# secrets or not, but don’t clear text

current state of secret management with kubernetes

FOSDEM 2021

lightning talks

romuald vandepoel (He/Him/His)
aka rom - romuald@redhat.com

# who am i?

fair question...

romuald aka rom

red hat consulting

automation & openshift & storage

# why secrets within k8s?

another fair question...

there are recurring topics from an implementation standpoint:

- networking
- storage
- security

# secrets?

something not to share with anyone...

... but to used to access something

from k8s view:
- a password
- a token
- a ssh key
- a configuration file

design dates from 2017

# how does k8s deal with secrets?

where do I save my db password?

options are:
- within pod
- within k8s internals
- outside of k8s internals

# saving my secrets...

my db password which is F0sd3M2O21

options are:

- ... ***within pods***
- within k8s internals
- outside of k8s internals

# saving my secrets...

my db password which is F0sd3M2O21

options are:

- within pods
- ***… within k8s internals***
- outside of k8s internals

```
echo -n "F0sd3M2O21" | kubectl create secret generic
mydbsecret -n secret-env-pod --dry-run=client
--from-literal=username=admin
--from-file=password=/dev/stdin -o yaml >
mydbsecret.yam

---mydbsecret.yaml---
apiVersion: v1
data:
  password: RjBzZDNNMk8yMQ==
  username: YWRtaW4=
kind: Secret
metadata:
  creationTimestamp: null
  name: mydbsecret
  namespace: secret-env-pod
---eof---

kubectl apply -f ./mydbsecret.yaml
```

```
---mypoddefinition.yaml---
apiVersion: v1
kind: Pod
metadata:
  name: secret-env-pod
spec:
  containers:
  - name: mycontainer
    image: redis
    env:
      - name: SECRET_USERNAME
        valueFrom:
          secretKeyRef:
            name: mydbsecret
            key: username
      - name: SECRET_PASSWORD
        valueFrom:
          secretKeyRef:
            name: mydbsecret
            key: password
  restartPolicy: Never
---eof---
```

within the container, call for:

- $SECRET_USERNAME
- $SECRET_PASSWORD

# saving my secrets...

my db password which is F0sd3M2O21

options are:

- within pods
- within k8s internals
- ... ***outside of k8s internals***

# saving my secrets outside of k8s internals

my db password which is F0sd3M2021

options are:

- hashicorp vault
- cyberark conjur
- keywhiz (from squares)
- seal secrets
- external secrets

# sealed secrets

embrace the everything-as-a-code

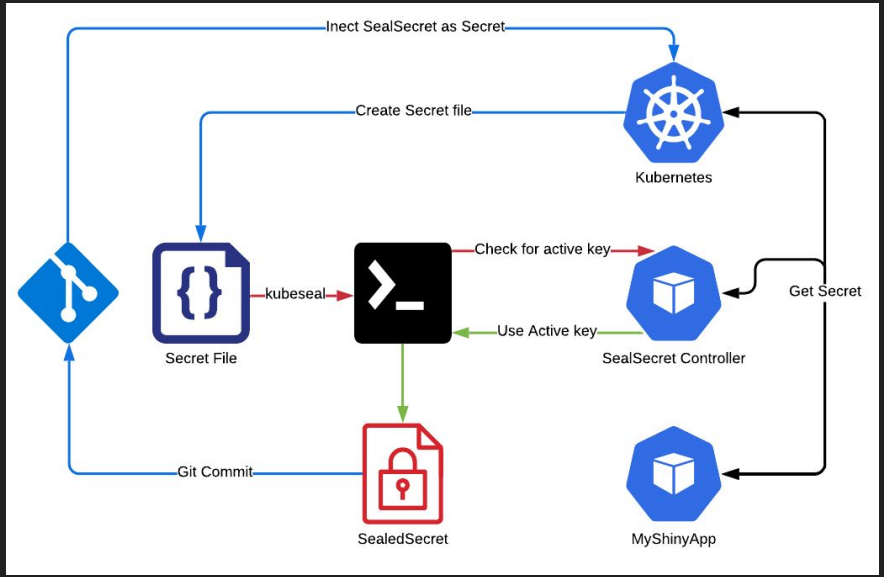


image source: romuald vandepoel - gpl license

```
echo -n "F0sd3M2O21" | kubectl create secret generic
mydbsecret -n secret-env-pod --dry-run=client
--from-literal=username=admin
--from-file=password=/dev/stdin -o yaml >
mydbsecret.yam

---mydbsecret.yaml---
apiVersion: v1
data:
  password: RjBzZDNNMk8yMQ==
  username: YWRtaW4=
kind: Secret
metadata:
  creationTimestamp: null
  name: mydbsecret
  namespace: secret-env-pod
---eof---

kubeseal --format yaml <mydbsecret.yaml
>sealedmydbsecret.yaml

kubectl apply -f sealedmydbsecret.yaml
```

```
---sealedmydbsecret.yaml---
apiVersion: bitnami.com/v1alpha1
kind: SealedSecret
metadata:
  creationTimestamp: null
  name: mydbsecret
  namespace: secret-env-pod
spec:
  encryptedData:
    password: AgAAKA...
    username: By3i+S...
---eof---
```

# external secrets

use your existing key management service (kms)

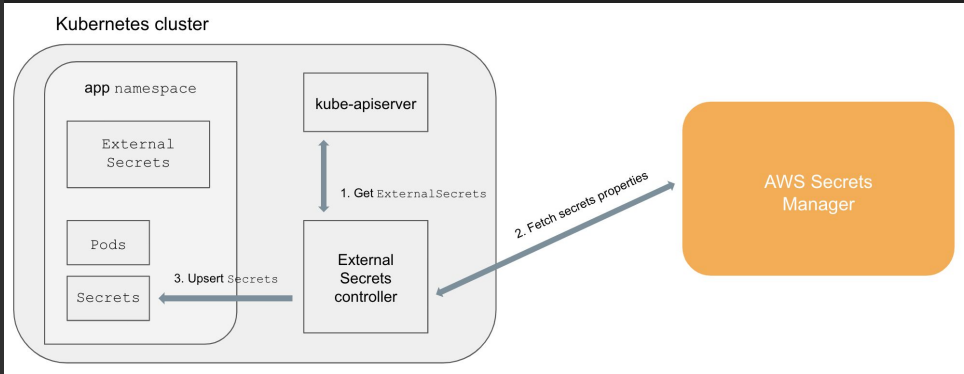


image source: https://github.com/external-secrets/kubernetes-external-secrets/blob/master/architecture.png

# conclusions

kiss principle

food for thoughts:

- kubernetes secrets resources
- do you have an existing kms?
- are you embracing the GitOps principles?
- why is secret management not having much love?

thank you

stay safe
like your secrets