TP: Hashservice

Notions: micro-service, scripting, infrastructure, automatisation, CI/CD
Outils: vagrant, jenkins, git, docker, ansible

Infrastructure locale à mettre en place:
Machine Master

- 1 Ubuntu Server 18.04 disposant de:
  - jenkins
  - ansible

Machines contrôlées:

- 2 Ubuntu Server 18.04
- 1 Centos 7

Créer un micro-service web - framework/language au choix - exposant les endpoints suivants:

GET /md5/<str>
GET /sha256/<str>

Chacune des ces routes renverra le hash MD5 ou SHA256 de la chaîne fournie en paramètre url (<str>)

Exemples d'utilisation
La requête http GET **/md5/hello** renverra le json suivant:

```
{
  "hash": "md5",
  "cleartext": "hello",
  "hashedtext": "5d41402abc4b2a76b9719d911017c592"
}
```

La requête http GET **/sha256/hello** renverra le json suivant:

```
{
  "hash": "sha256",
  "cleartext": "hello",
  "hashedtext": "2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824"
}
```

Créer un dépôt github qui contiendra:

- les sources du micro-service
- le Dockerfile permettant de conteneuriser ce micro-service
- un playbook ansible
- un fichier hosts

Créer une pipeline Jenkins ou Gitlab-ci effectuant les tâches suivantes:

- récupération du dépôt github

- build de l'image docker et démarrage de l'application conteneurisée
- test des endpoints par requête http et comparaison du retour avec le résultat attendu
- si le test réussit on exécute le playbook ansible de déploiement

La playbook ansible devra exécuter les tâches suivantes:

- sur les VMs ubuntu
    - build de l'image docker à partir du Dockerfile
    - lancement de 3 conteneurs sur la base de cette image (écoutant des ports différents, exemple: 3001, 3002, 3003)
    - facultatif: lancement d'un conteneur nginx écoutant le port 80 de la VM et répartissant la charge (load balancing) sur les serveurs applicatifs (https://docs.nginx.com/nginx/deployment-guides/load-balance-third-party/node-js/)
- sur la VM centos
    - téléchargement d'une application front statique (fichier .zip) à l'url suivante: https://opusidea-training.s3.eu-west-3.amazonaws.com/divers/hashservice-client.zip
    - désarchivage de l'application
    - éventuel remplacement d'adresses IP/ports à effectuer dans le fichier source: index.js (`API_URL = "http://localhost:3000/"`)
    - démarrage d'un serveur apache (installé ou conteneurisé) afin de servir l'application client

Livrables attendus (par mail / chat teams)

- adresse du dépôt du github
- descriptif de la pipeline (Jenkinsfile, gitlab-ci.yml)

Aperçu de l'application client dans le navigateur