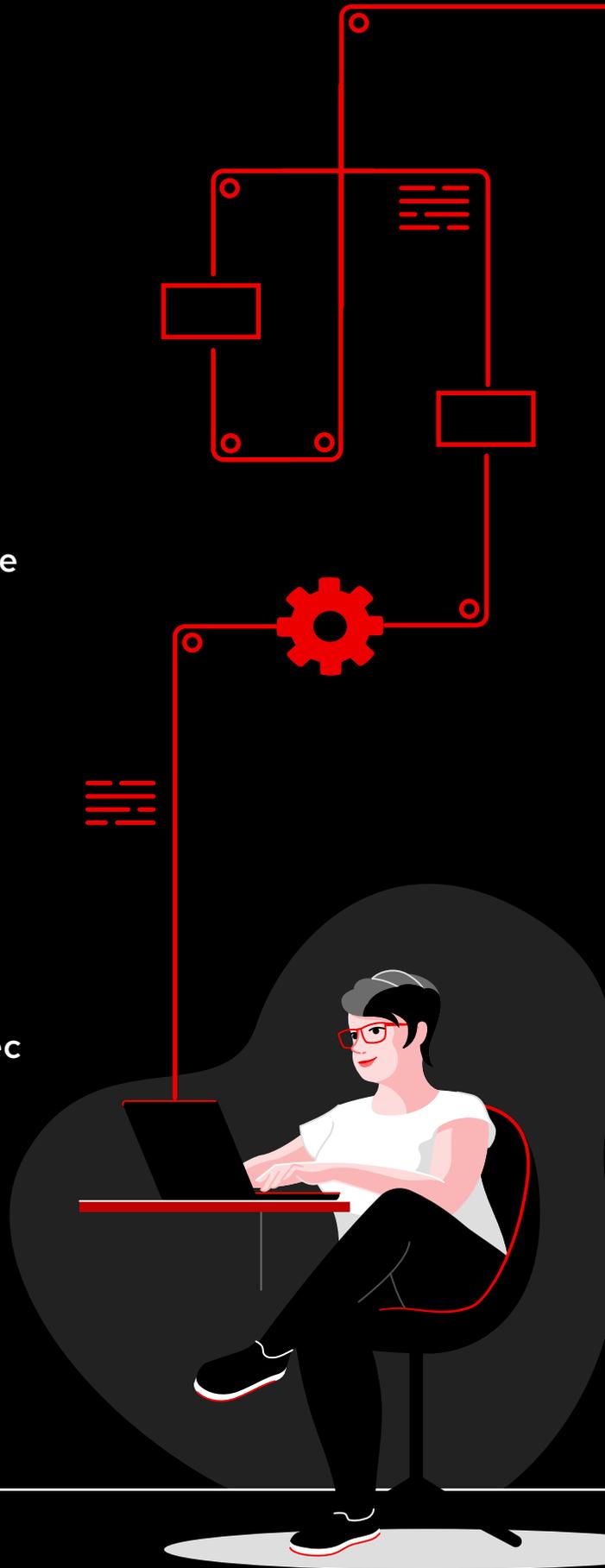


Automatiser votre réseau avec Red Hat

Un manuel technique pour mettre en œuvre les principales
tâches d'automatisation des réseaux avec la solution
Red Hat Ansible Automation Platform

Sommaire

- 1 Accélérer l'exploitation avec l'automatisation des réseaux
- 2 Installer et configurer Red Hat Ansible Automation Platform
- 3 Exécuter une première commande et un premier playbook
- 4 Créer votre inventaire
- 5 Mettre en œuvre des cas d'utilisation fréquents
- 6 Accéder au contenu utilisable avec Ansible Automation Platform
- 7 Aller encore plus loin dans l'automatisation
- 8 Se lancer



Accélérer l'exploitation avec l'automatisation des réseaux

Les approches traditionnelles et manuelles adoptées jusqu'ici pour configurer et mettre à jour les réseaux sont complexes et propices aux erreurs. Elles ne permettent donc pas de satisfaire efficacement les exigences en matière de transferts des données et d'applications qui évoluent en permanence. Avec des technologies d'automatisation logicielles et programmables, votre équipe sera en mesure de mieux soutenir et adapter les initiatives numériques de l'entreprise.

Grâce à l'automatisation des réseaux, les équipes d'exploitation des réseaux (NetOps) peuvent rapidement répondre à l'évolution des besoins en matière de capacités, de sécurité des applications, d'équilibrage de charge et d'intégrations multicloud. Elles peuvent aussi mettre en œuvre des activités réseau en libre-service et à la demande.

