

# Utilizzo di VMware vRealize Orchestrator Client

06 OTTOBRE 2020

vRealize Orchestrator 8.2

È possibile trovare la documentazione tecnica più aggiornata sul sito Web di VMware all'indirizzo:

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

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

**VMware, Inc.**  
Centro Leoni Palazzo A  
Via Spadolini 5  
Ground Floor  
Milan, MI 20121  
tel: +39 02 30412700  
fax: +39 02 30412701  
[www.vmware.com/it](http://www.vmware.com/it)

Copyright © 2008-2020 VMware, Inc. Tutti i diritti sono riservati. [Informazioni sul copyright e sui marchi](#)

# Sommario

<b>1</b>	<b>Utilizzo di VMware vRealize Orchestrator Client</b>	<b>6</b>
<b>2</b>	<b>VMware vRealize Orchestrator Client</b>	<b>7</b>
	Dashboard dell'utilizzo di vRealize Orchestrator Client	8
	Organizzazione dei contenuti in vRealize Orchestrator Client	9
	Creazione di una cartella o una sottocartella in vRealize Orchestrator Client	10
<b>3</b>	<b>Configurazione di vRealize Orchestrator Client</b>	<b>13</b>
	Gestione di ruoli e gruppi di vRealize Orchestrator Client	13
	Assegnazione di ruoli in vRealize Orchestrator Client	14
	Creazione di gruppi in vRealize Orchestrator Client	15
	Configurazione dei ruoli di vRealize Orchestrator Client in vRealize Automation	16
	Cronologia delle versioni di un oggetto di vRealize Orchestrator	17
	Ripristino della versione precedente di un workflow	17
	Confronto visivo tra le versioni di un workflow	18
	Reimpostazione di uno stato precedente dell'inventario dei contenuti di vRealize Orchestrator con Git	19
<b>4</b>	<b>Casi d'uso di vRealize Orchestrator</b>	<b>21</b>
	Come integrare Amazon Web Services in vRealize Orchestrator tramite Python	21
	Creazione dello script Python iniziale	22
	Creazione di un'azione di Amazon Web Services	23
	Debug di un'azione di Amazon Web Services	24
	Aggiornamento dell'azione di Amazon Web Services	27
	Come utilizzare la diramazione Git per gestire l'inventario degli oggetti di vRealize Orchestrator	28
	Preparazione dell'ambiente di GitLab	29
	Configurazione di una connessione a un repository Git	30
	Push delle modifiche in un repository Git	31
	Come utilizzare moduli di terze parti per richiamare l'API del progetto di vRealize Automation	33
	Creazione di uno script Python che richiami l'API del progetto di vRealize Automation	33
	Creazione di uno script Node.js che richiami l'API del progetto di vRealize Automation	36
	Creazione di uno script di PowerShell che richiami l'API del progetto di vRealize Automation	38
<b>5</b>	<b>Gestione dei workflow</b>	<b>42</b>
	Workflow standard nella libreria dei workflow di vRealize Orchestrator	43
	Creazione di workflow in vRealize Orchestrator Client	43

Modifica di workflow e azioni dal workflow principale	44
Designer del modulo di input di vRealize Orchestrator	44
Creazione della finestra di dialogo dei parametri di input del workflow in vRealize Orchestrator Client	45
Proprietà dei parametri di input in vRealize Orchestrator Client	45
Utilizzo delle azioni per convalidare gli input dei workflow di vRealize Orchestrator	46
Richieste di interazione dell'utente in vRealize Orchestrator Client	48
Pianificazione dei workflow in vRealize Orchestrator Client	48
Modifica di un'attività pianificata in vRealize Orchestrator Client	49
<b>6 Gestione delle azioni</b>	<b>50</b>
Creazione di azioni in vRealize Orchestrator Client	50
Esecuzione e debug delle azioni	51
Esecuzione di azioni in vRealize Orchestrator Client	52
Azioni di debug in vRealize Orchestrator Client	52
Concetti di base per gli script Python, Node.js e PowerShell	53
Limiti di runtime per script Python, Node.js e PowerShell	55
<b>7 Gestione degli elementi di configurazione</b>	<b>56</b>
Creazione di elementi di configurazione in vRealize Orchestrator Client	56
<b>8 Gestione dei criteri</b>	<b>58</b>
Creazione e applicazione di criteri in vRealize Orchestrator Client	58
Elementi criterio in vRealize Orchestrator Client	59
Gestione delle esecuzioni dei criteri in vRealize Orchestrator Client	60
<b>9 Gestione degli elementi delle risorse</b>	<b>61</b>
<b>10 Gestione dei pacchetti</b>	<b>62</b>
Creazione di un pacchetto in vRealize Orchestrator Client	62
Esportazione di un pacchetto in vRealize Orchestrator Client	63
Importazione di un pacchetto in vRealize Orchestrator Client	64
<b>11 Risoluzione dei problemi in vRealize Orchestrator Client</b>	<b>66</b>
Dati metrici in vRealize Orchestrator Client	66
Profilatura dei workflow in vRealize Orchestrator Client	66
Utilizzo del dashboard del sistema vRealize Orchestrator	67
Utilizzo della riproduzione del token del workflow in vRealize Orchestrator Client	68
Convalida dei workflow di vRealize Orchestrator	69
Convalida di un workflow e correzione degli errori di convalida in vRealize Orchestrator Client	70
Debug degli script dei workflow in vRealize Orchestrator Client	71

[Workflow di debug in base all'elemento dello schema](#) 72

# Utilizzo di VMware vRealize Orchestrator Client

1

L'argomento *Utilizzo di VMware vRealize Orchestrator Client* fornisce informazioni sulle funzionalità di automazione del workflow e sul funzionamento di vRealize Orchestrator Client.

## Destinatari

Queste informazioni sono destinate agli amministratori di sistema esperti che cercano uno strumento con cui eseguire e gestire i workflow di vRealize Orchestrator.

# VMware vRealize Orchestrator Client

## 2

Utilizzare vRealize Orchestrator Client per gestire i servizi e gli oggetti di vRealize Orchestrator.

