

Sujet TP – CI/CD

Table des matières

Présentation du Contexte.....	2
Organisation.....	2
Description détaillé des éléments attendus.....	2
Dockerfile.....	2
Cluster Kubernetes.....	2
Documentation.....	3
(Optional) Ajouter un pod de monitoring.....	3
Serveur de déploiement continu avec Jenkins.....	3
Rendu.....	3

Présentation du Contexte

Vous arrivez dans une entreprise qui gère des parcs informatiques. Le CTO a décidé de doter ses équipes d'outils pour automatiser le déploiement continu et la gestion de l'infrastructure

Vous allez donc devoir :

- Préparer un DockerFile
- Mettre en place un cluster Kubernetes
- Déployer en continu dans cet environnement depuis un JenkinsFile

Organisation

Cet exercice se fait en binôme et sera à rendre sous forme d'un dépôt Github commun au plus tard le mercredi 9 Juin 2021 à 17h00 .

Le lien du dépôt github sera à mettre dans le fichier Excel de rendu qui se trouve sur Teams en face de votre binôme.

Ce travail est à réaliser sur l'environnement vagrant que vous connaissez. A vous de fournir les vagrant file et si nécessaire les fichiers de provisioning pour permettre l'utilisation du cluster et le déploiement continu

Description détaillé des éléments attendus

Dockerfile

Il est nécessaire de préparer un dockerfile pour l'application Python, celui-ci s'appuiera sur l'image officielle de Python et lancera au démarrage du conteneur le fichier main.py qui est dans le dossier <https://github.com/vanessakovalsky/example-python/tree/master/vagrant/src>

Le docker file sera ensuite taggué et envoyé sur le hub docker pour être utilisé par le Cluster.

Cluster Kubernetes

Un Cluster Kubernetes composé des éléments suivants est attendu :

- Une application Python présent sur 3 environnement (dev, test, prod), avec 3 URLs différentes donc un Pod au sens Kubernetes en 3 exemplaires
- D'un point de vue technique un manifest de déploiement pour cette application (à partir du DockerFile fait à l'étape précédente), et un manifest pour le service associé pour l'exposé sur internet . Si nécessaire prévoir des fichiers pour gérer les éléments variables (sous forme de configmap ou de secret)

Documentation

Une documentation sur l'utilisation des différents manifests ainsi qu'un schéma de l'architecture du cluster est à mettre dans le dépôt au format que vous souhaitez.

(Optionnel) Ajouter un pod de monitoring

Si vous avez le temps, vous pouvez également fournir les fichiers manifests nécessaire pour déployer un pod de monitoring avec la stack ELK.

Astuce : vous pouvez transformer un fichier docker-compose en manifest avec Kompose :

<https://kompose.io/>

Serveur de déploiement continu avec Jenkins

A partir du JenkinsFile présent ici : <https://github.com/vanessakovalsky/example-python> vous aller devoir rajouter des étapes de déploiement continu.

Modifier l'étape compile pour récupérer votre propre dockerfile et builder l'image en local.

Ajouter alors une étape qui va pousser votre nouvelle image sur le hub docker.

A la suite de l'étape 'testcode' ajouter une étape 'déploiement'.

Cet étape devra appeler un playbook Ansible (à créer avec son fichier hosts) qui sera en charge de lancer un nouveau déploiement sur le cluster Kubernetes avec la dernière version de l'image

Rendu

Le rendu se fait sous la forme d'un dépôt github contenant à minima :

- le Dockerfile
- Les fichiers manifest (au format YAML) nécessaire pour le cluster Kubernetes
- Le (ou les) fichier(s) playbook d'ansible et son fichier hosts
- Le jenkinsfile mis à jour
- une documentation sous la forme de votre choix