Par conséquent, les équipes NetOps ont la capacité de devenir aussi agiles et flexibles que les équipes responsables des infrastructures et des applications afin de prendre en charge les nouvelles demandes métier.

Ressources

Consultez les ressources suivantes pour en savoir plus sur les bases de la solution Ansible Automation Platform :

- ▶ Formation en ligne :
[Les bases d'Ansible : présentation technique de l'automatisation](#)
- ▶ Livre numérique :
[L'automatisation des réseaux à la portée de tous](#)

Accélérer l'exploitation avec Red Hat Ansible Automation Platform

Avec la [solution Red Hat® Ansible® Automation Platform](#), nous avons adapté le célèbre projet communautaire Ansible aux entreprises, en y ajoutant les fonctions et fonctionnalités nécessaires aux équipes pour automatiser à grande échelle. Cette plateforme d'automatisation informatique puissante associe un langage d'automatisation simple et facile à lire à une architecture flexible ainsi qu'à des capacités de partage et de collaboration axées sur la sécurité. Comme elle n'exige aucune compétence en programmation, toutes vos équipes, quel que soit leur rôle, peuvent utiliser immédiatement Ansible Automation Platform.

La solution Ansible Automation Platform permet de rationaliser et gérer des environnements de datacenters complexes, des serveurs et réseaux aux applications et pratiques DevOps. Parce qu'elle assure la prise en charge des périphériques d'infrastructure réseau multifournisseurs dans les environnements cloud, sur le réseau local, dans les succursales ainsi qu'en périphérie, elle facilite l'automatisation de tous les systèmes au moyen d'un seul outil.

Ce livre numérique explique comment effectuer les tâches courantes d'automatisation réseau.



Remarque : les lignes de commandes présentées dans ce livre numérique sont écrites pour Ansible Automation Platform 2.x et ne sont pas utilisables avec Ansible Automation Platform 1.0, ni les versions antérieures.



Installer et configurer Red Hat Ansible Automation Platform

Installer Ansible Automation Platform

La solution Ansible Automation Platform est facile et rapide à installer et configurer.

Étape

1a

Installer l'interface en ligne de commande d'Ansible avec yum

Exécutez la commande suivante :

```
$ sudo yum install ansible
```

Lisez le [Guide d'installation Ansible](#) pour des instructions complètes.

Étape

1b

Installer Ansible Automation Platform à l'aide de l'outil d'installation

1. Assurez-vous d'avoir la [dernière version](#) ou téléchargez une version d'essai gratuite sur red.ht/try_ansible.

2. Désarchivez le fichier .tar (la version et le nom peuvent être différents) :

```
$ tar xvzf ansible-automation-platform-setup-bundle-1.2.1-1.tar.gz  
$ cd ansible-automation-platform-setup-bundle-<version>
```

3. Ouvrez le fichier d'inventaire avec un éditeur de texte :

```
$vi inventory
```

4. Configurez votre adresse IP ou nom de domaine pleinement qualifié (FQDN) pour les hôtes [automationhub] et [automationcontroller], puis définissez vos mots de passe :

- ▶ admin_password pour l'administration
- ▶ pg_password pour la base de données

5. Exécutez le script de configuration.

```
$sudo ./setup.sh
```

6. Une fois que l'installation est terminée, accédez à votre hôte Ansible Automation Platform à l'aide de Google Chrome ou Mozilla Firefox en utilisant le nom d'hôte ou l'adresse IP. Connectez-vous avec les informations d'identification de l'administrateur indiquées dans le fichier d'inventaire.

7. Ajoutez votre souscription via le menu des paramètres, dans l'interface utilisateur web.

Lisez le [Guide d'installation d'Ansible Automation Platform](#) et le [Guide de configuration rapide d'Automation Controller](#) pour des instructions complètes.

Étape

2

Installer Ansible Content Collections pour la mise en réseau

Red Hat propose le composant [Ansible Content Collections](#), compatible et certifié pour un grand nombre de périphériques, d'outils et d'infrastructures réseau via [Ansible Automation Hub](#). Chaque collection existe au sein d'un espace de noms qui contient une ou plusieurs collections. Utilisez la commande `ansible-galaxy` pour installer ces collections :

```
$ ansible-galaxy collection install namespace.collection_name
```

Suivez les instructions de la [documentation Ansible Automation Platform](#) pour configurer Ansible Automation Hub afin d'accéder à vos collections et de les gérer. Vous trouverez des collections communautaires sur [Ansible Galaxy](#).

