External Secrets Operator: A Cloud Native way to manage your secrets.

[EXTERNAL SECRETS OPERATOR]

Container Solutions

~ > whoami

- Charl Klein 🇿🇦🇳🇱
- Cloud Native Engineer @ Container-Solutions
- Experience with Client / Server Support, Networking, ITSM, Development (Integration / Backend), Pre-Sales, DevOps, and Cloud.
- Working with K8s since 2017/8.
- Passionate about Security
  - Pursuing a BSc in CyberSecurity (IU University - Germany)

@CharlKlein

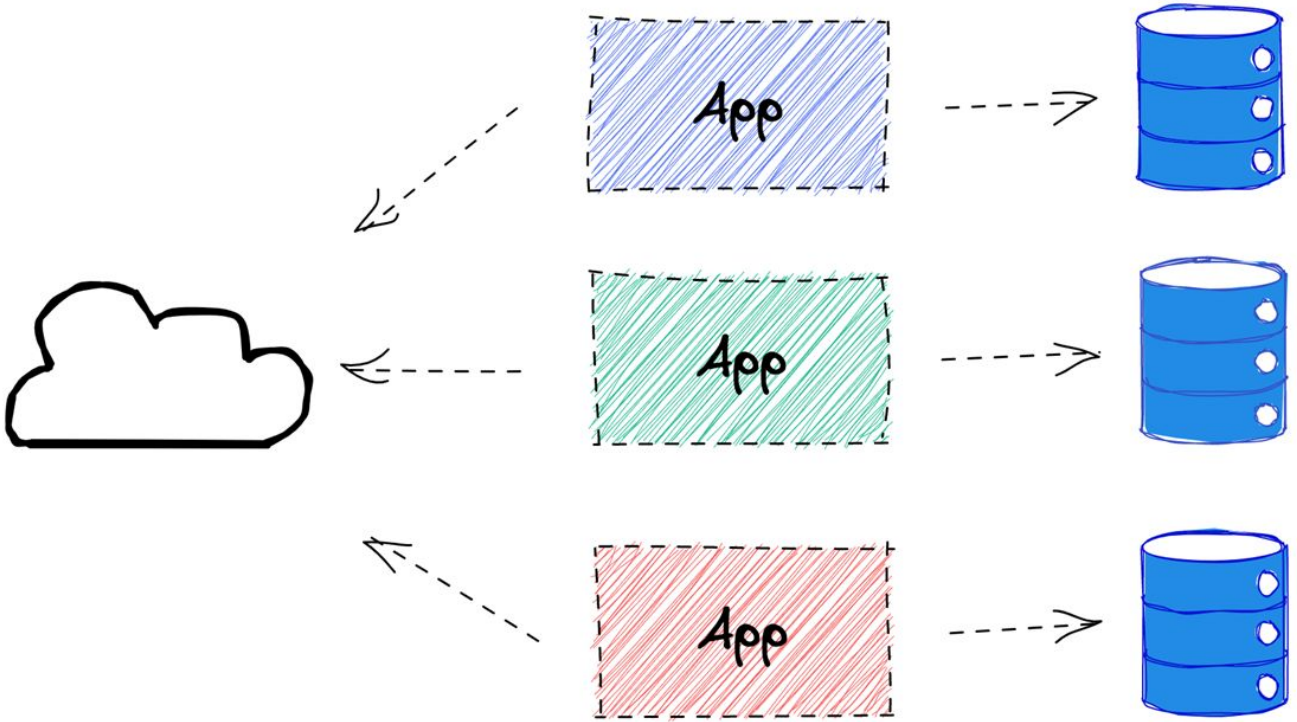
Agenda

