



Automatisation avec Ansible

Organiser les opérations de déploiement

7 novembre 2024

Table des matières

1	Introduction	3
1.1	Rôle	3
1.2	Tâche	3
1.3	YAML	3
2	Structurer le déploiement avec les rôles Ansible	4
2.1	Identifier les étapes pour installer MediaWiki	4
2.2	Transposer les étapes en rôles Ansible	4
3	Créer le premier rôle	5
3.1	Où placer les rôles ?	5
3.2	Créer le répertoire qui contiendra tous les rôles	5
3.3	Utiliser ansible-galaxy pour créer automatiquement un rôle	6
4	Créer l'arborescence des rôles MediaWiki	8
4.1	Créer le rôle Apache	8
4.2	Créer le rôle mariadb	8
4.3	Créer le rôle mediawiki	8
4.4	Créer le rôle commun	8
4.5	Créer le rôle confdb	9
4.6	Créer le rôle confapache	9
5	Améliorer l'inventaire en vue du déploiement	10

1 Introduction

On a pris connaissance du **contexte** dans lequel on va travailler, et on a mis en place un **environnement** technique favorable pour **automatiser** le **déploiement** de **MediaWiki** à l'aide d'**Ansible**.

Dans ce module, on va transposer et organiser les opérations nécessaires à l'installation de MediaWiki, à l'aide des rôles Ansible.

Le node manager et les nodes sont maintenant opérationnels. Il nous reste à l'installer le gestionnaire de wiki **MediaWiki** sur deux serveurs, afin. On va avoir besoin d'organiser les opérations, afin de les exécuter séquentiellement avec **Ansible**.

1.1 Rôle

Un **rôle** est une **structure arborescente** constituée de **répertoires** et de **fichiers de configuration** **YAML**, qui vont avoir pour fonction d'installer tel ou tel système. Les rôles peuvent être imbriqués et interdépendants les uns des autres. Un rôle est donc un ensemble de fichiers organisés dans une structure arborescente.

Le but des rôles est de pouvoir agglomérer des opérations cohérentes (dans les fichiers YAML), afin de pouvoir les réutiliser de façon modulaire. On peut voir un rôle comme un ensemble d'opérations qui ont un *rôle* commun, comme par exemple le rôle d'installer Apache, ou le rôle de configurer MariaDB.

Les **répertoires** sont tous optionnels, excepté le répertoire **tasks** qui doit contenir le fichier **main.yml**. Ansible va traiter en premier ce fichier à l'appel d'un rôle.

Certains répertoires doivent **obligatoirement** contenir un fichier **main.yml** pour être pris en compte.

1.2 Tâche

Une tâche est une instruction décrite en YAML dans un fichier de configuration. Chaque tâche utilise un module ainsi que quelques éventuels arguments supplémentaires.

De façon schématique, on peut retenir que :

- Un **rôle** contient un ou plusieurs **fichiers de configuration** (YAML) ;
- Un **fichier de configuration** contient une ou plusieurs **tâches**;
- Une **tâche** fait appel à un **module**.

1.3 YAML

[YAML](#) (*Yet Another Markup Language*) permet d'écrire des structures de données qui peuvent être spécifiées sous la forme de listes.

2 Structurer le déploiement avec les rôles Ansible

2.1 Identifier les étapes pour installer MediaWiki

On a déjà vu que l'installation de MediaWiki nécessitait **6 étapes**:

- 1) Installer le service web **Apache** sur le serveur **web1**.
- 2) Installer **PHP** sur le serveur **web1**.
- 3) Installer une base de données **MariaDB** sur le serveur **bd1**.
- 4) Télécharger les **fichiers sources de MediaWiki** et les mettre sur le **serveur web web1**.
- 5) Configurer le **serveur web web1** pour pointer vers l'URL **http://web1/mediawiki**.
- 6) Finaliser l'installation de MediaWiki via **le script d'installation** sur **web1**.

