Secrets faciles dans Kubernetes : Parce que je le Vault bien

DEEZER

~#whoami : Alexis Fala

~~Clown~~ Infrastructure engineer | SRE chez DEEZER

- → Passionné par l'open-source et le Cloud Native.
- → Guitariste du dimanche spécialisé en début de chansons.
- → Membre des **SRE du coeur** avec Idriss Neumann et Julien Briault.
- → Auteur à ses heures perdues sur Deezer.io.

LePotiBlagueur

@LePotiBlagueur

https://www.linkedin.com/in/alexis-fala/

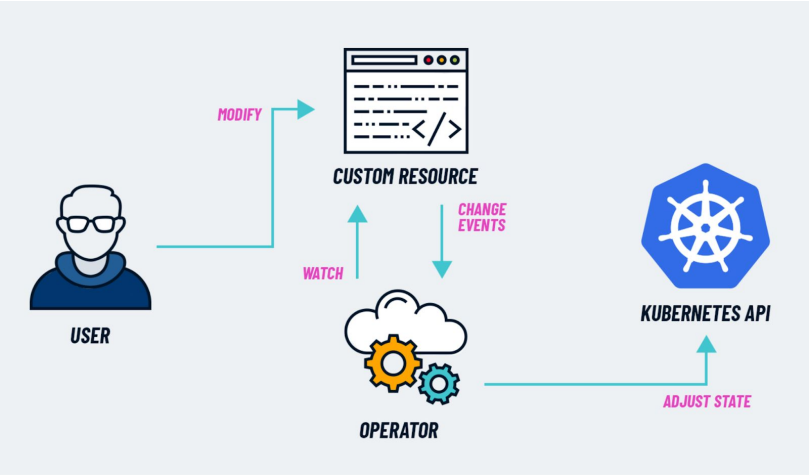
Sommaire

Qu’est-ce qu’un operator ?

Kubernetes Operators

Les opérateurs sont des extensions de l'API/Control plane de Kubernetes.

- Permettent de manager des applications via l'utilisation de CRD en les associant à un controller.
- Se basent sur les principes du control loop.
- Peut gérer des composants internes et externes au cluster.
- Il en existe une grande variété : cert-manager pour gérer les certificats TLS, CNPG pour déployer Postgrès...
- A la recherche d'un Operator ? https://operatorhub.io/



Source cncf.io

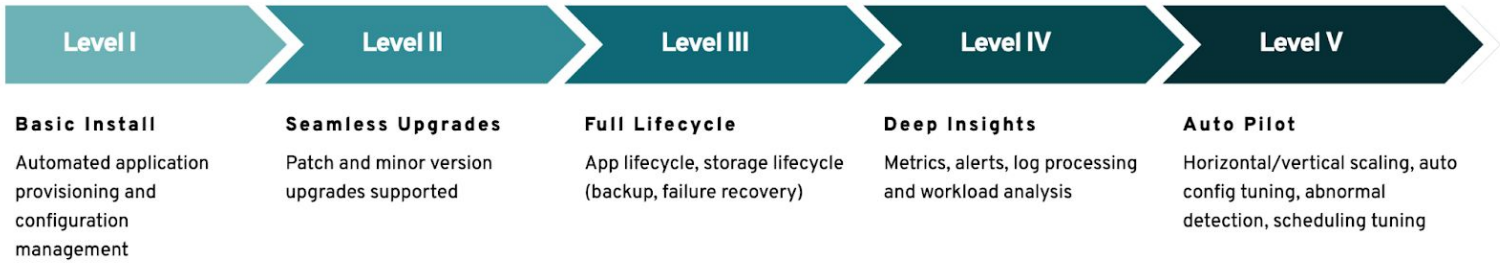
Vous pouvez écrire vos propres opérateurs.

Comment écrire son opérateur ?

S'aider d'un framework

- Operator Framework : https://operatorframework.io/
- Kubebuilder : https://kubebuilder.io/
- Kubernetes Operators Framework : https://kopf.readthedocs.io/en/stable/
- ...

