

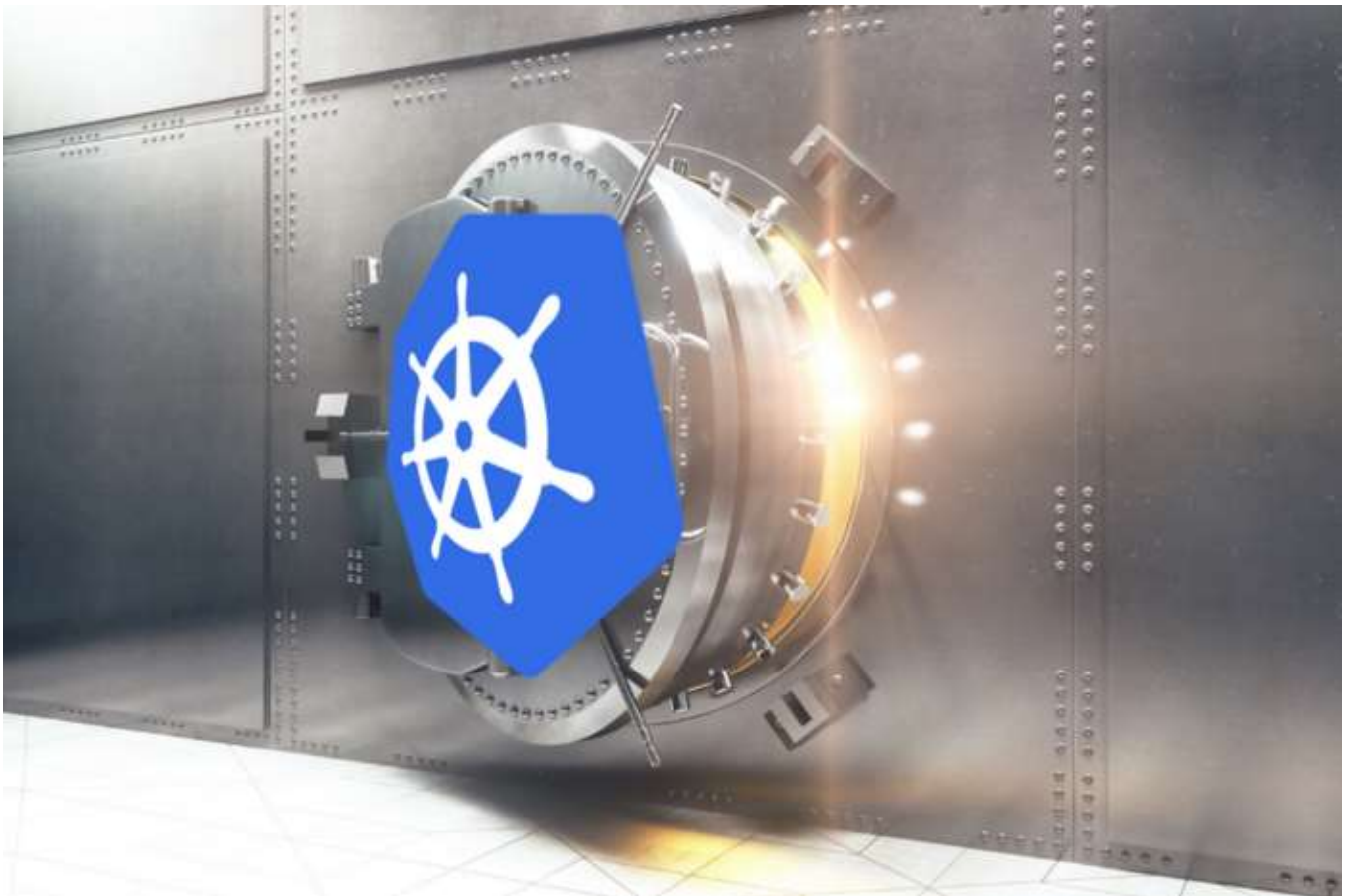
# Securing your secrets using vault-k8s in Kubernetes — Part 1



Pavan Kumar

Jul 18, 2020 · 3 min read

Kubernetes secrets let you store and manage sensitive data such as passwords, ssh keys, Tls certificates, etc. However, there are few limitations to using the build-in secret management for Kubernetes. So, we often tend to rely on some third-party tools to handle secret management. One such tool is HashiCorp Vault. In this series of articles let's learn to secure our secrets using HashiCorp Vault-k8s in Kubernetes.



Integration of Vault and Kubernetes with vault-k8s

## Installing vault in our cluster using Helm Chart :

### 1. Adding the helm repo for vault and Installing the helm chart:

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

```
helm install --name=vault --set='server.dev.enabled=true'
```

After the successful installation, you will find 2 pods running, lets exec in to our vault server to create our policies and roles

```
[root@master vault-injection]# k get po
NAME                                READY   STATUS    RESTARTS   AGE
vault-0                            1/1     Running   0           32h
vault-agent-injector-7d8b64b8d4-dt9g2 1/1     Running   0           32h
[root@master vault-injection]#
```

### 2. Creation of roles and policies and configuring the Kubernetes Auth Method.

```
kubectl exec -it vault-0 -- /bin/sh
```

Now let us configure the Kubernetes Auth Method

```
vault auth enable kubernetes
```

```
vault write auth/kubernetes/config \
  token_reviewer_jwt="$(cat
/var/run/secrets/kubernetes.io/serviceaccount/token)" \
  kubernetes_host=https://${KUBERNETES_PORT_443_TCP_ADDR}:443 \
```

```
kubernetes_ca_cert=@/var/run/secrets/kubernetes.io/serviceaccount/ca
.crt
```

Let's create a policy called prod-policy and prod-role. Add the name of the service account and the namespace to which you want to give access over this secret and let's create a secret called prod-secret.

```
vault policy write "prod-policy" -<<EOF
path "secret*" {
  capabilities = ["read", "update"]
}
EOF
```

```
vault write auth/kubernetes/role/prod-role \
  bound_service_account_names=prod-sa \
  bound_service_account_namespaces=default \
  policies=prod-policy \
  ttl=1h
```

```
vault kv put secret/prod-secret username=pavankumar password=Kubeadm
```

Hurrah!!! We created our first secret and configured the access over this secret to our Kubernetes ServiceAccount ( prod-sa ). In the next article, we would be learning about **How to Inject the vault secrets into Kubernetes.**

👏 **Join FAUN today and receive similar stories each week in your inbox!** Get your weekly dose of the must-read tech stories, news, and tutorials.

Follow us on [Twitter](#) 🐦 and [Facebook](#) 👤 and [Instagram](#) 📷 and join our [Facebook](#) and [Linkedin](#) Groups 💬

If this post was helpful, please click the clap 🖐️ button below a few times to show your support for the author! ↓

[Kubernetes](#)[Vault](#)[Hashicorp Vault](#)[Vault K8s](#)[Kubernetes Secret](#)

Get the Medium app

