

# Utilisation du client VMware vRealize Orchestrator

Février 2022

vRealize Orchestrator 8.7

Vous trouverez la documentation technique la plus récente sur le site Web de VMware, à l'adresse :

<https://docs.vmware.com/fr/>

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

**VMware France SAS.**  
Tour Franklin  
100-101 Terrasse Boieldieu  
92042 Paris La Défense 8 Cedex  
France  
[www.vmware.com/fr](http://www.vmware.com/fr)

Copyright © 2008-2022 VMware, Inc. Tous droits réservés. [Informations relatives aux copyrights et marques commerciales.](#)

# Table des matières

<b>1</b>	<b>Utilisation du client VMware vRealize Orchestrator</b>	<b>6</b>
<b>2</b>	<b>Présentation de vRealize Orchestrator Client</b>	<b>7</b>
	Tableau de bord de l'utilisation de vRealize Orchestrator Client	8
	Organisation de contenu dans vRealize Orchestrator Client	8
	Créer des dossiers ou des sous-dossiers	10
	Déplacer des objets et des dossiers	11
	Supprimer des dossiers ou des sous-dossiers	11
<b>3</b>	<b>Configuration de vRealize Orchestrator Client</b>	<b>13</b>
	Rôles et groupes de vRealize Orchestrator	13
	Attribuer des rôles dans vRealize Orchestrator Client	15
	Configurer des rôles vRealize Orchestrator Client dans vRealize Automation	16
	Créer des Groupes dans vRealize Orchestrator Client	17
	Historique des versions des objets vRealize Orchestrator	17
	Restaurer un workflow à une version antérieure	18
	Comparaison visuelle entre les versions de workflow	19
	Réinitialiser un inventaire de contenu vRealize Orchestrator à un état antérieur avec Git	20
<b>4</b>	<b>Cas d'utilisation de vRealize Orchestrator</b>	<b>22</b>
	Comment intégrer des services Amazon Web Services dans vRealize Orchestrator à l'aide de Python	22
	Créer un script Python initial	23
	Créer l'action Amazon Web Services (AWS)	24
	Déboguer l'action Amazon Web Services	25
	Mettre à jour l'action Amazon Web Services	29
	Comment puis-je utiliser le branchement Git pour gérer mon inventaire d'objets vRealize Orchestrator	29
	Préparer votre environnement GitLab	30
	Configurer une connexion à un référentiel Git	31
	Transférer les modifications vers un référentiel Git	32
	Comment utiliser des modules tiers pour appeler vRealize Automation Project API	35
	Créer un script Python qui appelle vRealize Automation Project API	35
	Créer un script Node.js qui appelle vRealize Automation Project API	37
	Créer un script PowerShell qui appelle vRealize Automation Project API	39
<b>5</b>	<b>Gestion des workflows</b>	<b>43</b>
	Workflows standard de la bibliothèque de workflows vRealize Orchestrator	44

Créer des workflows	44
Modifier les workflows et les actions du workflow parent	45
Concepteur de formulaires de saisie vRealize Orchestrator	45
Créer la boîte de dialogue des paramètres d'entrée de workflow dans vRealize Orchestrator Client	46
Propriétés des paramètres d'entrée dans vRealize Orchestrator Client	46
Utilisation d'actions pour valider les entrées de workflow vRealize Orchestrator	47
Demandes d'interaction des utilisateurs dans le client vRealize Orchestrator	49
Planifier des workflows	49
Modifier une tâche planifiée dans le client vRealize Orchestrator	50
Rechercher des références d'objets dans les workflows	51
<b>6 Gestion des actions</b>	<b>52</b>
Créer des actions	52
Exécution et débogage d'actions	53
Exécuter des actions	54
Actions de débogage	54
Concepts de base pour les scripts Python, Node.js et PowerShell	55
Limites d'exécution pour les scripts Python, Node.js et PowerShell	57
<b>7 Gestion des éléments de configuration</b>	<b>58</b>
Créer des éléments de configuration	58
<b>8 Gestion des stratégies</b>	<b>60</b>
Créer et appliquer des stratégies	60
Éléments de la stratégie	61
Gérer les exécutions de stratégies	62
<b>9 Gestion des éléments de ressource</b>	<b>63</b>
<b>10 Gestion des modules</b>	<b>64</b>
Créer des modules	64
Exporter des modules	65
Importer des modules	66
<b>11 Dépannage dans le client vRealize Orchestrator</b>	<b>68</b>
Données de mesure dans le client vRealize Orchestrator	68
Profiler des workflows dans le client vRealize Orchestrator	68
Utilisation du tableau de bord système de vRealize Orchestrator	69
Utilisation de la relecture de jeton de workflow dans vRealize Orchestrator Client	70
Validation des workflows vRealize Orchestrator	71

Valider un workflow et corriger les erreurs de validation dans vRealize Orchestrator Client	72
Déboguer les scripts de workflow dans vRealize Orchestrator Client	73
Workflows de débogage par élément de schéma	74
Configuration d'un conteneur Photon OS pour les modules Python	75

# Utilisation du client VMware vRealize Orchestrator

# 1

La section *Utilisation de VMware vRealize Orchestrator Client* fournit des informations sur les fonctionnalités d'automatisation de workflow et sur le fonctionnement de vRealize Orchestrator Client.

## Public visé

Ces informations sont destinées aux administrateurs système expérimentés qui recherchent un outil leur permettant d'exécuter et de gérer les workflows vRealize Orchestrator.

# Présentation de vRealize Orchestrator Client

# 2

Utilisez vRealize Orchestrator Client pour gérer les services et les objets vRealize Orchestrator.

Vous pouvez accéder à vRealize Orchestrator Client à l'adresse `https://your_orchestrator_FQDN/orchestration-ui`.

Élément de l'interface utilisateur	Description
Tableau de bord	Utilisez le tableau de bord et la fonctionnalité de profilage de vRealize Orchestrator Client pour collecter des données de mesure utiles sur votre environnement et vos workflows vRealize Orchestrator.
Workflows	Créez, modifiez, planifiez, exécutez et supprimez des workflows.
Actions	Créez, modifiez et supprimez des actions. L'éditeur d'actions prend en charge la saisie automatique des éléments de script communs inclus dans l'explorateur d'API vRealize Orchestrator.
Stratégies	Créez, modifiez, exécutez et supprimez des stratégies.
Modules	Créez, supprimez, exportez et importez des modules contenant des objets vRealize Orchestrator.
Configurations	Créez, exécutez et supprimez des éléments de configuration.
Ressources	Exportez, importez et mettez à jour des éléments de ressource.
Groupes	Les utilisateurs disposant de droits d'administrateur peuvent attribuer des rôles aux utilisateurs dans vRealize Orchestrator Client et les ajouter à des groupes.
Journaux d'audit	Affichez les différents événements, tels que la date de création d'un objet, qui sont enregistrés dans vRealize Orchestrator Client.
Référentiels Git	Créez une intégration à un référentiel Git et utilisez l'intégration pour gérer le développement de workflows et d'autres objets de vRealize Orchestrator dans plusieurs déploiements. Reportez-vous à <a href="#">Comment puis-je utiliser le branchement Git pour gérer mon inventaire d'objets vRealize Orchestrator</a> .
Éléments supprimés	Restaurer les objets de vRealize Orchestrator Client supprimés, tels que des workflows, des actions, des stratégies, des éléments de configuration et des éléments de ressource.
Explorateur d'API	Explorez les commandes d'API disponibles dans vRealize Orchestrator Client.  <b>Note</b> vRealize Orchestrator Client communique avec l'instance de REST API de vRealize Orchestrator via un proxy REST.

Ce chapitre contient les rubriques suivantes :

- [Tableau de bord de l'utilisation de vRealize Orchestrator Client](#)
- [Organisation de contenu dans vRealize Orchestrator Client](#)

## Tableau de bord de l'utilisation de vRealize Orchestrator Client

Le tableau de bord de vRealize Orchestrator Client fournit un outil utile pour surveiller, gérer et dépanner les workflows vRealize Orchestrator Client.

Les informations du tableau de bord de vRealize Orchestrator Client sont réparties entre cinq panneaux.

Fenêtre	Description
Exécutions de workflow	Fournit des données visuelles sur le nombre d'exécutions de workflows en cours, en attente et ayant échoué.
Workflows favoris	Affiche les workflows ajoutés aux Favoris.
En attente d'entrée	Affiche les exécutions de workflows en attente qui nécessitent une intervention supplémentaire de l'utilisateur. Ces workflows s'affichent également dans le menu des notifications dans le coin supérieur droit de l'interface utilisateur.
Exécutions de workflows récentes	Gérez les exécutions de workflows récentes. Ce panneau affiche le nom, l'état, la date de début et de fin de l'exécution de workflow.
Nécessite une attention	Affiche les exécutions de workflows ayant échoué et les mesures de performances des exécutions de workflow.

## Organisation de contenu dans vRealize Orchestrator Client

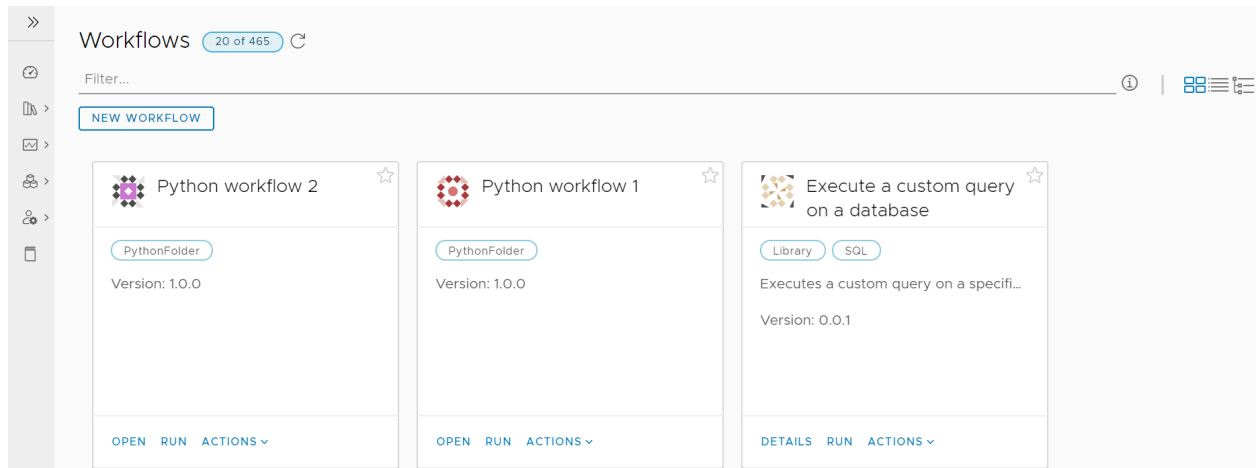
Gérez la manière dont votre inventaire d'objets vRealize Orchestrator s'affiche dans vRealize Orchestrator Client.

vRealize Orchestrator Client prend en charge trois types de vues différents pour les objets tels que les workflows, les actions, les stratégies, les ressources et les configurations : vue Carte, vue Liste et vue Arborescence. Vous pouvez modifier le type de vue actuel dans le coin supérieur droit de la page.

### Vue Carte

La vue Carte est le type de vue par défaut utilisé dans vRealize Orchestrator Client. Les informations sur l'objet d'inventaire individuel, tel qu'un workflow, s'affichent dans un élément de carte distinct.





## Vue Liste

La vue Liste affiche des informations sur vos objets vRealize Orchestrator organisés sous forme de liste. Pour plus d'informations sur les actions que vous pouvez effectuer sur l'objet, cliquez sur l'icône de points de suspension verticaux à gauche de l'objet.

	Name	Tags	Version	Description
⋮	Python workflow 2	PythonFolder	1.0.0	
⋮	Python workflow 1	PythonFolder	1.0.0	
⋮	Execute a custom query on a database	Library SQL	0.0.1	Executes a custom query on a specified database and returns the number of affected rows. You can run the workflow to update, delete, and insert queries.
⋮	JDBC URL generator	Library JDBC	0.0.8	Solicits information to generate a connection URL for JDBC database connections. The workflow emits the connection string it generates as output via the system log, and confirms the string can create a connection to the specified database.

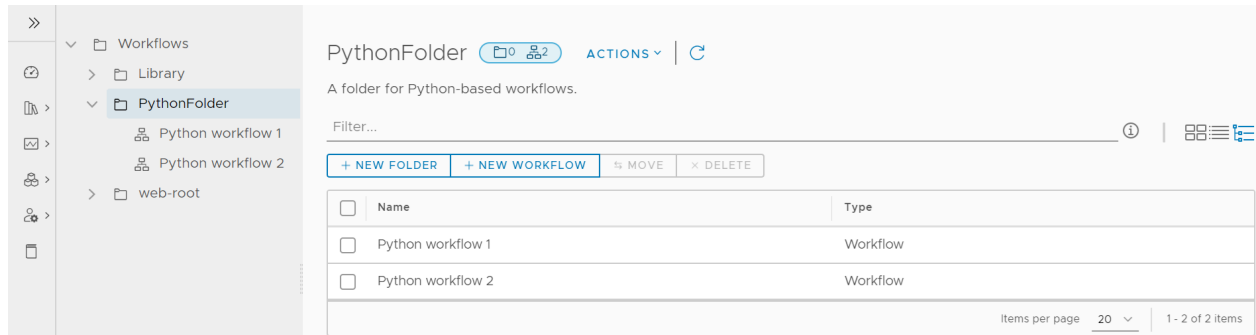
## Vue Arborescence

Vous pouvez organiser votre inventaire d'objets sous les dossiers hiérarchiques dans la vue Arborescence. Chaque type d'objet vRealize Orchestrator dispose d'un dossier de niveau racine. Vous ne pouvez pas créer d'objets, tels que des workflows, sous le dossier racine. Vous devez créer des dossiers distincts organisés sous le dossier racine. Chaque dossier contient des outils pour vous aider à gérer son contenu, tel qu'un filtre de contenu.

**Note** Chaque dossier dispose d'un filtre de contenu distinct. Vous ne pouvez pas filtrer le contenu dans les dossiers.

Pour de plus amples informations sur les dossiers, reportez-vous à la section [Créer un dossier ou un sous-dossier dans vRealize Orchestrator Client](#).

**Note** Lorsque vous sélectionnez un objet dans la vue Arborescence, il s'ouvre en mode lecture seule. Pour modifier le contenu de l'objet, tel que les variables de workflow ou le schéma de workflow, cliquez sur **Modifier** dans le menu des options supérieur.




## Créer un dossier ou un sous-dossier dans vRealize Orchestrator Client

Organisez vos objets vRealize Orchestrator à l'aide d'une structure hiérarchique de dossiers.

Vous pouvez créer des dossiers et des sous-dossiers pour organiser les types d'objets vRealize Orchestrator suivants :

- Workflows
- Actions
- Stratégies
- Éléments de configuration
- Éléments de ressource

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Dans le volet de navigation de gauche, sélectionnez une page d'objet, par exemple **Workflows**.
- 3 Dans le coin supérieur droit, sélectionnez l'icône de la vue Arborescence ()
- 4 (Facultatif) Pour créer un sous-dossier, sélectionnez un dossier parent dans la vue Arborescence à gauche.
- 5 Cliquez sur **Nouveau dossier**.
- 6 Entrez un nom et une description, puis cliquez sur **Enregistrer**.
- 7 Ajoutez des objets et des sous-dossiers au dossier récemment créé.


- 8 (Facultatif) Pour modifier le nom du dossier, sélectionnez **Actions > Modifier**.

## Déplacer des objets et des dossiers dans vRealize Orchestrator Client

Réorganisez votre contenu vRealize Orchestrator en déplaçant le contenu vers un autre dossier.

Vous ne pouvez pas déplacer des actions entre des modules d'action ni déplacer des objets vers un dossier racine. Le dossier racine comporte les principaux dossiers et sous-dossiers d'objets, mais ne peut pas être utilisé pour stocker des objets.

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Dans le volet de navigation de gauche, sélectionnez une page d'objet, par exemple **Workflows**.
- 3 Dans le coin supérieur droit, sélectionnez l'icône de la vue Arborescence () .
- 4 Développez la vue Arborescence et sélectionnez l'objet ou le dossier que vous souhaitez déplacer.
- 5 Faites glisser l'objet ou le dossier vers son nouveau dossier parent.

---

**Note** Vous pouvez également déplacer des objets dans de nouveaux dossiers directement depuis l'éditeur d'objets. Dans l'onglet **Résumé**, cliquez sur **Sélectionner un dossier**, puis sélectionnez le nouveau dossier parent de l'objet. Une autre option de déplacement consiste à sélectionner des objets dans le tableau sur la page Dossier. Cette option est utile pour effectuer des opérations de déplacement par lot qui incluent plusieurs objets vRealize Orchestrator.


---

## Supprimer un dossier ou un sous-dossier dans vRealize Orchestrator Client

Supprimez les dossiers ou les sous-dossiers obsolètes de votre instance de vRealize Orchestrator Client.

Vous ne pouvez pas supprimer le dossier de niveau racine correspondant à chaque type d'objet vRealize Orchestrator.

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Dans le volet de navigation de gauche, sélectionnez une page d'objet, par exemple **Workflows**.
- 3 Dans le coin supérieur droit, sélectionnez l'icône de la vue Arborescence () .

- 4 Cochez la case en regard du dossier que vous souhaitez supprimer.

