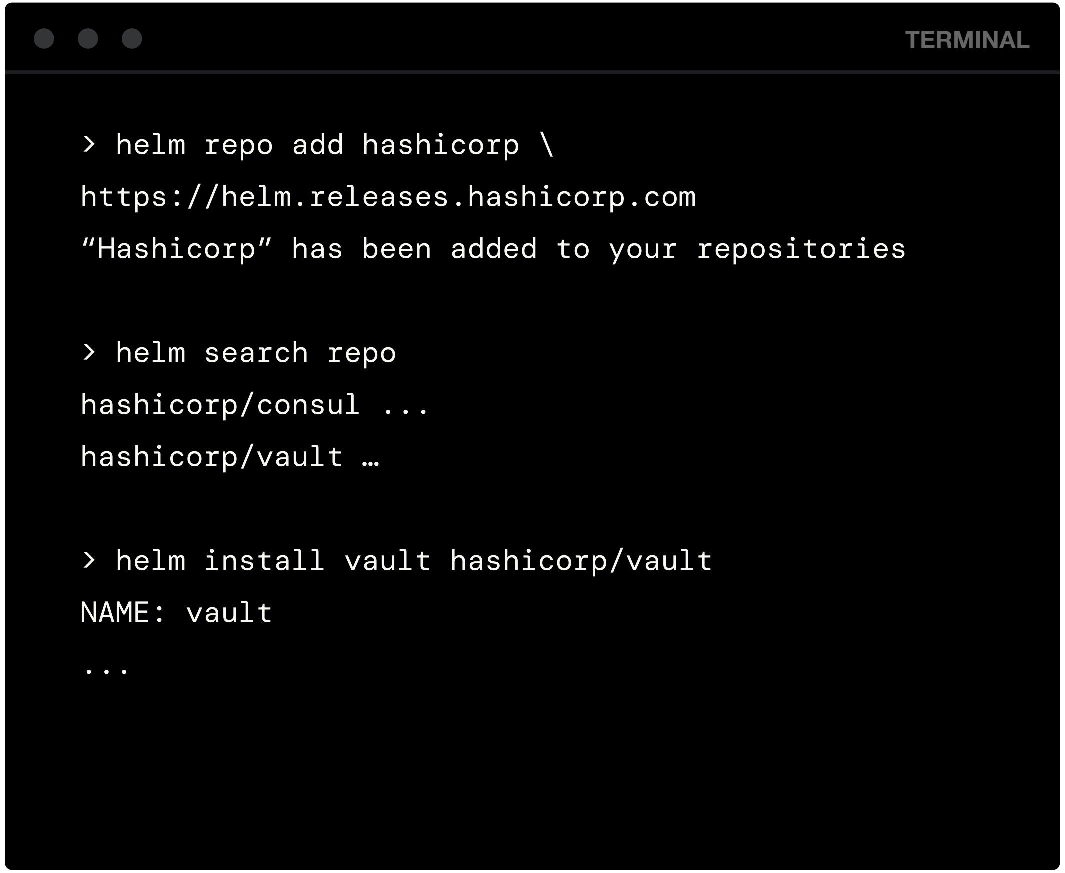
v0.18.0 Latest
15 days ago

+ 26 releases

Packages

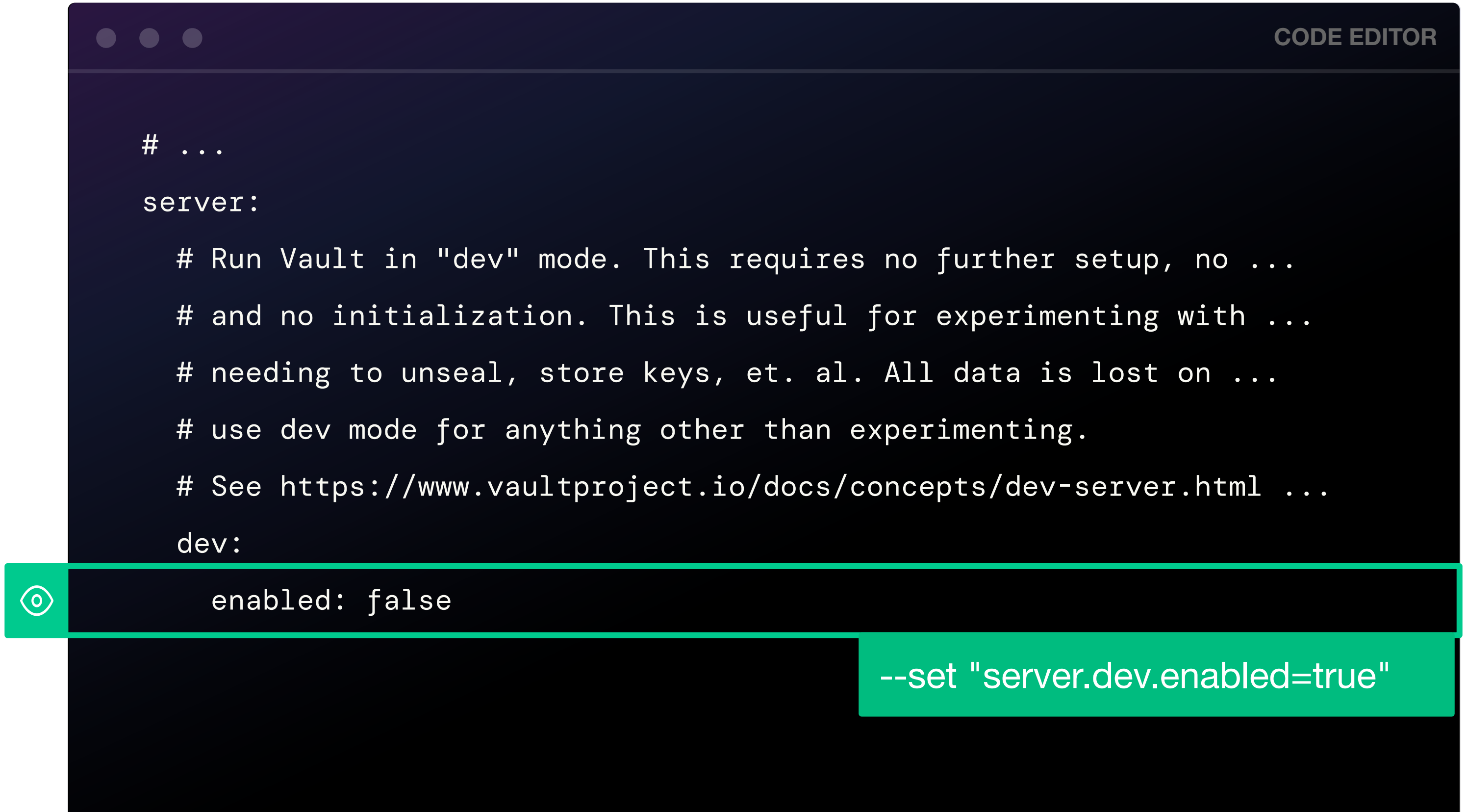
No packages published

Helm Repository


TERMINAL

```
> 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
