

TP 7 - Les rôles

Objectif

Bien qu'il soit possible d'écrire un playbook dans un fichier comme nous l'avons fait dans les labs précédents, vous souhaitez éventuellement réutiliser des fichiers et commencer à organiser les choses.

Lorsque vous créez un rôle, vous déconstruisez votre playbook en parties et ces parties se trouvent dans une structure de répertoires.

Ce Lab couvrira:

- la structure des dossiers d'un rôle Ansible
- comment construire un rôle Ansible
- création d'un playbook pour utiliser et exécuter un rôle

Création d'un rôle de base

Ansible recherche les rôles dans un sous-répertoire appelé `roles` dans le répertoire du projet. Cela peut être remplacé dans la configuration Ansible. Chaque rôle a son propre répertoire. Pour faciliter la création d'un nouveau rôle, l'outil `ansible-galaxy` peut être utilisé.

Astuce

Ansible Galaxy est votre hub pour trouver, réutiliser et partager le meilleur contenu Ansible. La commande `ansible-galaxy` aide à interagir avec Ansible Galaxy. Pour l'instant, nous allons simplement l'utiliser comme aide pour construire la structure du répertoire.

Nous allons créer un rôle qui installe et configure Apache pour servir un `virtual host`.

- Exécutez ces commandes dans votre répertoire `~/ansible-files`:

```
mkdir roles
ansible-galaxy init --offline roles/apache_vhost
```

- Jetez un œil aux répertoires de rôles et à leur contenu.

Création du fichier de tâches

Le fichier `main.yml` dans le sous-répertoire `tasks` du rôle doit faire ce qui suit :

- S'assurer que le paquet `apache2` est installé
- S'assurer que le service `apache2` est démarré et activé
- Mettre du contenu HTML dans la racine du document Apache
- Installer le template fourni pour configurer le vhost

AVERTISSEMENT

*Le `main.yml` (et d'autres fichiers éventuellement inclus par `main.yml`) ne peut contenir que des tâches, et non de *Playbook complet*!*

Accédez au répertoire `roles/apache_vhost` et éditez le fichier `tasks/main.yml`.

Ajoutez uniquement les tâches suivantes :

- Installez le package `apache2` à l'aide du module `apt`
- Ensuite, ajoutez les deux tâches supplémentaires suivantes pour garantir une structure de répertoire vhost.

```
- name: ensure vhost directory is present
  file:
    path: "/var/www/{{ ansible_hostname }}"
    state: directory
- name: deliver html content
  copy:
    src: web.html
    dest: "/var/www/{{ ansible_hostname }}/index.html"
```

Notez que le répertoire vhost est créé en utilisant le module `file`.

- La dernière tâche que nous ajoutons utilise le module `template` pour créer le fichier de configuration vhost à partir d'un modèle `j2`:

```
- name: template vhost file
  template:
    src: vhost.conf.j2
    dest: /etc/apache2/sites-enabled/vhost.conf
    owner: root
    group: root
    mode: 0644
  notify:
    - restart_apache2
```

Notez qu'il utilise un handler pour redémarrer apache2 après une mise à jour de configuration.

Création du handler

- Créez le handler dans le fichier `handlers/main.yml` pour redémarrer apache2 lorsqu'il est notifié par la tâche de template. Ajoutez uniquement la tâche de handler dans ce fichier.

```
- name: restart_apache2
  service:
    name: apache2
    state: restarted
```

Création des templates

Créez le contenu HTML qui sera servi par le serveur Web.

- Créez un fichier `web.html` dans le répertoire "src" du rôle, files:

```
mkdir ~/ansible-files/roles/apache_vhost/files
echo 'simple vhost index' > ~/ansible-files/roles/apache_vhost/files/web.html
```

- Ajoutez au fichier `templates/vhost.conf.j2` du rôle `apache_vhost`, le contenu suivant :

```
<VirtualHost *:80>
  ServerAdmin webmaster@{{ ansible_fqdn }}.fr
  ServerName www.{{ ansible_fqdn }}.fr
  DocumentRoot /var/www/{{ ansible_hostname }}/
</VirtualHost>
```

Test du rôle

Vous êtes prêt à tester le rôle sur node2. Mais comme un rôle ne peut pas être attribué directement à un nœud, créez d'abord un playbook qui connecte le rôle et l'hôte.

- Créez le fichier `test_apache_role.yml` dans le répertoire `~/ansible-files`:

```
---
- name: use apache_vhost role playbook
  hosts: node2
  become: true

  pre_tasks:
    - debug:
        msg: 'Beginning web server configuration.'

  roles:
    - apache_vhost

  post_tasks:
    - debug:
        msg: 'Web server has been configured.'
```

Notez les mots clés `pre_tasks` et `post_tasks`. Normalement, les tâches des rôles s'exécutent avant les tâches d'un playbook. Pour contrôler l'ordre d'exécution, des `pre_tasks` sont effectuées avant l'application des rôles. Les `post_tasks` sont effectuées une fois tous les rôles terminés.

- Maintenant vous pouvez exécuter votre playbook
- Ajoutez la ligne suivante au fichier `/etc/hosts` de votre machine windows.

```
192.168.201.12    www.node2.fr
```

- Testez avec le navigateur pour confirmer que le rôle a fonctionné (vérifiez le port : 80 ou 8080) :

```
www.node2.fr
```