

tower-tuto

sahibmartial

April 2020

1 Introduction

Ce tuto décrit toutes les étapes à suivre pour élaborer ce tp de Ansible Tower. Tower AWX a certes faculté de comprendre et interpréter les commandes ansible. Tower AWX est un outil d'orchestration des playbooks ansible utilisé dans un contexte de production. C'est dans un tel contexte que ce tp est réalisé afin de déployer l'application fake-backend pour notre client POZOS.

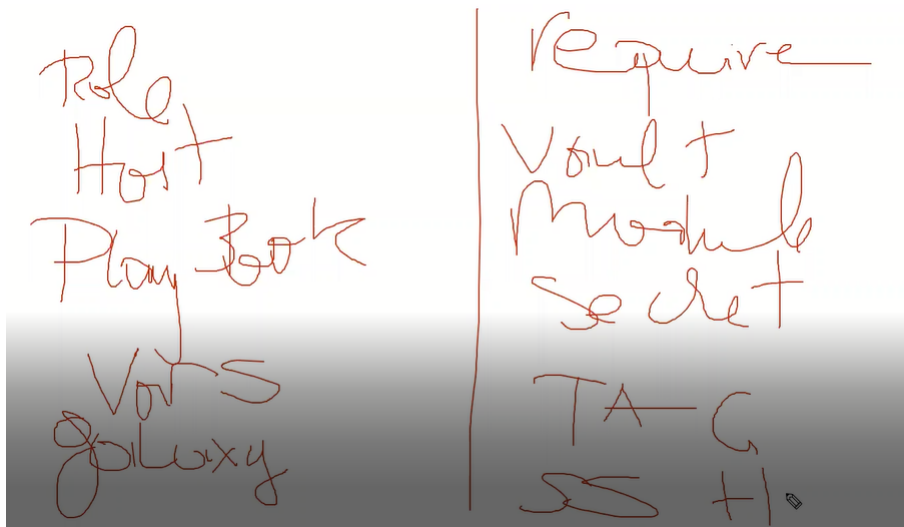


Figure 1: Ansible terme

2 AWS: lancement de notre stack

Depuis la plate-forme AWS nous allons créer nos instances à savoir:

- Tower AWX

- Build
- Preprod
- Prod

patientez le temps que cela s'installe .

3 Repo github

Notre repo github cible qui va servir pour l'occasion à notre tp pour le déploiement est **Fake-Backend-Tower**. le contenu du repo en image .

■ battleboat	commit application	10 hours ago
■ files/secrets	commit application	10 hours ago
■ group_vars	commit application	10 hours ago
■ roles	commit application	10 hours ago
📄 Dockerfile	commit application	10 hours ago
📄 Gopkg.lock	commit application	10 hours ago
📄 Gopkg.toml	commit application	10 hours ago
📄 Jenkinsfile	commit application	10 hours ago
📄 README.md	first commit	10 hours ago
📄 check_deploy_app.yml	commit application	10 hours ago
📄 config.go	commit application	10 hours ago
📄 docker-compose.yml	commit application	10 hours ago
📄 hosts	commit application	10 hours ago
📄 install_fake-backend.yml	commit application	10 hours ago
📄 main.go	commit application	10 hours ago
📄 ping.yml	commit application	10 hours ago

Figure 2: repo fake-backend

4 Configuration TOWER AWX

Tower AwX est un orchestrateur ansible pour gérer et interpréter tous nos play-books que nous avons écrit lors de la phase de développement de notre chaine CI/CD. Spécialisé pour le déploiement de nos applications en production.

Identifiant pour se connecter sont:

User:**admin**

Mot de passe :**password**

Une fois logger nous allons définir nos credentials:

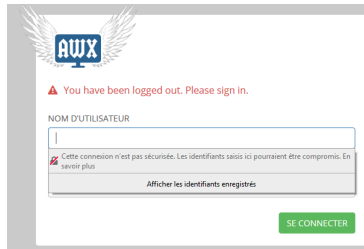


Figure 3: Tower AWX

1. vault-key : elle sert à décrypter nos secrets, donc en occurrence le fichier devops.yml
2. devops key-private : elle sert à établir des connexions ssh sur les hosts

4.1 Credentials

En fonction de la langue de ton navigateur Tower configurera la version de la langue compatible . Ainsi **vault**(anglais) sera **coffre fort**(french).

