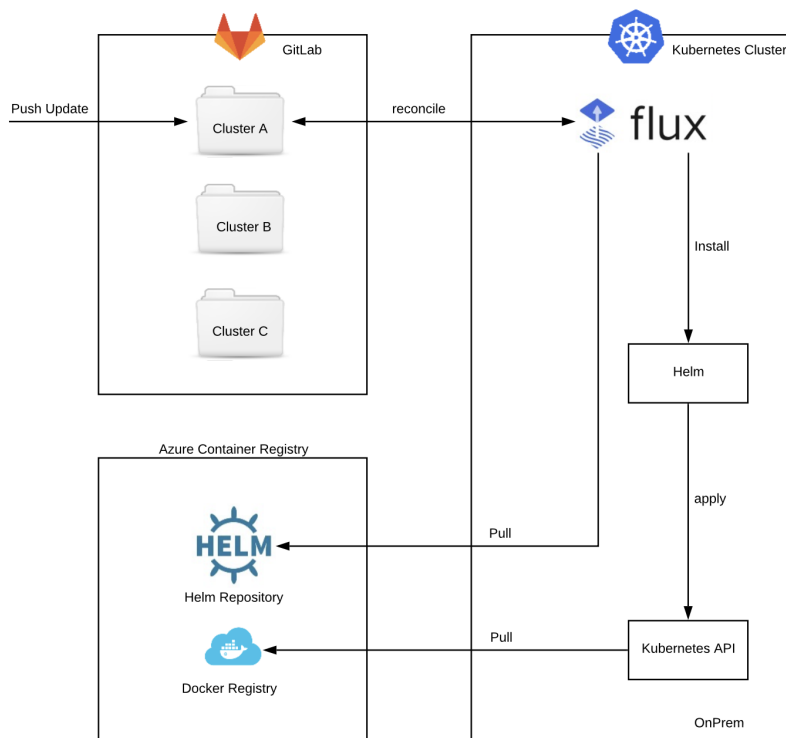


Deploying Helm Charts with Flux2

- [Overview](#)
- [Install Flux](#)
- [Generate a User Token in GitHub](#)
- [Bootstrap the Repository](#)
- [Create Helm Repository](#)
- [Create a Helm Release](#)
- [Helpfull Commands](#)
- [Debugging a Failed Helm Release](#)
- [References](#)

Overview



Install Flux

See <https://fluxcd.io/docs/installation/#install-the-flux-cli>

Linux/Mac

```
$ curl -s https://fluxcd.io/install.sh | sudo bash
or
$ curl -s https://fluxcd.io/install.sh | bash
```

Generate a User Token in GitHub

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>

Export User Token as Environment Variables

```
$ export GITHUB_TOKEN=xxxx
$ export GITHUB_USER=johnmehan
```

Bootstrap the Repository

Perform pre-check to make sure we are able to install flux.

```
$ flux check --pre

checking prerequisites
flux 0.19.1 <0.20.0 (new version is available, please upgrade)
Kubernetes 1.21.2 >=1.19.0-0
prerequisites checks passed
```

With Github

```
$ flux bootstrap github --owner=$GITHUB_USER --repository=<REPO_NAME> --branch=main --path=<CLUSTER_NAME> --personal

# example
$ flux bootstrap github --owner=$GITHUB_USER --repository=ft --branch=main --path=local --personal
```

With Gitlab

```
$ flux bootstrap gitlab --owner=<GROUP> --repository=<REPO_NAME> --path=<PATH> --token-auth --hostname=<HOST> --branch=<BRANCH>

#example
$ flux bootstrap gitlab --owner=MyGroup --repository=deploy_repo --path=cluster/lab --token-auth --hostname=gitlab1.company.com --branch=master
```

Git Clone New Repo

```
$ git clone git@github.com:johnmehan/ft.git
$ cd ft
$ cd local
```

Create Helm Repository

Create secret for flux to use to talk to the helm repository

```
$ kubectl create secret generic acr --from-literal username=mehan --from-literal "password=xxx" -n flux-system
```

Create a helm repo

```
$ flux create source helm ncyd-helm-virtual --url https://artifactory-fpark1.int.net.nokia.com/artifactory/ncyd-helm-virtual --interval 1m0s --secret-ref=acr --export > helmrepo-ncyd-helm-virtual.yaml
```

View Repo

```
---
apiVersion: source.toolkit.fluxcd.io/v1beta1
kind: HelmRepository
metadata:
  name: ncyd-helm-virtual
  namespace: flux-system
spec:
  interval: 1m0s
  secretRef:
    name: acr
  url: https://artifactory-fpark1.int.net.nokia.com/artifactory/ncyd-helm-virtual
  passCredentials: true
```

Commit the newly created file.

```
$ git add <file>
$ git commit
$ git push
```

Wait a bit and then try retrieving your list of helm sources:

```
$ flux get sources helm
```

NAME	READY	MESSAGE	SUSPENDED
REVISION			
ncyd	True	Fetches revision: 732d32e292bd0def8d5782211f139523d23880a1	
732d32e292bd0def8d5782211f139523d23880a1			False

Create a Helm Release

```
$ flux create helmrelease ssh-server --source=HelmRepository/ncyd-helm-virtual --chart ssh-server --release-name ssh-server --target-namespace default --interval 5m0s --export > helmrelease-ssh-server.yaml
```

You may want to update this helm release to override certain values

```
$ vi helmrelease-ssh-server.yaml
```

```
---
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: ssh-server
  namespace: flux-system
spec:
  chart:
    spec:
      chart: ssh-server
      version: '1.0.0-1362206'
      sourceRef:
        kind: HelmRepository
        name: ncyd-helm-virtual
  interval: 5m0s
  releaseName: ssh-server
  targetNamespace: default
  values:
    some: true
```

Now, commit this file to your git and wait a little bit before executing the following command:

```
$ flux get helmreleases
```

To force a reconciliation, you can issue the following command:

```
flux reconcile ks flux-system --with-source
```

Helpfull Commands

Command	Description
flux uninstall	Remove flux installation
flux get helmreleases	List the helmreleases
flux get hr	
flux reconcile ks flux-system --with-source	Force a reconcile
flux get sources helm	View Helm Sources
kubectl describe helmrelease <HR_NAME> -n flux-system	Get details of a helm release.
kubectl logs -f -n flux-system deployment/source-controller	See logs for source-controller

Debugging a Failed Helm Release

You may some of the following commands in order to determine the failure.

```
# get list of helmreleases
$ flux get hr -A

# get details about the helmrelease
$ kubectl describe helmrelease <HR_NAME> -n flux-system

# See logs in source controller
$ kubectl logs -f -n flux-system deployment/source-controller
```

Force a reconcile

```
flux reconcile ks flux-system --with-source
```

Delete a helm release and reconcile

```
kubectl -n fluxsystem delete hr <HR_NAME>
flux reconcile ks flux-system --with-source
```

References

References	URL
Flux2 Docs	https://fluxcd.io/docs/
Supercharge your Kubernetes deployments with Flux v2 and GitHub - Introduction	https://www.youtube.com/watch?v=N6UCKF7JD7k
Get Started with Flux	https://fluxcd.io/docs/get-started/
Install the Flux CLI	https://fluxcd.io/docs/installation/#install-the-flux-cli
flux2-kustomize-helm-example	https://github.com/fluxcd/flux2-kustomize-helm-example