

# Usar o Cliente VMware vRealize Orchestrator

12 AGOSTO 2021

vRealize Orchestrator 8.5

Este documento foi traduzido automaticamente do inglês. Se você observar algum erro de tradução, deixe seu feedback na parte inferior da página específica da publicação no VMware Docs.

Você pode encontrar a documentação técnica mais atualizada no site da VMware, em:

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

**VMware, Inc.**

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

**VMware Brasil**

Rua Surubim, 504 4º andar CEP 04571-050  
Cidade Monções  
São Paulo  
SÃO PAULO: 04571-050  
Brasil  
Tel: +55 11 55097200  
Fax: + 55. 11. 5509-7224  
[www.vmware.com/br](http://www.vmware.com/br)

Copyright © 2008-2021 VMware, Inc. Todos os direitos reservados. [Informações sobre direitos autorais e marca registrada.](#)

# Conteúdo

- 1 Usar o Cliente VMware vRealize Orchestrator 6**
- 2 Introdução ao Cliente vRealize Orchestrator 7**
  - Painel de uso do Cliente vRealize Orchestrator 8
  - Organização de conteúdo no vRealize Orchestrator Client 8
    - Criar pastas ou subpastas 10
    - Mover objetos e pastas 11
    - Excluir pastas ou subpastas 11
- 3 Configurar o vRealize Orchestrator Client 13**
  - Funções e grupos do vRealize Orchestrator 13
    - Atribuir funções no vRealize Orchestrator Client 15
    - Configurar funções do Cliente vRealize Orchestrator no vRealize Automation 16
    - Criar grupos no vRealize Orchestrator Client 17
  - Histórico da versão do objeto vRealize Orchestrator 17
    - Restaurar um fluxo de trabalho para uma versão anterior 18
    - Comparação visual entre versões do fluxo de trabalho 19
    - Redefinir inventário de conteúdo do vRealize Orchestrator para um estado anterior com o Git 20
- 4 Casos de uso do vRealize Orchestrator 22**
  - Como integrar os Amazon Web Services no vRealize Orchestrator usando Python 22
    - Criar script do Python inicial 23
    - Criar a ação do Amazon Web Services 24
    - Depurar a ação do Amazon Web Services 25
    - Atualizar a ação do Amazon Web Services 28
  - Como posso usar a ramificação do Git para gerenciar o inventário de objeto do meu vRealize Orchestrator 29
    - Preparar o seu ambiente do GitLab 30
    - Configurar uma conexão para um Repositório do Git 31
    - Enviar alterações para um repositório Git 32
  - Como posso usar módulos de terceiros para chamar a API do projeto do vRealize Automation 34
    - Criar um script do Python que chama a API de Projeto do vRealize Automation 34
    - Criar um script Node.js que chame a API de Projeto do vRealize Automation 37
    - Criar um script do PowerShell que chama a API de Projeto do vRealize Automation 39
- 5 Gerenciar fluxos de trabalho 43**
  - Fluxos de trabalho padrão na biblioteca do fluxo de trabalho do vRealize Orchestrator 44

Criar fluxos de trabalho	44
Editar fluxos de trabalho e ações do fluxo de trabalho principal	45
Designer do formulário de entrada do vRealize Orchestrator	45
Criar a caixa de diálogo de parâmetros de entrada do fluxo de trabalho no Cliente vRealize Orchestrator	46
Propriedades do parâmetro de entrada no Cliente vRealize Orchestrator	46
Usar ações para validar entradas de fluxo de trabalho do vRealize Orchestrator	47
Solicitações de interação do usuário no Cliente vRealize Orchestrator	49
Agendar fluxos de trabalho	49
Editar tarefa agendada no Cliente vRealize Orchestrator	50
Localizar referências de objeto em fluxos de trabalho	51
<b>6 Gerenciar ações</b>	<b>52</b>
Criar ações	52
Ações de depuração e execução	53
Executar ações	54
Depurar ações	54
Conceitos básicos para scripts do Python, Node.js e PowerShell	55
Limites do tempo de execução para scripts Python, Node.js e PowerShell	57
<b>7 Gerenciar elementos de configuração</b>	<b>58</b>
Gerenciar elementos de configuração	58
<b>8 Gerenciar políticas</b>	<b>60</b>
Criar e aplicar políticas	60
Elementos de política	61
Gerenciar execuções de política	62
<b>9 Gerenciar elementos de recursos</b>	<b>63</b>
<b>10 Gerenciar pacotes</b>	<b>64</b>
Criar pacotes	64
Exportar pacotes	65
Importar pacotes	66
<b>11 Solução de problemas no Cliente vRealize Orchestrator</b>	<b>68</b>
Dados de métrica no Cliente vRealize Orchestrator	68
Fluxos de trabalho de perfil no Cliente vRealize Orchestrator	68
Usar o painel do sistema do vRealize Orchestrator	69
Usar reprodução de token de fluxo de trabalho no Cliente vRealize Orchestrator	70
Validar fluxos de trabalho do vRealize Orchestrator	71

Validar um fluxo de trabalho e corrigir erros de validação no Cliente vRealize Orchestrator	
72	
Depurar scripts de fluxo de trabalho no Cliente vRealize Orchestrator	73
Depurar fluxos de trabalho pelo elemento de esquema	74
Configurando um contêiner de Photon OS para pacotes Python	75

# Usar o Cliente VMware vRealize Orchestrator

# 1

*Usar o Cliente VMware vRealize Orchestrator* fornece informações sobre os recursos de automação de fluxo de trabalho e a funcionalidade do vRealize Orchestrator Client.

## Público-alvo

Essas informações destinam-se a administradores de sistema experientes que estão procurando uma ferramenta que possa ajudá-los a executar e gerenciar fluxos de trabalho do vRealize Orchestrator.

# Introdução ao Cliente vRealize Orchestrator

## 2

Use o vRealize Orchestrator Client para gerenciar serviços e objetos do vRealize Orchestrator.

Você pode acessar o vRealize Orchestrator Client em `https://FQDN_do_seu_orchestrator/orchestration-ui`.

Elementos da interface do usuário	Descrição
<b>Painel</b>	Use o Painel do vRealize Orchestrator Client e o recurso de perfil para coletar dados de métrica úteis sobre o ambiente e os fluxos de trabalho do vRealize Orchestrator.
<b>Fluxos de trabalho</b>	Crie, edite, agende, execute e exclua fluxos de trabalho.
<b>Ações</b>	Crie, edite e exclua ações. O editor de ações oferece suporte à conclusão automática para elementos de script comuns incluídos no API Explorer do vRealize Orchestrator.
<b>Políticas</b>	Crie, edite, execute e exclua as políticas.
<b>Pacotes</b>	Crie, exclua, exporte e importe pacotes que contenham objetos do vRealize Orchestrator.
<b>Configurações</b>	Crie, execute e exclua elementos de configuração.
<b>Recursos</b>	Exporte, importe e atualize elementos de recurso.
<b>Grupos</b>	Os usuários com direitos de administrador podem atribuir funções aos usuários no vRealize Orchestrator Client e adicioná-los aos grupos.
<b>Logs de Auditoria</b>	Exiba os diferentes eventos, como quando um objeto é criado, que são registrados no vRealize Orchestrator Client.
<b>Repositórios do Git</b>	Crie uma integração com um repositório do Git e use a integração para gerenciar o desenvolvimento de fluxos de trabalho e outros objetos do vRealize Orchestrator em várias implantações.  Consulte o <a href="#">Como posso usar a ramificação do Git para gerenciar o inventário de objeto do meu vRealize Orchestrator</a> .
<b>Itens Excluídos</b>	Restaure objetos do vRealize Orchestrator Client excluídos, como fluxos de trabalho, ações, políticas, elementos de configuração e elementos de recurso.
<b>API Explorer</b>	Explore os comandos de API disponíveis no vRealize Orchestrator Client.  <b>Observação</b> O vRealize Orchestrator Client se comunica com a REST API do vRealize Orchestrator por meio de um proxy REST.

Este capítulo inclui os seguintes tópicos:

- [Painel de uso do Cliente vRealize Orchestrator](#)

## ■ Organização de conteúdo no vRealize Orchestrator Client

# Painel de uso do Cliente vRealize Orchestrator

O painel do vRealize Orchestrator Client fornece uma ferramenta útil para monitoramento, gerenciamento e solução de problemas dos fluxos de trabalho do vRealize Orchestrator Client.

As informações sobre o painel do vRealize Orchestrator Client se espalham entre cinco painéis.

Janela	Descrição
Execuções de fluxo de trabalho	Fornecer dados visuais sobre o número de execuções de fluxo de trabalho em execução, em espera e com falha.
Fluxos de trabalho favoritos	Exibe os fluxos de trabalho adicionados aos favoritos.
Aguardando entrada	Exibe as execuções pendentes do fluxo de trabalho que exigem maior interação do usuário. Esses fluxos de trabalho também são exibidos no menu de notificações no canto superior direito da interface do usuário.
Execuções de fluxo de trabalho recentes	Gerencie as execuções de fluxo de trabalho recentes. Mostra o nome, o estado, a data de início e a data de término da execução do fluxo de trabalho.
Exigindo atenção	Exibe as execuções de fluxo de trabalho com falha e as métricas de desempenho de execução do fluxo de trabalho.

## Organização de conteúdo no vRealize Orchestrator Client

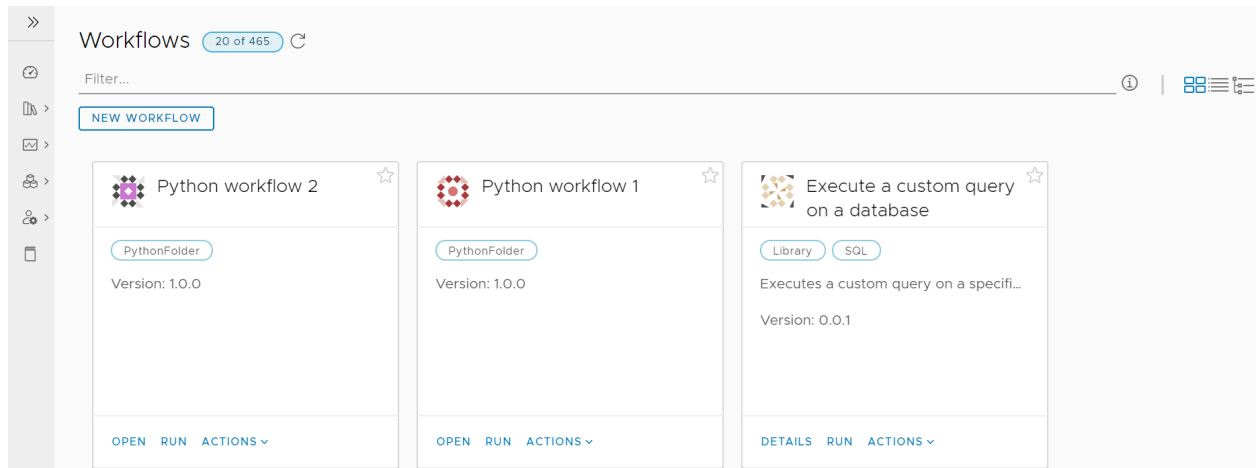
Gerencie como o seu inventário do objeto de vRealize Orchestrator é exibido no vRealize Orchestrator Client.

O vRealize Orchestrator Client suporta três tipos diferentes de exibição para objetos como fluxos de trabalho, ações, políticas, recursos e configurações: Exibição em Cartão, Exibição em Lista e Exibição em Árvore. Você pode alterar o tipo de visualização atual no canto superior direito da página.

### Exibição em Cartão

A Exibição em Cartão é o tipo de visualização padrão usado no vRealize Orchestrator Client. As informações sobre o objeto de inventário individual, como um fluxo de trabalho, são exibidas em um elemento de cartão separado.





## Exibição em Lista

A Exibição em Lista mostra informações sobre seus objetos de vRealize Orchestrator organizados como uma lista. Para obter mais informações sobre as ações que você pode executar no objeto, clique no ícone de reticências verticais à esquerda do objeto.

	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.

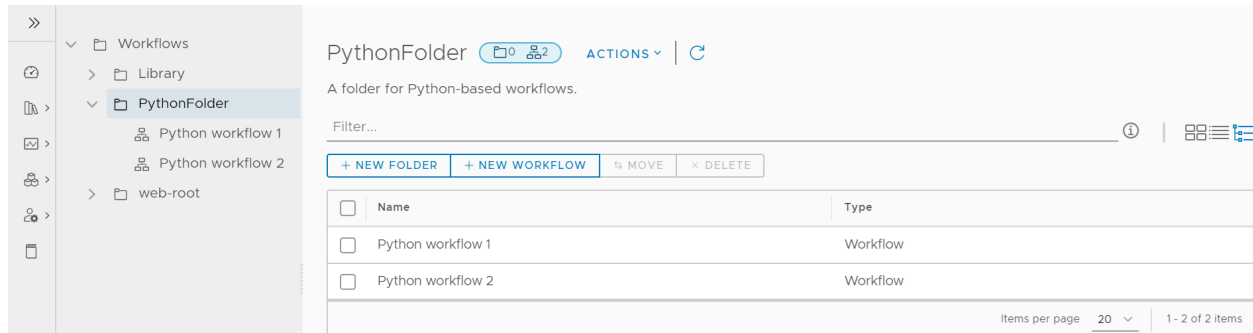
## Exibição em Árvore

Você pode organizar seu inventário de objeto em pastas hierárquicas no modo de Exibição em Árvore. Cada tipo de objeto vRealize Orchestrator tem uma pasta de nível raiz. Não é possível criar novos objetos, como fluxos de trabalho, sob a pasta raiz. Você deve criar pastas separadas organizadas sob a pasta raiz. Cada pasta inclui ferramentas para ajudá-lo a gerenciar seu conteúdo, como um filtro de conteúdo.

**Observação** Cada pasta tem um filtro de conteúdo separado. Você não pode filtrar conteúdo em pastas.

Para obter mais informações sobre pastas, consulte [Criar pasta ou subpasta no Cliente vRealize Orchestrator](#).

**Observação** Quando você seleciona um objeto na Exibição em Árvore, ele abre em modo somente leitura. Para editar o conteúdo do objeto, como variáveis do fluxo de trabalho ou o esquema do fluxo de trabalho, clique em **Editar** no menu de opções superior.




## Criar pasta ou subpasta no Cliente vRealize Orchestrator

Organize os objetos do vRealize Orchestrator usando uma estrutura de pastas hierárquica.

Você pode criar pastas e subpastas para organizar os seguintes tipos de objetos do vRealize Orchestrator:

- Fluxos de trabalho
- Ações
- Políticas
- Elementos de configuração
- Elementos de recurso

### Procedimentos


- 1 Faça login no vRealize Orchestrator Client.
- 2 No painel de navegação à esquerda, selecione uma página de objeto, como **Fluxos de Trabalho**.
- 3 No canto superior direito, selecione o ícone de exibição em árvore ()
- 4 (Opcional) Para criar uma subpasta, selecione uma pasta principal no modo de exibição de árvore à esquerda.
- 5 Clique em **Nova Pasta**.
- 6 Insira um nome e descrição e clique em **Salvar**.
- 7 Adicione objetos ou subpastas à pasta que acabou de ser criada.
- 8 (Opcional) Para editar o nome da pasta, selecione **Ações > Editar**.

## Mover objetos e pastas no vRealize Orchestrator Client

Reorganize o conteúdo do vRealize Orchestrator movendo o conteúdo para outra pasta.

Não é possível mover ações entre os módulos de ação ou mover qualquer objeto para uma pasta raiz. A pasta raiz inclui as pastas e subpastas principais do objeto, mas não pode ser usada para armazenar objetos.

### Procedimentos

- 1 Faça login no vRealize Orchestrator Client.
- 2 No painel de navegação à esquerda, selecione uma página de objeto, como **Fluxos de Trabalho**.
- 3 No canto superior direito, selecione o ícone de exibição em árvore ()
- 4 Expanda o modo de exibição de árvore e selecione o objeto ou a pasta que você deseja mover.
- 5 Arraste o objeto ou a pasta para a sua nova pasta principal.

---

**Observação** Você também pode mover objetos para novas pastas diretamente do editor de objetos. Na guia **Resumo**, clique em **Selecionar Pasta** e selecione a nova pasta principal para o objeto. Outra opção de movimento é selecionar objetos da tabela na página da pasta. Essa opção é útil para realizar operações de movimentação em lote que incluem vários objetos do vRealize Orchestrator.