Ces **6 étapes** comportent des **opérations d'installation** (étapes 1 à 3) et des **opérations de configuration** (étapes 4 à 6).

2.2 Transposer les étapes en rôles Ansible

On va maintenant transposer cette logique à Ansible en utilisant les rôles. On va créer 5 rôles :

- 1) Un rôle pour **installer** Apache et PHP : **apache**.
- 2) Un rôle pour **installer** MariaDB : **mariadb**.
- 3) Un rôle pour **configurer** Apache pour MediaWiki : **confapache**.
- 4) Un rôle pour **configurer** MariaDB pour MediaWiki : **confdb**.
- 5) Un rôle qui contiendra les **variables globales** : **commun**.

On va donc suivre cette liste pour créer chaque rôle un par un.

3 Créer le premier rôle

On va commencer par créer une **arborescence** de fichiers de configuration qui permettra de dérouler les étapes de déploiement de MediaWiki et dans le module suivant, on va écrire les **scripts** Ansible dans ces fichiers de configuration.

3.1 Où placer les rôles ?

Par défaut, Ansible va chercher les rôles dans le **répertoire roles** qui est placé dans le **répertoire de travail** (répertoire courant) : celui dans lequel les commandes Ansible sont lancées.

3.2 Créer le répertoire qui contiendra tous les rôles

On commence par créer le répertoire **roles** dans notre espace de travail. Ce répertoire contiendra tous les rôles qu'on va créer.

```
root@debian:~# su - user-ansible
```

```
user-ansible@debian:~$ source ansible10.5.0/bin/activate
```

```
(ansible10.5.0) user-ansible@debian:~$ mkdir roles
```

```
(ansible10.5.0) user-ansible@debian:~$ cd roles/
```

```
(ansible10.5.0) user-ansible@debian:~/roles$
```

Il est possible d'indiquer à Ansible un chemin différent en modifiant la variable d'environnement **DEFAULT_ROLES_PATH**, ou en modifiant l'option **roles_path** (par défaut /etc/ansible/roles) dans le fichier de configuration Ansible (/etc/ansible/ansible.cfg).

Pour créer une **arborescence** complète, on peut utiliser la commande **ansible-galaxy**.

Ansible Galaxy fait référence au [site web de Galaxy](#) où les utilisateurs peuvent partager des rôles (de la même façon que GitHub propose aux développeurs de stocker et de partager, publiquement ou non, le code qu'ils créent).

ansible-galaxy est un outil de ligne de commande pour télécharger, créer et gérer les rôles Ansible. À titre d'exemple, Il est possible d'utiliser un rôle de la Galaxy pour installer Apache.

3.3 Utiliser ansible-galaxy pour créer automatiquement un rôle

Créer le premier rôle **pour l'installation d'Apache** avec la commande **ansible-galaxy** et l'option **init** :

```
(ansible10.5.0) user-ansible@debian:~/roles$ ansible-galaxy init apache  
- Role apache was created successfully  
(ansible10.5.0) user-ansible@debian:~/roles$
```

Installer tree :

```
root@debian:~# apt install tree  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  tree  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 52.5 kB of archives.  
After this operation, 116 kB of additional disk space will be used.  
Get:1 http://deb.debian.org/debian bookworm/main amd64 tree amd64 2.1.0-1  
[52.5 kB]  
Fetched 52.5 kB in 0s (342 kB/s)  
Selecting previously unselected package tree.  
(Reading database ... 41080 files and directories currently installed.)  
Preparing to unpack .../tree_2.1.0-1_amd64.deb ...  
Unpacking tree (2.1.0-1) ...  
Setting up tree (2.1.0-1) ...  
Processing triggers for man-db (2.11.2-2) ...  
root@debian:~#
```