Étape

3

Créer vos environnements d'exécution

Les environnements d'exécution sont des conteneurs qui incluent toutes les dépendances requises pour automatiser le réseau. Ils remplacent les environnements virtuels Python et facilitent l'automatisation des systèmes. Vous pouvez créer des environnements d'exécution avec `ansible-builder`.

Installez `ansible-builder` :

```
$ pip install ansible-builder
```

Lisez la [documentation Ansible Builder](#) pour en savoir plus sur les exigences, l'installation et l'utilisation de ce composant.

Les équipes de développement en automatisation peuvent utiliser `ansible-navigator` pour tester et exécuter les playbooks en local dans un environnement d'exécution. Vous pouvez également importer le même conteneur dans Automation Controller afin d'exécuter des modèles de tâche d'automatisation du réseau.

Suivez les instructions de la [documentation Ansible Automation Platform](#) pour créer et utiliser des environnements d'exécution.

Configurer votre environnement réseau

Configurez votre environnement réseau pour Ansible Automation Platform selon les meilleures pratiques suivantes :



Assurer la connectivité à votre environnement réseau

Configurez un compte de service Ansible Automation Platform pour vous connecter à vos systèmes d'authentification, d'autorisation et de comptabilité. La solution Ansible Automation Platform prend en charge les méthodes d'authentification d'entreprise comme le système TACACS+ (Terminal Access Controller Access-Control System Plus), le service RADIUS (Remote Access Dial-In User Service) et le protocole LDAP (Lightweight Directory Access Protocol). Pour en savoir plus, lisez la [section Configurer l'authentification d'entreprise](#) de la documentation.



Créer votre référentiel de playbooks

Connectez Ansible Automation Platform à votre outil de gestion de code source (SCM) en [paramétrant un projet dans l'interface web](#) afin d'accéder à tous les playbooks dans ce référentiel de projet.



Configurer votre inventaire

Créez un [inventaire](#) des périphériques réseau que vous souhaitez automatiser. Ansible Automation Platform peut gérer plusieurs inventaires. Vous pouvez charger des inventaires de manière dynamique depuis des outils courants tels qu'Amazon Web Services (AWS) EC2, Microsoft Azure Resource Center et VMware vCenter à l'aide des [plug-ins d'inventaire](#). Vous pouvez également [charger des inventaires](#) à partir d'un projet Ansible Automation Platform. Le chapitre [Créer votre inventaire](#) de ce livre numérique explique plus en détail la création et l'utilisation des inventaires.



Paramétrer les règles de pare-feu de votre réseau

Configurez les règles de votre pare-feu pour permettre à la solution Ansible Automation Platform de se connecter aux routeurs et commutateurs via le port Secure Shell (SSH) 22 par défaut. Si vous le souhaitez, vous pouvez modifier le numéro de ce port à l'aide de la [variable d'hôte](#) `ansible_port`.



Paramétrer vos mots de passe pour Ansible Automation Platform

Créez un [identifiant](#) pour retenir vos mots de passe. Vous pouvez autoriser vos utilisateurs et vos équipes à utiliser les informations d'identification sans les leur divulguer.

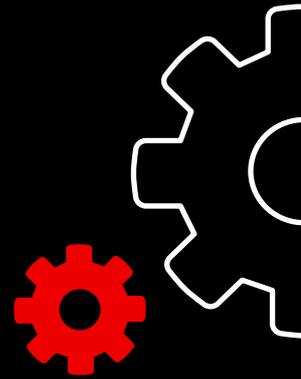


Créer un modèle de tâche Ansible

Créez un [modèle de tâche](#) pour connecter votre inventaire, votre identifiant et votre projet. Les modèles de tâche définissent des ensembles de paramètres pour exécuter les tâches d'automatisation, ce qui permet à plusieurs de vos équipes d'exécuter les mêmes ensembles de tâches plusieurs fois et d'en réutiliser le contenu. Chaque modèle de tâche inclut les éléments suivants :

- ▶ Un [projet](#) à partir duquel charger les playbooks Ansible
- ▶ Un [inventaire](#) ou une liste de cibles d'automatisation telles que des commutateurs réseau
- ▶ Un [identifiant](#) pour vous connecter aux appareils de votre inventaire et les automatiser
- ▶ Un [environnement d'exécution](#) pour extraire les dépendances nécessaires à la tâche d'automatisation

Exécuter une première commande et un premier playbook



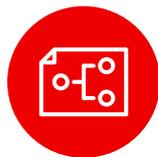
Se familiariser avec les playbooks

Les playbooks sont le langage utilisé par Ansible pour la configuration, le déploiement et l'orchestration. Ils se composent d'un ensemble d'instructions en langage naturel appelées « plays » qui définissent l'automatisation pour tout un inventaire d'hôtes. Chaque play comprend une ou plusieurs tâches qui ciblent un, plusieurs ou tous les hôtes d'un inventaire. Chacune des tâches appelle un module Ansible qui se charge d'une fonction spécifique, comme recueillir des informations utiles, sauvegarder des fichiers réseau, gérer la configuration des réseaux, exécuter des commandes ou valider la connectivité.

