Open Infrastructure FOUNDATION

INDONESIA OpenInfra Days

Make Kubernetes **Secret Object**

More Secret Using SealedSecret

Muhammad Fauzi Islami

IT Consultant at **PT Inovasi Informatika Indonesia**

Bogor, August 21, 2021

About Me

Muhammad Fauzi Islami

IT Consultant at PT Inovasi Informatika Indonesia

INOVASI INFORMATIKA INDONESIA

@fauzislami

Muhammad Fauzi Islami

islamifauzi@gmail.com

Agenda

- Secret Object in K8S
- The Concern of Managing Secret Manifest
- Secure K8S Secret Object Using Sealed Secret
- Quick Demo

Secret Object In K8S

It hides all sensitive data.

Definition of Secret Object

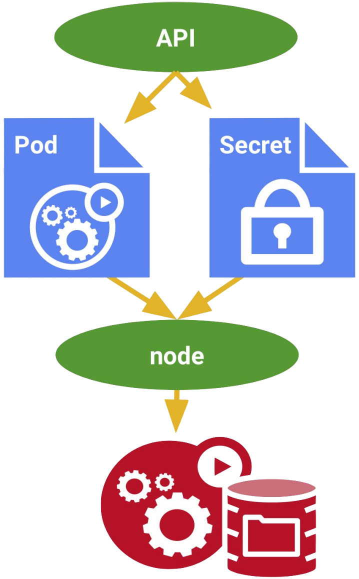
“ A **Secret** is an object that contains a small amount of sensitive data such as a password, certificate, a token, or a key. “

Source : https://kubernetes.io/docs/concepts/configuration/secret/

How Secret works

Decoupling any confidential data from application code.

Besides, it will be reusable for other resources.



INDONESIA
OpenInfra Days

The Concern Of Managing Secret Manifest

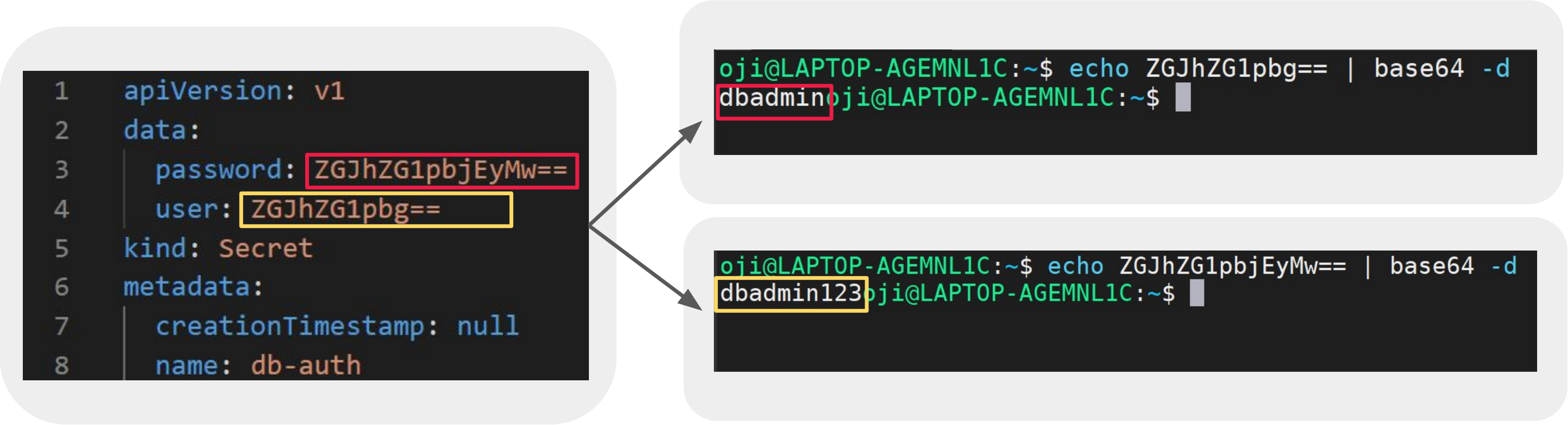
Secret Object is not as secret as it supposed to be.

A look at current “Infrastructure as Code” Trends

Managing kubernetes manifest on Git or any other SCM is a common task nowadays. We manage all of the configurations of K8S cluster in a shared git repository, run automated tests, and automatically deploy changes to the cluster from master . . .

except ***Secret***.