...
```

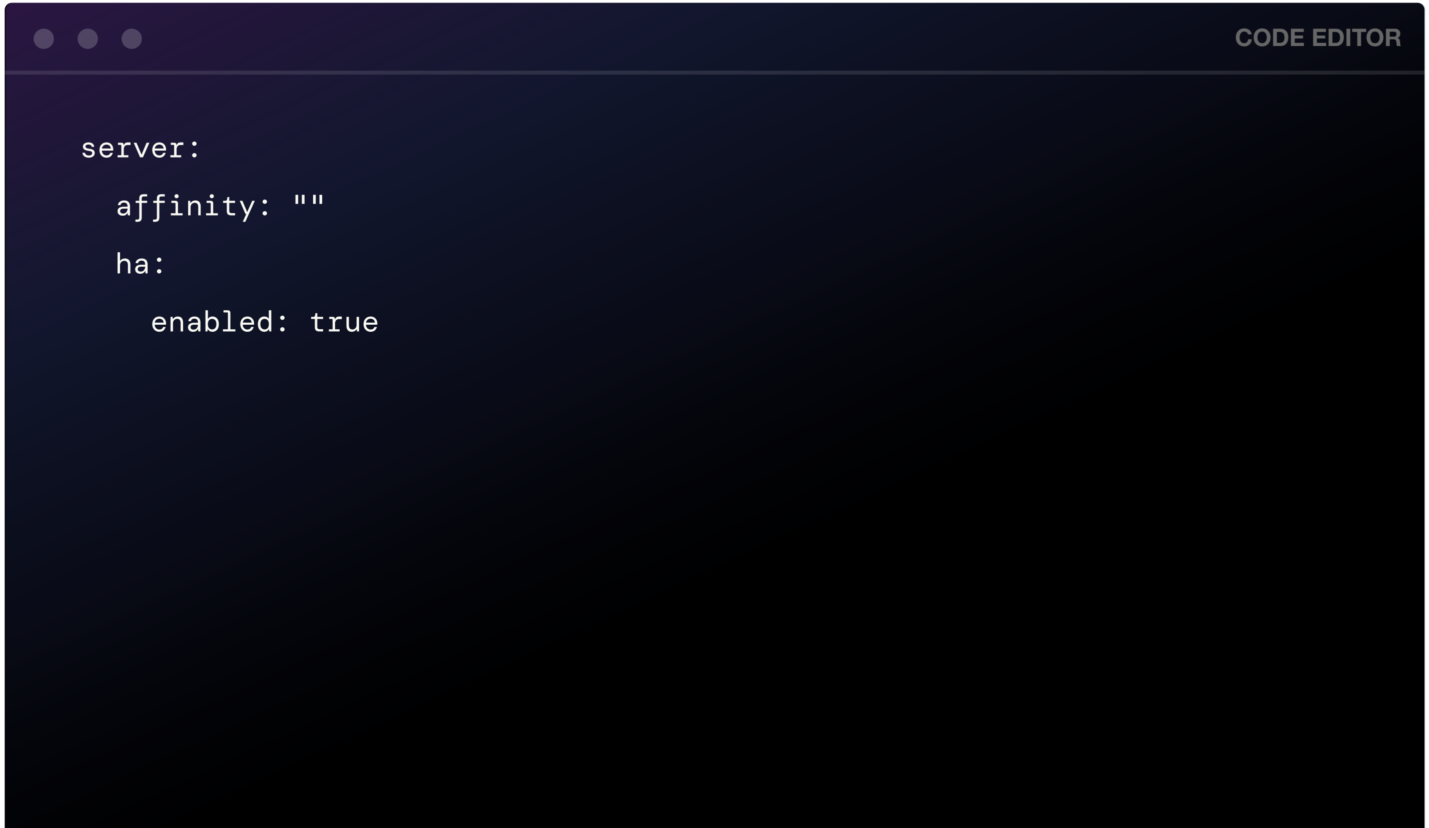
Default Values


CODE EDITOR

```
# ...
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
```

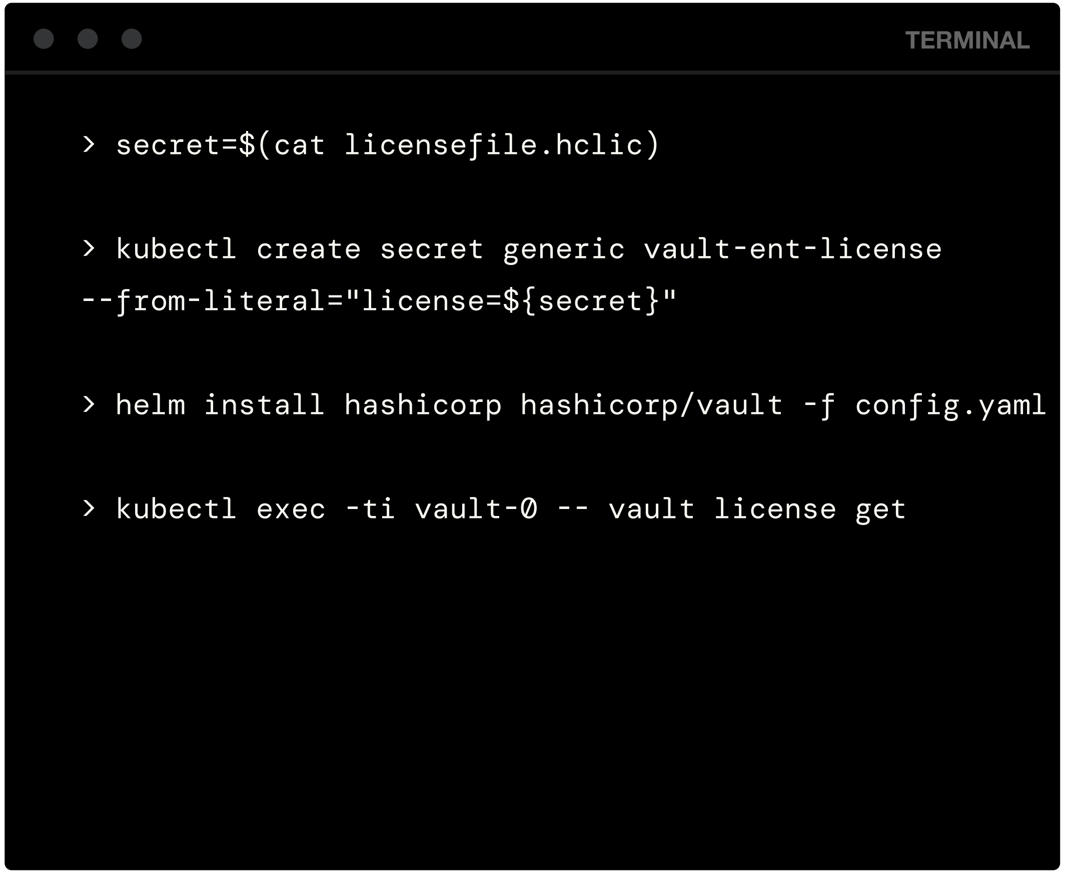
--set "server.dev.enabled=true"