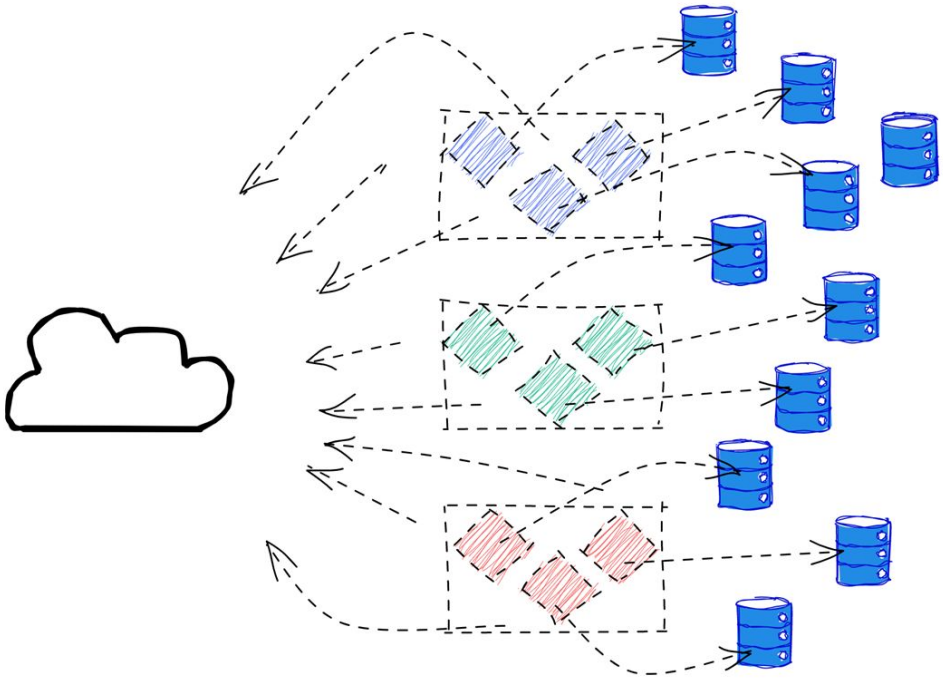
- Intro - Problem
- A Beautiful Open Source Story
- The current state of the Project
- How it Works
- DEMO!

Intro

The problem and solution


App


App
App
App

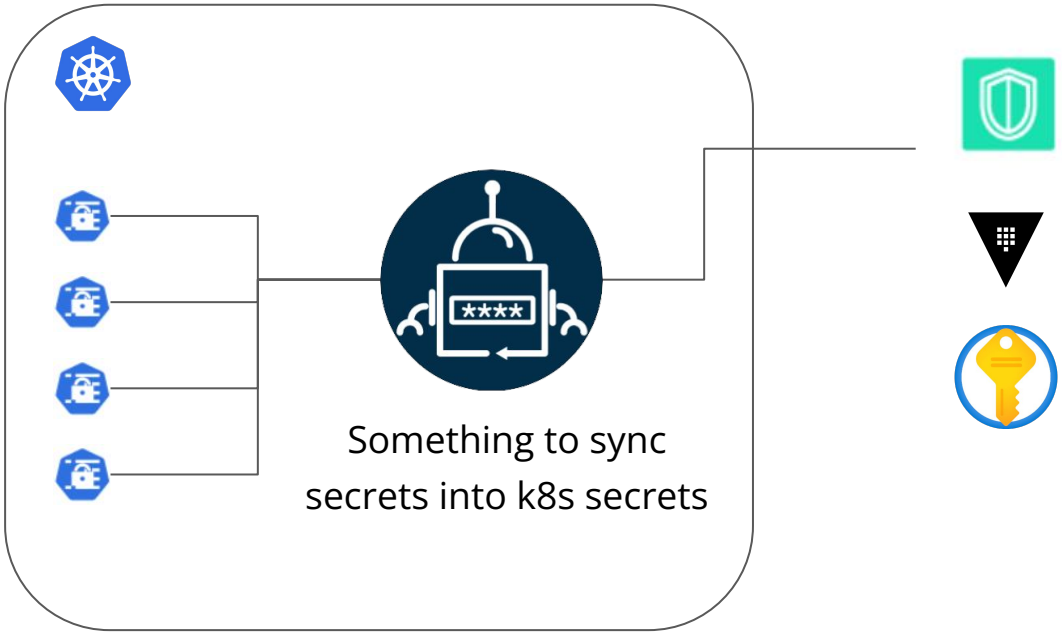


Problem

Challenges In a multi service and multi environment setup in k8s

- You can end up with hundreds of secrets to manage
- Hard to handle rotation
- Hard to onboard new services
- Hard to onboard new people with specific accesses
- Distributing secrets securely

