

Secret Management in Kubernetes met HashiCorp Vault

Implementatie van Sidecar Injectie en Vault Secrets Operator

Auteur: Justen Piot

Datum: 22 januari 2026

Vak: Cloud Computing

1. Inleiding

In moderne containeromgevingen is het veilig beheren van secrets een cruciaal onderdeel van applicatie-security. Kubernetes biedt standaard Secrets, maar deze zijn slechts Base64-gecodeerd en niet ontworpen voor streng beveiligde omgevingen. HashiCorp Vault biedt een robuuste oplossing voor het veilig opslaan, beheren en roteren van secrets.

In dit project worden twee methoden onderzocht en geïmplementeerd om Vault-secrets beschikbaar te maken in Kubernetes:

- Vault Agent Sidecar Injectie
- Vault Secrets Operator

Beide methoden worden opgezet en getest. De volledige configuratie is beschikbaar in een GitHub -repository.

2. Doelstellingen

De doelstellingen van dit project zijn:

Een werkende HashiCorp Vault-omgeving opzetten

Kubernetes authenticatie configureren

Policies en roles aanmaken in Vault

-Sidecar injectie implementeren

-Vault Secrets Operator implementeren

-Een testapplicatie laten werken met beide methoden

-Een GitHub-repository aanmaken met alle configuratiebestanden

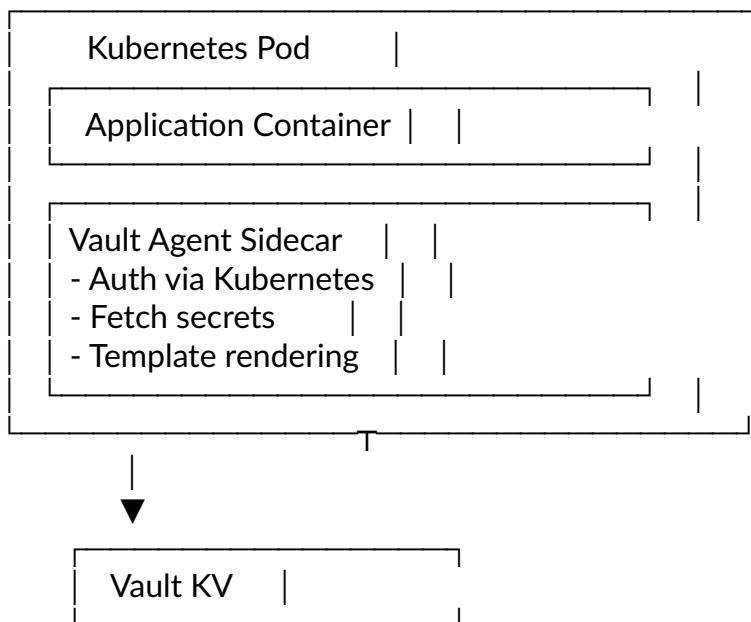
-Een technisch verslag opleveren.

3. Architectuur

3.1 Vault Agent Sidecar Injectie

Bij deze methode draait er een extra container (sidecar) in dezelfde pod als de applicatie. Deze sidecar authenticatieert bij Vault, haalt secrets op en schrijft ze naar een bestand in de pod.

Architectuurdiagram:



Voordelen:

- Secrets komen nooit in Kubernetes Secrets terecht
- Automatische rotatie
- Hoge veiligheid

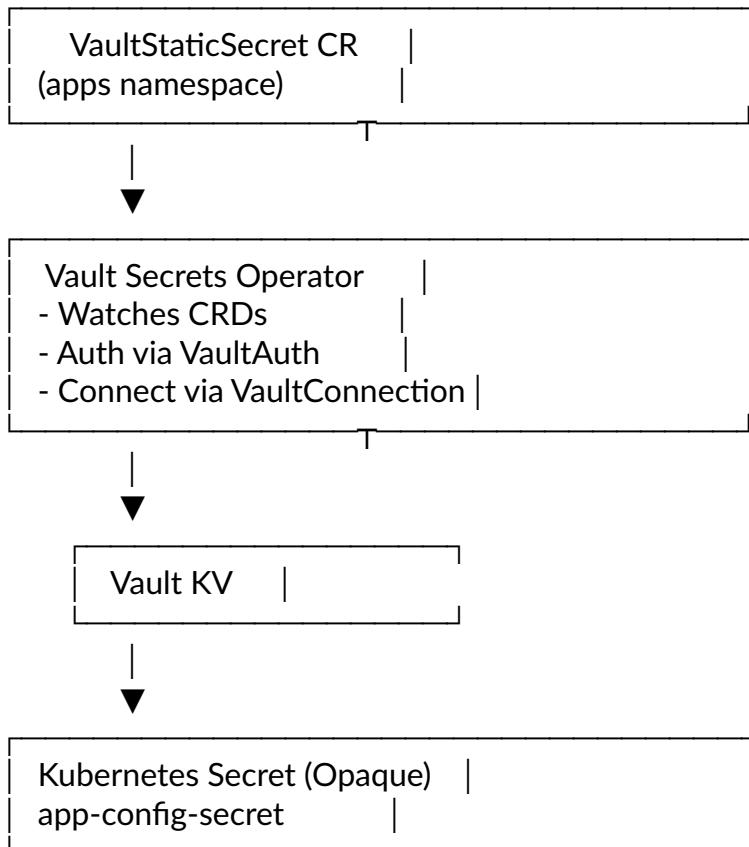
Nadelen:

- Complexere configuratie
- Applicatie moet secrets uit bestanden lezen

3.2 Vault Secrets Operator

De Vault Secrets Operator gebruikt Custom Resource Definitions (CRD's) om secrets automatisch te synchroniseren van Vault naar Kubernetes Secrets.