Override Default Values in a File



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

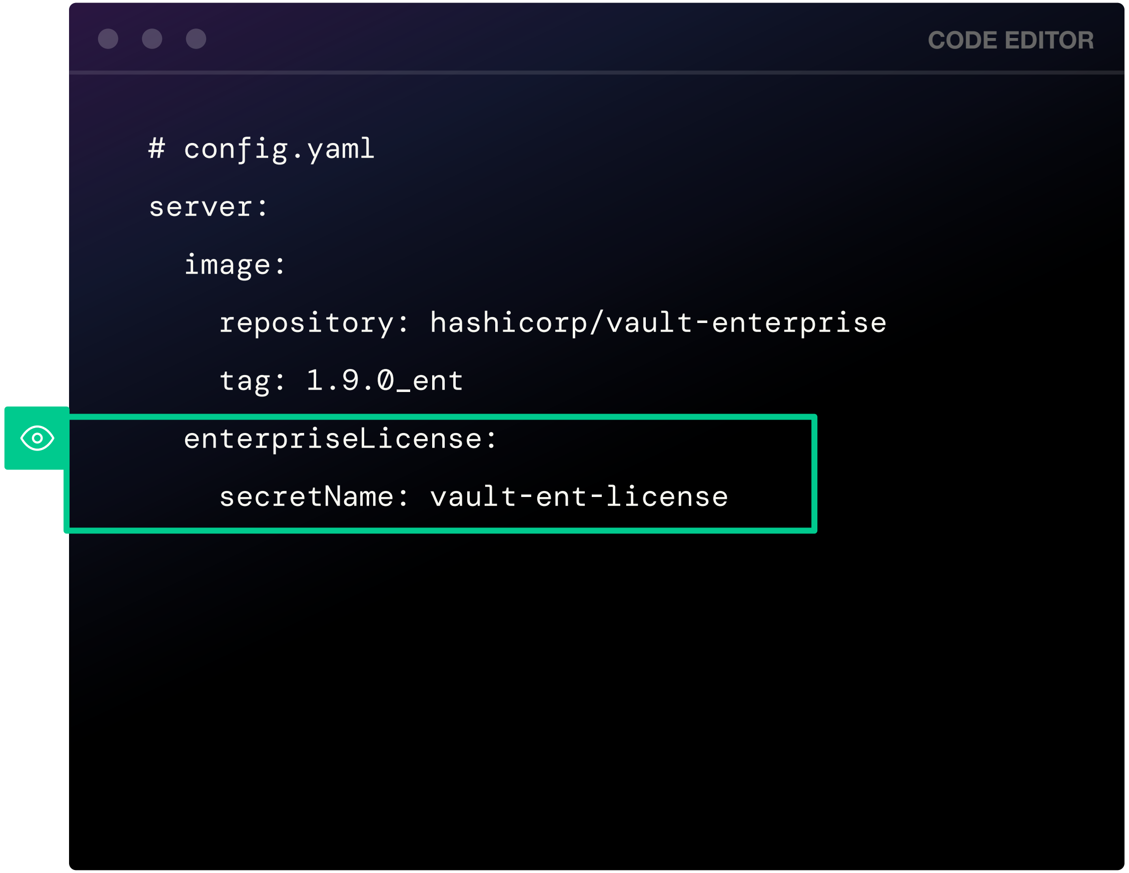
Licensing



TERMINAL

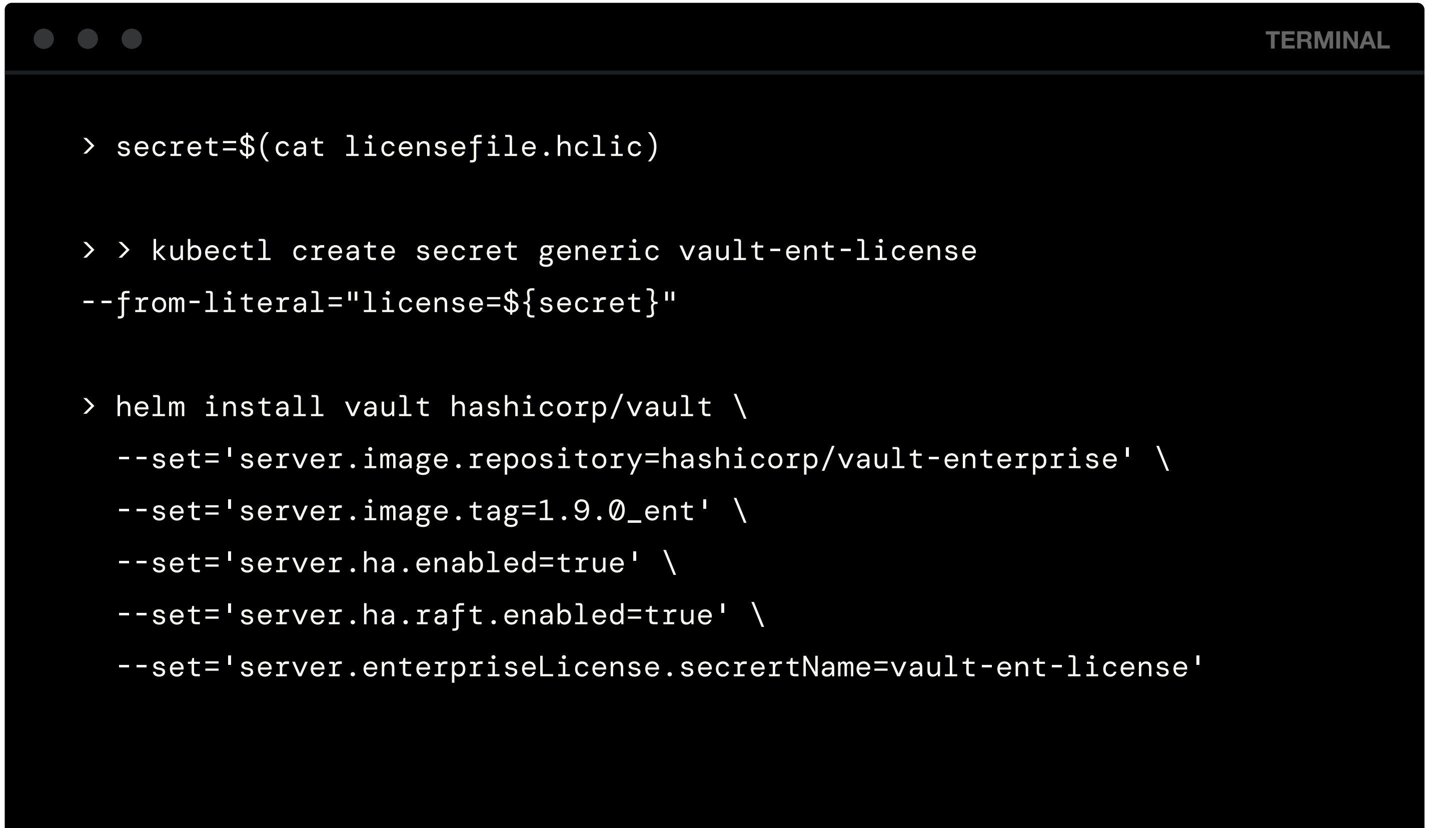
```
> secret=$(cat licensefile.hclic)

> 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



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

Primary HA Vault ENT Cluster Deployment


TERMINAL

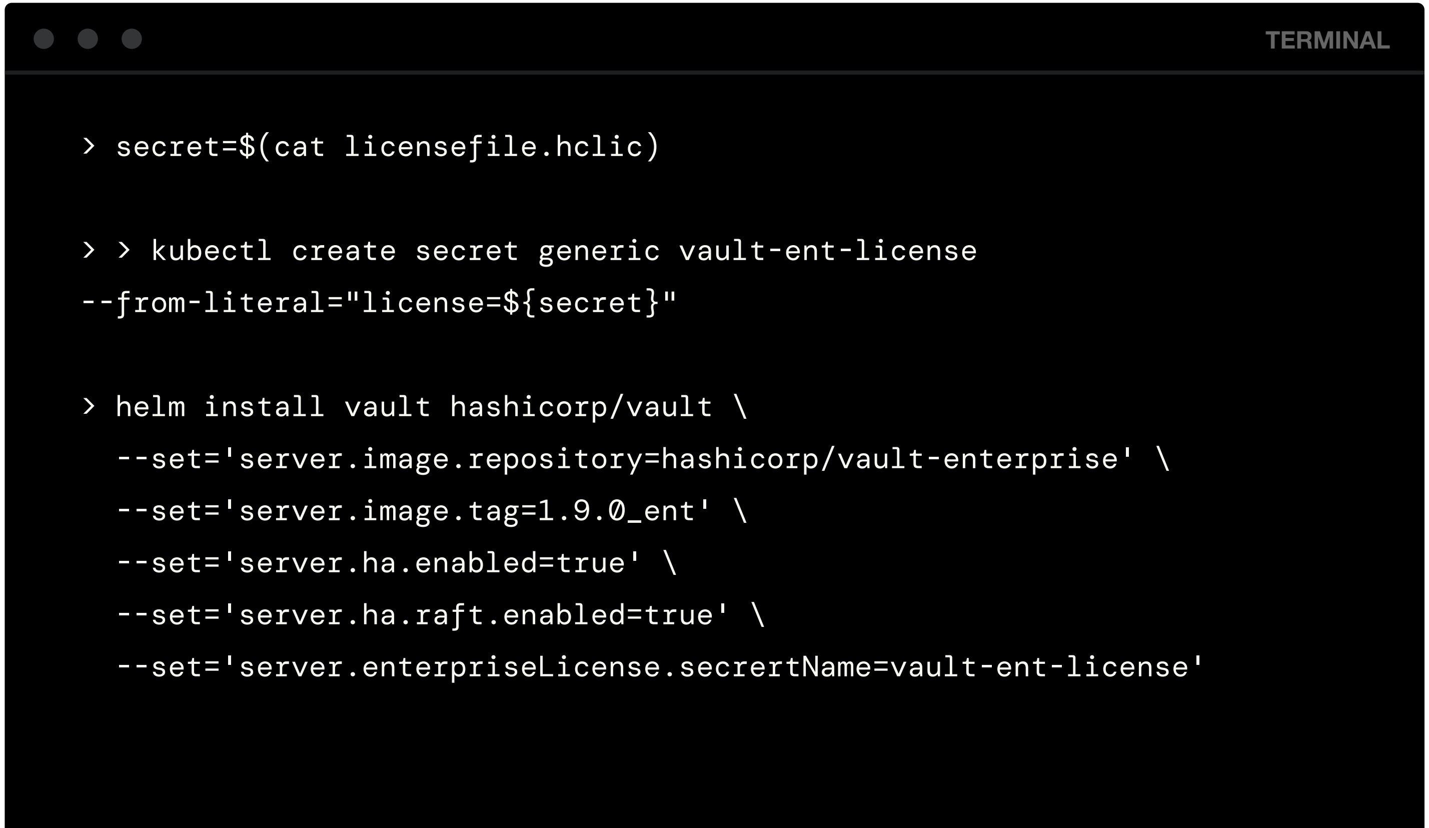
```
> secret=$(cat licensefile.hclic)

