HashiCorp
Vault

Vault Kubernetes Integration

Agenda

Helm Chart for Vault 01

Pod Secret Access 02

Vault Agent Injector 03

Container Storage Interface 04

Vault Secrets Operator 05

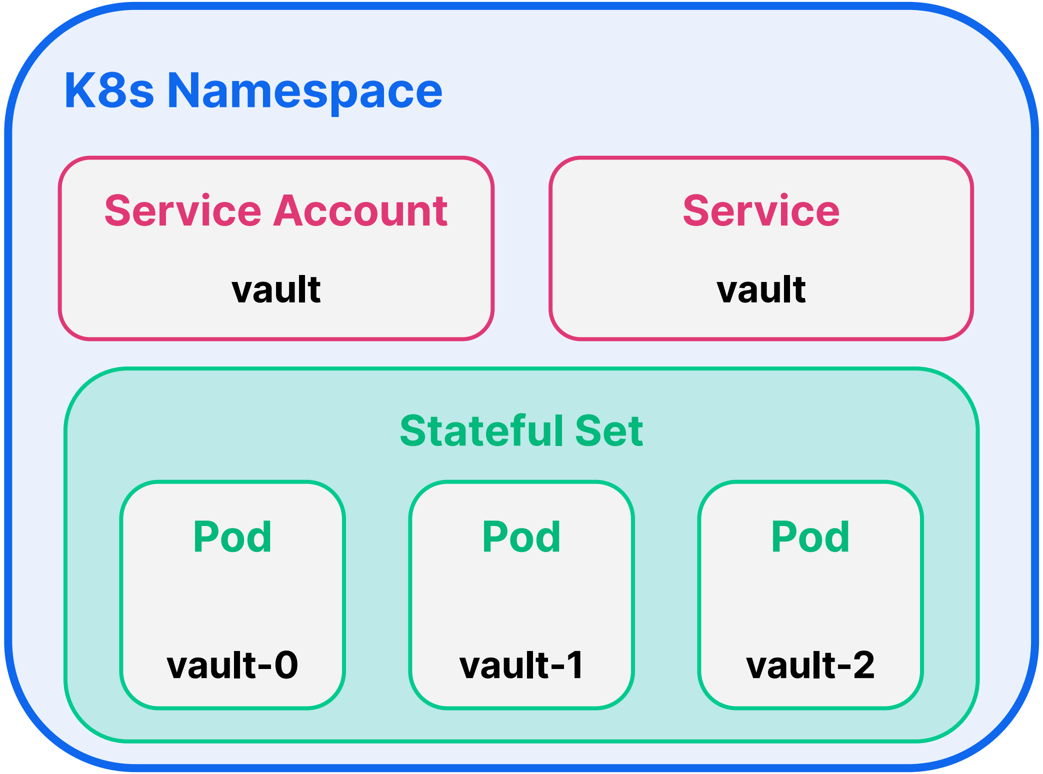
01

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's Helm chart is compatible with Helm 3.6+ and Kubernetes 1.16+

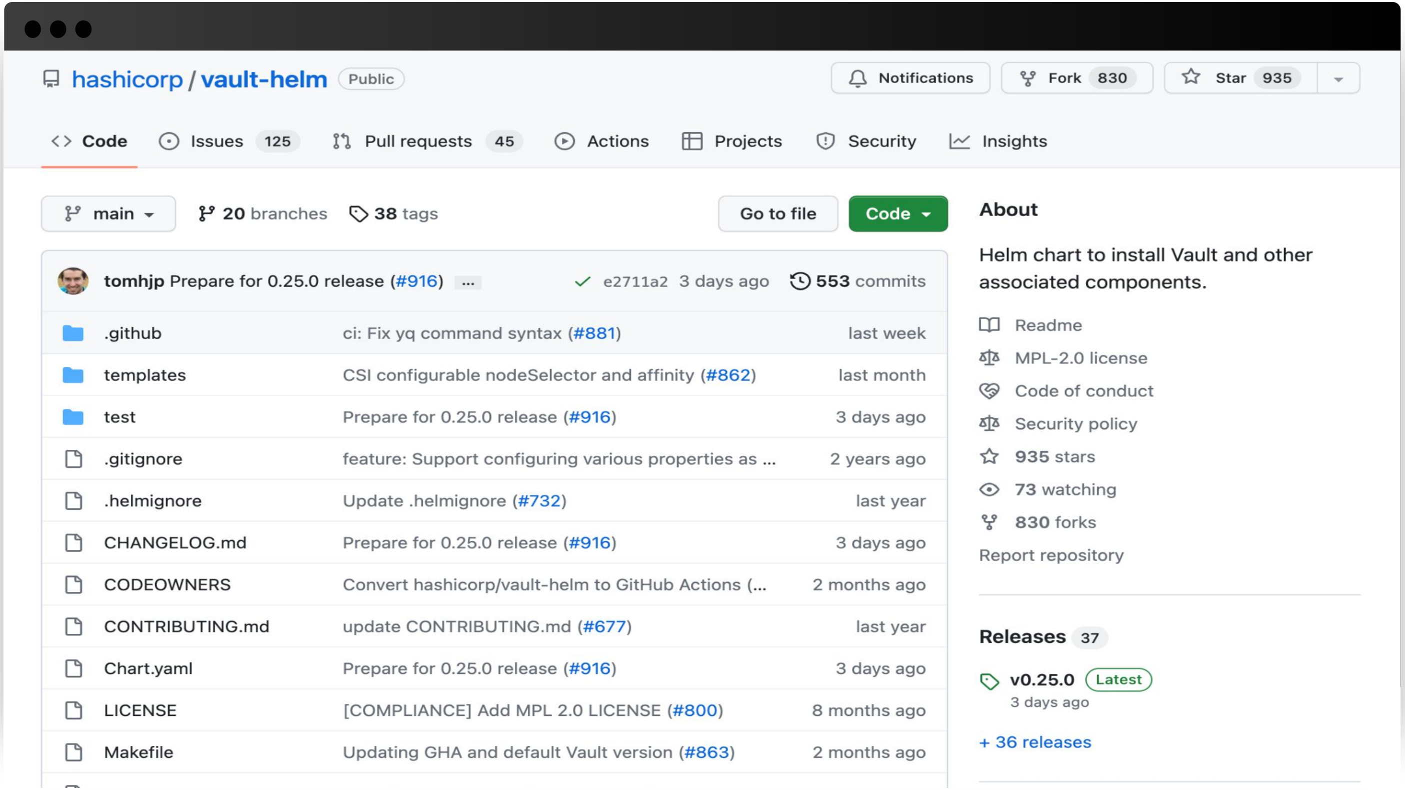
Vault in Kubernetes



- A dedicated Kubernetes cluster should be used for Vault
- Vault **should not** be deployed in the default namespace
- Vault is designed to run as an unprivileged user
- Vault is a stateful application that requires persistent storage

Vault Helm Chart

hashicorp/vault-helm



hashicorp / vault-helm Public

tomhjp Prepare for 0.25.0 release (#916) … e2711a2 3 days ago 553 commits

.github	ci: Fix yq command syntax (#881)	last week
templates	CSI configurable nodeSelector and affinity (#862)	last month
test	Prepare for 0.25.0 release (#916)	3 days ago
.gitignore	feature: Support configuring various properties as ...	2 years ago
.helmignore	Update .helmignore (#732)	last year
CHANGELOG.md	Prepare for 0.25.0 release (#916)	3 days ago
CODEOWNERS	Convert hashicorp/vault-helm to GitHub Actions (...	2 months ago
CONTRIBUTING.md	update CONTRIBUTING.md (#677)	last year
Chart.yaml	Prepare for 0.25.0 release (#916)	3 days ago
LICENSE	[COMPLIANCE] Add MPL 2.0 LICENSE (#800)	8 months ago
Makefile	Updating GHA and default Vault version (#863)	2 months ago

About

Helm chart to install Vault and other associated components.

- Readme
- MPL-2.0 license
- Code of conduct
- Security policy
- 935 stars
- 73 watching
- 830 forks

