

Cours 63 : Ansible, Puppet, Chef

Dans ce cours nous verrons ce qu'est Ansible, Puppet et Chef ainsi que leur fonctionnement. Nous ferons tout d'abord une introduction dans la configuration d'outils de gestion, puis nous verrons le fonctionnement de Ansible, Puppet, Chef.

Commençons tout d'abord par expliquer le concept de la configuration de drift.

Configuration drift se passe lorsque un changement individuel fait au fil du temps à cause de la configuration d'un appareil pour dévier depuis une configuration standard/correcte comme défini par la compagnie.

Chaque appareil aura une partie unique de sa configuration (Les adresses IP, les nom d'hôtes, etc.) la plupart de la configuration des appareils est défini dans un template standard désigné par le réseau d'architecte/ingénieurs de l'entreprise.

Lorsqu'un ingénieur individuel fait des changements sur l'appareil (par exemple pour résoudre et fixer un problème réseau, tester des configurations, etc.), la configuration d'un appareil peut dévier de son standard.

Les enregistrements de ces changements individuels et leurs raisons sont conservés, cela peut conduire à des problèmes futures.

Même sans outils d'automatisation, il est meilleur d'avoir des pratiques de gestion de configuration. Par exemple lorsqu'un changement est fait, sauvegarder la configuration comme fichier texte et le placer dans un dossier partagé.

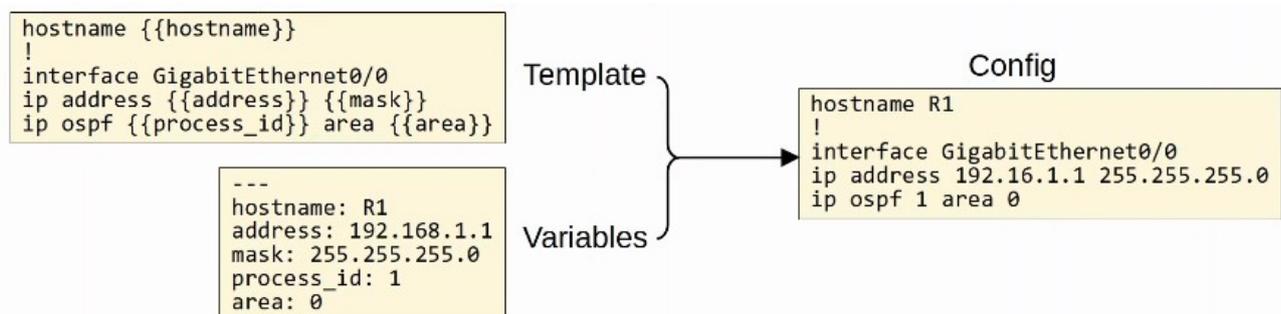
Un système de standard de nomage peut être utilisé par exemple : nom_prénom_date

Cela permet de suivre la trace des anciennes versions de la configuration et de voir quelles changements ont été fait. Il y a tout de même des défauts à ce système, comme lorsqu'un ingénieur peut oublier de placer la nouvelle configuration dans le fichier après avoir fait des changements, comment savoir lequel peut être considéré comme étant le bon fichier.

Même si la configuration est bien sauvegardé, cela ne garantie pas que la configuration corresponde bien au standard.

Le provisionnement de configuration se réfère à comment les changements de configuration sont appliqués aux appareils. Cela inclus configurer de nouveaux appareils aussi.