4.1.1 Credential vault-key

En image la configuration de ce credential.

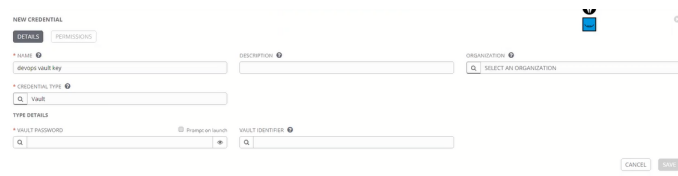


Figure 4: Credential vault key

4.1.2 Credential devops key-private

En image la configuration de ce credential.

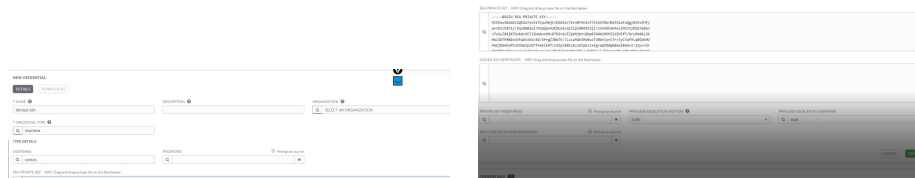


Figure 5: credentials ssh

5 Création du projet

Il va contenir tous nos jobs , on va donc configurer ce dernier afin de le lier a notre repo cible sur github *Fake-backend-Tower*.

En image la création du projet, cliquer sur projet dans l'interface Tower .

* NOM
Fake-backend-tower

DESCRIPTION

* ORGANISATION
Default

* TYPE SCM
Git

DÉTAILS DE LA SOURCE

* URL DU SCM
hub.com/sahibmartial/Fake-Backend-Tower.git

BRANCHE SCM/BALISE/VALIDATION
master

SCM REFSPEC

SCM INFORMATION D'IDENTIFICATION

OPTIONS DE MISE À JOUR SCM

- ☐ NETTOYER
- ☐ SUPPRIMER LORS DE LA MISE À JOUR
- ☐ METTRE À JOUR RÉVISION AU LANCEMENT
- ☐ ALLOW BRANCH OVERRIDE

ANNULER ENREGISTRER

Figure 6: Création du projet

5.1 Inventaire

En image la création de l'inventaire. Ensuite création de la source pour dire que

INVENTAIRES / fake-ops

fake-ops

DÉTAILS PERMISSIONS GROUPES HÔTES SOURCES JOBS COMPLÉTÉS

* NOM
fake-ops

DESCRIPTION

* ORGANISATION
Default

INFORMATION D'IDENTIFICATION À INSIGHTS

GROUPES D'INSTANCES

Figure 7: Inventaire

notre inventaire provient d'une source externe.

5.1.1 Source

fake-ops

DÉTAILS NOTIFICATIONS PLANNINGS

* NOM fake-ops DESCRIPTION * SOURCE Provenance d'un projet

DÉTAILS DE LA SOURCE

INFORMATION D'IDENTIFICATION

* PROJET fake-ops * FICHIER D'INVENTAIRE hosts

VERBOSITÉ 1 (INFO)

METTRE À JOUR LES OPTIONS

- ☐ REMPLACER
- ☐ REMPLACER LES VARIABLES
- ☐ METTRE À JOUR AU LANCEMENT
- ☐ METTRE À JOUR LORSQUE LE PROJET EST ACTUALISÉ

Figure 8: Source

Un refresh sur notre inventaire *fake-ops* crée, puis en cliquant sur groupes, hosts nous pouvons voir la magie de tower.

fake-ops

DÉTAILS PERMISSIONS GROUPES HÔTES SOURCES JOBS COMPLÉTÉS

RECHERCHER CLÉ ALL GROUPS ROOT GROUPS EXÉCUTER DES COMMANDES

GROUPES	ACTIONS
<input type="checkbox"/> build	
<input type="checkbox"/> preprod	
<input type="checkbox"/> prod	

ÉLÉMENTS 1 - 3

fake-ops

DÉTAILS PERMISSIONS GROUPES HÔTES SOURCES JOBS COMPLÉTÉS

RECHERCHER CLÉ EXÉCUTER DES COMMANDES

HÔTES	DESCRIPTION	GROUPES LIÉS	ACTIONS
<input type="checkbox"/> 100.27.31.130	imported	prod	
<input type="checkbox"/> 3.80.148.1	imported	build	
<input type="checkbox"/> 54.85.180.1	imported	preprod	

ÉLÉMENTS 1 - 3

Figure 9: Source

5.2 Jobs

Création des différentes tâches de notre projet.

