ESO Primer
Why, When, What, and How

$>whoami

Gustavo Carvalho

CTO @ External Secrets Inc.

Maintainer @external-secrets

@gusfcarvalho

@gusfcarvalho

Q/A Slido

Agenda

- **Why** do we need to care about Secrets Management?
- **What** is External Secrets Operator
- **When** can I use External Secrets Operator
- **How** can I use External Secrets Operator

Q/A Slido

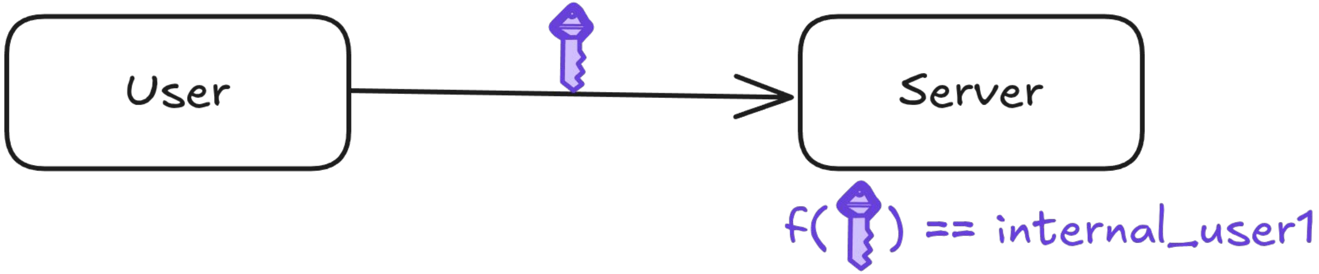
Why

Do we need to care about Secrets Management?

Q/A Slido

Why do we need to care about Secrets Management?

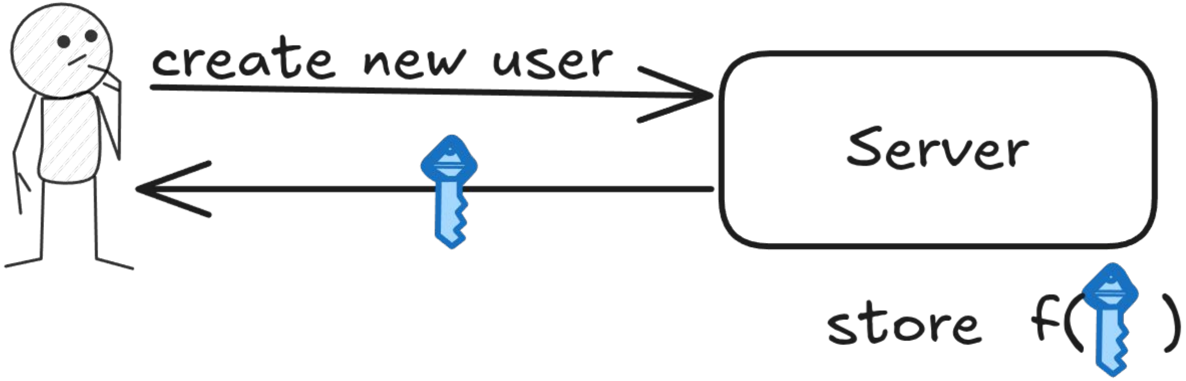
- **Secrets** (API keys, DB Credentials, etc.) are how a given system can connect to another
- It serves as the **identifying mechanism** on most systems



Q/A Slido

Why do we need to care about Secrets Management?

