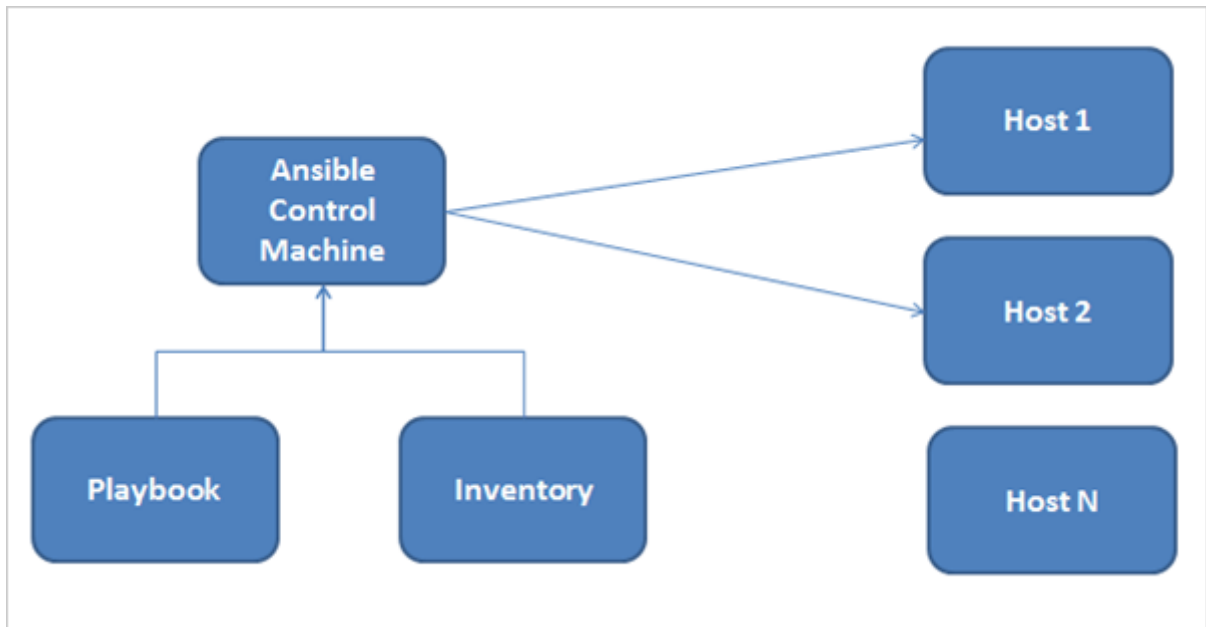


Lab2 _ commandes Ansible ad hoc

Utilisation d'Ansible



Ansible se compose de 3 composants principaux

- Machine de contrôle
- Inventaire
- Playbook

La machine de contrôle gère l'exécution du Playbook. Il peut être installé sur votre ordinateur portable ou sur n'importe quelle machine sur Internet.

Le fichier d'inventaire fournit une liste complète de toutes les machines cibles sur lesquelles différents modules sont exécutés en établissant une connexion ssh et en installant les logiciels nécessaires.

Le playbook se compose d'étapes que le mécanisme de contrôle exécutera sur les serveurs définis dans le fichier d'inventaire.

Commande ad hoc

- peu utilisé (en proportion) au profit de ansible-playbook
- permettre du test (ping, inventaire)
- permet de jouer des tâches
- beaucoup d'options similaires à la commande ansible-playbook

Syntaxe :

\$ ansible hostORgroup -m module_name -a 'arguments' -u username --become

Principales options à connaître :

- * -u : user distant utilisé
- * -b : passer les commandes en élévation de privilèges(sudo)
- * -k ou --ask-pass > password SSH
- * -K ou --ask-become-pass > password pour élévation privilèges
- * -D ou --diff : avoir un output de la diff
- * --key-file : lien direct vers la clef privée
- * -e ou --extra-vars : définir des variables
- * --ask-vault-pass : déchiffrer un secret vault
- * --vault-password-file : fichier pour déchiffrer
- * -vvv : verbose

Exemples de commandes ad hoc:

* affichage oneline

\$ ansible all -u ansible -m ping --one-line

* module command

\$ ansible all -u ansible -m command -a uptime

* passage d'une variable

\$ ansible all -b -e "var1=mavariabale" -m debug -a 'msg={{ var1 }}'

Modules Ansible

Les modules sont les principaux blocs de construction d'Ansible et sont essentiellement des scripts réutilisables qui sont utilisés par les playbooks Ansible. Ansible est livré avec un certain nombre de modules réutilisables. Ceux-ci incluent des fonctionnalités de contrôle des services, l'installation de progiciels, l'utilisation de fichiers et de répertoires, etc.

Jetons un coup d'œil à certains des modules Ansible les plus populaires et à leur utilisation via les commandes ad-hoc et plus tard dans le playbook.