Possible Solution



The appeal of syncing secrets

- AWS SM, GCP SM, Vault can be used with other services (integrations / providers)
- Keep your secret where it is secure (ie, In a Service that designed for it)
  - Abstract away the access
- Use great features from these services (like encryption at rest, easier rotation mechanisms)
- Declare secret reference in source code
- Avoid access to the actual secrets
- Environment specific secrets with same manifest

The Get together

Centralizing External Secrets Solutions
Beautiful open source history

How we got together

- Discussion started in a issue about a similar project [kubernetes-external-secrets/pull/47]
- A lot of similar solutions started to get mentioned in the issue
  - https://github.com/ContainerSolutions/externalsecret-operator
  - https://github.com/itscontained/secret-manager
  - https://github.com/mumoshu/aws-secret-operator
  - https://github.com/cmattoon/aws-ssm
  - https://github.com/tuenti/secrets-manager
  - https://github.com/kubernetes-sigs/k8s-gsm-tools
  - https://github.com/godaddy/kubernetes-external-secrets - **2k stars on github**
  - https://github.com/kubernetes-sigs/secrets-store-csi-driver

Consider merging with aws-secret-operator #47

Closed **max-lobur** opened this issue on Apr 16, 2019 · 19 comments

max-lobur commented on Apr 16, 2019

https://github.com/mumoshu/aws-secret-operator is a very similar concept, consider merging two projects.

👍 3

silasbw commented on Apr 16, 2019 • edited Member

Doh -- just when you think you've scoured the internet. Thanks **@max-lobur**. Yes, we agree, we should definitely consider merging.

👍 3 😄 2 🚀 1

lhotrifork commented on May 9, 2019

There's also this :)
https://github.com/ContainerSolutions/externalsecret-operator

silasbw commented on May 21, 2019 Member

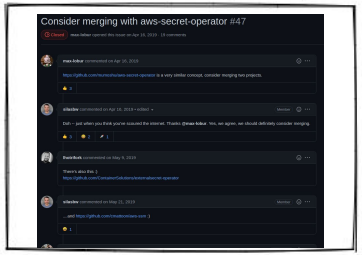
....and https://github.com/cmattoon/aws-ssm :)

😄 1

The First Steps

- Provide a standard implementation of External Secrets
  - Common CRD as first step (kubernetes-external-secrets/pull/477)
  - State of the art way to deal with secrets because we leverage external providers
- Make it easy to migrate between the different solutions

Issue on Godaddy's repo

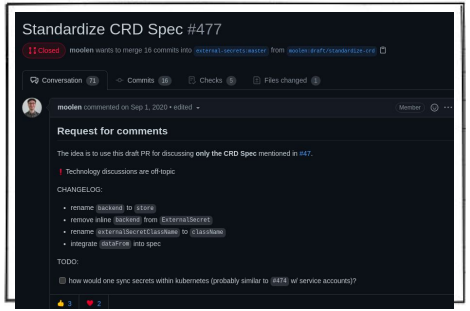


initial discussion about similar solutions and merging efforts

Similar solutions

https://github.com/ContainerSolutions/externalsecret-operator
https://github.com/itscontained/secret-manager
https://github.com/mumoshu/aws-secret-operator
https://github.com/cmattoon/aws-ssm
https://github.com/tuenti/secrets-manager
https://github.com/kubernetes-sigs/k8s-gsm-tools
https://github.com/godaddy/kubernetes-external-secrets
https://github.com/kubernetes-sigs/secrets-store-csi-driver