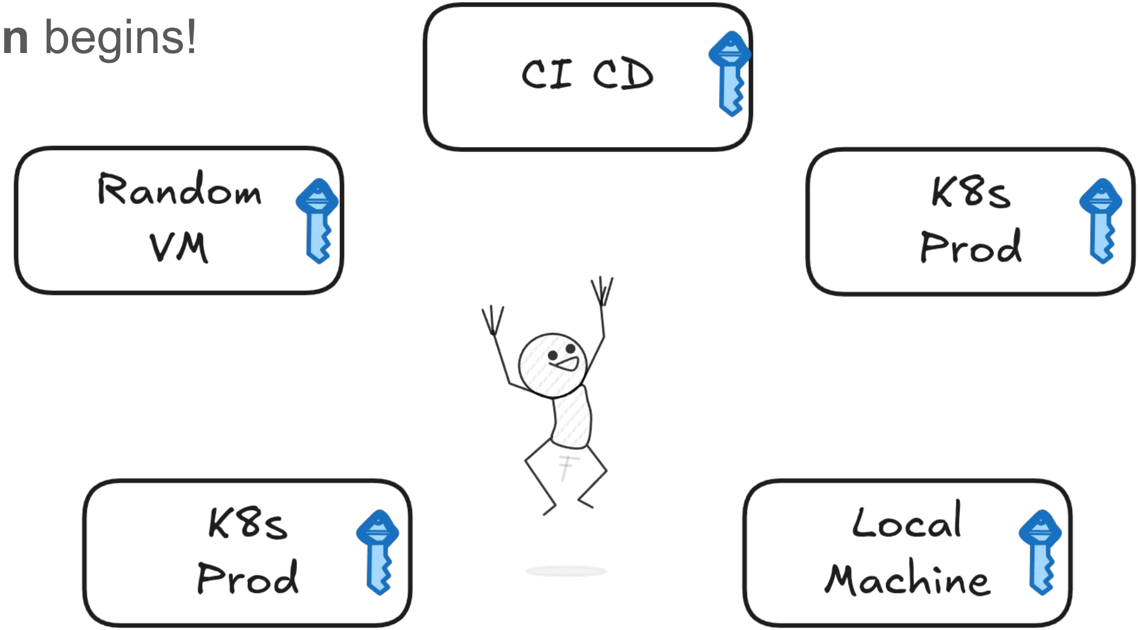
- The Server **does not store** the sensitive value.
- Authentication is done via an **indirect mechanism** (hashes, SCRAM, …)



Q/A Slido

Why do we need to care about Secrets Management?

- Distribution of the key becomes a **responsibility of the user** (typically the Server owner)
- Then, **fun** begins!



Q/A Slido

Why do we need to care about Secrets Management?

- This is why **IAM teams care** about it.
  - Identifying information such as Passwords should be private to the individual
  - MFAs are used to enforce people have a protection in case of a breach

- For Application Access, we’ve only got ways to remediate*:
  - Scope down who can access sensitive information
  - Store sensitive information on a ‘safe place’ (e.g. Open Bao)
  - Have short lived credentials as much as possible
  - Encrypt them when stored so unauthorized users cannot see them.

Q/A Slido

* my honest, humble, biased opinion. Take into account the opinion of people with more expertise into the subject if quoting this :)

Why do we need to care about Secrets Management?

- On Kubernetes:
  - Traditional Access Management (k8s RBAC) has a very bad UX to apply the principle of Least Privilege.
  - There is no verb to only show Secrets keys only (`list` and `get` both give you value access);
  - Needing to store information on Kubernetes is yet **another copy** of the sensitive data in the first place - should be avoided at all costs.

- Some Fallacies:
  - K8s Secrets are bad because they’re base64 encoded and that’s not encryption.
  - K8s Secrets are bad because they’re decrypted at rest.

Q/A Slido

What

is External Secrets Operator ?

Q/A Slido

What is External Secrets Operator?

