Intégration Continue - Mise en pratique - Gitlab CI

1. Initialiser un **nouveau repository git** qui vous permettra de sauvegarder les fichiers créés pendant le TP. Vous enverrez un zip du repository à la fin du TP avec vos réponses aux questions / exécutions et résultats sur la console dans un **fichier texte** (Markdown par exemple) par e-mail.
   Utilisez git progressivement ! (Ne pas faire qu'un seul commit à la fin)

2. Récupérer les identifiants de connexion à votre VM.

3. Installer Docker et Docker Compose sur votre VM.

4. Utiliser docker-compose pour installer GitlabCI sur votre VM
   https://docs.gitlab.com/ee/install/docker.html
   A l'aide de la documentation ci-dessus, rédiger un fichier docker-compose.yml plutôt que d'utiliser directement la commande docker fournie.
   Utiliser la version **gitlab-ce**
   a. Lors de l'installation de Gitlab dans Docker vous allez pouvoir choisir une adresse pour votre gitlab. Vous pouvez utiliser gitlab.example.com
   b. Ensuite modifier votre fichier /etc/hosts local et ajouter la ligne suivante afin que l'adresse gitlab.example.com dirige bien vers votre VM :

      <IP_VM> gitlab.example.com

5. L'installation de Gitlab peut prendre un peu de temps. Vous pouvez suivre ce que fait le conteneur docker avec la commande docker logs. Pendant ce temps, créer un fichier ANSWER et expliquer quelle bonne pratique Docker n'est pas respectée dans l'image Gitlab.

6. Récupérer le mot de passe du compte root, connectez vous et créez vous chacun un compte administrateur. Profitez-en pour explorer l'interface et les possibilités offertes par Gitlab. Noter dans le fichier ANSWERS les fonctionnalités offertes par Gitlab qui vous paraissent intéressantes dans le cadre d'une démarche DevOps.

7. Créer le premier commit contenant juste un README, le pousser sur un nouveau repository dans votre Gitlab (Utiliser **export GIT_SSL_NO_VERIFY=1**)

8. Dans un nouveau conteneur docker, configurer un runner gitlab. (Cf. le menu Runners dans l'administration de votre Gitlab)
   https://docs.gitlab.com/runner/install/docker.html

9. Créer un fichier gitlab-ci.yaml simple qui affiche "Hello World !" lorsqu'un commit est effectué. Puis tester l'exécution de la pipeline.
   https://docs.gitlab.com/ee/ci/yaml/

10. Créer deux classes python, une classe SimpleMath contenant une fonction statique “addition” prenant deux arguments. Et une classe TestSimpleMath qui hérite de unittest.TestCase et contient une fonction de test

11. Faites en sorte que la pipeline Gitlab effectue les tests lorsqu’un nouveau commit est effectué.

12. Créer la fonction soustraction et le test associé. Puis, pousser votre commit. Les tests sont effectués automatiquement via la pipeline.

13. Ajouter une étape de lint dans votre gitlab-ci.yaml. Utiliser pylint

Envoyer par e-mail à damien.montmoulinex@ynov.com un zip nommé **GITLAB-NOM1-NOM2.zip** contenant votre repository git avec :

- Tous les commits réalisés pendant la séance
- Les réponses aux questions et les copier/coller de votre terminal (commandes et résultats) dans des fichiers texte (Markdown si vous le souhaitez)
- Préciser l’adresse IP de votre machine dans le mail