Utiliser la commande **tree** pour afficher l'arborescence des fichiers.

```
(ansible10.5.0) user-ansible@debian:~/roles$ tree apache/  
apache/  
├── defaults  
│   └── main.yml  
├── files  
├── handlers  
│   └── main.yml  
├── meta  
│   └── main.yml  
├── README.md  
├── tasks  
│   └── main.yml  
├── templates  
├── tests  
│   ├── inventory  
│   └── test.yml  
└── vars  
    └── main.yml  
  
9 directories, 8 files  
(ansible10.5.0) user-ansible@debian:~/roles$
```

L'arborescence contient les répertoires suivants :

defaults	Valeurs par défaut (le fichier main.yml est obligatoire) avec une priorité moindre.
files	Tous les fichiers à copier sur le node.
handlers	Contient les tâches à exécuter après une notification(même chose pour les instructions handlers (le fichier main.yml est obligatoire)
meta	Dépendances du rôle et informations (auteur, licence, plateformes...) sur le rôle (le fichier main.yml est obligatoire)
tasks	Liste des instructions à exécuter (le fichier main.yml est obligatoire)
templates	Tous les fichiers de template Jinja
tests	Pour les tests
vars	Fichier contenant des déclarations de variables (le fichier main.yml est obligatoire) ; les variables définies ici sont prioritaires par rapport aux variables définies dans l'inventaire

À la création de l'arborescence, **les fichiers sont vides** ! Il conviendra de les compléter en fonction des opérations à réaliser.

On a compris qu'un rôle est avant tout une arborescence de fichiers qui contiendra des actions Ansible.

4 Créer l'arborescence des rôles MediaWiki

On va créer, un à un, la structure des 6 rôles pour le déploiement de MediaWiki.

4.1 Créer le rôle Apache

On a déjà créé la structure du rôle **Apache**.

Pour installer **Apache et PHP**, on a besoin d'installer des paquetages et activer le service Apache.

4.2 Créer le rôle mariadb

Pour le rôle **mariadb**, on a besoin d'installer des paquetages et démarrer un service.

On va créer le rôle **mariadb** :

```
(ansible10.5.0) user-ansible@debian:~/roles$ ansible-galaxy init mariadb  
- Role mariadb was created successfully
```

Le fichier **main.yml** contiendra les tâches à exécuter pour installer MariaDB

4.3 Créer le rôle mediawiki

Comme les opérations de configuration et les variables concernent spécifiquement le déploiement de MediaWiki, les rôles de configuration et commun vont être placés spécifiquement dans un répertoire **mediawiki**.

```
(ansible10.5.0) user-ansible@debian:~/roles$ ansible-galaxy init mediawiki  
- Role mediawiki was created successfully
```

4.4 Créer le rôle commun

Le rôle **commun** contiendra des **variables partagées** entre les rôles **confapache** et **confdb**. Ainsi, au lieu de définir les mêmes variables à deux endroits différents, il est préférable de créer un rôle commun, et d'utiliser une dépendance avec les autres rôles.

Pour cela, on a besoin du répertoire **defaults** et du fichiers **main.yml** qui contiendront les **variables globales**:

```
(ansible10.5.0) user-ansible@debian:~/roles$ ansible-galaxy init commun  
- Role commun was created successfully
```


4.5 Créer le rôle confdb

Pour le rôle **confdb**, on a besoin de **créer une base de données** et d'**attribuer des droits** sur cette base.

On doit aussi créer une dépendance avec le rôle **commun** pour partager des **variables globales**. Pour cela, on a besoin des répertoires **meta** et **tasks**.

```
(ansible10.5.0) user-ansible@debian:~/roles$ ansible-galaxy init confdb  
- Role confdb was created successfully
```

4.6 Créer le rôle confapache

On va partager des **variables globales** avec le rôle **confdb**, alors on ajoute une **dépendance** avec le rôle **commun**.

On va également créer le **répertoire d'installation** de MediaWiki, **télécharger** les fichiers MediaWiki sur le site officiel, lancer le **script d'installation** et **mettre à jour** la base de données.