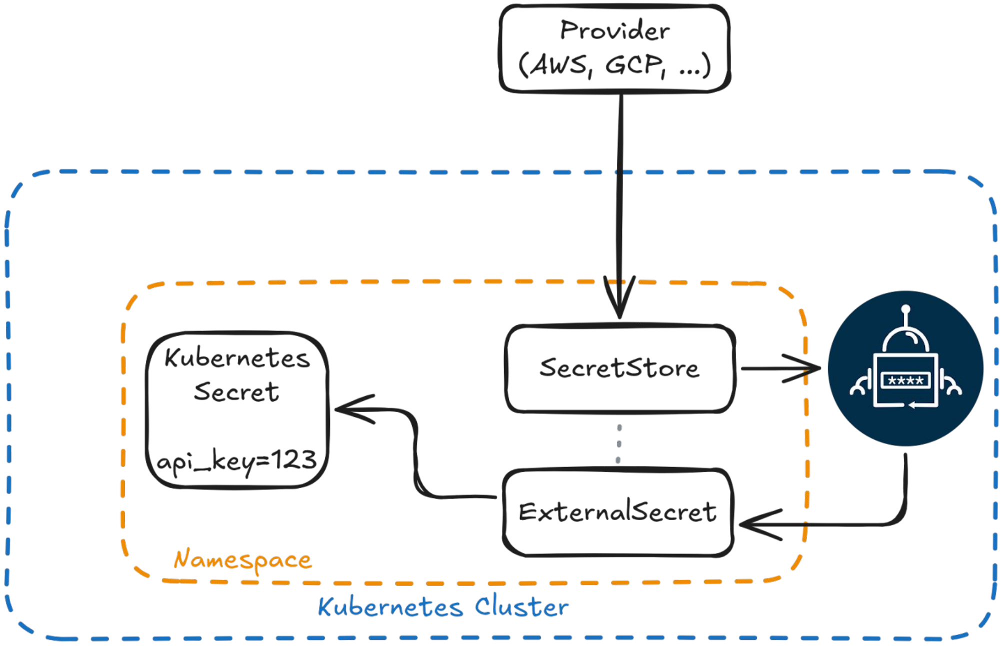
- **Kubernetes Controllers + Custom Resource Definitions** to Synchronize **External** Sources with Kubernetes **Secrets**.

- Makes possible a zero-access policy for developers on Kubernetes Secrets

Q/A Slido

What is External Secrets Operator?

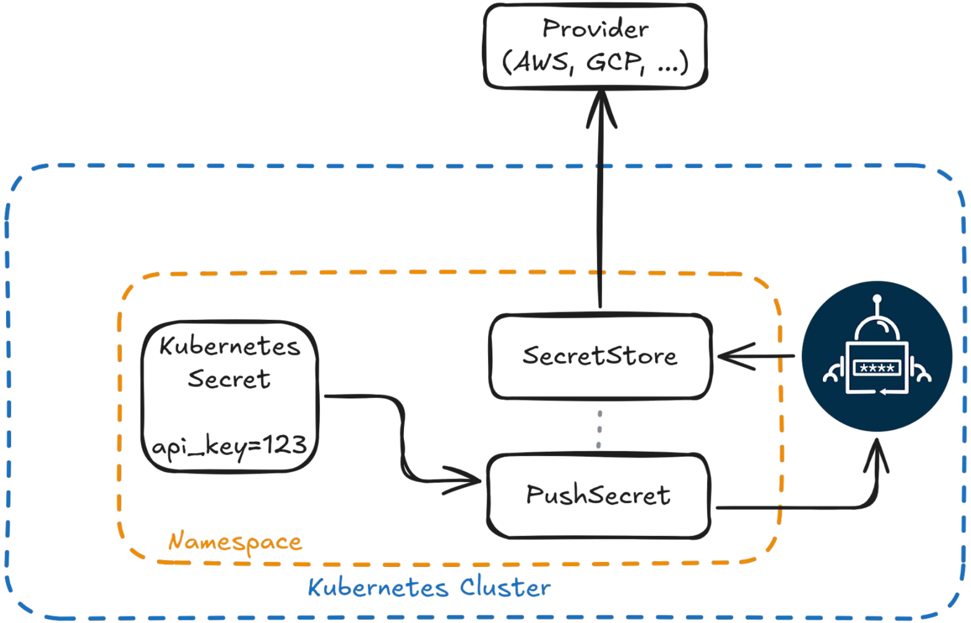
- **Pull to** the Cluster



Q/A Slido

What is External Secrets Operator?

- **Push from** the Cluster



Q/A Slido

When

should I use External Secrets Operator ?