---

## Excluir uma pasta ou subpasta no vRealize Orchestrator Client

Exclua pastas ou subpastas obsoletas do seu vRealize Orchestrator Client.

Não é possível excluir a pasta de nível raiz correspondente de cada tipo de objeto do vRealize Orchestrator.

### Procedimentos

- 1 Faça login no vRealize Orchestrator Client.
- 2 No painel de navegação à esquerda, selecione uma página de objeto, como **Fluxos de Trabalho**.
- 3 No canto superior direito, selecione o ícone de exibição em árvore ()
- 4 Marque a caixa de seleção ao lado da pasta que você deseja excluir.

---

**Observação** Para excluir uma subpasta, selecione a pasta principal na exibição em árvore e, em seguida, marque a caixa de seleção.

---

- 5 Clique em **Excluir**.

- 6 Se a pasta selecionada estiver vazia.
  - a Confirme se você deseja excluir a pasta.
  - b Clique em **Excluir**.
- 7 Se a pasta selecionada contiver objetos ou subpastas do vRealize Orchestrator Client.
  - a Confirme se você deseja excluir a pasta.
  - b Clique em **Excluir**.

Você recebe a mensagem Não foi possível excluir o item  
'your\_folder\_name': a pasta 'your\_folder\_name' não está vazia.
  - c Para excluir a pasta e todo o seu conteúdo, clique em **Impor exclusão**.
  - d Confirme se você deseja excluir a pasta e clique em **Excluir**.

---

**Observação** Você também pode realizar uma exclusão em lote selecionando vários objetos na tabela incluída no menu pasta.

---

# Configurar o vRealize Orchestrator Client

## 3

Para tirar proveito total da funcionalidade do vRealize Orchestrator Client, você deve configurar suas permissões de usuário e aprender como você pode usar o histórico de versões para gerenciar seus objetos.

Este capítulo inclui os seguintes tópicos:

- [Funções e grupos do vRealize Orchestrator](#)
- [Histórico da versão do objeto vRealize Orchestrator](#)

## Funções e grupos do vRealize Orchestrator

Os administradores do vRealize Orchestrator podem definir permissões que controlam o acesso a recursos e conteúdo no vRealize Orchestrator Client. Os direitos de acesso são separados em funções de usuário e permissões de grupo.

As funções controlam quais recursos do vRealize Orchestrator Client os usuários podem visualizar e usar. O acesso à funcionalidade de gerenciamento de função depende do tipo de licença do seu ambiente do vRealize Orchestrator.

**Tabela 3-1. Acesso com base em licença para o gerenciamento de funções do vRealize Orchestrator**

Licença	Autenticação	
	vSphere	vRealize Automation
<b>vSphere</b>	Não há suporte para o gerenciamento de função. Os grupos dão suporte apenas para permissões de execução.	
<b>vRealize Automation</b>	Gerencie funções no Cliente vRealize Orchestrator. Consulte o <a href="#">Atribuir funções no vRealize Orchestrator Client</a> .	Gerencie as funções por meio de <b>Gerenciamento de identidade e acesso</b> no vRealize Automation. Consulte o <a href="#">Configurar funções do Cliente vRealize Orchestrator no vRealize Automation</a> .

As permissões de grupo controlam o que os usuários de conteúdo do vRealize Orchestrator Client podem exibir e usar, como fluxos de trabalho, ações, políticas, elementos de configuração e elementos de recurso. O acesso a conteúdo pré-configurado do sistema vRealize Orchestrator, como fluxos de trabalho e ações padrão, é compartilhado entre todos os usuários, a menos que seja configurado de outra forma por meio de permissões do grupo.

Os direitos de acesso de usuários com funções de administrador e visualizador não são restritos por permissões de grupo. Os direitos de acesso de usuários sem uma função atribuída e os usuários com uma função de designer de fluxo de trabalho dependem do grupo atribuído a eles. Você pode estender os direitos de acesso desses usuários modificando suas permissões de grupo. Dessa forma, você pode organizar os usuários em projetos comuns. Por exemplo, você pode criar um grupo que inclua os usuários que trabalham no desenvolvimento de um plug-in personalizado do vRealize Orchestrator e permitir que eles modifiquem somente o conteúdo que é específico para o seu grupo.

**Tabela 3-2. Permissões para grupos e funções de usuário do vRealize Orchestrator**

Função	Direitos de acesso	
Administrador	<p>Os administradores podem acessar todos os recursos e conteúdo do Cliente vRealize Orchestrator incluindo o conteúdo criado por grupos específicos. Responsável pela configuração de funções do usuário, criação e exclusão de grupos e inclusão de usuários em grupos. Os administradores não são limitados pelas permissões de grupo.</p> <p><b>Observação</b> Os administradores de tenant de ambientes do vRealize Automation usados para autenticar o vRealize Orchestrator têm direitos de <b>Administrador</b> por padrão.</p>	
Espectador	<p>Os visualizadores têm acesso somente leitura a todo o conteúdo no Cliente vRealize Orchestrator, mas não podem criar, editar, executar ou exportar conteúdo. Os visualizadores também podem ver todos os grupos e o conteúdo dos grupos. Os visualizadores não são limitados por permissões de grupo.</p>	
Permissões de Grupo		
Nenhum grupo atribuído	Executar	Executar e editar

Tabela 3-2. Permissões para grupos e funções de usuário do vRealize Orchestrator (continuação)

Função	Direitos de acesso		
Designer de fluxo de trabalho	<ul style="list-style-type: none"> <li>■ Visualize o conteúdo do sistema.</li> <li>■ Visualize e execute suas próprias execuções.</li> <li>■ Crie, execute, edite e exclua seu próprio conteúdo.</li> </ul>	<ul style="list-style-type: none"> <li>■ Visualizar conteúdo do sistema</li> <li>■ Visualize e execute suas próprias execuções.</li> <li>■ Crie, execute, edite e exclua seu próprio conteúdo.</li> <li>■ Adicione conteúdo próprio ao grupo.</li> <li>■ Execute o conteúdo do grupo, mas não pode editá-lo.</li> </ul>	<ul style="list-style-type: none"> <li>■ Visualize o conteúdo do sistema.</li> <li>■ Visualize e execute suas próprias execuções.</li> <li>■ Crie, execute, edite e exclua seu próprio conteúdo.</li> <li>■ Adicione conteúdo próprio ao grupo.</li> <li>■ Execute e edite o conteúdo do grupo.</li> </ul>
	<p><b>Observação</b> Não disponível para instâncias do vRealize Orchestrator autenticadas com o vSphere.</p>		
Usuário sem uma função atribuída	<ul style="list-style-type: none"> <li>■ Visualize as próprias execuções.</li> </ul>	<ul style="list-style-type: none"> <li>■ Visualize e execute suas próprias execuções.</li> <li>■ Visualize e execute o conteúdo do grupo.</li> </ul>	<ul style="list-style-type: none"> <li>■ Visualize e execute suas próprias execuções.</li> <li>■ Visualize e execute o conteúdo do grupo.</li> </ul>
	<p><b>Observação</b> Para poder criar, editar e adicionar conteúdo, os usuários desse grupo devem receber uma função de designer de fluxo de trabalho.</p> <p><b>Observação</b> Não disponível para instâncias do vRealize Orchestrator autenticadas com o vSphere.</p>		

## Atribuir funções no vRealize Orchestrator Client

Como administrador, você pode adicionar usuários ao vRealize Orchestrator Client e definir quais recursos eles podem exibir e usar.

O gerenciamento de funções controla o acesso dos usuários do provedor de identidade do vRealize Orchestrator aos recursos do vRealize Orchestrator Client. O gerenciamento de funções abrange a interface do usuário do vRealize Orchestrator Client e a funcionalidade da API.

**Observação** O gerenciamento de funções do lado do cliente está disponível apenas para instâncias autenticadas do vRealize Orchestrator com o vSphere que usam uma licença do vRealize Automation. Para obter informações sobre como atribuir funções autenticadas do vRealize Orchestrator com o vRealize Automation, consulte [Configurar funções do Cliente vRealize Orchestrator no vRealize Automation](#).

## Procedimentos

- 1 Faça login no Cliente vRealize Orchestrator como administrador.
- 2 Vá para **Administração > Gerenciamento de Funções**.
- 3 Clique em **Adicionar**.
- 4 Procure o usuário ou grupo que você deseja adicionar ao vRealize Orchestrator Client.
- 5 Selecione a função do usuário. Para obter mais informações sobre funções, consulte [Funções e grupos do vRealize Orchestrator](#).
- 6 Clique em **Salvar**.

## Configurar funções do Cliente vRealize Orchestrator no vRealize Automation

Você pode atribuir funções de serviço para o vRealize Orchestrator Client na página **Gerenciamento de Identidade e Acesso** no vRealize Automation. As funções de serviço podem ser atribuídas às instâncias incorporadas do vRealize Orchestrator Client e autônomas do vRealize Orchestrator autenticadas com o vRealize Automation.

As funções de serviço do vRealize Orchestrator gerenciam quais recursos os usuários incorporados do vRealize Orchestrator Client podem acessar. Para obter mais informações sobre as funções do vRealize Orchestrator, consulte [Funções e grupos do vRealize Orchestrator](#).

---

**Observação** As instâncias autônomas do vRealize Orchestrator autenticadas com o vSphere que usam uma licença do vRealize Automation podem atribuir as funções diretamente no vRealize Orchestrator Client. Consulte o [Atribuir funções no vRealize Orchestrator Client](#).

---

## Pré-requisitos

- Verifique se os usuários e grupos apropriados são importados de uma instância válida do vIDM.
- Para atribuir uma função de serviço do vRealize Orchestrator ao seu usuário, verifique se o usuário tem uma função de organização atribuída no vRealize Automation. Consulte *Administrar usuários e grupos no vRealize Automation* em *Administrar vRealize Automation*.

## Procedimentos

- 1 No menu suspenso do cabeçalho superior à direita, selecione a opção **Gerenciamento de Identidade e Acesso**.
- 2 Na guia **Usuários Ativos**, procure o endereço de e-mail do usuário que você deseja atribuir ao vRealize Orchestrator.
- 3 Marque a caixa de seleção ao lado do usuário e clique em **Editar Funções**.
- 4 Clique em **Adicionar Acesso de Serviço**.
- 5 No menu suspenso à esquerda, selecione **Orchestrator**.



- 6 No menu suspenso à direita, selecione a função que você deseja atribuir ao usuário.
- 7 Clique em **Salvar**.

## Criar grupos no vRealize Orchestrator Client

Como administrador, você pode usar grupos para definir qual conteúdo do vRealize Orchestrator os usuários podem visualizar e acessar no vRealize Orchestrator Client.

Você pode usar o vRealize Orchestrator Client para definir permissões de grupo para fluxos de trabalho, ações, políticas, elementos de configuração, elementos de recurso e pacotes do vRealize Orchestrator.

---

**Observação** Os usuários de instâncias autenticadas do vRealize Orchestrator com o vSphere, só podem ter permissões de grupo **Executar**.

---

### Procedimentos

- 1 Faça login no Cliente vRealize Orchestrator como administrador.
- 2 Vá para **Administração > Grupos**.
- 3 Clique em **Novo Grupo**.
- 4 Na guia **Resumo**, adicione um nome e uma descrição para o grupo.
- 5 Na guia **Usuários**, clique em **Adicionar**.
  - a Procure um usuário que você deseja adicionar ao grupo.
  - b Atribua permissões de grupo ao usuário.
  - c Clique em **Adicionar**.
- 6 Na guia **Itens**, adicione objetos do vRealize Orchestrator ao grupo.

---

**Observação** Você também pode adicionar um objeto aos grupos existentes quando esse objeto estiver sendo criado no vRealize Orchestrator Client. Para adicionar o objeto, selecione o grupo no menu suspenso **Acessível por** na guia **Resumo/Geral** do editor de objetos.

---

- 7 Clique em **Salvar**.

## Histórico da versão do objeto vRealize Orchestrator

O vRealize Orchestrator Client mantém um registro de histórico de versão para cada objeto do vRealize Orchestrator. Usando o histórico de versões, você pode comparar diferentes versões de objetos do vRealize Orchestrator e reverter para uma versão anterior.

O vRealize Orchestrator cria um registro de histórico de versão para cada objeto do vRealize Orchestrator quando você salva o objeto. Alterações subsequentes nos objetos do vRealize Orchestrator criam um novo registro de histórico de versão. Os registros de histórico de versões anteriores são preservados e podem ser usados para rastrear alterações no objeto e reverter o objeto para uma versão anterior. A reversão de um objeto para uma versão anterior cria um novo registro de histórico de versão.

O vRealize Orchestrator Client rastreia o histórico de versões dos seguintes objetos do vRealize Orchestrator:

- Fluxos de trabalho
- Ações
- Pacotes
- Políticas
- Elementos de recurso
- Elementos de configuração

---

**Observação** Os fluxos de trabalho gerados não são exibidos no histórico de versões do fluxo de trabalho. Por exemplo, os fluxos de trabalho gerados pelo fluxo de trabalho **Gerar fluxos de trabalho CRUD para uma tabela** não aparecem na guia **Histórico de Versões** e não podem ser enviados para nenhum repositório Git configurado. Para incluir esses fluxos de trabalho no histórico de versões do vRealize Orchestrator, duplique os fluxos de trabalho gerados.

---

Você pode acessar o histórico de versões de um objeto na guia **Histórico de versões** da página do editor de objetos. Se você estiver tentando editar um objeto ao mesmo tempo que outro usuário, poderá ocorrer um conflito de mesclagem. Para resolver o conflito de mesclagem, clique em **Resolver** à direita da mensagem de erro. Na janela **Resolver conflitos**, você tem três opções:

- **Usar os deles.** Resolva o conflito de mesclagem usando as alterações feitas pelo outro usuário.
- **Usar os nossos.** Resolva o conflito de mesclagem usando suas alterações.
- **Resolver.** Resolva o conflito de mesclagem editando o modelo de alteração exibido. Se o modelo fornecido for inválido, essa opção não estará disponível.

## Restaurar um fluxo de trabalho para uma versão anterior

Você pode restaurar um fluxo de trabalho para uma versão salva anteriormente.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Biblioteca > Fluxos de Trabalho** e selecione um fluxo de trabalho.
- 3 Selecione a guia **Histórico de versões**.

- 4 Para exibir uma comparação entre as versões, selecione uma versão do fluxo de trabalho e selecione outra versão no menu suspenso **Diferença com**.

Uma janela exibe as diferenças entre a versão atual e a versão selecionada do fluxo de trabalho.

- 5 Para restaurar o fluxo de trabalho para outra versão, clique em **Restaurar**.

O estado do fluxo de trabalho é revertido para o estado da versão selecionada.

---

**Observação** Você também pode restaurar uma versão do fluxo de trabalho a partir do modo de exibição de ferramenta de diferença gráfica. Consulte o [Comparação visual entre versões do fluxo de trabalho](#).

---

## Comparação visual entre versões do fluxo de trabalho

Compare as alterações entre versões do fluxo de trabalho com a ferramenta de diferença gráfica.

Por padrão, o histórico da versão vRealize Orchestrator exibe as diferenças entre as versões do fluxo de trabalho em um formulário YAML. Você também pode realizar uma comparação visual entre diferentes versões do fluxo de trabalho. Você pode visualizar as alterações em:

- As informações gerais do fluxo de trabalho, como número da versão e descrição do fluxo de trabalho.
- As variáveis usadas no fluxo de trabalho.
- Os parâmetros de entrada e saída do fluxo de trabalho.
- O esquema do fluxo de trabalho.

### Pré-requisitos

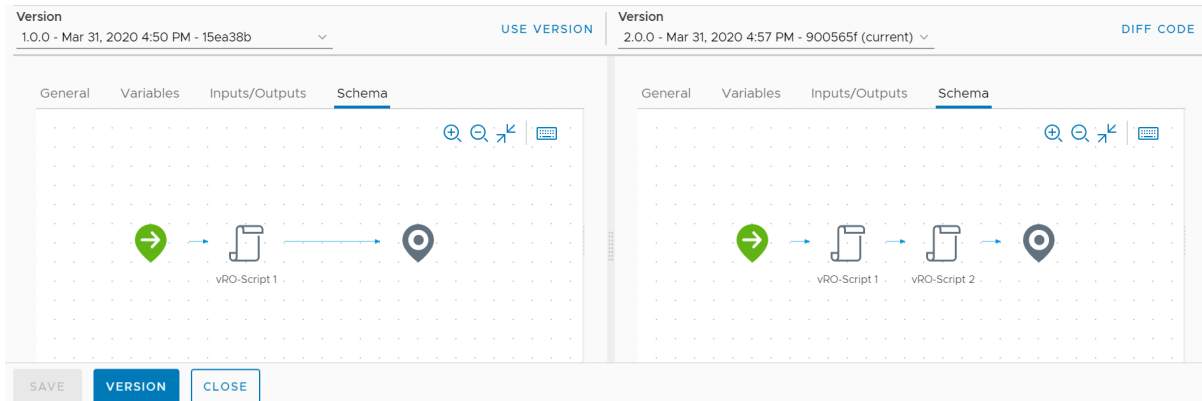
Crie um fluxo de trabalho.