> > 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.secrertName=vault-ent-license'
```

Primary HA Vault ENT Cluster Deployment

TERMINAL

```
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
```

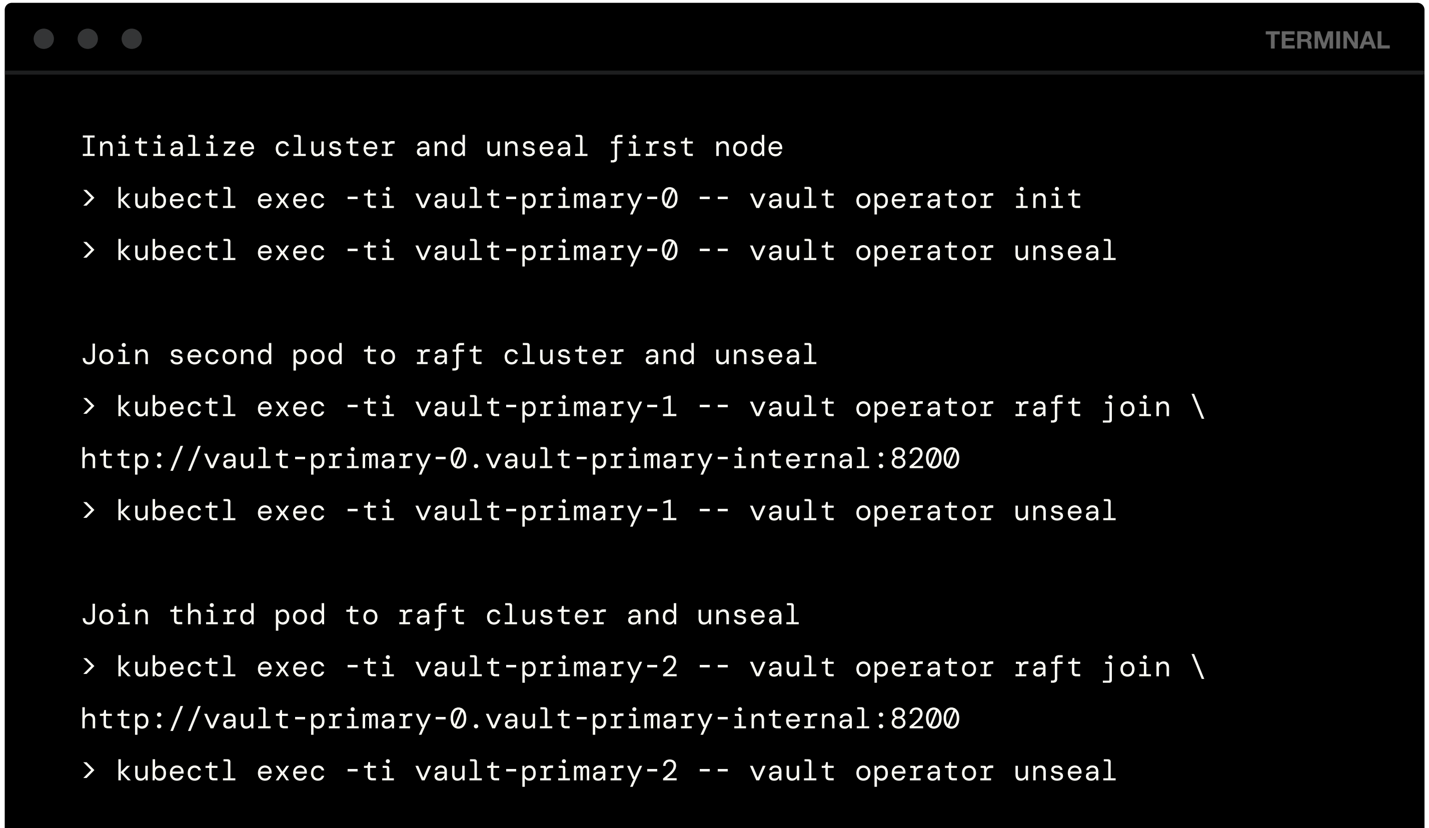
DR HA Vault ENT Cluster Deployment



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.secrertName=vault-ent-license'
```