È possibile creare e gestire oggetti di vRealize Orchestrator tramite vRealize Orchestrator Client. vRealize Orchestrator Client è disponibile all'indirizzo [https://your\\_orchestrator\\_FQDN/orchestration-ui](https://your_orchestrator_FQDN/orchestration-ui).

### **Comunicazione con REST API**

vRealize Orchestrator Client viene eseguito nel server del Centro di controllo di vRealize Orchestrator. Il client comunica con la REST API di vRealize Orchestrator tramite un proxy REST.

### **Gestione dei workflow**

È possibile creare, modificare, pianificare, eseguire ed eliminare workflow in vRealize Orchestrator Client.

### **Gestione delle azioni**

È possibile creare, modificare ed eliminare azioni in vRealize Orchestrator Client. L'editor di azioni supporta il completamento automatico degli elementi script comuni inclusi in API Explorer di vRealize Orchestrator.

### **Gestione dei criteri**

È possibile creare, modificare, eseguire ed eliminare criteri in vRealize Orchestrator Client.

### **Gestione della configurazione**

È possibile creare, eseguire ed eliminare elementi di configurazione in vRealize Orchestrator Client.

### **Gestione delle risorse**

È possibile esportare, importare e aggiornare elementi delle risorse in vRealize Orchestrator Client.

### **Integrazione Git**

Creare un'integrazione con un repository Git e utilizzare l'integrazione per gestire lo sviluppo di workflow e altre istanze di vRealize Orchestrator in più distribuzioni. Vedere [Come utilizzare la diramazione Git per gestire l'inventario degli oggetti di vRealize Orchestrator](#).

### Dati metrici

Utilizzare il dashboard del sistema e la funzionalità di profilatura di vRealize Orchestrator Client per raccogliere dati metrici utili relativi all'ambiente e ai workflow di vRealize Orchestrator.

### Gestione dei pacchetti

È possibile creare, eliminare, esportare e importare pacchetti contenenti oggetti di vRealize Orchestrator tramite vRealize Orchestrator Client.

### Gestione delle autorizzazioni

Gli utenti con diritti di amministratore possono assegnare ruoli agli utenti in vRealize Orchestrator Client e aggiungerli ai gruppi.

### API Explorer

Esplorare i comandi API disponibili in vRealize Orchestrator Client.

Questo capitolo include i seguenti argomenti:

- [Dashboard dell'utilizzo di vRealize Orchestrator Client](#)
- [Organizzazione dei contenuti in vRealize Orchestrator Client](#)

## Dashboard dell'utilizzo di vRealize Orchestrator Client

Il dashboard di vRealize Orchestrator Client offre uno strumento utile per il monitoraggio, la gestione e la risoluzione dei problemi dei workflow di vRealize Orchestrator Client.

Le informazioni nel dashboard di vRealize Orchestrator Client sono distribuite in cinque pannelli.

Finestra	Descrizione
Esecuzioni workflow	Include dati visivi relativi al numero di esecuzioni di workflow in corso, in attesa e non riuscite.
Workflow preferiti	Visualizza i workflow aggiunti ai Preferiti.
In attesa di input	Visualizza le esecuzioni dei workflow in sospeso che richiedono ulteriore interazione dell'utente. Questi workflow vengono visualizzati anche nel menu delle notifiche nell'angolo in alto a destra dell'interfaccia utente.
Esecuzioni dei workflow recenti	Consente di gestire le esecuzioni dei workflow recenti. Visualizza il nome, lo stato, la data di inizio e la data di fine dell'esecuzione del workflow.
Richiede attenzione	Visualizza le esecuzioni dei workflow non riuscite e le metriche delle prestazioni delle esecuzioni dei workflow.



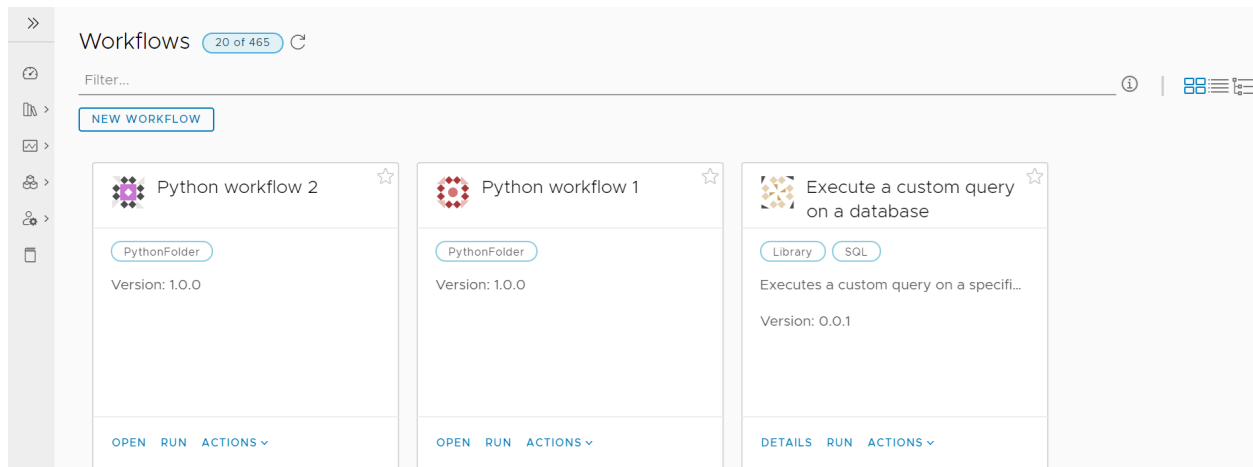
## Organizzazione dei contenuti in vRealize Orchestrator Client

È possibile gestire il modo in cui l'inventario degli oggetti di vRealize Orchestrator viene visualizzato in vRealize Orchestrator Client.

vRealize Orchestrator Client supporta tre tipi di visualizzazione diversi per oggetti come workflow, azioni, criteri, risorse e configurazioni: Visualizzazione scheda, Visualizzazione elenco e Visualizzazione albero. È possibile modificare il tipo di visualizzazione corrente nell'angolo in alto a destra nella pagina.

