Kubernetes secrets encryption using sops with Flux2

[**Eneko Pérez**](https://se.linkedin.com/in/enekoperez?trk=pulse-article_main-author-card)

**Eneko Pérez**

Cloud - Container Platform Reliability Engineer at Volvo Cars

Published Jul 20, 2021

[**+ Follow**](https://www.linkedin.com/signup/cold-join?session_redirect=https%3A%2F%2Fwww%2Elinkedin%2Ecom%2Fpulse%2Fkubernetes-secrets-encryption-using-sops-flux2-eneko-p%25C3%25A9rez&trk=pulse-article_main-author-card__cta-button)

I'm building my Kubernetes cluster running on Raspberry Pi using Flux2 (GitOps) to automate the deployment and creation of namespaces, secrets, applications and other objects like Helm releases. I have configured Flux2 to synchronise from a GitHub repository to deploy whatever I need automatically. Secrets, however, are a sensitive object in Kubernetes.

Secrets in Kubernetes are just base64-encoded, which means that with a simple command all the sensitive data can be revealed. As a rule of thumb, we should never publish passwords, secrets, credentials, certificates, keys, etc., in a (public) repository. I think it's common sense but still worths a mention. However, if we want Flux to deploy our secrets as well, they need to be in the repository. So, let's encrypt them!

SOPS comes from Secrets OPerationS. It's an open source Mozilla project available here:

Let's see how this works and how we can integrate it with Flux2.

* Create a local GPG key.

# Install GPG first. Tip: brew on Mac OS

# Note: Do not to specify a password for the key

gpg --full-generate-key

# List the key we just created

gpg --list-secret-keys

* Create a config file for SOPS

cat <<EOF > .sops.yaml

---

creation\_rules:

- encrypted\_regex: '(data|stringData)$'

pgp: >-

YOUR\_GPG\_KEY\_HERE

EOF

This simple configuration tells SOPS to encrypt only what is inside 'data' and 'stringData' keys inside a yaml file.

* Create a sample secret that we can use to test encryption

cat <<EOF > secret.yaml

apiVersion: v1

kind: Secret

metadata:

name: secret-test

namespace: test-namespace

stringData:

password: Abc.123

EOF

* Now we can encrypt it using SOPS

# Encryption in place

sops --encrypt --in-place secret.yaml

Now, if we take a look at the secret, it should look similar to this:



* Before digging into Flux2, we could apply this secret manually to Kubernetes but we have to decrypt it first:

sops --decrypt secret.yaml | kubectl apply -f -

* But I use Flux2 so I don't have to deploy anything manually, so let's configure Flux to decrypt these secrets:

# We need to create a secret with our GPG key in Kubernetes manually (or while bootstrapping the cluster) for Flux to be able to decrypt

gpg --export-secret-keys --armor YOUR\_GPG\_KEY\_HERE | kubectl create secret generic sops-gpg --namespace=flux-system --from-file=sops.asc=/dev/stdin

* Now, we have to configure Flux2 to use sops provider

# vim gotk-sync.yaml and add the following lines under spec:

decryption:

provider: sops

secretRef:

name: sops-gpg

* We are ready to push the changes (secret, sops config file and gotk-sync) to our repository and wait for Flux2 to reconcile and deploy.