



# Ansible

## Introduction

Glenn ROLLAND / [glenux@glenux.net](mailto:glenux@glenux.net)

# Présentation d'Ansible



**Logiciel libre, développé par Red-hat et Ansible**

**Permet la configuration et la gestion automatisée des ordinateurs**

- Déploiement de logiciels
- Exécution des tâches ad-hoc
- Gestion de configuration

**Site :** <https://ansible.com>

**Publié en février 2012**

**Aujourd'hui, de très nombreux utilisateurs**

- Hewlett-Packard
- Airbus
- La Poste
- Société Générale
- ...



**Host** : machine, caractérisée par son IP ou son nom de domaine

**Group** : ensemble de machine, caractérisée par un nom de groupe

**Command** (ou Module, ou Action) :

- éléments de base, exécutable en mode autonome
- permet d'interagir avec les systèmes
- des centaines sont fournis par défaut
- de nombreux autres sont rendus disponibles par la communauté



## Task :

- combinaison d'une action + des paramètres
- plus petit élément manipulé dans ansible
- peut contenir des directives pour conditions, boucles, etc.
- *ex: ajouter une ligne à un fichier, redémarrer un service...*

## Role :

- unité structurelle regroupant les tâches
- description d'un usage (serveur web, client NFS, montages, etc.)



## **Play :**

- structure YAML
- liste de rôles ou de tâches à appliquer
- spécifique pour un hôte ou un groupe

## **Playbook :**

- fichier yaml contenant plusieurs *Play*
- recettes globales que l'on joue pour mettre en place tout cela
- vise un ou plusieurs listes d'hotes ou de groupes



## Modularité

- Connexions via une simple liaison SSH
- Commandes peuvent être codé en tout langage (par défaut en python)
- Roles écrits en YAML
- Playbooks écrits en YAML



## Idempotence

- une opération a toujours le même effet
- applique une ou même plusieurs fois, la réappliquer on ne modifie pas le résultat

## Exemple

Un script de configuration qui ajoute une ligne de configuration :

- insérera la ligne si absente,
- modifiera si elle est présente mais erronée
- et ne fera rien si tout est déjà correct

# Architecture d'ansible



## Sans agent

Ansible contrôle les ordinateurs depuis un noeud maître

- il ne nécessite pas d'installation sur les ordinateurs contrôlés
- simple présence d'une connexion SSH entre le maître et la machine contrôlée
- seul python doit être présent sur la machine destination

## Exemple

Il est très facile d'intégrer de nouvelles machines dans une gestion par Ansible :

- récupérer la version de l'OS de centaines de serveurs, en une seule ligne de commande et sans artefact installé sur les serveurs.



## Orchestration

Au delà de l'automatisation de procédures et la mise en place de configurations, Ansible brille également dans l'orchestration d'enchaînements complexes. En jouant sur le niveau de parallélisation de son exécution on peut par exemple faire un déploiement roulant de mises à jour de tout un cluster applicatif, en intervenant sur chaque serveur en coordination avec le load-balancer pour éviter toute interruption. Il sait aussi faire ce que font Capistrano ou Fabric pour le déploiement applicatif. Avec de nombreux modules liés aux conteneurs et à la virtualisation il sait manipuler des infrastructures de type Cloud à géométrie variable.

Elle gère les différents nœuds à travers SSH et ne nécessite l'installation d'aucun logiciel supplémentaire sur ceux-ci.

Les modules communiquent via la sortie standard en notation JSON et peuvent être écrits dans n'importe quel langage de programmation.