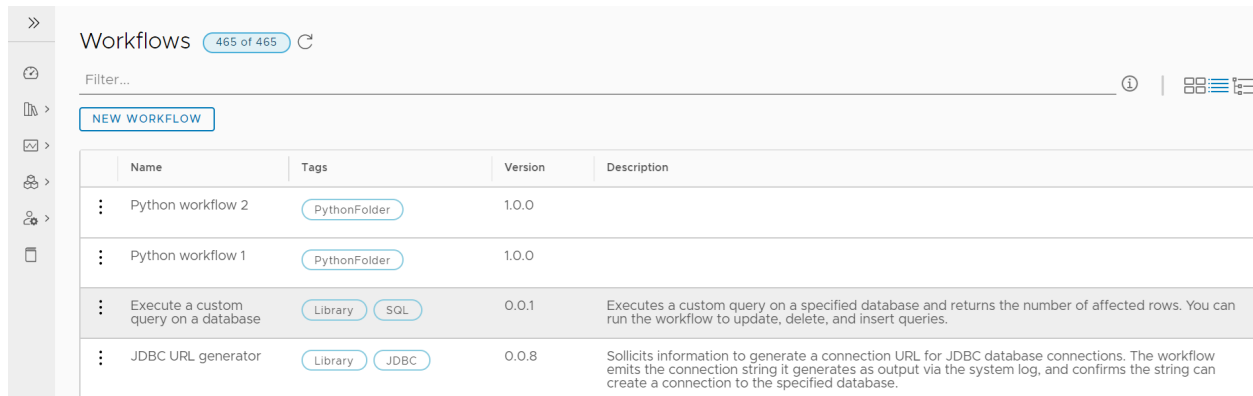
### Visualizzazione scheda

La Visualizzazione scheda è il tipo di visualizzazione predefinito utilizzato in vRealize Orchestrator Client. Le informazioni sul singolo oggetto dell'inventario, ad esempio un workflow, vengono visualizzate in un elemento scheda separato.



### Visualizzazione elenco

La Visualizzazione elenco include informazioni sugli oggetti di vRealize Orchestrator organizzati in un elenco. Per ulteriori informazioni sulle azioni che è possibile eseguire sull'oggetto, fare clic sull'icona con i puntini verticali a sinistra dell'oggetto.



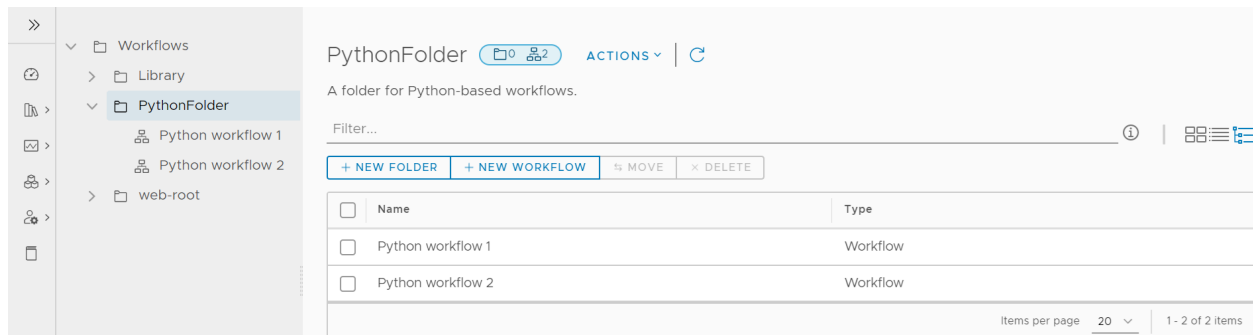
## Visualizzazione albero

È possibile organizzare l'inventario degli oggetti in cartelle gerarchiche nella Visualizzazione albero. Ogni tipo di oggetto di vRealize Orchestrator ha una cartella di livello root. Nella cartella root non è possibile creare nuovi oggetti, ad esempio workflow. È necessario creare cartelle separate organizzate nella cartella root. Ogni cartella include strumenti che consentono di gestire i contenuti, ad esempio un filtro dei contenuti.

**Nota** Ogni cartella ha un filtro dei contenuti separato. Non è possibile filtrare i contenuti di diverse cartelle contemporaneamente.

Per ulteriori informazioni sulle cartelle, vedere [Creazione di una cartella o una sottocartella in vRealize Orchestrator Client](#).

**Nota** Quando si seleziona un oggetto dalla Visualizzazione albero, viene aperto in modalità di sola lettura. Per modificare il contenuto dell'oggetto, ad esempio le variabili del workflow o lo schema del workflow, fare clic su **Modifica** nel menu delle opzioni in alto.



## Creazione di una cartella o una sottocartella in vRealize Orchestrator Client


Organizzare gli oggetti di vRealize Orchestrator utilizzando una struttura di cartelle gerarchica.

È possibile creare cartelle e sottocartelle per organizzare i seguenti tipi di oggetti di vRealize Orchestrator:

- Workflow
- Azioni
- Criteri
- Elementi di configurazione
- Elementi di risorse

### Procedura

- 1 Accedere a vRealize Orchestrator Client.


- 2 Nel riquadro di spostamento a sinistra, selezionare una pagina di oggetti, ad esempio **Workflow**.
- 3 In alto a destra, selezionare l'icona della Visualizzazione albero ()
- 4 (Facoltativo) Per creare una sottocartella, selezionare una cartella principale dalla Visualizzazione albero a sinistra.
- 5 Fare clic su **Nuova cartella**.
- 6 Immettere un nome e una descrizione e fare clic su **Salva**.
- 7 Aggiungere oggetti o sottocartelle alla cartella appena creata.
- 8 (Facoltativo) Per modificare il nome della cartella, selezionare **Azioni > Modifica**.

## Spostamento di oggetti e cartelle in vRealize Orchestrator Client

È possibile riorganizzare i contenuti di vRealize Orchestrator spostandoli in un'altra cartella.

