**LOG3000**

**Processus du génie logiciel**

**Hiver 2018**

TP5

Déploiements bleus et verts

Présenté par :

Taleb Souli (1770491)

David Tremblay (1748125)

18 Avril 2018

École Polytechnique de Montréal

Département de Génie Informatique et Logiciel

**Questions d’analyses**

**Q1. Expliquez comment effectuer le déploiement bleu/vert avec votre solution. (Les lignes de commandes seulement)**

Le déploiement bleu/vert n’est pas opérationnel dans notre travail pratique. La commande pour alterner entre les déploiements ne fonctionne donc pas.

**Q2. Qu’est-ce qu’un rôle dans Ansible?**

Un rôle est un ensemble de tâches et autres fichiers qui s'exécute automatiquement. Les différents types de tâches pouvant être inclus dans un rôle sont *task, handler, default, var, file, template* et *meta*. Les rôles permettent aussi de plus facilement partager du contenu entre différents utilisateurs et d’augmenter la réutilisabilité.

**Q3. Quelle est la différence entre Docker et Ansible?**

Docker permet de créer des conteneurs qui permettent d’instancier une application sur un client sans se soucier du OS. Le conteneur est beaucoup plus léger qu’une machine virtuelle classique et permet de facilement assurer que tous les utilisateurs travaille exactement sur le même environnement.

Ansible est un outil de déploiement et est plutôt utilisé pour faire des tâches qu’on peut facilement automatiser à l’aide de playbook et on peut regrouper le contenu similaire en rôles. Ansible permet aussi de déployer nos applications d’une manière facilement reproduite. Ansible et Docker vont donc très bien ensemble puisque qu’on peut automatiser les conteneurs Docker à l’aide d’Ansible.

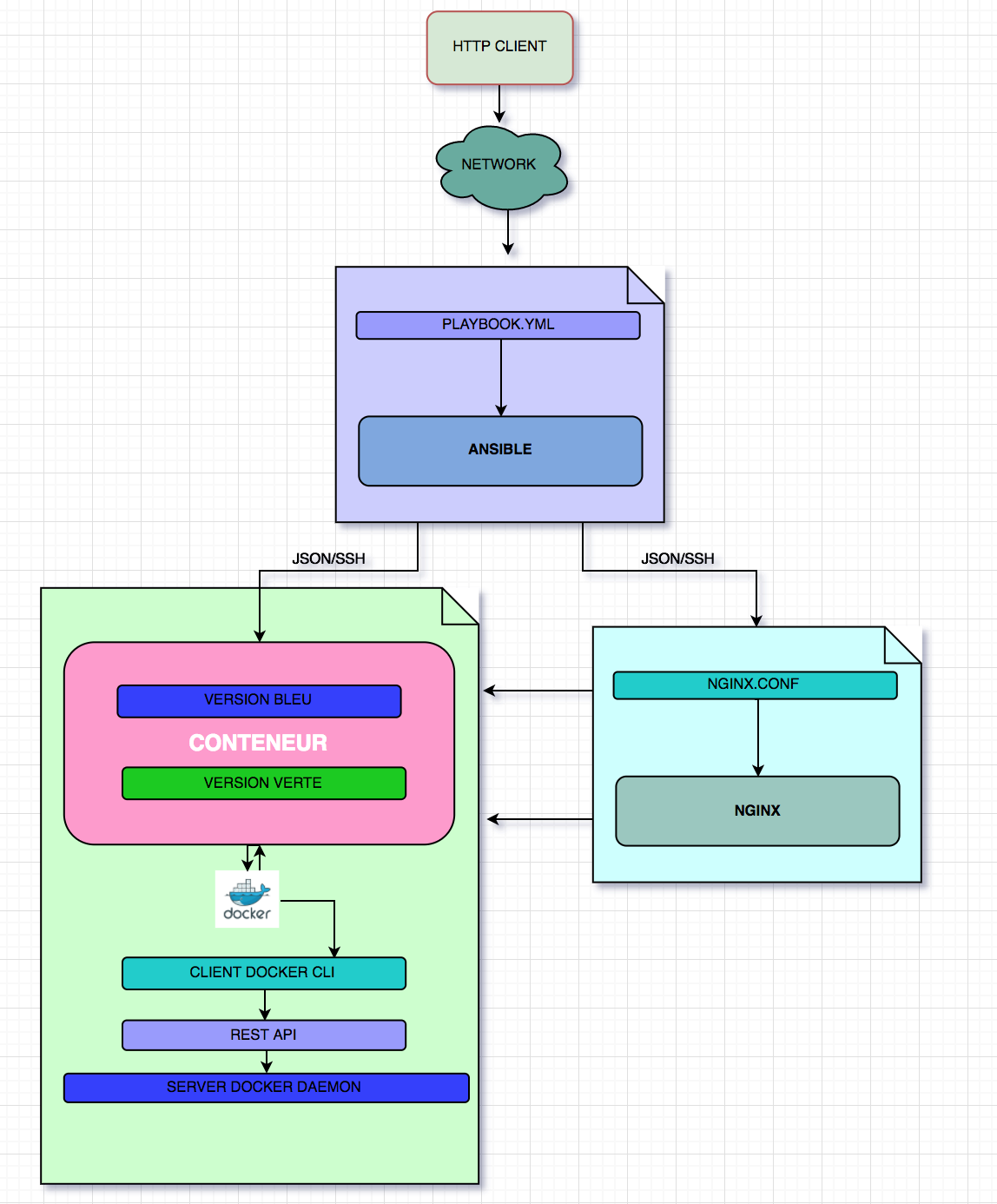
**Q4. Donnez un avantage de Ansible face à d’autres outils similaires tel que Puppet ou Chef.**