Initial effort on common CRD



New org hosting popular Godaddy's project and the new final de facto solution

https://github.com/external-secrets

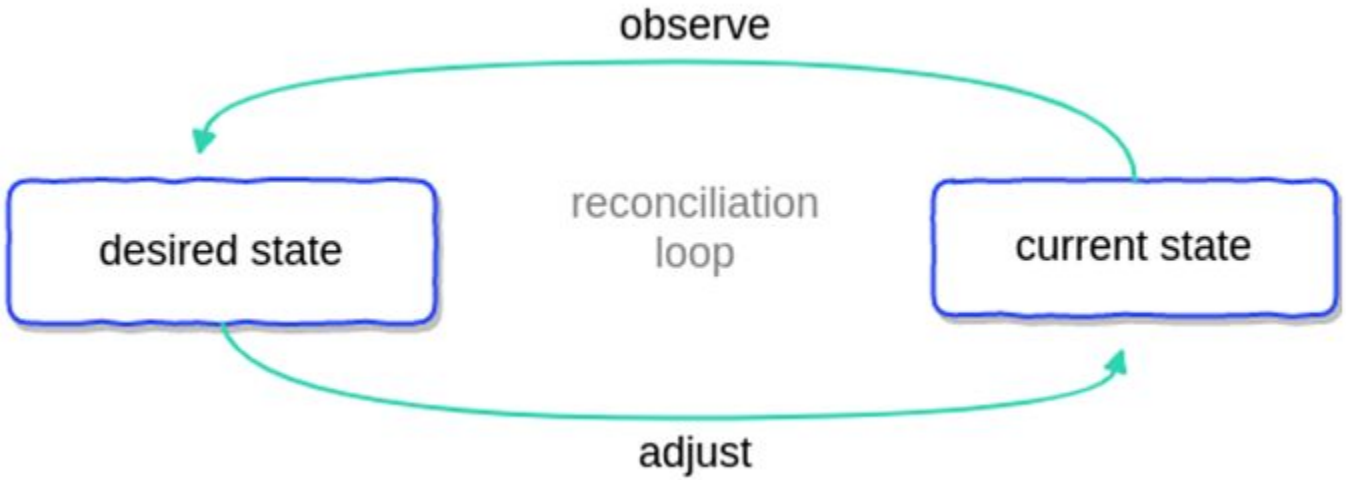
(goddady's)
https://github.com/external-secrets/kubernetes-external-secrets