---

**Note** Pour supprimer un sous-dossier, sélectionnez le dossier parent dans la vue Arborescence, puis cochez la case.

---

- 5 Cliquez sur **Supprimer**.
- 6 Si le dossier sélectionné est vide.
  - a Confirmez que vous voulez supprimer le dossier.
  - b Cliquez sur **Supprimer**.
- 7 Si le dossier sélectionné contient des objets ou des sous-dossiers vRealize Orchestrator Client.
  - a Confirmez que vous voulez supprimer le dossier.
  - b Cliquez sur **Supprimer**.

Vous recevez le message Impossible de supprimer l'élément « nom\_de\_votre\_dossier » : le dossier « nom\_de\_votre\_dossier » n'est pas vide.
  - c Pour supprimer le dossier et tout son contenu, cliquez sur **Forcer la suppression**.
  - d Confirmez que vous souhaitez supprimer le dossier, puis cliquez sur **Supprimer**.

---

**Note** Vous pouvez également effectuer une suppression par lot en sélectionnant plusieurs objets dans le tableau inclus dans le menu Dossier.

---

# Configuration de vRealize Orchestrator Client

## 3

Pour tirer pleinement parti des fonctionnalités de vRealize Orchestrator Client, vous devez configurer vos autorisations d'utilisateur et découvrir comment vous pouvez utiliser l'historique des versions pour gérer vos objets.

Ce chapitre contient les rubriques suivantes :

- [Rôles et groupes de vRealize Orchestrator](#)
- [Historique des versions des objets vRealize Orchestrator](#)

## Rôles et groupes de vRealize Orchestrator

Les administrateurs vRealize Orchestrator peuvent définir des autorisations qui contrôlent l'accès aux fonctionnalités et au contenu dans vRealize Orchestrator Client. Les droits d'accès sont séparés selon les autorisations des rôles et des groupes d'utilisateurs.

Les rôles vous permettent de contrôler les fonctionnalités de vRealize Orchestrator Client que les utilisateurs peuvent afficher et utiliser. L'accès à la fonctionnalité de gestion des rôles dépend du type de licence de votre environnement vRealize Orchestrator.

Tableau 3-1. Accès basé sur les licences pour la gestion des rôles vRealize Orchestrator

Licence	Authentification	
	vSphere	vRealize Automation
vSphere	La gestion des rôles n'est pas prise en charge. Les groupes prennent uniquement en charge les autorisations d'exécution.	
vRealize Automation	Gérez les rôles dans le client vRealize Orchestrator. Reportez-vous à <a href="#">Attribuer des rôles dans vRealize Orchestrator Client</a> .	Gérez les rôles via l'option <b>Gestion des identités et des rôles</b> dans vRealize Automation. Reportez-vous à <a href="#">Configurer des rôles vRealize Orchestrator Client dans vRealize Automation</a> .

Les autorisations de groupe contrôlent ce que les utilisateurs de contenu vRealize Orchestrator Client peuvent afficher et utiliser, tels que les workflows, les actions, les stratégies, les éléments de configuration et les éléments de ressources. L'accès au contenu préconfiguré du système vRealize Orchestrator, tel que les workflows et les actions standard, est partagé entre tous les utilisateurs, sauf s'il est configuré autrement via des autorisations de groupe.

Les droits d'accès des utilisateurs disposant des rôles d'administrateur et d'observateur ne sont pas limités par les autorisations de groupe. Les droits d'accès des utilisateurs sans rôle attribué et les utilisateurs disposant d'un rôle de concepteur de workflow dépendent du groupe qui leur est attribué. Vous pouvez étendre les droits d'accès de ces utilisateurs en modifiant leurs autorisations de groupe. Cela vous permet d'organiser les utilisateurs en projets communs. Par exemple, vous pouvez créer un groupe qui inclut les utilisateurs qui travaillent sur le développement d'un plug-in vRealize Orchestrator personnalisé et leur permettre de modifier uniquement le contenu spécifique au groupe.

**Tableau 3-2. Autorisations des rôles et des groupes d'utilisateurs vRealize Orchestrator**

Rôle	Droits d'accès	
Administrateur	<p>Les administrateurs peuvent accéder à toutes les fonctionnalités et au contenu du client vRealize Orchestrator, notamment le contenu créé par des groupes spécifiques. Responsable de la définition des rôles d'utilisateur, de la création et de la suppression de groupes et de l'ajout d'utilisateurs à des groupes. Les administrateurs ne sont pas limités par les autorisations de groupe.</p> <p><b>Note</b> Par défaut, les administrateurs de locataires des environnements vRealize Automation utilisés pour authentifier vRealize Orchestrator disposent des droits <b>Administrateur</b>.</p>	
Observateur	<p>Les observateurs disposent d'un accès en lecture seule à tout le contenu dans le client vRealize Orchestrator, mais ne peuvent pas créer, modifier, exécuter, ni exporter du contenu. Ils peuvent également afficher tous les groupes et le contenu des groupes. Les utilisateurs ne sont pas limités par les autorisations de groupe.</p>	
Autorisations de groupe		
Aucun groupe attribué	Exécuter	Exécuter et modifier

Tableau 3-2. Autorisations des rôles et des groupes d'utilisateurs vRealize Orchestrator (suite)

Rôle	Droits d'accès		
Concepteur de Workflow	■ Affichez le contenu du système.	■ Afficher le contenu du système	■ Affichez le contenu du système.
	■ Affichez et exécutez vos propres exécutions.	■ Affichez et exécutez vos propres exécutions.	■ Affichez et exécutez vos propres exécutions.
	■ Créez, exécutez, modifiez et supprimez votre propre contenu.	■ Créez, exécutez, modifiez et supprimez votre propre contenu.	■ Créez, exécutez, modifiez et supprimez votre propre contenu.
		■ Ajoutez votre propre contenu au groupe.	■ Ajoutez votre propre contenu au groupe.
		■ Exécutez le contenu du groupe, sans toutefois pouvoir le modifier.	■ Exécutez et modifiez le contenu du groupe.
			<b>Note</b> Non disponible pour les instances de vRealize Orchestrator authentifiées avec vSphere.
Utilisateur sans rôle attribué	■ Affichez vos propres exécutions.	■ Affichez et exécutez vos propres exécutions.	■ Affichez et exécutez vos propres exécutions.
		■ Affichez et exécutez le contenu du groupe.	■ Affichez et exécutez le contenu du groupe.
			<b>Note</b> Pour pouvoir créer, modifier et ajouter du contenu, un rôle Concepteur de workflow doit être attribué aux utilisateurs de ce groupe.
			<b>Note</b> Non disponible pour les instances de vRealize Orchestrator authentifiées avec vSphere.

## Attribuer des rôles dans vRealize Orchestrator Client

En tant qu'administrateur, vous pouvez ajouter des utilisateurs à vRealize Orchestrator Client et définir les fonctionnalités qu'ils peuvent afficher et utiliser.

La gestion des rôles contrôle l'accès des utilisateurs du fournisseur d'identité vRealize Orchestrator aux fonctionnalités de vRealize Orchestrator Client. La gestion des rôles couvre à la fois l'interface utilisateur de vRealize Orchestrator Client et la fonctionnalité de l'API.

**Note** La gestion des rôles côté client est uniquement disponible pour les instances de vRealize Orchestrator authentifiées avec vSphere qui utilisent une licence vRealize Automation. Pour plus d'informations sur l'attribution de rôles à vRealize Orchestrator authentifié avec vRealize Automation, reportez-vous à [Configurer des rôles vRealize Orchestrator Client dans vRealize Automation](#).

**Procédure**

- 1 Connectez-vous au client vRealize Orchestrator en tant qu'administrateur.
- 2 Accédez à **Administration > Gestion des rôles**.
- 3 Cliquez sur **Ajouter**.
- 4 Recherchez l'utilisateur ou le groupe que vous souhaitez ajouter à vRealize Orchestrator Client.
- 5 Sélectionnez le rôle de l'utilisateur. Pour plus d'informations sur rôles, consultez [Rôles et groupes de vRealize Orchestrator](#).
- 6 Cliquez sur **Enregistrer**.

## Configurer des rôles vRealize Orchestrator Client dans vRealize Automation

Vous pouvez attribuer des rôles de service pour vRealize Orchestrator Client sur la page **Gestion des identités et des accès** dans vRealize Automation. Des rôles de service peuvent être attribués pour l'instance intégrée de vRealize Orchestrator Client et pour les instances autonomes de vRealize Orchestrator authentifiées avec vRealize Automation.

Les rôles de service vRealize Orchestrator gèrent les fonctionnalités auxquelles ont accès les utilisateurs de l'instance intégrée de vRealize Orchestrator Client. Pour plus d'informations sur les rôles vRealize Orchestrator, reportez-vous à [Rôles et groupes de vRealize Orchestrator](#).

---

**Note** Les instances autonomes de vRealize Orchestrator authentifiées avec vSphere qui utilisent une licence vRealize Automation peuvent attribuer des rôles directement dans vRealize Orchestrator Client. Reportez-vous à [Attribuer des rôles dans vRealize Orchestrator Client](#).

---

**Conditions préalables**

- Vérifiez que les utilisateurs et les groupes appropriés sont importés à partir d'une instance valide de vIDM.
- Avant d'attribuer un rôle de service vRealize Orchestrator à votre utilisateur, vérifiez que celui-ci dispose d'un rôle d'organisation attribué dans vRealize Automation. Consultez la section *Administration d'utilisateurs et de groupes dans vRealize Automation* dans *Administration de vRealize Automation*.

**Procédure**

- 1 Dans le menu déroulant d'en-tête en haut à droite, sélectionnez l'option **Gestion des identités et des accès**.
- 2 Dans l'onglet **Utilisateurs actifs**, recherchez l'adresse e-mail de l'utilisateur que vous souhaitez attribuer à vRealize Orchestrator.
- 3 Cochez la case en regard de l'utilisateur, puis cliquez sur **Modifier les rôles**.
- 4 Cliquez sur **Ajouter l'accès au service**.
- 5 Dans le menu déroulant de gauche, sélectionnez **Orchestrator**.



- 6 Dans le menu déroulant de droite, sélectionnez le rôle que vous voulez attribuer à l'utilisateur.
- 7 Cliquez sur **Enregistrer**.

## Créer des Groupes dans vRealize Orchestrator Client

En tant qu'administrateur, vous pouvez utiliser des groupes pour définir les informations que les utilisateurs de contenu de vRealize Orchestrator peuvent afficher et auxquelles ils peuvent accéder dans vRealize Orchestrator Client.

Vous pouvez utiliser vRealize Orchestrator Client pour définir des autorisations de groupe sur les workflows, les actions, les stratégies, les éléments de configuration, les éléments de ressource et les modules de vRealize Orchestrator.

---

**Note** Les utilisateurs d'instances de vRealize Orchestrator authentifiées avec vSphere peuvent uniquement disposer d'autorisations de groupe **Exécuter**.

---

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator en tant qu'administrateur.
- 2 Accédez à **Administration > Groupes**.
- 3 Cliquez sur **Nouveau groupe**.
- 4 Dans l'onglet **Résumé** ajoutez un nom et une description du groupe.
- 5 Sous l'onglet **Utilisateurs**, cliquez sur **Ajouter**.
  - a Recherchez un utilisateur que vous souhaitez ajouter au groupe.
  - b Attribuez des autorisations de groupe à l'utilisateur.
  - c Cliquez sur **Ajouter**.
- 6 Dans l'onglet **Éléments**, ajoutez des objets vRealize Orchestrator au groupe.

---

**Note** Vous pouvez également ajouter un objet à des groupes existants lors de la création de l'objet dans vRealize Orchestrator Client. Pour ajouter l'objet, sélectionnez le groupe dans le menu déroulant **Accessible par** dans l'onglet **Résumé/Général** de l'éditeur d'objets.

---

- 7 Cliquez sur **Enregistrer**.

## Historique des versions des objets vRealize Orchestrator

vRealize Orchestrator Client conserve un enregistrement de l'historique des versions pour chaque objet vRealize Orchestrator. À l'aide de l'historique des versions, vous pouvez comparer différentes versions d'objets vRealize Orchestrator et restaurer une version antérieure.

vRealize Orchestrator crée un enregistrement de l'historique des versions de chaque objet vRealize Orchestrator lors de son enregistrement. Les modifications apportées ultérieurement aux objets vRealize Orchestrator créent un nouvel enregistrement de l'historique des versions. Les enregistrements de l'historique des versions précédentes sont conservés et peuvent être utilisés pour suivre les modifications apportées à l'objet et rétablir la version précédente de l'objet. La restauration d'un objet à une version antérieure crée un enregistrement d'historique des versions.

vRealize Orchestrator Client suit l'historique des versions des objets vRealize Orchestrator suivants :

- Workflows
- Actions
- Modules
- Stratégies
- Éléments de ressource
- Éléments de configuration

---

**Note** Les workflows générés ne s'affichent pas dans l'historique des versions de workflows. Par exemple, les workflows générés par le workflow **Générer des workflows CRUD pour une table** ne s'affichent pas sous l'onglet **Historique des versions** et ne peuvent pas être envoyés à des référentiels Git configurés. Pour inclure ces workflows dans l'historique des versions de vRealize Orchestrator, dupliquez les workflows générés.

---

Vous pouvez accéder à l'historique des versions d'un objet à partir de l'onglet **Historique des versions** dans la page de l'éditeur d'objets. Si vous tentez de modifier un objet en même temps qu'un autre utilisateur, cela peut créer un conflit de fusion. Pour résoudre ce conflit de fusion, cliquez sur **Résoudre** à droite du message d'erreur. Dans la fenêtre **Résoudre les conflits**, vous disposez de trois options :

- **Utiliser leurs.** Permet de résoudre le conflit de fusion à l'aide des modifications apportées par l'autre utilisateur.
- **Utiliser nos.** Permet de résoudre le conflit de fusion à l'aide de vos modifications.
- **Résoudre.** Permet de résoudre le conflit de fusion en modifiant le modèle de modification affiché. Si le modèle fourni n'est pas valide, cette option n'est pas disponible.

## Restaurer un workflow à une version antérieure

Vous pouvez restaurer un workflow à une version précédemment enregistrée.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Workflows**, et sélectionnez un workflow.
- 3 Sélectionnez l'onglet **Historique des versions**.

- 4 Pour afficher une comparaison entre les versions, sélectionnez une version de workflow et sélectionnez une autre version dans le menu déroulant **Comparaison par rapport à**.

Une fenêtre s'affiche avec les différences entre la version en cours et la version sélectionnée.

- 5 Pour restaurer le workflow vers une autre version, cliquez sur **Restaurer**.

L'état du workflow est restauré à la version sélectionnée.

---

**Note** Vous pouvez également restaurer une version de workflow à partir de la vue de l'outil de différences graphiques. Reportez-vous à [Comparaison visuelle entre les versions de workflow](#).

---

## Comparaison visuelle entre les versions de workflow

Comparez les modifications entre les versions de workflows à l'aide de l'outil de différences graphiques.

Par défaut, l'historique des versions de vRealize Orchestrator affiche les différences entre les versions de workflow dans un formulaire YAML. Vous pouvez également effectuer une comparaison visuelle entre différentes versions de workflow. Vous pouvez afficher les modifications dans :

- Les informations générales sur le workflow, telles que le numéro de version et la description du workflow.
- Les variables utilisées dans le workflow.
- Les paramètres d'entrée et de sortie du workflow.
- Le schéma du workflow.

### Conditions préalables

Créez un workflow.

### Procédure

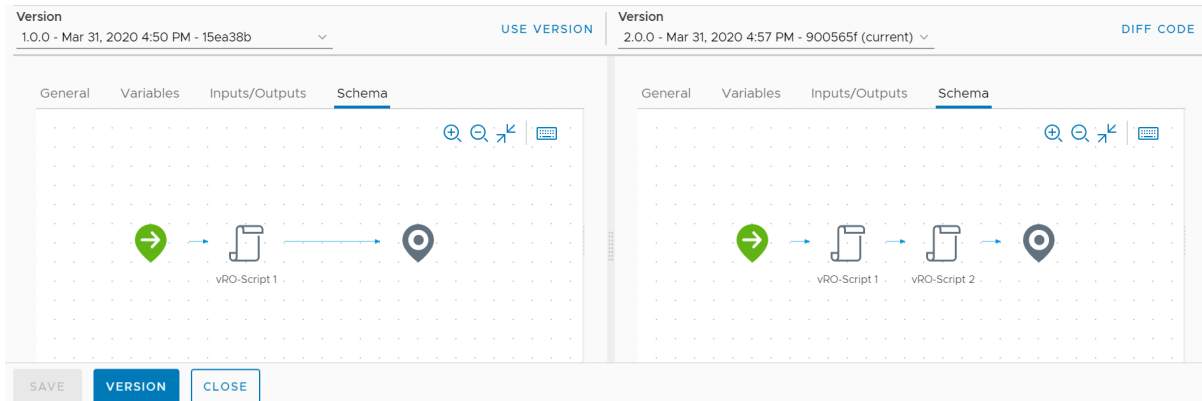
- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Accédez à **Bibliothèque > Workflows**, puis sélectionnez l'un de vos workflows.
- 3 Modifiez le contenu du workflow.

Par exemple, vous pouvez ajouter un élément supplémentaire de **Tâche pouvant contenir des scripts** dans l'onglet **Schéma**.

- 4 Cliquez sur **Enregistrer**.
- 5 Sélectionnez l'onglet **Historique des versions**.

## 6 Dans le coin supérieur droit, sélectionnez **Comparer visuellement**.

Vous pouvez maintenant effectuer une comparaison visuelle entre deux versions de workflow sélectionnées. Vous pouvez sélectionner les versions à comparer dans le menu déroulant **Version**.