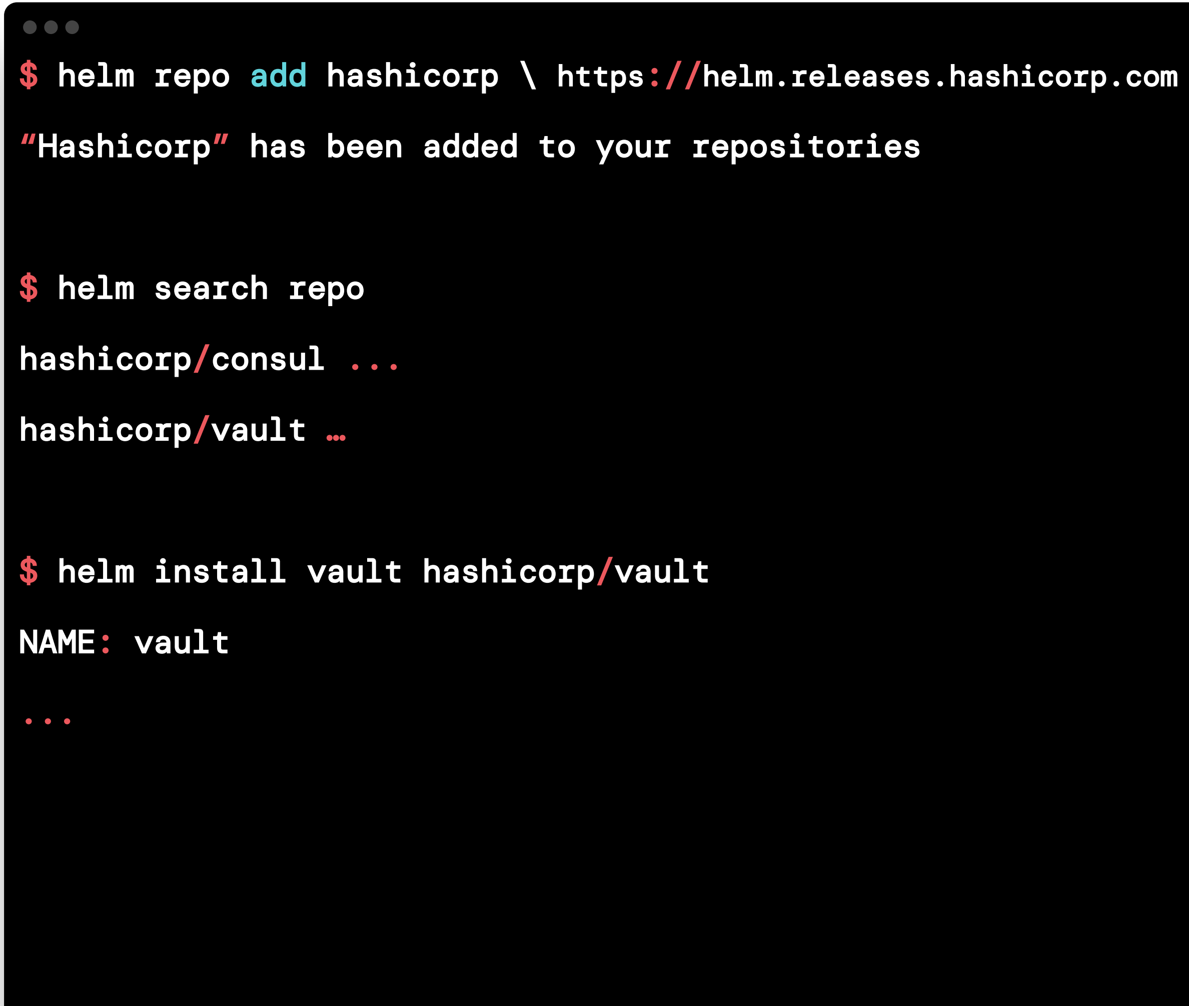
Report repository

Releases 37

v0.25.0 Latest
3 days ago

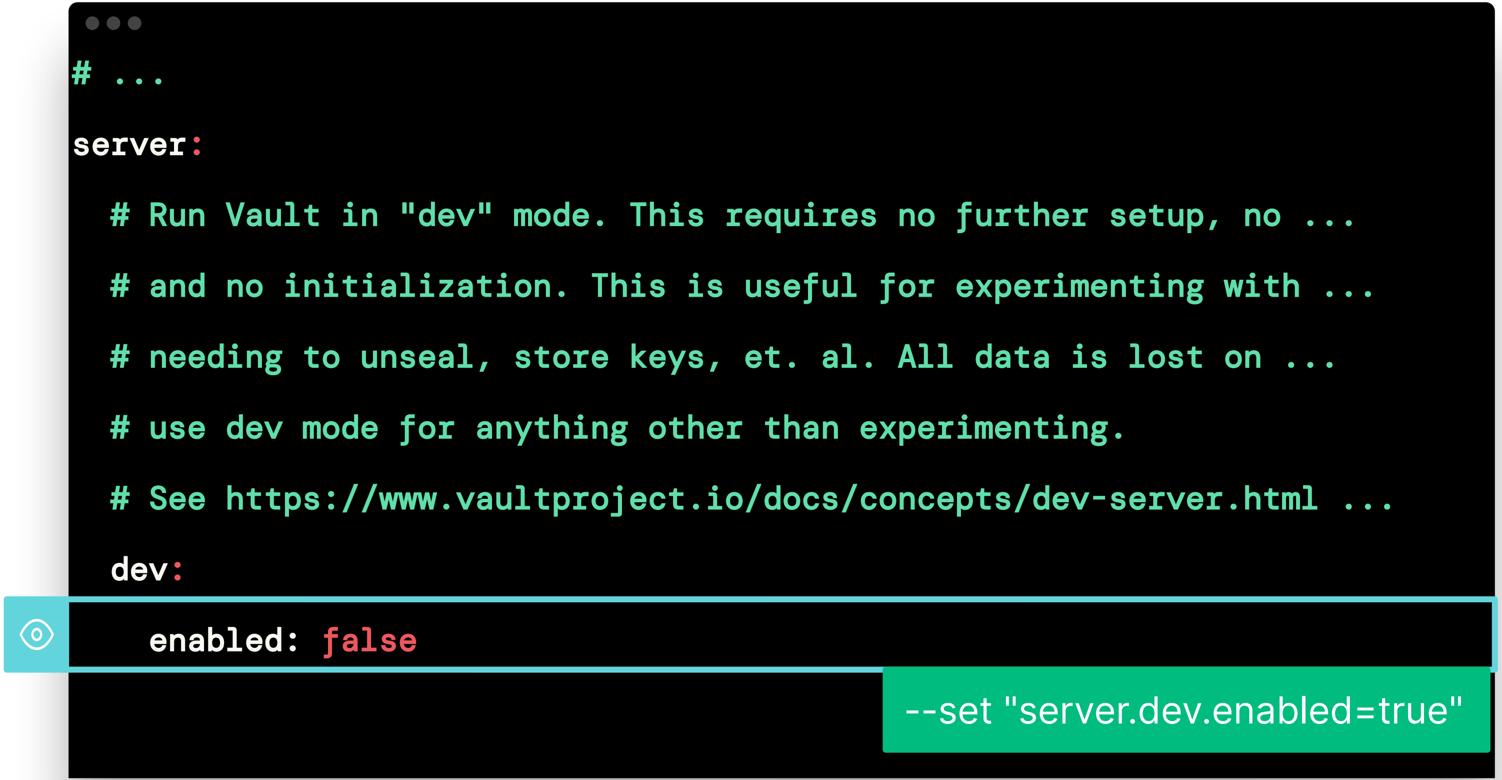
+ 36 releases

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

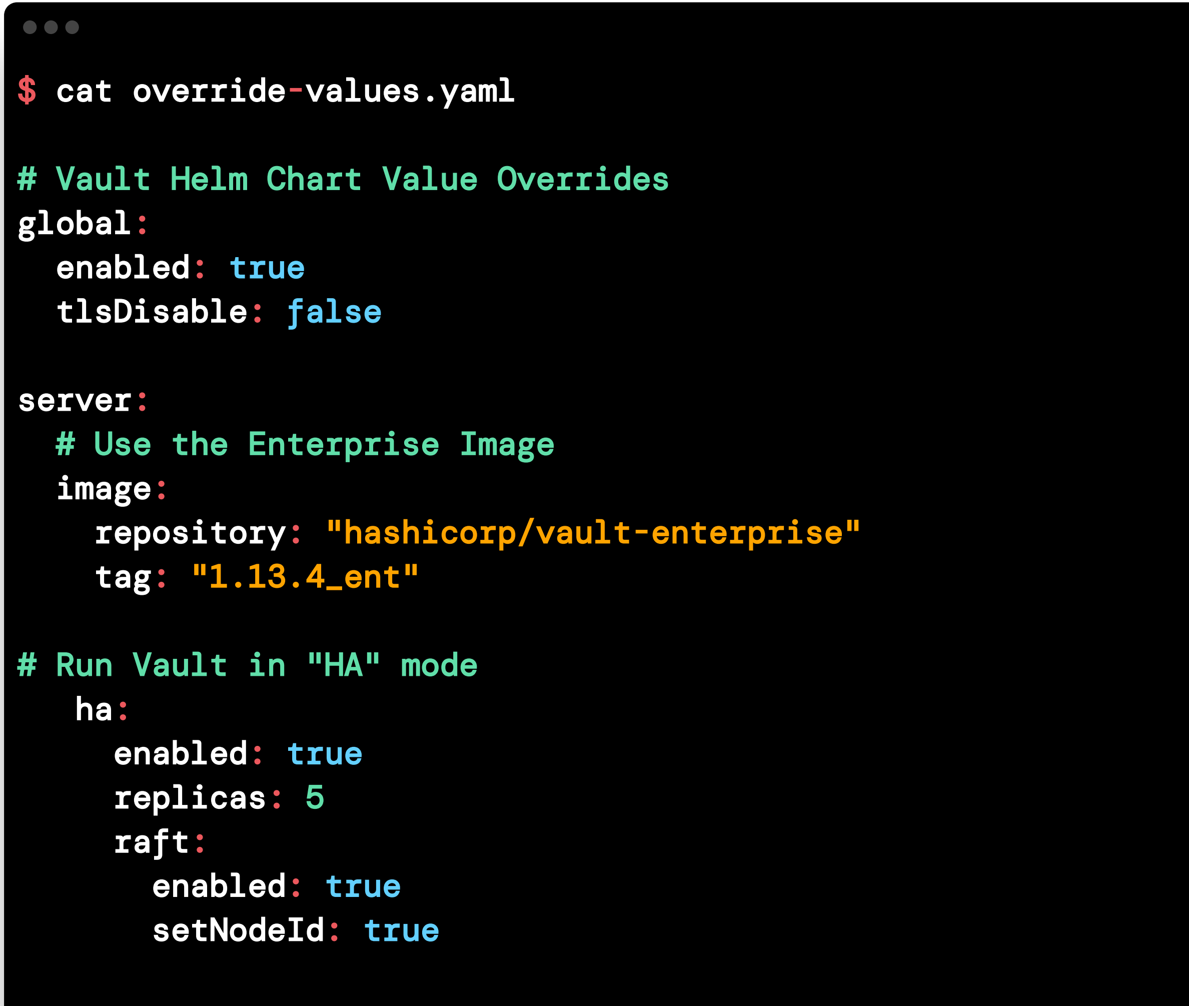


```
# ...
server:
  # Run Vault in "dev" mode. This requires no further setup, no ...
  # and no initialization. This is useful for experimenting with ...
  # needing to unseal, store keys, et. al. All data is lost on ...
  # use dev mode for anything other than experimenting.
  # See https://www.vaultproject.io/docs/concepts/dev-server.html ...
  dev:
    enabled: false
```

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

Create an Override File

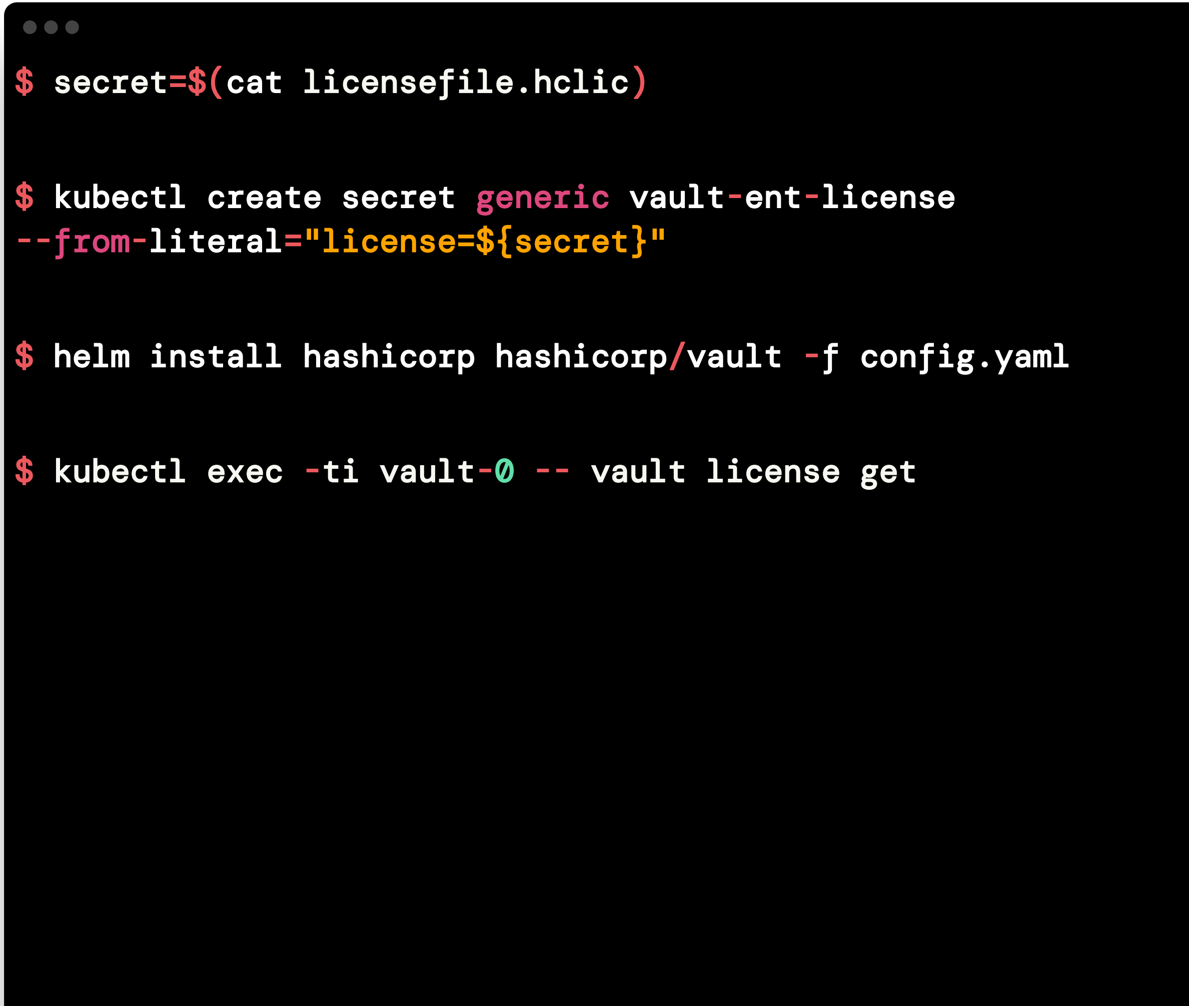
Configure Vault Helm Chart



```
$ cat override-values.yaml

# Vault Helm Chart Value Overrides
global:
  enabled: true
  tlsDisable: false

server:
  # Use the Enterprise Image
  image:
    repository: "hashicorp/vault-enterprise"
    tag: "1.13.4_ent"

# Run Vault in "HA" mode
  ha:
   enabled: true
   replicas: 5
   raft:
    enabled: true
    setNodeId: true
```