La structure des playbook est simple et structuré, ce qui permet de rapidement apprendre Ansible et facilement obtenir quelque chose de fonctionnel. Il fonctionne aussi sans agent.

|  |  |  |
| --- | --- | --- |
|  | **Avantage** | **Désavantage** |
| **Ansible** | * Excellente performance, installation et déploiement sans agent. * Faible frais généraux, basé sur les playbooks. | * Toujours très nouveau pas encore essayé et testé par beaucoup. * Pas de support pour Windows * interface graphique en cours de travail |
| **Puppet** | * Solution mature. * Bonne interface graphique. * Prise en charge de tous les principaux systèmes d'exploitation. * Installation facile | * Lent pour répondre et router les préoccupations des clients. * basé sur Ruby, performances discutables par rapport aux outils CM basés sur Python. * Bientôt, tous les clients doivent apprendre la marionnette DSL. |

**Questions de réflexion**

**Q1. À l’aide d’un schéma, expliquer l’architecture de votre déploiement. Expliquez votre schéma**

****

**Q2. Décrivez les avantages de déployer des applications avec un outil comme Ansible. [1]**

Le déploiement et processus peuvent très facilement être reproduis. Il est donc aisé d’obtenir un environnement consistent pour une application. Ansible permet aussi, comme vu lors de ce TP, d’obtenir des applications avec aucun downtime avec des déploiements bleu et vert. L’outil Ansible peut être exécuter de sa propre machine et il est très facile et rapide d’installation. La documentation est aussi bien structurée et contient beaucoup d’exemple pour faciliter l'apprentissage. Ansible contient aussi plusieurs modules de base par défaut. L’automatisation et le principe des états font sont aussi une force d’Ansible. Elles permettent d’effectuer des actions seulement lorsque nécessaire et automatiquement, ce qui est beaucoup plus ardu dans un script usuel. À travers les rôles, Ansible permet de regrouper les actions qui se répètent et ainsi augmenter la réutilisabilité du code, un principe très important en développement logiciel. Encore mieux, Ansible possède aussi des playbooks qui permettent d’appliquer des actions sur un ensemble de machines automatiquement.

