

TP 2 – Exécution de commandes Ad-hoc

Objectif

Pour la première utilisation d'ansible, vous allez exécuter quelques commandes Ad-hoc pour vous aider à comprendre comment il fonctionne. Les commandes Ad-Hoc Ansible vous permettent d'effectuer des tâches sur des nœuds distants sans avoir à écrire un playbook. Ils sont très utiles lorsque vous devez simplement faire une ou deux choses rapidement et souvent, sur de nombreux nœuds distants.

Cet TP couvrira

- Localisation et compréhension du fichier de configuration Ansible (`ansible.cfg`)
- Localisation et compréhension d'un fichier d'inventaire au format "ini"
- Exécution de commandes ad hoc

Travailler avec votre inventaire

Pour utiliser Ansible, vous devez fournir un fichier d'inventaire qui définit une liste d'hôtes à gérer. L'inventaire est un fichier au format ini répertoriant vos hôtes, trié en groupes, fournissant en outre certaines variables.

- Copiez le contenu ci-dessous dans le fichier d'inventaire `"$HOME/lab_inventory/hosts"`, et n'oubliez pas de créer le répertoire `lab_inventory` avant.

[web]

node1 ansible_host=192.168.201.11 ansible_user=vagrant ansible_ssh_private_key_file=~/.ssh/id_rsa

node2 ansible_host=192.168.201.12 ansible_user=vagrant ansible_ssh_private_key_file=~/.ssh/id_rsa

[control]

ansible ansible_host=192.168.201.1 ansible_connection=local

Un fichier d'inventaire peut contenir beaucoup plus d'informations, il peut organiser vos hôtes en groupes ou définir des variables.

Dans cet environnement, l'inventaire actuel a les groupes «web» et «control».

- Pour référencer les hôtes d'inventaire, vous fournissez un modèle d'hôte à la commande *ansible*. Ansible a une option *--list-hosts* qui peut être utile pour clarifier quels hôtes gérés sont référencés par le modèle d'hôte dans une commande ansible.

```
ansible -i /$HOME/lab_inventory/hosts node1 --list-hosts
ansible -i /$HOME/lab_inventory/hosts 'node*' --list-hosts
ansible -i /$HOME/lab_inventory/hosts web --list-hosts
ansible -i /$HOME/lab_inventory/hosts all --list-hosts
```

Astuce

L'inventaire peut contenir plus de données. Par exemple, si vous avez des hôtes qui s'exécutent sur des ports SSH non standard, vous pouvez mettre le numéro de port après le nom d'hôte avec deux points. Ou vous pouvez définir des noms spécifiques à Ansible et les faire pointer vers la "vraie" IP ou le nom d'hôte.

Les fichiers de configuration Ansible

Le comportement d'Ansible peut être personnalisé en modifiant les paramètres du fichier de configuration d'Ansible (format ini) : **ansible.cfg**

Astuce

La pratique recommandée est de créer un fichier `ansible.cfg` dans le répertoire à partir duquel vous exécutez les commandes Ansible. Ce répertoire contiendrait également tous les fichiers utilisés par votre projet Ansible, tels que l'inventaire et les playbooks. Une autre pratique recommandée est de créer un fichier `".ansible.cfg"` dans votre répertoire personnel.

- Dans le répertoire d'accueil de votre management ansible, créez un fichier `"~/.ansible.cfg"` dans le répertoire personnel de l'utilisateur courant et remplissez-le avec les détails suivants :

```
[defaults]
connection = smart
Timeout = 60
deprecation_warnings = False
host_key_checking = False
retry_files_enabled = False
inventory = $HOME/lab_inventory/hosts
```

La plupart des options de configuration ne sont pas intéressantes ici, mais assurez-vous de noter la dernière ligne: l'emplacement de l'inventaire est indiqué. C'est ainsi qu'Ansible savait dans les commandes précédentes à quelles machines se connecter.

Plusieurs options de configuration sont fournis. La plupart d'entre eux ne sont pas intéressantes ici, mais assurez-vous de noter la dernière ligne: l'emplacement de l'inventaire est indiqué. C'est ainsi qu'Ansible savait dans les commandes précédentes à quelles machines se connecter.

ping d'un hôte

Le module "*ping*" s'assure que les hôtes cibles sont accessible. Fondamentalement, il se connecte à l'hôte géré, y exécute un petit script et collecte les résultats. Cela garantit que l'hôte géré est accessible et qu'Ansible est capable d'exécuter correctement les commandes sur celui-ci.