https://github.com/external-secrets/external-secrets

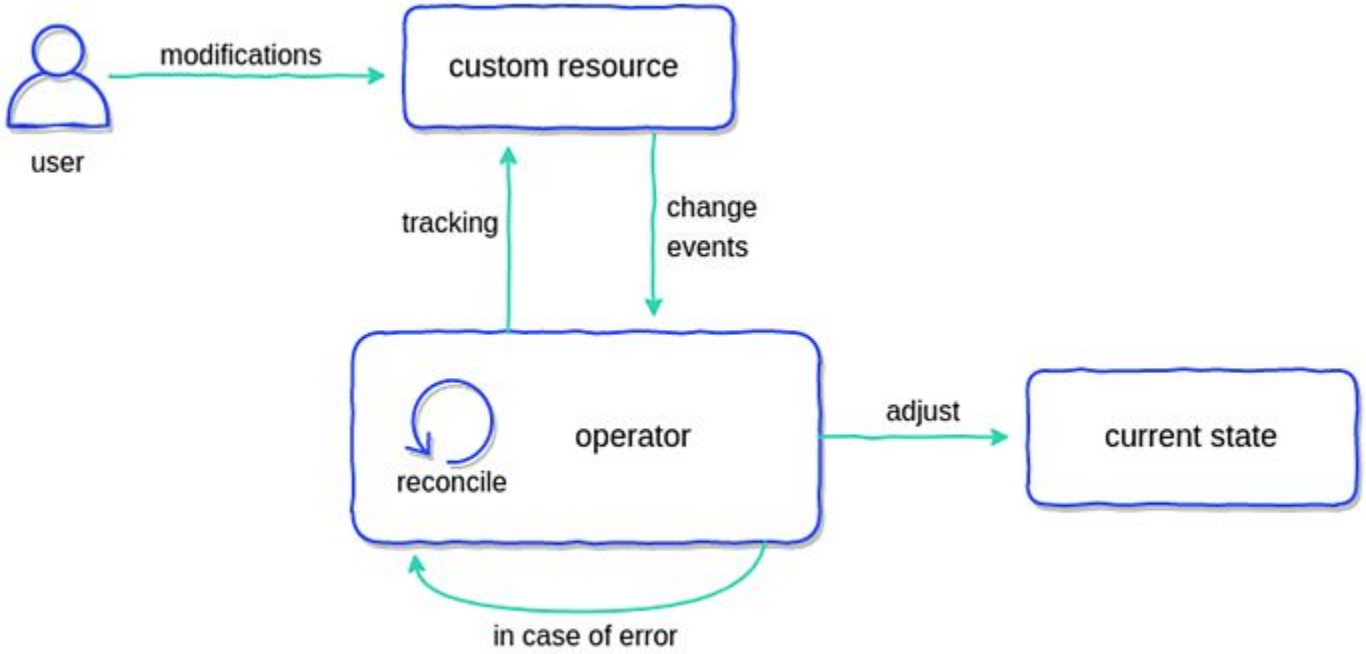
new shiny solution, everyone should migrate to this one ✨

How it Works

The Operator / Controller Pattern



The Operator / Controller Pattern



Our CRDs / APIs

SecretStore

ClusterSecretStore

ExternalSecret

ClusterExternalSecret

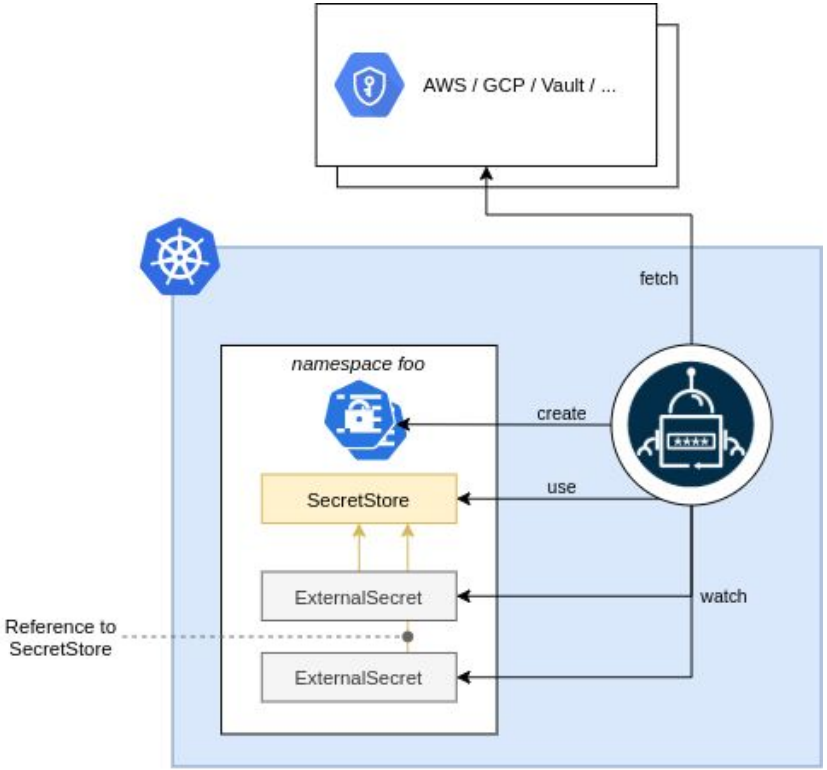
SecretStore/ClusterSecretStore

```yaml
apiVersion: external-secrets.io/v1beta1
kind: SecretStore
metadata:
  name: example
  namespace: example-ns
spec:

  # Used to select the correct ESO controller (think: ingress.ingressClassName)
  # The ESO controller is instantiated with a specific controller name
  # and filters ES based on this property
  # Optional
  controller: dev

  # You can specify retry settings for the http connection
  # these fields allow you to set a maxRetries before failure, and
  # an interval between the retries.
  # Current supported providers: IBM
  retrySettings:
    maxRetries: 5
    retryInterval: "10s"

  # provider field contains the configuration to access the provider
  # which contains the secret exactly one provider must be configured.
  provider:

```

SecretStore/ClusterSecretStore

```yaml
vault:
  server: "https://vault.acme.org"
  # Path is the mount path of the Vault KV backend endpoint
  path: "secret"
  # Version is the Vault KV secret engine version.
  # This can be either "v1" or "v2", defaults to "v2"
  version: "v2"
  # vault enterprise namespace: https://www.vaultproject.io/docs/enterprise/namespaces
  namespace: "a-team"
  # base64 encoded string of certificate
  caBundle: "..."
  # Instead of caBundle you can also specify a caProvider
  # this will retrieve the cert from a Secret or ConfigMap
  caProvider:
    # Can be Secret or ConfigMap
    type: "Secret"
    name: "my-cert-secret"
    key: "cert-key"
```

SecretStore/ClusterSecretStore



ExternalSecret / ClusterExternalSecret

```yaml
apiVersion: external-secrets.io/v1beta1
kind: ExternalSecret
metadata:
  name: "hello-world"

  # labels and annotations are copied over to the
  # secret that will be created
  labels:
    acme.org/owned-by: "q-team"
  annotations:
    acme.org/sha: 1234

spec:

  # SecretStoreRef defines which SecretStore to use when fetching the secret data
  secretStoreRef:
    name: secret-store-name
    kind: SecretStore  # or ClusterSecretStore

  # RefreshInterval is the amount of time before the values reading again from the
  SecretStore provider
  # Valid time units are "ns", "us" (or "µs"), "ms", "s", "m", "h" (from time.ParseDuration)
  # May be set to zero to fetch and create it once
  refreshInterval: "1h"

```

ExternalSecret / ClusterExternalSecret

```
27  target:
28
29    # The secret name of the resource
30    # Defaults to .metadata.name of the ExternalSecret
31    # It is immutable
32    name: my-secret
33
34    # Enum with values: 'Owner', 'Merge', or 'None'
35    # Default value of 'Owner'
36    # Owner creates the secret and sets .metadata.ownerReferences of the resource
37    # Merge does not create the secret, but merges in the data fields to the secret
38    # None does not create a secret (future use with injector)
39    creationPolicy: 'Merge'
40
41    # DeletionPolicy defines how/when to delete the Secret in Kubernetes
42    # if the provider secret gets deleted.
43    # Valid values are Delete, Merge, Retain
44    deletionPolicy: "Retain"
45
46    # Specify a blueprint for the resulting Kind=Secret
47    template:
48      type: kubernetes.io/dockerconfigjson # or TLS...
49
50      metadata:
51        annotations: {}
52        labels: {}
53
```