### Procedimentos

- 1 Faça login no vRealize Orchestrator Client.
- 2 Vá para **Biblioteca > Fluxo de trabalho**, e selecione um dos seus fluxos de trabalho.
- 3 Edite o conteúdo do fluxo de trabalho.  
  
Por exemplo, você pode adicionar um elemento extra da **Tarefa programável por script** na guia **Esquema**.
- 4 Clique em **Salvar**.
- 5 Selecione a guia **Histórico de versões**.

## 6 No canto superior direito, selecione **Diferença Visualmente**.

Agora você pode executar uma comparação visual entre duas versões de fluxo de trabalho selecionadas. Você pode selecionar quais versões devem ser comparadas a partir do menu suspenso **Versão**.



## 7 (Opcional) Você pode restaurar um fluxo de trabalho para outra versão selecionando **Usar Versão**.

# Redefinir inventário de conteúdo do vRealize Orchestrator para um estado anterior com o Git

Usando uma confirmação anterior do Git, você pode redefinir o conteúdo do vRealize Orchestrator para um estado anterior.

Você pode redefinir o conteúdo do seu vRealize Orchestrator para um estado anterior selecionando uma confirmação específica.

### Pré-requisitos

- Configure uma conexão com um repositório GitHub ou GitLab. Consulte o [Configurar uma conexão para um Repositório do Git](#).
- Envie um conjunto de alterações local para o repositório do Git configurado.

### Procedimentos

- 1 Faça login no vRealize Orchestrator Client.
- 2 Vá até **Administração > Histórico de Git**.
- 3 Selecione um conjunto de alterações que você deseja redefinir e clique em **Redefinir para este**.
- 4 Confirme que você deseja redefinir para essa confirmação específica e clique em **OK**.

O inventário de conteúdo do vRealize Orchestrator é redefinido para o estado especificado na confirmação. O conteúdo relevante do vRealize Orchestrator é revertido para uma versão anterior. Se o conteúdo não existisse quando a confirmação foi enviada, ela será removida do inventário.

### Próximo passo

Para restaurar o inventário do vRealize Orchestrator para o estado mais recente salvo no repositório do Git, execute um comando `Pull` na janela **Histórico de Git**.

# Casos de uso do vRealize Orchestrator

# 4

Esses casos de uso demonstram uma parte da funcionalidade da plataforma do vRealize Orchestrator.

Esses casos de uso apresentam apenas valores de exemplo. Sua própria estrutura de ambiente e convenções de nomenclatura podem variar.

Este capítulo inclui os seguintes tópicos:

- [Como integrar os Amazon Web Services no vRealize Orchestrator usando Python](#)
- [Como posso usar a ramificação do Git para gerenciar o inventário de objeto do meu vRealize Orchestrator](#)
- [Como posso usar módulos de terceiros para chamar a API do projeto do vRealize Automation](#)

## Como integrar os Amazon Web Services no vRealize Orchestrator usando Python

Esse caso de uso do vRealize Orchestrator mostra um exemplo de como você pode usar o Python para expandir os recursos da sua implantação do vRealize Orchestrator.

É possível usar os seguintes tempos de execução nos scripts de ação e fluxo de trabalho:

- Python 3,7
- Node.js 12
- PowerCLI 11/Powershell 6.2
- PowerCLI 12/Powershell 7

---

**Observação** O tempo de execução do PowerCLI inclui o PowerShell e os seguintes módulos: VMware.PowerCLI, PowerNSX, PowervRA.

---

---

**Importante** Você só poderá usar os novos tempos de execução se a sua implantação do vRealize Orchestrator usar uma licença do vRealize Automation.

---

Esse caso de uso demonstra como você pode criar um script Python que chama instâncias do EC2 no Amazon Web Services (AWS).

---

**Importante** Para desenvolver seu script personalizado, verifique se você está familiarizado com os principais conceitos de utilização dos scripts Python, Node.js e PowerShell, no vRealize Orchestrator. Consulte o [Conceitos básicos para scripts do Python, Node.js e PowerShell](#).

---

## Procedimentos

### 1 Criar script do Python inicial

Na sua máquina local, crie seu script do Python e empacote o script e uma biblioteca boto3 como uma pasta ZIP.

### 2 Criar a ação do Amazon Web Services

Crie uma ação do vRealize Orchestrator para usar o seu script do Python.

### 3 Depurar a ação do Amazon Web Services

A versão original do script do Python tem um erro interno deliberado, para que você possa aprender a depurar o script.

### 4 Atualizar a ação do Amazon Web Services

Importe o script Python atualizado e execute a ação novamente.

## Criar script do Python inicial

Na sua máquina local, crie seu script do Python e empacote o script e uma biblioteca boto3 como uma pasta ZIP.

### Pré-requisitos

- Baixe e instale o Python 3. Consulte a [página de downloads do Python](#).
- Baixe e instale o Visual Studio Code. Consulte a [página de download do Visual Studio Code](#).
- Verifique se você instalou a extensão do Python para o Visual Studio Code. Consulte o [Visual Studio Marketplace](#).

### Procedimentos

#### 1 Na sua máquina local, crie uma pasta vro-python-aws e instale o boto3 Python SDK nela.

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

#### 2 Abra um editor e crie o script do Python principal. Para esse caso de uso, você está usando o 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)
```

Este script do Python lista todas as instâncias do EC2 em execução em uma determinada região.

- 3 Salve o script criado como um arquivo `main.py` na pasta `vro-python-aws`.
- 4 Efetue login em uma interface da linha de comandos.
- 5 Vá para a pasta `vro-python-aws`.

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

- 6 Crie um pacote ZIP que contenha o script do Python.

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

---

**Observação** Você também pode criar o pacote ZIP usando uma ferramenta de utilitário ZIP, como 7-Zip.

---

## Resultados

Você criou a base do script do Python e a preparou para importação na implantação do vRealize Orchestrator.

## Criar a ação do Amazon Web Services

Crie uma ação do vRealize Orchestrator para usar o seu script do Python.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Biblioteca > Ações**.
- 3 Clique em **Nova Ação**.
- 4 Na guia **Geral**, insira um nome, um módulo e um número de versão para a ação.
- 5 Na guia **Script**, selecione **Python 3.7** como o tempo de execução e **Zip** como o tipo de script.
- 6 Clique em **Importar**.
- 7 Vá para a pasta `vro-python-aws` e selecione o pacote ZIP que contém o script do Python.



- 8 Na caixa de texto **Manipulador de entrada**, insira `main.handler`.

---

**Observação** O manipulador de entrada da ação é baseado no script principal no pacote ZIP importado. Como o script principal está localizado em um arquivo chamado `main.py` e uma função chamada de `manipulador`, o manipulador de entrada deve ser `main.handler`. Se você tiver intitulado o arquivo de script principal de forma diferente, altere o valor do manipulador de entrada de acordo.

---

- 9 Salve a ação e clique em **Executar**.

A ação executada encontra um erro.

- 10 Selecione a guia **Logs**.

Os logs da ação executada exibem uma mensagem de erro

`"botocore.exceptions.NoRegionError: Você deve especificar uma região."`. Esse é um comportamento esperado, pois o script do Python inicial não define uma região.

#### Próximo passo

Depure o script do Python. Consulte o [Depurar a ação do Amazon Web Services](#).

## Depurar a ação do Amazon Web Services

A versão original do script do Python tem um erro interno deliberado, para que você possa aprender a depurar o script.

#### Pré-requisitos

Faça login na sua conta do Amazon Web Services (AWS) e crie um usuário IAM especificamente para esse cenário de caso de uso. Consulte [Criando um Usuário IAM na sua Conta do AWS](#). O usuário IAM deve ter as seguintes permissões:

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

```

## Procedimentos

### 1 Prepare o vRealize Orchestrator Appliance.

---

**Cuidado** Não depure scripts em sua implantação do vRealize Orchestrator de produção. Depure a partir de uma implantação do vRealize Orchestrator de um único nó, que você usa para desenvolvimento e teste.

---

- a Faça login na linha de comando do vRealize Orchestrator Appliance sobre o SSH como **raiz**.
- b Execute o comando `vracli dev tools`.
- c Você será solicitado a confirmar que deseja continuar. Insira **sim** para continuar ou **não** para cancelar.

---

**Importante** Ao executar o comando `vracli dev tools`, você abre as portas necessárias para depurar o script do Python. Você deve sair da sessão atual do SSH aberta durante o processo de depuração.

---

### 2 Inicie a configuração de depuração.

- a Faça login no Cliente do vRealize Orchestrator.
- b Abra a ação AWS e clique em **Depurar**.  
O processo de depuração começa e a execução da ação é suspensa.
- c Selecione a guia **Configuração de Depuração**.  
A guia contém uma configuração `.json` que você pode anexar remotamente ao IDE para depurar o script do Python.
- d Copie o conteúdo da configuração manualmente ou clique em **Copiar para a Área de Transferência**.

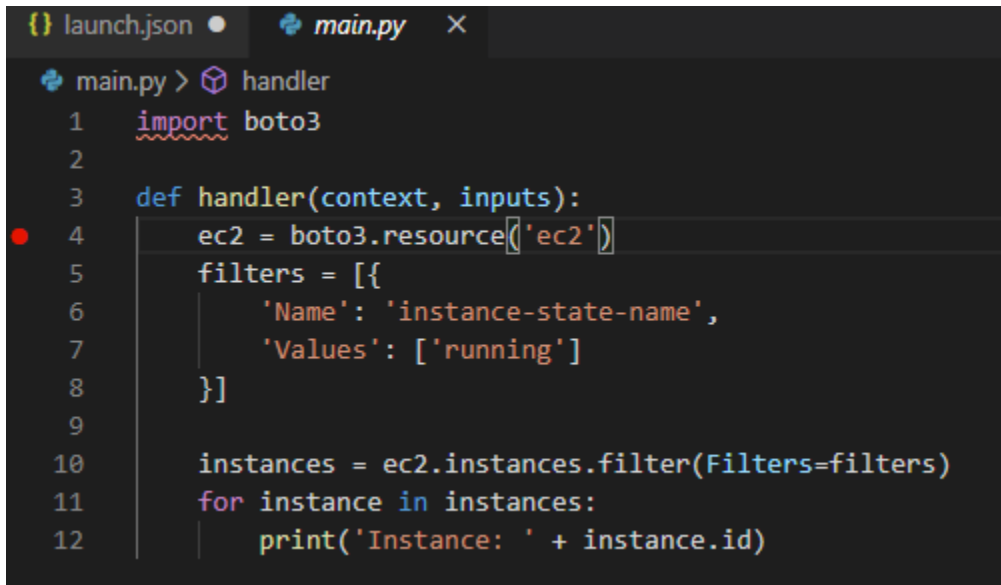
### 3 Depure o script do Python.

- a Abra o código do Visual Studio.
- b Abra a pasta `vro-python-aws`.
- c No painel de navegação superior, selecione **Executar > Abrir Configurações**.
- d Selecione **Arquivo Python**.