It contains ~~encrypted~~ encoded data



```
apiVersion: v1
data:
  password: ZGJhZG1pbjEyMw==
  user: ZGJhZG1pbg==
kind: Secret
metadata:
  creationTimestamp: null
  name: db-auth
```

```
oji@LAPTOP-AGEMNL1C:~$ echo ZGJhZG1pbg== | base64 -d
dbadminoji@LAPTOP-AGEMNL1C:~$ 
```

```
oji@LAPTOP-AGEMNL1C:~$ echo ZGJhZG1pbjEyMw== | base64 -d
dbadmin123oji@LAPTOP-AGEMNL1C:~$ 
```

Kubernetes secrets object stores sensitive information as a **base64 string**. Anyone can decode the base64 string to get the original data from the Secret manifest.

Some points are needed for this case

- ◎ Secure the secret object
- ◎ Possible to store in shared git repository safely
- ◎ Secret object remains reusable for other resources
- ◎ Secret object should work as usual in cluster

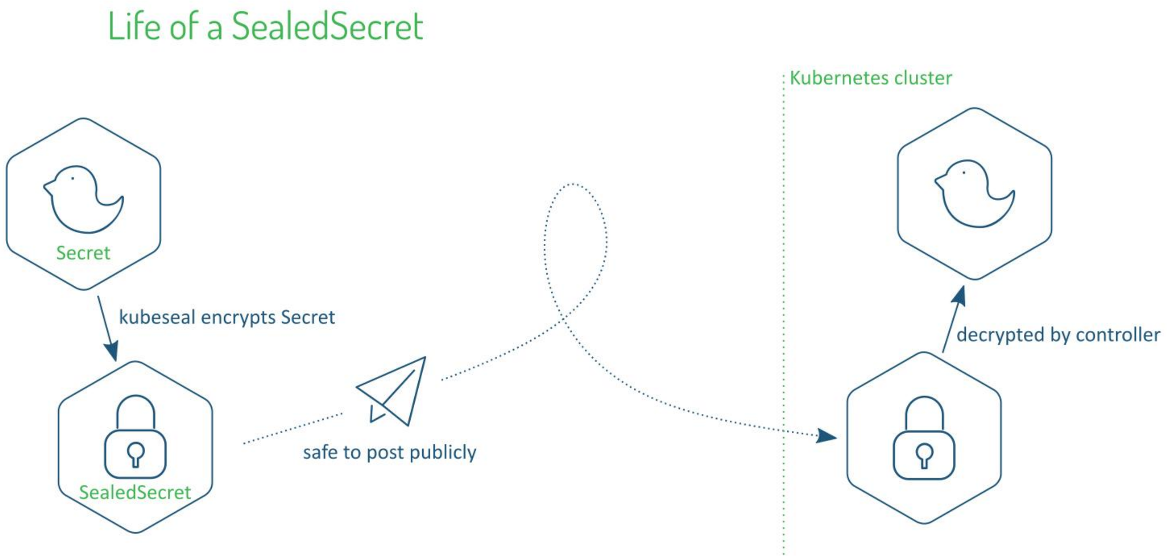
Secure K8S Secret Object Using SealedSecret

Encrypt you Secret into a SealedSecret.

SealedSecrets

SealedSecrets is a kubernetes controller that is developed by bitnami-labs which has a purpose to make a ***"one-way" encrypted Secret*** that can be created by anyone, but can *only* be decrypted by the controller running in the target cluster. The Sealed Secret is safe to share publicly, including to public git repository.