Plusieurs niveaux de capacité



Source operatorframework.io

Si ça ne tourne pas dans Kube, ça peut être géré via Kube.

DEEZER

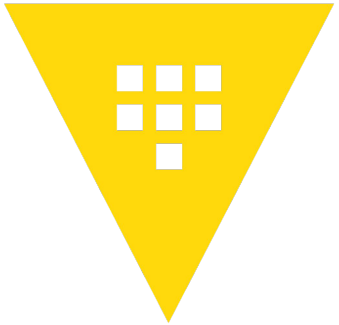
Introduction à Vault

DEEZER

Vault ? Kesako ?

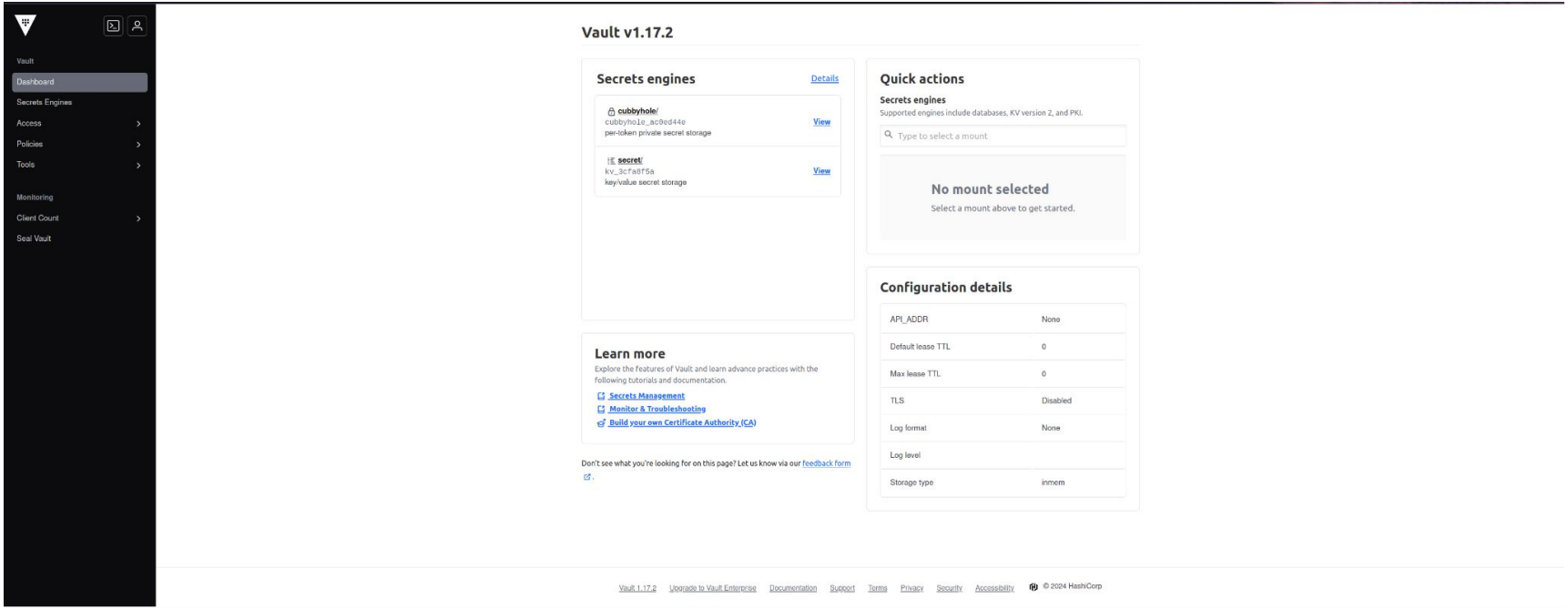
C’est l’outil de gestion centralisée de secrets le plus utilisé du marché.