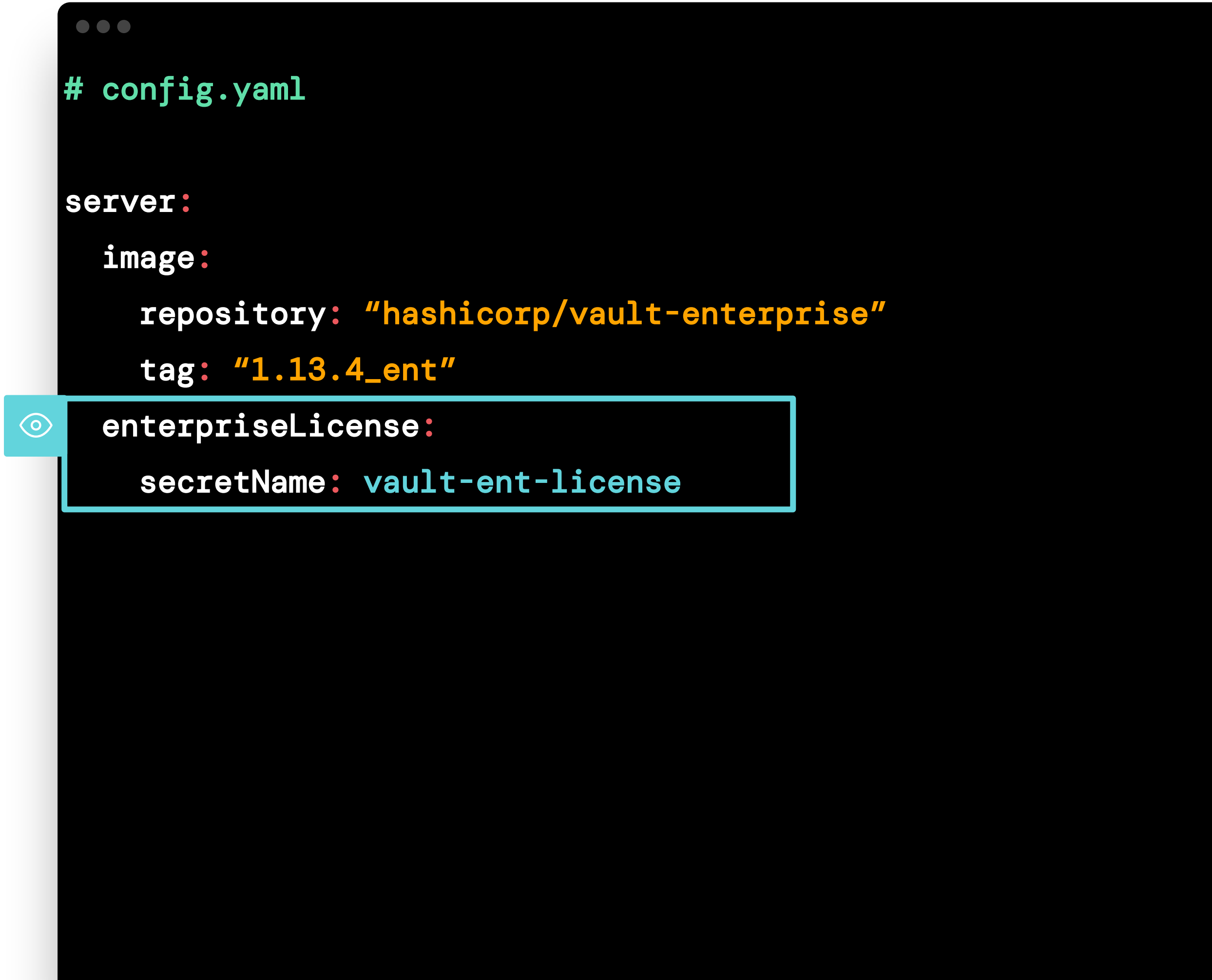
© HASHICORP

Licensing



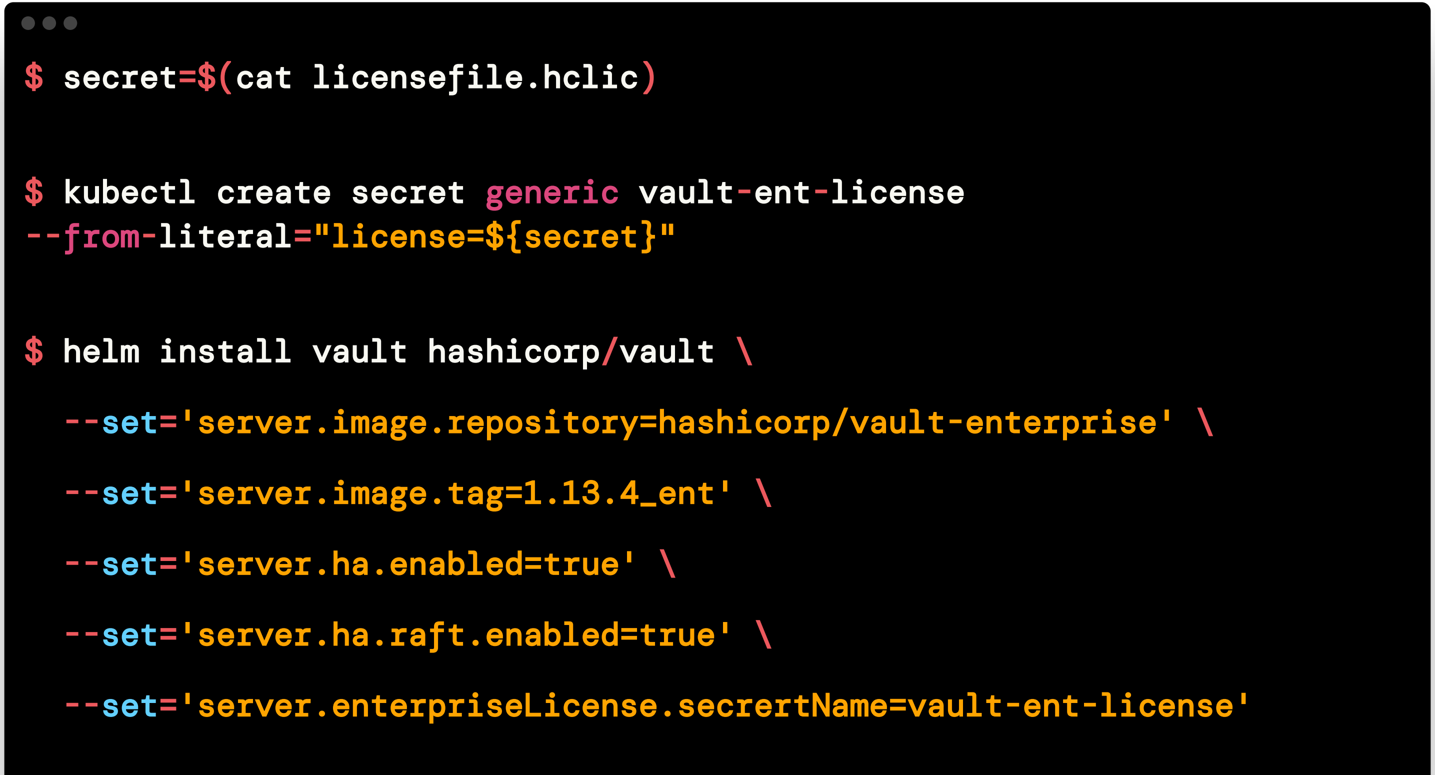
```
$ secret=$(cat licensefile.hclic)


$ 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



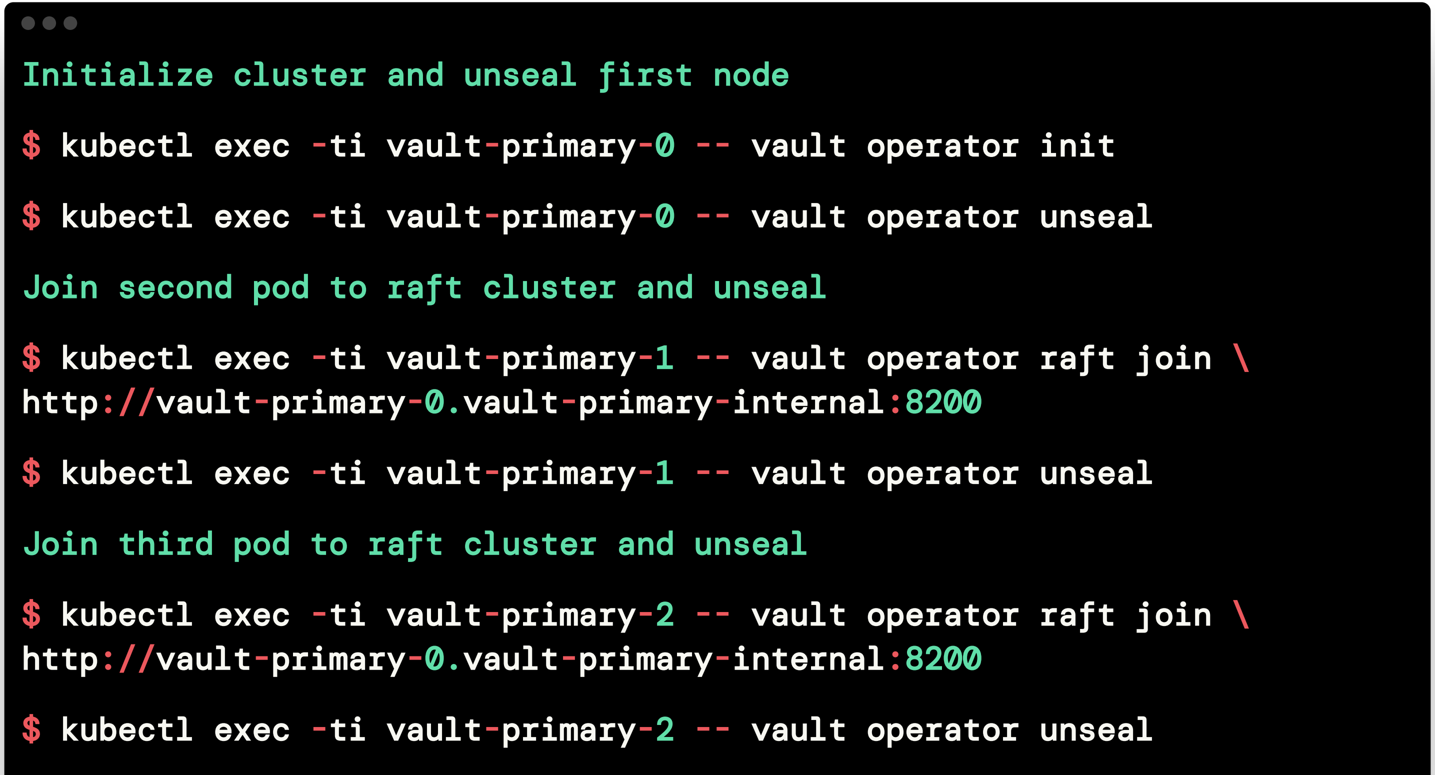
```
# config.yaml

server:
  image:
    repository: “hashicorp/vault-enterprise”
    tag: “1.13.4_ent”
  enterpriseLicense:
    secretName: vault-ent-license
```

Primary HA Vault ENT Cluster Deployment



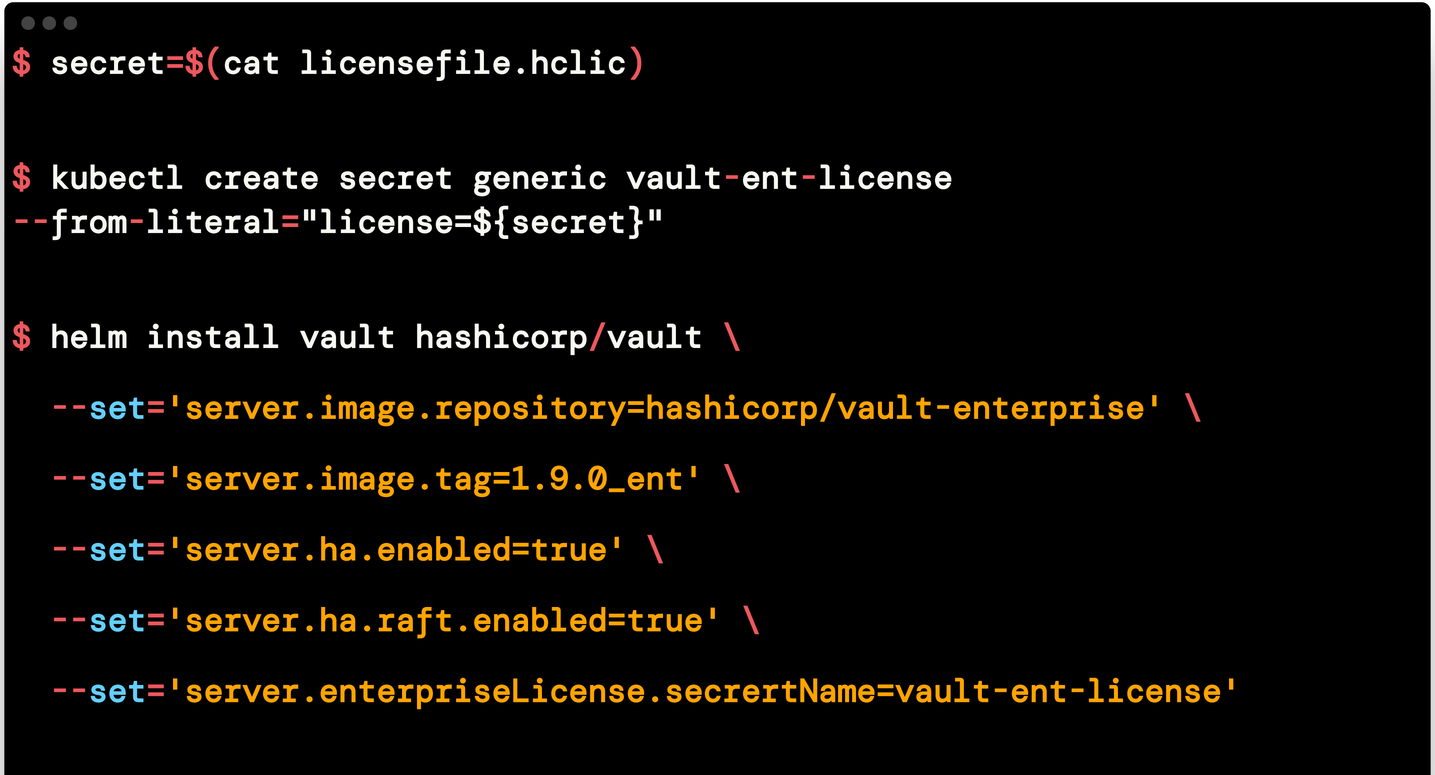
```
$ secret=$(cat licensefile.hclic)


$ 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=1.13.4_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



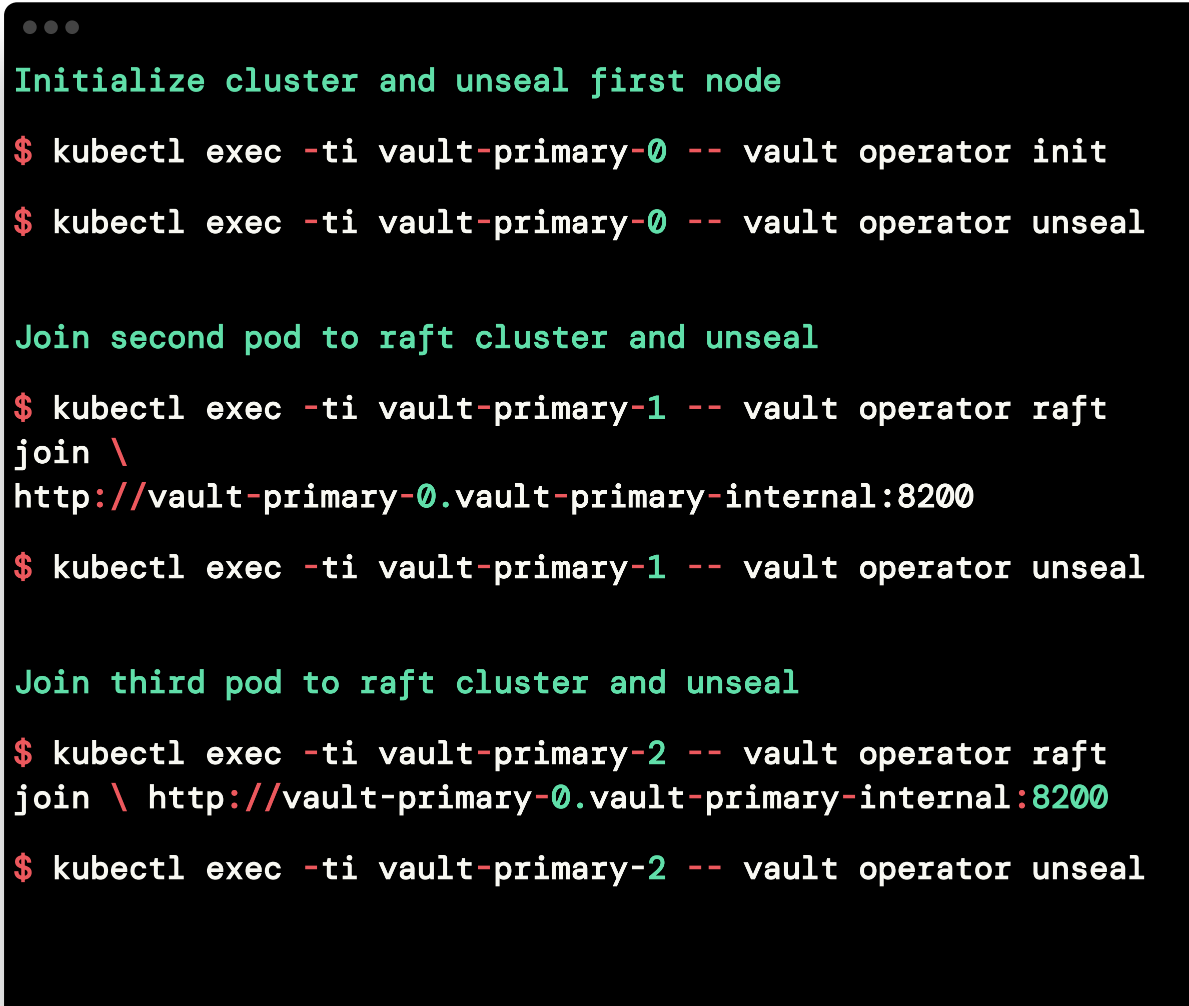
```
Initialize cluster and unseal first node

$ 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



