

Timon — Ethics

Devops niveau 2 : Intégration continue

Mise en place et travail avec une plateforme d'intégration
continue

Christophe Henry

Table des matières

Objectifs du cours	3
Détail du projet à rendre	3
Application de gestion des tâches	3
Vagrantfile	4
Ansible	4
Jenkins et Jenkinsfile	5
Compte-rendu de développement	5
Organisation du groupe	6
Modalités de rendu	6
Notation	6

Dans ce cours, nous allons apprendre à déployer et à utiliser une plateforme d'intégration continue basée sur Jenkins et à développer un projet en suivant les principes du devops et quelques pratiques de bases de la méthode *scrum*.

Objectifs du cours

Il vous sera demandé de développer, en groupe, un projet simple de gestion de liste de tâches en suivant en respectant des règles de qualité de code. Vous devrez pour ce faire, vous devrez développer, parallèlement au développement du projet, des scripts permettant de créer une machine virtuelle de manière reproductible à l'aide de Vagrant¹ et de la provisionner avec Ansible².

Vous pourrez vous appuyer sur les TP fournis pour mieux comprendre comment s'agencent tous ces composants.

Détail du projet à rendre

Application de gestion des tâches

Votre projet devra se présenter sous la forme d'un service web dont le front est accessible dans le navigateur sur `http://localhost/app`. Il devra permettre à l'utilisateur ou l'utilisatrice de :

- créer une tâche,
- supprimer une tâche,
- modifier une tâche.

Chaque tâche est constitué, au minimum d'un titre et d'une description. Vous pouvez ajouter des fonctionnalités si cela vous intéresse mais n'y passez pas trop de temps, il y a déjà bien assez de choses à faire.

Ces tâches devront être stockées dans une base de données installée (Postgres, MySQL, etc.), *non-embarquée* (pas de SQLite ou HSQL).

1. <https://www.vagrantup.com/>

2. <https://www.ansible.com/>

En plus du code lui-même, vous devrez veiller à la qualimétrie de votre projet. Votre code (front et back) devra être couvert à 80% par des tests unitaires ou d'intégration et vous devrez choisir et respecter les bonnes pratiques de style de code en utilisant des outils de lint comme `eslint` pour Javascript.

Vagrantfile

Vous devrez aussi rendre un **Vagrantfile** qui se trouvera à la racine du projet et qui permettra de créer une nouvelle machine virtuelle pour le déploiement du projet.

Ce **Vagrantfile** devra obligatoirement créer une machine virtuelle Linux. Vous êtes libres de choisir la *box* Vagrant qui vous convient parmi celles disponibles sur le site de Vagrant³. Certaines sont très génériques et ne fournissent qu'une distribution Linux de base d'autres viennent avec des frameworks pré-installés.

De même, vous êtes libres de choisir le système de virtualisation qui vous convient pour travailler. Vagrant permet de créer des VM sous Virtualbox et Hyper-V nativement (VM Ware est supportés mais moyennant l'achat d'une licence donc il n'est pas conseillé).

Sachez cependant que, pour la correction, le Vagrantfile qui se trouve à la racine du projet doit obligatoirement être adapté pour Virtualbox. Si vous souhaitez utiliser un autre système de virtualisation, vous pouvez écrire un autre **Vagrantfile**. Dans ce cas, veillez bien à tester le **Vagrantfile** principal avant le rendu du projet.

Ansible

Vagrant est capable de travailler avec de multiples systèmes de provisionnement de machines, notamment Ansible, qui fait l'objet du cours de devops niveau 1. Vous vous en servirez donc pour votre projet ainsi qu'une instance Jenkins qui sera chargée de compiler et exécuter les tests à chaque nouveau changement détecté dans le dépôt git.

3. <https://app.vagrantup.com/boxes/search>

Jenkins et Jenkinsfile

Afin de permettre à Jenkins de compiler et tester votre projet en permanence, vous placerez un `Jenkinsfile` à la racine de votre projet qui exécutera les différentes tâches de vérification de qualimétrie :

- compilation,
- tests,
- lint,
- autres.

Compte-rendu de développement

En plus du projet lui-même, vous devrez également rendre un livret de quelques pages en PDF (mention spéciale si écrit avec \LaTeX) dans lequel vous détaillerez les choix technologiques produits par le projet et les raisons pour lesquelles ces choix ont été faits.

Aucune contrainte extérieure ne pèse sur le projet en terme d'architecture technique ou d'organisation. Tout choix technologique sera considéré comme parfaitement valable au regard de la notation tant qu'il est justifié. Les justifications peuvent aller du plus conservateur (« c'est une technologie avec laquelle nous nous sentons à l'aise »), « c'est une technologie que nous pensons être pérenne ») au plus aventurier (« c'est une technologie que nous souhaitons apprendre »).

L'objectif ici est que vous soyez capable de justifier un choix technologique par des arguments objectifs plus que par des arguments de goûts personnels. Préférer rester sur une technologie connue pour maximiser les chances du projet de réussir est un argument aussi valable que de profiter d'un projet concept pour rajouter une corde technologique à son arc. L'important est de surtout prendre la décision en équipe et pour de bonnes raisons.

Vous pourrez également, si cela vous semble utile, détailler un ou plusieurs problèmes particulièrement ardues rencontrés lors du projet.

Organisation du groupe

La projet doit être développé en groupe de 5 ou 6 personnes. Lors de la première demi-journée, vous vous réunirez pour choisir les technologies que vous souhaitez adopter pour développer le projet. Vous êtes entièrement libres.

Vous serez libre de vous répartir les tâches à faire comme bon vous semble mais vous devrez vous réunir, tous les matins à heure fixe et en présence du formateur pour exposer chacun et chacune votre tour ce que vous avez fait lors de la dernière séance, ce que vous vous comptez faire lors de cette séance et les problèmes rencontrés, s'il y en a eu.

Modalités de rendu

Au terme du cours, vous devrez rendre le projet sous la forme d'une URL vers un dépôt *git* qui puisse être cloné anonymement en HTTP.

Le dépôt devra contenir le projet, le **Vagrantfile**, le playbook et les rôles Ansible permettant de monter la VM .

Notation

Afin d'obtenir la note maximum, les objectifs suivants devront être remplis :

- aucune commande autre que **vagrant up** ne doit être nécessaire pour démarrer la machine virtuelle et accéder à Jenkins et à l'application,
- aucun effet de bord (aucune installation, aucune création de fichier ou dossier, aucun téléchargement), ne pourra être fait sur la machine hôte en dehors du répertoire qui contient le **Vagrantfile** pour des raisons de sécurité de la machine du correcteur...
- vous pouvez écrire autant de **Vagrantfile** que vous le désirez mais celui qui sera placé à la racine du projet sera celui utilisé pour la correction et devra obligatoirement fonctionner avec Virtualbox,
- Jenkins doit exposer les afficher les jobs et les rapport de build pour les utilisateurs et utilisatrices non-connectées mais ne pas permettre

de faire des opérations (lancer un build ou modifier des paramètres) sans l'être,

- la machine virtuelle devra exposer un serveur Jenkins sur `http://localhost/jenkins` qui redirigera vers `http://localhost:8080/jenkins` de la machine hôte et devra exposer l'application de gestion de tâches `http://localhost/app` qui redirigera vers `http://localhost:8080/app` de la machine hôte,
- la couverture de test du code devra être de 80% à la fois pour le code front et le code back et devra être exposé clairement sur Jenkins,