Les playbooks Ansible peuvent être partagés et réutilisés par plusieurs équipes afin de créer un schéma d'automatisation reproductible.



Développement



Ingénierie



Exploitation informatique



Exploitation réseau



Externalisations



Playbooks Ansible

Anatomie d'un playbook

L'exemple suivant montre les parties principales d'un playbook Ansible.

```

1 ---
2 - name: Add VLANs
3   hosts: arista
4   gather_facts: false
5
6   vars:
7     vlans:
8       - name: desktops
9         vlan_id: 20
10      - name: servers
11        vlan_id: 30
12      - name: DMZ
13        vlan_id: 50
14
15   tasks:
16     - name: Add VLAN configuration
17       arista.eos.eos_vlans:
18         state: merged
19         config: "{{ vlans }}"

```

Indique le début d'un playbook

Appelle un périphérique ou groupe de périphériques nommé `arista`

Paramètre facultatif pour récupérer des faits

Définition de la variable

Dans ce playbook, la valeur des variables est définie directement.

Si vous utilisez l'interface web d'Ansible Automation Platform, vous pouvez aussi [créer un formulaire](#) pour inviter vos utilisateurs à saisir la valeur des variables lorsqu'ils exécutent votre playbook. Dans ce cas, il faut remplacer les lignes 9, 11 et 13 par :

```
# {{variable_name}} input
```

Pour en savoir plus, lisez la section [Créer un formulaire Ansible Automation Platform](#) à la page 7.

Tâches

Les tâches et les modules Ansible ont une corrélation 1:1. Cette section appelle des modules afin de configurer des réseaux locaux virtuels (VLAN) pour chacune des trois variables définies dans la section `vars`.

Créer un formulaire Red Hat Ansible

Les formulaires définissent des variables supplémentaires pour votre playbook dans un format question/réponse pratique pour les utilisateurs. Pour créer un formulaire :

1. Cliquez sur le bouton **Add survey** (Ajouter un formulaire) dans l'interface web d'Ansible Automation Platform.
2. Pour chaque question, indiquez les informations suivantes :
 - ▶ **Question** : la question à poser à l'utilisateur
 - ▶ **Description (facultative)** : une description de la réponse attendue
 - ▶ **Nom de la variable de réponse** : le nom de la variable Ansible dans laquelle la réponse sera stockée
 - ▶ **Type de réponse** : le format de la réponse (une ou plusieurs lignes de texte, mot de passe, choix multiple, nombre entier ou décimal)
 - ▶ **Réponse par défaut (facultative)** : la valeur par défaut de la variable
 - ▶ **Longueur minimale et maximale (facultative)** : la longueur de réponse minimale et maximale autorisée
 - ▶ **Obligatoire** : indique si la question est facultative ou non
3. Cliquez sur le bouton + pour ajouter la question à votre formulaire.
4. Répétez l'étape 3 pour ajouter d'autres questions à votre formulaire.
5. Cliquez sur le bouton **Save** (Enregistrer) pour enregistrer le formulaire lorsque vous avez terminé.

Pour en savoir plus, lisez la [section sur les formulaires](#) dans la documentation d'Ansible Automation Platform.

Exécuter votre playbook

L'exécution d'un playbook n'a rien de compliqué. Toutefois, la procédure est différente entre l'interface en ligne de commande d'Ansible et l'interface web d'Ansible Automation Platform.

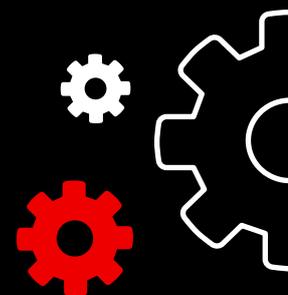
Ligne de commande Ansible

Exécutez la commande suivante :

```
ansible-navigator run <playbook name> -i <inventory file>
```

Interface web de Red Hat Ansible Automation Platform

Cliquez sur le bouton de lancement de la tâche (fusée) à côté de votre modèle sur l'interface web d'Ansible Automation Platform.