- Coffre-fort “API driven” avec un système d’authentification et de secret engine universel.
- Permet de stocker mot de passe, tokens, certificats, ...
- Son fonctionnement est basé sur des tokens, associés à des Policies.
- Chaque Policy est basée sur un “path”.
- Chaque path est limité par des Policy Rules.
- Authentification → Validation → Autorisation → Accès



HashiCorp
Vault

Vault UI

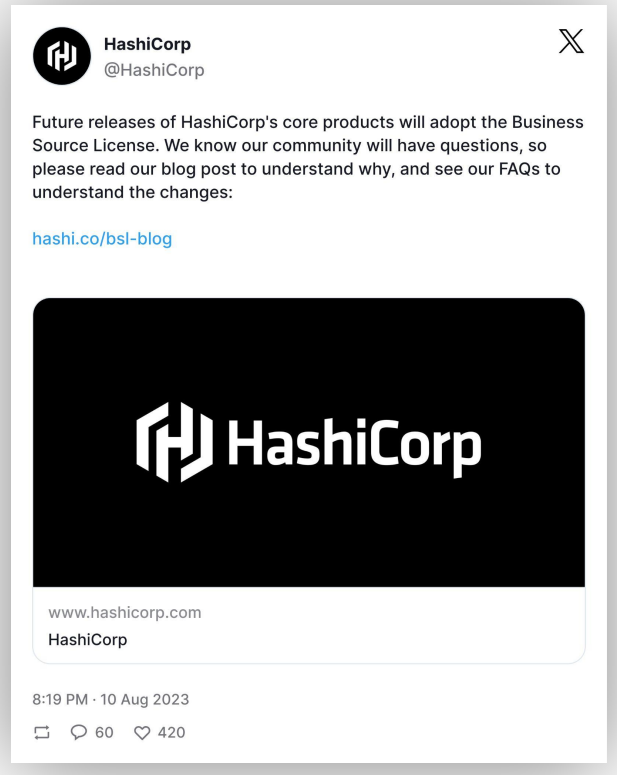


DEEZER

Alerte drama

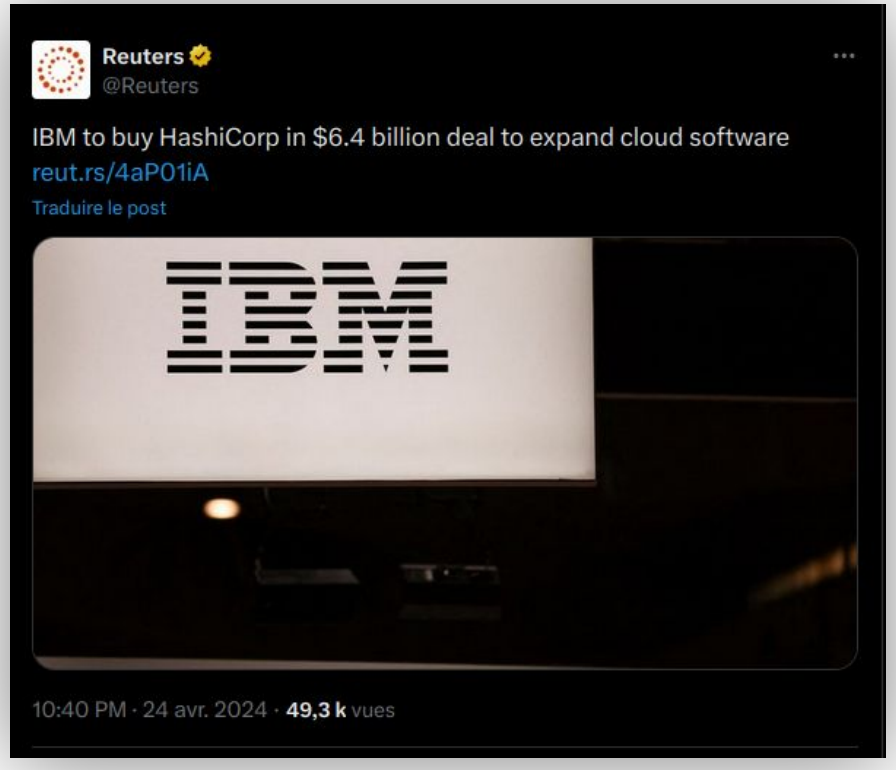
DEEZER

La Business Source Licence



DEEZER

Le rachat par IBM



Les secrets dans Kubernetes

DEEZER

Un secret dans le cluster

Un secret certes... Mais pas si secret

- Contient des informations sensibles
- Permet de stocker mot de passe, tokens, certificats, ...
- Encodé en base64 😬

```yaml
apiVersion: v1
kind: Secret
metadata:
 name: mon-super-secret
stringData:
 password: password1234
```