De manière traditionnelle, la configuration fournis est appliqué en connectant aux appareils un par un via SSH. Ce n'est pas pratique pour de grands réseaux. Des outils de gestion de configuration comme Ansible, Puppet et Chef permettent de faire des changements aux appareils à grande échelle en très peu de temps/effort. Il y a deux composants essentiels : templates et les variables pour former une configuration :



Voyons les différents outils de gestion de configuration. Les outils de gestion de configuration sont des outils d'automatisation qui facilitent un contrôle centralisé d'un grand nombre d'appareils. Ces outils ont été originellement développés après l'augmentation des Vms, pour permettre à un administrateur serveur système d'automatiser le processus de création, configuration et de suppression des Vms. Elles sont aussi largement utilisés pour gérer des appareils d'un réseau. Ces outils peuvent être utilisés pour faire des tâches comme par exemple :

- Générer des configurations pour de nouveaux appareils à grande échelle.
- Faire fonctionner des changement de configuration sur des appareils (tous les appareils ou une partie d'entre eux). Vérifier que la configuration des appareils est en accord avec les standards définis. Il est aussi possible de comparer des configurations entre plusieurs appareils, et entre différentes versions de configurations sur le même appareil.



Ansible est un outil de gestion de configuration appartenant à Red Hat.

Ansible est écrit en Python. Ansible est « agentless » cela signifie que cela ne requière pas de logiciel spécial pour être lancé dans des appareils de gestion.

Ansible utilise SSH pour connecter aux appareils, faire des changements de configuration, extraire des informations, etc.

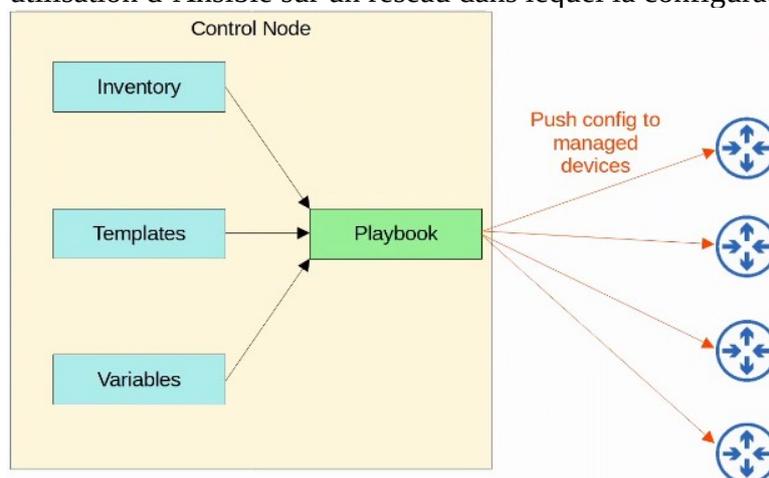
Ansible utilise des modèle de push. Le serveur Ansible (Noeud de contrôle) utilise SSH pour gérer des appareils et pousser leurs changements de configurations.

Puppet et Chef utilise un modèle de pull.

Après avoir installé Ansible on peut créer plusieurs fichiers textes :

- Playbooks : ces fichiers sont des « blueprint d'automatisation de tâche ». Ils ont en dehors de la logique et des actions de tâches que Ansible devrait faire. Ecrite en YAML.
- Inventaire : Ces fichiers listent les appareils qui peuvent être gérés par Ansible, comme les caractéristiques de chaque appareil comme leur rôle d'appareil (Accéder à un Switch, Switch Core, Routeur WAN, Firewall, etc..) écrit en INI, YAML et dans d'autres formats.
- Templates : Ces fichiers représentent le fichier de configuration d'appareils, mais pour des valeurs spécifique pour des variables non fournis. Ecrite en format Jinja2.
- Variables : Ces fichiers listent les variables et leurs valeurs. Ces valeurs sont substitués en des templates pour créer des fichiers de configuration complets. Ecrit en YAML.