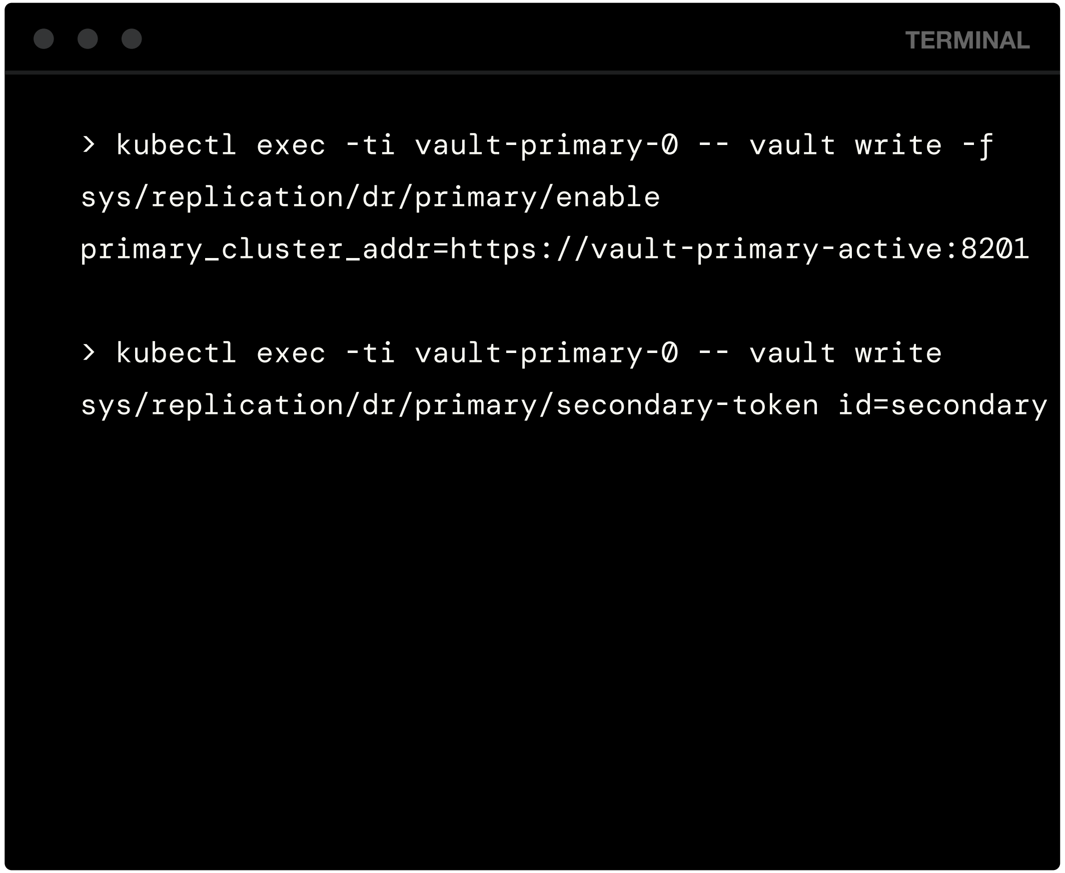
DR HA Vault ENT Cluster Deployment



```
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

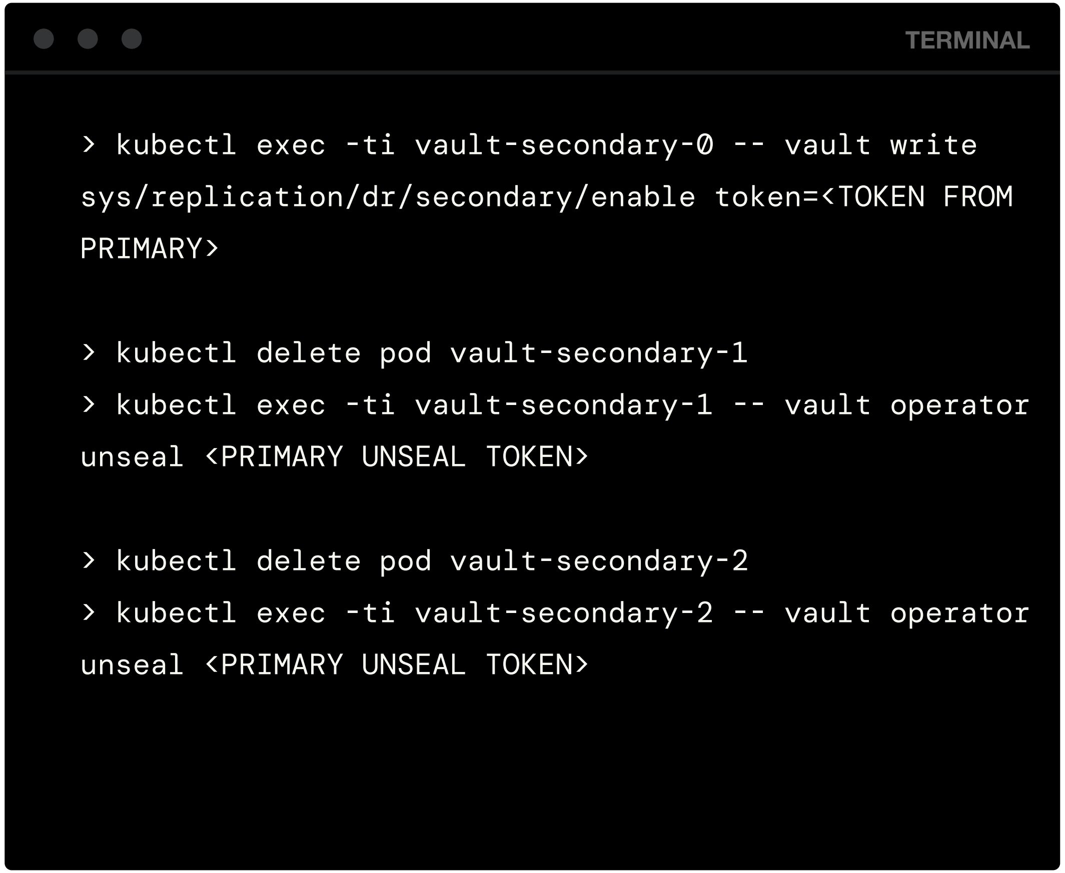


TERMINAL