## 7 (Facultatif) Vous pouvez restaurer un workflow à une autre version en sélectionnant **Utiliser la version**.

# Réinitialiser un inventaire de contenu vRealize Orchestrator à un état antérieur avec Git

En utilisant une validation Git antérieure, vous pouvez réinitialiser votre contenu vRealize Orchestrator à un état antérieur.

Vous pouvez réinitialiser un contenu vRealize Orchestrator à un état antérieur en sélectionnant un commit spécifique.

### Conditions préalables

- Configurez une connexion à un référentiel GitHub ou GitLab. Reportez-vous à [Configurer une connexion à un référentiel Git](#).
- Transférez un ensemble de modifications locales vers le référentiel Git configuré.

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Accédez à **Administration > Historique Git**.
- 3 Sélectionnez un ensemble de modifications que vous souhaitez réinitialiser et cliquez sur **Réinitialiser vers ce**.
- 4 Confirmez que vous souhaitez revenir à ce commit spécifique et cliquez sur **OK**.

L'inventaire vRealize Orchestrator est réinitialisé à l'état spécifié dans la validation. Le contenu vRealize Orchestrator pertinent est remis à une version antérieure. Si le contenu n'existait pas lors du transfert du commit, il est supprimé de l'inventaire.

### Étape suivante

Pour restaurer l'inventaire vRealize Orchestrator à l'état le plus récent enregistré dans le référentiel Git, exécutez une commande `Pull` à partir de la fenêtre **Historique Git**.

# Cas d'utilisation de vRealize Orchestrator

# 4

Ces cas d'utilisation illustrent une partie des fonctionnalités de la plate-forme vRealize Orchestrator.

Ils exposent des exemples de valeurs uniquement. Votre propre structure d'environnement et les conventions de dénomination peut varier.

Ce chapitre contient les rubriques suivantes :

- [Comment intégrer des services Amazon Web Services dans vRealize Orchestrator à l'aide de Python](#)
- [Comment puis-je utiliser le branchement Git pour gérer mon inventaire d'objets vRealize Orchestrator](#)
- [Comment utiliser des modules tiers pour appeler vRealize Automation Project API](#)

## Comment intégrer des services Amazon Web Services dans vRealize Orchestrator à l'aide de Python

Ce cas d'utilisation de vRealize Orchestrator illustre comment vous pouvez utiliser Python pour développer les capacités de votre déploiement de vRealize Orchestrator.

Vous pouvez utiliser les environnements d'exécution suivants dans vos scripts d'action et de workflow :

- Python 3.7
- Node.js 14
- PowerCLI 11/Powershell 6.2
- PowerCLI 12.3.0/Powershell 7.1

---

**Note** L'exécution de PowerCLI inclut PowerShell et les modules suivants : VMware.PowerCLI, PowerNSX, PowervRA.

---

**Important** Vous pouvez uniquement utiliser les nouveaux environnements d'exécution si votre déploiement vRealize Orchestrator utilise une licence vRealize Automation.

---

Ce cas d'utilisation illustre comment vous pouvez créer un script Python qui appelle des instances d'EC2 dans les services Amazon Web Services (AWS).

---

**Important** Avant de commencer à développer votre script personnalisé, vérifiez que vous maîtrisez bien les concepts principaux d'utilisation des scripts Python, Node.js et PowerShell dans vRealize Orchestrator. Reportez-vous à [Concepts de base pour les scripts Python, Node.js et PowerShell](#).

---

## Procédure

### 1 Créer un script Python initial

Sur votre machine locale, créez votre script Python et compressez le script et une bibliothèque boto3 sous la forme d'un dossier ZIP.

### 2 Créer l'action Amazon Web Services (AWS)

Créez une action vRealize Orchestrator qui utilise votre script Python.

### 3 Déboguer l'action Amazon Web Services

La version d'origine du script Python comporte une erreur délibérée qui vous permet d'apprendre à déboguer votre script.

### 4 Mettre à jour l'action Amazon Web Services

Importez le script Python mis à jour, puis exécutez à nouveau l'action.

## Créer un script Python initial

Sur votre machine locale, créez votre script Python et compressez le script et une bibliothèque boto3 sous la forme d'un dossier ZIP.

### Conditions préalables

- Téléchargez et installez Python 3. Reportez-vous à la [page Téléchargements Python](#).
- Téléchargez et installez Visual Studio Code. Reportez-vous à la [page Téléchargement de Visual Studio Code](#).
- Vérifiez que vous avez installé l'extension Python pour Visual Studio Code. Reportez-vous à [Visual Studio Marketplace](#).

### Procédure

- 1 Sur votre machine locale, créez un dossier `vro-python-aws`, puis installez-y le SDK boto3 Python.

```
mkdir vro-python-aws
cd vro-python-aws
mkdir lib
pip install boto3 -t lib/
```

- 2 Ouvrez un éditeur et créez le script Python principal. Pour ce cas d'utilisation, vous utilisez Visual Studio Code.

```
import boto3

def handler(context, inputs):
    ec2 = boto3.resource('ec2')
    filters = [{
        'Name': 'instance-state-name',
        'Values': ['running']
    }]

    instances = ec2.instances.filter(Filters=filters)
    for instance in instances:
        print('Instance: ' + instance.id)
```

Ce script Python répertorie toutes les instances d'EC2 en cours d'exécution dans une région donnée.

- 3 Enregistrez le script créé sous la forme d'un fichier `main.py` dans le dossier `vro-python-aws`.
- 4 Connectez-vous à une interface de ligne de commande.
- 5 Accédez au dossier `vro-python-aws`.

```
cd vro-python-aws
```

- 6 Créez un module ZIP qui contient le script Python.

```
zip -r --exclude=*.zip -X vro-python-aws.zip .
```

**Note** Vous pouvez également créer le module ZIP à l'aide d'un outil d'utilitaire ZIP, tel que 7-Zip.

## Résultats

Vous avez créé le script Python de base et vous l'avez préparé pour importation dans votre déploiement de vRealize Orchestrator.

## Créer l'action Amazon Web Services (AWS)

Créez une action vRealize Orchestrator qui utilise votre script Python.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Actions**.
- 3 Cliquez sur **Nouvelle action**.
- 4 Dans l'onglet **Général** entrez un nom, un module et un numéro de version pour l'action.



- 5 Dans l'onglet **Script**, sélectionnez **Python 3.7** comme environnement d'exécution, et **Zip** comme type de script.
- 6 Cliquez sur **Importer**.
- 7 Accédez au dossier `vro-python-aws`, puis sélectionnez le module ZIP contenant le script Python.
- 8 Dans la zone de texte **Gestionnaire d'entrée**, entrez `main.handler`.

---

**Note** Le gestionnaire d'entrée de l'action est basé sur le script principal du module ZIP importé. Étant donné que le script principal se trouve dans un fichier nommé `main.py` et une fonction nommée `handler`, le gestionnaire d'entrée doit être `main.handler`. Si vous avez nommé différemment votre fichier de script principal, modifiez la valeur du gestionnaire d'entrée en conséquence.

---

- 9 Enregistrez l'action et cliquez sur **Exécuter**.

L'exécution de l'action rencontre une erreur.

- 10 Sélectionnez l'onglet **Journaux**.

Les journaux de l'exécution de l'action affichent le message d'erreur  
« `botocore.exceptions.NoRegionError: You must specify a region.` ». Ce comportement est normal, car le script Python initial ne définit pas de région.

#### Étape suivante

Débuguez le script Python. Reportez-vous à [Déboguer l'action Amazon Web Services](#).

## Débuguer l'action Amazon Web Services

La version d'origine du script Python comporte une erreur délibérée qui vous permet d'apprendre à déboguer votre script.

#### Conditions préalables

Connectez-vous à votre compte AWS (Amazon Web Services) et créez un utilisateur IAM spécifiquement pour ce scénario de cas d'utilisation. Reportez-vous à la section [Création d'un utilisateur IAM dans votre compte AWS](#). L'utilisateur IAM doit disposer des autorisations suivantes :

```
"Effect": "Allow",
"Action": "ec2:DescribeInstances",
"Resource": ""
```

## Procédure

### 1 Préparez le dispositif vRealize Orchestrator Appliance.

---

**Attention** Ne déboguez pas les scripts dans votre déploiement vRealize Orchestrator de production. Déboguez à partir d'un déploiement vRealize Orchestrator à nœud unique, que vous utilisez pour le développement et le test.

---

- a Connectez-vous à l'interface de ligne de commande du dispositif vRealize Orchestrator Appliance en tant qu'utilisateur **racine**.
- b Exécutez la commande `vracli dev tools`.
- c Un message vous demande de confirmer que vous souhaitez continuer. Entrez **oui** pour continuer ou **non** pour arrêter.

---

**Important** En exécutant la commande `vracli dev tools`, vous ouvrez les ports requis pour déboguer le script Python. Vous devez laisser la session SSH actuelle ouverte pendant le processus de débogage.

---

### 2 Démarrez la configuration de débogage.

- a Connectez-vous au client vRealize Orchestrator.
- b Ouvrez l'action AWS, puis cliquez sur **Débogage**.  
Le processus de débogage commence et l'exécution de l'action est interrompue.
- c Sélectionnez l'onglet **Configuration de débogage**.  
L'onglet contient une configuration `.json` que vous pouvez attacher à distance à votre IDE pour déboguer le script Python.
- d Copiez manuellement le contenu de la configuration ou cliquez sur **Copier dans le Presse-papiers**.

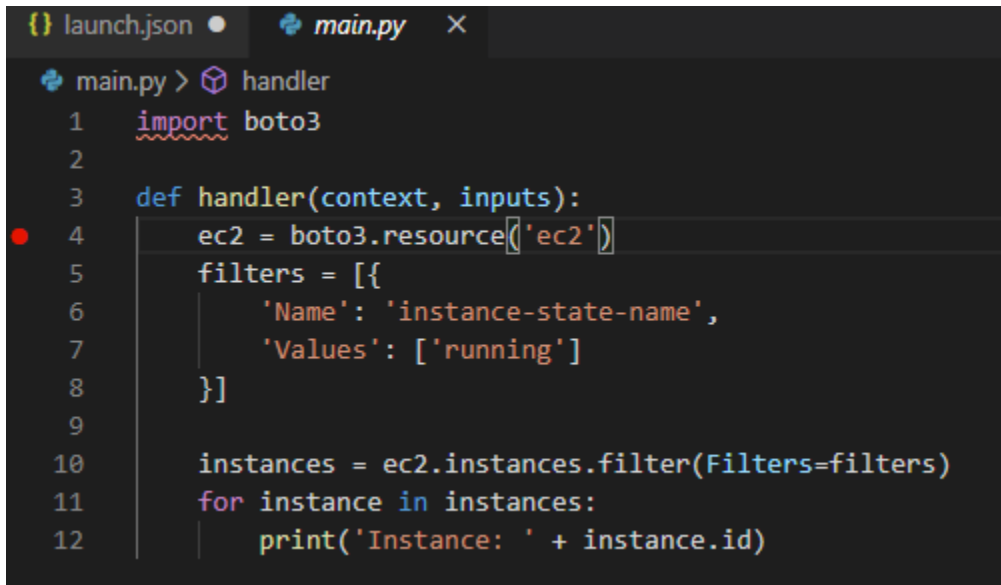
### 3 Déboguez le script Python.

- a Ouvrez Visual Studio Code.
- b Ouvrez le dossier `vro-python-aws`.
- c Dans le volet de navigation supérieur, sélectionnez **Exécuter > Configurations ouvertes**.
- d Sélectionnez **Fichier Python**.

- e Laissez les attributs "version" et "configuration" à leurs positions actuelles, et collez le contenu de la configuration .json copiée à partir de vRealize Orchestrator Client. Le fichier launch.json généré doit ressembler à cela :

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "request": "attach",
      "port": 18281,
      "name": "vRO Python debug 8302f4c7-5beb-40da-848a-5003c0296f7b",
      "host": "es-sof-vc-vm-225-190.sof-mbu.eng.vmware.com",
      "type": "python",
      "pathMappings": [
        {
          "localRoot": "${workspaceFolder}",
          "remoteRoot": "/var/run/vco-polyglot/function"
        }
      ]
    }
  ]
}
```

- f Sélectionnez le fichier de script main.py et ajoutez un point de rupture à la ligne `ec2 = boto3.resource('ec2')`.



```
{ } launch.json  main.py X
main.py > handler
1  import boto3
2
3  def handler(context, inputs):
4      ec2 = boto3.resource('ec2')
5      filters = [{
6          'Name': 'instance-state-name',
7          'Values': ['running']
8      }]
9
10     instances = ec2.instances.filter(Filters=filters)
11     for instance in instances:
12         print('Instance: ' + instance.id)
```

- g Dans le volet de navigation supérieur, sélectionnez **Exécuter > Démarrer le débogage**.

- h Lorsque le débogueur atteint le point de rupture, effectuez une opération de passage.

L'exécution de débogage indique que le script Python ne dispose pas d'une région spécifiée et d'une clé d'accès AWS.

- i Revenez à la session vRealize Orchestrator Appliance ouverte et appuyez sur **Entrée** pour fermer les ports que vous avez ouverts pour cette session de débogage.

#### 4 Ajoutez les informations manquantes au script Python.

- a Dans Visual Studio Code, créez un fichier appelé `awsconfig` contenant la clé d'accès AWS de l'utilisateur IAM et la région AWS pour laquelle vous souhaitez effectuer un test ping avec le script Python.

```
[default]
aws_access_key_id=your key ID
aws_secret_access_key=your secret access key
region=your-region
```

- b Enregistrez `awsconfig` sous la forme d'un fichier de configuration (`.cfg`) dans le dossier `vro-python-aws`.
- c Ouvrez le fichier `main.py` et modifiez-le pour que la bibliothèque `boto3` puisse utiliser le fichier `awsconfig.cfg`.

```
import boto3

import os
os.environ['AWS_CONFIG_FILE'] = os.getcwd() + '/awsconfig.cfg'

def handler(context, inputs):
    ec2 = boto3.resource('ec2')
    filters = [{
        'Name': 'instance-state-name',
        'Values': ['running']
    }]

    instances = ec2.instances.filter(Filters=filters)
    for instance in instances:
        print('Instance: ' + instance.id)
```

- d Créez un nouveau module ZIP contenant le fichier `main.py`, le fichier `awsconfig.cfg` et la bibliothèque `boto3`.