DEEZER

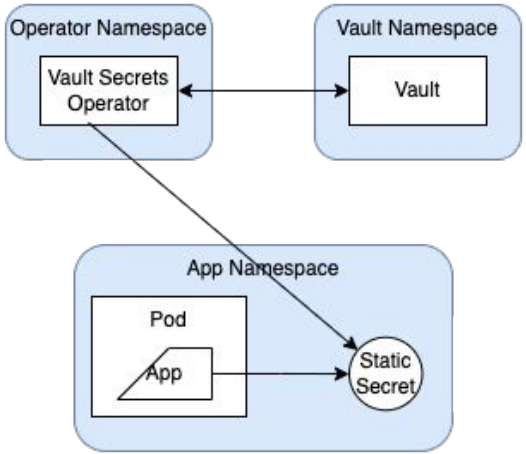
Le Vault Secrets Operator

DEEZER

Un operator pour Vault par Vault

Son nom ne marche qu'avec Vault

- A pour but de synchroniser les secrets contenus dans Vault et les secrets Kubernetes.
- Supporte toutes les features de Vault (TLS certificates, Secret engines, secrets dynamiques et statiques...)
- Permet de rollout restart dynamiquement un déploiement ou un pod.
- Possède un grand nombre de CRD compatible Vault Cloud et on-prem.



Source hashicorp.com

Comment le déployer ?

C'est simple !

```
helm install vault hashicorp/vault -n vault
--create-namespace --values vault/vault-values.yaml
```

https://artifacthub.io/packages/helm/hashicorp/vault-secrets-operator

Les étapes pour faire marcher l'opérateur

Création d'un service account

Première étape

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
 name: operator-auth
```

Création d'un service account

Première étape (bis)

```yaml
apiVersion: v1
kind: Secret
metadata:
 name: operator-auth
 annotations:
   kubernetes.io/service-account.name: operator-auth
type: kubernetes.io/service-account-token
```

Création d'un service account

Première étape (ter)

```
apiVersion:
rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
 name: role-tokenreview-binding
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: ClusterRole
 name: system:auth-delegator
subjects:
 - kind: ServiceAccount
   name: operator-auth
   namespace: [namespace du SA]
```

La configuration du Vault

Seconde étape : on écrit une policy

```
vault policy write vault-secret-operator
                ./policy.hcl
```

```
path "kvv2/data/demo/*" {
   capabilities = ["read", "list"]
}
```

La configuration du Vault

Seconde étape (bis) : on créé notre authentification Kube

```
vault auth enable -path vault-secret-operator kubernetes
```

La configuration du Vault

Seconde étape (ter) : on créé la config de notre authentification

```
vault write auth/vault-secret-operator/config
token reviewer jwt="$TOKEN REVIEW_JWT"
kubernetes host="$KUBE HOST"
kubernetes_ca_cert="$KUBE_CA_CERT"
```

La configuration du Vault

Seconde étape (quater) : on créé un rôle à notre authentification

```
vault write
auth/vault-secret-operator/role/default
bound service account names=operator-auth
bound service account namespaces=”*”
policies=vault-secret-operator
```

La connexion au Vault

Troisième étape : l'utilisation d'une VaultConnection

```
kind: VaultConnection
metadata:
 name: myvaultconnection
 namespace: vault
spec:
 address: [adresse de votre vault]
 skipTLSVerify: false
```

L’authentification au Vault

Quatrième étape : l’utilisation d’une VaultAuth

```yaml
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultAuth
metadata:
 name: myvaultauth
spec:
 vaultConnectionRef: myvaultconnection
 method: kubernetes
 mount: vault-secret-operator
 kubernetes:
   role: default
   serviceAccount: operator-auth
```

La déclaration d'un secret

Cinquième (et dernière) étape : on déclare un secret !

