

AWX

Présentation

AWX est l'interface utilisateur Web open-source, l'API REST et le moteur de tâches pour l'automatisation Ansible. Il vous donne un tableau de bord centralisé pour gérer les playbooks, les inventaires, les informations d'identification, les modèles de tâches et la planification.

Prise en main

On va commencer par rassembler dans un dépôt git awx-lamp notre playbook lamp.yml, notre rôle lamp, notre fichier inventaire, et un fichier requirements.yml dans un nouveau répertoire collections:

ansible-lamp/

```
|— collections
|   |— requirements.yml
|— hosts
|— lamp
|   |— defaults
|   |   |— main.yml
|   |— handlers
|   |   |— main.yml
|   |— meta
|   |   |— main.yml
|   |— README.md
|   |— tasks
|   |   |— clean_dbuser.yml
|   |   |— Debian-phpmyadmin.yml
|   |   |— main.yml
|   |   |— RedHat-phpmyadmin.yml
|   |   |— test_dbuser.yml
|   |— templates
|   |   |— index.j2
```

```

|   |— tests
|   |   |— inventory
|   |   |   |— test.yml
|   |   |— vars
|   |   |   |— Debian.yml
|   |   |   |— main.yml
|   |   |   |— RedHat.yml
|   |— lamp.yml

```

Le fichier requirements.yml contiendra les collections utilisées :

collections:

- name: community.general
- name: community.mysql

Ensuite :

- Identifiez vous sur awx avec le login `admin` et le mot de passe configuré à l'installation.
- Vous pouvez changer le mot de passe...
- Dans la section Ressources → Informations d'identification, choisissez « Contrôle de la source ». Nom d'utilisateur « git » et dans la section clé privée copiez le contenu de notre clef privée. Inutile pour un dépôt git anonyme...
- Dans la section Ressources → Projet, importez le projet ansible-lamp. Comme Type de Contrôle de la source, choisissez « Git », comme URL Contrôle de la source : `git@ip_debian:ansible-lamp.git/`, la branche : master et l'Identifiant Contrôle de la source, celui que vous venez de créer. Synchronisez. Vous pouvez choisir aussi « Mettre à jour Révision au lancement » pour s'assurer que le dépôt git est synchronisé au lancement.
- Dans la section Ressources → Informations d'identification, choisissez « Machine ». Dans la section clé privée copiez le contenu de notre clef privée.
- Créez une ressource inventaire. Créez simplement l'inventaire avec un nom au départ. Une fois créé vous pouvez aller dans la section `source` et choisir de l'importer depuis notre projet, sélectionnez le fichier `hosts`. Synchronisez.
- Pour tester tout cela vous pouvez lancer une tâche ad-hoc `ping` depuis la section inventaire → `hosts` en sélectionnant une machine et en cliquant sur le bouton « exécuter commande ».

- Allez dans la section modèle de job et créez un job en sélectionnant le projet et l'inventaire que vous venez de créer, le playbook `lamp.yml` et les Informations d'identification Machine...
- Exécutez ensuite le job en cliquant sur la fusée. Vous vous retrouvez sur la page de job de AWX. La sortie ressemble à celle de la commande mais vous pouvez en plus explorer les tâches exécutées en cliquant dessus.
- Il est possible également d'utiliser Ansible vault. Dans la section Ressources → Informations d'identification, choisissez « Archivage sécurisé ». Puis, dans le modèle de job, ajoutez l'information d'identification correspondante.
- Les modèles de flux de travail AWX vous permettent de configurer une séquence de modèles de tâches qui peuvent s'exécuter dans un ordre spécifique, permettant des tâches d'automatisation complexes. Ils utilisent une structure de type graphique avec des nœuds représentant différents emplois ou actions, permettant une meilleure gestion et un meilleur suivi des processus d'automatisation.

T.P. Créer un modèle de flux de travail (ci-cd gitOps) qui synchronise le dépôt git, le fichier d'inventaire avant de lancer le rôle `awx-lamp`...

Créez un nouveau container que vous ajouterez à votre inventaire

Lancez votre flux de travail

Allez voir la sortie du job

Modifiez votre modèle : dans la section **Programmation** configurer l'exécution du job toutes les 5 minutes.

Allez dans la section planification. Puis visitez l'historique des Jobs.