5.2.1 Ping

The screenshot shows the 'ping-tower' configuration form. At the top, there are tabs: 'DÉTAILS' (selected), 'PERMISSIONS', 'NOTIFICATIONS', 'JOBS COMPLÉTÉS', 'PLANNINGS', and 'AJOUTER UN QUESTIONNAIRE'. The form is divided into several sections:

- * NOM:** 'ping-tower'
- * INVENTAIRE:** 'fake-ops' (with a checkbox 'ME LE DEMANDER AU LANCEMENT')
- INFORMATIONS D'IDENTIFICATION:** Includes a search bar with 'devops-salt' and 'devops-vault-key' suggestions, and a checkbox 'ME LE DEMANDER AU LANCEMENT'.
- * VERBOSITÉ:** A dropdown menu.
- LIBELLÉS:** A text input field.
- TIMEOUT:** A numeric input field set to '0'.
- DESCRIPTION:** A text input field.
- * PROJET:** 'Fake-backend-tower' (with a checkbox 'ME LE DEMANDER AU LANCEMENT')
- FORMES:** A numeric input field set to '0'.
- BALISES DU JOB:** A text input field (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- BALISES DE SAUT:** A text input field (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- * TYPE DE JOB:** A dropdown menu set to 'Exécuter'.
- * PLAYBOOK:** 'ping.yml' (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- LIMITE:** A text input field (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- GROUPE D'INSTANCES:** A search bar with '1'.
- JOBS SLICING:** A numeric input field set to '1'.
- AFFICHER LES MODIFICATIONS:** A checkbox 'ME LE DEMANDER AU LANCEMENT'.
- OPTIONS:** A checkbox 'ACTIVER L'ÉLEVATION DES PRIVILÈGES' (checked).

Figure 10: Ping All hosts

5.2.2 Build

The screenshot shows the 'build-tower' configuration form. At the top, there are tabs: 'DÉTAILS' (selected), 'PERMISSIONS', 'NOTIFICATIONS', 'JOBS COMPLÉTÉS', 'PLANNINGS', and 'AJOUTER UN QUESTIONNAIRE'. The form is divided into several sections:

- * NOM:** 'build-tower'
- * INVENTAIRE:** 'fake-ops' (with a checkbox 'ME LE DEMANDER AU LANCEMENT')
- INFORMATIONS D'IDENTIFICATION:** Includes a search bar with 'devops-salt' and 'devops-vault-key' suggestions, and a checkbox 'ME LE DEMANDER AU LANCEMENT'.
- * VERBOSITÉ:** A dropdown menu set to '0 (Normal)'.
- LIBELLÉS:** A text input field.
- TIMEOUT:** A numeric input field.
- DESCRIPTION:** A text input field.
- * PROJET:** 'Fake-backend-tower' (with a checkbox 'ME LE DEMANDER AU LANCEMENT')
- FORMES:** A numeric input field set to '0'.
- BALISES DU JOB:** A dropdown menu set to 'build' (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- BALISES DE SAUT:** A text input field (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- * TYPE DE JOB:** A dropdown menu.
- * PLAYBOOK:** 'install_fake-backend.yml' (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- LIMITE:** A text input field (with a checkbox 'ME LE DEMANDER AU LANCEMENT').
- GROUPE D'INSTANCES:** A search bar.
- JOBS SLICING:** A numeric input field set to '1'.
- AFFICHER LES MODIFICATIONS:** A checkbox 'ME LE DEMANDER AU LANCEMENT'.
- OPTIONS:** A checkbox 'ACTIVER L'ÉLEVATION DES PRIVILÈGES' (checked).