```
zip -r --exclude=*.zip -X vro-python-aws.zip .
```

---

**Note** Vous pouvez également créer le module ZIP à l'aide d'un outil d'utilitaire ZIP, tel que 7-Zip.

---

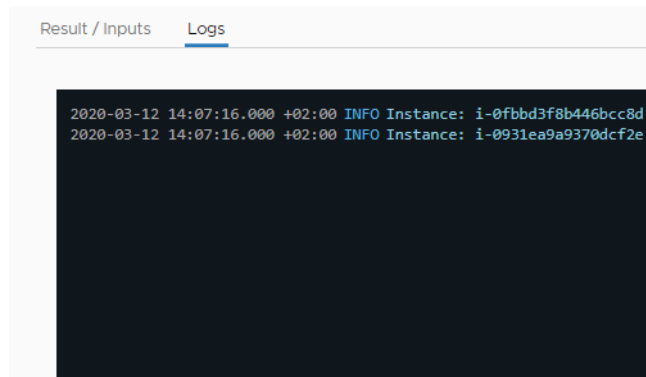
## Mettre à jour l'action Amazon Web Services

Importez le script Python mis à jour, puis exécutez à nouveau l'action.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Actions** et sélectionnez l'action AWS (Amazon Web Services) d'origine.
- 3 (Facultatif) Dans l'onglet **Général**, modifiez le numéro de version.
- 4 Supprimez l'ancien module ZIP, puis cliquez sur **Importer**.
- 5 Sélectionnez le module ZIP mis à jour.
- 6 Enregistrez l'action et cliquez sur **Exécuter**.
- 7 Une fois l'exécution de l'action terminée, sélectionnez l'onglet **Journaux**.

Les journaux affichent les instances d'EC2 interrogées par



l'action.

### Étape suivante

Créez un workflow vRealize Orchestrator qui utilise l'action AWS mise à jour en tant qu'**Élément d'action**.

## Comment puis-je utiliser le branchement Git pour gérer mon inventaire d'objets vRealize Orchestrator

Utilisez le branchement pour organiser la manière dont le contenu vRealize Orchestrator est géré dans votre référentiel Git.

À l'aide de Git, vous pouvez augmenter la flexibilité de vos développeurs vRealize Orchestrator en fournissant un référentiel centralisé. Par exemple, vous pouvez utiliser Git pour gérer le développement de workflows dans plusieurs environnements vRealize Orchestrator.

---

**Note** Pour utiliser Git pour gérer votre inventaire d'objets, votre déploiement de vRealize Orchestrator doit utiliser une licence vRealize Automation. Pour plus d'informations, reportez-vous à *Activation des fonctionnalités vRealize Orchestrator avec des licences* dans *Installation et configuration de vRealize Orchestrator*.

---

Vous pouvez désormais transférer et extraire des objets vers et depuis des branches. Vous pouvez utiliser le branchement pour gérer le développement de groupes spécifiques d'objets vRealize Orchestrator, avant qu'ils ne soient fusionnés dans votre branche principale.

Dans ce cas d'utilisation, vous utilisez un projet GitLab pour gérer des objets vRealize Orchestrator qui utilisent l'environnement d'exécution Python. Ce cas d'utilisation représente un exemple de la fonctionnalité Git dans vRealize Orchestrator et ne représente pas les limites de l'étendue de la fonctionnalité.

---

**Note** Si vous connaissez bien GitHub, vous pouvez utiliser un référentiel GitHub pour ce cas d'utilisation.

---

## Procédure

### 1 Préparer votre environnement GitLab

Créez une branche Git pour vos objets vRealize Orchestrator Python.

### 2 Configurer une connexion à un référentiel Git

En tant qu'**administrateur**, vous pouvez configurer une connexion entre votre déploiement vRealize Orchestrator et un référentiel ou un projet Git.

### 3 Transférer les modifications vers un référentiel Git

Transférez les modifications apportées aux objets vRealize Orchestrator locaux vers votre référentiel Git intégré. Pour ce cas d'utilisation, nous transférons les modifications à une action vRealize Orchestrator basée sur Python vers une branche Git spécifique.

## Préparer votre environnement GitLab

Créez une branche Git pour vos objets vRealize Orchestrator Python.

### Conditions préalables

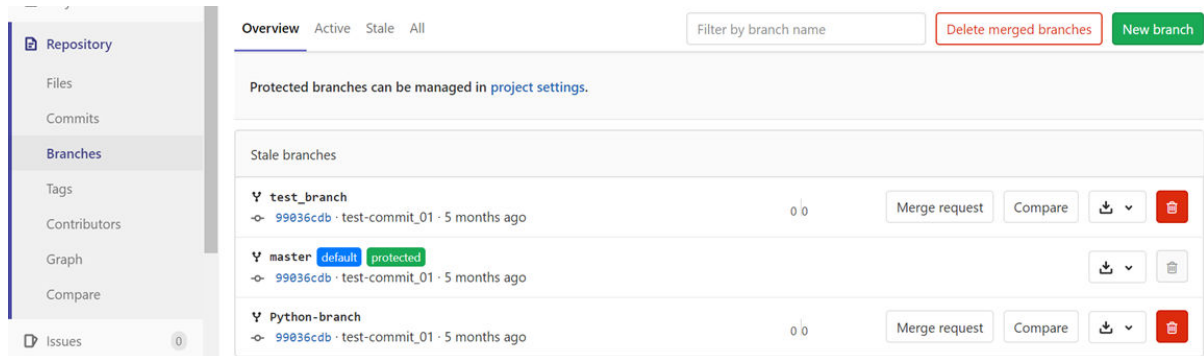
Créez un projet GitLab pour votre environnement vRealize Orchestrator. Reportez-vous à la section [Créer un projet](#).

### Procédure

- 1 Connectez-vous à votre compte GitLab.
- 2 Accédez à votre projet GitLab.

- 3 Dans le volet de navigation de gauche, sélectionnez **Référentiel > Branches**.
- 4 Dans l'onglet **Présentation**, cliquez sur **Nouvelle branche**.
- 5 Sous **Nom de la branche**, entrez **Python-branch**.
- 6 Pour l'option **Créer depuis**, conservez **master**.
- 7 Cliquez sur **Créer une branche**.

Vous avez créé une branche pour vos objets vRealize Orchestrator basés sur Python.



## Configurer une connexion à un référentiel Git

En tant qu'**administrateur**, vous pouvez configurer une connexion entre votre déploiement vRealize Orchestrator et un référentiel ou un projet Git.

Pour utiliser Git pour la gestion de votre inventaire d'objets vRealize Orchestrator, vous devez configurer la connexion à votre référentiel Git à l'aide de vRealize Orchestrator Client.

**Note** Vous ne pouvez pas ajouter plusieurs référentiels Git à partir de différents comptes via SSH, car vRealize Orchestrator crée une clé SSH pour chaque instance. Pour ajouter plusieurs référentiels Git, vous pouvez le faire via HTTP, comme décrit dans cette documentation.

### Conditions préalables

- Vérifiez que votre environnement vRealize Orchestrator utilise une licence vRealize Automation.
- Générez un jeton d'accès pour votre projet GitLab et copiez-le dans le Presse-papiers afin de pouvoir l'utiliser pendant le processus de configuration. Reportez-vous à [Création d'un jeton d'accès personnel](#).

**Note** Pour ce cas d'utilisation, vous utilisez un projet GitLab. Si vous êtes familiarisé avec GitHub, vous pouvez utiliser un référentiel GitHub. Pour plus d'informations sur la génération d'un jeton GitHub, reportez-vous à [Création d'un jeton d'accès personnel pour la ligne de commande](#).

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client en tant qu'**administrateur**.

2 Accédez à **Administration > Référentiels Git**.

3 Cliquez sur **Ajouter un référentiel**.

4 Entrez l'adresse URL de votre référentiel Git.

Par exemple, <https://gitlab.com/myusername/my-vro-repo>.

---

**Note** Vous pouvez également établir une connexion avec le protocole SSH.

---

5 Entrez le nom d'utilisateur de votre profil Git.

6 Entrez le jeton d'accès de votre référentiel Git.

7 Pour valider la connexion au référentiel Git, cliquez sur **Valider**.

8 (Facultatif) Modifiez le nom utilisé pour identifier le référentiel dans le vRealize Orchestrator Client.

9 (Facultatif) Ajoutez une brève description du référentiel Git connecté.

10 Pour activer le référentiel Git connecté, cliquez sur **Créer un référentiel actif**.

---

**Note** Vous ne pouvez utiliser qu'un seul référentiel Git à la fois. Vous pouvez modifier le référentiel Git actif sur la page **Référentiels Git**.

---

11 Sélectionnez la branche à laquelle vous souhaitez transférer vos modifications. Pour ce cas d'utilisation, vous utilisez **Python-branch**. Reportez-vous à [Préparer votre environnement GitLab](#).

---

**Note** Vous pouvez modifier la branche Git sélectionnée à tout moment après avoir terminé la configuration Git initiale.

---

12 Cliquez sur **Enregistrer** pour terminer le processus de configuration.

#### Étape suivante

Revenez au menu **Référentiels Git** et vérifiez que l'état du référentiel est **Actif**.

## Transférer les modifications vers un référentiel Git

Transférez les modifications apportées aux objets vRealize Orchestrator locaux vers votre référentiel Git intégré. Pour ce cas d'utilisation, nous transférons les modifications à une action vRealize Orchestrator basée sur Python vers une branche Git spécifique.

Vous pouvez transférer un ensemble de modifications locales vers un référentiel Git. Chaque ensemble de modifications peut se composer d'un ou de plusieurs objets vRealize Orchestrator modifiés.

---

**Note** Le processus de transfert et d'annulation des ensembles de modifications vers un référentiel Git n'est pas limité par les autorisations de groupe. Par conséquent, un développeur de workflow d'un groupe peut transférer ou annuler des modifications locales effectuées par un autre développeur.

---



## Conditions préalables

- Vérifiez que vous avez créé une branche Git. Reportez-vous à [Préparer votre environnement GitLab](#).
- Vérifiez que vous avez configuré une connexion avec un référentiel Git. Reportez-vous à [Configurer une connexion à un référentiel Git](#).
- Vérifiez que votre intégration Git est configurée pour transférer les modifications apportées à la branche Git **Python-Branch**.
- Créez un objet vRealize Orchestrator basé sur Python. Par exemple, reportez-vous à la section [Comment intégrer des services Amazon Web Services dans vRealize Orchestrator à l'aide de Python](#).

## Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Modifiez votre action Python.
  - a Accédez à **Bibliothèque > Actions**, puis sélectionnez votre action Python.
  - b Apportez des modifications mineures à l'action, telles que la modification de la description.
  - c Enregistrez l'action.

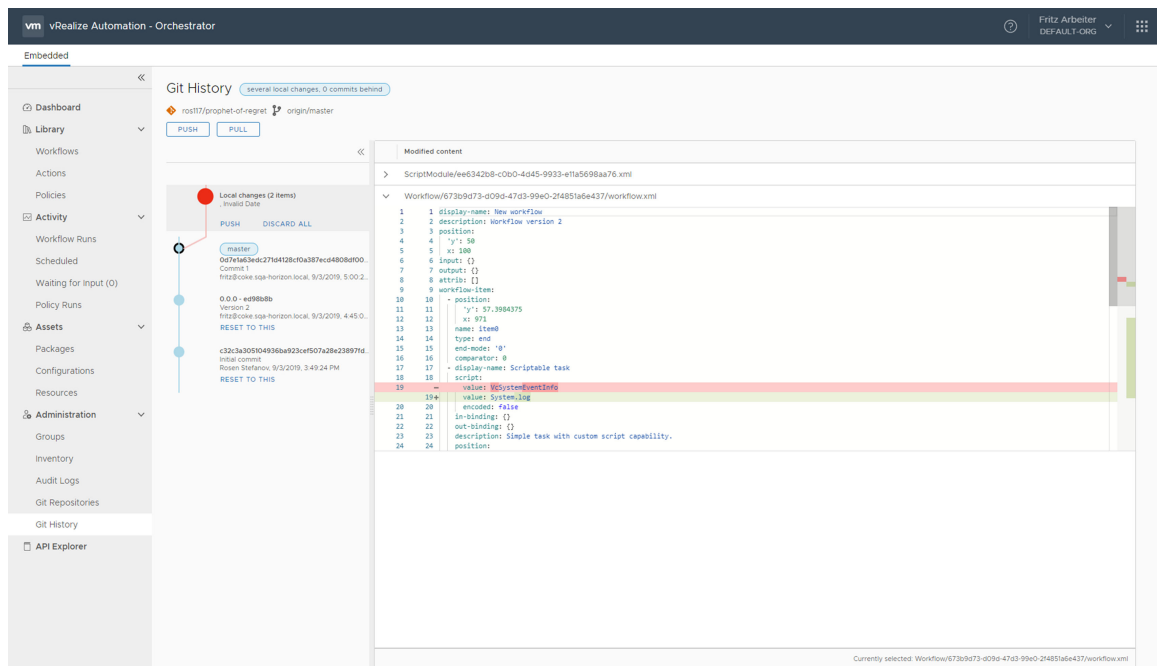
### 3 Envoyez vos modifications au référentiel Git.

**Note** Vous pouvez également transférer les modifications locales au niveau de chaque objet en cliquant sur l'option **Version** affichée en bas de l'éditeur d'objets.

#### a Accédez à **Administration > Historique Git**.

**Historique Git** affiche les différences actuelles entre la branche de version locale et la branche du référentiel Git sélectionnée. Vous pouvez développer l'entrée pour tout objet vRealize Orchestrator modifié afin d'afficher les différences de version.

**Note** Vous pouvez ignorer un ensemble de modifications locales en sélectionnant **Tout ignorer**.



#### b Cliquez sur **Transférer**.

#### c Entrez le titre du commit.

#### d (Facultatif) Entrez une brève description du commit.

#### e Sélectionnez les modifications apportées à votre action Python que vous souhaitez transférer vers le référentiel Git.

### 4 Pour terminer le transfert de l'ensemble des modifications locales vers le référentiel Git, cliquez sur **Transférer**.

#### Étape suivante

Une fois que vous avez terminé le développement de votre branche Git, fusionnez-la à la branche principale. Reportez-vous à [Comment créer une demande de fusion](#).

# Comment utiliser des modules tiers pour appeler vRealize Automation Project API

Ce cas d'utilisation vRealize Orchestrator illustre comment appeler vRealize Automation Project API à l'aide de modules tiers.

Vous pouvez utiliser les environnements d'exécution suivants dans vos scripts d'action et de workflow :

- Python 3.7
- Node.js 14
- PowerCLI 11/Powershell 6.2
- PowerCLI 12.3.0/Powershell 7.1

---

**Note** L'exécution de PowerCLI inclut PowerShell et les modules suivants : VMware.PowerCLI, PowerNSX, PowervRA.

---

Dans ce cas d'utilisation, vous apprenez à créer des actions vRealize Orchestrator qui utilisent des modules de dépendance tiers pour se connecter à vRealize Automation Project API.

---

**Important** Avant de commencer à développer votre script personnalisé, vérifiez que vous maîtrisez bien les concepts principaux d'utilisation des scripts Python, Node.js et PowerShell dans vRealize Orchestrator. Reportez-vous à [Concepts de base pour les scripts Python, Node.js et PowerShell](#) .

---

## Créer un script Python qui appelle vRealize Automation Project API

Créez un exemple de script qui utilise Python pour appeler vRealize Automation Project API.

### Conditions préalables

Vérifiez que vous avez installé Python 3 et le programme d'installation du module PIP. Consultez [Page de téléchargements Python](#) et [Index de module Python](#).

### Procédure

- 1 Sur votre machine locale, ouvrez un interpréteur de ligne de commande.
- 2 Créez un dossier `vro-python-vra`.

```
mkdir vro-python-vra
```

- 3 Accédez au dossier `vro-python-vra`.

```
cd vro-python-vra
```

- 4 Créez un script Python appelé `handler.py`.

```
touch handler.py
```

Le script `handler.py` doit définir une fonction qui accepte deux arguments, le contexte de l'exécution du workflow vRealize Orchestrator et les entrées vRealize Orchestrator liées.

```
def handler(context, inputs):
    print('Hello, your inputs were ' + inputs)
    return None
```

**Note** Si vous utilisez des bibliothèques de journalisation standard, tout ce que vous connectez à l'action qui utilise votre script est également indiqué dans le journal du workflow. Les entrées et le retour de votre script doivent avoir des paramètres d'entrée et des types de retour correspondants configurés dans le vRealize Orchestrator Client. Par exemple, l'entrée `vRAUrl` dans votre script doit avoir un paramètre d'entrée correspondant appelé `vRAUrl` dans vRealize Orchestrator Client. De même, si votre script renvoie une valeur de chaîne, le type de retour configuré dans vRealize Orchestrator Client doit également être un type de chaîne. Si votre action renvoie un objet complexe, vous pouvez utiliser le type de retour `Properties` ou `Composite Type`.

## 5 Installez le module de demandes Python.

**Important** Les modules de dépendance tiers doivent être installés dans un dossier de niveau racine dans votre dossier de scripts `vro-python-vra` principal. Pour ce cas d'utilisation, vous créez un dossier `lib` pour votre module de demandes.

### a Créez un dossier `lib`.

```
mkdir lib
```

### b Installez le module de demandes.

```
pip3 install requests -t lib/
```

## 6 Ajoutez le module de demandes au script `handler.py`.

```
import requests

def handler(context, inputs):
    print('Hello, your inputs were ' + inputs)
    return None
```

## 7 Créez une demande GET pour vRealize Automation Project API.

```
token = ''
vRAUrl = ''
r = requests.get(vRAUrl + '/iaas/api/projects', headers={'Authorization': 'Bearer ' + token})

print('Got response ' + r.text)
```

**8** Définissez les valeurs `token` et `vRAUrl` .

- a Récupérez le jeton d'accès à l'aide de vRealize Automation Identity Service API. Consultez [Obtenir votre jeton d'accès pour vRealize Automation API](#)
- b Pour la valeur `vRAUrl`, définissez le script pour qu'il utilise un paramètre d'entrée vRealize Orchestrator portant le même nom.

```
vRAUrl = inputs["vRAUrl"]
```

- c Ajoutez les nouvelles valeurs au fichier `handler.py`.

```
import requests

def handler(context, inputs):
    token = 'ACCESS_TOKEN'
    vRAUrl = inputs["vRAUrl"]

    r = requests.get(vRAUrl + '/iaas/api/projects', headers={'Authorization': 'Bearer ' + token})

    print('Got response ' + r.text)

    return r.json()
```

**Note** Comme la réponse de vRealize Automation Project API est renvoyée au format JSON, utilisez un type de retour `Properties` ou `Composite Type` pour votre action vRealize Orchestrator.

- 9 Créez un module ZIP qui contient le fichier `handler.py` et le dossier `lib` de votre module de demande.

```
zip -r --exclude=*.zip -X vro-python-vra.zip .
```

**Étape suivante**

Importez le script PowerShell dans une action de vRealize Orchestrator. Reportez-vous à [Créer des actions dans vRealize Orchestrator Client](#).

**Créer un script Node.js qui appelle vRealize Automation Project API**

Créez un exemple de script qui utilise Node.js pour appeler vRealize Automation Project API.

**Conditions préalables**

Téléchargez et installez Node.js 14. Consultez la page [Téléchargements de Node.js](#).

**Procédure**

- 1 Sur votre machine locale, ouvrez un interpréteur de ligne de commande.

**2** Créez un dossier `vro-node-vra`.

```
mkdir vro-node-vra
```

**3** Accédez au dossier `vro-node-vra`.

```
cd vro-node-vra
```

**4** Créez un script Node.js appelé `handler.js`.

```
touch handler.js
```

Le script `handler.js` doit définir une fonction qui accepte deux arguments, le contexte de l'exécution du workflow vRealize Orchestrator et les entrées vRealize Orchestrator liées.

```
exports.handler = (context, inputs) => {
  console.log('Hello, your inputs were ' + inputs);
  return null;
}
```

**Note** Si vous utilisez des bibliothèques de journalisation standard, tout ce que vous connectez à l'action qui utilise votre script est également indiqué dans le journal du workflow. Les entrées et le retour de votre script doivent avoir des paramètres d'entrée et des types de retour correspondants configurés dans le vRealize Orchestrator Client. Par exemple, l'entrée `vRAUrl` dans votre script doit avoir un paramètre d'entrée correspondant appelé `vRAUrl` dans vRealize Orchestrator Client. De même, si votre script renvoie une valeur de chaîne, le type de retour configuré dans vRealize Orchestrator Client doit également être un type de chaîne. Si votre action renvoie un objet complexe, vous pouvez utiliser le type de retour `Properties` ou `Composite Type`.

**5** Installez le module de demandes Node.js.

```
npm install request
```

**Important** Les modules de dépendance tiers doivent être installés dans le dossier de racine `node_modules` de votre dossier de scripts `vro-node-vra` principal. Ne déplacez pas ou ne renommez pas ce dossier.

**6** Ajoutez le module demandes au script de `handler.js`.