- e Deixe os atributos "version" e "configuration" em suas posições atuais e cole o conteúdo da configuração .json copiada do Cliente do vRealize Orchestrator. O arquivo gerado launch.json deve ser semelhante a este:

```
{
  "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 Selecione o arquivo de script main.py e adicione um ponto de interrupção à linha `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 No painel de navegação superior, selecione **Executar > Iniciar Depuração**.

- h Quando o depurador atinge o ponto de interrupção, realize uma operação de depuração. A execução de depuração indica que o script do Python não tem uma região especificada e a chave de acesso AWS.
- i Volte para a sessão do vRealize Orchestrator Appliance aberta e pressione **Enter** para fechar as portas que você abriu para esta sessão de depuração.

**4** Adicione as informações que estão faltando ao script do Python.

- a No código do Visual Studio, crie um arquivo chamado `awsconfig` que contém a chave de acesso AWS do usuário IAM e a região de AWS que você deseja executar ping com o script do Python.

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

- b Salve `awsconfig` como um arquivo de configuração (.cfg) na pasta `vro-python-aws`.
- c Abra o arquivo `main.py` e modifique-o para que a biblioteca boto3 possa usar o arquivo `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 Crie um novo pacote ZIP que contenha o arquivo `main.py`, `awsconfig.cfg` e a biblioteca boto3.

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

---

**Observação** Você também pode criar o pacote ZIP usando uma ferramenta de utilitário ZIP, como 7-Zip.

---

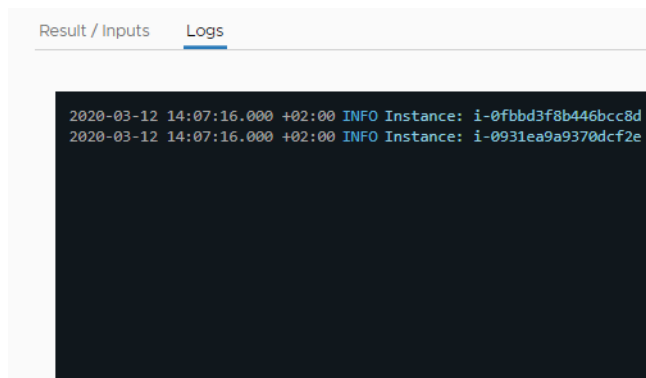
## Atualizar a ação do Amazon Web Services

Importe o script Python atualizado e execute a ação novamente.

## Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá até **Biblioteca > Ações** e selecione a ação original do Amazon Web Services (AWS).
- 3 (Opcional) Na guia **Geral**, altere o número da versão.
- 4 Remova o pacote ZIP antigo e clique em **Importar**.
- 5 Selecione o pacote ZIP atualizado.
- 6 Salve a ação e clique em **Executar**.
- 7 Após a execução da ação ser concluída, selecione a guia **Logs**.

Os logs exibem as instâncias do EC2 consultadas pela ação.



## Próximo passo

Crie um fluxo de trabalho do vRealize Orchestrator que usa a ação AWS atualizada como **Elemento de ação**.

## Como posso usar a ramificação do Git para gerenciar o inventário de objeto do meu vRealize Orchestrator

Use a ramificação para organizar como o conteúdo do vRealize Orchestrator é gerenciado no seu repositório do Git.

Ao usar o Git, você pode aumentar a flexibilidade dos seus desenvolvedores do vRealize Orchestrator fornecendo um repositório centralizado. Por exemplo, você pode usar o Git para gerenciar o desenvolvimento de fluxos de trabalho em vários ambientes do vRealize Orchestrator.

---

**Observação** Para usar o Git para gerenciar seu inventário de objetos, sua implantação do vRealize Orchestrator deve usar uma licença do vRealize Automation. Para obter mais informações, consulte *Ativação do recurso do vRealize Orchestrator com licenças em Instalar e configurar o vRealize Orchestrator*.

---

Agora você pode enviar e puxar objetos de e para ramificações. Você pode usar ramificação para gerenciar o desenvolvimento de grupos específicos de objetos do vRealize Orchestrator, antes que eles sejam mesclados novamente na sua ramificação principal.

Neste caso de uso, você está usando um projeto GitLab para gerenciar objetos do vRealize Orchestrator que usam o tempo de execução do Python. Esse caso de uso representa um exemplo da funcionalidade do Git no vRealize Orchestrator e não representa os limites do escopo do recurso.

---

**Observação** Se você estiver mais familiarizado com o GitHub, poderá usar um repositório GitHub para esse caso de uso.

---

## Procedimentos

### 1 Preparar o seu ambiente do GitLab

Crie uma ramificação Git para os seus objetos do vRealize Orchestrator Python.

### 2 Configurar uma conexão para um Repositório do Git

Como **administrador**, você pode configurar uma conexão entre sua implantação vRealize Orchestrator e um repositório ou projeto Git.

### 3 Enviar alterações para um repositório Git

Envie suas alterações para os objetos locais do vRealize Orchestrator para o seu repositório Git integrado. Para esse caso de uso, estamos colocando as alterações em uma ação de vRealize Orchestrator baseada em Python em uma ramificação específica do Git.

## Preparar o seu ambiente do GitLab

Crie uma ramificação Git para os seus objetos do vRealize Orchestrator Python.

### Pré-requisitos

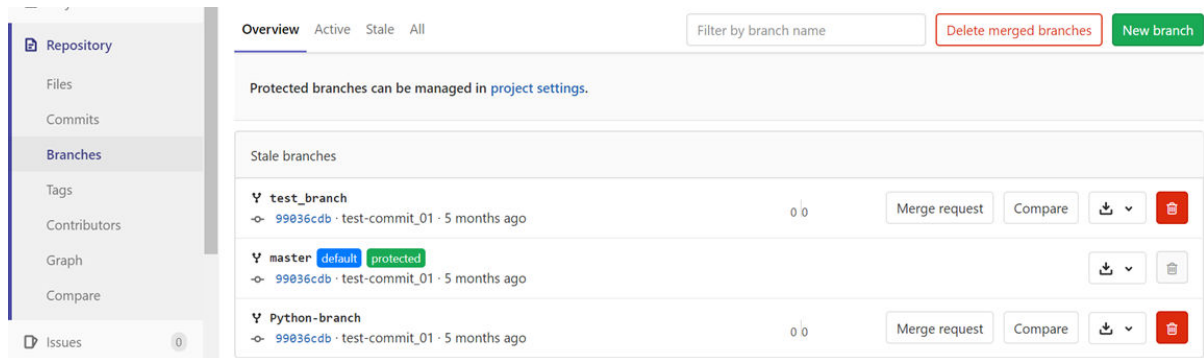
Crie um projeto do GitLab para o seu ambiente do vRealize Orchestrator. Consulte [Criar um projeto](#).

### Procedimentos

- 1 Faça login na sua conta do GitLab.
- 2 Vá para o seu projeto do GitLab.
- 3 No painel de navegação à esquerda, selecione **Repositório > Ramificações**.
- 4 Na guia **Visão geral**, clique em **Nova Ramificação**.
- 5 Em **Nome da ramificação**, insira **Python-branch**.
- 6 Deixe a opção **Criar de** como **mestre**.

## 7 Clique em **Criar ramificação**.

Você criou uma ramificação para os seus objetos do vRealize Orchestrator baseados em Python.



## Configurar uma conexão para um Repositório do Git

Como **administrador**, você pode configurar uma conexão entre sua implantação vRealize Orchestrator e um repositório ou projeto Git.

Para usar o Git para gerenciar seu inventário de objetos vRealize Orchestrator, você deve configurar uma conexão com seu repositório Git usando o vRealize Orchestrator Client.

### Pré-requisitos

- Verifique se o seu ambiente vRealize Orchestrator utiliza uma licença do vRealize Automation.
- Gere um token de acesso para o seu projeto GitLab e copie-o para a sua área de transferência para uso durante o processo de configuração. Consulte [Criando um token de acesso pessoal](#).

**Observação** Para esse caso de uso, você está usando um projeto GitLab. Se você estiver mais familiarizado com o GitHub, poderá usar um repositório do GitHub. Para obter informações que geram um token do GitHub, consulte [Criando um token de acesso pessoal para a linha de comando](#).

### Procedimentos

- 1 Efetue login no vRealize Orchestrator Client como um **administrador**.
- 2 Vá para **Administração > Repositórios do Git**.
- 3 Clique em **Adicionar Repositório**.
- 4 Insira o endereço da URL do seu repositório do Git.

Por exemplo, `https://gitlab.com/myusername/my-vro-repo`.

**Observação** Você também pode estabelecer uma conexão com o protocolo SSH.

- 5 Insira o nome de usuário para o seu perfil Git.
- 6 Insira o token de acesso do seu repositório do Git.

- 7 Para validar a conexão com o repositório do Git, clique em **Validar**.
- 8 (Opcional) Altere o nome usado para identificar o repositório no vRealize Orchestrator Client.
- 9 (Opcional) Adicione uma breve descrição para o repositório do Git conectado.
- 10 Para ativar o repositório do Git conectado, clique em **Tornar o repositório ativo**.

---

**Observação** Somente um repositório do Git pode estar ativo por vez. Você pode alterar o repositório do Git ativo na página **Repositórios do Git**.

---

- 11 Selecione a ramificação para a qual você deseja enviar suas alterações. Para esse caso de uso, você está usando a **ramificação Python**. Consulte o [Preparar o seu ambiente do GitLab](#).

---

**Observação** Você pode alterar a ramificação Git selecionada a qualquer momento após concluir a configuração inicial do Git.

---

- 12 Para concluir o processo de configuração, clique em **Salvar**.

#### Próximo passo

Volte para o menu **Repositórios do Git** e confirme se o status do repositório está **Ativo**.

## Enviar alterações para um repositório Git

Envie suas alterações para os objetos locais do vRealize Orchestrator para o seu repositório Git integrado. Para esse caso de uso, estamos colocando as alterações em uma ação de vRealize Orchestrator baseada em Python em uma ramificação específica do Git.

Você pode enviar um conjunto local de alterações para um repositório do Git. Cada conjunto de alterações pode consistir em um ou mais objetos modificados do vRealize Orchestrator.

---

**Observação** O processo de enviar e descartar conjuntos de alterações para um repositório do Git não é limitado por permissões do grupo. Portanto, um desenvolvedor de fluxo de trabalho de um grupo pode enviar ou descartar as alterações locais feitas por outro desenvolvedor.

---

#### Pré-requisitos

- Verifique se você criou uma ramificação Git. Consulte o [Preparar o seu ambiente do GitLab](#).
- Verifique se você configurou uma conexão com um repositório do Git. Consulte o [Configurar uma conexão para um Repositório do Git](#).
- Verifique se a sua integração do Git está configurada para enviar alterações para a ramificação Python **Python-branch** do Git.
- Crie um objeto do vRealize Orchestrator baseado em Python. Por exemplo, consulte [Como integrar os Amazon Web Services no vRealize Orchestrator usando Python](#).

#### Procedimentos

- 1 Faça login no vRealize Orchestrator Client.



## 2 Edite a ação do Python.

- Vá para **Biblioteca > Ações** e selecione sua ação do Python.
- Faça algumas pequenas alterações na ação, como alterar a descrição.
- Salve a ação.

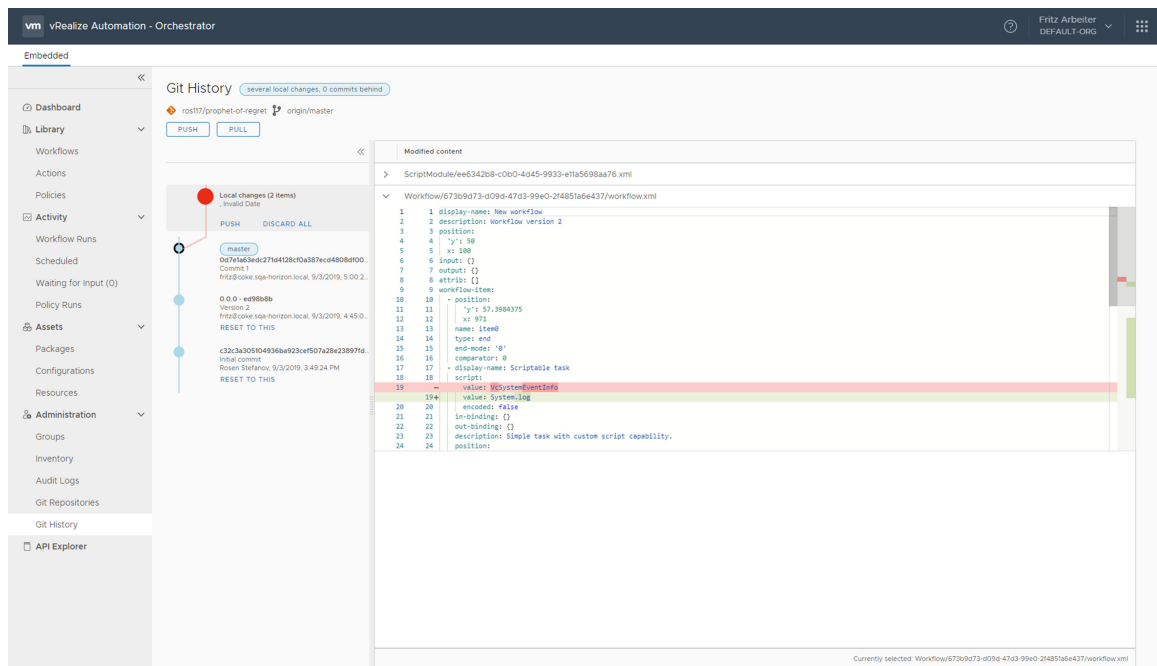
## 3 Envie suas alterações para o repositório do Git.

**Observação** Você também pode enviar alterações locais em um nível por objeto clicando na opção **Versão** exibida na parte inferior do editor de objetos.

- Vá até **Administração > Histórico de Git**.

O **Histórico de Git** exibe as diferenças atuais entre a ramificação da versão local e a ramificação selecionada do repositório do Git. Você pode expandir a entrada para qualquer objeto modificado do vRealize Orchestrator para exibir as diferenças de versão.

**Observação** Você pode descartar um conjunto de alterações local selecionando **Descartar tudo**.



- Clique em **Enviar**.
- Insira um título de confirmação.
- (Opcional) Insira uma rápida descrição para o compromisso.
- Selecione as alterações na ação do Python que você deseja enviar para o repositório do Git.

## 4 Para concluir o envio do conjunto de alterações locais para o repositório do Git, clique em **Enviar**.

### Próximo passo

Depois de concluir o desenvolvimento na sua ramificação do Git, mescle-a com a ramificação principal. Consulte [Como criar uma solicitação de mesclagem](#).

## Como posso usar módulos de terceiros para chamar a API do projeto do vRealize Automation

Esse caso de uso do vRealize Orchestrator demonstra como você pode chamar a API do Projeto do vRealize Automation usando módulos de terceiros.

É possível usar os seguintes tempos de execução nos scripts de ação e fluxo de trabalho:

- Python 3,7
- Node.js 12
- PowerCLI 11/Powershell 6.2
- PowerCLI 12/Powershell 7

---

**Observação** O tempo de execução do PowerCLI inclui o PowerShell e os seguintes módulos: VMware.PowerCLI, PowerNSX, PowervRA.

---

Nesse caso de uso, você aprende a criar ações do vRealize Orchestrator que usam módulos de dependência de terceiros para se conectar à API do Projeto do vRealize Automation.

---

**Importante** Para desenvolver seu script personalizado, verifique se você está familiarizado com os principais conceitos de utilização dos scripts Python, Node.js e PowerShell, no vRealize Orchestrator. Consulte o [Conceitos básicos para scripts do Python, Node.js e PowerShell](#).

---

## Criar um script do Python que chama a API de Projeto do vRealize Automation

Crie um script de amostra que usa o Python para chamar a API de Projeto do vRealize Automation.

### Pré-requisitos

Verifique se você instalou o Python 3 e o instalador de pacotes PIP. Consulte a [página Downloads do Python](#) e o [Índice de Pacote do Python](#).

### Procedimentos

- 1 Na sua máquina local, abra um shell de linha de comando.
- 2 Crie uma pasta `vro-python-vra`.

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

**3** Vá para a pasta `vro-python-vra`.

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

**4** Crie um script do Python chamado `handler.py`.

```
touch handler.py
```

O script `handler.py` deve definir uma função que aceite dois argumentos, o contexto do fluxo de trabalho do vRealize Orchestrator executado e as entradas do vRealize Orchestrator vinculadas.

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

**Observação** Usando bibliotecas de registro padrão, tudo o que você registra na ação que usa seu script também é mostrado no registro do fluxo de trabalho. As entradas e o retorno do seu script devem ter parâmetros de entrada e tipos de retorno correspondentes configurados no Cliente do vRealize Orchestrator. Por exemplo, a entrada de `vRAUrl`, no seu script, deve ter um parâmetro de entrada correspondente chamado `vRAUrl` no Cliente do vRealize Orchestrator. Da mesma forma, se o script retornar um valor de cadeia de caracteres, o tipo de retorno configurado no Cliente do vRealize Orchestrator também deverá ser um tipo de cadeia de caracteres. Se a sua ação retornar um objeto complexo, você poderá usar o tipo de retorno `Properties` ou `Composite Type`.

**5** Instale o módulo de solicitações do Python.

**Importante** Os módulos de dependência de terceiros devem ser instalados em uma pasta de nível da raiz na sua pasta de script principal `vro-python-vra`. Para esse caso de uso, crie uma pasta `lib` para o seu módulo de solicitações.

**a** Crie uma pasta `lib`.

```
mkdir lib
```

**b** Instale o módulo de solicitações.

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

**6** Adicione o módulo de solicitações ao script `handler.py`.

```
import requests

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

**7** Crie uma solicitação GET para a API de Projeto do vRealize Automation.

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

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

**8** Defina os valores `token` e `vRAUrl`.

- Recupere o token de acesso usando a API do Serviço de Identidade do vRealize Automation. Consulte [Obter o Token de Acesso para a API do vRealize Automation](#)
- Para o valor `vRAUrl`, defina o script para que ele use um parâmetro de entrada do vRealize Orchestrator com o mesmo nome.

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

- Adicione os novos valores ao arquivo `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()
```

**Observação** Como a resposta da API de Projeto do vRealize Automation é retornada em um formato JSON, use um tipo de retorno `Properties` ou `Composite Type` para a ação do vRealize Orchestrator.

**9** Crie um pacote ZIP que contenha o arquivo `handler.py` e a pasta `lib` do seu módulo de solicitação.

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

**Próximo passo**

Importe o script do PowerShell para uma ação do vRealize Orchestrator. Consulte o [Criar ações no Cliente do vRealize Orchestrator](#).

## Criar um script Node.js que chame a API de Projeto do vRealize Automation

Crie um script de amostra que usa o Node.js para chamar a API de Projeto do vRealize Automation.

### Pré-requisitos

Baixe e instale o Node.js 12. Consulte [Downloads do Node.js](#).

### Procedimentos

- 1 Na sua máquina local, abra um shell de linha de comando.
- 2 Crie uma pasta `vro-node-vra`.

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

- 3 Vá para a pasta `vro-node-vra`.

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

- 4 Crie um script Node.js chamado `handler.js`.

```
touch handler.js
```

O script `handler.js` deve definir uma função que aceite dois argumentos, o contexto do fluxo de trabalho de vRealize Orchestrator executado e as entradas do vRealize Orchestrator vinculadas.

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

---

**Observação** Usando bibliotecas de registro padrão, tudo o que você registra na ação que usa seu script também é mostrado no registro do fluxo de trabalho. As entradas e o retorno do seu script devem ter parâmetros de entrada e tipos de retorno correspondentes configurados no Cliente do vRealize Orchestrator. Por exemplo, a entrada de `vRAUrl`, no seu script, deve ter um parâmetro de entrada correspondente chamado `vRAUrl` no Cliente do vRealize Orchestrator. Da mesma forma, se o script retornar um valor de cadeia de caracteres, o tipo de retorno configurado no Cliente do vRealize Orchestrator também deverá ser um tipo de cadeia de caracteres. Se a sua ação retornar um objeto complexo, você poderá usar o tipo de retorno `Properties` ou `Composite Type`.

---

## 5 Instale o módulo de solicitações Node.js.

```
npm install request
```

**Importante** Os módulos de dependência de terceiros devem ser instalados na pasta `node_modules` no nível da raiz na pasta principal do script `vro-node-vra`. Não mova ou renomeie esta pasta.

## 6 Adicione o módulo de solicitações para o script `handler.js`.

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

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

## 7 Crie uma solicitação GET para a API de Projeto do vRealize Automation.

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

**8** Defina os valores `token` e `vRAUrl` .

- a Recupere o token de acesso usando a API do Serviço de Identidade do vRealize Automation. Consulte [Obter o Token de Acesso para a API do vRealize Automation](#).
- b Para o valor `vRAUrl`, defina o script para que ele use um parâmetro de entrada do vRealize Orchestrator com o mesmo nome.

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

- c Adicione os novos valores ao arquivo `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));
  });
}
```

**Observação** Como a resposta da API de Projeto do vRealize Automation é retornada em um formato JSON, use um tipo de retorno `Properties` ou `Composite Type` para a ação do vRealize Orchestrator.