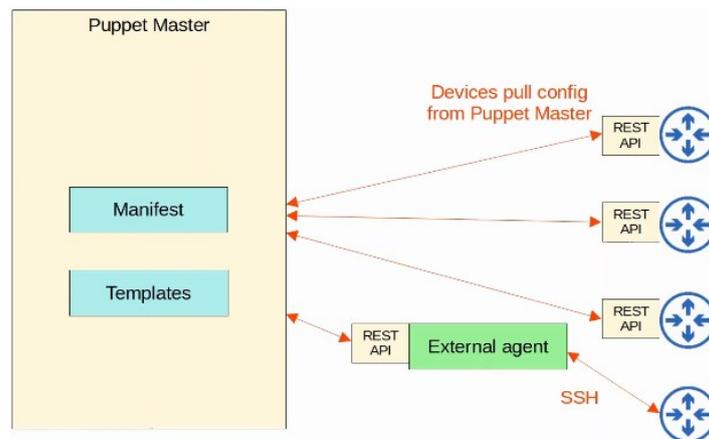
Voici un exemple d'utilisation d'Ansible sur un réseau dans lequel la configuration est poussé :



Puppet est un outil de gestion de configuration écrit en Ruby. Puppet est basé sur des agents. Des logiciels spécifiques doivent être installés sur des appareils gérés. Pas tous les appareils Cisco supportent les agent Puppet. Puppet peut être lancé sans agent, dans lequel un agent proxy lance un hôte externe, et l'agent proxy utilisé SSH pour se connecter à l'appareil géré et communiquer avec eux. Le serveur Puppet est appelé « Puppet Master ». Puppet utilise un modèle de « pull » (les clients « pull » la configuration depuis le Puppet Master). Les clients utilise TCP 8140 pour communiquer avec le Puppet Master. Au lieu de YAML, il utilise un langage propriétaire pour les fichiers. Les fichiers textes requis dans le Puppet Master incluent :

- Manifest : Ce fichier définit l'état de la configuration voulue d'un appareil réseau.
- Templates : De manière similaire à des Templates Ansible. Utilisé pour générer des Manifests.

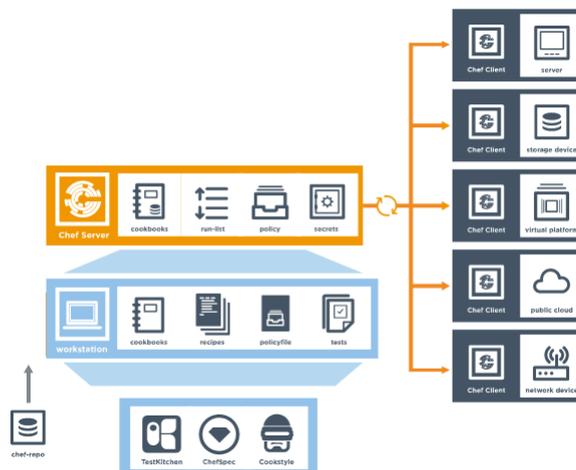
Voici un réseau utilisant Puppet :



Chef est un outil de gestion de configuration écrit en Ruby. Chef est basé sur des agent. Des logiciels doivent être installés dans les appareils gérés. Pas tous les appareils Cisco supportent des agent Chef. Chef utilise un modèle Pull. Le serveur utilise TCP 10002 pour envoyer des configurations aux clients. Les fichiers utilisent un DSL (Domain-Specific Language) basé sur Ruby. Les fichiers textes utilisés par Chef incluent :

- Ressources : les « ingrédients » dans une recette. L'objet de configuration est géré par Chef. Les recettes dans un livre de recette. En dehors de la logique et des actions des tâches fonctionnant sur les ressources.
- Cookbook : ou livre de recette en Français est plusieurs recettes regroupés ensemble.
- Run-list : Est une liste ordonné de recettes qui sont lancés pour apporter l'appareil vers l'état de configuration voulue.

Voici un réseau utilisant Chef :



Voici un tableau comparant ces différents outils :

	Ansible	Puppet	Chef
Fichiers Clefs définis par des actions	Playbook	Manifest	Recipe, Run-list
Protocole de communication	SSH	HTTPS (Par REST API)	HTTPS (par REST API)
Port clef	22 (Port SSH)	8140	10002
Basé sur agent/sans agent	Sans Agent	Basé sur agent (Ou sans agent)	Basé sur agent
Push/Pull	Push	Pull	Pull