Figure 11: Build Images

5.2.3 Deploy

The screenshot shows the 'deploy-tower' configuration page. The 'NOM' field is 'deploy-tower'. The 'INVENTAIRE' field is 'fake-ops'. The 'PROJET' field is 'fake-backend-tower'. The 'PLAYBOOK' field is 'install_fake-backend.yml'. The 'LIMITES' field is 'prod'. The 'VERBOSITÉ' field is '0 (Normal)'. The 'LIBELLÉS' field is empty. The 'TIMEOUT' field is '0'. The 'DESCRIPTION' field is empty. The 'FORKS' field is '0'. The 'BALISES DU JOB' field is 'prod'. The 'BALISES DE SAUT' field is empty. The 'GROUPES D'INSTANCES' field is empty. The 'JOB SLICING' field is '1'. The 'AFFICHER LES MODIFICATIONS' checkbox is checked. The 'OPTIONS' section is empty.

Figure 12: Deploy Application fake-backend

5.2.4 Check-Deploy

The screenshot shows the 'check-deploy-tower' configuration page. The 'NOM' field is 'check-deploy-tower'. The 'INVENTAIRE' field is 'fake-ops'. The 'PROJET' field is 'fake-backend-tower'. The 'PLAYBOOK' field is 'check_deploy_app.yml'. The 'LIMITES' field is 'prod'. The 'VERBOSITÉ' field is '0 (Normal)'. The 'LIBELLÉS' field is empty. The 'TIMEOUT' field is '0'. The 'DESCRIPTION' field is empty. The 'FORKS' field is '0'. The 'BALISES DU JOB' field is 'prod'. The 'BALISES DE SAUT' field is empty. The 'GROUPES D'INSTANCES' field is empty. The 'JOB SLICING' field is '1'. The 'AFFICHER LES MODIFICATIONS' checkbox is checked. The 'OPTIONS' section has 'ACTIVER L'ÉLEVATION DES PRIVILÈGES' checked.

Figure 13: Check-Deploy application

6 Workflow

Synchronisation des jobs et du webhook. on crée notre workflow comme ceci, puis enregistrer Ensuite sur l'icône *visualisateur du workflow* pour dessiner les étapes de notre workflow voir image

Figure 14: création du workflow

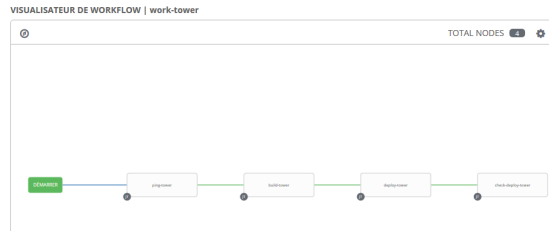


Figure 15: visualisateur du workflow

7 Webhook

Après avoir créé notre workflow, ce dernier nous génère le token et l'URL pour notre webhook figure 14. Nous allons copier ces valeurs en éditant un webhook sur notre repo cible comme ceci.

Figure 16: create webhook

8 Notification

Nous allons réaliser une notification depuis tower via slack. Ici j'utilise mon espace personnel slack.

8.1 Notificatiion slack

Ici l'url pour créer notre api slack: <https://api.slack.com/apps>.

J'ai donc créer mon api :**devops-bot** ensuite dans **Settings** puis dans **Basic Information**.

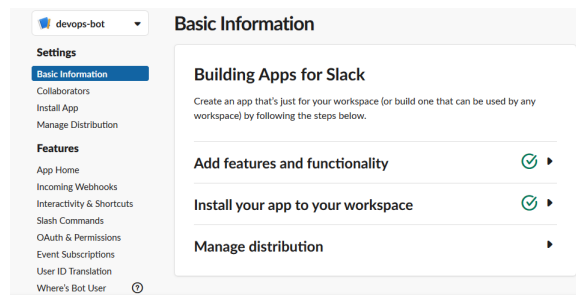


Figure 17: Basic information

On peut voir sur cette image la liste des éléments à configurer pour faire fonctionner notre notification en voici quelques étapes importantes. En premier effectuer les **Add features and functionality** voir figure ce-dessus.