Créer votre inventaire

Comprendre les inventaires

Un **inventaire** est une collection d'hôtes sur lesquels vous pouvez agir à l'aide de commandes et de playbooks Ansible. Les fichiers d'inventaire classent les hôtes en groupes et peuvent servir de source fiable pour votre réseau. Ces fichiers peuvent avoir des formats simples, comme INI ou YAML. De nombreuses entreprises choisissent le langage YAML pour leurs inventaires afin d'assurer la cohérence avec leurs playbooks. Avec un fichier d'inventaire, un playbook unique peut assurer la maintenance de centaines de périphériques réseau à l'aide d'une seule commande.

Ce chapitre explique comment créer un fichier d'inventaire.

Créer un inventaire de base au format INI

Commencez par créer des groupes logiques dans votre inventaire. Les meilleures pratiques conseillent de regrouper les serveurs et périphériques réseau selon leur **fonction** (*quoi*) (application, pile ou microservice), leur **localisation** (*où*) (datacenter ou région) et l'**étape** (*quand*) (étape de développement).

- ▶ **Quoi**: db, web, leaf, spine
- ▶ **Où**: east, west, floor_19, building_A
- ▶ **Quand**: dev, test, staging, prod

Cet exemple de code, au format INI, illustre une structure de groupe de base pour un petit datacenter. Vous pouvez rassembler les groupes à l'aide de la syntaxe `[metagroupname:children]`, en listant les groupes comme membres du « metagroup ».

Ici, le groupe `network` inclut toutes les « leafs » et les « spines ». Le groupe `datacenter` inclut tous les périphériques réseau et tous les serveurs web.

Pour en savoir plus, lisez la [section Créer votre inventaire](#) de la documentation d'Ansible. Vous trouverez également un [exemple de playbook de rapport d'inventaire](#) sur GitHub.

```
1  [leafs]
2  leaf01
3  leaf02
4
5  [spines]
6  spine01
7  spine02
8
9  [network:children]
10 leafs
11 spines
12
13 [webserver]
14 webserver01
15 webserver02
16
17 [datacenter:children]
18 network
19 webserver
```

Anatomie d'un inventaire au format YAML

```

1 ---
2 all:
3   vars:
4     ansible_user: admin
5     ansible_password: password123
6     ansible_become_pass: password123
7     ansible_become: True
8     ansible_become_method: enable
9     ansible_network_cli_ssh_type: libssh
10  children:
11    routers:
12      children:
13        arista:
14        cisco:
15        juniper:
16  arista:
17    hosts:
18      rtr2:
19        ansible_host: 172.16.100.2
20      rtr4:
21        ansible_host: 172.16.100.4
22    vars:
23      ansible_network_os: arista.eos.eos
24      ansible_connection: ansible.netcommon.network_cli
25  cisco:
26    hosts:
27      rtr1:
28        ansible_host: 172.16.100.1
29    vars:
30      ansible_network_os: cisco.ios.ios
31      ansible_connection: ansible.netcommon.network_cli
32  juniper:
33    hosts:
34      rtr3:
35        ansible_host: 172.16.100.3
36    vars:
37      ansible_network_os: junipernetworks.junos.junos
38      ansible_connection: ansible.netcommon.netconf

```

Indique le début d'un playbook

Définit les variables applicables à tous les hôtes de l'inventaire, quel que soit le groupe

Hierarchie du groupe

Les lignes 10 à 15 identifient les groupes d'hôtes dans cet inventaire. Dans ce cas, le groupe `routers` contient trois sous-groupes : `arista`, `cisco` et `juniper`.

Définition du groupe

La commande `hosts` indique quels hôtes appartiennent à chaque groupe. Dans cet exemple, le groupe `arista` contient deux hôtes identifiés par leur adresse IP.

Variables de groupe

Chaque groupe peut avoir son propre ensemble de variables. Cet inventaire définit le système d'exploitation et le type de connexion pour chaque groupe. Ces deux variables pointent vers des éléments contenus dans des collections de contenus.

Pour en savoir plus sur ces variables, consultez la [section Grouper vos variables](#) à la page 11.

Créer des groupes par plateforme

Si votre inventaire se développe, il est conseillé de regrouper les périphériques par plateforme. Ainsi, vous pourrez définir facilement des variables spécifiques à une plateforme et applicables à tous les périphériques de cette plateforme. Les lignes 10 à 15 de l'exemple d'inventaire identifient la hiérarchie des groupes d'hôtes pour cet inventaire. Le groupe `routers` contient trois enfants ou sous-groupes classés par plateforme : `arista`, `cisco` et `juniper`. Chacun de ces sous-groupes contient un ou plusieurs hôtes, définis par leur adresse IP aux lignes 16 à 21, 25 à 28 et 32 à 35.