```
const request = require('request');

exports.handler = (context, inputs) => {
  console.log('Hello, your inputs were ' + inputs);
  return null;
}
```

## 7 Créez une demande GET pour vRealize Automation Project API.

```
const token = '';
const vRAUrl = '';
request.get(vRAUrl + '/iaas/api/projects', { 'auth': { 'bearer': token } }, function
(error, response, body) {
    console.log('Got response ' + body);
});
```

## 8 Définissez les valeurs token et vRAUrl .

- Récupérez le jeton d'accès à l'aide de vRealize Automation Identity Service API. Consultez [Obtenir votre jeton d'accès pour vRealize Automation API](#).
- Pour la valeur vRAUrl, définissez le script pour qu'il utilise un paramètre d'entrée vRealize Orchestrator portant le même nom.

```
const vRAUrl = inputs.vRAUrl;
```

- Ajoutez les nouvelles valeurs au fichier handler.js.

```
const request = require('request');
exports.handler = (context, inputs, callback) => {
    const vRAUrl = inputs.vRAUrl;
    const token = 'ACCESS_TOKEN';
    request.get(vRAUrl + '/iaas/api/projects', { 'auth': { 'bearer': token } },
function (error, response, body) {
    console.log('Got response ' + body);
    callback(null, JSON.parse(body));
});
}
```

**Note** Comme la réponse de vRealize Automation Project API est renvoyée au format JSON, utilisez un type de retour `Properties` ou `Composite Type` pour votre action vRealize Orchestrator.

## 9 Créez un module ZIP qui contient le fichier handler.js et le dossier node\_modules de votre module de demande.

```
zip -r --exclude=*.zip -X vro-node-vra.zip .
```

### Étape suivante

Importez le script Node.js dans une action vRealize Orchestrator. Reportez-vous à [Créer des actions dans vRealize Orchestrator Client](#).

## Créer un script PowerShell qui appelle vRealize Automation Project API

Créez un exemple de script qui utilise PowerShell pour appeler vRealize Automation Project API.

## Procédure

- 1 Sur votre machine locale, ouvrez un interpréteur de ligne de commande.
- 2 Créez un dossier `vro-powershell-vra`.

```
mkdir vro-powershell-vra
```

- 3 Accédez au dossier `vro-powershell-vra`.

```
cd vro-powershell-vra
```

- 4 Créez un script PowerShell appelé `handler.ps1`.

```
touch handler.ps1
```

Le script `handler.ps1` doit définir une fonction qui accepte deux arguments, le contexte de l'exécution du workflow vRealize Orchestrator et les entrées vRealize Orchestrator liées.

```
function Handler {  
    Param($context, $inputs)  
  
    $inputsString = $inputs | ConvertTo-Json -Compress  
    Write-Host "Inputs were $inputsString"  
}
```

---

**Note** Si vous utilisez des bibliothèques de journalisation standard, tout ce que vous connectez à l'action qui utilise votre script est également indiqué dans le journal du workflow. Les entrées et le retour de votre script doivent avoir des paramètres d'entrée et des types de retour correspondants configurés dans le vRealize Orchestrator Client. Par exemple, l'entrée `vRAUrl` dans votre script doit avoir un paramètre d'entrée correspondant appelé `vRAUrl` dans vRealize Orchestrator Client. De même, si votre script renvoie une valeur de chaîne, le type de retour configuré dans vRealize Orchestrator Client doit également être un type de chaîne. Si votre action renvoie un objet complexe, vous pouvez utiliser le type de retour `Properties` ou `Composite Type`.

---



## 5 Installez le module d'assertion PowerShell.

**Important** Les modules de dépendance tiers doivent être installés dans un dossier de niveau racine dans votre dossier de scripts `vro-powershell-vra` principal. Pour ce cas d'utilisation, vous créez un dossier `Modules` pour votre module d'assertion.

- a Créez un dossier `Modules`.

```
mkdir Modules
```

- b Installez le module d'assertion.

```
pwsh -c "Save-Module -Name Assert -Path ./Modules/ -Repository PSGallery"
```

## 6 Ajoutez le module d'assertion au script `handler.ps1`.

```
Import-Module Assert

function Handler {
    Param($context, $inputs)

    $inputsString = $inputs | ConvertTo-Json -Compress
    Write-Host "Inputs were $inputsString"
}
```

## 7 Créez une demande GET à vRealize Automation Project API qui utilise l'applet de commande `Invoke-RestMethod`.

```
$token = ''
$vRAUrl = ''
$projectsUrl = $vRAUrl + "/project-service/api/projects"
$response = Invoke-RestMethod $projectsUrl + '/iaas/api/projects' -Headers
@{'Authorization' = "Bearer $token"} -Method 'GET'

Write-Host "Got response: $response"
```

## 8 Définissez les valeurs `token` et `vRAUrl`.

- a Récupérez le jeton d'accès à l'aide de vRealize Automation Identity Service API. Consultez [Obtenir votre jeton d'accès pour vRealize Automation API](#).
- b Ajoutez les attributs du module d'assertion `Assert-NotNull` et `Assert-Type`.

```
$token | Assert-NotNull
$token | Assert-Type String
```

- c Pour la valeur `vRAUrl`, définissez le script pour qu'il utilise un paramètre d'entrée vRealize Orchestrator portant le même nom.

```
$vRAUrl = $inputs.vRAUrl
```

- d Ajoutez les nouvelles valeurs au fichier `handler.ps1`.

```
Import-Module Assert
$ErrorActionPreference = "Stop"
function Handler {
    Param($context, $inputs)
    $token = "ACCESS_TOKEN"
    $token | Assert-NotNull
    $token | Assert-Type String
    $vRAUrl = $inputs.vRAUrl
    $projectsUrl = $vRAUrl + "/project-service/api/projects"
    $response = Invoke-RestMethod $projectsUrl -Headers @{'Authorization' = "Bearer $token"} -Method 'GET'

    Write-Host "Got response: $response"

    return $response
}
```

**Note** Comme la réponse de vRealize Automation Project API est renvoyée au format JSON, utilisez un type de retour `Properties` ou `Composite Type` pour votre action vRealize Orchestrator.

- 9 Créez un module ZIP qui contient le fichier `handler.ps1` et le dossier `Modules` de votre module d'assertion.

```
zip -r --exclude=*.zip -X vro-powershell-vra.zip .
```

### Étape suivante

Importez le script PowerShell dans une action de vRealize Orchestrator. Reportez-vous à [Créer des actions dans vRealize Orchestrator Client](#).

# Gestion des workflows

# 5

Un workflow est une série d'actions et de décisions que l'on exécute de façon séquentielle. vRealize Orchestrator fournit une bibliothèque de workflows qui effectuent des tâches de gestion courantes. vRealize Orchestrator fournit également des bibliothèques des actions individuelles réalisées par les workflows.

Les workflows associent actions, décisions et résultats qui, s'ils sont exécutés dans un ordre précis, réalisent une tâche ou un processus particulier dans un environnement virtuel. Les workflows réalisent des tâches telles que le provisionnement de machines virtuelles, la sauvegarde, la maintenance régulière, l'envoi d'e-mails, les opérations SSH, la gestion de l'infrastructure physique et d'autres opérations utilitaires générales. Ils acceptent en entrée ce qui correspond à leur fonction. Vous pouvez créer des workflows qui s'exécutent selon des planifications définies ou si certains événements prévus surviennent. Les informations peuvent être fournies par vous, par d'autres utilisateurs ou par un processus externe, comme l'appel à un service Web par une application. Le workflow effectue une validation et un filtrage des informations avant qu'elles s'exécutent.

Les workflows peuvent faire appel à d'autres workflows. Par exemple, vous pouvez disposer d'un workflow qui appelle un autre workflow pour créer une machine virtuelle.

Créez des workflows à l'aide de l'IDE (Integrated Drive Electronics) de l'interface de vRealize Orchestrator Client, qui donne accès à la bibliothèque de workflows et la possibilité d'exécuter des workflows sur le moteur de workflow. Le moteur de workflow peut aussi se servir d'objets issus de bibliothèques externes à connecter sur vRealize Orchestrator. Cette fonctionnalité vous permet de personnaliser les processus ou de mettre en œuvre les fonctions que des applications tierces fournissent.

Ce chapitre contient les rubriques suivantes :

- [Workflows standard de la bibliothèque de workflows vRealize Orchestrator](#)
- [Créer des workflows dans vRealize Orchestrator Client](#)
- [Modifier les workflows et les actions du workflow parent](#)
- [Concepteur de formulaires de saisie vRealize Orchestrator](#)
- [Demandes d'interaction des utilisateurs dans le client vRealize Orchestrator](#)
- [Planifier les workflows dans le client vRealize Orchestrator](#)
- [Rechercher des références d'objets dans les workflows](#)

## Workflows standard de la bibliothèque de workflows vRealize Orchestrator

vRealize Orchestrator fournit une bibliothèque standard de workflows que vous pouvez utiliser pour automatiser les opérations de votre infrastructure virtuelle. Les workflows de la bibliothèque standard sont verrouillés en lecture seule. Pour personnaliser un workflow standard, vous devez le dupliquer. Les workflows en double et les workflows personnalisés que vous créez sont entièrement modifiables.

Le contenu de la bibliothèque de workflows est accessible via le menu **Bibliothèque > Workflows** de vRealize Orchestrator Client basé sur HTML5. Les workflows standard et personnalisés du client sont organisés à l'aide de balises. Par exemple, vous pouvez accéder au workflow **Générer une paire de clés** en entrant **SSH** dans la zone de recherche de la bibliothèque de workflows.

---

**Note** Vous ne pouvez pas ajouter de nouvelles balises aux workflows standard, sauf si vous dupliquez le workflow.

---

## Créer des workflows dans vRealize Orchestrator Client

Vous pouvez utiliser vRealize Orchestrator Client pour créer et modifier des workflows.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Sélectionnez **Bibliothèque > Workflows**.
- 3 Cliquez sur **Nouveau workflow**.
- 4 Entrez le nom du nouveau workflow et cliquez sur **Créer**.
- 5 Utilisez l'éditeur de workflow pour configurer les variables, les entrées et les sorties de workflow, la structure de schéma et la présentation du workflow.
- 6 Pour terminer la modification du workflow, cliquez sur **Enregistrer**.

---

**Note** Vous pouvez suivre les modifications apportées aux workflows dans l'onglet **Historique des versions**. Pour plus d'informations, reportez-vous à [Historique des versions des objets vRealize Orchestrator](#).

---

### Étape suivante

Vous pouvez utiliser la fonctionnalité de relecture de jeton de vRealize Orchestrator pour optimiser les performances de vos workflows. Pour plus d'informations, reportez-vous à [Utilisation de la relecture de jeton de workflow dans vRealize Orchestrator Client](#).

## Modifier les workflows et les actions du workflow parent

Modifiez les workflows et les actions directement à partir du workflow parent dans l'instance de vRealize Orchestrator Client.

La modification des workflows et des actions enfants directement à partir du workflow parent peut aider à rationaliser le développement du workflow.

### Conditions préalables

Créez un workflow qui appelle un autre workflow, une autre action ou les deux.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Workflows** et sélectionnez votre workflow.
- 3 Sélectionnez l'onglet **Schéma**.
- 4 En fonction du type d'objet, double-cliquez sur **Élément de workflow** ou sur **Élément d'action** dans le canevas du workflow.
- 5 Modifiez l'objet.
- 6 Pour terminer la modification du workflow ou de l'action enfant, cliquez sur **Enregistrer**.
- 7 Pour revenir au Workflow parent, fermez l'éditeur d'objets.

## Concepteur de formulaires de saisie vRealize Orchestrator

Si un workflow nécessite des paramètres d'entrée, il ouvre une boîte de dialogue dans laquelle les utilisateurs saisissent les valeurs requises. Vous pouvez organiser le contenu, la disposition ou la présentation de cette boîte de dialogue dans le concepteur de formulaires de saisie.

Le concepteur de formulaires de saisie se trouve dans l'onglet **Formulaire de saisie** de l'éditeur de workflow. Ce concepteur se compose d'un menu de navigation, d'un canevas de conception et d'un menu de propriétés. Vous pouvez faire glisser des entrées et des éléments génériques du menu de gauche vers le canevas de conception. Dans le canevas, vous pouvez définir la position des paramètres d'entrée, les organiser en onglets de saisie distincts et configurer les propriétés des paramètres d'entrée.

---

**Note** Vous ne pouvez pas utiliser le contenu de l'onglet **Variables** de l'éditeur de workflow dans le concepteur de formulaires de saisie. Vous pouvez uniquement utiliser les paramètres de l'onglet **Entrée/sortie**.

---

### Éléments génériques

Vous pouvez ajouter des éléments génériques tels que des menus déroulants et des zones de texte de mot de passe au concepteur de formulaires de saisie. Les éléments génériques

ne correspondent pas aux paramètres d'entrée réels, mais peuvent être liés à des paramètres d'entrée.

## Créer la boîte de dialogue des paramètres d'entrée de workflow dans vRealize Orchestrator Client

Vous pouvez utiliser le concepteur de formulaires de saisie pour créer et personnaliser la boîte de dialogue de saisie des paramètres du workflow.

### Conditions préalables

Vérifiez que le workflow possède une liste définie de paramètres d'entrée.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Workflows**.
- 3 Sélectionnez votre workflow personnalisé.
- 4 Cliquez sur l'onglet **Formulaire de saisie**.
- 5 (Facultatif) Créez des onglets à utiliser dans la boîte de dialogue de saisie.  
Vous pouvez utiliser les onglets pour organiser la structure de votre boîte de dialogue.
- 6 Sélectionnez vos paramètres d'entrée.
- 7 Modifiez les propriétés des paramètres d'entrée.  
Pour plus d'informations sur les propriétés des paramètres d'entrée, reportez-vous à [Propriétés des paramètres d'entrée dans vRealize Orchestrator Client](#).
- 8 (Facultatif) Ajoutez des éléments génériques au canevas et liez-les aux paramètres d'entrée.
- 9 (Facultatif) Ajoutez une validation externe aux paramètres d'entrée. Pour plus d'informations, reportez-vous à [Utilisation d'actions pour valider les entrées de workflow vRealize Orchestrator](#).
- 10 Cliquez sur **Enregistrer**.

### Résultats

Vous avez créé la mise en forme de la boîte de dialogue du workflow et vous avez défini les propriétés des paramètres d'entrée.

## Propriétés des paramètres d'entrée dans vRealize Orchestrator Client

Vous pouvez définir des propriétés de paramètre pour fixer des contraintes aux paramètres d'entrée que les utilisateurs fournissent lorsqu'ils exécutent des workflows vRealize Orchestrator.

Avec vRealize Orchestrator, vous pouvez définir les propriétés de paramètre utilisées pour quantifier les valeurs des paramètres d'entrée utilisées dans les workflows. Les propriétés de paramètre que vous définissez imposent des limites aux types et aux valeurs des paramètres d'entrée que les utilisateurs fournissent dans les workflows vRealize Orchestrator.

Les propriétés de paramètre valident les paramètres d'entrée et modifient la présentation des zones de texte qui s'affichent dans la boîte de dialogue des paramètres d'entrée. Certaines propriétés de paramètre peuvent créer des dépendances entre les paramètres.

Propriété de paramètre	Description
Étiquette	Permet de définir l'étiquette du paramètre d'entrée.
Type d'affichage	Permet de définir le type d'affichage de la zone de texte.
Visibilité	Permet de définir la visibilité du paramètre d'entrée.
Lecture seule	Permet de définir la zone de texte d'entrée en lecture seule.
Aide personnalisée	Permet de définir la description du signpost du paramètre d'entrée.
Valeur par défaut	Permet de définir la valeur par défaut du paramètre d'entrée.
Étape	Utilisé pour les entrées numériques. Permet de définir le taux d'augmentation de la valeur du paramètre d'entrée par clic.
Obligatoire	Permet de définir si la valeur du paramètre d'entrée est obligatoire ou non.
Expression régulière	Permet de valider l'entrée à l'aide d'une expression régulière.
Valeur minimale	Permet de définir la valeur ou la longueur minimale du paramètre.
Valeur maximale	Permet de définir la valeur ou la longueur maximale du paramètre.
Zone de texte de correspondance	Permet de définir la valeur du paramètre d'entrée pour qu'elle corresponde à celle d'un autre paramètre d'entrée.
Source de la valeur	Permet de définir la source de valeur des propriétés de paramètre dans les onglets <b>Apparence</b> , <b>Valeur</b> et <b>Contraintes</b> .  <b>Note</b> Vous pouvez importer la valeur des actions externes à l'aide de l'option <b>Source externe</b> . Le filtrage des actions disponibles est effectué par type de paramètre.

## Utilisation d'actions pour valider les entrées de workflow vRealize Orchestrator

Utilisez des actions externes pour valider les saisies de vos workflows personnalisés.

## Conditions préalables

Créez un workflow personnalisé avec des paramètres d'entrée. Pour plus d'informations, reportez-vous à [Créer des workflows dans vRealize Orchestrator Client](#).

Vous pouvez utiliser le concepteur de formulaires de saisie pour créer des validations externes pour vos entrées de workflow. Les validations externes utilisent des scripts d'action qui renvoient une valeur de chaîne lorsque la valeur du paramètre de saisie contient une erreur. Si la valeur du paramètre de saisie est valide, la validation externe ne renvoie rien.

## Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Créez une action de validation.
  - a Accédez à **Bibliothèque > Actions**.
  - b Cliquez sur **Nouvelle action**.
  - c Entrez les informations requises dans l'onglet **Résumé**.
  - d Entrez les paramètres d'entrée de l'action de validation.

---

**Note** Les noms des paramètres d'entrée de l'action de validation doivent être identiques aux noms des paramètres d'entrée du workflow en cours de validation.

---

- e Entrez le script de l'action de validation sur l'onglet **Script**.