ExternalSecret / ClusterExternalSecret

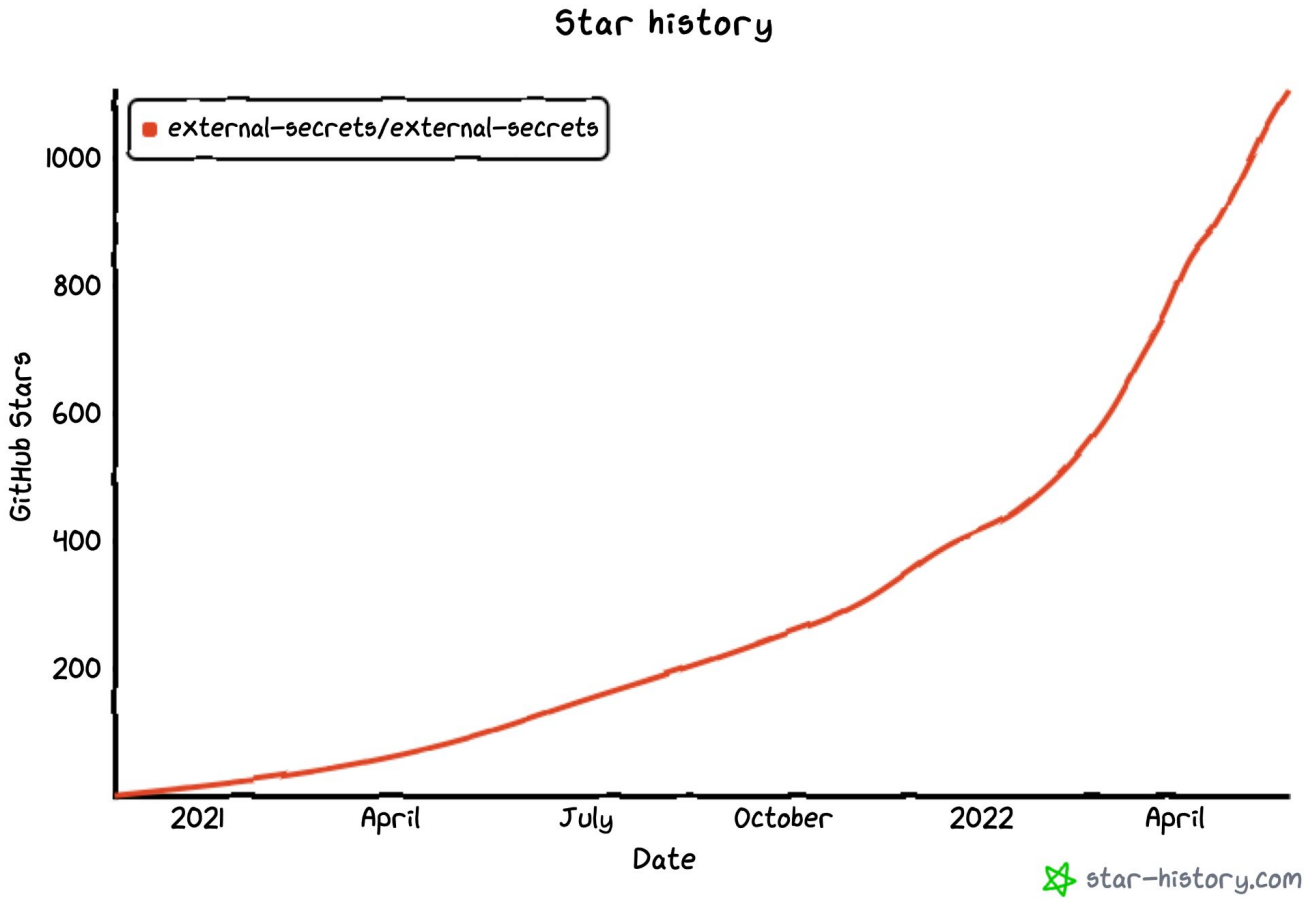
```
54      # Use inline templates to construct your desired config file that contains your secret
55      data:
56        config.yml: |
57          endpoints:
58          - https://{{ .data.user }}:{{ .data.password }}@api.exmaple.com
59
60      # Uses an existing template from configmap
61      # Secret is fetched, merged and templated within the referenced configMap data
62      # It does not update the configmap, it creates a secret with: data["alertmanager.yml"]
        = ...result...
63      templateFrom:
64      - configMap:
65          name: alertmanager
66          items:
67          - key: alertmanager.yaml
68
69    # Data defines the connection between the Kubernetes Secret keys and the Provider data
70    data:
71      - secretKey: secret-key-to-be-managed
72        remoteRef:
73          key: provider-key
74          version: provider-key-version
75          property: provider-key-property
76
```