Pour en savoir plus, lisez la [section Créer des groupes par plateforme](#) de la documentation d'Ansible.

```
10  children:
11    routers:
12      children:
13        arista:
14        cisco:
15        juniper:
16  arista:
17    hosts:
18      rtr2:
19        ansible_host: 172.16.100.2
20      rtr4:
21        ansible_host: 172.16.100.4
```

Paramétrer vos variables

Dans l'inventaire, vous pouvez définir les valeurs de nombreuses variables dont vous avez eu besoin pour votre première commande et ainsi les omettre dans la commande `ansible-playbook`. Les lignes 2 à 9 de l'exemple d'inventaire définissent des variables qui s'appliquent à tous les hôtes listés dans l'inventaire, quel que soit leur groupe.

Vous pouvez paramétrer et stocker des variables dans plusieurs fichiers différents. Les meilleures pratiques conseillent de paramétrer les variables utilisées pour se connecter aux périphériques, notamment des informations de connexion ou une adresse IP, dans des fichiers d'inventaire ou des playbooks. Configurez les variables en lien avec la configuration d'un périphérique dans des fichiers séparés stockés dans le répertoire `group_vars`. Pour en savoir plus, lisez la [section Organiser les variables de groupe et d'hôte](#) de la documentation d'Ansible.

```
2  all:
3    vars:
4      ansible_user: admin
5      ansible_password: password123
6      ansible_become_pass: password123
7      ansible_become: True
8      ansible_become_method: enable
9      ansible_network_cli_ssh_type: libssh
```

Grouper vos variables

Lorsque des périphériques dans un groupe partagent les mêmes valeurs de variables, notamment le système d'exploitation ou l'utilisateur SSH, vous pouvez réduire les doublons et simplifier la maintenance en consolidant ces valeurs dans des variables de groupe. Les variables de groupe sont paramétrées au sein de leurs définitions de groupes respectives. Les lignes 22 à 24, 29 à 31 et 36 à 38 de l'exemple d'inventaire configurent des valeurs de variables de groupe pour chacun des trois groupes hôtes.

```
22     vars:
23         ansible_network_os: arista.eos.eos
24         ansible_connection: ansible.netcommon.network_cli
```

Cet exemple définit les variables de système d'exploitation réseau et de type de connexion pour chacun des trois sous-groupes. Dans ce cas, ces variables pointent vers des éléments contenus dans des collections Ansible Content Collections. Les éléments de ces collections ont le format suivant : `namespace.collection_name.item`. Par exemple, `arista.eos.eos` pointe vers le plug-in de système d'exploitation EOS dans la collection EOS servie par l'espace de noms Arista, tandis que `ansible.netcommon.network_cli` pointe vers le plug-in de la CLI réseau dans la collection Netcommon servie par l'espace de noms Ansible.

Syntaxe des variables

La syntaxe des valeurs de variables est différente dans l'inventaire, les playbooks et les fichiers `group_vars`. Même si les playbooks et les fichiers `group_vars` sont tous deux écrits en YAML, les variables s'utilisent différemment.

Fichiers d'inventaire de style INI

Utilisez la syntaxe `key=value` pour les valeurs des variables :

```
ansible_network_os=cisco.ios.ios
```

Fichiers `group_vars` et playbook

Utilisez le nom complet de la clé :

```
ansible_network_os: cisco.ios.ios
```

Fichiers avec extensions `.YML` et `.YAML`

Utilisez la syntaxe YAML :

```
key: value
```

Pour en savoir plus, lisez la [section Syntaxe des variables](#) de la documentation d'Ansible.

Protéger les variables sensibles

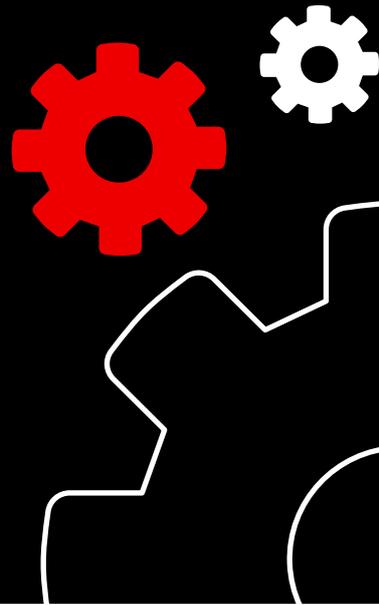
Les meilleures pratiques conseillent d'utiliser une protection supplémentaire pour les variables sensibles, telles que les mots de passe.