```
> 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

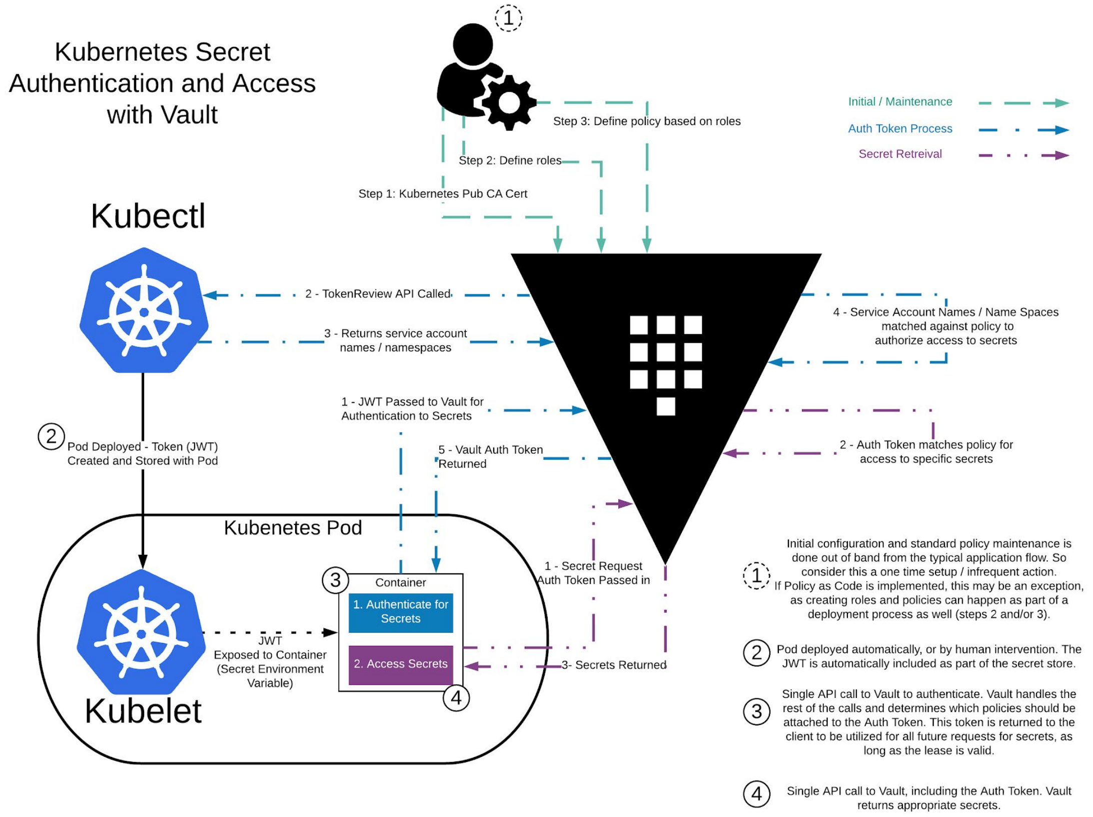

TERMINAL

```
> 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
1
Step 3: Define policy based on roles
Step 2: Define roles
Step 1: Kubernetes Pub CA Cert
Initial / Maintenance
Auth Token Process
Secret Retreival
Kubectl
2 - TokenReview API Called
3 - Returns service account names / namespaces
4 - Service Account Names / Name Spaces matched against policy to authorize access to secrets
1 - JWT Passed to Vault for Authentication to Secrets
5 - Vault Auth Token Returned
2 - Auth Token matches policy for access to specific secrets
2
Pod Deployed - Token (JWT) Created and Stored with Pod
Kubenetes Pod
1 - Secret Request Auth Token Passed in
3
Container
1. Authenticate for Secrets
2. Access Secrets
4
JWT Exposed to Container (Secret Environment Variable)
3- Secrets Returned
Kubelet

1 Initial configuration and standard policy maintenance is done out of band from the typical application flow. So consider this a one time setup / infrequent action. If Policy as Code is implemented, this may be an exception, as creating roles and policies can happen as part of a deployment process as well (steps 2 and/or 3).

2 Pod deployed automatically, or by human intervention. The JWT is automatically included as part of the secret store.

3 Single API call to Vault to authenticate. Vault handles the rest of the calls and determines which policies should be attached to the Auth Token. This token is returned to the client to be utilized for all future requests for secrets, as long as the lease is valid.

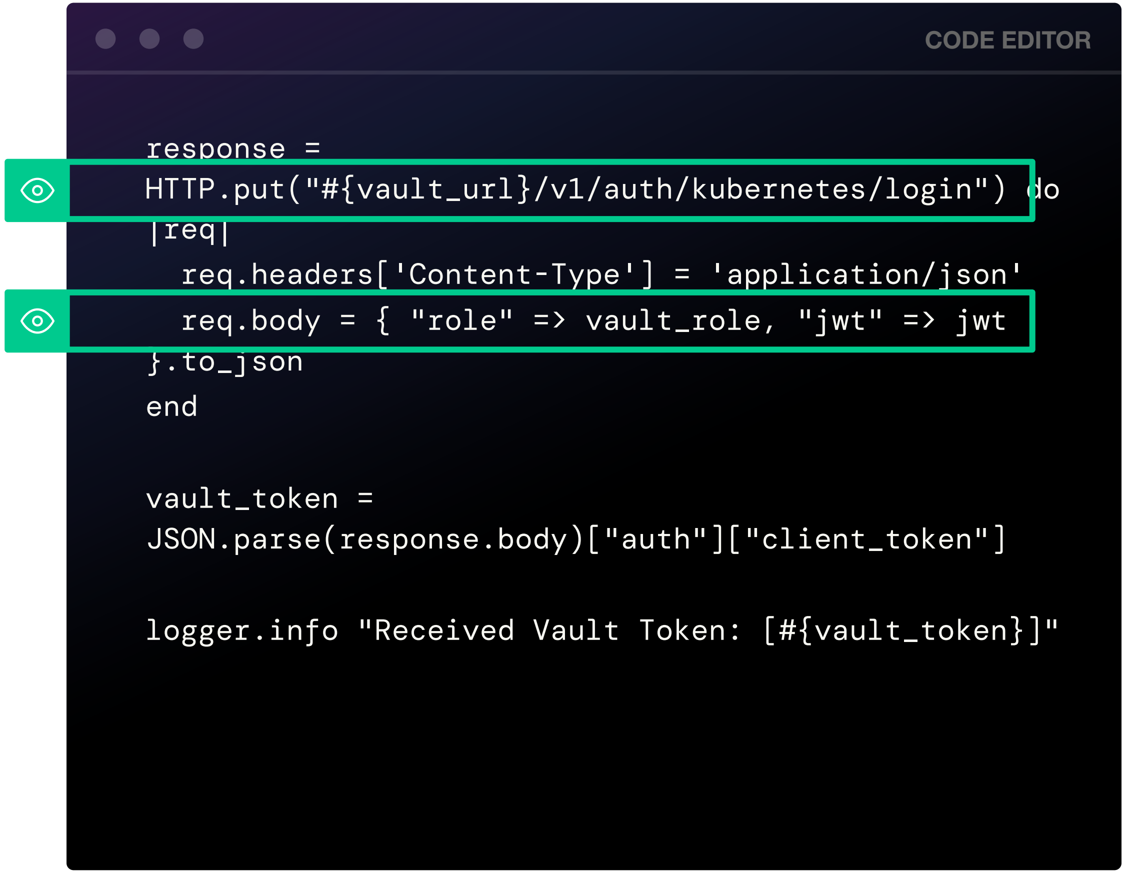
4 Single API call to Vault, including the Auth Token. Vault returns appropriate secrets.

Application Pod Definition

CODE EDITOR