Pour cela, on a besoin des répertoires **meta** et **tasks**:

```
(ansible10.5.0) user-ansible@debian:~/roles$ ansible-galaxy init confapache  
- Role confapache was created successfully
```

Ce qui donne l'arborescence suivante :

```
(ansible10.5.0) user-ansible@debian:~/roles$ ls -l  
total 24  
drwxr-xr-x 10 user-ansible user-ansible 4096 Oct 31 12:37 apache  
drwxr-xr-x 10 user-ansible user-ansible 4096 Oct 31 12:38 commun  
drwxr-xr-x 10 user-ansible user-ansible 4096 Oct 31 12:40 confapache  
drwxr-xr-x 10 user-ansible user-ansible 4096 Oct 31 12:39 confdb  
drwxr-xr-x 10 user-ansible user-ansible 4096 Oct 31 12:37 mariadb  
drwxr-xr-x 10 user-ansible user-ansible 4096 Oct 31 12:37 mediawiki
```

5 Améliorer l'inventaire en vue du déploiement

Toujours dans l'esprit de **modularité**, on va modifier le fichier **inventaire.ini** pour séparer les deux **nodes** dans deux **groupes** distincts.

Ce qui permettra par la suite de s'adresser à un **groupe** plutôt qu'aux **nodes** directement. De cette façon, si on veut ajouter un node, on a simplement à ajouter le node dans le bon groupe dans le fichier inventaire, et à rejouer vos scripts sans rien changer d'autre.

Se connecter sur le node manager :

```
root@debian:~# su - user-ansible
```

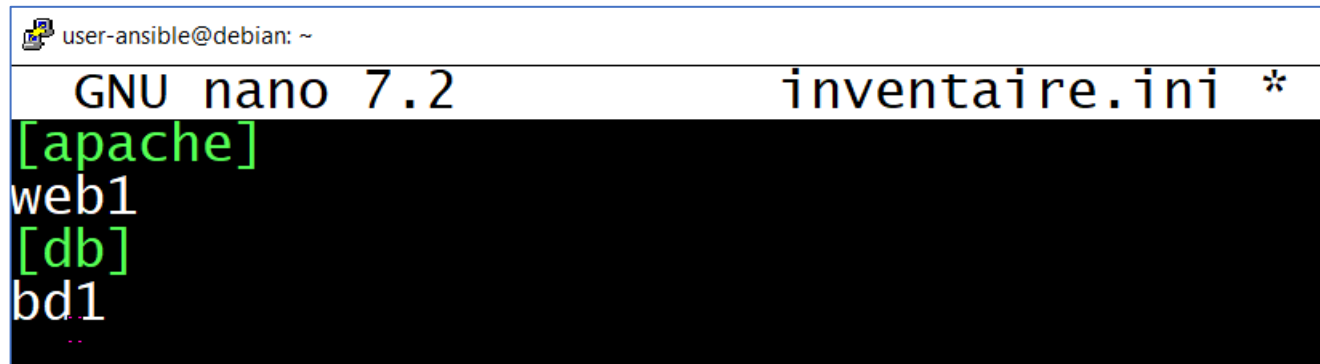
Activer l'environnement virtuel :

```
user-ansible@debian:~$ source ansible10.5.0/bin/activate  
(ansible10.5.0) user-ansible@debian:~$
```

Éditer le fichier **inventaire.ini**

```
(ansible10.5.0) user-ansible@debian:~$ nano inventaire.ini
```

```
[apache]  
web1  
[db]  
bd1
```



```
user-ansible@debian: ~  
GNU nano 7.2                               inventaire.ini *  
[apache]  
web1  
[db]  
bd1
```

On a deux groupes entre crochets qui contiennent un node chacun.

On déclare un groupe de serveurs en mettant le nom du groupe entre crochets [].

Le fichier inventaire est très souple, il permet des combinaisons intéressantes et surprenantes :

- node[1:2] pour 2 nodes
- [linux:children] pour un groupe qui contient d'autres groupes