Non è possibile spostare le azioni tra moduli di azioni o spostare gli oggetti in una cartella root. La cartella root include le cartelle e le sottocartelle di oggetti principali, ma non può essere utilizzata per archiviare gli oggetti.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Nel riquadro di spostamento a sinistra, selezionare una pagina di oggetti, ad esempio **Workflow**.
- 3 In alto a destra, selezionare l'icona della Visualizzazione albero ()
- 4 Espandere la Visualizzazione albero e selezionare l'oggetto o la cartella che si desidera spostare.
- 5 Trascinare l'oggetto o la cartella nella nuova cartella principale.

---

**Nota** È inoltre possibile spostare gli oggetti in nuove cartelle direttamente dall'editor di oggetti. Nella scheda **Riepilogo**, fare clic su **Seleziona cartella** e selezionare la nuova cartella principale per l'oggetto. Un'altra opzione di spostamento consiste nel selezionare gli oggetti dalla tabella nella pagina della cartella. Questa opzione è utile per eseguire operazioni di spostamento batch che includono più oggetti di vRealize Orchestrator.

---

## Eliminazione di una cartella o una sottocartella in vRealize Orchestrator Client

Eliminare le cartelle o le sottocartelle obsolete da vRealize Orchestrator Client.

Non è possibile eliminare la cartella di livello root corrispondente di ogni tipo di oggetto di vRealize Orchestrator.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.

- 2 Nel riquadro di spostamento a sinistra, selezionare una pagina di oggetti, ad esempio **Workflow**.

- 3 In alto a destra, selezionare l'icona della Visualizzazione albero ()

- 4 Selezionare la casella di controllo accanto alla cartella che si desidera eliminare.

---

**Nota** Per eliminare una sottocartella, selezionare la cartella principale nella Visualizzazione albero, quindi selezionare la casella di controllo.

---

- 5 Fare clic su **Elimina**.

- 6 Se la cartella selezionata è vuota.

- a Confermare che si desidera eliminare la cartella.
- b Fare clic su **Elimina**.

- 7 Se la cartella selezionata contiene oggetti o sottocartelle di vRealize Orchestrator Client.

- a Confermare che si desidera eliminare la cartella.
- b Fare clic su **Elimina**.

Viene visualizzato il messaggio Impossibile eliminare l'elemento 'nome\_cartella': la cartella 'nome\_cartella' non è vuota.

- c Per eliminare la cartella e tutto il suo contenuto, fare clic su **Forza eliminazione**.
- d Confermare che si desidera eliminare la cartella e fare clic su **Elimina**.

---

**Nota** È inoltre possibile eseguire un'eliminazione in batch selezionando più oggetti nella tabella inclusa nel menu della cartella.

---

# Configurazione di vRealize Orchestrator Client

## 3

Per sfruttare appieno le funzionalità di vRealize Orchestrator Client, è necessario configurare le autorizzazioni utente e imparare a utilizzare la cronologia delle versioni per gestire gli oggetti.

Questo capitolo include i seguenti argomenti:

- [Gestione di ruoli e gruppi di vRealize Orchestrator Client](#)
- [Cronologia delle versioni di un oggetto di vRealize Orchestrator](#)

## Gestione di ruoli e gruppi di vRealize Orchestrator Client

Un amministratore può utilizzare vRealize Orchestrator Client per impostare ruoli utente e autorizzazioni di gruppi per le funzionalità e i contenuti di vRealize Orchestrator.

Dopo l'autenticazione dell'istanza di vRealize Orchestrator, l'amministratore può impostare le autorizzazioni che controllano l'accesso alle funzionalità e ai contenuti. Le autorizzazioni in vRealize Orchestrator Client sono suddivise in autorizzazioni dei gruppi e di gestione dei ruoli. Con la gestione dei ruoli, è possibile controllare quali funzionalità di vRealize Orchestrator Client gli utenti possono visualizzare e utilizzare. Con le autorizzazioni dei gruppi, è possibile controllare quali contenuti di vRealize Orchestrator Client gli utenti possono visualizzare e utilizzare. L'accesso ai contenuti regolato dalle autorizzazioni dei gruppi include workflow, azioni, criteri, elementi di configurazione ed elementi di risorse. È possibile utilizzare i gruppi per organizzare gli utenti in progetti comuni. È ad esempio possibile creare un gruppo che includa gli utenti che collaborano allo sviluppo di un plug-in di vRealize Orchestrator personalizzato.

---

**Nota** L'accesso ai contenuti di vRealize Orchestrator preconfigurati, come i workflow e le azioni standard, viene condiviso tra tutti gli utenti, a meno che le autorizzazioni dei gruppi non specifichino una configurazione diversa.

---

### Ruoli

La gestione dei ruoli sul lato client è disponibile solo per le istanze di vRealize Orchestrator autenticate con vSphere che utilizzano una licenza di vRealize Automation. Per le distribuzioni che utilizzano un'autenticazione di vRealize Automation, è necessario utilizzare la funzionalità Gestione identità e accessi di vRealize Automation. Vedere [Configurazione dei ruoli di vRealize Orchestrator Client in vRealize Automation](#).

Ruolo	Descrizione
<b>Amministratore</b>	<p>Può accedere a tutte le funzionalità e i contenuti di vRealize Orchestrator Client, inclusi i contenuti creati da gruppi specifici. È responsabile dell'impostazione dei ruoli utente, della creazione e dell'eliminazione di gruppi, nonché dell'aggiunta di utenti ai gruppi.</p> <p><b>Nota</b> Gli amministratori tenant dell'ambiente di vRealize Automation utilizzato per l'autenticazione di vRealize Orchestrator dispongono di diritti di <b>Amministratore</b> per impostazione predefinita.</p>
<b>Designer workflow</b>	<p>Può creare, eseguire, modificare ed eliminare i propri contenuti di vRealize Orchestrator Client. Può aggiungere i propri contenuti al gruppo assegnato. Non può accedere alle funzionalità di amministrazione e risoluzione dei problemi di vRealize Orchestrator Client.</p>

**Nota** Gli utenti di vRealize Automation che non dispongono di un ruolo predefinito possono comunque accedere a vRealize Orchestrator Client, ma hanno accesso limitato alle funzionalità del client. Se fanno parte di un gruppo, questi utenti possono visualizzare ed eseguire i contenuti associati a tale gruppo.