```
if (in_1=="invalid") {
    return "in_1 can't be invalid!";
}

if (in_2=="invalid") {
    return "in_2 can't be invalid!";
}

//inputs are valid, return nothing
```

---

**Note** Le script précédent est un exemple simple et ne représente pas la portée complète des scripts de validation qui peuvent être utilisés.

---

- f Cliquez sur **Enregistrer**.
- 3 Appliquez la validation externe.
  - a Accédez à **Bibliothèque > Workflows**.
  - b Sélectionnez votre workflow personnalisé.
  - c Sélectionnez l'onglet **Formulaire de saisie**.
  - d Sélectionnez l'icône du Presse-papiers en haut à gauche de l'écran.
  - e Faites glisser un élément de validation de vRealize Orchestrator dans le canevas.



- f Sélectionnez l'élément de validation, entrez une étiquette de validation et sélectionnez l'action de validation.
  - g (Facultatif) Créez des éléments de validation supplémentaires.
  - h Cliquez sur **Enregistrer**.
- 4 Exécutez le workflow.

Si la validation rencontre une erreur, elle renvoie une chaîne. Si la validation réussit, la validation ne renvoie rien et l'exécution du workflow continue.

#### Résultats

Vous avez créé une validation externe de votre workflow vRealize Orchestrator personnalisé.

## Demandes d'interaction des utilisateurs dans le client vRealize Orchestrator

Une entrée utilisateur supplémentaire peut être nécessaire pour terminer les workflows.

Les workflows nécessitant une interaction supplémentaire des utilisateurs interrompent les opérations jusqu'à ce que les paramètres d'entrée demandés soient fournis par l'utilisateur. Les workflows définissent les utilisateurs qui peuvent fournir les informations demandées et envoyer les demandes d'interaction en conséquence. Les workflows en attente d'une entrée utilisateur s'affichent dans le panneau **Exécutions de workflows récentes** du tableau de bord de vRealize Orchestrator Client et dans le menu des notifications situé en haut à droite.

## Planifier les workflows dans le client vRealize Orchestrator

Vous pouvez utiliser la planification pour automatiser les exécutions de vos workflows vRealize Orchestrator.

Lorsque vous planifiez des exécutions de workflow, vous définissez la date, l'heure et les intervalles auxquels la tâche planifiée s'exécute.

#### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Sélectionnez votre workflow dans le menu **Bibliothèque** et cliquez sur **Planifier** dans le panneau de workflow.
- 3 Configurez les paramètres des tâches planifiées dans les catégories **Général**, **Planification** et **Workflow**.

---

**Note** La catégorie de paramètre **Workflow** est uniquement visible pour les workflows qui nécessitent des paramètres d'entrée.

---

Paramètre	Description
Nom	Le nom de la tâche planifiée.
Description	Une brève description détaillant l'objectif de la tâche planifiée.
Démarrer	La date et l'heure de la première exécution planifiée du workflow.
Démarrer si dans le passé	Indiquez si le workflow doit démarrer lorsque l'heure planifiée est passée. <b>Oui</b> démarre immédiatement le workflow planifié. <b>Non</b> démarre le workflow à la prochaine récurrence planifiée.
Planifier	Définissez le schéma de récurrence et les entrées du déclencheur d'événements pour la tâche planifiée.
Date de fin	Visible uniquement si l'option <b>Aucune récurrence</b> est sélectionnée. Définissez la date et l'heure de fin de la tâche planifiée.
Workflow	Saisissez les paramètres d'entrée du workflow.

4 Cliquez sur **Créer**.

#### Résultats

Vous avez créé une tâche planifiée pour le workflow. Les workflows planifiés s'affichent sous **Activité > Planifié**. Vous pouvez supprimer des tâches planifiées en cliquant sur **Supprimer** dans le panneau de planification.

## Modifier une tâche planifiée dans le client vRealize Orchestrator

Les tâches planifiées peuvent être modifiées pour changer les paramètres tels que la date, l'heure et la récurrence du workflow planifié.

#### Conditions préalables

Créez une tâche de workflow planifiée.

#### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Sélectionnez votre tâche planifiée dans **Activité > Planifié**.
- 3 Cliquez sur **Modifier** dans le panneau de workflow.
- 4 Modifiez la planification et cliquez sur **Enregistrer**.

**Note** Les paramètres d'entrée définis lors de la création de la tâche planifiée sont en lecture seule et ne peuvent pas être modifiés. Pour modifier ces paramètres, créez une nouvelle tâche planifiée pour ce workflow.

## Rechercher des références d'objets dans les workflows

En tant que développeur de workflows, vous pouvez utiliser les informations de référence d'objet pour optimiser votre cycle de vie de développement.

vRealize Orchestrator Client vous permet d'obtenir des informations de référence sur l'objet. Cette fonctionnalité vous permet de :

- **Rechercher des dépendances** : obtenez des informations sur les dépendances d'objets dans les workflows. Les dépendances peuvent inclure d'autres workflows, actions, éléments de ressource et éléments de configuration.
- **Rechercher des utilisations** : déterminez si le workflow sélectionné est utilisé dans d'autres workflows de la bibliothèque de vRealize Orchestrator Client.

Vous pouvez accéder à des informations sur les références d'objet à partir de l'éditeur de workflow ou de la bibliothèque de vRealize Orchestrator Client dans la vue de fiche, de liste ou d'arborescence. Pour plus d'informations sur les différents types d'organisation de contenu de la bibliothèque de vRealize Orchestrator Client, reportez-vous à la section [Organisation de contenu dans vRealize Orchestrator Client](#).

La procédure suivante montre comment vous pouvez accéder aux références d'objets à partir de l'éditeur de workflow.

### Conditions préalables

Développez un workflow qui inclut au moins une référence d'objet.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Workflows** et sélectionnez votre workflow.
- 3 Pour trouver des informations sur les dépendances d'objet, cliquez sur **Rechercher des dépendances**.

---

**Note** Dans la fenêtre contextuelle des dépendances, vous pouvez sélectionner des objets référencés dans la liste. La sélection d'un objet ouvre un onglet vRealize Orchestrator Client distinct dans lequel vous pouvez afficher les détails de l'objet sélectionné ou le modifier.

---

- 4 Pour obtenir des informations sur les autres workflows dans lesquels votre workflow est utilisé, cliquez sur **Rechercher des utilisations**.

# Gestion des actions

# 6

Vous pouvez modifier vos workflows vRealize Orchestrator en ajoutant des scripts d'action.

Le vRealize Orchestrator Client fournit des bibliothèques d'actions prédéfinies et un éditeur d'actions pour les scripts d'action personnalisés. Les actions représentent des fonctions individuelles que vous utilisez en tant que composants de base dans les workflows.

Les actions sont des fonctions JavaScript. Les actions peuvent présenter plusieurs paramètres d'entrée et une seule valeur de retour. Elles peuvent appeler n'importe quel objet de l'API vRealize Orchestrator ou des objets de n'importe quelle API que vous importez dans vRealize Orchestrator via un plug-in.

Lorsqu'un workflow s'exécute, l'action récupère les paramètres d'entrée auprès des variables du workflow. Ces variables peuvent correspondre soit aux paramètres d'entrée initiaux du workflow, soit aux variables que les autres éléments du workflow définissent au cours de leur exécution.

L'éditeur d'actions inclut une fonctionnalité de saisie automatique pour les scripts et un Explorateur d'API contenant les types de scripts disponibles ainsi que leur documentation.

Ce chapitre contient les rubriques suivantes :

- [Créer des actions dans vRealize Orchestrator Client](#)
- [Exécution et débogage d'actions](#)
- [Concepts de base pour les scripts Python, Node.js et PowerShell](#)
- [Limites d'exécution pour les scripts Python, Node.js et PowerShell](#)

## Créer des actions dans vRealize Orchestrator Client

Vous pouvez utiliser vRealize Orchestrator Client pour créer, modifier et supprimer des scripts d'action.

Vous pouvez utiliser les environnements d'exécution suivants lors de la création d'actions :

- Python 3.7
- Node.js 14
- PowerCLI 11/Powershell 6.2

- PowerCLI 12.3.0/PowerShell 7.1

---

**Note** L'exécution de PowerCLI inclut PowerShell et les modules suivants : VMware.PowerCLI, PowerNSX, PowervRA.

---

### Conditions préalables

Avant de créer un script Python, Node.js ou PowerShell, vérifiez que vous maîtrisez bien les concepts essentiels du développement de scripts compatibles vRealize Orchestrator qui utilisent ces exécutions. Reportez-vous à [Concepts de base pour les scripts Python, Node.js et PowerShell](#).

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Actions**.
- 3 Cliquez sur **Nouvelle action**.
- 4 Dans l'onglet **Général**, entrez le nom et le nom du module de l'action.

---

**Note** Le nom et le nom du module doivent être uniques pour chaque action. Le nom de l'action doit être une fonction JavaScript valide. Le nom de l'action doit être un mot unique qui ne peut contenir que des lettres, des chiffres, ainsi que les symboles dollar (« \$ ») et trait de soulignement (« \_ »). Le nom du module doit être composé de mots séparés par un point (« . »).

---

- 5 (Facultatif) Créez une description, un numéro de version, des balises et des autorisations de groupe pour l'action.
- 6 Dans l'onglet **Script**, ajoutez des entrées d'action, sélectionnez le type de retour de la sortie et écrivez le script.

---

**Note** En sélectionnant **Zip** dans le menu déroulant **Type**, vous pouvez importer une source de script externe et, le cas échéant, ses modules de dépendance.

---

- 7 Pour terminer la modification de l'action, cliquez sur **Enregistrer**.

Un message indique que l'action est sauvegardée.

### Étape suivante

Pour voir un exemple de cas d'utilisation illustrant le mode d'emploi des actions vRealize Orchestrator, consultez [Comment intégrer des services Amazon Web Services dans vRealize Orchestrator à l'aide de Python](#).

## Exécution et débogage d'actions

Vous pouvez améliorer vos actions en les exécutant et en les déboguant directement à partir de l'éditeur d'actions.

Vous pouvez exécuter et déboguer des actions directement à partir de l'éditeur d'actions de vRealize Orchestrator Client. Avec cette fonctionnalité, vous pouvez garantir que vos actions s'exécutent comme prévu lorsqu'elles sont intégrées dans vos workflows.

## Exécuter des actions dans vRealize Orchestrator Client

En tant que concepteur de workflows, vous souhaitez exécuter vos actions avant de les intégrer à un workflow.

### Conditions préalables

Créez une action. Reportez-vous à [Créer des actions dans vRealize Orchestrator Client](#).

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Accédez à **Bibliothèque > Actions** et sélectionnez l'action à exécuter.
- 3 Cliquez sur **Exécuter**.
- 4 Entrez les paramètres d'entrée requis et cliquez sur **Exécuter**.

À la fin de l'exécution de l'action, cliquez sur l'onglet **Résultats/entrées**. Si l'exécution de l'action a rencontré une erreur, elle s'affiche dans cet onglet avec une couleur rouge. Vous pouvez afficher les détails de l'exécution de l'action à partir de l'élément **Résultats de l'action**.

---

**Note** Les résultats de l'exécution de l'action ne sont pas enregistrés.

---

## Actions de débogage dans vRealize Orchestrator Client

En tant que concepteur de workflows, vous pouvez déboguer des actions en insérant des points de rupture dans votre script.

vRealize Orchestrator comporte un outil de débogage intégré que vous pouvez utiliser pour déboguer le script et les propriétés d'entrée de votre action. Le processus de débogage peut être lancé dans l'éditeur d'actions en insérant des points de rupture dans les lignes de script de votre action.

---

**Note** L'outil de débogage intégré fonctionne uniquement avec les actions qui utilisent l'exécution JavaScript par défaut. Pour obtenir un exemple de débogage des scripts d'action qui utilisent des exécutions différentes, reportez-vous à la section [Déboguer l'action Amazon Web Services](#).

---

### Conditions préalables

Créez une action. Reportez-vous à [Créer des actions dans vRealize Orchestrator Client](#).

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Actions** et sélectionnez l'action à déboguer.

3 Dans l'éditeur d'actions, ajoutez des points de rupture aux lignes de votre script d'action que vous souhaitez déboguer.

4 Cliquez sur **Débogage**.

5 Entrez les paramètres d'entrée de votre action, puis cliquez sur **Exécuter**.

Une exécution d'action en mode de débogage commence.

6 Lorsque l'exécution de l'action s'interrompt à chaque point de rupture, sélectionnez l'une des options suivantes :

Option	Description
Continuer	Reprend l'exécution de l'action jusqu'à ce qu'un autre point de rupture soit atteint ou que l'exécution de l'action se termine.
Entrer dans	Entrer dans la fonction d'action actuelle. Si le débogueur ne peut pas approfondir la ligne actuelle de la fonction, il effectue une opération <b>Passer</b> .
Passer	Le débogueur continue sur la ligne suivante de la fonction actuelle.
Retour de l'étape	Le débogueur entre dans la ligne qui sera effectuée lors du retour de la fonction actuelle.

7 (Facultatif) Dans l'onglet **Débogueur**, ajoutez des expressions.

8 (Facultatif) Dans l'onglet **Débogueur**, modifiez la valeur de vos variables.

## Concepts de base pour les scripts Python, Node.js et PowerShell

Lorsque vous créez un script pour une utilisation dans vRealize Orchestrator, vous devez vérifier qu'il a la structure et la mise en forme appropriées.

### Exécutions prises en charge

Pour développer des actions et des workflows vRealize Orchestrator, vous pouvez utiliser les environnements d'exécution suivants :

- Python 3.7
- Node.js 14
- PowerCLI 11/Powershell 6.2
- PowerCLI 12.3.0/Powershell 7.1

**Note** L'exécution de PowerCLI inclut PowerShell et les modules suivants : VMware.PowerCLI, PowerNSX, PowervRA.

Vous pouvez ajouter n'importe quel code source personnalisé aux nouvelles exécutions, mais pour accepter le contexte et les entrées, et renvoyer le résultats depuis et vers le moteur vRealize Orchestrator, vous devez respecter le format fonctionnel approprié.

## Recommandations relatives aux scripts

Pour les tâches d'écriture de scripts les plus simples, vous pouvez ajouter des éléments **Tâche pouvant contenir des scripts** à votre schéma de workflow. Vous pouvez utiliser les actions vRealize Orchestrator pour les tâches d'écriture de scripts plus complexes.

L'utilisation d'actions offre deux avantages spécifiques :

- Les actions peuvent être créées, mises à jour, importées et exportées indépendamment des workflows.
- Les actions sont des objets autonomes qui peuvent être exécutés et débogués dans leur propre environnement, ce qui peut entraîner un processus de développement plus fluide. Reportez-vous à [Exécution et débogage d'actions](#) .

## Conditions requises pour la fonction de script

Le nom par défaut de votre fonction de script est **handler**. La fonction accepte deux arguments, context et input. L'argument context est un objet de mappage qui contient des informations système. Par exemple, `vroURL` peut contenir l'URL de l'instance de vRealize Orchestrator que vous souhaitez appeler, alors que `executionId` contient l'ID de jeton d'une exécution de workflow.

Une entrée est un objet de carte contenant toutes les entrées fournies aux actions. Par exemple, si vous définissez une entrée dans votre action appelée `myInput`, vous pouvez y accéder à partir de l'argument input, tel que `inputs.myInput` ou `inputs["myInput"]`, selon votre exécution. Tout ce que la fonction renvoie est le résultat de l'action. Par conséquent, le type de retour de votre action doit correspondre au type de contenu renvoyé par le script dans vRealize Orchestrator. Si vous renvoyez une primitive, le type de retour de l'action doit être un type de nombre. Si vous renvoyez une chaîne, le type de retour de l'action doit être un type de chaîne. Si vous renvoyez un objet complexe, le type de retour doit être mappé à `Properties` ou `Composite Type`. Ces mêmes principes s'appliquent également aux tableaux.

Types de paramètres d'entrée et de sortie pris en charge pour les exécutions Python, Node.js et PowerShell :

- `String`
- `Number`
- `Boolean`
- `Date`
- `Properties`
- `Composite Type`



## Définir le gestionnaire d'entrée

Par défaut, la valeur du gestionnaire d'entrée est `handler.handler`. Cette valeur signifie que le moteur de vRealize Orchestrator recherche un fichier de niveau supérieur dans votre module ZIP appelé `handler.py`, `handler.js` ou `handler.ps1`, qui inclut une fonction appelée `handler`. Les différences entre les noms des fichiers de fonction et de gestionnaire doivent être reflétées dans la valeur du gestionnaire d'entrées. Par exemple, si votre gestionnaire principal est appelé `index.js` et que votre fonction est appelée `callMe`, vous devez définir la valeur du gestionnaire d'entrée sur `index.callMe`.

## Déboguer les scripts d'exécution dans un IDE externe

vRealize Orchestrator prend en charge le débogage de scripts Python et Node.js dans un IDE externe. Vous ne pouvez pas déboguer des scripts PowerShell dans un IDE externe.

## Limites d'exécution pour les scripts Python, Node.js et PowerShell

Certains scripts Python, Node.js ou PowerShell peuvent nécessiter la modification des valeurs de mémoire et de délai d'expiration dans vRealize Orchestrator Client.

vRealize Orchestrator Client utilise un ensemble de valeurs de mémoire et de délai d'expiration par défaut pour les scripts d'action Python, Node.js et PowerShell :

- Mémoire : 64 Mo
- Délai d'expiration : 180 secondes

Si votre script d'action dépasse une de ces valeurs par défaut ou les deux, l'exécution de l'action échoue. Par exemple, votre script d'action peut utiliser plusieurs modules de dépendance tiers. Dans ce cas, la limite de mémoire par défaut de 64 Mo peut être insuffisante.

Pour éviter l'échec d'exécution d'actions en raison de ressources insuffisantes, modifiez les valeurs de mémoire et de délai d'expiration depuis l'éditeur d'actions.

---

**Note** Vous pouvez également envisager de diviser votre script en plusieurs éléments de tâche pouvant contenir des scripts, qui peuvent être ajoutés à vos workflows.

---

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Accédez à **Bibliothèque > Actions**, puis sélectionnez votre action.
- 3 Sélectionnez l'onglet **Script**.
- 4 Sous **Limites d'exécution**, modifiez les valeurs de mémoire et de délai d'expiration.
- 5 Cliquez sur **Enregistrer**.
- 6 Pour tester les nouvelles limites d'exécution, cliquez sur **Déboguer**.

# Gestion des éléments de configuration

# 7

Un élément de configuration est une liste de variables que vous pouvez utiliser pour configurer les constantes tout au long du déploiement d'un serveur vRealize Orchestrator.

Vous pouvez utiliser les éléments de configuration pour rendre les variables disponibles pour l'ensemble des workflows, des actions et des stratégies s'exécutant sur le serveur vRealize Orchestrator.

Si vous créez un module contenant un workflow, une action ou une stratégie qui utilise une variable provenant d'un élément de configuration, vRealize Orchestrator inclut automatiquement cet élément de configuration dans le module. Si vous importez un module contenant un élément de configuration dans un autre serveur vRealize Orchestrator, vous pouvez également importer les valeurs des variables de l'élément de configuration. Par exemple, si vous créez un workflow qui s'appuie sur des valeurs de variables dépendant du serveur vRealize Orchestrator sur lequel il s'exécute, la définition de ces variables dans un élément de configuration vous permet d'exporter ce workflow de manière à ce qu'un autre serveur vRealize Orchestrator puisse l'utiliser. Les éléments de configuration vous permettent donc d'échanger plus facilement les workflows, les actions et les stratégies entre les serveurs.

---

**Note** Vous ne pouvez pas importer les valeurs d'une variable d'élément de configuration à partir d'un élément de configuration exporté depuis vRealize Orchestrator 5.1 ou version antérieure.

---

Ce chapitre contient les rubriques suivantes :

- [Créer des éléments de configuration dans vRealize Orchestrator Client](#)

## Créer des éléments de configuration dans vRealize Orchestrator Client

Les éléments de configuration vous permettent de définir des variables communes sur un serveur vRealize Orchestrator. Tous les éléments s'exécutant sur le serveur peuvent utiliser les variables que vous avez définies dans un élément de configuration.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Ressources > Configurations**.

- 3 Sélectionnez **Nouvelle configuration**.
- 4 Entrez le nom de l'élément de configuration.
- 5 Sélectionnez l'onglet **Variables**.
- 6 Pour créer variable locale, cliquez sur **Nouveau**.
  - a Entrez le nom de la variable.
  - b Sélectionnez le type de variable.

---

**Note** Pour créer un groupe de variables de configuration, cochez la case **Groupe**.

---

- c (Facultatif) Entrez une valeur pour la variable de configuration.
  - d Cliquez sur **Enregistrer**.
- 7 Pour terminer la création d'un élément de configuration, cliquez sur **Enregistrer**.

#### Étape suivante

Vous pouvez utiliser cet élément de configuration pour fournir des variables aux workflows, aux actions ou aux stratégies.

# Gestion des stratégies

# 8

Les stratégies constituent des déclencheurs d'événement qui surveillent l'activité du système. Elles répondent à des événements prédéfinis issus de modifications de l'état ou des performances d'objets spécifiques de vRealize Orchestrator.

Les stratégies constituent une série de règles, d'indicateurs de stratégies, de seuils et de filtres d'événement qui exécutent certains workflows ou scripts lors de la survenue d'événements prédéfinis dans vRealize Orchestrator ou dans les technologies auxquelles vRealize Orchestrator accède via des plug-ins. vRealize Orchestrator évalue constamment les règles de stratégie pendant l'exécution de cette dernière. Vous pouvez par exemple mettre en œuvre des indicateurs de stratégies et des seuils pour surveiller le comportement des objets vCenter Server de types `VC:HostSystem` et `VC:VirtualMachine`.

Ce chapitre contient les rubriques suivantes :

- [Créer et appliquer des stratégies dans vRealize Orchestrator Client](#)
- [Éléments de stratégie dans vRealize Orchestrator Client](#)
- [Gérer l'exécution des stratégies dans vRealize Orchestrator Client](#)

## Créer et appliquer des stratégies dans vRealize Orchestrator Client

Vous pouvez utiliser des stratégies pour surveiller l'activité du système vRealize Orchestrator lors d'événements spécifiques.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Bibliothèque > Stratégies**.
- 3 Sélectionnez **Nouvelle stratégie**.  
Vous avez créé une stratégie vide.
- 4 Entrez un nom de stratégie et un numéro de version.
- 5 Sélectionnez l'onglet **Variables**.

**6** Pour créer variable locale, cliquez sur **Nouveau**.

- a Entrez le nom de la variable.
- b Sélectionnez le type de variable.

---

**Note** Pour créer un groupe de variables de stratégie, cochez la case **Groupe**.

---

- c Entrez la valeur de la variable.

---

**Note** Vous pouvez utiliser la fonction **Lier à la configuration** pour importer la valeur de variable d'un élément de configuration.

---

- d Cliquez sur **Enregistrer**.

**7** Dans l'onglet **Définition**, vous pouvez ajouter des éléments de stratégie et définir des gestionnaires d'événements.

Pour plus d'informations sur les éléments de stratégie, reportez-vous à [Éléments de stratégie dans vRealize Orchestrator Client](#).

**8** Cliquez sur **Enregistrer**.

Vous avez configuré la stratégie.

**Étape suivante**

Pour démarrer une stratégie, sélectionnez la stratégie et cliquez sur **Exécuter**. Entrez le nom d'exécution de la stratégie et, si vous y êtes invité, les paramètres d'entrée requis.

Pour afficher l'état de la stratégie, accédez à **Activité > Exécution de stratégies**.

## Éléments de stratégie dans vRealize Orchestrator Client

Vous pouvez utiliser des éléments de stratégie pour exécuter des workflows ou des scripts prédéfinis de vRealize Orchestrator lorsqu'un événement se produit.

Vous pouvez ajouter un élément de stratégie pour déclencher des exécutions de workflows ou de scripts en réponse à des événements déclenchés par des objets. L'élément d'événement périodique vous permet de planifier des exécutions de workflows ou de scripts. L'élément racine vous permet de définir le comportement de démarrage ou d'arrêt des stratégies. Les éléments de stratégie peuvent avoir des gestionnaires d'événements qui définissent le moment auquel les éléments de stratégie doivent être exécutés.

---

**Note** Les gestionnaires d'événements qui activent les éléments de stratégie peuvent être des workflows ou des scripts d'action. Si vous ajoutez un workflow et un script à un gestionnaire d'événements, la stratégie ignore le déclencheur de script et n'utilise que le déclencheur de workflow.

---