```
$ 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'
```

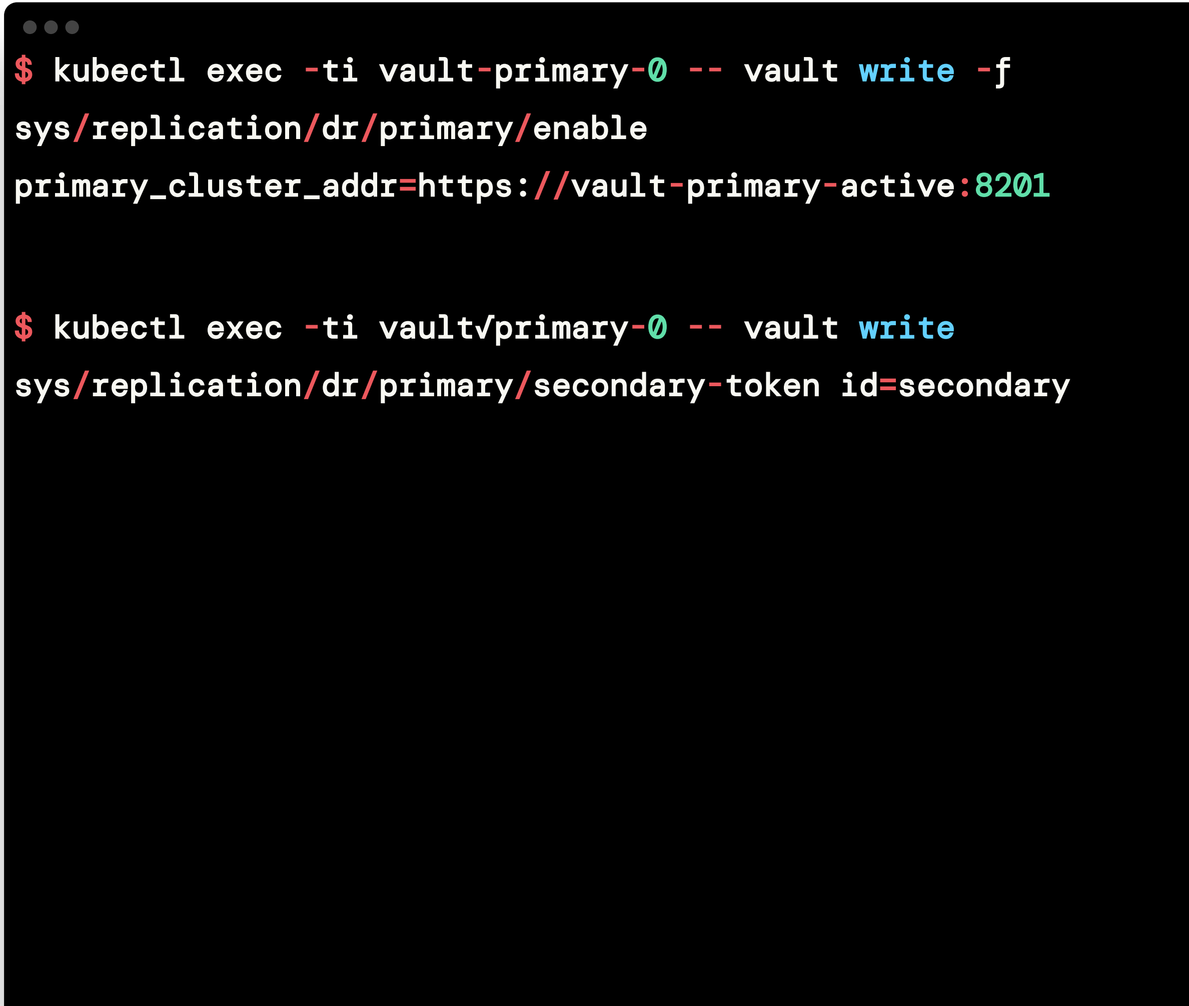
DR HA Vault ENT Cluster Deployment



```
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

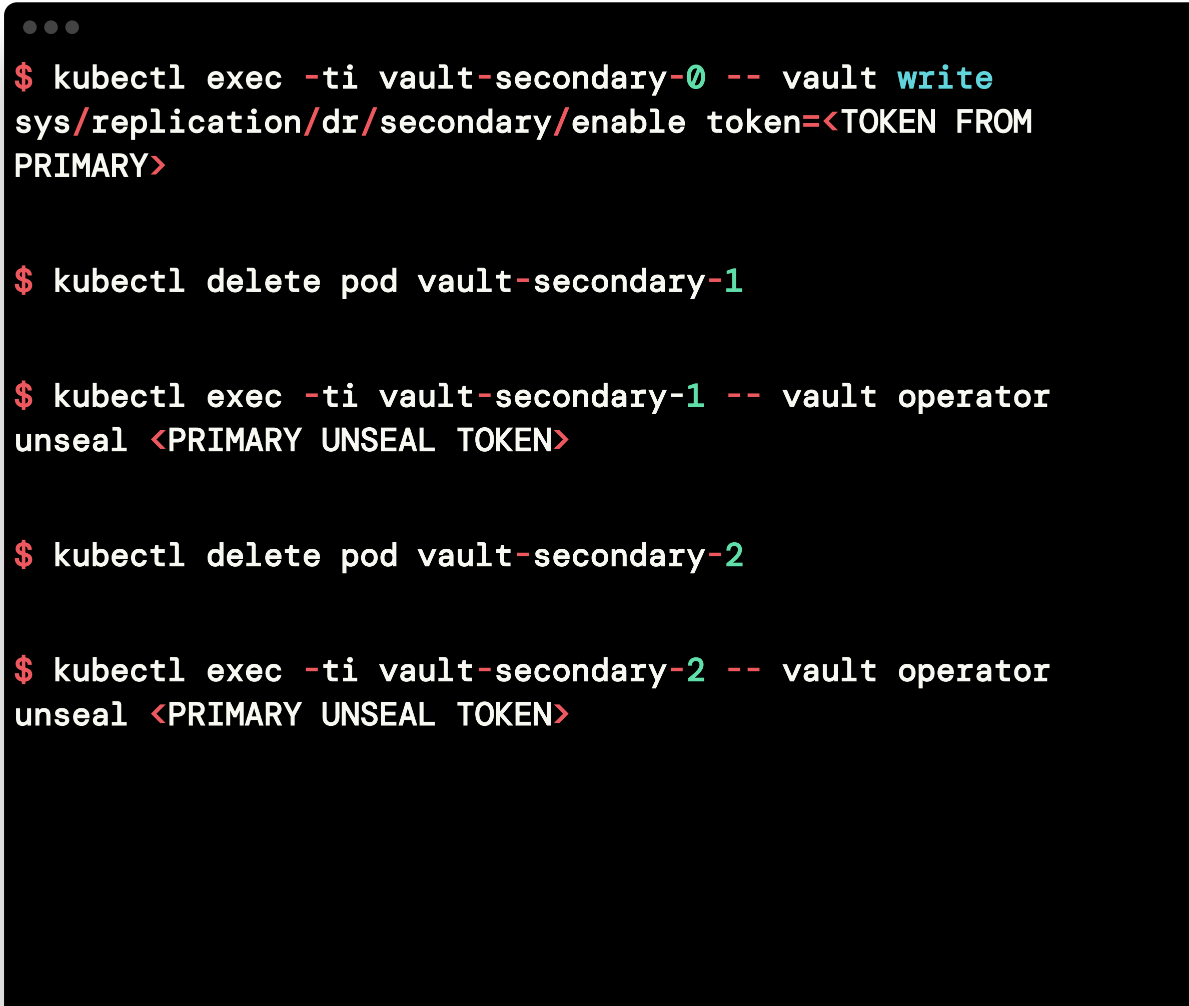
Primary Cluster



```
$ 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



```
$ 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>
```

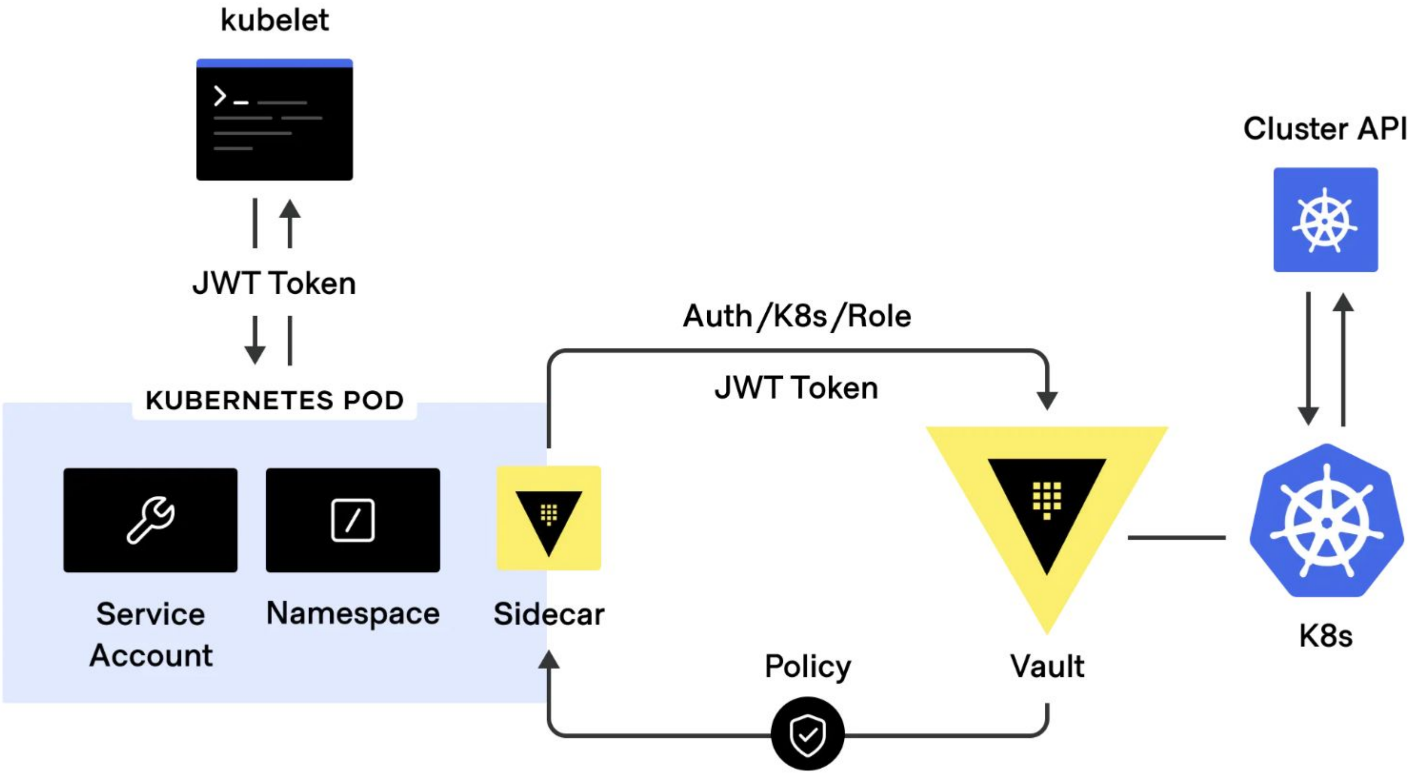
Upgrading Vault

- Always backup Vault data via snapshot before beginning any upgrade
- Follow general [Vault upgrade pattern]() and use the Helm chart to update Vault server StatefulSet
- Vault StatefulSet uses ***`OnDelete`*** (instead of ***`RollingUpdate`***) to ensure standby nodes are updated before the active primary node
- Helm will install the latest chart found in a repo by default, best practice is to specify the chart version when upgrading

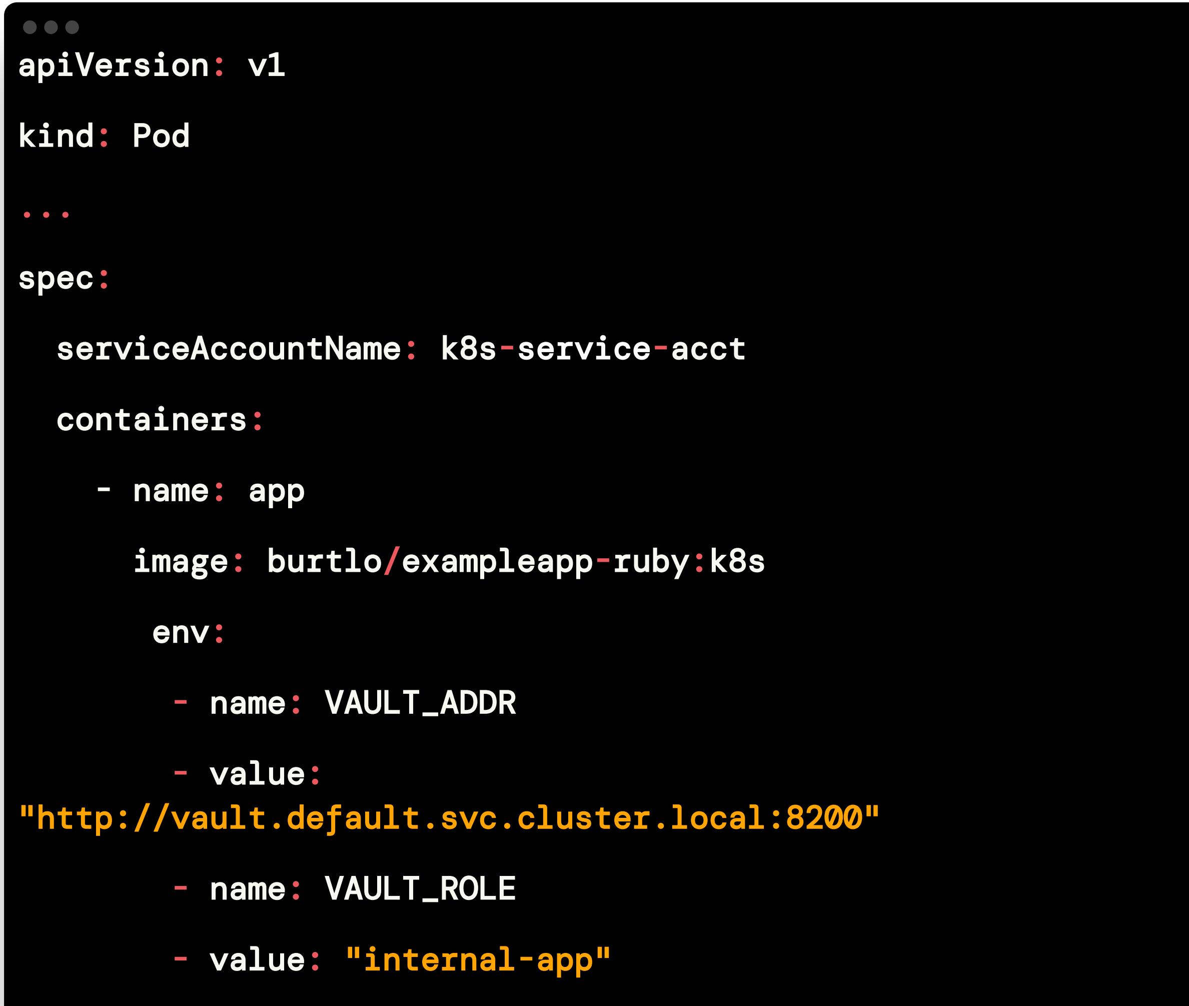
02

Pod Secret Access

Kubernetes Auth Flow

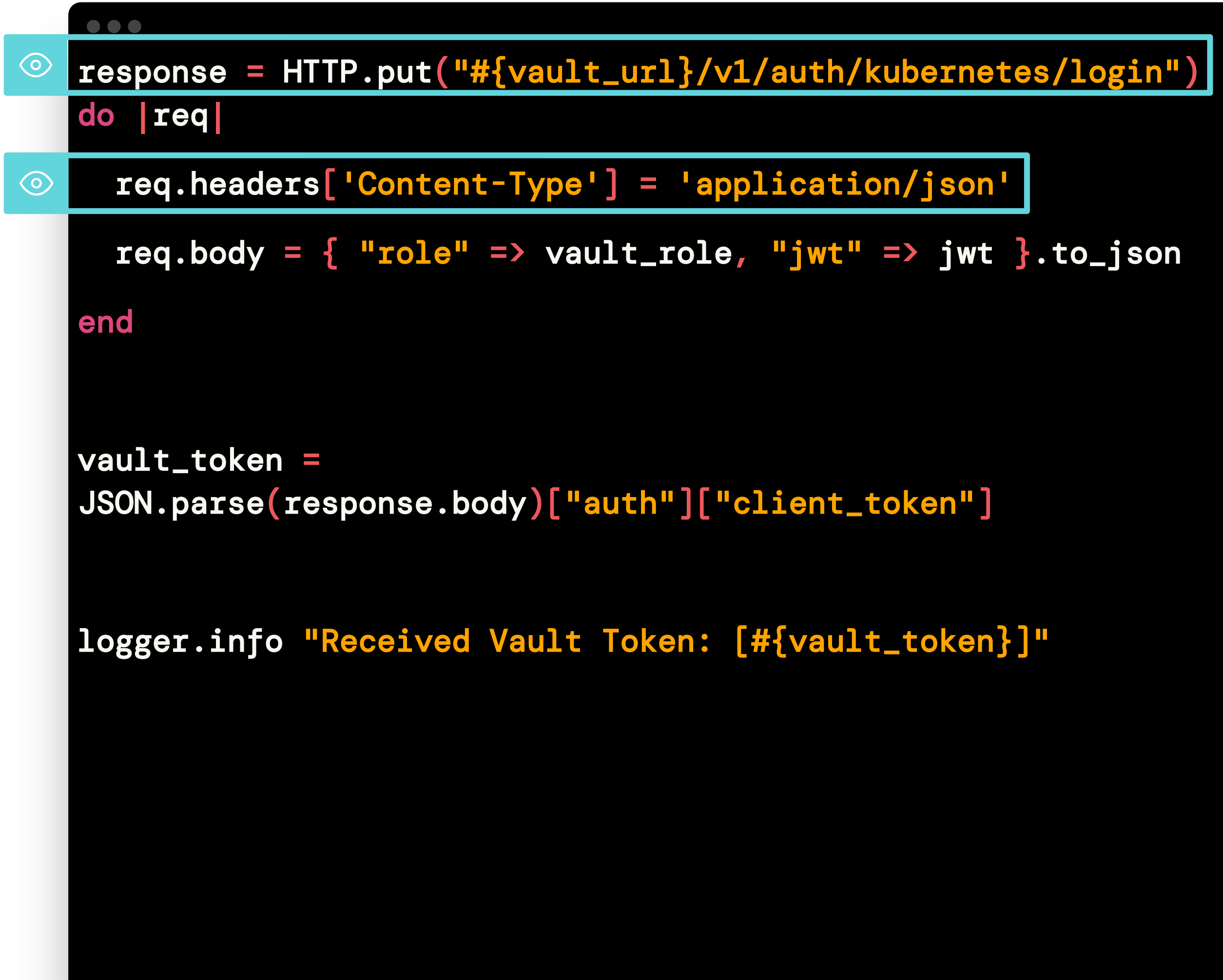


Application Pod Definition