8.1.1 APP Home

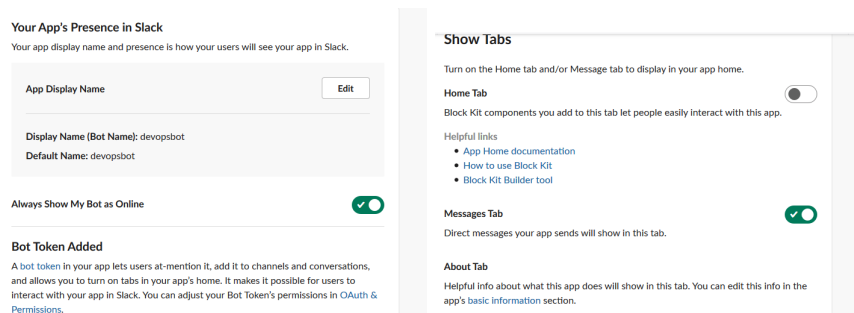


Figure 18: Edit name of Api

8.1.2 Incoming Webhooks

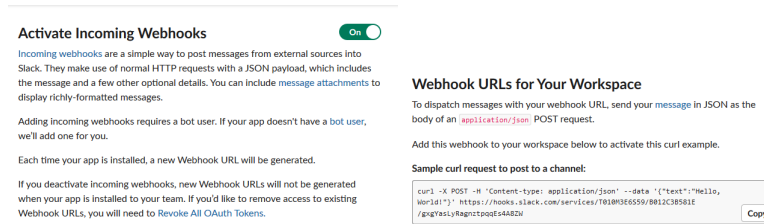


Figure 19: Incoming Webhooks

8.1.3 OAuth et Permissions

Génération du token d'accès pour notre Api.



Figure 20: Token bot

NB: C'est ce token que nous utiliser lorsque nous allons configurer la notification sur tower.

Ajouter les scopes comme ceci: Après avoir réaliser nos *Add features and func-*

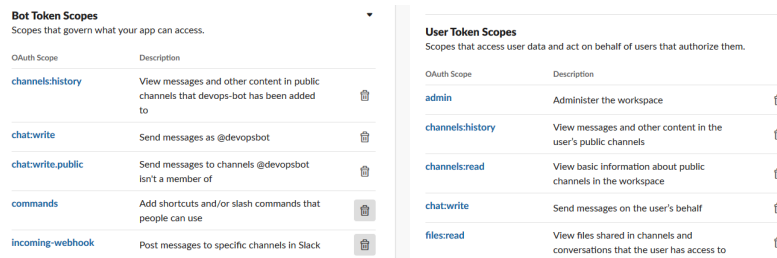


Figure 21: Scopes

tionality nous allons **:Install your app to your workspace** voir figure 15.

Une fois ceci terminé vous devez avoir un message dans le channel choisir.



Figure 22: Test notification

8.2 Notification Tower

8.2.1 Création de la notification

notification-slack

* NOM: notification-slack

DESCRIPTION: tower notification

* ORGANISATION: Default

* TYPE: Slack

DÉTAILS SUR LE TYPE

* CANAL DE DESTINATION: #tower

* JETON: *****

COULEUR DES NOTIFICATIONS: vert

notification-slack: Notification sent.

MODÈLES DE NOTIFICATION

NOM	TYPE	ACTIONS
notification-slack	Slack	

Figure 23: Création d'une notification

NB: le token à utiliser c'est celui de la figure 18 pour mon cas. Ensuite en cliquant sur la clochette on peut voir si notre notification fonctionne bien. Un message sera envoyé dans le canal que nous avons choisir.

8.2.2 Notification workflow

Se rendre sur notre workflow puis sur notification ensuite choisir la notification que nous venons de créer et activer les options comme ceci. Ensuite lets go pour le test.

work-tower

DÉTAILS PERMISSIONS NOTIFICATIONS JOBS COMPLÉTÉS PLANNINGS

RECHERCHER

CLÉ

NOM	TYPE	APPROVAL	START	RÉUSSITE	DÉFAILLANCE
notification-slack	Slack	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
tower-mail	Email	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Workflow Job #158 'fake-test-pipeline' running: <https://towerhost/#/workflows/158>

Workflow Job #158 'fake-test-pipeline' successful: <https://towerhost/#/workflows/158>

Figure 24: notification et résultat on channel

9 Conclusion

AWX nous facilite l'exécution des playbooks ansible dans un environnement de production .

“Imitative mène à la victoire”