## Gruppi

Le autorizzazioni dei gruppi in vRealize Orchestrator Client possono essere utilizzate per connettere più utenti che collaborano a un'attività di vRealize Orchestrator comune, ad esempio lo sviluppo di un plug-in personalizzato.

Autorizzazioni degli utenti dei gruppi	Descrizione
<b>Esegui e modifica</b>	<p>Disponibile solo per le istanze di vRealize Orchestrator che utilizzano una licenza di vRealize Automation.</p> <p>Consente di creare, modificare, aggiungere ed eseguire oggetti di vRealize Orchestrator da utilizzare nel gruppo.</p>
<b>Esegui</b>	<p>Consente di visualizzare ed eseguire oggetti di vRealize Orchestrator inclusi nel gruppo.</p>

**Nota** Le autorizzazioni del gruppo sono collegati al sistema di gestione dei ruoli in vRealize Orchestrator Client. Ad esempio, gli utenti che non dispongono di un ruolo predefinito possono avere le autorizzazioni **Esegui e modifica**, ma possono solo visualizzare ed eseguire i propri contenuti o i contenuti del gruppo senza poter creare, modificare e aggiungere contenuti.

## Assegnazione di ruoli in vRealize Orchestrator Client

Un amministratore può aggiungere utenti in vRealize Orchestrator Client e impostare le funzionalità che possono visualizzare e utilizzare.

La gestione dei ruoli controlla l'accesso degli utenti dal provider di identità di vRealize Orchestrator alle funzionalità di vRealize Orchestrator Client. La gestione dei ruoli riguarda sia l'interfaccia utente di vRealize Orchestrator Client sia la funzionalità dell'API.

---

**Nota** La gestione dei ruoli sul lato client è disponibile solo per le istanze di vRealize Orchestrator autenticate con vSphere che utilizzano una licenza di vRealize Automation. Per informazioni sull'assegnazione di ruoli a vRealize Orchestrator autenticato con vRealize Automation, vedere [Configurazione dei ruoli di vRealize Orchestrator Client in vRealize Automation](#).

---

#### Procedura

- 1 Accedere a vRealize Orchestrator Client come amministratore.
- 2 Passare a **Amministrazione > Gestione ruoli**.
- 3 Fare clic su **Aggiungi**.
- 4 Cercare l'utente o il gruppo che si desidera aggiungere in vRealize Orchestrator Client.
- 5 Selezionare il ruolo dell'utente. Per ulteriori informazioni sui ruoli, vedere [Gestione di ruoli e gruppi di vRealize Orchestrator Client](#).
- 6 Fare clic su **Salva**.

## Creazione di gruppi in vRealize Orchestrator Client

Un amministratore può utilizzare i gruppi per impostare i contenuti di vRealize Orchestrator a cui gli utenti possono accedere e che possono visualizzare in vRealize Orchestrator Client.

È possibile utilizzare vRealize Orchestrator Client per impostare le autorizzazioni di gruppo per workflow, azioni, criteri, elementi di configurazione, elementi di risorse e pacchetti di vRealize Orchestrator.

---

**Nota** Gli utenti delle istanze di vRealize Orchestrator autenticati in vSphere possono disporre solo dell'autorizzazione di gruppo **Esegui**.

---

#### Procedura

- 1 Accedere a vRealize Orchestrator Client come amministratore.
- 2 Passare a **Amministrazione > Gruppi**.
- 3 Fare clic su **Nuovo gruppo**.
- 4 Nella scheda **Riepilogo**, aggiungere un nome e una descrizione per il gruppo.
- 5 Nella scheda **Utenti**, fare clic su **Aggiungi**.
  - a Cercare l'utente che si desidera aggiungere al gruppo.
  - b Assegnare autorizzazioni di gruppo all'utente.
  - c Fare clic su **Aggiungi**.

- 6 Nella scheda **Elementi**, aggiungere oggetti di vRealize Orchestrator al gruppo.

---

**Nota** È inoltre possibile aggiungere un oggetto ai gruppi esistenti quando l'oggetto viene creato in vRealize Orchestrator Client. Per aggiungere l'oggetto, selezionare il gruppo nel menu a discesa **Accessibile da** della scheda **Riepilogo/Generale** dell'editor di oggetti.

---

- 7 Fare clic su **Salva**.

## Configurazione dei ruoli di vRealize Orchestrator Client in vRealize Automation

È possibile assegnare ruoli di servizio per vRealize Orchestrator Client nella pagina **Gestione identità e accessi** di vRealize Automation. I ruoli di servizio possono essere assegnati per le istanze di vRealize Orchestrator Client incorporate e le istanze di vRealize Orchestrator autonome autenticate con vRealize Automation.

I ruoli di servizio di vRealize Orchestrator consentono di gestire le funzionalità dell'istanza di vRealize Orchestrator Client incorporata a cui gli utenti possono accedere. Per ulteriori informazioni sui ruoli di vRealize Orchestrator, vedere [Gestione di ruoli e gruppi di vRealize Orchestrator Client](#).

---

**Nota** Le istanze di vRealize Orchestrator autonome autenticate con vSphere che utilizzano una licenza di vRealize Automation possono assegnare ruoli direttamente in vRealize Orchestrator Client. Vedere [Assegnazione di ruoli in vRealize Orchestrator Client](#).

---

### Prerequisiti

- Verificare che gli utenti e i gruppi appropriati vengano importati da un'istanza di vIDM valida.
- Prima di assegnare un ruolo di servizio di vRealize Orchestrator all'utente, verificare che all'utente sia assegnato un ruolo di organizzazione in vRealize Automation. Vedere *Amministrazione di utenti e gruppi in vRealize Automation* in *Amministrazione di vRealize Automation*.

