Présentation travaux CI/CD Apps

Démo Point de Synchro du 22/03/2024

Épisode précédent

- Remplacer Concourse et Trackbone
- Besoin de CI simple, besoin de déploiement complexe
- Solutions retenues par l'équipe Apps :
  - CI : Gitlab CI (DT-Store)
  - CD : surprise…

Épisode précédent

> *Est-ce que c'est Flux? – Ala*

Réponse :

> *En ce moment l'équipe apps travaille sur Flux – Vincent*

Donc… oui ! (grosse surprise)

Plan initial

1. POC min pour valider ces outils chez apps avant présentation :
    - S’approprier Flux et son déploiement
    - Utiliser le Gitlab DT-Store comme source de code et CI
2. Implication de tout PF pour structurer leur utilisation

Pourquoi Flux ?

- Comparaison de nombreuses solutions
- Peu de projets viables (maturité, compétences dispo, k8s natif, etc.) :
  - Flux CD
  - Argo CD

Pourquoi Flux ?

- Léger et **décentralisé** vs Argo “lourd” et centralisé (en gros)
- Appropriation simple : des objets k8s en YAML, repos git et helm
- Prise à charge de Terraform via un *controller*
- Bonus : se popularise au sein de l’entreprise

Contexte initial

- Vision “PF”
- Gestion de bout en bout des clusters
- Gestion du bootstrap
- Architecture “caascad” :
    - cluster parent -> clusters enfants

Problème !

Ce n'est plus l'optique actuelle

Contexte actuel

- Clusters et objets “cloud” fournis par CaaS-CNP
- Gestion de Flux… ??? (pour le moment eux, à voir, pinguez Vincent)
- Architecture “flat” (pour le déploiement)