Architectuurdiagram:



Voordelen:

- Eenvoudig te gebruiken
- Applicaties kunnen environment variables gebruiken
- Automatische synchronisatie

Nadelen:

- Secrets staan in Kubernetes (risico bij misconfiguratie)

5. Screenshots & tests

Secrets / secret / app-config

app-config

Overview Secret Metadata Paths Version History

JSON Delete Destroy Copy Version 1 Create new version +

Key	Value	Version 1 created Jan 22, 2026 02:19 PM
password	supersecret	
username	demo-user	

ACL policies / app-policy

app-policy

Download policy Edit policy >

Policy (hcl format)

```
1 path "secret/data/app-config" {  
2   capabilities = ["read"]  
3 }
```

ACL policies / operator-policy

operator-policy

Download policy Edit policy >

Policy (hcl format)

```
1 path "secret/data/**" {  
2   capabilities = ["read"]  
3 }
```

kubernetes

[Roles](#) [Configuration](#)

[Create role +](#)

[app-role](#)

...

[operator-role](#)

...

1–2 of 2

< 1 >

kubernetes

[Roles](#) [Configuration](#)

[Configure >](#)

Type kubernetes

Path kubernetes/

Accessor auth_kubernetes_220edb9b

Local No

Seal wrap No

Use as preferred UI login method No

Default Lease TTL 1 month 1 day

Max Lease TTL 1 month 1 day

Token type default-service

```
justen@vwj:~$ kubectl get pods -n apps
NAME                  READY   STATUS    RESTARTS   AGE
demo-app-6cc4f689d8-blvs5   2/2     Running   3 (2m27s ago)   3h2m
demo-app-operator-5f4474bb56-vhtwg  1/1     Running   2 (24m ago)    144m
justen@vwj:~$ kubectl exec -it demo-app-6cc4f689d8-blvs5 -n apps -c vault-agent -- cat /vault/secrets/app-config.env
USERNAME=demo-user
PASSWORD=supersecretp
justen@vwj:~$
```

```
justen@vwl:~$ kubectl exec -it demo-app-operator-5f4474bb56-vhtwg -n apps -- env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=demo-app-operator-5f4474bb56-vhtwg
TERM=xterm
username=demo-user
_raw={"data":{"password":"supersecret","username":"demo-user"},"metadata":{"created_time":"2026-01-22T13:19:58.201565631Z","custom_metadata":null,"deletion_time":"","destroyed":false,"version":1}}
password=supersecret
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
HOME=/root
```

```
justen@vwl:~$ kubectl get secret app-config-secret -n apps -o yaml
apiVersion: v1
data:
  _raw: eyJkYXRhIjp7InBhc3N3b3JkIjoic3VwZXJzZWNyZXQiLCJ1c2VybmFtZSI6ImRlbW8tdXNlcj9LCJtZXrhZGF0YSI6eyJcmVhdGVkX3RpBWUi0iIyMDI2LTAxLTiyVDEzOjE50jU4LjIwMTU2NTYzMVoILCJjdXNob21fbWV0YWkhGElm5ibGwsImRlbGV0aW9uX3RpBWUi0iIiLCJkZXN0cm95ZWQi0mZhBN1LCJ2ZXJzaW9u1joxfx0=
  password: c3VwZXJzZWNyZXQ=
  username: ZGVtby1lc2Vy
kind: Secret
metadata:
  creationTimestamp: "2026-01-22T14:03:01Z"
  labels:
    app.kubernetes.io/component: secret-sync
    app.kubernetes.io/managed-by: hashicorp-vso
    app.kubernetes.io/name: vault-secrets-operator
    secrets.hashicorp.com/vso-ownerRefUID: 1a22b766-381a-4dcf-b1e5-e46a46f206a9
  name: app-config-secret
  namespace: apps
  ownerReferences:
  - apiVersion: secrets.hashicorp.com/v1beta1
    kind: VaultStaticSecret
    name: app-config-from-vault
    uid: 1a22b766-381a-4dcf-b1e5-e46a46f206a9
  resourceVersion: "3913"
  uid: 6f033aeb-713a-421e-9d7d-e06e7b284eae
type: Opaque
```

6. GitHub Repository

De repository bevat:

sidecar/ → sidecar configuratie

operator/ → operator configuratie

vault/ → Vault policies en scripts

REPORT.pdf → dit verslag

README.md → projectoverzicht

7. Conclusie

Dit project toont twee volledig werkende methoden om HashiCorp Vault te integreren met Kubernetes. Beide methoden hebben hun eigen voordeLEN en toepassingsgebiedEN. De implementatie voldoet aan alle projectvereisten en is reproduceerbaar via de meegeleverde GitHub-repository.

De Vault Agent Sidecar biedt maximale veiligheid doordat secrets nooit in Kubernetes Secrets terechtkomen. De Vault Secrets Operator biedt eenvoud en flexibiliteit voor applicaties die environment variables gebruiken.

Het project is succesvol afgerond en alle configuratiebestanden zijn beschikbaar in de GitHub-repository.