

Lab8_ Gestion de l'infrastructure avec AWX

Gérer une infrastructure

On va parler maintenant de Ansible Tower. Ansible a été racheté il y a quelques années par Red Hat et Tower est en réalité un outil payant pour centraliser et contrôler votre infrastructure Ansible. Il existe un produit identique de la communauté qui s'appelle AWX qu'on verra par la suite.

Tower est un outil de centralisation pour contrôler toute votre infrastructure Ansible avec un tableau de bord visuel. Il permet de contrôler les accès grâce à ce qu'on appelle les RBAC, les Role Base Access Controllers. Vous pouvez planifier des jobs et vous pouvez aussi configurer des identifications.

Tower est aussi une API REST qui permet en fait de le rendre ouvert et disponible pour les autres outils de votre infrastructure, comme des outils d'analyse de logs, des choses comme ça. Ensuite, vous avez accès à un dashboard. Vous verrez l'état des hôtes, l'état des inventaires et l'état des jobs. Vous avez cette notion de real time job. Ça veut dire qu'en réalité vous avez, pour chaque playbook, la possibilité de le suivre au fur et à mesure de son exécution, de voir son état, de voir s'il est en réussite ou en échec, d'avoir la sortie standard. Vous avez les deux vues. Le status et ici la sortie standard de la commande Ansible. Et vous avez la possibilité de visualiser l'état de l'ensemble des plays qui sont automatisés par flux.

Vous avez aussi la possibilité de gérer les logs, de les paramétrer, de les personnaliser et de les récupérer sur l'environnement Tower et vous pouvez aussi les exporter vers d'autres outils d'analyse, votre infrastructure, grâce bien sûr à cette notion d'API REST, qui est disponible au niveau de Tower.

On retrouve les mêmes capacités avec AWX qui est un outil de la communauté, donc Tower reste dans la partie payante d'Ansible et de ce qu'on appelle la Red Hat Automation Platform. C'est l'outil de Red Hat pour pouvoir mettre en avant Ansible et Tower.

Installer AWX

À partir d'une recherche Internet ou sur le site Red Hat Ansible, vous pouvez trouver le lien pour aller récupérer le projet AWX sur GitHub.

On est sur le GitHub Ansible AWX et vous pouvez cloner le Git en local. Une fois que le Git est cloné, vous allez avoir un répertoire installer, avec un ensemble de rôles ici, et vous avez ici le playbook principal d'installation.

Ce playbook va exécuter les rôles en question. Par défaut, il s'exécute sur un inventaire qui est par défaut local, sauf si vous voulez l'exécuter sur d'autres machines cibles. On va installer AWX avec Ansible. Si on va sur la partie documentation de l'installation, ensuite, il faut installer des outils de compilation, GNU Make, bien sûr le Git et du Node.js. Voilà pour la partie prérequis.

Si on descend plus bas dans la doc, vous allez avoir ici trois formes d'exécution. Ça sera du Docker mais soit sur OpenShift soit sur du Kubernetes soit avec un simple Docker et un Docker Compose.

J'ai choisi la solution la plus simple avec Docker Compose. Une fois que vous avez cloné le Git, vous l'avez cloné quelque part au niveau de l'arborescence user ansible, je vais l'installer sur la machine control et si on fait un ls dans l'arborescence AWX installer, vous avez les roles, le script d'installation Ansible qu'on va pouvoir en fait utiliser.

```
ansible-playbook -i inventory playbook install.yml
```

Pour vérifier l'installation, on va faire un docker ps et vous verrez qu'il y a cinq containers qui permettent d'exécuter l'environnement AWX. Une fois les containers démarrés, les services sont disponibles, et on va pouvoir se connecter sur le frontal qui reste sur du port 80. On va pouvoir faire un HTTP sur la machine de contrôle. Le user par défaut est admin, le mot de passe par défaut est password. Voilà, on a installé l'environnement AWX pour pouvoir centraliser notre infrastructure Ansible.

Planifier les tâches Ansible

Nous allons voir quelle est la logique de cet outil. D'abord, il faut créer ce qu'on appelle des « projets ». Il y a un projet qui est déjà existant, Il faut lui donner un nom, une organisation, bien sûr une description. Ensuite, vous avez ici les types de connexion : est-ce qu'on est sur du Git, du Manuel, etc. ? Si on est sur du Git, donc, on donne ici l'URL du Git et ensuite on peut spécifier des balises de validation, des choses comme ça.

On peut aussi donner des délais d'attente au niveau de la gestion du cache, ainsi de suite. Une fois qu'on a créé le projet, il faut créer un inventaire.

L'inventaire, ici, est uniquement l'inventaire local. Il y a Détails, il y a des Permissions. On peut créer des Groupes sur un inventaire, des Hosts sur un inventaire et, rappelez-vous, on peut créer des variables de hôtes ou des variables de groupes. Ici, on a juste la notion de localhost qui est utilisée.

Une fois qu'on a fait ça, on va passer sur la partie Modèles et on va créer un modèle d'exécution, un modèle de job. On sélectionne ce modèle ; ici on a le nom du modèle ; le type du modèle. Ensuite, on donne ici l'inventaire, on le sélectionne parmi la liste des inventaires disponibles. On l'associe à un projet. Et, lorsqu'on l'a associé à un projet, on peut en fait dérouler les menus pour avoir la liste des fichiers qui sont disponibles sur ce projet. Ensuite, le mode « Verbosity » : Normal, ou 1, ou 2, ou 3, 3-v, etc. Et ensuite, on sauvegarde le modèle.

Lorsqu'on va sur le modèle, pour pouvoir exécuter un modèle, on a en fait ici la notion de Démarrer le job. On peut démarrer donc le modèle. Et on aura donc les deux fenêtres : la fenêtre de Détails et la fenêtre d'exécution du job.

C'est juste un aperçu de l'utilisation de cet outil, AWX, qui est pratiquement identique au Tower d'Ansible Red Hat et qui permet de centraliser l'ensemble de votre infrastructure Ansible.

- Se connecter en tant que super-utilisateur (admin)
- Examiner le “Tower Dashboard”
- La page des paramètres
- Réviser les données de l’Organization”
- Ajouter un User dans l’Organization”
- Créer un nouvel “Inventory” et l’ajouter à l’Organization”
 - Groups : `webserver`
 - Hosts : `app2`
 - Vars: `ansible_connection: ssh`
- Créer un “Credential”
 - type “Machine”
- Mettre en place un “Project”
 - nom : “demo-lab”
 - URL du SCM : <https://github.com/ansible/tower-example>
- Créer un nouveau “Job Template”
- Lancement