

CI/CD AVEC AWX ET PODMAN

Un retour d'expérience

QUI SUIS-JE ?

James Regis

RHCA #100-110-133

Opensource

+

Robotique

=

Automatisation

PLAN

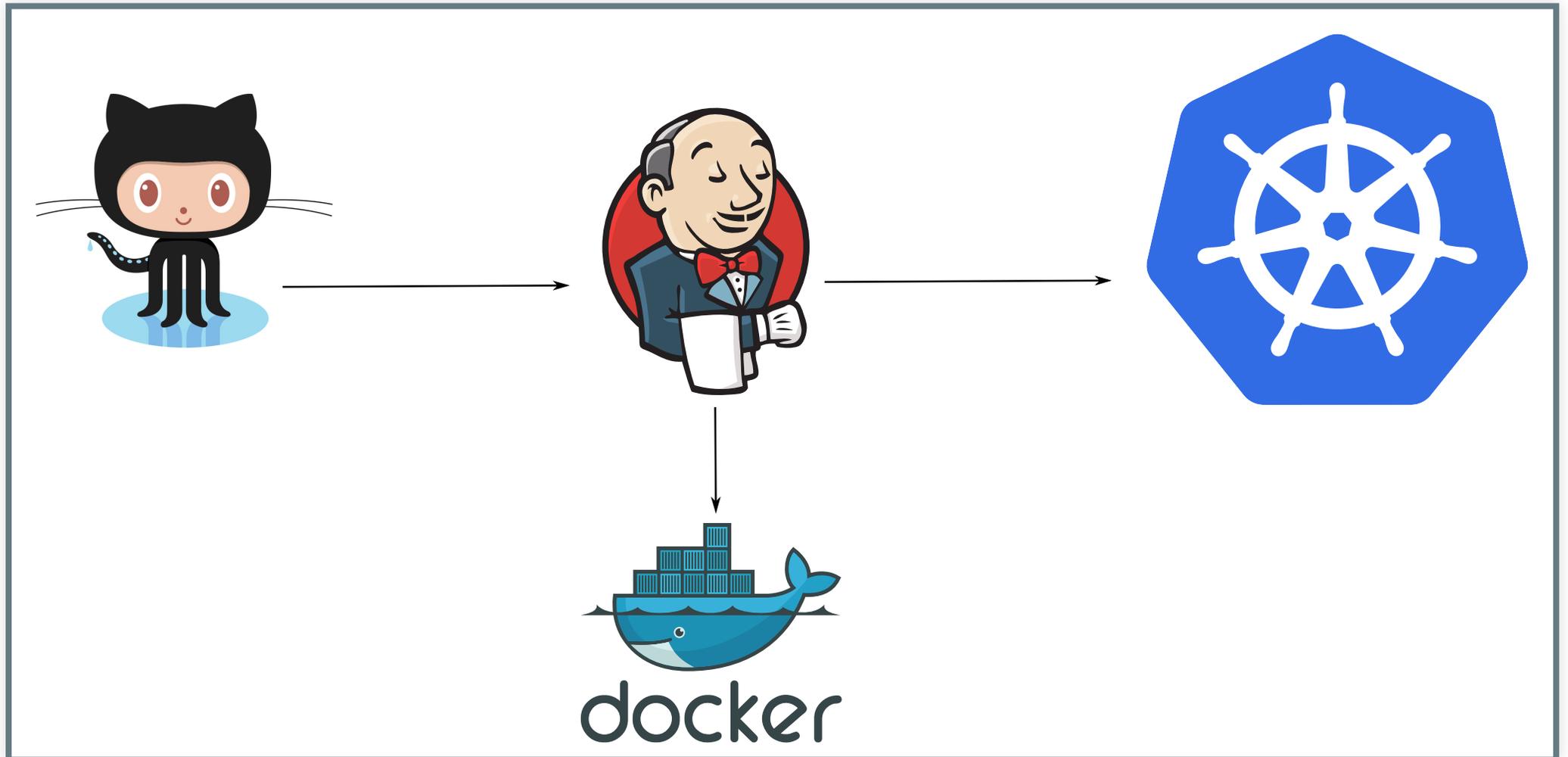
Existant

Objectif

Demo

Retour d'expérience

EXISTANT



LE CHANGEMENT

NOUVEAU SCHÉMA



AWX - WEBHOOK (1)