La solution Ansible Automation Platform fournit un gestionnaire d'identifiants pour les mots de passe et les informations de clé. Sur la page **Credentials** (Informations d'identification) de l'interface web, vous pouvez autoriser vos utilisateurs et vos équipes à utiliser les informations d'identification sans les leur divulguer. Pour en savoir plus, lisez la [section sur les informations d'identification](#) de la documentation d'Ansible.

Notez que la plateforme Ansible Automation Platform peut être exécutée sur des systèmes où le [mode FIPS \(Federal Information Processing Standards\)](#) est activé.



Mettre en œuvre des cas d'utilisation fréquents



Ce chapitre présente des exemples de playbooks pour des cas d'utilisation courants de l'automatisation réseau, comme l'ajout d'un VLAN, le regroupement de faits, la récupération d'informations sur une ressource et la sauvegarde des configurations.

Ajouter un VLAN

La configuration d'un VLAN sur plusieurs périphériques réseau est une activité récurrente pour l'équipe NetOps. Ansible facilite la création d'un VLAN et sa propagation sur votre réseau.

```
1 ---
2 - name: Add VLANs
3   hosts: arista
4   gather_facts: false
5   vars:
6     vlans:
7       - name: desktops
8         vlan_id: 20
9       - name: servers
10        vlan_id: 30
11       - name: DMZ
12        vlan_id: 50
13
14   tasks:
15     - name: Add VLAN configuration
16       arista.eos.eos_vlans:
17         state: merged
18         config: "{{ vlans }}"
```

Regrouper des faits

La plupart des réseaux comprennent un grand nombre de plateformes et de périphériques différents. Ansible peut interroger les données de votre réseau (versions logicielles, informations sur les interfaces, etc.), les stocker et générer des rapports.

```
1 ---
2 - name: Use facts module
3   hosts: cisco
4   gather_facts: false
5
6   tasks:
7     - name: Retrieve facts
8       cisco.ios.ios_facts:
9
10    - name: Display version
11      debug:
12        msg: "{{ ansible_net_version }}"
13
14    - name: Display serial number
15      debug:
16        msg: "{{ ansible_net_serialnum }}"
```

Récupérer des informations sur une ressource

Les **modules de ressource réseau** Ansible simplifient et standardisent la gestion de plusieurs périphériques réseau. Tous les modules de ressource peuvent utiliser `state: gathered` pour récupérer des informations à propos de ressources réseau.

```
1 ---
2 - name: Retrieve interface information
3   hosts: cisco
4   gather_facts: false
5
6   tasks:
7     - name: Use state gathered
8       cisco.ios.ios_interfaces:
9         state: gathered
10      register: interfaces_info
11
12    - name: Print out interfaces information
13      debug:
14        msg: "{{ interfaces_info }}"
```

Sauvegarder des configurations

Le stockage des sauvegardes des configurations est une activité essentielle pour les équipes NetOps. Ansible Automation Platform facilite l'extraction des données relatives à l'ensemble ou à certaines parties seulement d'une configuration depuis un périphérique réseau.

```
1 ---
2 - hosts: cisco
3   gather_facts: false
4
5   tasks:
6     - name: Back up config
7       cisco.ios.ios_config:
8         backup: yes
```

Accéder au contenu utilisable avec Ansible Automation Platform



Vous pouvez accéder à du contenu d'automatisation prêt à l'emploi afin d'adopter Red Hat Ansible Automation Platform plus rapidement.

Ansible Content Collections

Une collection [Ansible Content Collection](#) représente un format standardisé de distribution de contenu Ansible qui peut inclure des exemples de playbooks, des rôles, des modules, des plug-ins et bien plus encore. Ansible Automation Platform donne accès à des collections Ansible Content Collections certifiées et prises en charge, afin d'augmenter les capacités de la plateforme, d'intégrer des technologies tierces, d'étendre l'automatisation à d'autres domaines et de faciliter son adoption. Vous pouvez également accéder à des collections de contenus validés qui incluent des références testées et orientées (« opinionated »), notamment des protocoles de routage, des interfaces, des listes d'accès et des configurations de mise en réseau de base. Vous pouvez installer des [collections de contenus certifiés Red Hat Ansible](#) entièrement prises en charge et accéder à des [contenus de référence validés](#) depuis [Ansible Automation Hub](#), un composant inclus dans votre souscription Ansible Automation Platform.

Rôles Ansible

Les rôles Ansible groupent le contenu d'automatisation pour le rendre réutilisable. Au lieu de créer de longs playbooks contenant des centaines de tâches, vous pouvez classer et diviser les tâches en unités de travail distinctes plus petites en utilisant les rôles. Un rôle inclut l'ensemble des tâches, variables et gestionnaires nécessaires pour former une unité de travail complète. Les rôles sont distribués comme des entités autonomes ou des parties d'Ansible Content Collections.