```yaml
apiVersion: secrets.hashicorp.com/v1beta1
kind: VaultStaticSecret
metadata:
 name: vault-static-secret
spec:
 vaultAuthRef: [namespace]/myvaultauth
 mount: kvv2
 type: kv-v2
 path: demo/config
 refreshAfter: 10s
 destination:
   create: true
   name: my-secret
 rolloutRestartTargets:
 - kind: Deployment
   name: myapp
```

Et voilà !

Petite ouverture...

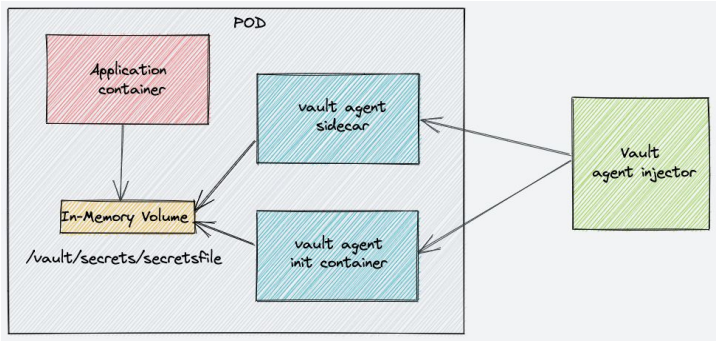
DEEZER


L'opérateur ne renforce pas la sécurité de vos secrets

Le vault agent injector

Là, on est vraiment secure...

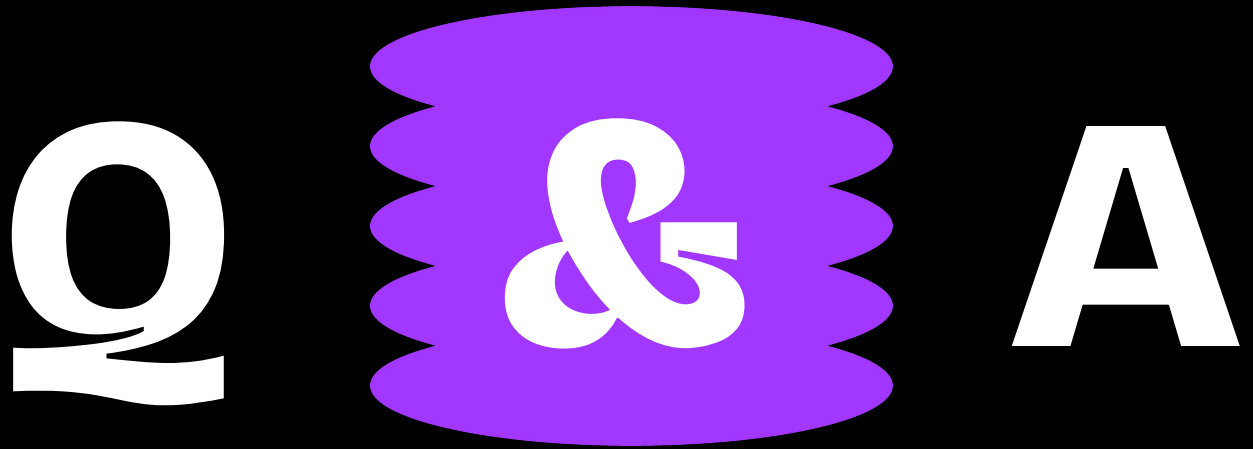
- Agent sidecar appelé par des annotations dans vos déploiements/pods
- Injecte directement vos secrets à l'intérieur des pods
- Supporte le *consul-template*
- Forte dépendance à la disponibilité de Vault **: risque de bloquer vos déploiements !**



Source devopscube.com

D...D...Demo !

DEEZER


Q
&
A

Laissez-moi un commentaire sur

openfeedback

J’suis gentil : je vous donne accès au talk !

DEEZER