Astuce

Considérez un module comme un outil conçu pour accomplir une tâche spécifique.

- Ansible doit savoir qu'il doit utiliser le module *ping*: L'option *-m* définit le module ansible à utiliser. Les options peuvent être passées au module spécifié en utilisant l'option *-a*.

En utilisant le module "*ping*", Vérifiez la connectivité des hôtes du groupe web.

```
ansible web -m ping
```

Comme vous le voyez, chaque nœud donne un resultat positif - ici "pong".

Liste des modules et obtention d'aide

Ansible est livré avec de nombreux modules par défaut.

- Exécutez la commande suivante pour répertorier tous les modules ansible :

```
ansible-doc -l
```

- Utilisez *grep* pour trouver le module *user*.

```
ansible-doc -l | grep -i user
```

- Pour obtenir de l'aide pour un module spécifique on utilise la commande *ansible-doc* suivie par le nom du module.
Affichez l'aide du module "user".

Astuce

Les options obligatoires sont marquées d'un "=" dans `ansible-doc`.

Utilisation du module « `command` »

Voyons maintenant comment exécuter une bonne commande Linux à l'ancienne à l'aide du module `command`. Celui-ci exécute simplement la commande spécifiée sur un hôte géré.

- Appliquez la tâche qui exécute la commande `id` sur le nœud `node1`.

```
ansible node1 -m command -a "id"
```

Dans ce cas, le module est appelé "`command`" et l'option passée avec "-a" est la commande réelle à exécuter.

- Essayez d'exécuter cette commande Ad-hoc sur tous les hôtes gérés en utilisant le modèle d'hôte `all`, en affichant les versions du noyau que vos hôtes :

```
ansible all -m command -a 'uname -r'
```

Astuce

Comme de nombreuses commandes Linux, `ansible` permet des options de forme longue aussi bien que de forme courte. Par exemple, `ansible web --module-name ping` est identique à l'exécution de `ansible web -m ping`. Nous allons utiliser les options abrégées tout au long de ces TP.

Les modules de copie et les autorisations

En utilisant le module `copy`, exécutez une commande Ad-hoc sur `node1` pour changer le contenu du fichier `/etc/motd`. **Le contenu est remis au module via une option dans ce cas.**

- Exécutez ce qui suit, mais **attendez-vous à une erreur** :

```
ansible node1 -m copy -a 'content="Managed by Ansible\n" dest=/etc/motd'
```

Comme mentionné, cela produit une **erreur** :

La sortie de la commande Ad-hoc vous crie **FAILED** en rouge. Pourquoi ? Parce que l'utilisateur **vagrant** n'est pas autorisé à écrire le fichier `motd`.

Vous devez demander à Ansible d'utiliser *sudo* pour exécuter la commande en tant que root en utilisant le paramètre *-b* (pensez "become").

Astuce

*Ansible se connectera aux machines en utilisant votre nom d'utilisateur actuel (vagrant dans ce cas), tout comme SSH le ferait. Pour remplacer le nom d'utilisateur distant, vous pouvez utiliser le paramètre *-u*.*

Il est normal de se connecter en tant que **vagrant** car *sudo* est configuré.

- Modifiez la commande pour utiliser le paramètre *-b* et exécutez à nouveau :

```
ansible node1 -m copy -a 'content="Managed by Ansible\n" dest=/etc/motd' -b
```

Cette fois, la commande a été exécutée avec succès.

- Utilisez *ansible* avec le module générique "command" pour vérifier le contenu du fichier */etc/motd* :

```
ansible node1 -m command -a 'cat /etc/motd'
```

- Exécutez à nouveau la commande *ansible node1 -m copy...* Notez:
 - La couleur de sortie différente (configuration de terminal appropriée fournie).
 - Le changement de "changed": true, à "changed": false,.
 - La première ligne indique SUCCESS au lieu de CHANGED.

Astuce

Cela permet de repérer plus facilement les changements et ce qu'Ansible a réellement fait.