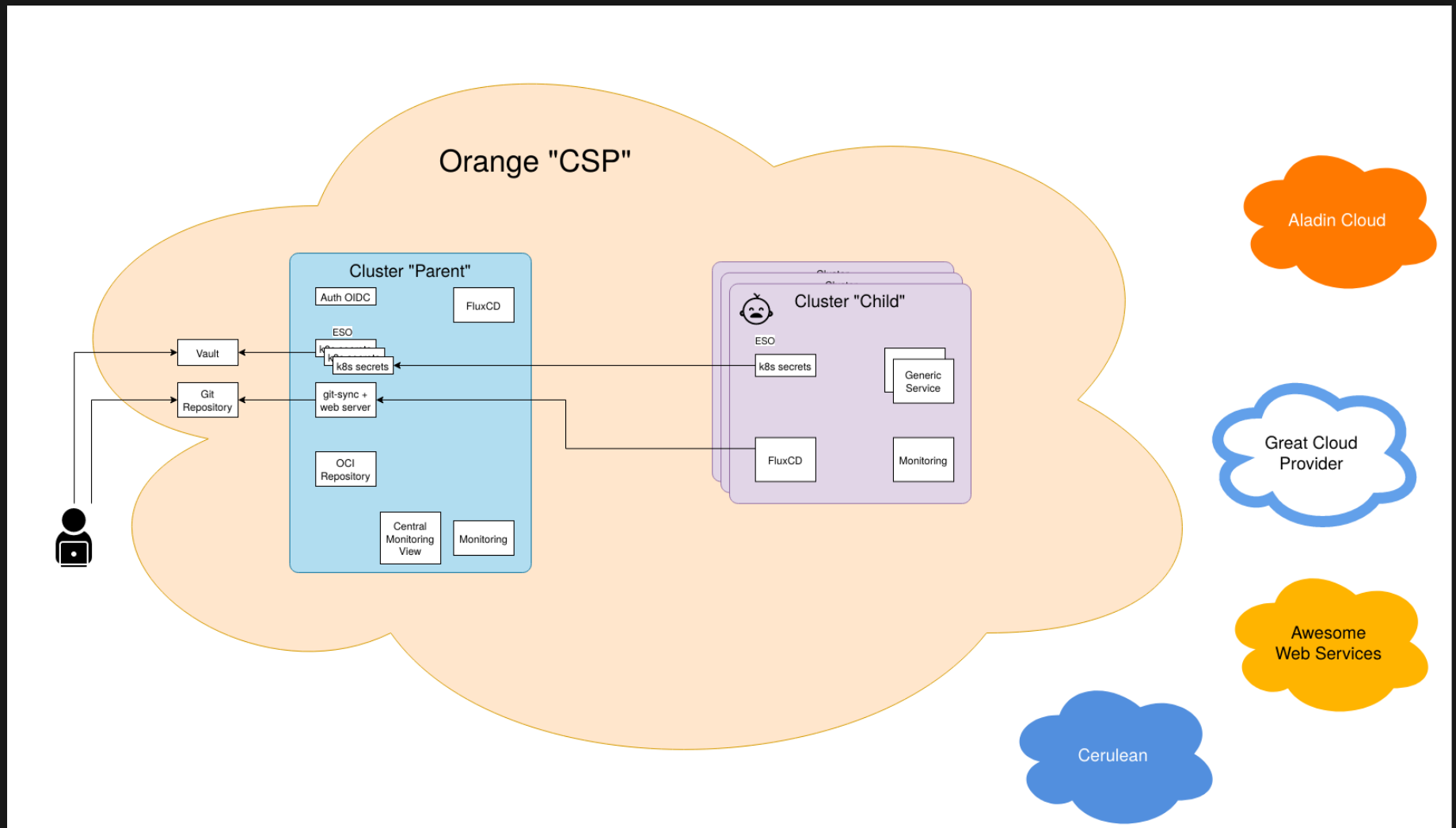
Nouveau plan™

1. Attendre les présentations de Flux à la PF (c’est fait)
2. Présentation des sujets étudiés et testés par Apps (**c’est maintenant**)
3. Ateliers et autres pour construire *new observability stack avec un nom moins chiant*

POC en tant que simple utilisateur Flux

- Déploiement d'une application sur un env CaaS-CNP
  - Utilisation de Kustomize
  - Réflexion sur l'organisation des repositories

Architecture initialement envisagée


Orange "CSP"
Cluster "Parent"
Auth OIDC
FluxCD
ESO
k8s secrets
Vault
Git Repository
git-sync + web server
OCI Repository
Central Monitoring View
Monitoring
Cluster "Child"
ESO
k8s secrets
Generic Service
FluxCD
Monitoring
Aladin Cloud
Great Cloud Provider
Awesome Web Services
Cerulean

Dépendances identifiées

- **git** : parent -> gitlab, enfants -> git-sync (sur le parent)
- **registry OCI** (containers + charts) : parent -> harbor, enfants -> cache k8s
- **secrets** : parent -> vault, enfants -> synchro secrets k8s

POC de déploiement de l’architecture via Flux

- Déploiement d’un cluster parent via un laptop opérateur
  - Run de code Terraform
- Déploiement d’un cluster enfant via le cluster parent
  - Run de code Terraform via Flux

Changement de sujet

- Pause des réflexions sur Flux en attendant d’y voir plus clair
- Mise en place d’une registry OCI (Harbor)
- Bascule sur la prise en main de la CI

POC Registry OCI

- Structuration
- Stockage de containers
- Stockage de charts
- “Robots”
- Proxy-cache (vers docker, quay.io, etc.)

POC Gitlab CI

- Création de pipelines pour les tâches communes
  - Mirroring de charts
  - Mirroring de containers
  - Packaging de charts
  - Build de containers

POC Gitlab CI & Harbor

- Présentation détaillée à venir
  - Utilisation
  - Outils validés pour le build, push…
  - etc.

Réflexions

- Le passage au “GitOps” nous amène à :
  - Faire évoluer le modèle opérationnel
  - Repenser la structure de nos repositories
- Les travaux vont devoir être répartis dans les équipes

Proposition : Organisation git

- La notion d'**Application** correspond bien à notre modèle actuel :
  - applications upstream : grafana, privx, etc.
  - "services" (= appli) cloud : aws-route53, fe-rds, etc.
  - développements "maison" : customers-manager, etc.
  - "enabler" (= operator k8s = appli)

Proposition : Environnements

- DEV : gros manque sur Caascad/NGOT. Idéalement déploiement continue automagique.
- STG : rebuild propre + process de validation pour iso PRD (ticket MES)
- PRD : pas de rebuild, copie des artefacts validés sur STG, validation (ticket MEP)

La suite…

- Répartition des tâches par les POs
- Définir les ateliers