```
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

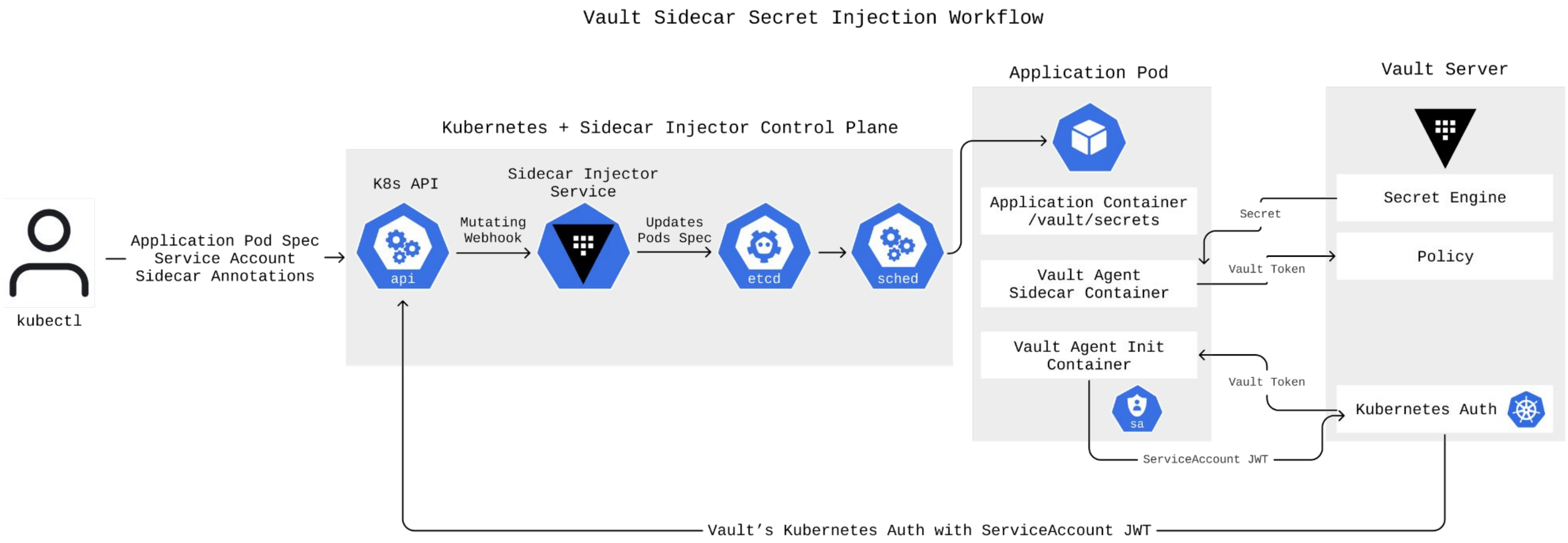


```
response = HTTP.put("#{vault_url}/v1/auth/kubernetes/login")
do |req|
  req.headers['Content-Type'] = 'application/json'
  req.body = { "role" => vault_role, "jwt" => jwt }.to_json
end


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


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

03

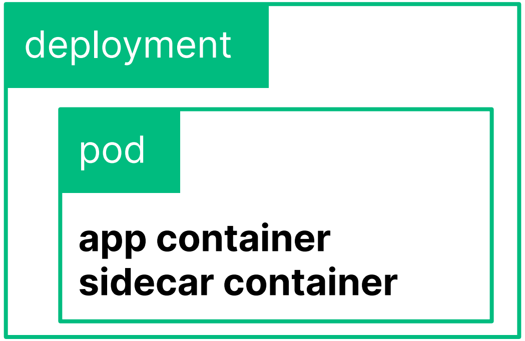
Vault Agent Injector

Sidecar Pattern Workflow



Sidecar Pattern

- Vault Agent injector uses the Kubernetes Sidecar pattern to append a Vault Agent container to pods
- Vault Agent renders Vault secrets to a shared memory volume
- Agent injector is a Kubernetes mutating webhook controller
- Sidecar container needs:
  - Vault address
  - Vault authentication role
  - Vault secret path



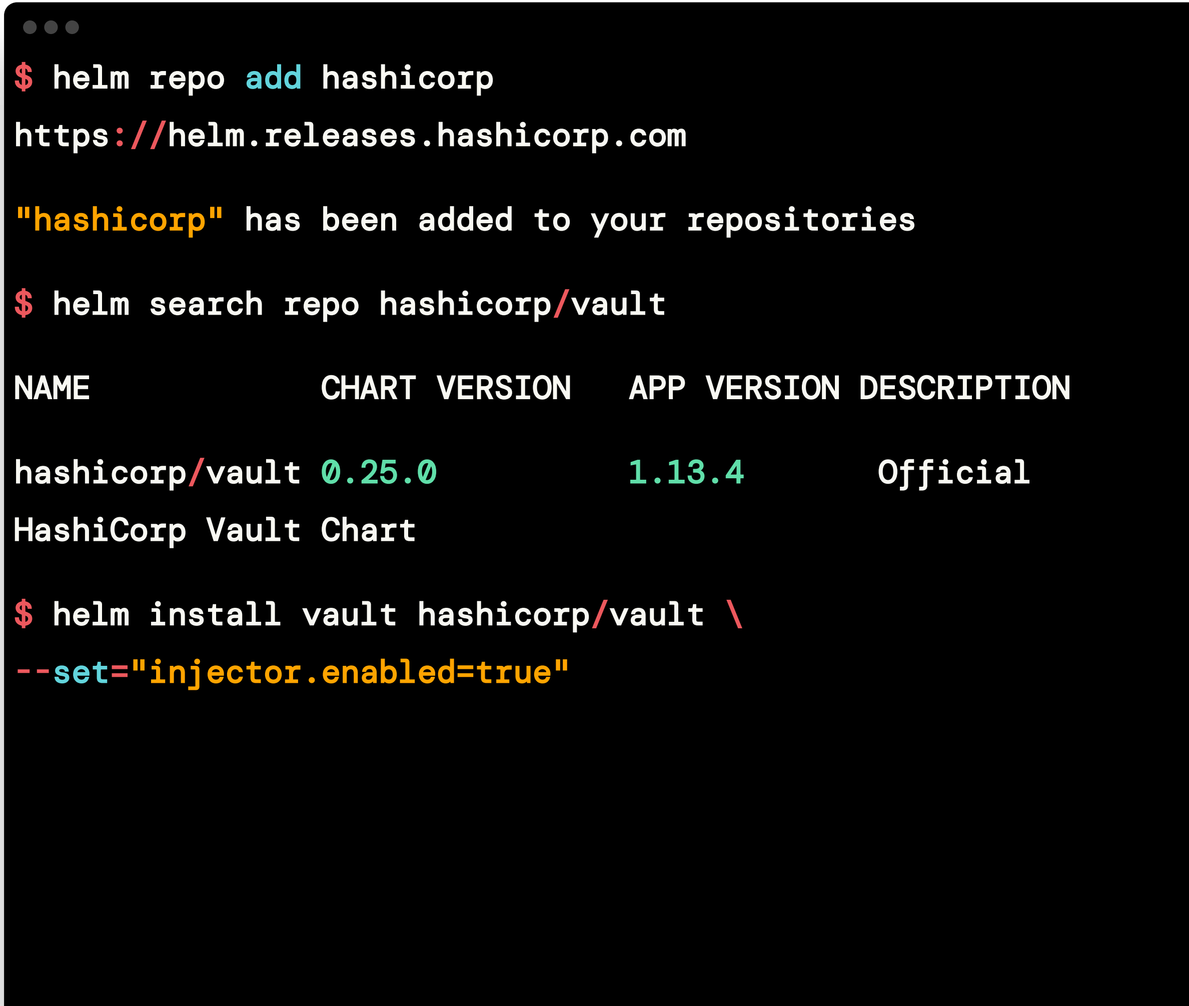
deployment

pod

app container
sidecar container

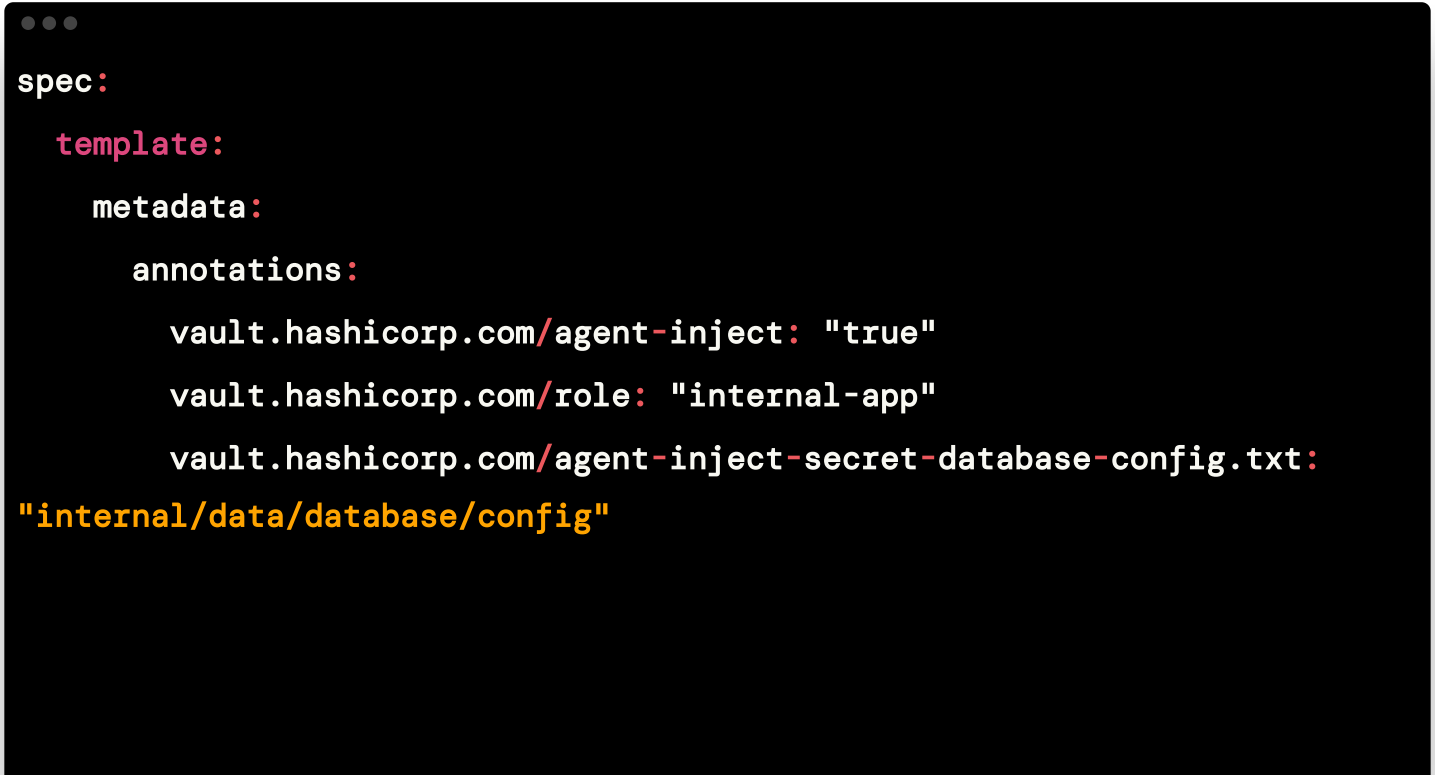
Install Agent Injector

Installation Guide



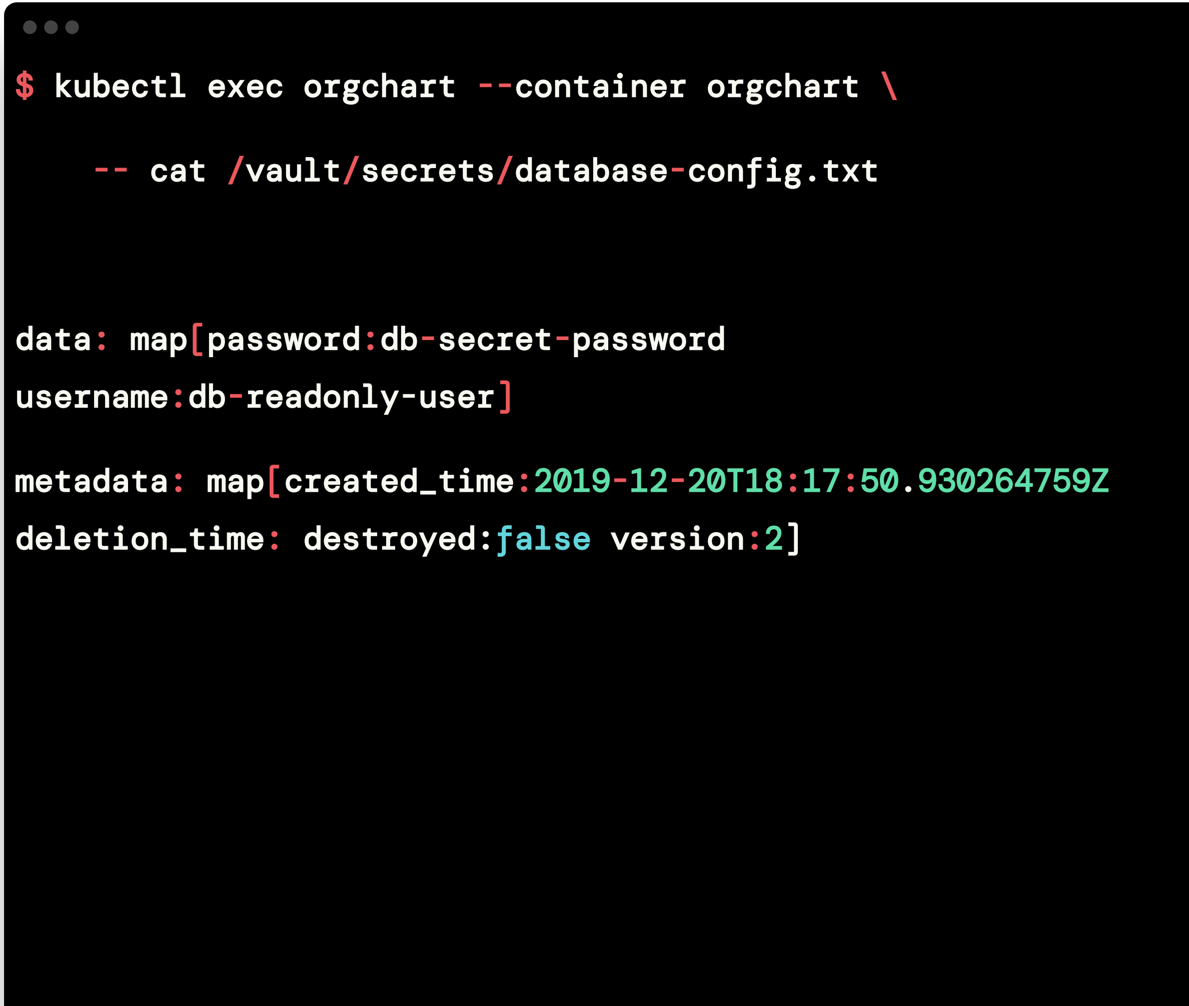
```
$ helm repo add hashicorp
https://helm.releases.hashicorp.com

