Objectif du projet:

1. déployer une plateforme d'intégration continue (PIC) intégrant des outils de CI/CD.

2. développer une chaîne complète de déploiement continue permettant de faire des mise en production (MEP) automatiquement, avec possibilité de rollback.

---

Livrables attendus:

1. le projet sera constitué d'un repo git contenant:

   - une courte documentation (README)

   - les scripts pour les différentes actions à mener dans le projet.

2. A la question "doit-on automatiser cette partie?" la réponse sera toujours OUI, sauf s'il y a une impossibilité technique à la faire.

3. La priorité reste néanmoins de livrer une solution fonctionnelle, a minima. A chaque équipe de s'organiser pour concilier vitesse de livraison/automatisation.

4. Chaque équipe s'organisera en SCRUM avec un ou plusieurs sprints pour livrer le projet (choix laissé à l'équipe).

5. Le projet sera noté UNIQUEMENT sur le projet git (github ou gitlab) et la tableau SCRUM (sur Trello ou avec des post-its). Tout ce qui ne sera ni sur l'un ni sur l'autre ne sera pas pris en compte, notamment les scripts stockés en local, etc.

6. La date de livraison du projet est le vendredi 21 février 2020, à 17h, heure de Paris. Tous les commits postérieurs à cette date sur le repo git ne sera pas pris en compte pour la notation du livrable.

7. Les objectifs 1. et 2. du projet ne sont pas chronologiques et ne supposent donc pas que l'un soit réalisé avant l'autre. Chaque équipe fera ses propres choix sur les priorités de livraison.

---

Descriptif du projet:

1. Déployer une PIC as Code:

   - Votre client a besoin d'une plateforme complète de CI/CD contenant:

     a. un SCM (github ou gitlab au choix). Ce dernier peut être installé en local (uniquement gitlab dans ce cas).

     b. un orhchestrateur qui sera Jenkins. Ce dernier se déclinera en un Master et un noeud contenant un docker-engine. Jenkins pourra se connecter en SSH au noeud.

     c. un serveur pour l'environnement de test avec un docker-egnine.

     d. un serveur pour l'environnement de prod avec un docker-engine.

   - Tous les serveurs locaux seront installés sur des machines vagrant. Le provisioning de ces serveurs se fera via ansible.

   - Vous organiserez le provisioning ansible par des rôles spécifiques.

   - Les livrables attendus pour cette partie sera donc une liste de vagrantfile et de rôles ansible.

   - Il est parfaitement possible d'automatiser la configuration des serveurs et pas uniquement leur installation. A chaque équipe d'évaluer le temps qu'elle souhaite passer sur cette tâche.

2. Chaîne de production complète:

   - Votre client a besoin d'une chaîne de déploiement complète qui répond aux exigences suivantes:

     a. Les features développées par les équipes de dev doivent être systématiquement déployées sur l'environnement de test, via une image docker (qui sera stockée sur le Hub Docker).

     b. Les évolutions sur la branche master du SCM seront systématiquement déployées sur l'environnement de production, via une image docker.

     c. Si erreur de déploiement en production, un job Jenkins de rollback devra pouvoir être lancé pour revenir sur une ancienne version de manière automatique.

   - La chaîne de production sera contsitué d'un seul pipeline Jenkins permettant de:

     a. Récupérer le projet sur le SCM

     b. Réaliser le build du projet avec Maven, en incluant un rapport de test à chaque build.

     c. Builder une image Docker avec l'environnement d'exécution java qui permet de démarrer l'application

     d. Pousser l'image Docker sur Hub Docker.

     e. Lancer un script ansible qui déploie un conteneur avec l'image en question et qui la rend accesible sur un port spécifique.

   - Le code en question sera un projet springboot dont l'URL est : <https://gitlab.com/RaphaeldeGail/devopsapp.git>

   - Il sera possible d'utiliser un webhook pour déclencher les jobs Jenkins.

3. Partie BONUS à ne réaliser que si les deux premières parties ont déjà été entièrement livrées:

   - Répétez le même exercice mais en déployant l'application dans un cluster kubernetes.

Chaque équipe devra tester sa chaîne de production en essayant de commiter sur différentes branches.