Mise à disposition d'une API pour recevoir des données

```
EXTRA VARIABLES ⓘ  YAML  JSON
1 tower_webhook_event_guid: b7d56a00-37d5-11ea-8057-ad885c26760a
2 tower_webhook_event_ref: null
3 tower_webhook_event_type: ping
4 tower_webhook_payload:
5   hook:
6     active: true
7     config:
8       content_type: json
9       insecure_ssl: '1'
10      secret: '*****'
11      url: 'https://cb37189d.ngrok.io:443/api/v2/job_templates/7/github/'
12      created_at: '2020-01-15T20:29:24Z'
13      events:
14        - push
15      id: 175270916
16      last_response:
17        code: null
18        message: null
19        status: unused
20      name: web
21      ping_url: 'https://api.github.com/repos/jbradberry/awx/hooks/175270916/pings'
22      test_url: 'https://api.github.com/repos/jbradberry/awx/hooks/175270916/test'
23      type: Repository
24      updated_at: '2020-01-15T20:29:24Z'
25      url: 'https://api.github.com/repos/jbradberry/awx/hooks/175270916'
26      hook_id: 175270916
27      repository:
28        archive_url: 'https://api.github.com/repos/jbradberry/awx/{archive_format}/{ref}'
29        archived: false
30        assignees_url: 'https://api.github.com/repos/jbradberry/awx/assignees{/user}'
31        blobs_url: 'https://api.github.com/repos/jbradberry/awx/git/blobs{/sha}'
32        branches_url: 'https://api.github.com/repos/jbradberry/awx/branches{/branch}'
33        clone_url: 'https://github.com/jbradberry/awx.git'
34        collaborators_url: 'https://api.github.com/repos/jbradberry/awx/collaborators{/collaborator}'
35        comments_url: 'https://api.github.com/repos/jbradberry/awx/comments{/number}'
36        commits_url: 'https://api.github.com/repos/jbradberry/awx/commits{/sha}'
37        compare_url: 'https://api.github.com/repos/jbradberry/awx/compare/{base}...{head}'
38        contents_url: 'https://api.github.com/repos/jbradberry/awx/contents/{+path}'
39        contributors_url: 'https://api.github.com/repos/jbradberry/awx/contributors'
40        created_at: '2018-12-20T19:39:26Z'
41        default_branch: devel
42        deployments_url: 'https://api.github.com/repos/jbradberry/awx/deployments'
43        description: AWX Project
44        disabled: false
45        downloads_url: 'https://api.github.com/repos/jbradberry/awx/downloads'
46        events_url: 'https://api.github.com/repos/jbradberry/awx/events'
```

AWX - WEBHOOK (2)

```
tower_webhook_event_guid: b918273fb75717f2e58133390078558bb4d9a092
tower_webhook_event_ref: 05db47517bda5bcd577543587c21204383217dd9
tower_webhook_event_type: Push Hook
tower_webhook_payload:
  after: 05db47517bda5bcd577543587c21204383217dd9
  before: f5180269af7df67e0d0649df95c91edff0a313ad
  checkout_sha: 05db47517bda5bcd577543587c21204383217dd9
  commits:
  - id: 9d1cd7f60886c0f31097f754283e8c89741db182
    message: |
      Modification mineure
    title: Moidification mineure
    timestamp: '2020-09-17T01:27:31+02:00'
    url: >-
      https://gitlab.com/jamesregis/simple-frontend/-/commit/9d1cd7f60886c0f31097f754283e8c89741db182
    author:
      name: James Regis
```

PODMAN

Remplace Docker

Construire les images depuis *awx_task*

Déposer les images sur un registre

PODMAN - PROBLÈMES

Installation de podman dans AWX

- **Creation d'un environnement virtuel**
- Installation du module Podman pour AWX (pip)
- Installation du binaire podman

PODMAN - PROBLÈMES

Construction d'une image dans un conteneur

Fonctionne bien pour les images simples mais pas pour les images complexes ...

nécessité de modifier libpod, utiliser fuse-overlay, etc ..

Est-ce finalement une bonne idée ?

Utilisation d'une ferme de machines

PODMAN - PROBLÈMES

Accès sécurisé au registre

[podman/common.py](#)

```
if self.validate_certs is not None:
    args.append('--tls-verify')
if self.validate_certs:
    args.append('--tls-verify')
else:
    args.append('--tls-verify=false')
```

Absent dans les nouvelles version du moteur Ansible

DÉPLOIEMENT - 1

Utilisation du module k8s avec des gabarits.

Les variables seront paramétrables.

```
- name: Create namespace
  k8s:
    kubeconfig: "{{ kubeconfig }}"
    context: "{{ cluster.context }}"
    verify_ssl : yes
    api_version: v1
    state: present
    definition: "{{ lookup('template', 'namespace.yml') | from_yaml }}"
```