- 9** Crie um pacote ZIP que contenha o arquivo `handler.js` e a pasta `node_modules` do seu módulo de solicitação.

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

**Próximo passo**

Importe o script Node.js para uma ação do vRealize Orchestrator. Consulte o [Criar ações no Cliente do vRealize Orchestrator](#).

## Criar um script do PowerShell que chama a API de Projeto do vRealize Automation

Crie um script de amostra que usa o PowerShell para chamar a API de Projeto do vRealize Automation.

**Procedimentos**

- 1** Na sua máquina local, abra um shell de linha de comando.
- 2** Crie uma pasta `vro-powershell-vra`.

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

**3** Vá para a pasta `vro-powershell-vra`.

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

**4** Crie um script do PowerShell chamado `handler.ps1`.

```
touch handler.ps1
```

O script `handler.ps1` deve definir uma função que aceite dois argumentos, o contexto do fluxo de trabalho de vRealize Orchestrator executado e as entradas do vRealize Orchestrator vinculadas.

```
function Handler {
    Param($context, $inputs)

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

**Observação** Usando bibliotecas de registro padrão, tudo o que você registra na ação que usa seu script também é mostrado no registro do fluxo de trabalho. As entradas e o retorno do seu script devem ter parâmetros de entrada e tipos de retorno correspondentes configurados no Cliente do vRealize Orchestrator. Por exemplo, a entrada de `vRAUrl`, no seu script, deve ter um parâmetro de entrada correspondente chamado `vRAUrl` no Cliente do vRealize Orchestrator. Da mesma forma, se o script retornar um valor de cadeia de caracteres, o tipo de retorno configurado no Cliente do vRealize Orchestrator também deverá ser um tipo de cadeia de caracteres. Se a sua ação retornar um objeto complexo, você poderá usar o tipo de retorno `Properties` ou `Composite Type`.

**5** Instale o módulo de afirmação do PowerShell.

**Importante** Os módulos de dependência de terceiros devem ser instalados em uma pasta de nível da raiz na sua pasta de script principal `vro-powershell-vra`. Para esse caso de uso, crie uma pasta `Modules` para o seu módulo de afirmação.

**a** Crie uma pasta `Módulos`.

```
mkdir Modules
```

**b** Instale o módulo de afirmação.

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

**6** Adicione o módulo de afirmação ao script `handler.ps1`.

```
Import-Module Assert

function Handler {
```



```
Param($context, $inputs)

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

- 7** Crie uma solicitação GET para a API de Projeto do vRealize Automation que usa o cmdlet do `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** Defina os valores `token` e `vRAUrl`.
- Recupere o token de acesso usando a API do Serviço de Identidade do vRealize Automation. Consulte [Obter o Token de Acesso para a API do vRealize Automation](#).
  - Adicione os atributos de módulo de afirmação `Assert-NotNull` e `Assert-Type`.

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

- c Para o valor `vRAUrl`, defina o script para que ele use um parâmetro de entrada do vRealize Orchestrator com o mesmo nome.

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

- d Adicione os novos valores ao arquivo `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
}
```

**Observação** Como a resposta da API de Projeto do vRealize Automation é retornada em um formato JSON, use um tipo de retorno `Properties` ou `Composite Type` para a ação do vRealize Orchestrator.

- 9 Crie um pacote ZIP que contenha o arquivo `handler.ps1` e a pasta `Módulos` do seu módulo de afirmação.

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

### Próximo passo

Importe o script do PowerShell para uma ação do vRealize Orchestrator. Consulte o [Criar ações no Cliente do vRealize Orchestrator](#).

# Gerenciar fluxos de trabalho

# 5

Um fluxo de trabalho é uma série de ações e decisões que você executa sequencialmente. O vRealize Orchestrator fornece uma biblioteca de fluxos de trabalho que executam tarefas comuns de gerenciamento. O vRealize Orchestrator também fornece bibliotecas das ações individuais que os fluxos de trabalho executam.

Os fluxos de trabalho combinam ações, decisões e os resultados que, quando executados em uma ordem específica, concluem uma tarefa específica ou um processo específico em um ambiente virtual. Os fluxos de trabalho realizam tarefas como provisionamento de máquinas virtuais, backup, manutenção regular, envio de e-mails, realização de operações de SSH, gerenciamento da infraestrutura física e outras operações de utilitário gerais. Os fluxos de trabalho aceitam entradas de acordo com suas funções. Você pode criar fluxos de trabalho que são executados de acordo com as agendas definidas, ou que são executados se ocorrerem determinados eventos previstos. As informações podem ser fornecidas por você, por outros usuários, por outro fluxo de trabalho ou ação, ou por um processo externo, como uma chamada de serviço Web de um aplicativo. Os fluxos de trabalho executam alguma validação e filtragem de informações antes de serem executadas.

Os fluxos de trabalho podem demandar outros fluxos de trabalho. Por exemplo, você pode ter um fluxo de trabalho que chama outro fluxo de trabalho para criar uma nova máquina virtual.

Você cria fluxos de trabalho usando o ambiente de desenvolvimento integrado (IDE) da interface do vRealize Orchestrator Client, que fornece acesso à biblioteca de fluxo de trabalho e à capacidade de executar fluxos de trabalho no mecanismo de fluxo de trabalho. O mecanismo de fluxo de trabalho também pode tirar objetos de bibliotecas externas que você conecta ao vRealize Orchestrator. Esse recurso permite que você personalize processos ou implemente funções fornecidas por aplicativos de terceiros.

Este capítulo inclui os seguintes tópicos:

- Fluxos de trabalho padrão na biblioteca do fluxo de trabalho do vRealize Orchestrator
- Criar fluxos de trabalho no Cliente do vRealize Orchestrator
- Editar fluxos de trabalho e ações do fluxo de trabalho principal
- Designer do formulário de entrada do vRealize Orchestrator
- Solicitações de interação do usuário no Cliente vRealize Orchestrator
- Agendar fluxos de trabalho no Cliente vRealize Orchestrator

- [Localizar referências de objeto em fluxos de trabalho](#)

## Fluxos de trabalho padrão na biblioteca do fluxo de trabalho do vRealize Orchestrator

O vRealize Orchestrator fornece uma biblioteca padrão de fluxos de trabalho que você pode usar para automatizar as operações na sua infraestrutura virtual. Os fluxos de trabalho na biblioteca padrão estão bloqueados no estado somente leitura. Para personalizar um fluxo de trabalho padrão, você deve duplicar esse fluxo. Os fluxos de trabalho duplicados ou personalizados que você cria são totalmente editáveis.

O conteúdo da biblioteca de fluxos de trabalho pode ser acessado pelo menu **Biblioteca > Fluxos de Trabalho** do vRealize Orchestrator Client com base em HTML5. Os fluxos de trabalho padrão e personalizado no cliente são organizados por meio de etiquetas. Por exemplo, você pode acessar o fluxo de trabalho **Gerar par de chaves** inserindo **SSH** na caixa de pesquisa da biblioteca de fluxos de trabalho.

---

**Observação** Não é possível adicionar novas etiquetas aos fluxos de trabalho padrão, a menos que você duplique o fluxo de trabalho.

---

## Criar fluxos de trabalho no Cliente do vRealize Orchestrator

Você pode usar o vRealize Orchestrator Client para criar e editar fluxos de trabalho.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Selecione **Biblioteca > Fluxos de Trabalho**.
- 3 Clique em **Novo Fluxo de Trabalho**.
- 4 Insira o nome do novo fluxo de trabalho e clique em **Criar**.
- 5 Use o editor de fluxo de trabalho para configurar as variáveis, as entradas e saídas de fluxo de trabalho, a estrutura do esquema e a apresentação do fluxo de trabalho.
- 6 Para concluir a edição do fluxo de trabalho, clique em **Salvar**.

---

**Observação** Você pode rastrear as alterações nos fluxos de trabalho na guia **Histórico de versões**. Para obter mais informações, consulte [Histórico da versão do objeto vRealize Orchestrator](#).

---

### Próximo passo

Você pode usar o recurso de repetição de token do vRealize Orchestrator para otimizar o desempenho de seus fluxos de trabalho. Para obter mais informações, consulte [Usar reprodução de token de fluxo de trabalho no Cliente vRealize Orchestrator](#).

## Editar fluxos de trabalho e ações do fluxo de trabalho principal

Edite fluxos de trabalho e ações diretamente do fluxo de trabalho principal no vRealize Orchestrator Client.

A edição de ações e fluxos de trabalho herdeiros diretamente do fluxo de trabalho principal pode ajudar a simplificar o desenvolvimento do fluxo de trabalho.

### Pré-requisitos

Crie um fluxo de trabalho que chame outro fluxo de trabalho, ação ou ambos.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá até **Biblioteca > Fluxos de Trabalho** e selecione seu fluxo de trabalho.
- 3 Selecione a guia **Esquema**.
- 4 Dependendo do tipo de objeto, clique duas vezes no **elemento de fluxo de trabalho** ou **Elemento de ação** na tela de fluxo de trabalho.
- 5 Edite o objeto.
- 6 Para concluir a edição da ação ou do fluxo de trabalho herdeiro, clique em **Salvar**.
- 7 Para retornar ao fluxo de trabalho principal, feche o editor de objetos.

## Designer do formulário de entrada do vRealize Orchestrator

Se um fluxo de trabalho exigir parâmetros de entrada, ele abrirá uma caixa de diálogo na qual os usuários inserem os valores necessários. Você pode organizar o conteúdo, o layout e a apresentação desta caixa de diálogo com o designer de formulário de entrada.

O designer de formulário de entrada está localizado na guia **Formulário de Entrada** do editor de fluxo de trabalho. Esse designer consiste em um menu de navegação, uma tela de criação e um menu de propriedades. Você pode arrastar entradas e elementos genéricos do menu à esquerda para a tela de criação. Na tela, você pode definir a posição dos parâmetros de entrada, organizá-los em guias de entrada separadas e configurar as propriedades do parâmetro de entrada.

---

**Observação** Não é possível usar o conteúdo da guia **Variáveis** do editor de fluxo de trabalho no designer de formulário de entrada. Você só pode usar parâmetros na guia **Entrada/Saída**.

---

### Elementos genéricos

Você pode adicionar elementos genéricos, como menus suspensos e caixas de texto de senha, ao designer de formulário de entrada. Os elementos genéricos não correspondem aos parâmetros de entrada reais, mas podem ser vinculados a parâmetros de entrada.

## Criar a caixa de diálogo de parâmetros de entrada do fluxo de trabalho no Cliente vRealize Orchestrator

Você pode usar o designer de formulário de entrada para criar e personalizar a caixa de diálogo do parâmetro de entrada de fluxo de trabalho.

### Pré-requisitos

Verifique se o fluxo de trabalho tem uma lista definida de parâmetros de entrada.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Acesse **Biblioteca > Fluxos de Trabalho**.
- 3 Selecione o fluxo de trabalho personalizado.
- 4 Clique na guia **Formulário de Entrada**.
- 5 (Opcional) Crie guias para uso na caixa de diálogo de entrada.  
Você pode usar guias para organizar a estrutura da sua caixa de diálogo.
- 6 Selecione os parâmetros de entrada.
- 7 Edite as propriedades dos parâmetros de entrada.  
Para obter mais informações sobre as propriedades do parâmetro de entrada, consulte [Propriedades do parâmetro de entrada no Cliente vRealize Orchestrator](#).
- 8 (Opcional) Adicione elementos genéricos à tela e associe-os aos parâmetros de entrada.
- 9 (Opcional) Adicione a validação externa aos parâmetros de entrada. Para obter mais informações, consulte [Usar ações para validar entradas de fluxo de trabalho do vRealize Orchestrator](#).
- 10 Clique em **Salvar**.

### Resultados

Você criou o layout da caixa de diálogo do fluxo de trabalho e definiu as propriedades dos parâmetros de entrada.

## Propriedades do parâmetro de entrada no Cliente vRealize Orchestrator

Você pode definir as propriedades de parâmetro para restringir os parâmetros de entrada que os usuários fornecem quando executam fluxos de trabalho do vRealize Orchestrator.

Com o vRealize Orchestrator, você pode definir as propriedades de parâmetro usadas para quantificar os valores de parâmetro de entrada usados em fluxos de trabalho. As propriedades de parâmetro que você define impõem limites sobre os tipos e os valores dos parâmetros de entrada que os usuários podem fornecer nos fluxos de trabalho do vRealize Orchestrator.

As propriedades do parâmetro validam os parâmetros de entrada e modificam a apresentação das caixas de texto que aparecem na caixa de diálogo dos parâmetros de entrada. Algumas propriedades de parâmetro podem criar dependências entre os parâmetros.

Propriedade de parâmetro	Descrição
<b>Rótulo</b>	Defina o rótulo do parâmetro de entrada.
<b>Tipo de exibição</b>	Defina o tipo de exibição da caixa de texto de entrada.
<b>Visibilidade</b>	Defina a visibilidade do parâmetro de entrada.
<b>Somente leitura</b>	Defina a caixa de texto de entrada como somente leitura.
<b>Ajuda personalizada</b>	Defina a descrição do sinal do parâmetro de entrada.
<b>Valor padrão</b>	Defina o valor padrão do parâmetro de entrada.
<b>Etapa</b>	Usado para entradas de tipo de número. Defina por quanto tempo o valor do parâmetro de entrada aumenta por clique.
<b>Obrigatório</b>	Define se o valor do parâmetro de entrada é obrigatório ou não.
<b>Expressão regular</b>	Valida a entrada usando uma expressão regular.
<b>Valor mínimo</b>	Defina o valor mínimo ou o comprimento do parâmetro.
<b>Valor máximo</b>	Defina o valor máximo ou o comprimento do parâmetro.
<b>Caixa de texto de correspondência</b>	Defina o valor do parâmetro de entrada para corresponder ao valor de outro parâmetro de entrada.
<b>Origem do valor</b>	Defina a origem do valor das propriedades do parâmetro nas guias <b>Aparência</b> , <b>Valor</b> e <b>Restrições</b> .  <b>Observação</b> Você pode importar o valor de ações externas usando <b>Origem externa</b> . A filtragem de ações disponíveis é feita por tipo de parâmetro.

## Usar ações para validar entradas de fluxo de trabalho do vRealize Orchestrator

Use ações externas para validar as entradas de seus fluxos de trabalho personalizados.

### Pré-requisitos

Crie um fluxo de trabalho personalizado com os parâmetros de entrada. Para obter mais informações, consulte [Criar fluxos de trabalho no Cliente do vRealize Orchestrator](#).

Você pode usar o designer de formulário de entrada para criar validações externas para suas entradas de fluxo de trabalho. As validações externas usam scripts de ação que retornam um valor de cadeia de caracteres quando o valor do parâmetro de entrada contém um erro. Se o valor do parâmetro de entrada for válido, a validação externa não retornará nada.

## Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Crie uma ação de validação.
  - a Vá para **Biblioteca > Ações**.
  - b Clique em **Nova Ação**.
  - c Insira as informações necessárias na guia **Resumo**.
  - d Insira os parâmetros de entrada da ação de validação.

---

**Observação** Os nomes dos parâmetros de entrada da ação de validação devem ser idênticos aos nomes dos parâmetros de entrada de fluxo de trabalho que estão sendo validados.

---

- e Insira o script da ação de validação na guia **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
```

---

**Observação** O script anterior é um exemplo simples e não representa o escopo completo dos scripts de validação que podem ser usados.

---

- f Clique em **Salvar**.
- 3 Aplique a validação externa.
  - a Vá para **Biblioteca > Fluxos de trabalho**.
  - b Selecione o fluxo de trabalho personalizado.
  - c Selecione a guia **Formulário de Entrada**.
  - d Selecione o ícone da área de transferência no canto superior esquerdo da tela.
  - e Arraste um elemento de validação do vRealize Orchestrator para a tela.
  - f Selecione o elemento de validação, insira um rótulo de validação e selecione a ação de validação.



g (Opcional) Crie elementos de validação adicionais.

h Clique em **Salvar**.

#### 4 Execute o fluxo de trabalho.

Se a validação encontrar um erro, ela retornará uma cadeia de caracteres. Se a validação for bem-sucedida, a validação não retornará nada e a execução do fluxo de trabalho continuará.

#### Resultados

Você criou uma validação externa para o fluxo de trabalho personalizado do vRealize Orchestrator.

## Solicitações de interação do usuário no Cliente vRealize Orchestrator

Os fluxos de trabalho podem solicitar entrada de usuário adicional antes que possam ser concluídos.