**Q3. Dans un tableau, comparez les features principaux de Ansible avec ceux de docker-compose.**

|  |  |
| --- | --- |
| **Ansible** | **docker-compose** |
| Sans agent | Configuration de plusieurs conteneurs qu’on peut partir en une seule commande |
| Les playbooks peuvent être configurés pour être exécuter à un temps précis | Facilité de lier les conteneurs |
| Basé sur les états | composé de services eux-mêmes composés d’images, volumes, ports, etc. |
| Des centaines de modules déjà intégrés par défaut | docker-compose up pour démarrer l’application |
| Automatisation des applications | Défini les services de l’application |
| Configuration puissante | Plusieurs environnements isolés sur un même hôte |
| Simple | Persistance des données des volumes |
| Puissant | Seulement les volumes modifiés sont recréés |
| Facile à apprendre | Passer du modèle à l'application en quelques minutes |
| Flexible | Configuration rapide de l'environnement de développement |
|  | Configuration de yaml simple |
|  | Installation simple |

**Question de rétroaction**

**Q1. Combien de temps avez-vous passé sur le travail pratique, en heures-personnes, en sachant que deux personnes travaillant pendant trois heures correspondent à six heures-personnes. Est-ce que l'effort demandé pour ce laboratoire est adéquat ?**

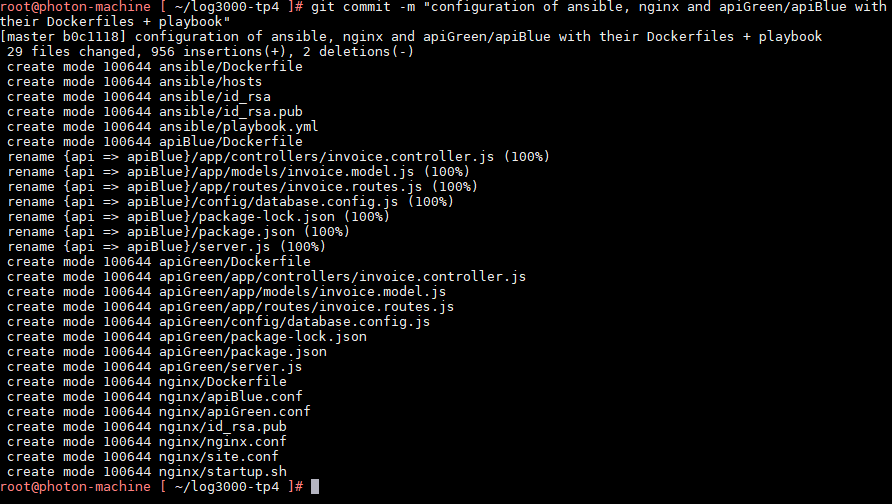
Nous avons passé environ 12 heures sur ce travail pratique. Il était intéressant d’intégrer ce que nous avions appris au TP4 (Docker) avec Ansible dans ce TP afin de voir comment les deux technologies peuvent offrir des possibilités intéressantes. Par contre, le travail était peu encadré et la documentation sur les déploiements bleu et vert était limitée et nous n’avons pas réussi à faire le malgré un nombre d’heures passées supérieur au travaux pratique précédents.

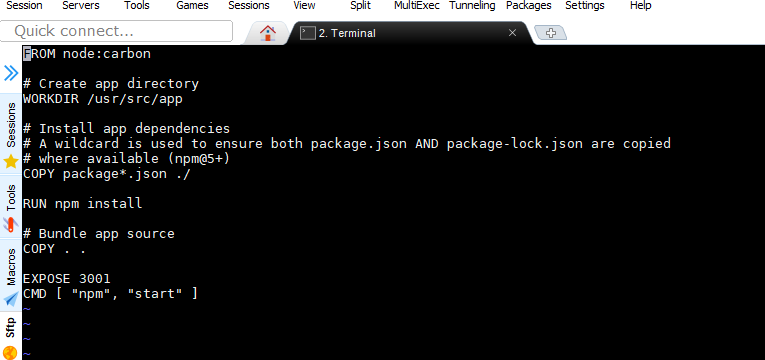
**Q2. Quelles difficultés avez vous rencontré lors de ce laboratoire?**

