

AVANTAGES ET FONCTIONNEMENT D'ANSIBLE

Pourquoi avons-nous besoin d'Ansible ?

Bien avant de vous expliquer ce qu'est Ansible, il est de la plus haute importance de comprendre les **problèmes rencontrés avant l'utilisation Ansible**. Prenons un petit retour en arrière sur le début de l'informatique lorsque le déploiement et la gestion des serveurs de manière fiable et efficace ont été un défi. Auparavant, les administrateurs système géraient les serveurs manuellement, installaient les logiciels, changeaient les configurations et administraient les services sur des serveurs individuels.

À mesure que les Datacenters se développaient et que les applications hébergées devenaient plus complexes, les administrateurs ont réalisé qu'ils ne pouvaient pas faire évoluer leur gestion manuelle des systèmes aussi rapidement que les applications qu'ils activaient. Cela a également entravé la vitesse du travail des développeurs car l'équipe de développement était agile et publiait fréquemment des logiciels, mais les opérations informatiques passaient plus de temps à configurer les systèmes. C'est pourquoi que des outils de provisionnement de serveur et de gestion de la configuration automatique ont prospéré. Une solution des solutions au problème mentionné ci-dessus est Ansible.

L'outil Ansible vise à fournir des gains de productivité importants à une grande variété de défis d'automatisation. Comme vu sur le chapitre précédent, c'est un outil qui se veut simple à l'utilisation mais suffisamment puissant pour automatiser des environnements d'applications complexes à plusieurs niveaux.

Avantages d'Ansible

Voici une liste des avantages d'Ansible :

- **Gratuit**: Ansible est un outil open source.
- **Simple** : Ansible utilise une syntaxe simple écrite en YAML. Aucune compétence en programmation particulière n'est nécessaire pour créer les playbooks d'Ansible. Il est également simple à installer.
- **Puissant**: Ansible vous permet de modéliser des workflows très complexes.
- **Flexible**: Ansible vous fournit des centaines de modules prêts à l'emploi pour gérer vos tâches, quel que soit l'endroit où ils sont déployés. Vous pouvez réutiliser le même playbook sur un parc de machines Red Hat, Ubuntu ou autres.
- **Agentless** : vous n'avez pas besoin d'installer d'autres logiciels ou d'ouvrir des ports de pare-feu supplémentaires sur les systèmes clients que vous souhaitez automatiser. Ansible réduit encore l'effort requis pour que votre équipe commence à automatiser immédiatement.
- **Efficace**: Parce que vous n'avez pas besoin d'installer de logiciel supplémentaire, il y a plus de place pour les ressources d'application sur votre serveur.

Que peut faire Ansible?

Ansible peut être utilisé de différentes manières. J'en ai mentionné quelques-unes ci-dessous:

Déploiement d'applications

Ansible vous permet de déployer rapidement et facilement des applications à plusieurs niveaux. Vous n'aurez pas besoin d'écrire du code personnalisé pour automatiser vos systèmes; vous listez les tâches à effectuer en écrivant un playbook, et Ansible trouvera comment amener vos systèmes à l'état dans lequel vous voulez qu'ils soient. En d'autres termes, vous n'aurez pas à configurer manuellement les applications sur chaque machine . Lorsque vous exécutez un playbook à partir de votre machine de contrôle, Ansible utilisera le protocole SSH pour communiquer avec les hôtes distants et exécuter toutes les tâches (Tasks).

Orchestration

Comme son nom l'indique, l'orchestration consiste à amener différents éléments à interagir ensemble sans incohérence. Par exemple, avec le déploiement d'applications, vous devez gérer non seulement les services frontend, mais également les services backend comme les bases de données, le réseau, le stockage, etc... Vous devez également vous assurer que toutes les tâches sont gérées dans le bon ordre. Grâce à Ansible vous orchestrez les éléments de votre infrastructure à l'aide des playbooks Ansible, et vous pouvez les réutiliser sur différents types de machines, grâce à la portabilité des modules Ansible.

Conformité et sécurité

Comme pour le déploiement d'applications, des politiques de sécurité de votre entreprise (telles que des règles de pare-feu ou le verrouillage des utilisateurs) peuvent être mises en œuvre avec d'autres processus automatisés. Si vous configurez les détails de sécurité sur la machine de contrôle et exécutez le playbook

associé, tous les hôtes distants seront automatiquement mis à jour avec ces détails. Cela signifie que vous n'aurez pas besoin de surveiller chaque machine pour vérifier la conformité de la sécurité en continu manuellement. De plus, tous les identifiants (identifiants et mots de passe des utilisateurs admin) qui sont stockés dans vos playbooks ne sont récupérables en brut par aucun utilisateur.

Provisionnement du cloud

La première étape de l'automatisation du cycle de vie de vos applications consiste à automatiser l'approvisionnement de votre infrastructure. Avec Ansible, vous pouvez provisionner des plateformes cloud, des hôtes virtualisés, des périphériques réseau et des serveurs physiques.