Gestionnaire d'événements	Description
<b>OnInit</b>	L'élément de stratégie est déclenché à chaque démarrage de la stratégie.
<b>OnExit</b>	L'élément de stratégie est déclenché à chaque arrêt de la stratégie.
<b>OnExecute</b>	Utilisé par l'élément d'événement périodique. Déclenche l'élément de stratégie lors de la période spécifiée dans l'élément d'événement périodique.

**Note** Les technologies connectées à la base de données vRealize Orchestrator peuvent posséder des gestionnaires d'événements uniques. Par exemple, le plug-in SNMP (protocole simple de gestion de réseau) vous permet d'utiliser le gestionnaire d'événements **OnTrap** lors de la création d'éléments de stratégie basés sur SNMP.

Les éléments de stratégie sont configurés dans l'onglet **Définition** de la fenêtre d'édition de la stratégie.

## Gérer l'exécution des stratégies dans vRealize Orchestrator Client

Vous pouvez utiliser vRealize Orchestrator Client pour gérer la priorité des stratégies et le comportement de démarrage du serveur vRealize Orchestrator lors de son redémarrage.

### Conditions préalables

Créez et exécutez une stratégie. Pour plus d'informations, reportez-vous à [Créer et appliquer des stratégies dans vRealize Orchestrator Client](#).

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator en tant qu'administrateur.
- 2 Accédez à **Activité > Exécution des stratégies**.
- 3 Cliquez sur l'exécution de stratégie que vous souhaitez gérer.
- 4 Cliquez sur **Arrêter**.  
L'état de la stratégie passe à **Arrêté**.
- 5 Dans l'onglet **Général**, définissez la priorité de la stratégie et le comportement de démarrage du serveur.
- 6 Pour redémarrer la stratégie, cliquez sur **Exécuter**.  
L'état de la stratégie passe à **En cours d'exécution**.

# Gestion des éléments de ressource

## 9

Les workflows peuvent utiliser des objets que vous créez indépendamment de vRealize Orchestrator en tant qu'attributs. Pour utiliser les objets externes en tant qu'attributs dans les workflows, importez-les dans le serveur en tant qu'éléments de ressource.

Les objets que les workflows vRealize Orchestrator peuvent utiliser en tant qu'éléments de ressource incluent les fichiers image, les scripts, les modèles XML, les fichiers HTML, etc. Tous les workflows s'exécutant sur le serveur vRealize Orchestrator peuvent utiliser n'importe quel élément de ressource que vous importez dans vRealize Orchestrator.

Après avoir importé un objet en tant qu'élément de ressource dans vRealize Orchestrator, vous pouvez apporter des modifications à l'objet à un emplacement unique et les répercuter automatiquement à l'ensemble des workflows utilisant cet élément de ressource.

La taille maximale d'un élément de ressources est de 16 Mo.

Vous pouvez importer, exporter, restaurer, mettre à jour et supprimer un élément de ressource.

# Gestion des modules

# 10

Utilisez vRealize Orchestrator Client pour créer, exporter et importer des modules. Les modules peuvent être utilisés pour exporter des objets de workflow pour une utilisation sur les autres instances de vRealize Orchestrator.

Les modules peuvent contenir des workflows, des actions, des stratégies, des éléments de configuration ou des éléments de ressources.

Lorsque vous ajoutez un élément à un module, vRealize Orchestrator vérifie les dépendances et ajoute tous les éléments dépendants au module. Par exemple, si vous ajoutez un workflow qui utilise des actions ou d'autres workflows, vRealize Orchestrator ajoute ces actions et ces workflows au module.

Lorsque vous importez un module, le serveur compare les versions des différents éléments de son contenu aux éléments locaux correspondants. La comparaison indique les différences de versions entre les éléments locaux et importés. L'utilisateur peut décider s'il souhaite importer le module ou sélectionner des éléments spécifiques.

Pour la plupart des objets créés dans vRealize Orchestrator Client, à l'exception des éléments de ressource, les modules sont le seul moyen d'exporter et d'importer ces objets.

Les modules s'appuient sur la gestion des droits numériques pour contrôler la façon dont le serveur récepteur peut utiliser leur contenu. vRealize Orchestrator signe les modules et les chiffre pour protéger les données. Les modules peuvent identifier les utilisateurs qui exportent et redistribuent les éléments grâce aux certificats X509.

## Créer un module dans vRealize Orchestrator Client

Vous pouvez exporter et importer les workflows, les stratégies, les actions, les références de plug-ins, les éléments de ressource et les éléments de configuration dans les modules. Tous les éléments dépendants associés aux objets du module sont automatiquement ajoutés à ce dernier afin d'assurer la compatibilité entre les versions. Pour supprimer des éléments dépendants, vous devez d'abord supprimer l'objet du module associé.

Pour la plupart des objets créés dans vRealize Orchestrator Client, à l'exception des éléments de ressource, les modules sont le seul moyen d'exporter et d'importer ces objets.



### Conditions préalables

Vérifiez que le serveur vRealize Orchestrator contient des objets tels que des workflows, des actions et des stratégies que vous pouvez ajouter à un module.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Actifs > Modules**.
- 3 Cliquez sur **Nouveau module**.
- 4 Dans l'onglet **Général** entrez un nom et une description du module.

---

**Note** Vous ne pouvez pas utiliser de caractères spéciaux lorsque vous nommez des modules dans vRealize Orchestrator Client.

---

- 5 Dans l'onglet **Contenu**, cliquez sur **Ajouter**.
- 6 Sélectionnez les objets à ajouter au module et cliquez sur **Ajouter**.

---

**Note** Les éléments dépendants sont ajoutés automatiquement au module, mais ne s'affichent pas dans l'onglet **Contenu** lors de la création du module. Pour afficher les éléments dépendants, sélectionnez l'onglet **Contenu** après la création du module.

---

- 7 Pour terminer la création du module, cliquez sur **Créer**.

## Exporter un module dans vRealize Orchestrator Client

Vous pouvez utiliser vRealize Orchestrator Client pour exporter des modules vers un autre environnement vRealize Orchestrator.

### Conditions préalables

Créez un module contenant les objets vRealize Orchestrator à exporter. Pour plus d'informations, reportez-vous à [Créer un module dans vRealize Orchestrator Client](#).

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Actifs > Modules**.
- 3 Cliquez sur **Exporter** sur le module.

#### 4 (Facultatif) Sélectionnez les options d'exportation supplémentaires.

Option	Description
Ajouter les valeurs d'attribut de configuration au module	Exportez les valeurs d'attribut des éléments de configuration.
Ajouter les valeurs d'attribut SecureString de configuration au module	Exportez les valeurs d'attribut de configuration <code>SecureString</code> .
Ajouter les balises globales au module	Exportez les balises globales.

#### 5 Définissez les droits d'accès des utilisateurs qui importent le module.

Option	Description
Afficher le contenu	L'utilisateur peut afficher le contenu du module.
Ajouter au module	L'utilisateur peut ajouter du contenu du module importé à d'autres modules.
Modifier le contenu	L'utilisateur peut modifier le contenu du module.

#### 6 Cliquez sur **OK**.

**Note** Les fichiers présentant l'extension `.package` sont enregistrés dans un dossier par défaut sur votre machine locale. Pour définir un dossier personnalisé, vous pouvez modifier les paramètres de stockage dans votre navigateur.

### Résultats

Vous venez d'exporter le module. Vous pouvez désormais utiliser les objets exportés dans un autre environnement vRealize Orchestrator.

## Importer un module dans vRealize Orchestrator Client

Utilisez vRealize Orchestrator Client pour importer des modules de workflow. En important des modules, vous pouvez réutiliser les objets d'un serveur vRealize Orchestrator sur un autre serveur.

### Conditions préalables

- Sauvegardez tous les objets standard de vRealize Orchestrator que vous avez modifiés.
- Sur le serveur distant, créez et exportez un module avec les objets à importer.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Actifs > Modules**.
- 3 Cliquez sur **Importer**, recherchez le fichier `.package` à importer et cliquez sur **Ouvrir**.

4 Vérifiez les informations du module importé.

- a L'onglet **Général** contient des informations sur le module importé, comme le nom, la description, le nombre d'éléments contenus et les informations de certificat.

Vous serez peut-être invité à indiquer que vous approuvez le certificat de l'éditeur de l'instance source de vRealize Orchestrator avant de pouvoir importer le fichier.

- b L'onglet **Éléments du module** répertorie les objets inclus dans le fichier d'importation. Si la version d'un objet du module est plus récente que celle qui se trouve sur le serveur, le système sélectionne cette version de l'objet à importer. Les versions antérieures d'éléments vRealize Orchestrator doivent être sélectionnées manuellement.
- c Désélectionnez **Importer les valeurs d'attribut de configuration** si vous ne souhaitez pas importer les valeurs d'attribut des éléments de configuration issus du module.
- d Dans le menu déroulant, indiquez si vous souhaitez importer les balises.

5 Cliquez sur **Importer**.

# Dépannage dans le client vRealize Orchestrator

# 11

Vous pouvez dépanner et surveiller votre instance de vRealize Orchestrator à l'aide des mesures, de la relecture de jeton, de la validation et du débogage.

Ce chapitre contient les rubriques suivantes :

- [Données de mesure dans le client vRealize Orchestrator](#)
- [Utilisation de la relecture de jeton de workflow dans vRealize Orchestrator Client](#)
- [Validation des workflows vRealize Orchestrator](#)
- [Déboguer les scripts de workflow dans vRealize Orchestrator Client](#)
- [Workflows de débogage par élément de schéma](#)
- [Configuration d'un conteneur Photon OS pour les modules Python](#)

## Données de mesure dans le client vRealize Orchestrator

Les administrateurs vRealize Orchestrator peuvent utiliser le profilage de workflow et des mesures du tableau de bord système pour dépanner le système et les workflows vRealize Orchestrator.

La fonctionnalité de profilage collecte des données de mesure à propos des exécutions de workflow. Le profilage de workflow est activé par défaut. Vous pouvez désactiver le profilage automatique dans **Centre de contrôle > Propriétés de l'extension > profiler-8.7.0**.

L'autre source pour les données de mesure dans vRealize Orchestrator Client est le tableau de bord système qui fournit des mesures de niveau système. Pour plus d'informations, reportez-vous à [Utilisation du tableau de bord système de vRealize Orchestrator](#).

## Profilage des workflows dans le client vRealize Orchestrator

Vous pouvez profiler vos exécutions de workflow pour dépanner et optimiser votre environnement vRealize Orchestrator.

Vous pouvez utiliser la fonctionnalité de profilage vRealize Orchestrator Client pour recueillir des données de mesure utiles sur vos exécutions de workflow. Ces données peuvent être utilisées pour optimiser les performances de vos workflows. Par défaut, les exécutions de workflow sont profilées automatiquement. Vous pouvez désactiver le profilage automatique à partir de la page **Propriétés de l'extension** du centre de contrôle vRealize Orchestrator et exécuter le profileur manuellement. Pour exécuter un profilage manuel, recherchez votre workflow dans la bibliothèque et sélectionnez **Actions > Profil**.

#### Conditions préalables

Exécutez un workflow.

#### Procédure

- 1 Connectez-vous au client vRealize Orchestrator.
- 2 Accédez à **Activité > Exécution du workflow**.
- 3 Sélectionnez une exécution de workflow.

Sur le schéma d'exécution du workflow, vous pouvez voir des données sur les éléments de workflow individuels. Les données incluent la durée totale d'exécution, la durée maximale et le nombre d'exécutions d'éléments. Vous pouvez filtrer ces informations dans le menu déroulant en haut à droite de la page.

- 4 Sélectionnez l'onglet **Performances**.

Cet onglet fournit des données de mesure sur les temps CPU d'exécution du workflow, la durée d'exécution, la taille de jeton et les données d'élément de workflow.

---

**Note** Si le workflow est interrompu (lorsqu'il attend d'autres saisies, par exemple), la mesure du temps CPU capture uniquement le thread d'exécution qui se produit avant la fin.

---

#### Étape suivante

Utilisez les données collectées à partir du profilage pour optimiser votre workflow.

## Utilisation du tableau de bord système de vRealize Orchestrator

En tant qu'administrateur, vous pouvez utiliser le tableau de bord système de vRealize Orchestrator Client pour collecter des données de mesure utiles sur les nœuds de votre environnement vRealize Orchestrator.

Vous pouvez accéder au tableau de bord système en cliquant sur l'onglet **Système** en haut de la page du tableau de bord de vRealize Orchestrator Client. Les données fournies incluent :

- État du nœud
- Propriétés du nœud
- Paramètres du cluster. Vous pouvez uniquement afficher les paramètres du cluster à partir du tableau de bord système. Pour modifier ces paramètres, accédez à la page **Gestion du cluster Orchestrator** du centre de contrôle vRealize Orchestrator.

- Infos sur les threads
- Mémoire de segment de mémoire
- Mémoire sans segment de mémoire
- Utilisation du système fichier
- Données d'authentification
- Pool de connexions de base de données Orchestrator
- Traiter les arguments de saisie

Ces données peuvent être utilisées pour surveiller l'état des nœuds individuels de votre environnement vRealize Orchestrator et résoudre les problèmes. Pour naviguer entre les nœuds individuels, cliquez sur l'onglet associé à un nœud en haut du tableau de bord système.

## Utilisation de la relecture de jeton de workflow dans vRealize Orchestrator Client

Vous pouvez utiliser la fonctionnalité de relecture de jeton pour afficher les transitions entre les éléments dans les exécutions de workflow.

La fonctionnalité de relecture de jeton enregistre des informations contextuelles pour chaque transition entre les éléments Workflow. Pour chaque élément de workflow, la relecture de jeton enregistre l'heure de début et de fin d'exécution du workflow ainsi que les variables qui ont été modifiées à la fin de l'exécution de l'élément de workflow. La relecture de jeton fait également référence aux messages de journal de script générés pour chaque élément workflow.

---

**Note** Les données sur les transitions d'éléments de workflow sont stockées dans la base de données PostgreSQL de vRealize Orchestrator. Ces données sont supprimées de la base de données lors de la suppression de l'exécution du workflow.

---

### Conditions préalables

- Activez la fonctionnalité de relecture de jeton depuis le centre de contrôle.
  - a Connectez-vous au centre de contrôle en tant qu'utilisateur **racine**.
  - b Sélectionnez **Propriétés de l'extension**.
  - c Cliquez sur **tokenreplay-8.7.0**.
  - d Pour activer la fonctionnalité de relecture de jeton, cliquez sur **Activer**.
  - e Cliquez sur **Enregistrer**.

---

**Note** L'actualisation du serveur vRealize Orchestrator peut prendre jusqu'à 5 minutes.

---

- Exécutez un workflow.

**Note** Par défaut, la relecture de jeton ne s'exécute pas automatiquement pour toutes les exécutions de workflow sur votre serveur vRealize Orchestrator. Vous pouvez exécuter des relectures de jetons individuelles pour chaque workflow ou activer l'extension de relecture de jeton pour tous les workflows à partir de la page **Propriétés de l'extension** du centre de contrôle.

#### Procédure

- 1 (Facultatif) Activez la relecture de jeton pour toutes les exécutions de workflow sur votre serveur vRealize Orchestrator.

**Note** Pour exécuter des relectures de jetons individuelles sans activer la fonctionnalité depuis le centre de contrôle, cliquez sur **Exécuter avec relecture** sur la page de l'éditeur de workflow.

- a Connectez-vous au centre de contrôle en tant qu'utilisateur **racine**.
- b Sélectionnez **Propriétés de l'extension**.
- c Cliquez sur **tokenreplay-8.7.0**.
- d Pour activer la fonctionnalité de relecture de jeton pour tous les workflows, vérifiez que la fonctionnalité **Enregistrer la relecture pour toutes les exécutions de workflow** est activée.
- e Cliquez sur **Enregistrer**.

**Note** L'actualisation du serveur vRealize Orchestrator peut prendre jusqu'à 5 minutes.

- 2 Connectez-vous au client vRealize Orchestrator en tant qu'administrateur.
- 3 Accédez à **Activité > Exécution du workflow**.
- 4 Sélectionnez une exécution de workflow.
- 5 Sélectionnez un élément d'exécution de workflow dans le menu de gauche.

Les **Variables** et **Journaux** affichent désormais des informations spécifiques pour cet élément workflow.

## Validation des workflows vRealize Orchestrator

vRealize Orchestrator fournit un outil de validation des workflows. Valider un workflow aide à identifier les erreurs qui peuvent s'y trouver et contrôle que les données s'enchaînent correctement d'un élément au suivant.

Par défaut, vRealize Orchestrator procède systématiquement à la validation d'un workflow lors de son exécution.

Lorsque vous validez un workflow, l'outil de validation crée une liste de toutes les erreurs et avertissements éventuels. Cliquer sur une erreur de la liste met en évidence l'élément de workflow qui contient l'erreur.

Si vous lancez l'outil de validation dans l'éditeur de workflow, il proposera une suggestion de correctifs rapides pour les erreurs détectées. Certains correctifs rapides nécessitent des informations supplémentaires ou des paramètres d'entrée. D'autres correctifs rapides régleront l'erreur pour vous.

La validation des workflows contrôle les liaisons et connexions de données entre les éléments. La validation des workflows ne contrôle pas le traitement des données qu'effectue chaque élément du workflow. Par conséquent, un workflow valide peut s'exécuter de façon incorrecte et produire des résultats erronés si une fonction au sein d'un élément du schéma est incorrecte.

## Valider un workflow et corriger les erreurs de validation dans vRealize Orchestrator Client

Vous devez valider un workflow avant de pouvoir l'exécuter. Vous ne pourrez corriger les erreurs de validation que si vous avez ouvert le workflow pour modification.

### Conditions préalables

Vérifiez que vous disposez d'un workflow complet à valider, avec les éléments liés du schéma et les liaisons définies.

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator en tant qu'administrateur.
- 2 Accédez à **Bibliothèque > Workflows** et sélectionnez le workflow à valider.
- 3 Cliquez sur **Modifier**.
- 4 Cliquez sur **Valider** dans le menu supérieur.

Si le workflow est valide, un message de confirmation s'affiche. Si le workflow n'est pas valide, une liste des erreurs apparaît.

- 5 En cas de workflow non valide, cliquez sur l'un des messages d'erreur et prenez les mesures appropriées pour résoudre le problème.

L'outil de validation met en évidence l'élément du schéma dans lequel l'erreur s'est produite en l'affublant d'une icône rouge. Lorsque c'est possible, l'outil de validation affiche une mesure de correction rapide.

- Si vous êtes d'accord avec la mesure de correction rapide proposée, cliquez dessus pour l'appliquer.
- Si vous n'êtes pas d'accord avec la mesure de correction rapide proposée, refermez la boîte de dialogue Validation des workflows et corrigez l'élément du schéma par vous-même.

---

**Important** Vérifiez toujours que le correctif proposé par vRealize Orchestrator convient.

---

Par exemple, la mesure proposée pourrait être de supprimer un attribut inutilisé alors qu'en fait il se peut tout simplement que la liaison de cet attribut ne soit pas correcte.



- 6 Répétez les étapes qui précèdent jusqu'à avoir éliminé la totalité des erreurs de validation.

### Résultats

Vous venez de valider un workflow et d'en avoir corrigé les erreurs de validation.

### Étape suivante

Vous pouvez désormais exécuter le workflow.

## Déboguer les scripts de workflow dans vRealize Orchestrator Client

Vous pouvez déboguer les exécutions de workflow en insérant des points de rupture dans le script des éléments de workflow.

Lorsque vous atteignez un point de rupture, vous disposez de plusieurs options pour continuer le processus de débogage. Lorsque vous déboguez un élément du schéma de workflow, vous pouvez consulter les informations générales sur l'exécution du workflow, modifier les variables du workflow, ajouter des expressions à regarder et afficher les messages du journal.

---

**Note** Effectuez le débogage de script dans un environnement hors production.

---

### Procédure

- 1 Connectez-vous au client vRealize Orchestrator en tant qu'administrateur.
- 2 Sélectionnez un workflow dans la bibliothèque.
- 3 Ouvrez le schéma de workflow, sélectionnez un élément de workflow, puis cliquez sur l'onglet **Script**.
- 4 Pour insérer un point de rupture, cliquez sur le cercle rouge à gauche du numéro de ligne.

---

**Note** Vous pouvez uniquement insérer des points de rupture dans des éléments de workflow avec des scripts.

---

- 5 Pour exécuter le workflow en mode débogage, cliquez sur **Déboguer**.

Si le workflow demande des paramètres d'entrée, vous devez en indiquer.

- 6 Lorsque l'exécution du workflow s'interrompt à chaque point de rupture, sélectionnez l'une des options disponibles.

Option	Description
Continuer	Reprend l'exécution du workflow jusqu'à ce qu'un autre point de rupture soit atteint ou que l'exécution du workflow se termine.
Entrer dans	Vous pouvez utiliser cette option pour accéder à un élément de workflow. Vous ne pouvez pas entrer dans un élément de workflow imbriqué lorsque vous déboguez un workflow dans l'éditeur de workflows.
Passer	Ignore l'élément en cours dans le schéma et interrompt l'exécution du workflow sur l'élément suivant.

**Note** Vous pouvez demander au débogueur d'ignorer le point de rupture actuel en cliquant dessus. Cela modifie le symbole de point de rupture par un triangle vert.

- 7 (Facultatif) Insérez des expressions à surveiller dans l'onglet **Débogueur**.  
Vous pouvez utiliser des expressions pour suivre l'achèvement de variables spécifiques.
- 8 (Facultatif) Modifiez les valeurs des variables dans l'onglet **Débogueur**.

## Workflows de débogage par élément de schéma

En tant que concepteur de workflows, vous pouvez déboguer des éléments de schéma individuels.

### Procédure

- 1 Connectez-vous à vRealize Orchestrator Client.
- 2 Accédez à **Bibliothèque > Workflows** et sélectionnez votre workflow.
- 3 Sélectionnez l'onglet **Schéma**.
- 4 Sélectionnez l'élément de workflow que vous souhaitez déboguer, puis cliquez sur le bouton Déboguer situé en haut à gauche de l'élément.

**Note** En ajoutant un point de rupture à un élément de schéma **Élément de workflow**, vous pouvez déboguer les workflows enfants directement à partir du workflow parent. Lorsque le débogueur atteint l'élément de schéma **Élément de workflow**, il ouvre la vue Schéma du workflow enfant.

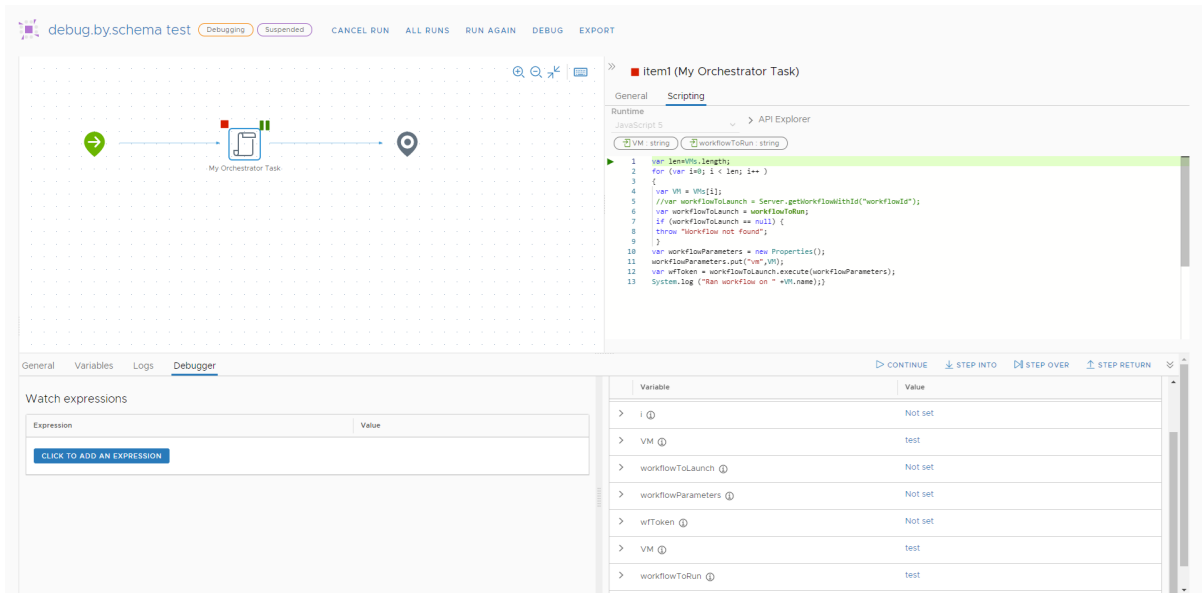
- 5 Répétez les étapes pour tous les autres éléments de schéma que vous souhaitez déboguer.
- 6 Cliquez sur **Débogage**.
- 7 Entrez les valeurs de paramètres d'entrée demandées et cliquez sur **Exécuter**.

L'exécution du workflow commence et est interrompue lorsque le débogueur atteint un élément de schéma incluant un point de rupture.

8 Au niveau d'un point de rupture, sélectionnez l'une des options suivantes :

Option	Description
Continuer	Reprend l'exécution du workflow jusqu'à ce qu'un autre point de rupture soit atteint ou que l'exécution du workflow se termine.
Entrer dans	Entrer dans la fonction de workflow en cours. Si le débogueur ne peut pas approfondir la ligne actuelle de la fonction, il effectue une opération <b>Passer</b> .
Passer	Le débogueur continue sur la ligne suivante de la fonction actuelle.
Retour de l'étape	Le débogueur entre dans la ligne qui sera effectuée lors du retour de la fonction actuelle.

9 (Facultatif) Dans l'onglet **Variables**, modifiez la valeur de vos variables du workflow.



## Configuration d'un conteneur Photon OS pour les modules Python

En fonction du système d'exploitation utilisé pour compiler le script Python, les workflows ou actions peuvent échouer après l'importation de l'archive ZIP correspondante dans le client vRealize Orchestrator.

Le système d'exploitation du conteneur d'exécution utilisé pour Python dans vRealize Orchestrator est basé sur Photon 3.0. Les modules de script Python compilés pour un autre système d'exploitation, comme Linux, sont incompatibles avec le conteneur d'exécution. Ce problème peut entraîner l'échec du script Python, lorsque vous tentez de l'utiliser dans le cadre de vos workflows ou actions de vRealize Orchestrator. Dans ce cas, vous recevez le message d'erreur suivant dans vos journaux :