DÉPLOIEMENT - 2

Création individuelle de chaque ressource

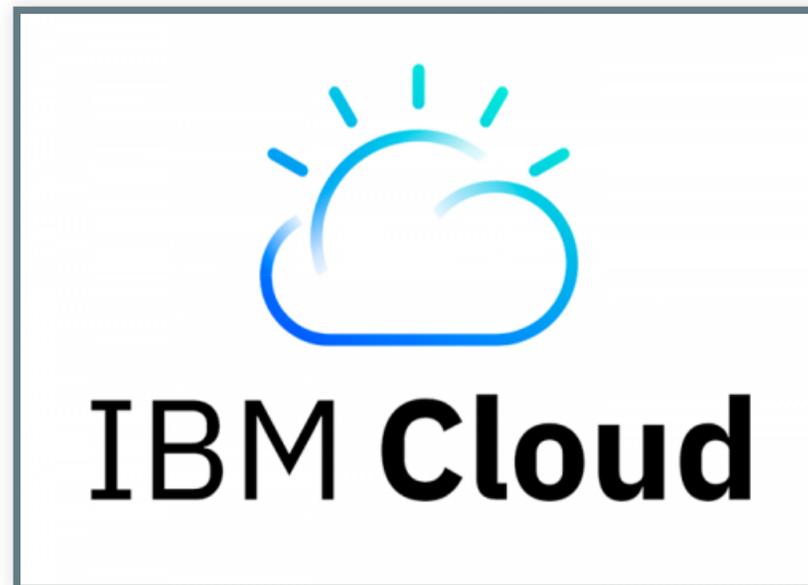
```
templates
├── etc
│   ├── backend-deployment.yml
│   ├── cluster-ip.yml
│   ├── config_map.yml.j2
│   ├── database-secret.yml
│   ├── deployment.yml
│   ├── docker-gcr-access-for-dev.yml
│   ├── gcr-secret.yml
│   ├── horizontal_autoscaler.yml
│   ├── ingress.yml
│   ├── kong-certificate.yml
│   ├── kong-ingress.yml
│   ├── kong-KongIngress-for-certificate-issuing.yml
│   ├── kong-KongIngress.yml
│   ├── namespace.yml
│   ├── nginx-certificate.yml
│   ├── nginx-ingress.yml
│   ├── service-clusterip.yml
│   ├── service-nodeport.yml
│   └── varloop.j2
```

```
tasks
├── backend-deployment.yml
├── check-deployment.yml
├── config-maps.yml
├── create-dns-record.yml
├── create-nginx-file-and-certs.yml
├── create-or-check-gcr-json-file.yml
├── deployment.yml
├── kong-as-controller.yml
├── main.yml
└── nginx-as-controller.yml
```

CONCLUSION

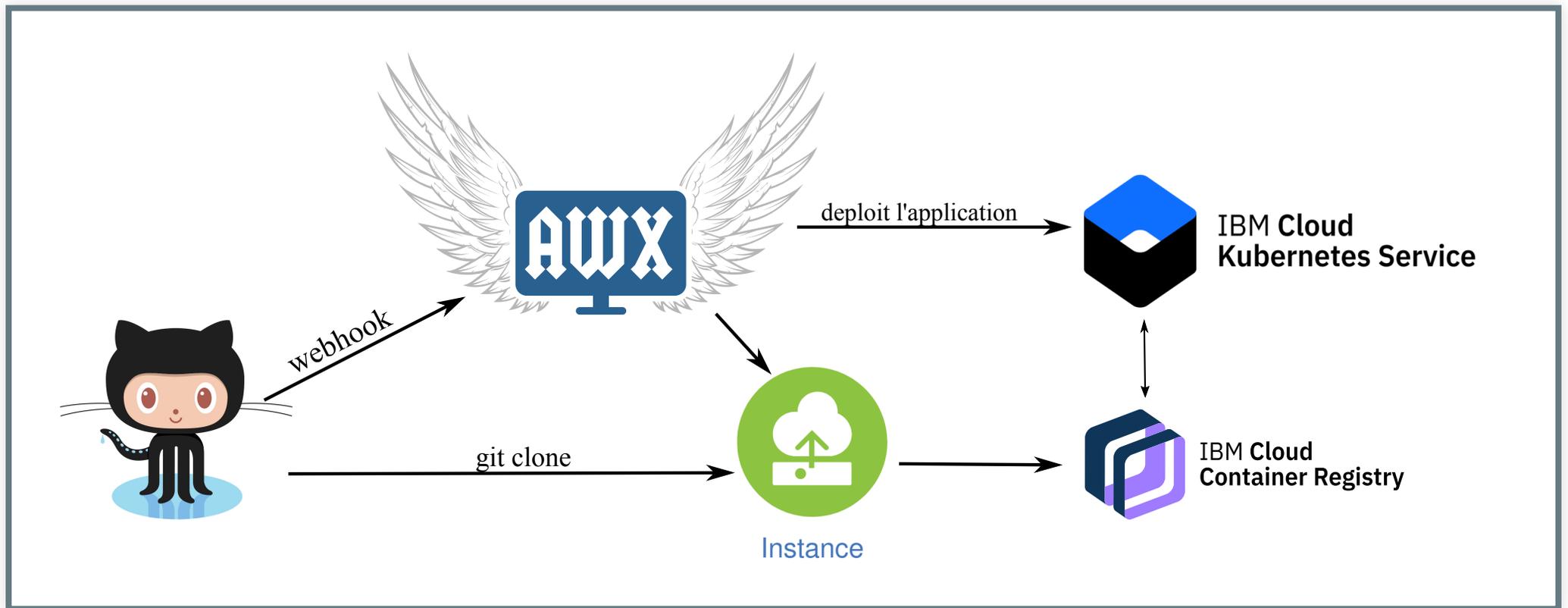
- Installation du module podman
- Construction de l'image sur une machine dédiée (à la demande)
- Déploiement selon les paramètres de l'application

UTILISATION D'IBM CLOUD



Avec les modules Ansible

DÉMO



CONCLUSIONS

- Fonctionne bien
- "build" un peu long : meilleure gestion des ressources
- Voir du côté d'Argo-CD et Tekton (k8s)