Os fluxos de trabalho que exigem interação do usuário suspendem as operações até que os parâmetros de entrada solicitados sejam fornecidos pelo usuário. Os fluxos de trabalho definem quais usuários podem fornecer as informações solicitadas e enviar solicitações de interação de forma adequada. Os fluxos de trabalho que aguardam a entrada do usuário são exibidos no painel **Execuções de Fluxo de Trabalho Recentes** do painel do vRealize Orchestrator Client e no menu de notificação no canto superior direito.

## Agendar fluxos de trabalho no Cliente vRealize Orchestrator

Você pode usar o agendamento para automatizar as execuções do fluxo de trabalho do vRealize Orchestrator.

Quando você agenda as execuções de fluxo de trabalho, defina a data, a hora e os intervalos em que a tarefa agendada será executada.

#### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Selecione seu fluxo de trabalho no menu **Biblioteca** e, no painel do fluxo de trabalho, clique em **Agendar**.
- 3 Configure os parâmetros de tarefa agendada nas guias **Geral**, **Agendamento** e **Fluxo de Trabalho**.

---

**Observação** A categoria do parâmetro do **Fluxo de trabalho** é visível apenas para fluxos de trabalho que exigem parâmetros de entrada.

---

Parâmetro	Descrição
<b>Nome</b>	O nome da tarefa agendada.
<b>Descrição</b>	Uma breve descrição detalhando a finalidade da tarefa agendada.
<b>Iniciar</b>	A data e a hora da primeira execução agendada do fluxo de trabalho.
<b>Iniciar se estiver no passado</b>	Selecione se deseja iniciar o fluxo de trabalho, se a hora agendada estiver no passado. <b>Sim</b> inicia o fluxo de trabalho agendado imediatamente. <b>Não</b> inicia o fluxo de trabalho na próxima recorrência agendada.
<b>Agendar</b>	Defina o padrão de recorrência e as entradas de gatilho de evento da tarefa agendada.
<b>Data de término</b>	Visível somente se <b>Sem Reincidência</b> estiver selecionado. Defina a data e a hora de quando a tarefa agendada terminar.
<b>Fluxo de Trabalho</b>	Insira os parâmetros de entrada do fluxo de trabalho.

#### 4 Clique em **Criar**.

#### Resultados

Você criou uma tarefa agendada para o fluxo de trabalho. Os fluxos de trabalho agendados aparecem em **Atividade > Agendada**. Você pode excluir tarefas agendadas clicando em **Excluir** no painel de agendamento.

## Editar tarefa agendada no Cliente vRealize Orchestrator

As tarefas agendadas podem ser editadas para alterar parâmetros, como data, hora e recorrência do fluxo de trabalho agendado.

#### Pré-requisitos

Crie uma tarefa agendada no fluxo de trabalho.

#### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Selecione a tarefa agendada em **Atividade > Agendada**.
- 3 Clique em **Editar** no painel do fluxo de trabalho.
- 4 Edite a agenda e clique em **Salvar**.

**Observação** Os parâmetros de entrada definidos ao se criar a tarefa agendada são somente leitura e não podem ser editados. Para alterar esses parâmetros, crie uma nova tarefa agendada para este fluxo de trabalho.

## Localizar referências de objeto em fluxos de trabalho

Como desenvolvedor de fluxo de trabalho, você pode usar informações de referência de objeto para otimizar o ciclo de vida de desenvolvimento.

Com o vRealize Orchestrator Client, você pode encontrar informações de referência de objeto. Esse recurso tem duas funções:

- **Localizar Dependências:** encontre informações sobre dependências de objeto em seus fluxos de trabalho. As dependências podem incluir outros fluxos de trabalho, ações, elementos de recurso e elementos de configuração.
- **Localizar Usos:** saiba se o fluxo de trabalho selecionado é usado em outros fluxos de trabalho na biblioteca do vRealize Orchestrator Client.

Você pode acessar informações sobre referências de objetos no editor de fluxo de trabalho ou na biblioteca do vRealize Orchestrator Client no Modo de Exibição de Cartão, de Lista ou de Árvore. Para obter mais informações sobre os diferentes tipos de organização de conteúdo da biblioteca do vRealize Orchestrator Client, consulte [Organização de conteúdo no vRealize Orchestrator Client](#).

O procedimento a seguir demonstra como você pode acessar referências de objeto no editor de fluxo de trabalho.

### Pré-requisitos

Desenvolva um fluxo de trabalho que inclua pelo menos uma referência de objeto.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá até **Biblioteca > Fluxos de Trabalho** e selecione seu fluxo de trabalho.
- 3 Para encontrar informações sobre dependências de objeto, clique em **Localizar Dependências**.

---

**Observação** Enquanto estiver na janela pop-up de dependências, você poderá selecionar objetos referenciados na lista. A seleção de um objeto abre uma guia vRealize Orchestrator Client separada, onde você pode visualizar os detalhes do objeto selecionado ou editá-lo.

---

- 4 Para encontrar informações sobre onde o fluxo de trabalho selecionado é usado, clique em **Localizar Usos**.

# Gerenciar ações

## 6

Você pode modificar seus fluxos de trabalho do vRealize Orchestrator adicionando os scripts de ação.

O vRealize Orchestrator Client fornece bibliotecas de ações predefinidas e um editor de ações para scripts de ação personalizados. As ações representam funções individuais que você usa como base nos fluxos de trabalho.

As ações são funções de JavaScript. As ações podem ter vários parâmetros de entrada e ter um único valor de retorno. As ações podem chamar qualquer objeto na API do vRealize Orchestrator ou objetos em qualquer API que você importa para o vRealize Orchestrator usando um plug-in.

Quando um fluxo de trabalho é executado, uma ação toma parâmetros de entrada das variáveis do fluxo de trabalho. Essas variáveis podem ser os parâmetros de entrada inicial do fluxo de trabalho, ou as variáveis que outros elementos do fluxo de trabalho definem quando são executados.

O editor de ações inclui um recurso de preenchimento automático para scripts e um API Explorer, apresentando os tipos de scripts disponíveis e sua documentação.

Este capítulo inclui os seguintes tópicos:

- [Criar ações no Cliente do vRealize Orchestrator](#)
- [Ações de depuração e execução](#)
- [Conceitos básicos para scripts do Python, Node.js e PowerShell](#)
- [Limites do tempo de execução para scripts Python, Node.js e PowerShell](#)

## Criar ações no Cliente do vRealize Orchestrator

Você pode usar o vRealize Orchestrator Client para criar, editar e excluir scripts de ação.

Você pode usar os seguintes tempos de execução ao criar ações:

- Python 3,7
- Node.js 12
- PowerCLI 11/Powershell 6.2

## ■ PowerCLI 12/Powershell 7

---

**Observação** O tempo de execução do PowerCLI inclui o PowerShell e os seguintes módulos: VMware.PowerCLI, PowerNSX, PowervRA.

---

### Pré-requisitos

Antes de criar um script do Python, Node.js ou PowerShell, verifique se você está familiarizado com os conceitos essenciais para o desenvolvimento de scripts compatíveis com o vRealize Orchestrator que usam esses tempos de execução. Consulte o [Conceitos básicos para scripts do Python, Node.js e PowerShell](#).

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Biblioteca > Ações**.
- 3 Clique em **Nova Ação**.
- 4 Na guia **Geral**, insira o nome e o nome do módulo da ação.

---

**Observação** O nome e o nome do módulo devem ser exclusivos para cada ação. O nome da ação deve ser uma função JavaScript válida. O nome da ação deve ser uma única palavra que pode conter apenas letras, números e os símbolos de dólar ("\$\$") e sublinhado ("\_"). O nome do módulo deve ser composto por palavras separadas pelo caractere ponto (".").

---

- 5 (Opcional) Crie uma descrição, um número de versão, tags e permissões de grupo para a ação.
- 6 Na guia **Script**, adicione entradas de ação, selecione o tipo de retorno da saída e grave o script.

---

**Observação** Ao selecionar **Zip** no menu suspenso **Tipo**, você pode importar uma origem de script externo e, se aplicável, seus módulos de dependência.

---

- 7 Para concluir a edição da ação, clique em **Salvar**.

Uma mensagem informa que a ação foi salva.

### Próximo passo

Para exibir um exemplo de caso de uso de como você pode usar as ações do vRealize Orchestrator, consulte [Como integrar os Amazon Web Services no vRealize Orchestrator usando Python](#).

## Ações de depuração e execução

Você pode melhorar suas ações executando-as e depurando-as diretamente do editor de ações.

Você pode executar e depurar ações diretamente do editor de ação do Cliente vRealize Orchestrator. Com esse recurso, você pode garantir que suas ações sejam realizadas conforme o esperado quando estão integradas aos seus fluxos de trabalho.

## Executar ações no Cliente vRealize Orchestrator

Como designer do fluxo de trabalho, você deseja executar suas ações antes de integrá-las a um fluxo de trabalho.

### Pré-requisitos

Crie uma ação. Consulte o [Criar ações no Cliente do vRealize Orchestrator](#).

### Procedimentos

- 1 Faça login no Cliente vRealize Orchestrator.
- 2 Acesse **Biblioteca > Ações** e selecione a ação que você deseja executar.
- 3 Clique em **Executar**.
- 4 Insira os parâmetros de entrada necessários e clique em **Executar**.

Com a execução da ação concluída, clique na guia **Resultados/Entradas**. Se a execução da ação tiver encontrado um erro, ele será exibido nessa guia em vermelho. Você pode visualizar os detalhes da ação executada no elemento **Resultados da Ação**.

---

**Observação** Os resultados da execução da ação não são salvos.

---

## Depurar ações no Cliente vRealize Orchestrator

Como um designer de fluxo de trabalho, você pode depurar ações inserindo pontos de interrupção no seu script.

O vRealize Orchestrator inclui uma ferramenta de depuração integrada que você pode usar para depurar o script e as propriedades de entrada de sua ação. O processo de depuração pode ser iniciado no editor de ações inserindo pontos de interrupção nas linhas de script da sua ação.

---

**Observação** A ferramenta de depuração integrada funciona somente com ações que usam o tempo de execução JavaScript padrão. Para obter um exemplo de como você pode depurar scripts de ação que usam diferentes tempos de execução, consulte [Depurar a ação do Amazon Web Services](#).

---

### Pré-requisitos

Crie uma ação. Consulte o [Criar ações no Cliente do vRealize Orchestrator](#).

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Biblioteca > Ações** e selecione a ação que deseja depurar.

3 No editor de ações, adicione pontos de interrupção às linhas do script de ação que você deseja depurar.

4 Clique em **Depurar**.

5 Insira os parâmetros de entrada da sua ação e clique em **Executar**.

Uma ação executada no modo de depuração é iniciada.

6 Quando a execução da ação estiver pausada depois de atingir um ponto de interrupção, selecione uma das seguintes opções:

Opção	Descrição
<b>Continuar</b>	Retoma a execução da ação até que outro ponto de interrupção seja alcançado ou a execução da ação seja concluída.
<b>Entrar em</b>	Entra na função de ação atual. Se o depurador não puder ficar mais profundo na linha atual da função, ele executará uma operação <b>Passar sobre</b> .
<b>Passar sobre</b>	O depurador continua na próxima linha da função atual.
<b>Retorno de etapa</b>	O depurador vai para a linha que será executada quando a função atual retornar.

7 (Opcional) Na guia **Depurador**, adicione expressões.

8 (Opcional) Na guia **Depurador**, edite o valor das suas variáveis.

## Conceitos básicos para scripts do Python, Node.js e PowerShell

Ao criar seu script para uso no vRealize Orchestrator, você deve verificar se o seu script tem a estrutura e a formatação corretas.

### Tempo de execução com suporte

Para desenvolver ações e fluxos de trabalho do vRealize Orchestrator, você pode usar os seguintes tempos de execução:

- Python 3,7
- Node.js 12
- PowerCLI 11/Powershell 6.2
- PowerCLI 12/Powershell 7

**Observação** O tempo de execução do PowerCLI inclui o PowerShell e os seguintes módulos: VMware.PowerCLI, PowerNSX, PowervRA.

Você pode adicionar qualquer código-fonte personalizado aos novos tempos de execução, mas para aceitar contexto e entradas e retornar um resultado do e para o mecanismo do vRealize Orchestrator, você deve seguir o formato funcional correto.

## Recomendações de script

Para tarefas de script mais simples, você pode adicionar elementos de **Tarefa programável por script** ao seu esquema de fluxo de trabalho. Você pode usar ações do vRealize Orchestrator para tarefas de script mais complexas.

O uso de ações fornece dois benefícios específicos:

- As ações podem ser criadas, atualizadas, importadas e exportadas de forma independente dos fluxos de trabalho.
- As ações são objetos independentes que podem ser executados e depurados no próprio ambiente, o que pode levar a um processo de desenvolvimento mais fácil. Consulte o [Ações de depuração e execução](#).

## Requisitos de Função de Script

O nome padrão para a sua função de script é **manipulador**. A função aceita dois argumentos, contexto e entrada. Context é um objeto de mapa, contendo informações do sistema. Por exemplo, `vroURL` pode conter a URL da instância do vRealize Orchestrator que você deseja chamar, enquanto `executionId` contém o ID do token de uma execução de fluxo de trabalho.

Uma entrada é um objeto de mapa que contém todas as entradas fornecidas para as ações. Por exemplo, se você definir uma entrada em sua ação chamada `myInput`, poderá acessá-la no argumento de entradas, como `inputs.myInput` ou `inputs["myInput"]`, dependendo do seu tempo de execução. Qualquer coisa que você retornar da função é o resultado da ação. Portanto, o tipo de retorno de sua ação deve corresponder ao tipo de conteúdo que o script retorna no vRealize Orchestrator. Se você retornar um número primitivo, o tipo de retorno de ação deverá ser um tipo de número. Se você retornar uma cadeia de caracteres, o tipo de retorno da ação deve ser um tipo de cadeia de caracteres. Se você retornar um objeto complexo, o tipo de retorno deverá ser mapeado para `Properties` ou `Composite Type`. Esses mesmos princípios também se aplicam a matrizes.

Tipos de parâmetro de entrada e saída com suporte para tempos de execução Python, Node.js e PowerShell:

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



## Definir o Manipulador de Entrada

Por padrão, o valor do manipulador de entrada é `handler.handler`. Esse valor significa que o mecanismo do vRealize Orchestrator procura um arquivo de nível superior no seu pacote ZIP chamado `handler.py`, `handler.js`, ou `handler.ps1`, que inclui uma função chamada `handler`. Quaisquer diferenças nos nomes da função e do arquivo manipulador devem ser refletidas no valor do manipulador de entrada. Por exemplo, se o manipulador principal é chamado `index.js` e sua função é chamada `callMe`, você deve definir o valor para o manipulador de entrada para `index.callMe`.

## Depurar Scripts de Tempo de Execução em um IDE Externo

vRealize Orchestrator oferece suporte à depuração de scripts do Python e Node.js em um IDE externo. Não é possível depurar scripts do PowerShell em um IDE externo.

## Limites do tempo de execução para scripts Python, Node.js e PowerShell

Alguns scripts Python, Node.js ou PowerShell podem exigir que você altere os valores de memória e tempo limite no vRealize Orchestrator Client.

O vRealize Orchestrator Client usa um conjunto de valores padrão de memória e tempo limite para os scripts de ação Python, Node.js e PowerShell:

- Memória: 64 MB
- Tempo limite: 180 segundos

Se o script de ação exceder um ou ambos os valores padrão, haverá falha na execução da ação. Por exemplo, o script de ação pode usar vários módulos de dependência de terceiros. Nesse cenário, o limite de memória padrão de 64 MB pode não ser suficiente.

Para evitar execuções de ação com falha devido a recursos insuficientes, altere os valores de memória e tempo limite do editor de ações.

---

**Observação** Você também pode considerar a divisão do script em vários elementos de tarefas programáveis por script, que podem ser adicionados aos seus fluxos de trabalho.

---

### Procedimentos

- 1 Faça login no vRealize Orchestrator Client.
- 2 Vá até **Biblioteca > Ações** e selecione sua ação.
- 3 Selecione a guia **Script**.
- 4 Em **Limites de tempo de execução**, altere os valores de memória e tempo limite.
- 5 Clique em **Salvar**.
- 6 Para testar os novos limites de tempo de execução, clique em **Depuração**.

# Gerenciar elementos de configuração

# 7

Um elemento de configuração é uma lista de variáveis, que você pode usar para configurar constantes em toda a implantação do servidor do vRealize Orchestrator.