### Procedura

- 1 Nel menu a discesa in alto a destra, selezionare l'opzione **Gestione identità e accessi**.
- 2 Nella scheda **Utenti attivi**, cercare l'indirizzo e-mail dell'utente che si desidera assegnare a vRealize Orchestrator.
- 3 Selezionare la casella di controllo accanto all'utente e fare clic su **Modifica ruoli**.
- 4 Fare clic su **Aggiungi accesso servizio**.
- 5 Nel menu a discesa a sinistra, selezionare **Orchestrator**.
- 6 Nel menu a discesa a destra, selezionare il ruolo che si desidera assegnare all'utente.
- 7 Fare clic su **Salva**.



## Cronologia delle versioni di un oggetto di vRealize Orchestrator

vRealize Orchestrator Client mantiene un record della cronologia delle versioni per ogni oggetto di vRealize Orchestrator. Utilizzando la cronologia delle versioni, è possibile confrontare diverse versioni di oggetti di vRealize Orchestrator e ripristinare una versione precedente.

vRealize Orchestrator crea un record della cronologia delle versioni per ogni oggetto di vRealize Orchestrator quando si salva l'oggetto. Le modifiche successive apportate agli oggetti di vRealize Orchestrator creano un nuovo record della cronologia delle versioni. I record della cronologia delle versioni precedenti vengono conservati e possono essere utilizzati per tenere traccia delle modifiche apportate all'oggetto, nonché ripristinare una versione precedente dell'oggetto. Se si ripristina una versione precedente di un oggetto, viene creato un nuovo record della cronologia delle versioni.

vRealize Orchestrator Client tiene traccia della cronologia delle versioni dei seguenti oggetti di vRealize Orchestrator:

- Workflow
- Azioni
- Pacchetti
- Criteri
- Elementi di configurazione

È possibile accedere alla cronologia delle versioni di un oggetto dalla scheda **Cronologia versioni** della pagina dell'editor di oggetti. Se si sta tentando di modificare un oggetto contemporaneamente a un altro utente, è possibile che si verifichi un conflitto di unione. Per risolvere il conflitto di unione, fare clic su **Risolvi** a destra del messaggio di errore. Nella finestra **Risoluzione di conflitti** sono disponibili tre opzioni:

- **Usa il LORO.** Consente di risolvere il conflitto di unione utilizzando le modifiche apportate dall'altro utente.
- **Usa il NOSTRO.** Consente di risolvere il conflitto di unione utilizzando le proprie modifiche.
- **Risolvi.** Consente di risolvere il conflitto di unione modificando il modello di modifica visualizzato. Se il modello fornito non è valido, questa opzione non è disponibile.

## Ripristino della versione precedente di un workflow

È possibile ripristinare una versione precedentemente salvata di un workflow.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Workflow** e selezionare un workflow.
- 3 Selezionare la scheda **Cronologia versioni**.

- 4 Per visualizzare un confronto tra le versioni, selezionare una versione del workflow, quindi scegliere un'altra versione nel menu a discesa **Diff. rispetto a**.

In una finestra vengono visualizzate le differenze tra la versione corrente del workflow e la versione del workflow selezionata.

- 5 Per ripristinare un'altra versione del workflow, fare clic su **Ripristina**.

Per il workflow viene ripristinato lo stato della versione selezionata.

---

**Nota** È inoltre possibile ripristinare una versione del workflow dalla visualizzazione dello strumento di differenza grafica. Vedere [Confronto visivo tra le versioni di un workflow](#).

---

## Confronto visivo tra le versioni di un workflow

È possibile confrontare le modifiche tra le versioni del workflow con lo strumento che rileva le differenze grafiche.

Per impostazione predefinita, in Cronologia versioni di vRealize Orchestrator vengono visualizzate le differenze tra le versioni del workflow in un modulo YAML. È inoltre possibile eseguire un confronto visivo tra versioni del workflow diverse. È possibile visualizzare le modifiche apportate:

- Nelle informazioni generali sul workflow, ad esempio il numero di versione e la descrizione del workflow.
- Nelle variabili utilizzate nel workflow.
- Nei parametri di input e output del workflow.
- Nello schema del workflow.

### Prerequisiti

Creare un workflow.

### Procedura

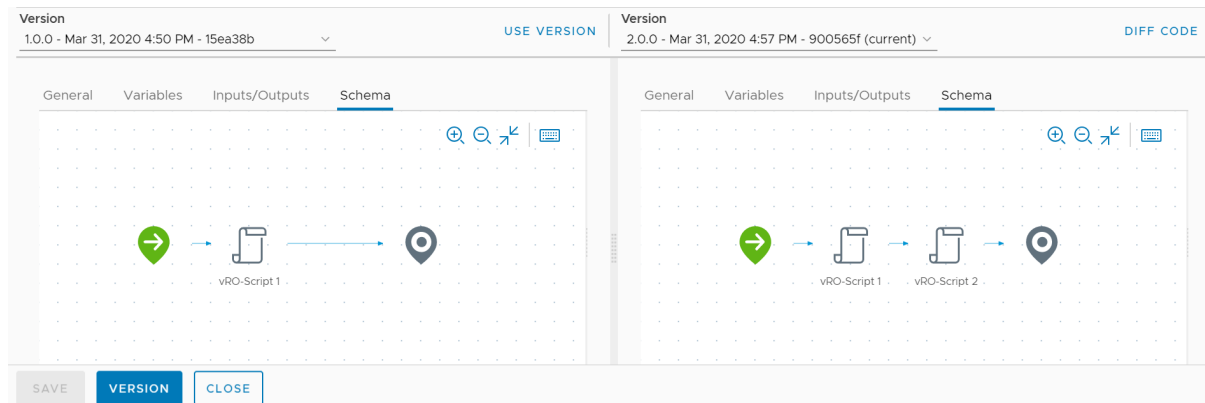
- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Workflow** e selezionare uno dei workflow.
- 3 Modificare il contenuto del workflow.

È ad esempio possibile aggiungere un ulteriore elemento **Attività gestibile tramite script** nella scheda **Schema**.

- 4 Fare clic su **Salva**.
- 5 Selezionare la scheda **Cronologia versioni**.

## 6 In alto a destra, selezionare **Diff. visivamente**.

Ora è possibile eseguire un confronto visivo tra due versioni del workflow selezionate. È possibile selezionare le versioni da confrontare nel menu a discesa **Versione**.



