

# Plateforme de CI/CD

Hervé Meftah  
hmeftah@system-dev.io

October 16, 2020

## Introduction

Ce document répond à la spécification suivante:

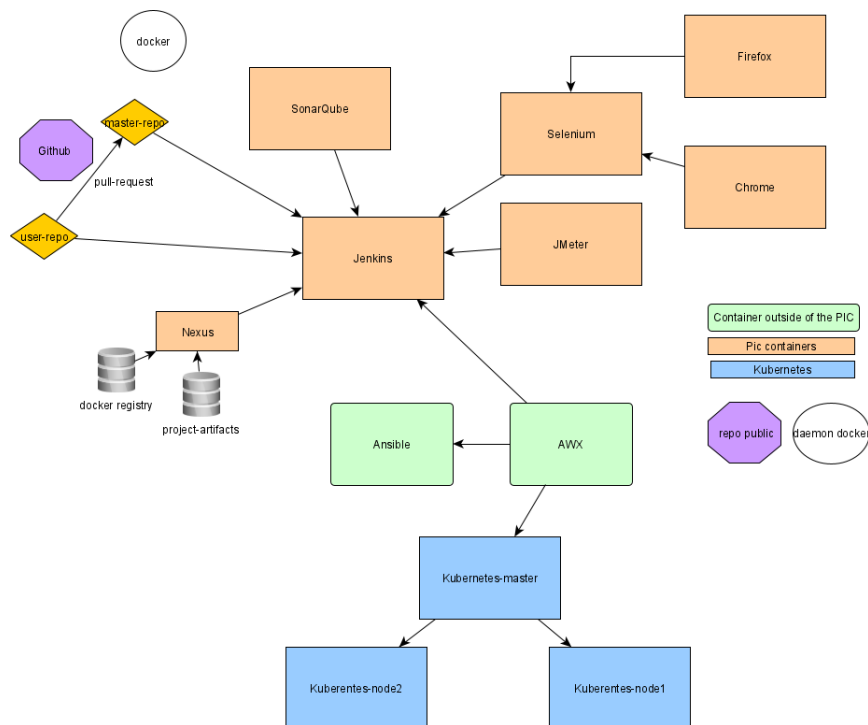
- Expression des besoins** Définir comment déployer, dans Kubernetes, une instance d'un projet à chaque fois qu'un pull-request est envoyée et mettre en place l'automatisation qui le fait.

## 1 Présentation de la plateforme CI/CD

### 1.1 Le schéma

Le schéma ci-dessus décrit les différents composants qui seront utilisés pour déployer les applications dans Kubernetes. Tous ces composants sont sous Docker, communiquant les uns aux autres par le DNS virtuel de

Figure 1: plateforme CI/CD



Docker. Des disques/volumes sont utilisés pour permettre la persistance des données entre les redémarrages de la plateforme.

Github est le gestionnaire de source, il gere les pull-request du *master-repo* qui sont initiés par le repo d'un utilisateur connecté a *user-repo*. Des *webhooks* sont placés comme déclencheurs de la chaine d'intégration continue (CI/CD) suite à la validation, par le *code reviewer*, des corrections ou ajouts faits dans le code source.

L'image du container Jenkins est construite à partir d'un Dockerfile s'appuyant sur la version LTS de Jenkins. Par défaut, elle inclus: le langage Go, la dernière version de python (3.9) et le gestionnaire de build Maven. A l'initialisation le mot de passe administrateur est présent dans les logs du container.

Pour des buts de formation, les plugins Jenkins ne sont installés par défaut. Une installation sans plugin est préférable pour bien s'approprier les concepts de Jenkins.

Le container SonarQube scanne le code source pour effectuer un controle qualité statique, il vérifie les duplications, le code "mort", la couverture des tests unitaires. En sortie de SonarQube, une tâche Jenkins effectue un *docker build* pour injecter le code source dans une image docker.

Le container Nexus, correspond à un Artifactory, archive les images, les sources intermediaires pour un usage ultérieur. Jmeter et Selenium sont des outils pour respectivement tester les montees en charge et les tests fonctionnels a travers l'interface Web ou GUI de l'application.

L'automatisation de cette chaine d'intégration, pilotée par Jenkins, inclus Ansible et la version open-source Tower: AWX. Le provisionning d'infrastructure est fait avec Ansible, et en fin de chaine, Ansible se connecte à la plateforme Kubernetes pour déployer les artifacts validés par les tests et stockés dans une image Docker présent dans Nexus.

## 2 Installation de la plateforme

### 2.1 Pré-requis

Quelques outils doivent etre installes en premier pour faciliter la gestion de la plateforme.

#### Command Line

```
## Pre-requis pour Centos 7
sudo yum -y update    # update all packages
sudo yum -y install git wget    # install git and wget
sudo yum -y install epel-release # added extra packages
sudo yum -y install htop iotop iftop # added monitoring tools
//Fork
//      https://github.com/crunchy-devops/tp-devops1.git
git clone your personnal repository of tp-devops1
git clone https://github.com/<your-repo>/tp-devops1.git
cd tp-devops1
```

### 2.2 Docker

Docker permet la création d'applications virtualisées dans des containers, qui sont plus légers que des machines virtuelles Docker fait partie des concepts d'*Infrastructure As Code (IaC)* ou votre environnement applicatif est "codé" par des scripts exécutables, versionnables et donc tracable tout au long du cycle de développement.