A life of a SealedSecret



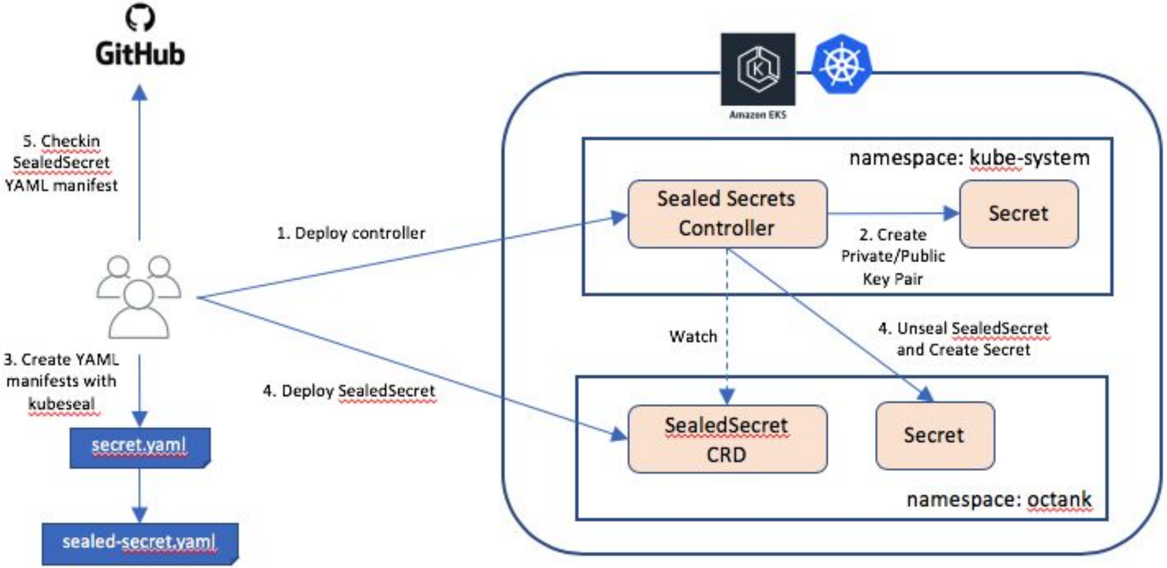
SealedSecret : Protecting your passwords before the reach Kubernetes
https://engineering.bitnami.com/articles/sealed-secrets.html

SealedSecret Component

The **SealedSecret** implementation consists of two components :

- A controller that runs in-cluster, and implements a new SealedSecret Kubernetes API object via the "third party resource" mechanism.
- A `kubeseal` command line tool that encrypts a regular Kubernetes Secret object (as YAML or JSON) into a SealedSecret.

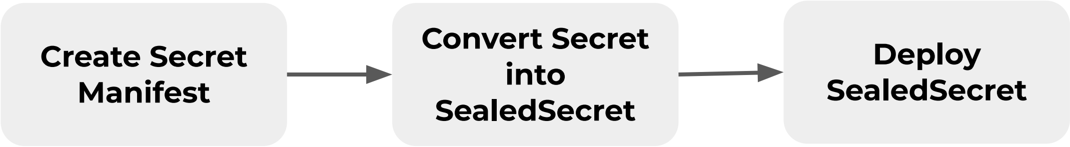
How SealedSecret works


GitHub
5. Checkin SealedSecret YAML manifest
1. Deploy controller
3. Create YAML manifests with kubeseal
secret.yaml
sealed-secret.yaml
4. Deploy SealedSecret
Amazon EKS
namespace: kube-system
Sealed Secrets Controller
Secret
2. Create Private/Public Key Pair
Watch
4. Unseal SealedSecret and Create Secret
SealedSecret CRD
Secret
namespace: octank

Managing secrets deployment in Kubernetes using Sealed Secrets
https://aws.amazon.com/blogs/opensource/managing-secrets-deployment-in-kubernetes-using-sealed-secrets/

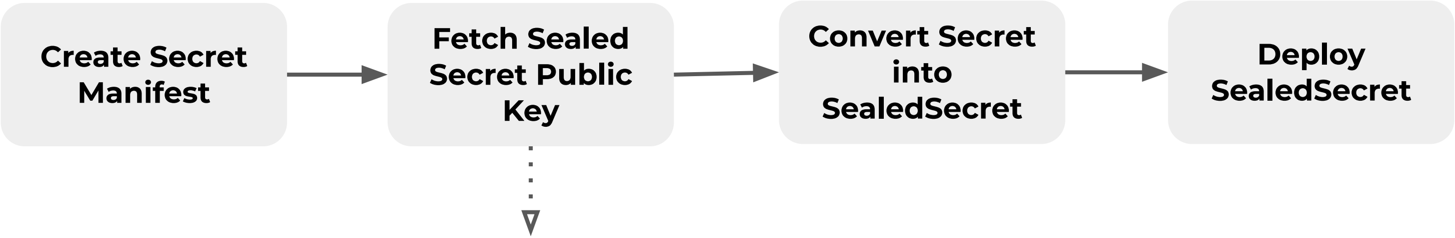
2 ways in using SealedSecret

- Have access to cluster



Create Secret Manifest → Convert Secret into SealedSecret → Deploy SealedSecret

- Not have any access to cluster



Create Secret Manifest → Fetch Sealed Secret Public Key → Convert Secret into SealedSecret → Deploy SealedSecret

Only if public key doesn’t exist

SealedSecret Scopes

SealedSecret has 3 scopes in order to make boundaries and certain rules.

