

# Formation Ansible, automatiser la gestion des serveurs – Verspieren

## Correction des travaux pratiques

Matthieu Falce

Octobre 2025

## 1 Ansible

### 1.1 Vagrant pour créer une machine

Réponses :

1. on peut générer un fichier de configuration avec vagrant : `vagrant ssh-config > ssh.config`
2. on indique à ansible dans l'inventaire d'utiliser la configuration ssh pour se connecter aux VM créées avec vagrant.
3. on utilise `ansible -i inventory.cfg all -m setup -a 'gather_subset=!all,!min,network'` pour aller lire les informations depuis les `facts` ansible, sinon on peut aller lire depuis des commandes shell précises : `ansible -i inventory.cfg all -a "df -h"` ou `ansible -i inventory.cfg all -a "free"`

### 1.2 Architecture 3 tiers

Réponses :

1. les rôles permettent de mieux encapsuler les différentes tâches, en proposant une organisation logique et de l'arborescence de fichiers
  - sans rôle nous devons lancer des playbooks différents selon chaque types de machine s dans l'inventaire
  - avec le playbook, on peut n'en lancer qu'un
2. l'idempotence permet de lancer plusieurs fois le playbook, sans réfléchir aux conséquences (une ligne n'est rajoutée dans un fichier que si elle n'y est pas déjà). C'est une propriété très importante d'**ansible** qui permet d'avoir plus confiance dans les déploiements
3. copy effectue une copie simple alors que **template** permet de changer des variables à l'intérieur du fichier
  - pour le `zshrc` : une copie peut suffire
  - pour créer un fichier HTML avec le `hostname` du serveur, un **template** est plus adapté (un seul fichier qui aura la variable remplacée différemment sur chaque machines)
4. pour ne plus avoir les vaches, il faut (au choix) :
  - `export ANSIBLE_NOCOWS=1` dans le terminal
  - rajouter `nocows=1` dans `ansible.cfg`

### 1.3 Gestion de l'inventaire

Réponses :

1. on peut utiliser le langage **ruby** dans les **Vagrantfile**. Ainsi, pour créer plusieurs machines, il suffit de créer une boucle et de lancer autant de machine identiques que l'on veut.
2. Par défaut, **Ansible** ne s'exécute que sur 5 machines en parallèle. Pour le lancer en parallèle sur toutes les machines, on modifie le paramètre de **fork**
  - soit dans la ligne de commande en rajoutant `-f NB_MACHINES`
  - soit dans le `ansible.cfg` en rajoutant `forks = NB_MACHINES`

3. on utilise `--list-hosts`
4. la `strategy free` va permettre à chaque machine de s'exécuter à sa vitesse, sans se synchroniser avec les autres. Ainsi, dans notre exemple, les machines les plus puissantes ne vont pas attendre les autres et finir plus vite.

## 1.4 Projet – Installation de wordpress

Il existe un exemple de playbook d'exemple d'installation de **Wordpresss** sur le site github de **Ansible**<sup>1</sup>. Il y a également de nombreux projets sur **galaxy** pour installer cet outils.

L'idée est de s'en inspirer, tout en comprenant ce qu'ils font.

Vous trouverez une correction dans le dossier **projet**.

---

1. <https://github.com/ansible/ansible-examples/tree/master/wordpress-nginx>