Você pode usar elementos de configuração para disponibilizar as variáveis para todos os fluxos de trabalho, ações e políticas em execução no servidor do vRealize Orchestrator.

Se você criar um pacote contendo um fluxo de trabalho, uma ação ou uma política que usa uma variável de um elemento de configuração, o vRealize Orchestrator incluirá automaticamente o elemento de configuração no pacote. Se você importar um pacote que contém um elemento de configuração para outro servidor do vRealize Orchestrator, também será possível importar os valores de variáveis de elemento de configuração. Por exemplo, se você criar um fluxo de trabalho que exige valores de variáveis que dependem do servidor do vRealize Orchestrator no qual ele é executado, a configuração dessas variáveis em um elemento de configuração permitirá exportar esse fluxo de trabalho para que outro servidor do vRealize Orchestrator possa usá-lo. Os elementos de configuração, portanto, permitem que você troque fluxos de trabalho, ações e políticas entre servidores com mais facilidade.

---

**Observação** Não é possível importar valores de uma variável de elemento de configuração de um elemento de configuração exportado do vRealize Orchestrator 5.1 ou anterior.

---

Este capítulo inclui os seguintes tópicos:

- [Criar elementos de configuração no Cliente do vRealize Orchestrator](#)

## Criar elementos de configuração no Cliente do vRealize Orchestrator

Com elementos de configuração, você pode definir variáveis comuns em um servidor do vRealize Orchestrator. Todos os elementos que estão em execução no servidor podem usar as variáveis definidas em um elemento de configuração.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Ativos > Configurações**.
- 3 Selecione **Nova Configuração**.

- 4 Insira o nome do elemento de configuração.
- 5 Selecione a guia **Variáveis**.
- 6 Para criar uma variável local, clique em **Novo**.
  - a Insira o nome da variável.
  - b Selecione o tipo de variável.

---

**Observação** Para criar uma matriz de variáveis de configuração, marque a caixa de seleção **Matriz**.

---

- c (Opcional) Insira um valor para a variável de configuração.
  - d Clique em **Salvar**.
- 7 Para concluir a criação de um elemento de configuração, clique em **Salvar**.

#### Próximo passo

Você pode usar o elemento de configuração para fornecer variáveis para fluxos de trabalho, ações ou políticas.

# Gerenciar políticas

# 8

As políticas são gatilhos de eventos que monitoram a atividade do sistema. As políticas respondem a eventos predefinidos emitidos por alterações no status ou no desempenho de objetos específicos do vRealize Orchestrator.

As políticas são uma série de regras, medidores, limites e filtros de eventos que executam certos fluxos de trabalho ou scripts quando ocorrem eventos predefinidos específicos no vRealize Orchestrator ou nas tecnologias que o vRealize Orchestrator acessa através de plug-ins. O vRealize Orchestrator avalia constantemente as regras de política enquanto a política está em execução. Por exemplo, você pode implementar medidores e limites de políticas que monitoram o comportamento de objetos do vCenter Server dos tipos `VC:HostSystem` e `VC:VirtualMachine`.

Este capítulo inclui os seguintes tópicos:

- [Criar e aplicar políticas no Cliente do vRealize Orchestrator](#)
- [Elementos de política no Cliente vRealize Orchestrator](#)
- [Gerenciar execuções de política no Cliente vRealize Orchestrator](#)

## Criar e aplicar políticas no Cliente do vRealize Orchestrator

Você pode usar as políticas para monitorar a atividade do sistema do vRealize Orchestrator para eventos específicos.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Biblioteca > Políticas**.
- 3 Selecione **Nova Política**.  
Você criou uma política em branco.
- 4 Insira um nome de política e um número de versão.
- 5 Selecione a guia **Variáveis**.

**6** Para criar uma variável local, clique em **Novo**.

- a Insira o nome da variável.
- b Selecione o tipo de variável.

---

**Observação** Para criar uma matriz de variáveis de política, marque a caixa de seleção **Matriz**.

---

- c Insira o valor da variável.

---

**Observação** Para importar o valor de uma variável de elemento de configuração, você pode usar **Associar à configuração**.

---

- d Clique em **Salvar**.

**7** Na guia **Definição**, adicione elementos de política e defina os identificadores de eventos.

Para obter mais informações sobre elementos de políticas, consulte [Elementos de política no Cliente vRealize Orchestrator](#).

**8** Clique em **Salvar**.

Você configurou a política.

#### Próximo passo

Para iniciar uma política, selecione a política e clique em **Executar**. Insira o nome de execução da política e, se solicitado, os parâmetros de entrada necessários.

Para exibir o status da política, vá para **Atividade > Execuções de Política**.

## Elementos de política no Cliente vRealize Orchestrator

Você pode usar elementos de política para executar fluxos de trabalho ou scripts predefinidos do vRealize Orchestrator quando ocorre um evento.

Você pode adicionar um elemento de política para acionar fluxo de trabalho ou script é executado como resposta para eventos disparados por objetos. Com o elemento de evento periódico, você pode agendar execuções de fluxo de trabalho ou de script. Com o elemento raiz, você pode definir o comportamento de início ou parada das políticas. Os elementos de política podem ter manipuladores de eventos que definem quando os elementos da política devem ser executados.

---

**Observação** Os manipuladores de eventos que ativam elementos de política podem ser fluxos de trabalho ou scripts de ação. Se você adicionar um fluxo de trabalho e um script a um manipulador de eventos, a política ignorará o gatilho de script e usará somente o gatilho de fluxo de trabalho.

---

Manipulador de eventos	Descrição
<b>OnInit</b>	O elemento de política é acionado toda vez que você inicia a política.
<b>OnExit</b>	O elemento de política é acionado toda vez que você interrompe a política.
<b>OnExecute</b>	Usado pelo elemento de evento periódico. Aciona o elemento de política durante o tempo especificado no elemento de evento periódico.

**Observação** As tecnologias conectadas ao banco de dados do vRealize Orchestrator podem ter manipuladores de eventos exclusivos. Por exemplo, por meio do plug-in SNMP, você pode usar o manipulador de eventos **OnTrap** ao criar elementos de política com base em SNMP.

Os elementos de política são configurados na guia **Definição** da janela de edição de política.

## Gerenciar execuções de política no Cliente vRealize Orchestrator

Você pode usar o vRealize Orchestrator Client para gerenciar a prioridade da política e o comportamento de inicialização do servidor de políticas para quando o servidor do vRealize Orchestrator for reiniciado.

### Pré-requisitos

Crie e execute uma política. Para obter mais informações, consulte [Criar e aplicar políticas no Cliente do vRealize Orchestrator](#).

### Procedimentos

- 1 Faça login no Cliente vRealize Orchestrator como administrador.
- 2 Vá até **Atividade > Execuções de Política**.
- 3 Clique na execução de política que você deseja gerenciar.
- 4 Clique em **Parar**.  
O estado da política muda para **Parado**.
- 5 Na guia **Geral**, defina a prioridade da política e o comportamento de inicialização do servidor.
- 6 Para reiniciar a política, clique em **Executar**.  
O estado da política muda para **Em execução**.

# Gerenciar elementos de recursos

## 9

Os fluxos de trabalho podem usar objetos que você cria de forma independente do vRealize Orchestrator como atributos. Para usar objetos externos como atributos em fluxos de trabalho, importe-os para o servidor como elementos de recurso.

Os objetos que os fluxos de trabalho do vRealize Orchestrator podem usar como elementos de recurso incluem arquivos de imagem, scripts, modelos XML, arquivos HTML e assim por diante. Todos os fluxos de trabalho que são executados no servidor do vRealize Orchestrator podem usar qualquer elemento de recurso que você importar para o vRealize Orchestrator.

Depois de importar um objeto para o vRealize Orchestrator como elemento de recurso, você pode fazer alterações no objeto em um único local e propagar essas alterações automaticamente para todos os fluxos de trabalho que usam esse elemento de recurso.

O tamanho máximo para um elemento de recurso é 16 MB.

Você pode importar, exportar, restaurar, atualizar e excluir um elemento de recurso.

# Gerenciar pacotes

# 10

Use o vRealize Orchestrator Client para criar, exportar e importar pacotes. Os pacotes podem ser usados para exportar objetos de fluxo de trabalho para uso em outras instâncias do vRealize Orchestrator.

Os pacotes podem conter fluxos de trabalho, ações, políticas, elementos de configuração ou elementos de recursos.

Quando você adiciona um elemento a um pacote, o vRealize Orchestrator verifica se há dependências e adiciona quaisquer elementos dependentes ao pacote. Por exemplo, se você adicionar um fluxo de trabalho que usa ações ou outros fluxos de trabalho, o vRealize Orchestrator adicionará essas ações e fluxos de trabalho ao pacote.

Quando você importa um pacote, o servidor compara as versões dos diferentes elementos do seu conteúdo para os elementos locais correspondentes. A comparação mostra as diferenças nas versões entre os elementos locais e importados. O usuário pode decidir se deseja importar o pacote ou pode selecionar elementos específicos a serem importados.

Para a maioria dos objetos criados no vRealize Orchestrator Client, além dos elementos de recurso, os pacotes são a única maneira de exportar e importar esses objetos.

Os pacotes usam o gerenciamento de direitos digitais para controlar como o servidor de recebimento pode usar o conteúdo do pacote. O vRealize Orchestrator assina pacotes e criptografa os pacotes para proteção de dados. Os pacotes podem rastrear quais usuários exportam e redistribuem os elementos usando certificados X509.

## Criar um pacote no Cliente do vRealize Orchestrator

Você pode exportar e importar fluxos de trabalho, políticas, ações, referências de plug-in, elementos de recurso e elementos de configuração em pacotes. Todos os elementos dependentes relacionados aos objetos de pacote são adicionados automaticamente ao pacote, para garantir a compatibilidade entre as versões. Para excluir elementos dependentes, você deve primeiro remover o objeto de pacote relacionado.

Para a maioria dos objetos criados no vRealize Orchestrator Client, além dos elementos de recurso, os pacotes são a única maneira de exportar e importar esses objetos.



### Pré-requisitos

Verifique se o servidor do vRealize Orchestrator contém objetos como fluxos de trabalho, ações e políticas, que você pode adicionar a um pacote.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Ativos > Pacotes**.
- 3 Clique em **Novo Pacote**.
- 4 Na guia **Geral**, insira um nome e uma descrição para o pacote.

---

**Observação** Não é possível usar caracteres especiais ao atribuir nomes aos pacotes no vRealize Orchestrator Client.

---

- 5 Na guia **Conteúdo**, clique em **Adicionar**.
- 6 Selecione os objetos que você deseja adicionar ao pacote e clique em **Adicionar**.

---

**Observação** Os elementos dependentes são adicionados automaticamente ao pacote, mas não são exibidos na guia **Conteúdo** durante a criação do pacote. Para visualizar os elementos dependentes, selecione a guia **Conteúdo** após a criação do pacote.

---

- 7 Para concluir a criação do pacote, clique em **Criar**.

## Exportar um pacote no Cliente vRealize Orchestrator

Você pode usar o vRealize Orchestrator Client para exportar pacotes para outro ambiente do vRealize Orchestrator.

### Pré-requisitos

Crie um pacote que contenha os objetos do vRealize Orchestrator que você deseja exportar. Para obter mais informações, consulte [Criar um pacote no Cliente do vRealize Orchestrator](#).

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Ativos > Pacotes**.
- 3 Clique em **Exportar** no pacote.

#### 4 (Opcional) Selecione as opções de exportação adicionais.

Opção	Descrição
<b>Adicionar valores de atributo de configuração ao pacote</b>	Exporte os valores de atributo dos elementos de configuração.
<b>Adicionar valores de atributo SecureString de configuração ao pacote</b>	Exporte os valores de atributo de configuração do <code>SecureString</code> .
<b>Adicionar tags globais ao pacote</b>	Exporte as tags globais.

#### 5 Defina os direitos de acesso para os usuários que importam o pacote.

Opção	Descrição
<b>Exibir conteúdo</b>	O usuário pode visualizar o conteúdo do pacote.
<b>Adicionar ao pacote</b>	O usuário pode adicionar conteúdo do pacote importado para outros pacotes.
<b>Editar conteúdo</b>	O usuário pode editar o conteúdo do pacote.

#### 6 Clique em **OK**.

**Observação** Os arquivos com a extensão `.package` são salvos em uma pasta padrão na sua máquina local. Para definir uma pasta personalizada, você pode alterar as configurações de armazenamento no seu navegador.

### Resultados

Você exportou o pacote. Agora você pode usar os objetos exportados em outro ambiente do vRealize Orchestrator.

## Importar um pacote no Cliente vRealize Orchestrator

Use o vRealize Orchestrator Client para importar pacotes de fluxo de trabalho. Ao importar pacotes, você pode reutilizar objetos de um servidor do vRealize Orchestrator em outro servidor.

### Pré-requisitos

- Faça backup de todos os objetos padrão do vRealize Orchestrator que você tenha modificado.
- No servidor remoto, crie e exporte um pacote com os objetos que você deseja importar.

### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Vá para **Ativos > Pacotes**.
- 3 Clique em **Importar**, navegue até o arquivo `.package` que você deseja importar e clique em **Abrir**.

#### 4 Revise as informações do pacote importado.

- a A guia **Geral** contém informações sobre o pacote importado, como o nome, a descrição, o número de itens contidos e as informações do certificado.

Você pode ser solicitado a indicar que confia no certificado do editor da instância de origem do vRealize Orchestrator antes de importar o arquivo.

- b A guia **Elementos do pacote** lista os objetos incluídos no arquivo de importação. Se a versão de um objeto no pacote for posterior à versão no servidor, o sistema selecionará a versão do objeto para importação. As versões anteriores de elementos do vRealize Orchestrator devem ser selecionadas manualmente.
- c Desmarque **Importar Valores de Atributo de Configuração** se você não quiser importar os valores de atributo dos elementos de configuração do pacote.
- d No menu suspenso, selecione se deseja importar tags.

#### 5 Clique em **Importar**.

# Solução de problemas no Cliente vRealize Orchestrator

# 11

Você pode solucionar e monitorar sua instância do vRealize Orchestrator usando métricas, repetição de token, validação e depuração.

Este capítulo inclui os seguintes tópicos:

- Dados de métrica no Cliente vRealize Orchestrator
- Usar reprodução de token de fluxo de trabalho no Cliente vRealize Orchestrator
- Validar fluxos de trabalho do vRealize Orchestrator
- Depurar scripts de fluxo de trabalho no Cliente vRealize Orchestrator
- Depurar fluxos de trabalho pelo elemento de esquema
- Configurando um contêiner de Photon OS para pacotes Python

## Dados de métrica no Cliente vRealize Orchestrator

Os administradores do vRealize Orchestrator podem usar os perfis do fluxo de trabalho e as métricas do Painel do Sistema para solucionar os fluxos de trabalho e o sistema do vRealize Orchestrator.

O recurso de perfil coleta dados de métrica sobre as execuções do fluxo de trabalho. O perfil de fluxo de trabalho está ativado por padrão. Você pode desativar os perfis automáticos no **Centro de Controle > Propriedades da Extensão > profiler-8.5.0**.

A outra fonte para dados de métrica no vRealize Orchestrator Client é o Painel do Sistema, que fornece métricas de nível do sistema. Para obter mais informações, consulte [Usar o painel do sistema do vRealize Orchestrator](#).

## Fluxos de trabalho de perfil no Cliente vRealize Orchestrator

Você pode criar o perfil de suas execuções de fluxo de trabalho para solucionar e otimizar o ambiente do vRealize Orchestrator.

Você pode usar o recurso de perfil do vRealize Orchestrator Client para obter dados de métrica úteis sobre as execuções de fluxo de trabalho. Esses dados podem ser usados para otimizar o desempenho de seus fluxos de trabalho. Por padrão, as execuções de fluxo de trabalho são importadas automaticamente. Você pode desativar a criação de perfil automática na página **Propriedades da Extensão** do Centro de Controle do vRealize Orchestrator e executar o gerador de perfil manualmente. Para fazer uma execução manual da criação de perfil, localize seu fluxo de trabalho na biblioteca e selecione **Ações > Perfil**.

#### Pré-requisitos

Execute um fluxo de trabalho.

#### Procedimentos

- 1 Faça login no Cliente do vRealize Orchestrator.
- 2 Acesse **Atividade > Execuções de Fluxo de Trabalho**.
- 3 Execute uma execução do fluxo de trabalho.

