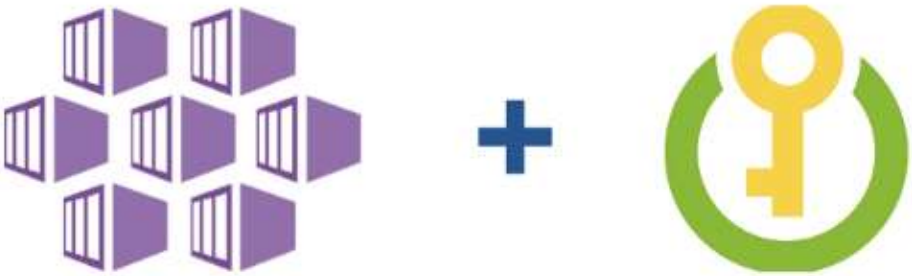
Integrate Azure Key Vault with AKS — Using “FlexVolume” (Part 2/3)

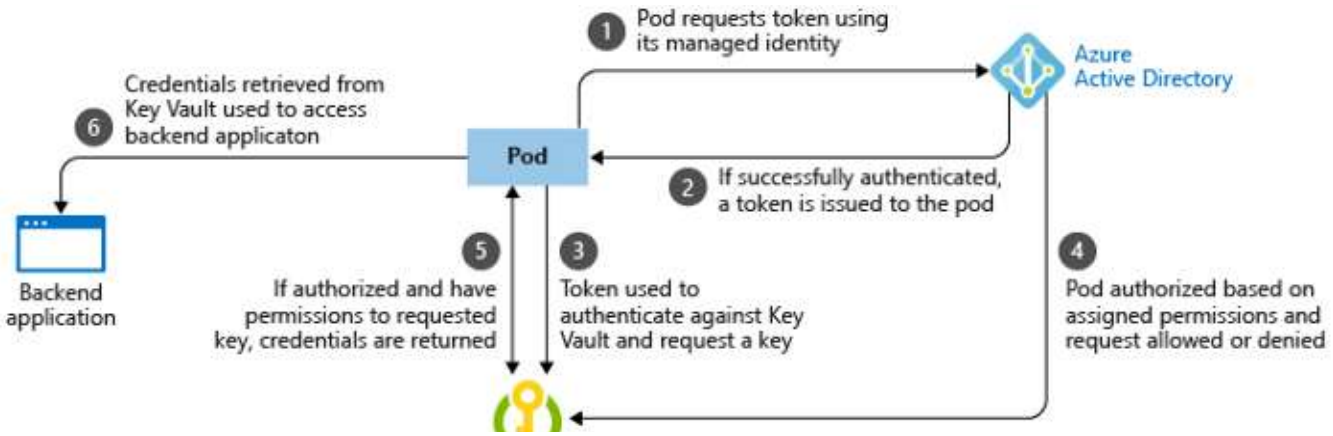
Leon Jalfon
Apr 28, 2020 · 3 min read

In this 3-parts tutorial we will explain how to integrate AKS with Azure Key Vault using “FlexVolumes” and “Azure Key Vault to Kubernetes”. However, before we get down to work let’s talk a little about each approach.



In this section

With FlexVolumes Key Vault secrets, keys, and certificates become a volume accessible to pods. Once the volume is mounted, its data is available directly in the container filesystem for your application.



For more information visit the official documentation:
https://github.com/Azure/kubernetes-keyvault-flexvol

What will we do in this tutorial?

- Configure your environment (set some environment variables)
- Install Key Vault FlexVolume in your AKS cluster
- Configure FlexVolume (create a secret with service principal details)
- Deploy a pod that access to a Key Vault secret
- Cleanup

Configure your environment

Let’s configure some environment variables that will be used during the tutorial

```
KEY_VAULT_NAME=<your-key-vault-name>
KEY_VAULT_SECRET_NAME=<your-secret-name>
SERVICE_PRINCIPAL_CLIENT_ID=<your-service-principal-client-id>
SERVICE_PRINCIPAL_CLIENT_SECRET=<your-service-principal-secret>
SERVICE_PRINCIPAL_TENANT_ID=<your-service-principal-tenant-id>
```

Installation

Deploy Key Vault FlexVolume to your AKS cluster

```
kubectl create -f
https://raw.githubusercontent.com/Azure/kubernetes-keyvault-
flexvol/master/deployment/kv-flexvol-installer.yaml
```

To validate Key Vault FlexVolume is running as expected, run the following command:

```
kubectl get pods -n kv
```

The output should show keyvault-flexvolume pods running on each agent node:

```
NAME                          READY     STATUS    RESTARTS   AGE
keyvault-flexvolume-f7bx8     1/1       Running   0          3m
keyvault-flexvolume-rcxbl     1/1       Running   0          3m
keyvault-flexvolume-z6jm6     1/1       Running   0          3m
```

Configuration

There are 4 options to configure FlexVolume: Service Principal, Pod identity, VMSS User Assigned Managed Identity or VMSS System Assigned Managed Identity (in this tutorial we will use a service principal due it’s the simplest option)

Add your service principal credentials as a Kubernetes secret

```
kubectl create secret generic kvcreds --from-literal
clientid=${SERVICE_PRINCIPAL_CLIENT_ID} --from-literal
clientsecret=${SERVICE_PRINCIPAL_CLIENT_SECRET} --type=azure/kv
```

Usage

Let’s create a pod to test our configuration:

```
cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: flex-kv-test
spec:
  containers:
  - name: flex-kv-test
    image: nginx
    volumeMounts:
    - name: test
      mountPath: /kvmnt
      readOnly: true
  volumes:
  - name: test
    flexVolume:
      driver: "azure/kv"
      secretRef:
        name: kvcreds
      options:
        usepodidentity: "false"
        keyvaultname: "${KEY_VAULT_NAME}"
        keyvaultobjectnames: ${KEY_VAULT_SECRET_NAME}
        keyvaultobjecttypes: secret
        keyvaultobjectversions: ""
```

```
          tenantid: "${SERVICE_PRINCIPAL_TENANT_ID}"
EOF
```

Note that the flexvolume configuration receive the following options:

- **usepodidentity:** (optional) if not provided, will default to “false”
- **keyvaultname:** the name of the KeyVault
- **keyvaultobjectnames:** list of KeyVault object types: secret, key or cert (semi-colon separated)
- **keyvaultobjecttypes:** list of KeyVault object types: secret, key or cert (semi-colon separated)
- **keyvaultobjectversions:** (optional) list of KeyVault object versions (semi-colon separated), default:latest
- **tenantid:** the tenant ID of the KeyVault

Ensure that you have access to keyvault from the deployed pod

```
kubectl exec -it flex-kv-test cat /kvmnt/${KEY_VAULT_SECRET_NAME}
```

Cleanup

Delete the test pod

```
kubectl delete pod flex-kv-test
```

Delete the secret that store the service principal

```
kubectl delete secret kvcreds
```

Uninstall FlexVolume from your AKS cluster

```
kubectl delete daemonset keyvault-flexvolume -n kv
kubectl delete namespace kv
```

Integrate Azure Key Vault with AKS — Introduction (Part 1/3)

Integrate Azure Key Vault with AKS — Using “akv2k8s” (Part 3/3)

DevOps Kubernetes Azure Keyvault Devsecops