## 7 (Facoltativo) È possibile ripristinare un'altra versione per un workflow selezionando **Usa versione**.

# Reimpostazione di uno stato precedente dell'inventario dei contenuti di vRealize Orchestrator con Git

Utilizzando un commit Git precedente, è possibile reimpostare uno stato precedente dei contenuti di vRealize Orchestrator.

È possibile reimpostare uno stato precedente dei contenuti di vRealize Orchestrator selezionando un commit specifico.

### Prerequisiti

- Configurare una connessione a un repository GitHub o GitLab. Vedere [Configurazione di una connessione a un repository Git](#).
- Eseguire il push di un set di modifiche locale nel repository Git configurato.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Amministrazione > Cronologia Git**.
- 3 Selezionare il set di modifiche che si desidera reimpostare e fare clic su **Reimposta su questo**.
- 4 Confermare che si desidera eseguire la reimpostazione su questo commit specifico e fare clic su **OK**.

L'inventario dei contenuti di vRealize Orchestrator viene reimpostato sullo stato specificato nel commit. Per i contenuti di vRealize Orchestrator pertinenti viene ripristinata una versione precedente. I contenuti che non esistevano quando è stato eseguito il push del commit vengono rimossi dall'inventario.

### Operazioni successive

Per ripristinare per l'inventario di vRealize Orchestrator lo stato più recente salvato nel repository Git, eseguire un comando Pull dalla finestra **Cronologia Git**.

# Casi d'uso di vRealize Orchestrator

# 4

Questi casi d'uso illustrano parte della funzionalità della piattaforma di vRealize Orchestrator.

I casi d'uso includono solo valori di esempio. La struttura e le convenzioni di denominazione del proprio ambiente possono variare.

Questo capitolo include i seguenti argomenti:

- [Come integrare Amazon Web Services in vRealize Orchestrator tramite Python](#)
- [Come utilizzare la diramazione Git per gestire l'inventario degli oggetti di vRealize Orchestrator](#)
- [Come utilizzare moduli di terze parti per richiamare l'API del progetto di vRealize Automation](#)

## Come integrare Amazon Web Services in vRealize Orchestrator tramite Python

Questo caso d'uso di vRealize Orchestrator illustra un esempio di come sia possibile utilizzare Python per espandere le funzionalità della distribuzione di vRealize Orchestrator.

A partire da vRealize Orchestrator 8.1, è possibile utilizzare tre nuovi runtime nelle azioni e negli script dei workflow.

- Python 3.7
- Node.js 12
- PowerCLI 11/Powershell 6.2
- PowerCLI 12/Powershell 7

---

**Nota** Il runtime PowerCLI include PowerShell e i seguenti moduli: VMware.PowerCLI, PowerNSX, PowervRA.

---

**Importante** È possibile utilizzare i nuovi runtime solo se la distribuzione di vRealize Orchestrator usa una licenza di vRealize Automation.

---

Questo caso d'uso illustra in che modo è possibile creare uno script Python che richiami le istanze di EC2 in Amazon Web Services (AWS).

---

**Importante** Prima di iniziare a sviluppare lo script personalizzato, assicurarsi di conoscere i concetti di base dell'utilizzo degli script Python, Node.js e PowerShell in vRealize Orchestrator. Vedere [Concetti di base per gli script Python, Node.js e PowerShell](#) .

---

## Procedura

### 1 Creazione dello script Python iniziale

Sulla macchina locale, creare lo script Python e inserirlo in un pacchetto insieme a una libreria Boto3 come cartella ZIP.

### 2 Creazione di un'azione di Amazon Web Services

Creare un'azione di vRealize Orchestrator che utilizzi lo script Python.

### 3 Debug di un'azione di Amazon Web Services

Nella versione originale dello script Python è stato integrato intenzionalmente un errore che consente di imparare come eseguire il debug dello script.

### 4 Aggiornamento dell'azione di Amazon Web Services

Importare lo script Python aggiornato ed eseguire di nuovo l'azione.

## Creazione dello script Python iniziale

Sulla macchina locale, creare lo script Python e inserirlo in un pacchetto insieme a una libreria Boto3 come cartella ZIP.

## Prerequisiti

- Scaricare e installare Python 3. Vedere la [pagina dei download di Python](#).
- Scaricare e installare Visual Studio Code. Vedere la [pagina di download di Visual Studio Code](#).
- Verificare di aver installato l'estensione Python per Visual Studio Code. Vedere [Visual Studio Marketplace](#).

## Procedura

- 1 Sulla macchina locale, creare una cartella vro-python-aws e installare l'SDK Python Boto3 in tale cartella.