#### 2.2.1 Installation de Docker

Docker peut etre installé par des OS packages, mais la maniere la plus rapide est d'utiliser Ansible.

#### Command Line

```
sudo yum -y install python3
python3 -m venv venv # install virtualenv module dans la directory venv
source venv/bin/activate # activate the python virtualenv
pip3 install wheel # install pip package wheel for permission usage
pip3 install ansible # install ansible
ansible --version # check
ansible-playbook -i inventory playbook.yml
```

Fusce varius orci ac magna dapibus porttitor. In tempor leo a neque bibendum sollicitudin. Nulla pretium fermentum nisi, eget sodales magna facilisis eu. Praesent aliquet nulla ut bibendum lacinia. Donec vel mauris vulputate, commodo ligula ut, egestas orci. Suspendisse commodo odio sed hendrerit lobortis. Donec finibus eros erat, vel ornare enim mattis et.

### 3 Implementation

Proin lobortis efficitur dictum. Pellentesque vitae pharetra eros, quis dignissim magna. Sed tellus leo, semper non vestibulum vel, tincidunt eu mi. Aenean pretium ut velit sed facilisis. Ut placerat urna facilisis dolor suscipit vehicula. Ut ut auctor nunc. Nulla non massa eros. Proin rhoncus arcu odio, eu lobortis metus sollicitudin eu. Duis maximus ex dui, id bibendum diam dignissim id. Aliquam quis lorem lorem. Phasellus sagittis aliquet dolor, vulputate cursus dolor convallis vel. Suspendisse eu tellus feugiat, bibendum lectus quis, fermentum nunc. Nunc euismod condimentum magna nec bibendum. Curabitur elementum nibh eu sem cursus, eu aliquam leo rutrum. Sed bibendum augue sit amet pharetra ullamcorper. Aenean congue sit amet tortor vitae feugiat.

In congue risus leo, in gravida enim viverra id. Donec eros mauris, bibendum vel dui at, tempor commodo augue. In vel lobortis lacus. Nam ornare ullamcorper mauris vel molestie. Maecenas vehicula ornare turpis, vitae fringilla orci consectetur vel. Nam pulvinar justo nec neque egestas tristique. Donec ac dolor at libero congue varius sed vitae lectus. Donec et tristique nulla, sit amet scelerisque orci. Maecenas a vestibulum lectus, vitae gravida nulla. Proin eget volutpat orci. Morbi eu aliquet turpis. Vivamus molestie urna quis tempor tristique. Proin hendrerit sem nec tempor sollicitudin.

```
hello.py

#!/usr/bin/python

import sys
sys.stdout.write("Hello World!\n")
```

Fusce eleifend porttitor arcu, id accumsan elit pharetra eget. Mauris luctus velit sit amet est sodales rhoncus. Donec cursus suscipit justo, sed tristique ipsum fermentum nec. Ut tortor ex, ullamcorper varius congue in, efficitur a tellus. Vivamus ut rutrum nisi. Phasellus sit amet enim efficitur, aliquam nulla id, lacinia mauris. Quisque viverra libero ac magna maximus efficitur. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vestibulum mollis eros in tellus fermentum, vitae tristique justo finibus. Sed quis vehicula nibh. Etiam nulla justo, pellentesque id sapien at, semper aliquam arcu. Integer at commodo arcu. Quisque dapibus ut lacus eget vulputate.

#### Command Line

```
$ chmod +x hello.py
$ ./hello.py

Hello World!
```

Vestibulum sodales orci a nisi interdum tristique. In dictum vehicula dui, eget bibendum purus elementum eu. Pellentesque lobortis mattis mauris, non feugiat dolor vulputate a. Cras porttitor dapibus lacus at pulvinar. Praesent eu nunc et libero porttitor malesuada tempus quis massa. Aenean cursus ipsum a velit ultricies sagittis. Sed non leo ullamcorper, suscipit massa ut, pulvinar erat. Aliquam erat volutpat. Nulla non lacus vitae mi placerat tincidunt et ac diam. Aliquam tincidunt augue sem, ut vestibulum est volutpat eget. Suspendisse potenti. Integer condimentum, risus nec maximus elementum, lacus purus porta arcu, at ultrices diam nisl eget urna. Curabitur sollicitudin diam quis sollicitudin varius. Ut porta erat ornare laoreet euismod. In tincidunt purus dui, nec egestas dui convallis non. In vestibulum ipsum in dictum scelerisque.



**Notice:** In congue risus leo, in gravida enim viverra id. Donec eros mauris, bibendum vel dui at, tempor commodo augue. In vel lobortis lacus. Nam ornare ullamcorper mauris vel molestie. Maecenas vehicula ornare turpis, vitae fringilla orci consectetur vel. Nam pulvinar justo nec neque egestas tristique. Donec ac dolor at libero congue varius sed vitae lectus. Donec et tristique nulla, sit amet scelerisque orci. Maecenas a vestibulum lectus, vitae gravida nulla. Proin eget volutpat orci. Morbi eu aliquet turpis. Vivamus molestie urna quis tempor tristique. Proin hendrerit sem nec tempor sollicitudin.