Q/A Slido

When Should I use ESO?

Q/A Slido

When Should I use ESO?

- You have a **Secrets Management tool available to use**;
- Your Cloud Native tools are **expecting a Kubernetes Secret** to work
  - Ingress Controllers;
  - Most of Open Source tooling helm charts
  - App Deployments expecting a Secret ref (`env.valueFrom.secretKeyRef`)
- **Gitops used** to install applications (ArgoCD, Flux, Fleet, Sveltos, ...)
- You need to **push from Kubernetes** into your Secrets Manager

Q/A Slido

When Should I **NOT** use ESO?

- **Compliance requirement** to not use Kubernetes Secrets

- You **do not have** a Secrets Management tool;

- **Your users cannot use** the Secrets Management tool (directly or indirectly).

Q/A Slido

How

can I use External Secrets Operator ?

Q/A Slido

How can I use ESO?

- Pulling keys multiple stores
- Managing multiple entries: `dataFrom`
- Decoding & Metadata Policies
- Generators and Refresh Policies
- `spec.target` and Templates
- Creation & Deletion Policies
- Pushing to downstream clusters

Q/A Slido

Pulling keys from multiple stores

```
spec:
  data:
  - secretKey:  my-key
    remoteRef:
      key: my-key-in-store-1
    sourceRef:
      storeRef:
        name:  store-1
        kind: SecretStore
  - secretKey:  second-key
    remoteRef:
      key: my-key-in-store-2
    sourceRef:
     storeRef:
        name:  store-2
        kind: SecretStore
```

Q/A Slido

Managing Multiple entries: dataFrom

```
spec:
  dataFrom:
  - extract:  # all keys from a JSON object
      key: remote-key-in-store
    sourceRef: {}
  - find:
      name:
        regexp: .* # all secret  keys in provider
  - find:
      tags:
        foo: bar # All secrets matching given tags
    rewrite:
     - regexp:
        source: “foo-(.*)”
        target: “bar-$1”
```

Q/A Slido

MetadataPolicy & DecodingStrategy

```
spec:
  data:
  - secretKey: metadata
    remoteRef:
      key: my-remote-secret
      metadataPolicy: Fetch
  dataFrom:
  - extract:
      key: my-remote-secret
      metadataPolicy: Fetch
      decodingStrategy: Base64
  - find:
     tags:
        encoding: base64
     decodingStrategy: Base64
```

Q/A Slido

Generators and RefreshPolicy

```
spec:
 refreshPolicy: Periodic/OnChange/CreatedOnce
 refreshInterval: 1h
 dataFrom:
 - sourceRef:
   generatorRef:
    apiVersion: generators.external-secrets.io/v1alpha1
    kind: Password
    name: my-password-generator
  rewrite:
  - regexp:
    source: “password”
    target: “db-password”
```

Q/A Slido

`spec.target` and Templates

```
spec:
  target:
    name: custom-secret-name
    template:
      metadata:
        annotations:
          custom-annotations:  yes-please
      data:
        combined-key-values: “{{.key1}}:{{ .key2 | toUpper }}”
  data: {...}
  dataFrom: {...}
```

Q/A Slido

`spec.target` and Templates

```
spec:
  target:
    name: custom-secret-name
    template:
      templateFrom:
      - target: Data/Annotations/Labels
        configMap:
          name: my-custom-configmap
          items:
           - key: my-cm-key
             templateAs: Values/KeysAndValues
      - target: Data
        literal: |
           “key-1-{{ .key1 }}”: “key-2-{{ .key2 }}”
  data: {...}
  dataFrom: {...}
```

Q/A Slido

creationPolicy and deletionPolicy

```
spec:
  target:
    name: custom-secret-name
    type: kubernetes.io/dockerconfigjson
    creationPolicy: Owner/Merge/Orphan/None
    deletionPolicy: Retain/Delete/Merge
  data: {...}
  dataFrom: {...}
```

Q/A Slido

Questions?

Q/A Slido