The current state

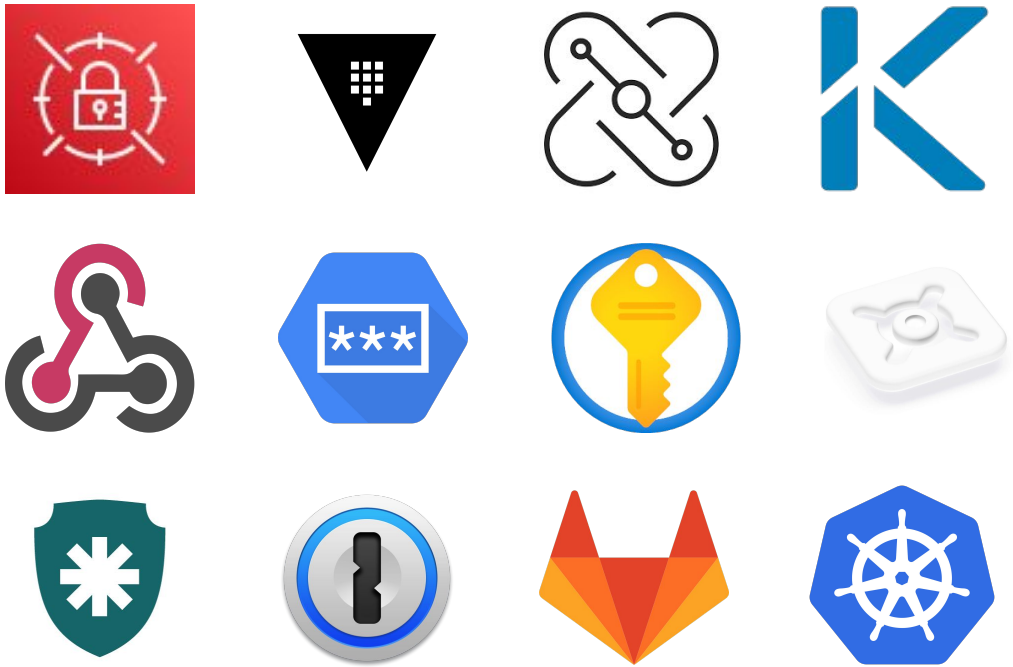
What is happening with ESO

Status

- < 0.2.0:
  - AWS Parameter Store
  - AWS Secret Manager
  - Hashicorp Vault
  - Templating functionalities
  - GCP Secret manager
  - Azure Key Vault
- 0.3.0:
  - Introduction of Creation Policy
  - Huge addition to docs
- 0.4.0:
  - Reconciler for SecretStores
  - Reporter for Kubernetes events
  - Hashicorp Vault and GCP Secret Manager promoted to stable
- 0.4.3:
  - New Templating engine
  - Promoted AWS providers to stable
  - Promoted Azure provider to beta
- 0.5.0 (Current):
  - CRD promotion to beta
- 0.x.x
  - Ability to write out to Provider


Star history
external-secrets/external-secrets
1000
800
600
400
200
GitHub Stars
2021
April
July
October
2022
April
Date
star-history.com

Supported Providers



- AWS
- Azure
- Google
- IBM
- Akeyless
- HashiCorp Vault
- Yandex
- Gitlab
- Oracle
- 1Password
- Webhooks
- Kubernetes
- Senhasegura

DEMO!

You can reach us and contribute

https://github.com/external-secrets/external-secrets

https://external-secrets.io/

https://kubernetes.slack.com/messages/external-secrets

https://blog.container-solutions.com/the-birth-of-the-external-secrets-community

Container Solutions is Hiring

Current Remote Positions:

- Cloud Native Engineer
- Cloud Native Architect
- Resource Manager

https://www.container-solutions.com/careers

Thank You

Cloud Native Computing
Johannesburg