```
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



```
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

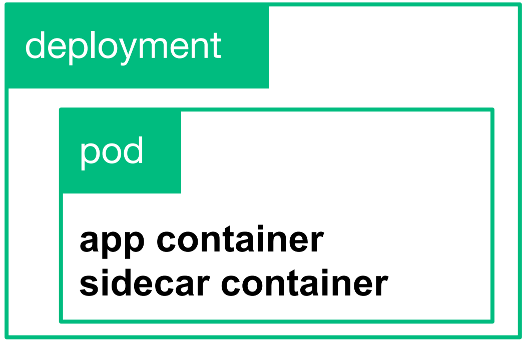
Vault Agent Injector

Sidecar Pattern

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

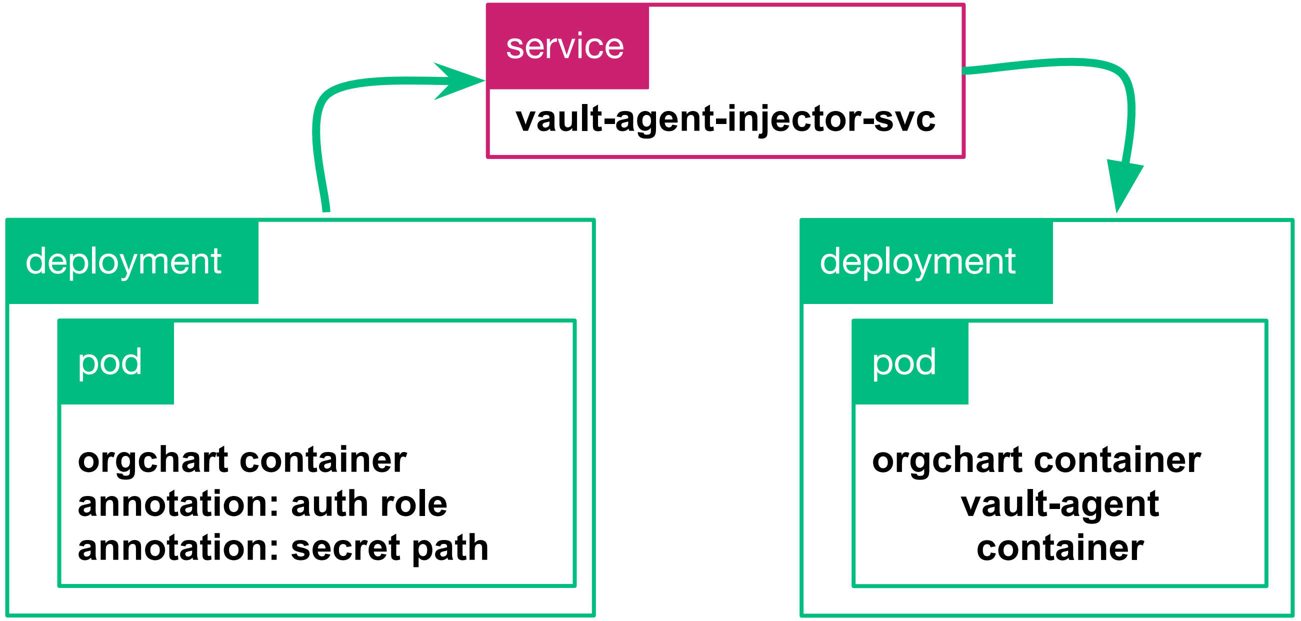
Sidecar container needs:

- Vault address
- Vault authentication role
- Vault secret path

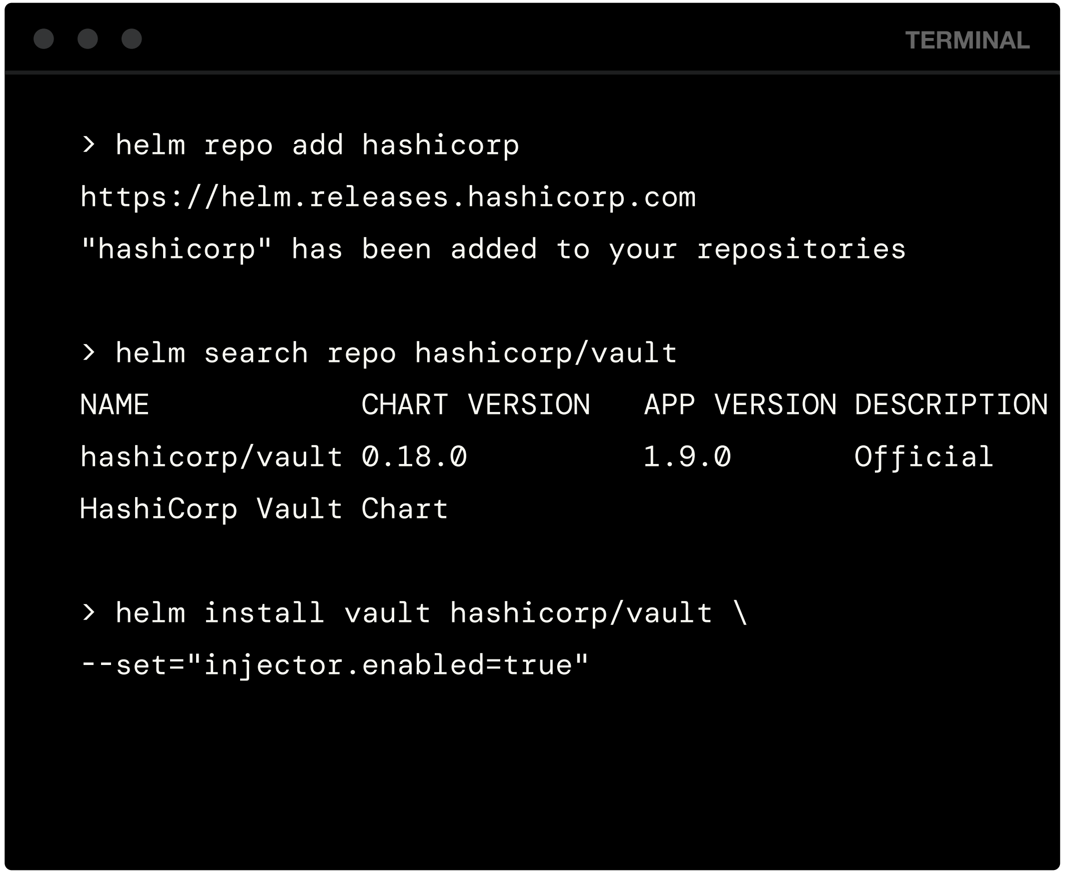


Sidecar Pattern

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



Install Agent Injector


TERMINAL

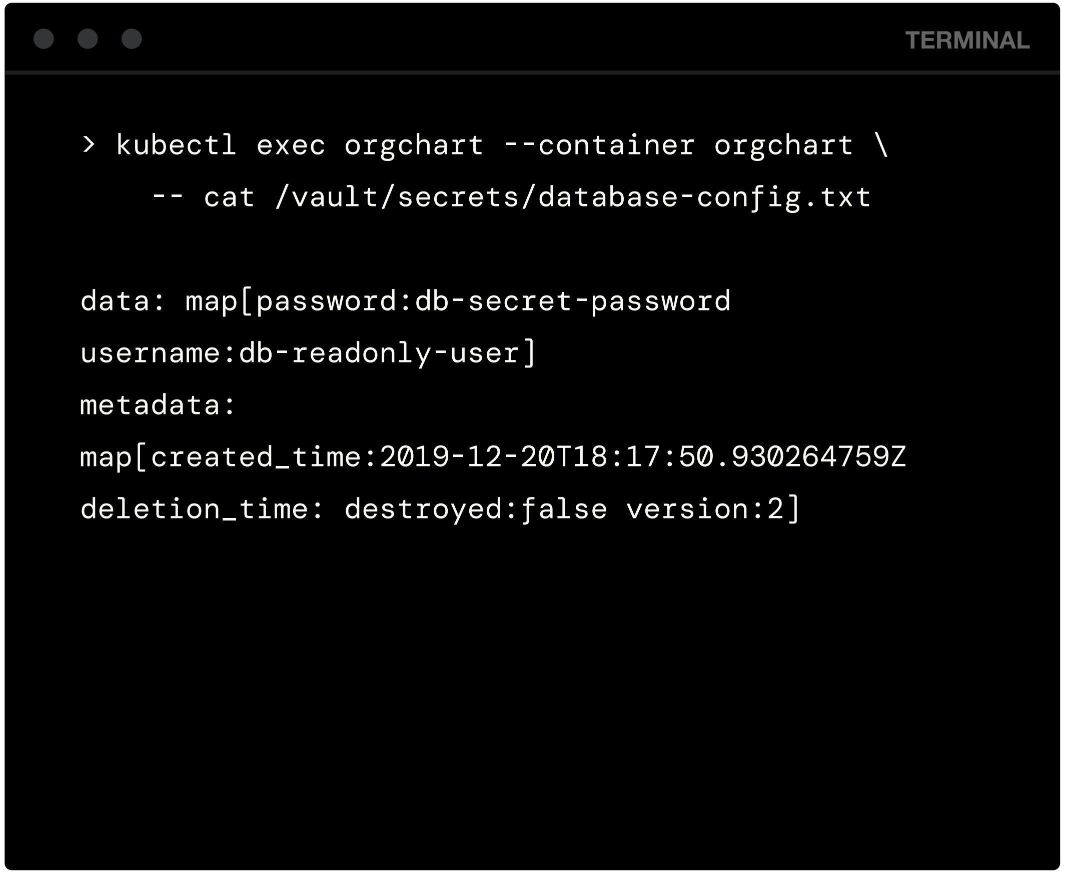
```
> helm repo add hashicorp
https://helm.releases.hashicorp.com
"hashicorp" has been added to your repositories

> 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

CODE EDITOR

```
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


TERMINAL

```
> 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

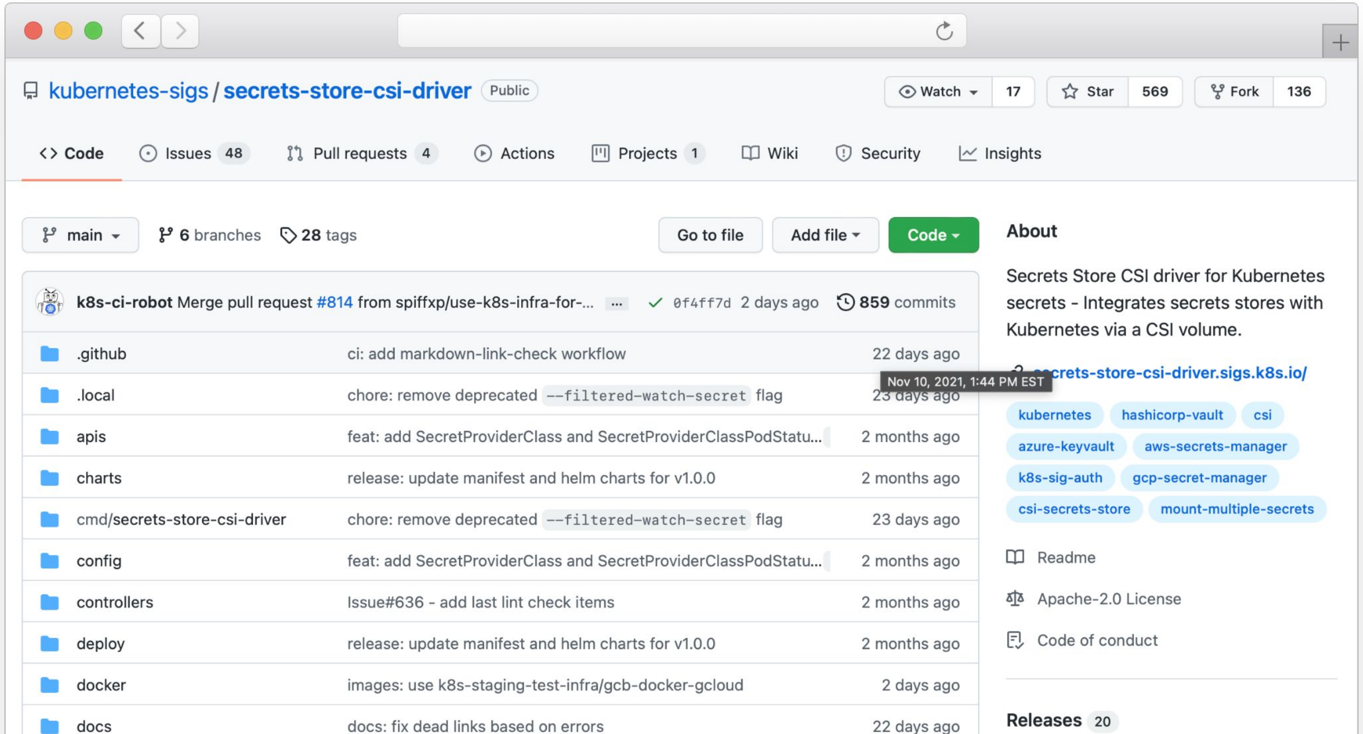
Overview

Secrets Store CSI driver for Kubernetes secrets - Integrates secrets stores with Kubernetes via a Container Storage Interface (CSI) volume.