Ansible Automation Hub

[Ansible Automation Hub](#) donne aux titulaires de souscription Red Hat Ansible Automation Platform accès à des collections de contenus entièrement prises en charge et certifiées, développées, testées et mises à jour par Red Hat et ses partenaires technologiques. Il vous fournit un portail sécurisé vers Ansible Content Collections ainsi qu'un réseau privé pour le contenu d'automatisation interne et tiers. Automation Hub est le référentiel standard pour le contenu utilisé dans les environnements d'automatisation en production.

Ansible Galaxy

[Ansible Galaxy](#) abrite toutes les collections Ansible communautaires, ainsi que les rôles autonomes. Vous pouvez aussi contribuer à la communauté en déposant dans Ansible Galaxy les collections et rôles que vous créez.

Aller encore plus loin dans l'automatisation

Créer des workflows d'automatisation avancés

Après avoir acquis de l'expérience et pris confiance dans votre gestion des réseaux avec Ansible Automation Platform, vous pouvez vous lancer dans des cas d'utilisation avancés qui font appel à l'automatisation orientée événements. Cette dernière est la suite logique de la transition vers une automatisation informatique et réseau de bout en bout. Elle permet de prédéterminer vos réponses aux changements des conditions réseau, en évitant toute intervention manuelle. Inclus avec Ansible Automation Platform, le composant **Event-Driven Ansible** s'appuie sur les mêmes briques élémentaires que les playbooks, pour une mise en œuvre facile à tout moment. Par exemple, vous pouvez utiliser Event-Driven Ansible pour regrouper automatiquement des faits et ainsi améliorer les tickets d'assistance, gérer les tâches d'administration des utilisateurs telles que la réinitialisation des mots de passe, ou encore résoudre des problèmes simples.

Vous pouvez également utiliser **Red Hat Ansible Lightspeed with IBM watsonx Code Assistant** afin que vos équipes d'automatisation apprennent, créent et gèrent des contenus Ansible Automation Platform (comme des playbooks) plus efficacement. Ce service d'intelligence artificielle (IA) générative accélère la création, la recherche, la compréhension et l'optimisation de vos contenus d'automatisation.

En savoir plus

Red Hat met à votre disposition de nombreuses ressources sur la solution Ansible Automation Platform, notamment de la documentation détaillée, des articles, des vidéos et des forums de discussion. La plupart de ces ressources sont disponibles à l'adresse ansible.com et sur le [portail client Red Hat](#).

- ▶ Site web du produit :
[Red Hat Ansible Network Automation](#)
- ▶ Travaux pratiques :
[Ateliers interactifs Ansible Automation Platform](#)
- ▶ Documentation :
[Index des plateformes réseau](#)
[Documentation communautaire d'Ansible](#)
- ▶ Guides de l'utilisateur :
[Inventaires et variables](#)
[Formulaires](#)
[Informations d'identification](#)
- ▶ Livre numérique :
[Guide sur l'automatisation des réseaux](#)
- ▶ Cours de formation
[Les bases d'Ansible : présentation technique de l'automatisation](#)
[Ansible pour l'automatisation du réseau](#)
- ▶ Version d'essai gratuite :
[version d'évaluation d'Ansible Automation Platform \(à télécharger\)](#)

Vous souhaitez automatiser votre réseau ?

À l'aide d'un langage naturel compréhensible et intuitif, la solution Red Hat Ansible Automation Platform vous offre une manière simple et puissante de moderniser l'exploitation de votre réseau, tout en prenant en charge vos processus et votre infrastructure actuels. Avec son framework flexible et évolutif, vous pouvez plus facilement améliorer le niveau de disponibilité de l'infrastructure, de productivité de vos équipes, de sécurité de vos réseaux et de conformité de la configuration.

Essayer gratuitement Ansible Automation Platform : red.ht/try_ansible

Accélérer le déploiement avec l'aide des spécialistes Red Hat

L'automatisation du réseau peut sembler difficile, mais l'équipe de consulting Red Hat peut vous aider. Tous les contrats de consulting Red Hat débutent par une session de découverte gratuite d'une demi-journée sur site. Au cours de cette session, les spécialistes Red Hat vous aident à identifier vos défis les plus urgents, les meilleures approches pour les relever et les résultats souhaités pour la mise en œuvre de l'automatisation de votre réseau.

Organiser une session de découverte gratuite : redhat.com/consulting