```
-04:00errorCannot find module action
```

Pour résoudre ce problème, vous devez installer le module Python requis dans un dossier de conteneur Photon OS.

### Conditions préalables

Installez Docker. Reportez-vous à la section [Obtenir Docker](#).

### Procédure

- 1 Accédez au dossier parent de votre script Python.
- 2 Créez un conteneur avec l'image Photon de base en montant un dossier de conteneur dans le dossier parent.

---

**Note** Le script suivant est une commande Docker unique que vous devez exécuter dans son intégralité pour créer un conteneur approprié.

---

```
docker run -ti -v
$(pwd) /<name_of_folder_that_contains_your_python_script>:/:
<name_of_folder_that_contains_your_python_script>
photon:3.0
```

- 3 Installez Python dans le conteneur.

```
tdnf install -y python3-3.7.5-5.ph3 python3-pip-3.7.5-5.ph3
```

- 4 Accédez au dossier conteneur qui inclut votre script Python.
- 5 Ajoutez votre script Python et vos modules.

---

**Note** Installez les modules requis pour votre script Python dans le dossier `lib`.

---

```
pip3 install <package_name> -t lib/
```

- 6 Quittez le conteneur et accédez au dossier local que vous avez monté dans le conteneur.
- 7 Comprimez tous les fichiers et dossiers pertinents dans une archive ZIP.
- 8 Importez l'archive ZIP dans le client vRealize Orchestrator et validez le script en l'exécutant dans le cadre d'une action.