La configuration de la machine virtuelle nous avait causé bien des problèmes au dernier TP mais c’était la même procédure pour ce travail alors nous n’avons pas eu de problème à ce niveau. Le seul problème rencontré à été de coder le travail pratique puisque malgré une compréhension d’Ansible en théorie nous ne comprenions pas vraiment comment réaliser le déploiement bleu vert et nous avons perdu plusieurs heures sans vraiment avancer le travail. Il n’y avait pas beaucoup d’exemples en ligne et peu d’informations dans l’énoncé. Nous avons passé beaucoup de temps sur le travail et avons essayé plusieurs moyens d’arriver au résultat désiré sans succès. Cela dit, nous avons mis une structure de fichier de base permettant d’utiliser Docker et les playbooks d’Ansibles.

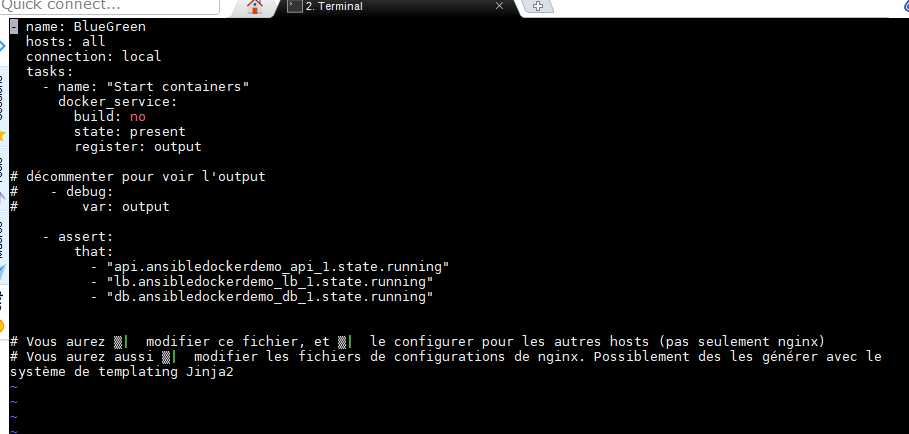
**ANNEXE**

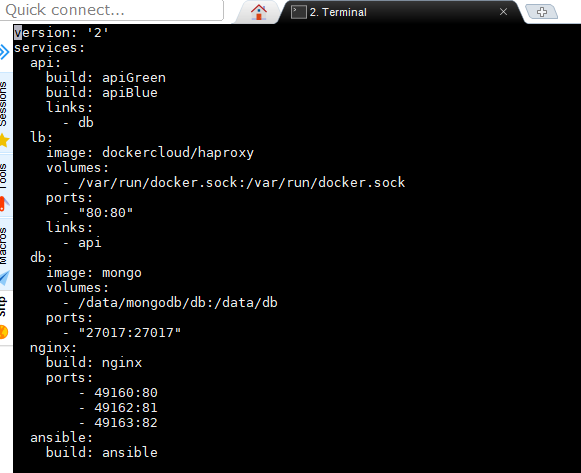
* **Commit:**

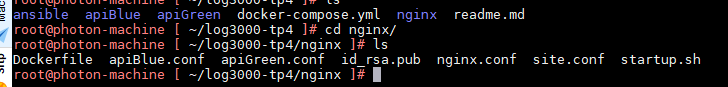
****

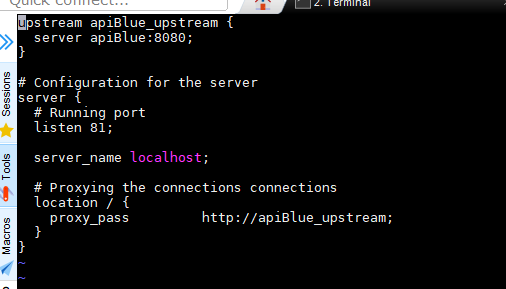
****

****

****

****

****

****

**RÉFÉRENCES**

**[1]** <https://www.fierdecoder.fr/2017/05/ansible-loutil-coup-de-coeur-pour-gerer-son-infrastructure/>