1) Module de configuration

Pour obtenir des informations sur le réseau ou le matériel ou la version du système d'exploitation ou des informations relatives à la mémoire, le module de configuration vous aidera à rassembler les mêmes informations sur les machines cibles. Sur le contrôle, la machine exécute la commande ci-dessous.

```
$ ansible webservers -m setup
```

2) Module de commande

Le module de commande exécute simplement une commande spécifique sur la machine cible et donne la sortie.

Certains des exemples sont donnés ci-dessous

```
$ ansible webservers -m command -a 'uptime'
```

```
$ ansible webservers -m command -a 'hostname'
```

3) Module Shell

Pour exécuter n'importe quelle commande dans le shell de votre choix, vous pouvez utiliser le module Shell. Les commandes du module shell sont exécutées dans / bin / sh shell et vous pouvez utiliser les opérateurs comme «>» ou «|» (symbole de tube ou même des variables d'environnement).

Donc, la principale différence entre le module Shell et le module de commande est que si vous n'avez pas besoin d'utiliser des opérateurs comme ceux mentionnés, vous pouvez utiliser le module de commande.

```
$ ansible webservers -m shell -a 'ls -l > temp.txt'
```

Sur les machines sous le groupe de serveurs Web, vérifiez le fichier créé et exécutez la commande pour afficher le fichier texte.

```
$ ansible webservers -m command -a 'cat temp.txt'
```

4) Module utilisateur

En utilisant ce module, on peut créer ou supprimer des utilisateurs.

Pour ajouter un utilisateur

```
$ ansible webservers -m user -a 'name=user1 password=user1' --become
```

Pour supprimer l'utilisateur

```
$ ansible webservers -m user -a 'name=user1 state=absent' -become
```

5) Module de fichiers

Ce module est utilisé pour créer des fichiers, des répertoires, définir ou modifier les autorisations et la propriété des fichiers, etc.

Exemple 1: Créer un fichier

```
$ ansible webservers -m file -a 'dest=/home/ansible/niranjana.txt state=touch mode=600 owner=ansible group=ansible'
```

Exemple 2: Créer un répertoire

Pour créer un répertoire à l'aide du module de fichiers, vous devez définir deux paramètres.

- Path (alias - name, dest) - C'est le chemin absolu du répertoire à créer.
- État - Vous devez saisir la valeur 'répertoire'. Par défaut, la valeur est 'fichier'.

```
$ ansible webservers -m file -a 'dest=/home/ansible/vndir state=directory mode=755'
```

Exemple 4: Supprimer un répertoire

Vous pouvez supprimer un répertoire en définissant la valeur du paramètre d'état sur absent . Le répertoire et tout son contenu seront supprimés.

```
$ ansible webservers -m file -a 'dest=/home/ansible/vndir state=absent'
```

6) Module de copie

Il est utilisé pour copier des fichiers sur plusieurs machines cibles.

```
$ ansible webservers -m copy -a 'src=sample.txt dest=/home/ansible/sample.txt'
```

7) Gestion des packages logiciels

Si vous devez installer des progiciels via «yum» ou «apt», vous pouvez utiliser les commandes ci-dessous.

Exemple 1: Installez GIT

```
$ ansible webservers -m apt -a "name=git state=present" -become
```

Exemple 2: Vérifiez si le package est installé et mettez-le à jour avec la dernière version.

```
$ ansible webserver -m apt -a "name=git state=latest"
```

Dans la commande ci-dessus, état = dernier mettra à jour le package vers la dernière version uniquement.

Exemple 3: Installez Apache Webserver

```
$ ansible webserver -m apt -a 'name=httpd state=present' --become
```

Exemple 4: Vérifiez si Maven est installé ou non.

```
$ ansible webserver -m apt -a 'name=maven state=absent' --become
```

8) Module de gestion des services

Pour gérer les services avec ansible, nous utilisons un module '**un service**'.

Démarrer un service

```
$ ansible webserver -m service -a "name=httpd state=started" --become
```

Arrêter un service

```
$ ansible webserver -m service -a "name=httpd state=stopped" --become
```

Redémarrer un service

```
$ ansible webserver -m service -a "name=httpd state=restarted" --become
```

Application

En utilisant Ansible en ligne de commande, vous allez réaliser les actions suivantes :

- * Créer les groupes `rennes`, `lille` et `paris`
- * Créer un utilisateur `core`
- * Modifier l'utilisateur `core` pour qu'il ait l'UID 10000
- * Modifier l'utilisateur `core` pour qu'il soit dans le groupe `rennes`
- * Installer le logiciel `tree`
- * Stopper le service `crond`
- * Désactiver le service `atd`
- * créer un fichier vide `/tmp/test` avec les droits 644

- * Mettre à jour le système sur la VM cliente
- * Redémarrer la VM cliente (une petite recherche s'impose !)

[WARNING]

====

Utilisez les modules appropriés plutôt que le module shell.
Pour rappel, les modules disponibles sont listés à cette
adresse :

http://docs.ansible.com/ansible/modules_by_category.html

====

[NOTE]

====

Vérifier sur le client que les actions sont bien effectuées.