"hashicorp" has been added to your repositories

$ helm search repo hashicorp/vault

NAME                CHART VERSION   APP VERSION DESCRIPTION

hashicorp/vault 0.25.0          1.13.4      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

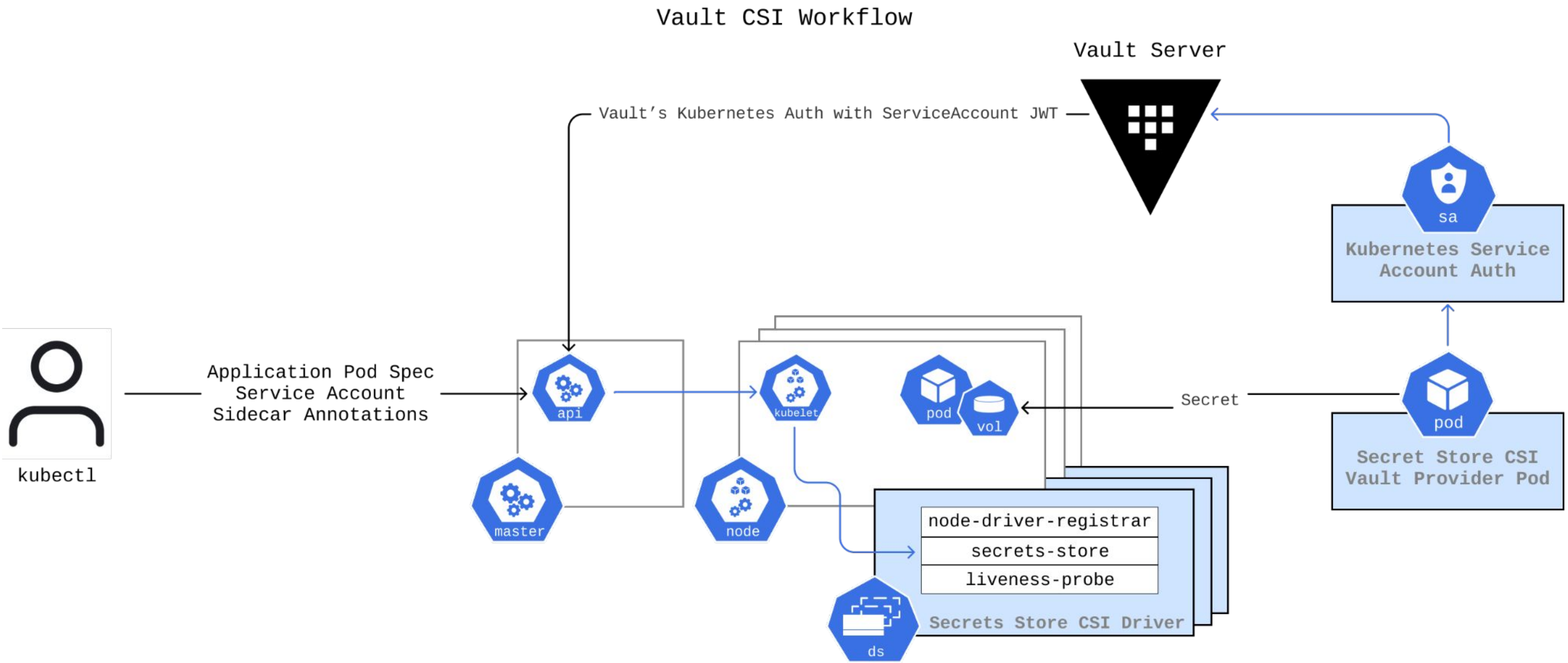


```
$ kubectl exec orgchart --container orgchart \

    -- 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

CSI Driver Workflow

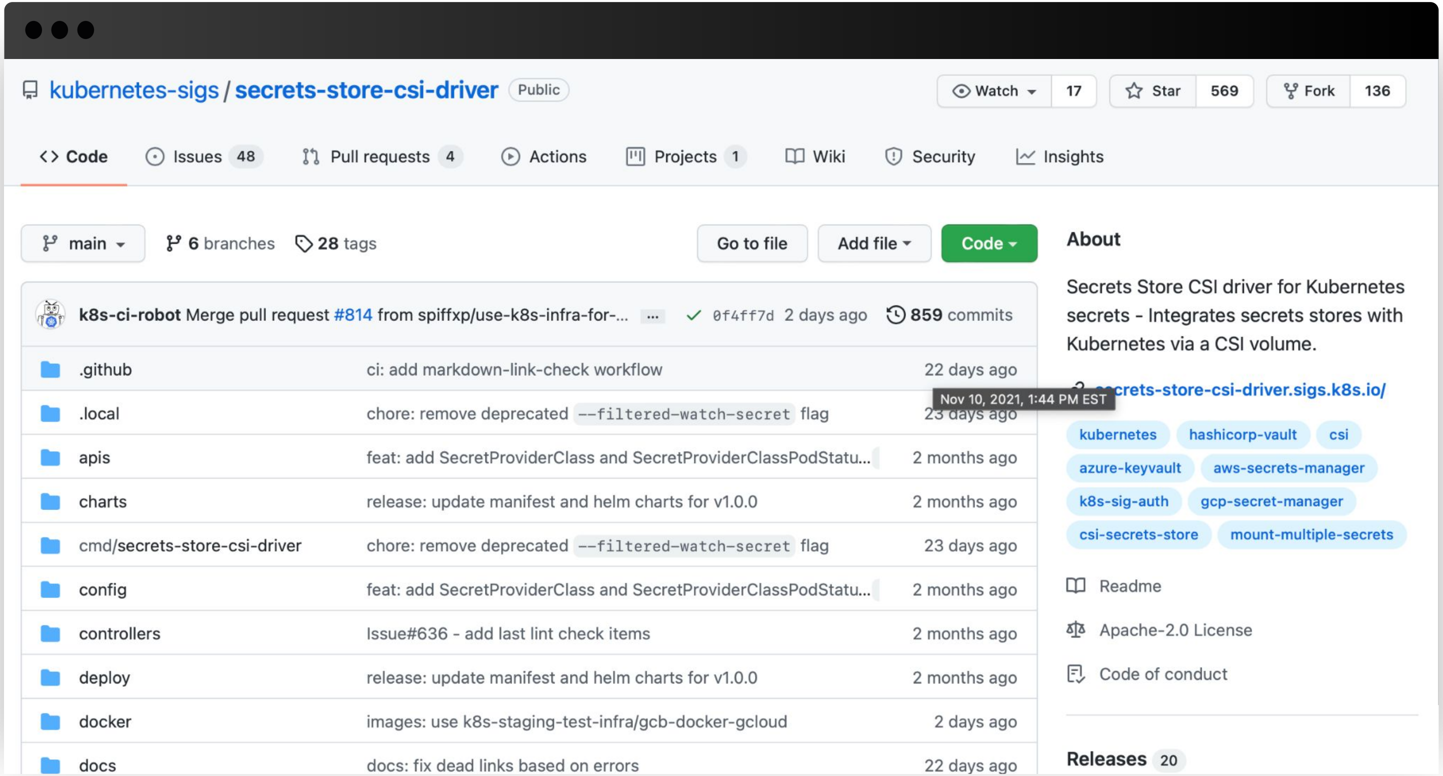


Vault CSI Driver

- Integrates secrets stores with Kubernetes via a Container Storage Interface (CSI) volume
- Is deployed as a daemonset on every node in the Kubernetes cluster
- Uses the Secret Provider Class specified and the pod’s service account to retrieve secrets from Vault, and mount them into the pod’s CSI volume
- Uses ***`hostPath`*** to mount ephemeral volumes into the pods (disabled by default in OpenShift)

Secrets Store CSI Driver

CSI Driver



kubernetes-sigs / secrets-store-csi-driver Public

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

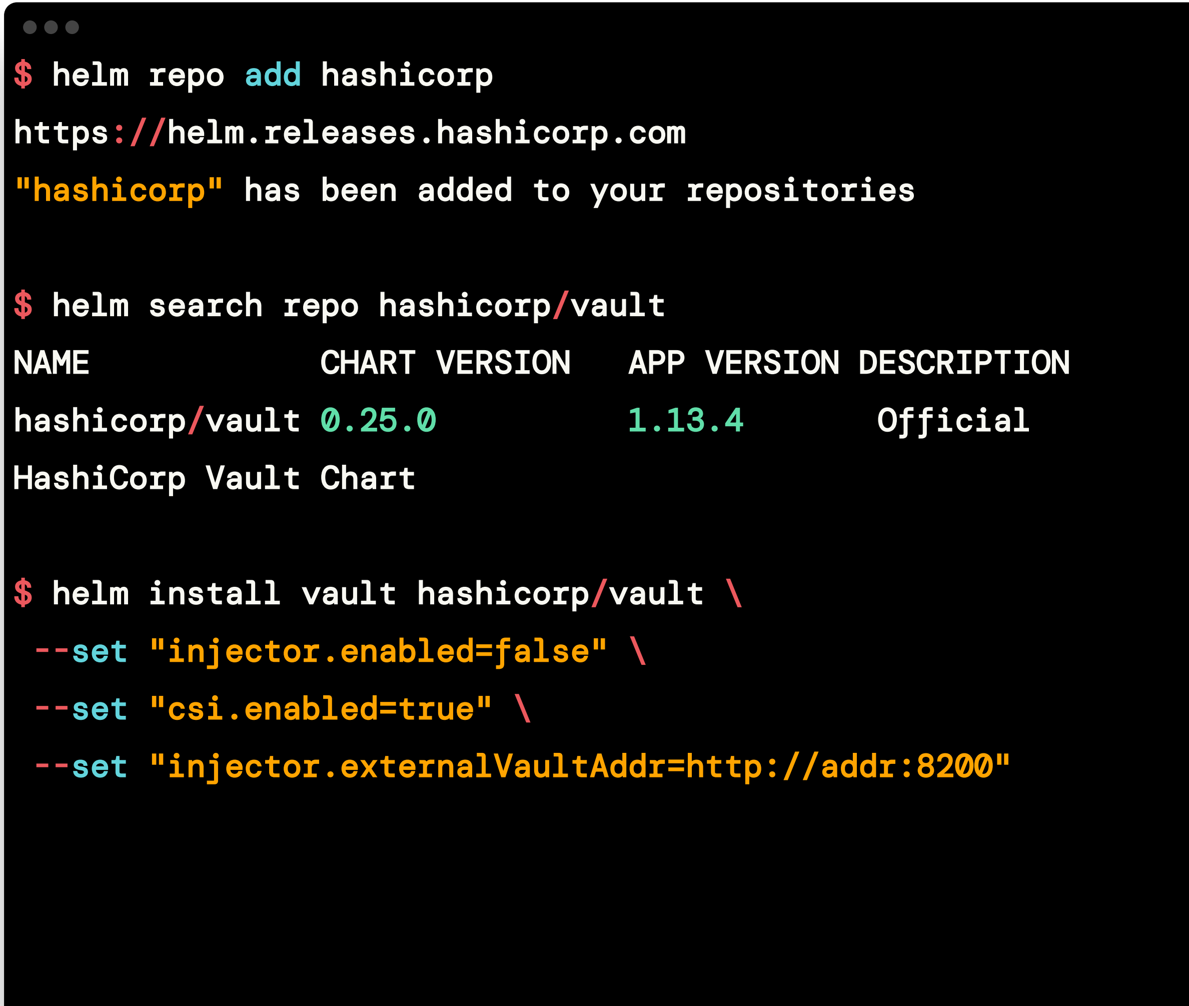
About

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

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

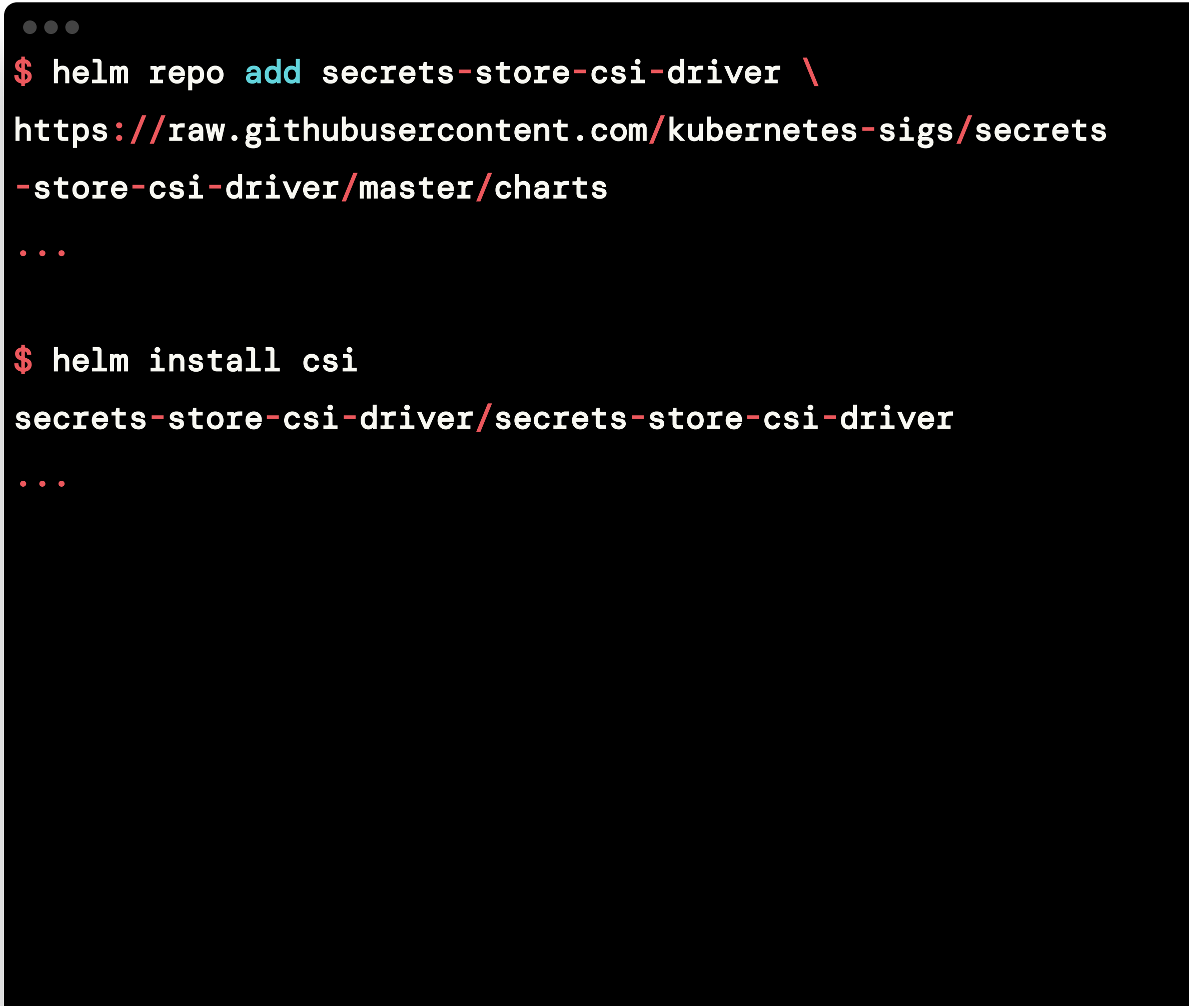
Install Container Storage Interface

Installation Guide



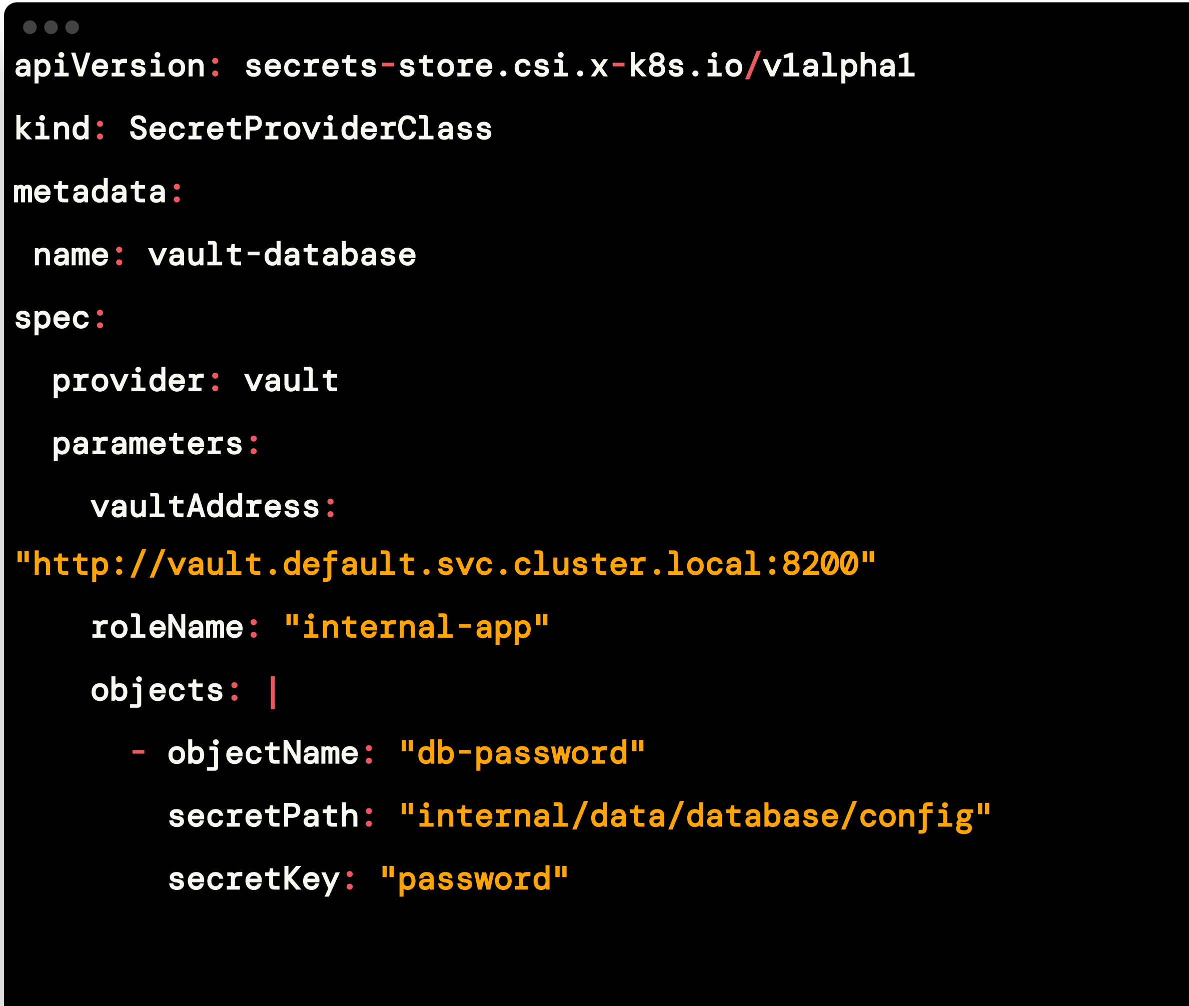
```
$ helm repo add hashicorp
https://helm.releases.hashicorp.com
"hashicorp" has been added to your repositories