There are 3 scopes we can create our SealedSecrets with :

- **strict** (default) : In this case, we need to seal our Secret considering the name and the namespace. We can't change the name and the namespaces of SealedSecret once we have created it. If we try to do that, we get a decryption error.
- ***Namespace-wide*** : This scope allows us to freely rename the SealedSecret within the namespace for which we have sealed the Secret.
- ***Cluster-wide*** : This scope allows us to freely move the Secret to any namespace and give it any name we wish.

Quick Demo

Let’s give it a try !

Installations

Cluster Side

Apply Manifest

```
kubectl apply -f https://github.com/bitnami-labs/sealed-secrets/releases/download/v0.16.0/controller.yaml
```

OR

Using Helm

```
$ helm repo add sealed-secrets https://bitnami-labs.github.io/sealed-secrets
$ helm repo update
$ helm install -n kube-system demo-sealed-secrets sealed-secrets/sealed-secrets
```

Installations

Client Side

Kubeseal Installation

```
$ wget https://github.com/bitnami-labs/sealed-secrets/releases/download/v0.16.0/kubeseal-linux-amd64 -O kubeseal

$ sudo install -m 755 kubeseal /usr/local/bin/kubeseal
```

Secret Object Creation

Create Secret #1

```
$ kubectl create secret generic dummy-auth \
  --from-literal=user=admin \
  --from-literal=password=admin123 \
  --dry-run=client -o yaml > dummy-auth.yaml
```

Create Secret #2

```
$ kubectl create secret generic db-auth \
  --from-literal=user=dbadmin \
  --from-literal=password=dbadmin123 \
  --dry-run=client -o yaml > db-auth.yaml
```

SealedSecret Creation (Need access to the cluster)

Encrypt secret using SealedSecret

```
$ kubeseal --format=yaml --controller-namespace=kube-system \
  --controller-name=sealed-secrets-controller < dummy-auth.yaml \
  > sealed-dummy-auth.yaml
```

Apply SealedSecret Manifest

```
$ kubectl apply -f sealed-dummy-auth.yaml
$ kubectl get secret
$ kubectl get sealedsecret
```

SealedSecret Creation (No need access to the cluster)

Fetch public key (Only if public key doesn’t exist)

```
$ kubeseal --fetch-cert --controller-namespace=kube-system \
  --controller-name=sealed-secrets-controller > sealedsecret-pubkey.pem
```

Encrypt secret using SealedSecret

```
$ kubeseal --format=yaml \
  --cert=sealedsecret-pubkey.pem < db-auth.yaml > sealed-db-auth.yaml

$ kubectl apply -f sealed-db-auth.yaml
```

Compare strict with cluster-wide

Change SealedSecret name (Strict)

```
Change → Name: db-auth ---> Name: db-auth-2

$ kubectl apply -f sealed-db-auth.yaml
```

Change SealedSecret name (Cluster-Wide)

```
$ kubeseal --scope=cluster-wide \
  --format=yaml --cert=sealedsecret-pubkey.pem < \
  db-auth.yaml > sealed-db-auth.yaml
```

```
Change → Name: db-auth ---> Name: db-auth-2

$ kubectl apply -f sealed-db-auth.yaml
```

Sponsored by:

Open Infrastructure
FOUNDATION

NVIDIA®

boer
technology

Biznet
GioCloud

EasyStack
open cloud computing

BRI

Open Networking
Indonesia

imtek

INDOCENTER

Hosted by:

OPENSTACK USER GROUP INDONESIA

OpenStack Indonesia
Indonesia OpenStack Foundation Community
www.openstack.id

Community Partners:

sdX Indonesia

openSUSE ID

KLAS

gimpscape ID

CLOUD NATIVE
INDONESIA

Adaptive Network
Laboratory

NETWORK
EVOLUTION

Open Infrastructure FOUNDATION

INDONESIA OpenInfra Days

Thanks!

Do you have any questions?

youremail@google.com
+62 9903113781
yourcompany.com

Platinum sponsor :

NVIDIA

Gold sponsor :

boer technology

Biznet GioCloud

EasyStack

Silver sponsor :

BRI

Custom sponsor :

Open Networking Indonesia

imtek

INDOCENTER