Comment fonctionne Ansible

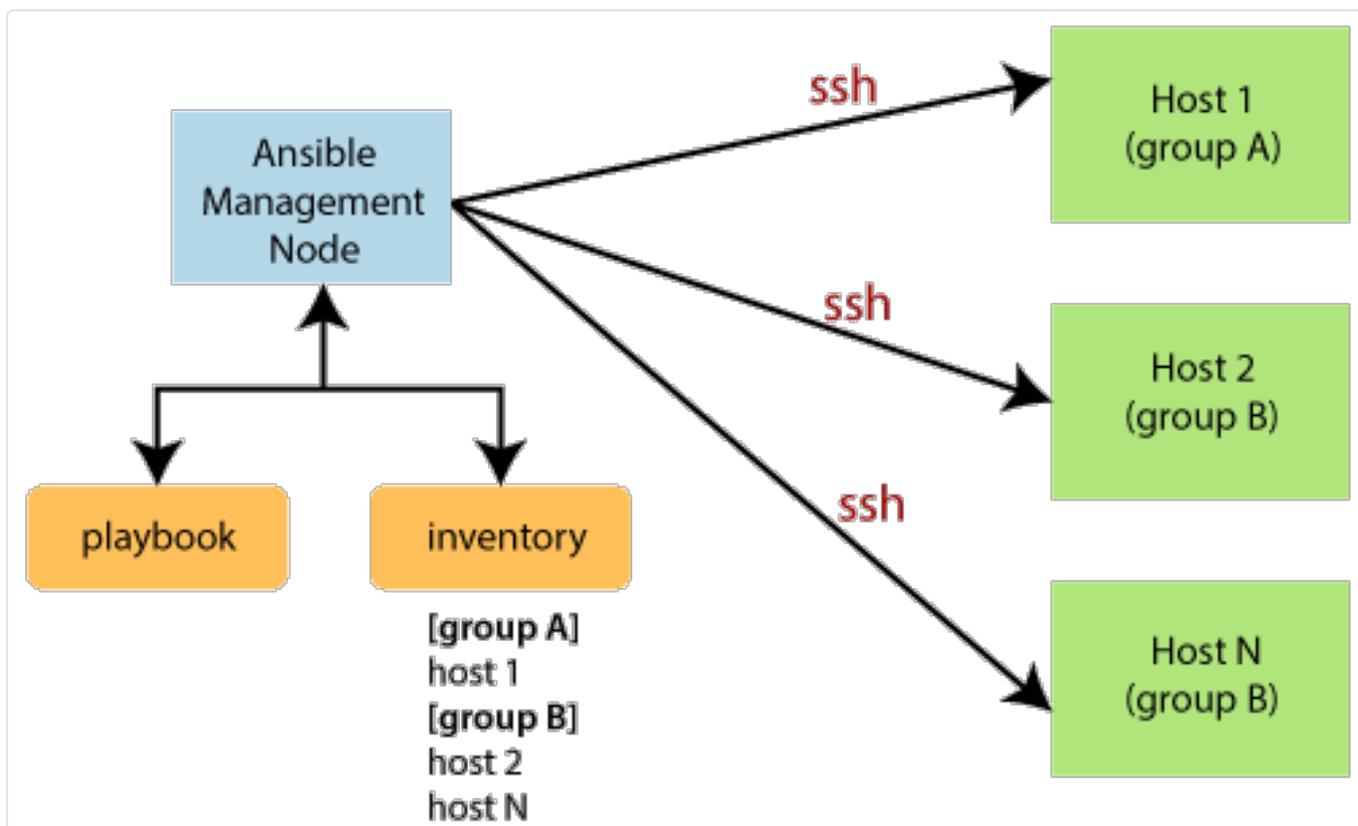
Dans Ansible, il existe deux catégories d'ordinateurs: le **nœud maître** (master) et les **nœuds esclaves** (slaves). Le nœud maître est une machine sur laquelle est installé l'outil Ansible. Il doit y avoir au moins un nœud maître, bien qu'un nœud maître de sauvegarde puisse également exister.

Ansible fonctionne en se connectant à vos nœuds en **SSH** et en y poussant de petits programmes, appelés **modules**. Ces modules sont définis dans un fichier nommé le **Playbook**. Le nœud maître, se base sur un fichier d'inventaire qui fournit la liste des hôtes sur lesquels les modules Ansible doivent être exécutés.

Ansible exécute ces modules en SSH et les supprime une fois terminé. La seule condition requise pour cette interaction est que votre nœud maître Ansible dispose d'un accès de connexion aux nœuds esclaves. Les clés SSH sont le moyen le plus courant de fournir un accès, mais d'autres formes d'authentification sont également

prises en charge.

Voici un schéma qui reprend notre explication :



Dans la prochaine partie, nous découvrirons comment configurer notre environnement d'Ansible.