$ helm search repo hashicorp/vault
NAME            CHART VERSION   APP VERSION DESCRIPTION
hashicorp/vault 0.25.0          1.13.4       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



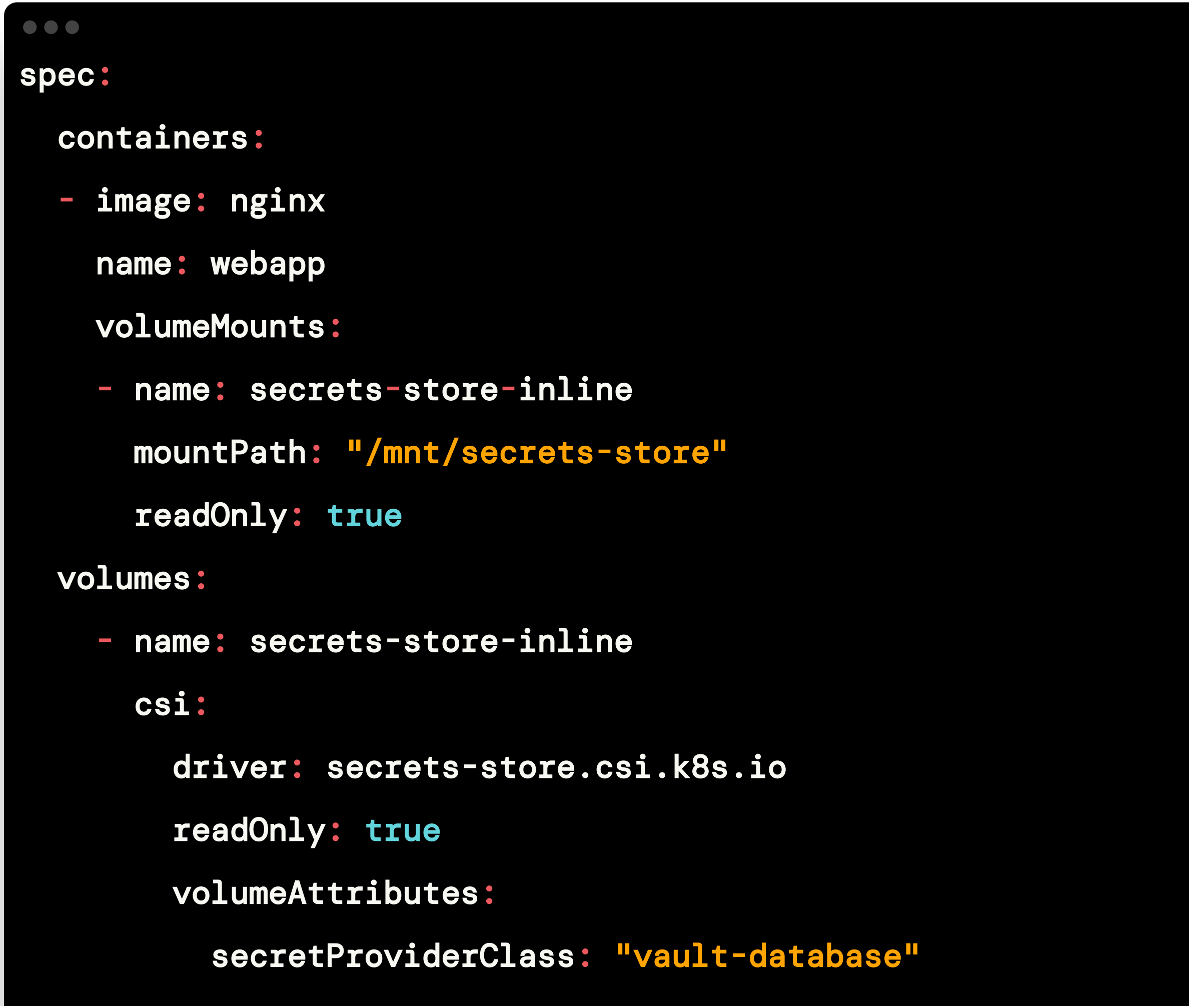
```
$ helm repo add secrets-store-csi-driver \
https://raw.githubusercontent.com/kubernetes-sigs/secrets
-store-csi-driver/master/charts
...

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

Install Secrets Store CSI Driver