No esquema de execução do fluxo de trabalho, você pode ver dados sobre os itens de fluxo de trabalho individuais. Os dados incluem a duração da execução total, a duração máxima e o número de execuções de item. Você pode filtrar essas informações no menu suspenso no canto superior direito da página.

- 4 Selecione a guia **Desempenho**.

Essa guia fornece dados de métricas sobre os tempos de CPU da execução do fluxo de trabalho, a duração da execução, o tamanho do token e os dados do item do fluxo de trabalho.

---

**Observação** Se a execução do fluxo de trabalho for suspensa, por exemplo, quando o fluxo de trabalho estiver aguardando uma entrada adicional, a métrica dos tempos da CPU capturará apenas o segmento de tempo de execução que ocorreu antes da conclusão.

---

#### Próximo passo

Use os dados coletados do perfil para otimizar seu fluxo de trabalho.

## Usar o painel do sistema do vRealize Orchestrator

Como administrador, você pode usar o Painel do Sistema do vRealize Orchestrator Client para obter dados de métrica úteis sobre os nós do seu ambiente do vRealize Orchestrator.

Você pode acessar o Painel do Sistema clicando na guia **Sistema**, na parte superior da página do painel do vRealize Orchestrator Client. Os dados fornecidos incluem:

- Status do nó
- Propriedades do nó

- Configurações do cluster. Você só pode visualizar as configurações do cluster no Painel do Sistema. Para alterar essas configurações, acesse a página **Gerenciamento de cluster do Orchestrator** do Centro de Controle do vRealize Orchestrator.
- Informações sobre segmentos
- Memória heap
- Memória não heap
- Uso do sistema de arquivos
- Dados de autenticação
- Pool de conexões da base de dados do Orchestrator
- Argumentos de entrada do processo

Esses dados podem ser usados para monitorar o estado dos nós individuais do ambiente do seu vRealize Orchestrator e solucionar problemas. Para navegar entre nós individuais, clique na guia associada a um nó na parte superior do Painel do Sistema.

## Usar reprodução de token de fluxo de trabalho no Cliente vRealize Orchestrator

Você pode usar o recurso de repetição de token para exibir as transições entre os itens nas execuções de fluxo de trabalho.

O recurso de repetição de token registra informações contextuais para cada transição entre os itens de fluxo de trabalho. Para cada item de fluxo de trabalho, a repetição de token grava quando o fluxo de trabalho é executado, finalizado e quais variáveis foram alteradas no final da execução do item do fluxo de trabalho. A repetição de token também faz referência às mensagens de log de script geradas para cada item de fluxo de trabalho.

---

**Observação** Os dados sobre as transições dos itens do fluxo de trabalho são armazenados no banco de dados PostgreSQL do vRealize Orchestrator. Esses dados são removidos do banco de dados quando a execução do fluxo de trabalho é excluída.

---

### Pré-requisitos

- Ative o recurso de repetição de token no Centro de Controle.
  - a Faça login no Centro de Controle como **raiz**.
  - b Selecione **Propriedades da Extensão**.
  - c Clique em **tokenreplay-8.5.0**.
  - d Para ativar o recurso de repetição de token, clique em **Ativar**.

- e Clique em **Salvar**.

---

**Observação** Pode levar até 5 minutos para o servidor do vRealize Orchestrator atualizar a extensão.

---

- Execute um fluxo de trabalho.

---

**Observação** Por padrão, a repetição de token não é executada automaticamente para todas as execuções de fluxo de trabalho no servidor do vRealize Orchestrator. Você pode executar as repetições de token para cada fluxo de trabalho individualmente ou ativar a extensão de repetição de token para todos os fluxos de trabalho na página **Propriedades da Extensão** do Centro de Controle.

---

#### Procedimentos

- 1 (Opcional) Ative a repetição de token para todas as execuções de fluxo de trabalho no servidor do vRealize Orchestrator.

---

**Observação** Para executar repetições de token individuais sem ativar o recurso no Centro de Controle, clique em **Executar com a repetição** na página do editor de fluxo de trabalho.

---

- a Faça login no Centro de Controle como **raiz**.
- b Selecione **Propriedades da Extensão**.
- c Clique em **tokenreplay-8.5.0**.
- d Para ativar o recurso de repetição de token para todos os fluxos de trabalho, verifique se a opção **Gravar repetição para todas as execuções de fluxo de trabalho** está ativada.
- e Clique em **Salvar**.

---

**Observação** Pode levar até 5 minutos para o servidor do vRealize Orchestrator atualizar a extensão.

---

- 2 Faça login no Cliente vRealize Orchestrator como administrador.
- 3 Acesse **Atividade > Execuções de Fluxo de Trabalho**.
- 4 Execute uma execução do fluxo de trabalho.
- 5 Selecione um item de execução do fluxo de trabalho no menu à esquerda.

As guias **Variável** e **Logs** agora exibem informações específicas para esse item do fluxo de trabalho.

## Validar fluxos de trabalho do vRealize Orchestrator

O vRealize Orchestrator fornece uma ferramenta de validação de fluxo de trabalho. A validação de um fluxo de trabalho ajuda a identificar erros no fluxo de trabalho e verifica se os dados fluem de um elemento para o próximo corretamente.

Por padrão, o vRealize Orchestrator sempre executa uma validação de fluxo de trabalho quando você executa um fluxo de trabalho.

Quando você valida um fluxo de trabalho, a ferramenta de validação cria uma lista de erros ou avisos. Clicar em um erro na lista realça o elemento de fluxo de trabalho que contém o erro.

Se você executar a ferramenta de validação no editor de fluxo de trabalho, a ferramenta fornecerá correções sugeridas para os erros que ela detectar. Algumas correções rápidas exigem informações adicionais ou parâmetros de entrada. Outras correções rápidas resolvem o erro para você.

A validação de fluxo de trabalho verifica as associações de dados e as conexões entre os elementos. A validação de fluxo de trabalho não verifica o processamento de dados que cada elemento do fluxo de trabalho executa. Como resultado, um fluxo de trabalho válido poderá ser executado incorretamente e produzir resultados errados se uma função em um elemento de esquema estiver incorreta.

## Validar um fluxo de trabalho e corrigir erros de validação no Cliente vRealize Orchestrator

Você deve validar um fluxo de trabalho para poder executá-lo. Você só poderá corrigir erros de validação se tiver aberto o fluxo de trabalho para edição.

### Pré-requisitos

Verifique se você tem um fluxo de trabalho completo para validar, com elementos de esquema vinculados e associações definidos.

### Procedimentos

- 1 Faça login no Cliente vRealize Orchestrator como administrador.
- 2 Vá para **Biblioteca > Fluxos de Trabalho** e selecione o fluxo de trabalho que você deseja validar.
- 3 Clique em **Editar**.
- 4 Clique em **Validar** no menu superior.

Se o fluxo de trabalho for válido, será exibida uma mensagem de confirmação. Se o fluxo de trabalho for inválido, será exibida uma lista de erros.

- 5 Para um fluxo de trabalho inválido, clique em uma mensagem de erro e tome as medidas necessárias para resolver o problema.

A ferramenta de validação destaca o elemento de esquema no qual o erro ocorre adicionando um ícone vermelho a ele. Sempre que possível, a ferramenta de validação exibe uma ação de correção rápida.

- Se você concordar com a ação de correção rápida proposta, clique nela para realizar essa ação.



- Se você discordar com a ação de correção rápida proposta, feche a caixa de diálogo Validação de fluxo de trabalho e corrija o elemento de esquema manualmente.

---

**Importante** Sempre verifique se a correção que o vRealize Orchestrator propõe é apropriado.

---

Por exemplo, a ação proposta pode ser excluir um atributo não utilizado, quando, na verdade, esse atributo pode não estar vinculado corretamente.

- 6 Repita as etapas anteriores até que você tenha eliminado todos os erros de validação.

#### Resultados

Você validou um fluxo de trabalho e corrigiu os erros de validação.

#### Próximo passo

Você pode executar o fluxo de trabalho.

## Depurar scripts de fluxo de trabalho no Cliente vRealize Orchestrator

Você pode depurar execuções de fluxo de trabalho inserindo pontos de interrupção no script dos itens de fluxo de trabalho.

Quando um ponto de interrupção é atingido, você tem várias opções para continuar o processo de depuração. Ao depurar um elemento do esquema de fluxo de trabalho, você pode visualizar as informações gerais sobre a execução do fluxo de trabalho, modificar as variáveis do fluxo de trabalho, adicionar expressões para observar e exibir mensagens de log.

---

**Observação** Realize toda a depuração de scripts em um ambiente que não seja de produção.

---

#### Procedimentos

- 1 Faça login no Cliente vRealize Orchestrator como administrador.
- 2 Selecione um fluxo de trabalho na biblioteca.
- 3 Abra o esquema de fluxo de trabalho, selecione um elemento de fluxo de trabalho e clique na guia **Scripting**.
- 4 Para inserir um ponto de interrupção, clique no círculo vermelho à esquerda do número da linha.

---

**Observação** Você só pode inserir pontos de interrupção em elementos de fluxo de trabalho com scripts.

---

- 5 Para executar o fluxo de trabalho no modo de depuração, clique em **Depurar**.  
Se o fluxo de trabalho exigir parâmetros de entrada, você deverá fornecê-los.

- 6 Quando a execução do fluxo de trabalho for pausada depois de atingir um ponto de interrupção, selecione uma das opções disponíveis.

Opção	Descrição
<b>Continuar</b>	Retoma a execução do fluxo de trabalho até que outro ponto de interrupção seja alcançado ou a execução do fluxo de trabalho seja concluída.
<b>Entrar em</b>	Você pode usar essa opção para entrar em um elemento de fluxo de trabalho. Não é possível entrar em um elemento de fluxo de trabalho aninhado ao depurar um fluxo de trabalho no editor de fluxo de trabalho.
<b>Passar sobre</b>	Ignora o elemento atual no esquema e pausa a execução do fluxo de trabalho no próximo elemento.

**Observação** Você pode instruir o depurador a ignorar o ponto de interrupção atual clicando no ponto de interrupção. Isso altera o símbolo do ponto de interrupção para um triângulo verde.

- 7 (Opcional) Na guia **Depurador**, insira as expressões para observar.  
 Você pode usar expressões para seguir a conclusão de variáveis específicas.
- 8 (Opcional) Na guia **Depurador**, modifique os valores das variáveis.

## Depurar fluxos de trabalho pelo elemento de esquema

Como um designer de fluxo de trabalho, você pode depurar elementos individuais do esquema.

### Procedimentos

- 1 Faça login no vRealize Orchestrator Client.
- 2 Vá até **Biblioteca > Fluxos de Trabalho** e selecione seu fluxo de trabalho.
- 3 Selecione a guia **Esquema**.
- 4 Selecione o elemento de fluxo de trabalho que você deseja depurar e clique no botão Depurar no canto superior esquerdo do elemento.

**Observação** Ao adicionando um ponto de interrupção a um elemento de esquema do **Elemento de Fluxo de Trabalho**, você pode depurar fluxos de trabalho herdeiro diretamente do fluxo de trabalho principal. Quando o depurador atinge o elemento de esquema do **Elemento de Fluxo de Trabalho**, ele abre a visualização de esquema do fluxo de trabalho herdeiro.

- 5 Repita o procedimento para todos os outros elementos de esquema que você deseja depurar.
- 6 Clique em **Depurar**.

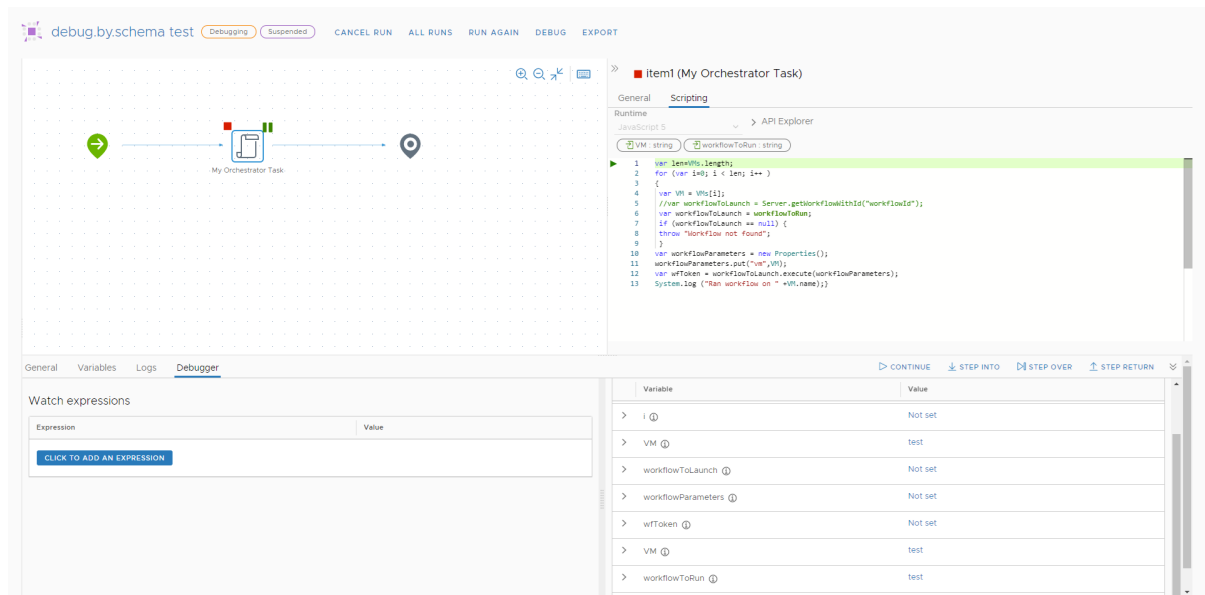
- 7 Insira os valores de parâmetro de entrada solicitados e clique em **Executar**.

A execução do fluxo de trabalho é iniciada e é suspensa quando o depurador atinge um elemento de esquema com um ponto de interrupção.

- 8 Quando estiver em um ponto de interrupção, selecione uma das seguintes opções:

Opção	Descrição
<b>Continuar</b>	Retoma a execução do fluxo de trabalho até que outro ponto de interrupção seja alcançado ou a execução do fluxo de trabalho seja concluída.
<b>Entrar em</b>	Entra na função de fluxo de trabalho atual. Se o depurador não puder ficar mais profundo na linha atual da função, ele executará uma operação <b>Passar sobre</b> .
<b>Passar sobre</b>	O depurador continua na próxima linha da função atual.
<b>Retorno de etapa</b>	O depurador vai para a linha que será executada quando a função atual retornar.

- 9 (Opcional) Na guia **Variáveis**, edite o valor das suas variáveis de fluxo de trabalho.



## Configurando um contêiner de Photon OS para pacotes Python

Dependendo do sistema operacional usado para compilar o script do Python, seus fluxos de trabalho ou ações podem falhar após a importação do arquivo ZIP relevante para o Cliente vRealize Orchestrator.

O sistema operacional do contêiner de tempo de execução usado para Python no vRealize Orchestrator é baseado no Photon 3.0. Os pacotes de script do Python compilados para outro sistema operacional, como o Linux, por exemplo, são incompatíveis com o contêiner de tempo de execução. Esse problema pode fazer com que o script do Python falhe quando você tentar usá-lo como parte de seus fluxos de trabalho ou ações do vRealize Orchestrator. Nesse cenário, você recebe a seguinte mensagem de erro em seus logs:

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

Para resolver esse problema, você deve instalar o pacote Python necessário em uma pasta de contêiner do Photon OS.

### Pré-requisitos

Instale o Docker. Consulte [Obter o Docker](#).

### Procedimentos

- 1 Navegue até a pasta principal do seu script do Python.
- 2 Crie um contêiner com a imagem base do Photon montando uma pasta de contêiner para a sua pasta principal.

---

**Observação** O script a seguir é um comando singular do Docker que você deve executar em sua totalidade para criar um contêiner adequado.

---

```
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 Instale o Python no contêiner.

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

- 4 Navegue até a pasta do contêiner que inclui o script do Python.
- 5 Adicione seu script e pacotes do Python.

---

**Observação** Instale os pacotes necessários para o seu script do Python na pasta `lib`.

---

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

- 6 Saia do contêiner e navegue até a pasta local que você montou no contêiner.
- 7 Compacte todos os arquivos e pastas relevantes em um arquivo ZIP.
- 8 Importe o arquivo ZIP para o Cliente vRealize Orchestrator e valide o script executando-o como parte de uma ação.