The Secrets Store CSI driver secrets-store.csi.k8s.io 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 in it is mounted into the container's file system.

Secrets Store CSI Driver

kubernetes-sigs/secrets-store-csi-driver



k8s-ci-robot Merge pull request #814 from spiffxp/use-k8s-infra-for-... ✓ 0f4ff7d 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

Nov 10, 2021, 1:44 PM EST

About

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

secrets-store-csi-driver.sigs.k8s.io/

kubernetes · hashicorp-vault · csi · azure-keyvault · aws-secrets-manager · k8s-sig-auth · gcp-secret-manager · csi-secrets-store · mount-multiple-secrets

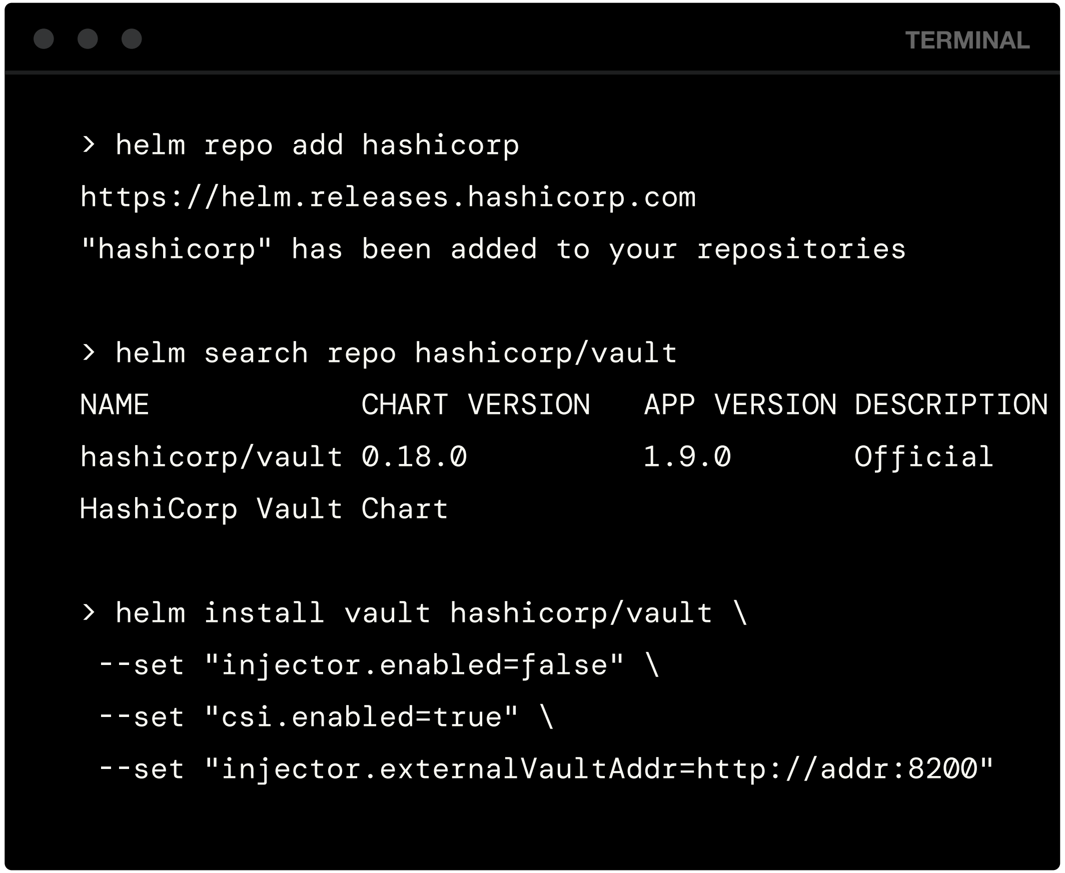
Readme

Apache-2.0 License

Code of conduct

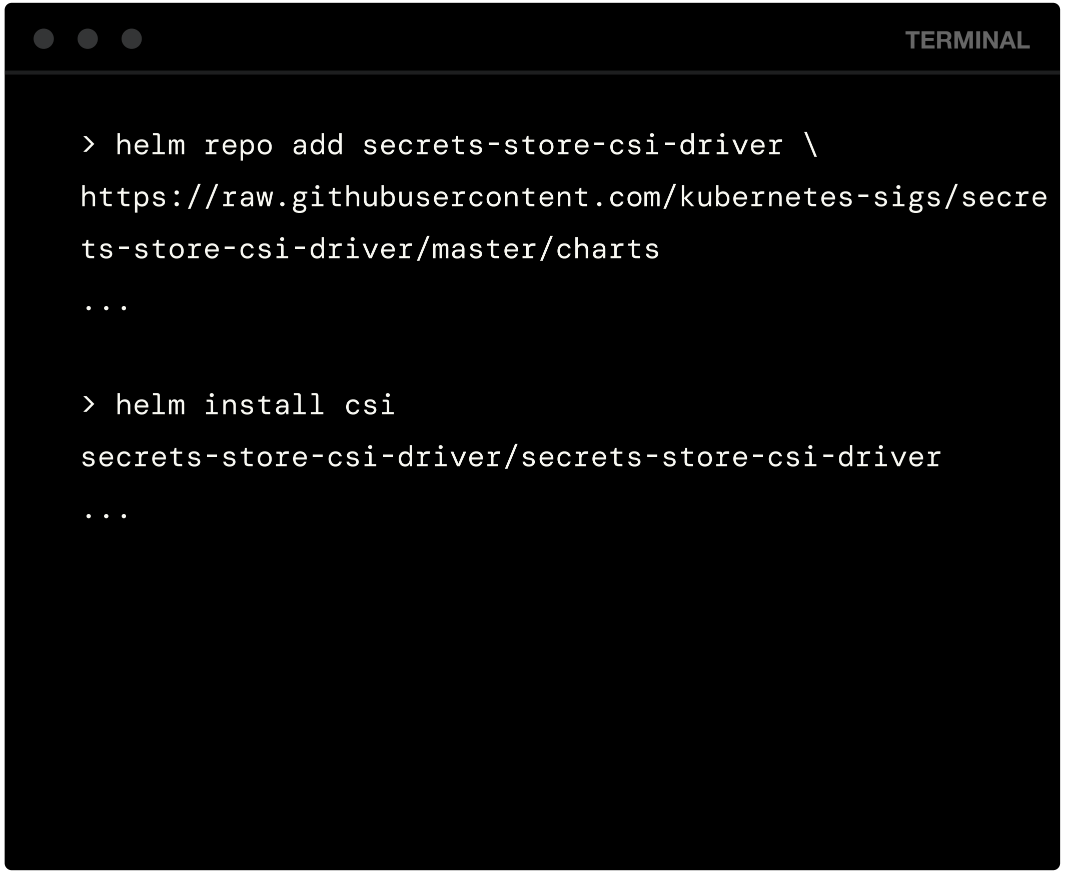
Releases 20

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



```
TERMINAL

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

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

Define SecretProviderClass

CODE EDITOR

```yaml
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"
```

Define a Pod with a Volume

CODE EDITOR

```
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"
```

05

Resources

Resources

- hashicorp/vault-helm
- Vault Enterprise License Management - Kubernetes
- Running Vault - OpenShift
- Examples
- Vault on Kubernetes Reference Architecture | Vault
- Vault on Kubernetes Security Considerations | Vault

Resources

- Injecting Secrets into Kubernetes Pods via Vault Agent Containers | Vault
- Mount Vault Secrets through Container Storage Interface (CSI) Volume | Vault
- Integrate a Kubernetes Cluster with an External Vault | Vault

06

Q & A