```
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



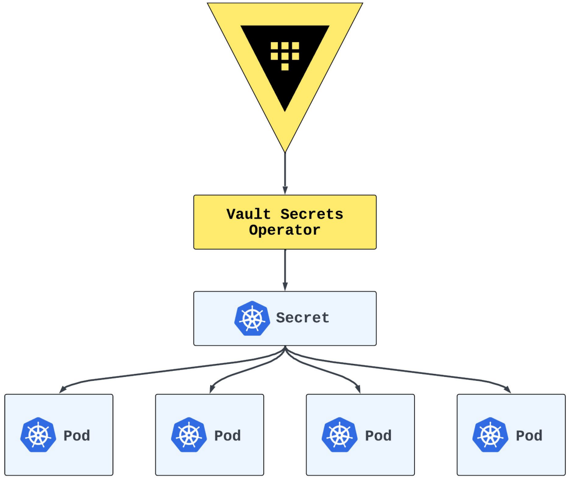
```
spec:
 containers:
 - image: nginx
  name: webapp
  volumeMounts:
  - name: secrets-store-inline
   mountPath: "/mnt/secrets-store"
   readOnly: true
 volumes:
  - name: secrets-store-inline
   csi:
    driver: secrets-store.csi.k8s.io
    readOnly: true
    volumeAttributes:
     secretProviderClass: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

05

Vault Secrets Operator

VSO Overview

Installation Guide

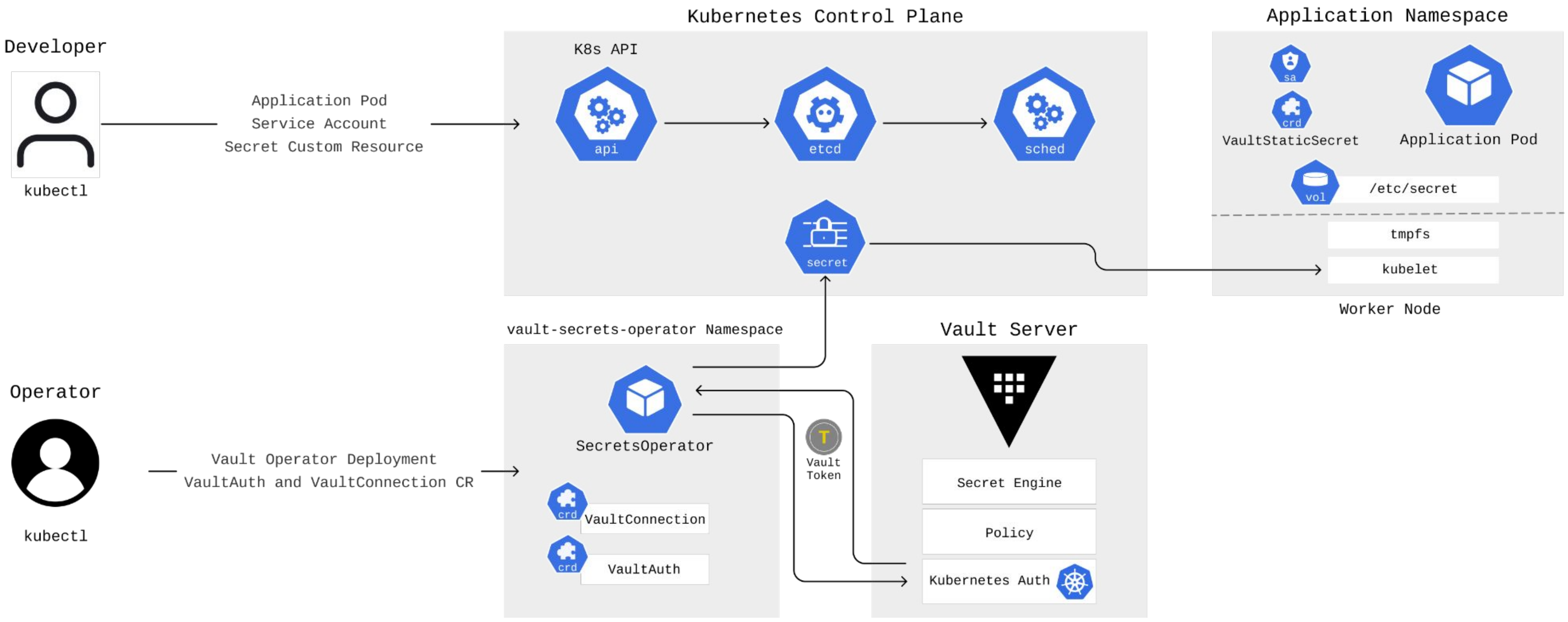


- Vault Secrets Operator (VSO) uses Kubernetes custom resources (CRDs) to manage secrets for services
- Secrets are managed by Vault and orchestrated in Kubernetes using custom resources
- The Vault Secrets Operator reconciles the current state with the desired state specified in the CRDs using declarative patterns.
- The operator facilitates secrets rotation, dynamic secrets management, and auditing capabilities
- Secret rotation is supported for ***`Deployment`***, ***`ReplicaSet`***, and ***`StatefulSet`*** resource types
- Requires:
  - Kubernetes 1.22+
  - Vault 1.11.0+

Custom Resources Deployment

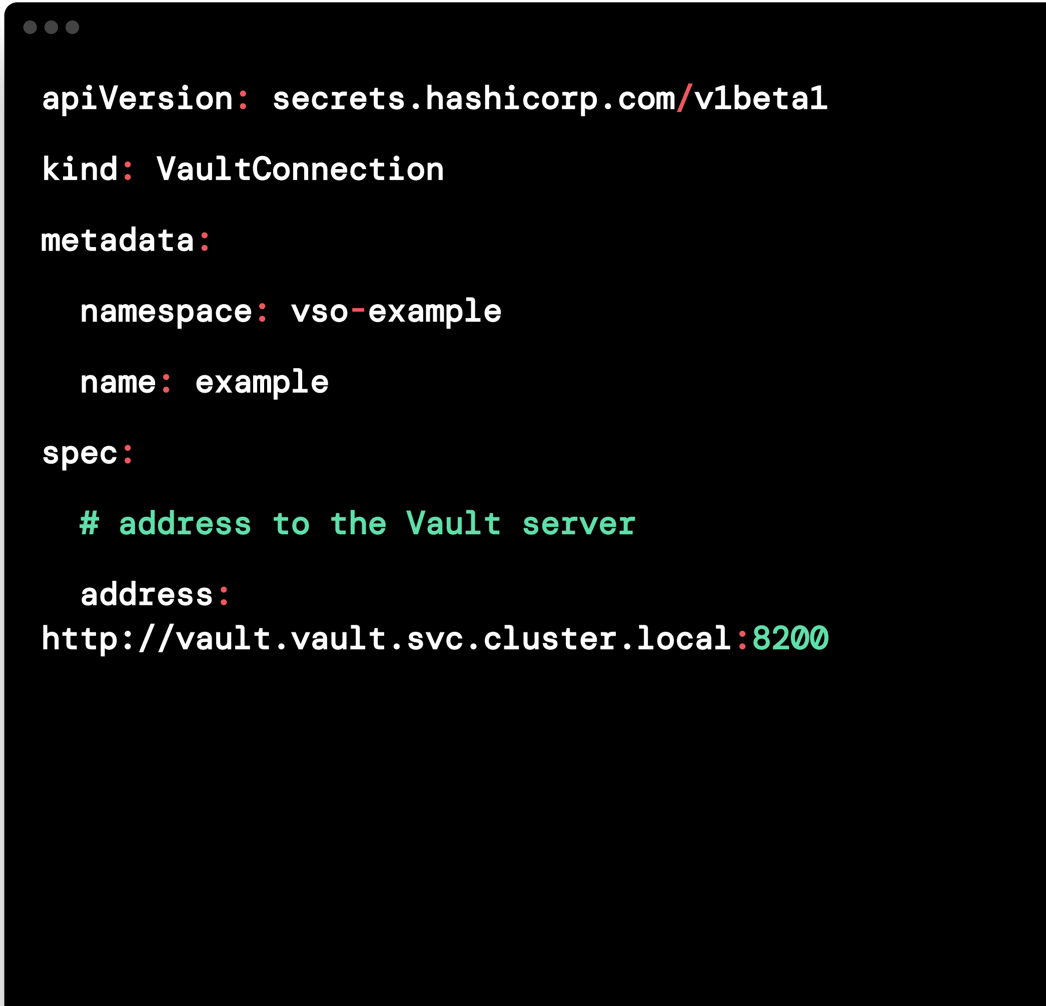
- VSO is dependent on custom resources, to set up secret sync, users first deploy the Operator through one of two supported methods:
  - Helm: Our Helm chart is available in the HashiCorp Helm repository, at https://helm.releases.hashicorp.com
  - Kustomize: The Operator's GitHub repo includes the artifacts necessary for deploying an instance of the Operator with Kustomize.
- Once sync is configured the Operator handles any of the supported Secret CRs
- VSO supports the following CRs:
  - VaultConnection
  - VaultAuth
  - VaultDynamicSecret
  - VaultPKISecret

VSO Workflow



Vault Server Connection

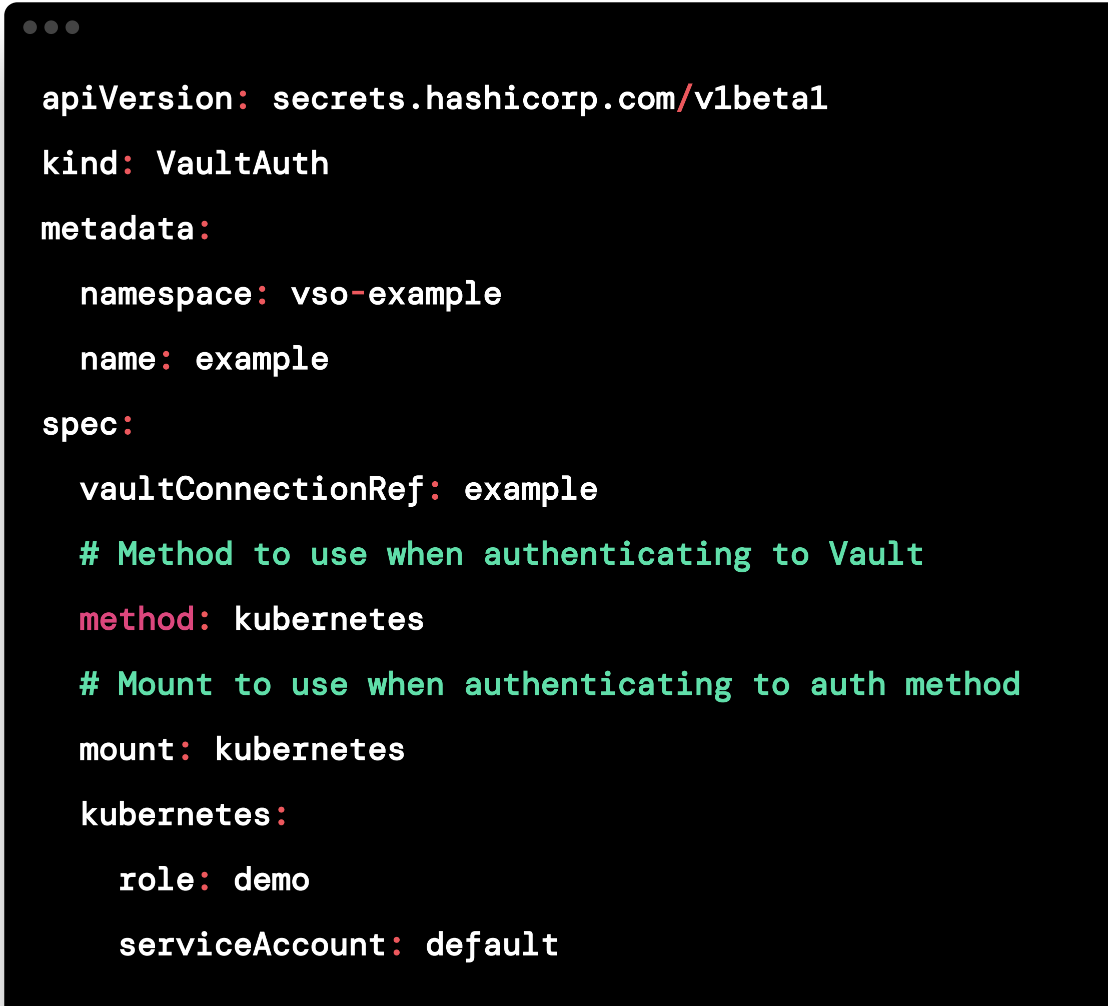
[API Reference]



```yaml
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultConnection
metadata:
  namespace: vso-example
  name: example
spec:
  # address to the Vault server
  address:
http://vault.vault.svc.cluster.local:8200
```

Vault Auth

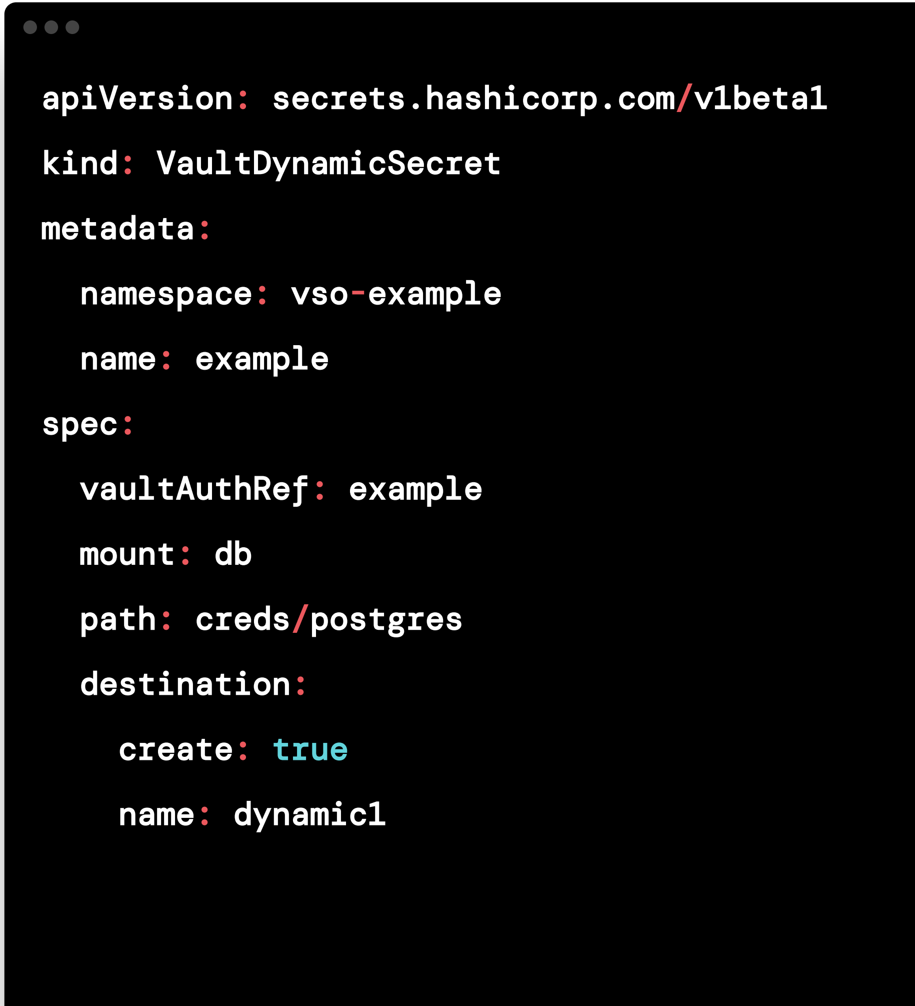
API Reference



```
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultAuth
metadata:
  namespace: vso-example
  name: example
spec:
  vaultConnectionRef: example
  # Method to use when authenticating to Vault
  method: kubernetes
  # Mount to use when authenticating to auth method
  mount: kubernetes
  kubernetes:
    role: demo
    serviceAccount: default
```

Vault Secret

[API Reference]



```yaml
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultDynamicSecret
metadata:
  namespace: vso-example
  name: example
spec:
  vaultAuthRef: example
  mount: db
  path: creds/postgres
  destination:
    create: true
    name: dynamic1
```

Required Permissions

VSO requires specific Kubernetes permissions to function correctly

Object	Permission	Reason
Secret	create, read, update, delete, watch	Sync operations, Vault auth
ServiceAccount	read token creation	Vault auth
Deployment	read, update, watch	Postsecret rotation actions

Pattern Comparison

Comparison Blog Article

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

Resources

Resources

- Vault on Kubernetes Security Considerations
- Vault on Kubernetes Deployment Guide
- Vault Helm Chart
- Vault Enterprise License Management - Kubernetes
- Helm Chart Examples
- Upgrading Vault on Kubernetes
- Running Vault - OpenShift
- Tutorials - Vault Installation to Managed Kubernetes Services
    - Google GKE
    - Azure AKS
    - Amazon EKS
- Comparison of Vault Integration Methods
- Tutorial: Vault Agent Injector
- Documentation: Agent Sidecar Injector
- Tutorial: Container Storage Interface (CSI)
- Documentation: CSI Provider
- Tutorial: Vault Secrets Operator (VSO)
- Documentation: VSO
- Integrate a Kubernetes Cluster with an External Vault

Q&A

Thank you

customer.success@hashicorp.com

www.hashicorp.com/customer-success