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

- 2 Aprire un editor e creare lo script Python principale. Per questo caso d'uso, si utilizza 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)
```

Questo script Python elenca tutte le istanze di EC2 in esecuzione in una determinata regione.

- 3 Salvare lo script creato come file `main.py` nella cartella `vro-python-aws`.
- 4 Accedere a un'interfaccia della riga di comando.
- 5 Passare alla cartella `vro-python-aws`.

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

- 6 Creare un pacchetto ZIP contenente lo script Python.

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

**Nota** È inoltre possibile creare il pacchetto ZIP utilizzando uno strumento dell'utilità ZIP, ad esempio 7-Zip.

## Risultati

Lo script Python di base è stato creato e preparato per l'importazione nella distribuzione di vRealize Orchestrator.

## Creazione di un'azione di Amazon Web Services

Creare un'azione di vRealize Orchestrator che utilizzi lo script Python.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Azioni**.
- 3 Fare clic su **Nuova azione**.
- 4 Nella scheda **Generale**, immettere un nome, un modulo e un numero di versione per l'azione.
- 5 Nella scheda **Script**, selezionare **Python 3.7** come runtime e **Zip** come tipo di script.
- 6 Fare clic su **Importa**.

- 7 Passare alla cartella vro-python-aws e selezionare il pacchetto ZIP contenente lo script Python.
- 8 Nella casella di testo **Gestore voci**, immettere **main.handler**.

---

**Nota** Il gestore delle voci dell'azione si basa sullo script principale nel pacchetto ZIP importato. Poiché lo script principale si trova in un file denominato `main.py` e una funzione denominata **handler**, il gestore delle voci deve essere **main.handler**. Se per il file dello script principale è stato specificato un nome diverso, modificare il valore del gestore delle voci di conseguenza.

---

- 9 Salvare l'azione e fare clic su **Esegui**.

Durante l'esecuzione dell'azione si verifica un errore.

- 10 Selezionare la scheda **Registri**.

Nei registri dell'esecuzione dell'azione viene visualizzato il messaggio di errore "botocore.exceptions.NoRegionError: è necessario specificare una regione.". Si tratta di un comportamento previsto, perché lo script Python iniziale non definisce una regione.

#### Operazioni successive

Eseguire il debug dello script Python. Vedere [Debug di un'azione di Amazon Web Services](#).

## Debug di un'azione di Amazon Web Services

Nella versione originale dello script Python è stato integrato intenzionalmente un errore che consente di imparare come eseguire il debug dello script.

#### Prerequisiti

Accedere all'account Amazon Web Services (AWS) e creare un utente IAM specifico per lo scenario di questo caso d'uso. Vedere [Creazione di un utente IAM nell'account AWS](#). L'utente IAM deve disporre delle autorizzazioni seguenti:

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



## Procedura

### 1 Preparare vRealize Orchestrator Appliance.

---

**Attenzione** Non eseguire il debug degli script nella distribuzione di produzione di vRealize Orchestrator. Eseguire il debug in una distribuzione di vRealize Orchestrator con un solo nodo, utilizzata a scopo di sviluppo e test.

---

- a Accedere alla riga di comando di vRealize Orchestrator Appliance tramite SSH come **root**.
- b Eseguire il comando `vracli dev tools`.
- c Viene richiesto di confermare se si desidera continuare. Immettere **yes** per continuare, oppure **no** per annullare l'operazione.

---

**Importante** Se si esegue il comando `vracli dev tools`, vengono aperte le porte necessarie per eseguire il debug dello script Python. Durante il processo di debug, è necessario lasciare aperta la sessione SSH corrente.

---

### 2 Avviare la configurazione del debug.

- a Accedere a vRealize Orchestrator Client.
- b Aprire l'azione di AWS e fare clic su **Debug**.  
Il processo di debug inizia e l'esecuzione dell'azione viene sospesa.
- c Selezionare la scheda **Configurazione debug**.  
La scheda contiene una configurazione .json che è possibile collegare in remoto all'IDE per eseguire il debug dello script Python.
- d Copiare manualmente il contenuto della configurazione oppure fare clic su **Copia negli Appunti**.

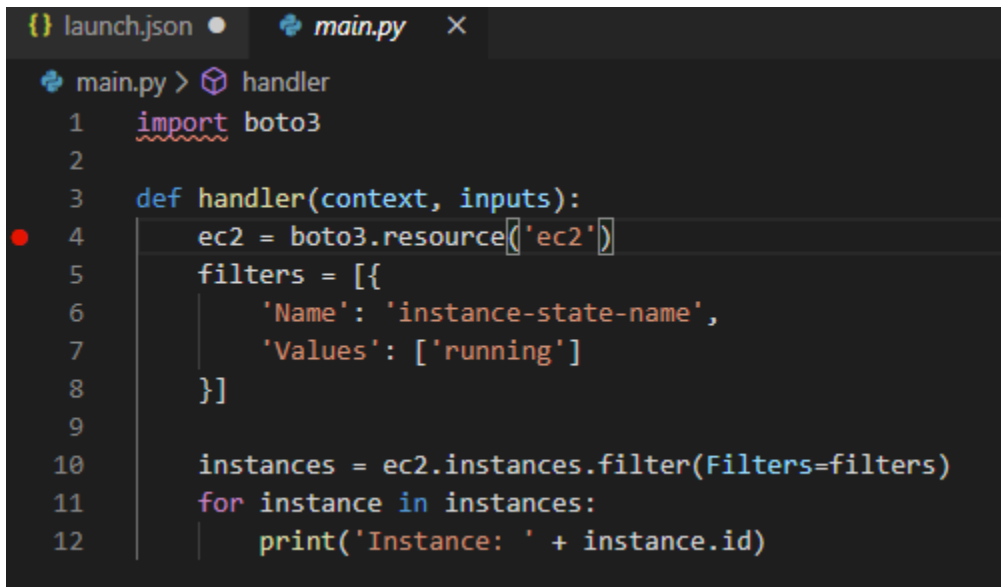
### 3 Eseguire il debug dello script Python.

- a Aprire Visual Studio Code.
- b Aprire la cartella `vro-python-aws`.
- c Nel riquadro di spostamento superiore, selezionare **Esegui > Apri configurazioni**.
- d Selezionare **File Python**.

- e Lasciare gli attributi "version" e "configuration" nelle posizioni correnti e incollare il contenuto della configurazione .json copiato da vRealize Orchestrator Client. Il file launch.json generato deve essere simile al seguente:

```
{
  "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 Selezionare il file di script main.py e aggiungere un punto di interruzione alla riga `ec2 = boto3.resource('ec2')`.



```
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 Nel riquadro di spostamento superiore, selezionare **Esegui > Avvia debug**.

- h Quando il debugger raggiunge il punto di interruzione, eseguire un'operazione stepover.  
L'esecuzione del debug indica che nello script Python manca una regione specifica e la chiave di accesso di AWS.
- i Tornare alla sessione di vRealize Orchestrator Appliance aperta e premere **Invio** per chiudere le porte aperte per questa sessione di debug.

#### 4 Aggiungere le informazioni mancanti nello script Python.

- a In Visual Studio Code, creare un file denominato `awsconfig` che contenga la chiave di accesso di AWS dell'utente IAM e la regione di AWS in cui si desidera eseguire il ping con lo script Python.

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

- b Salvare `awsconfig` come file di configurazione (`.cfg`) nella cartella `vro-python-aws`.
- c Aprire il file `main.py` e modificarlo in modo che la libreria `boto3` possa utilizzare il file `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 Creare un nuovo pacchetto ZIP contenente il file `main.py`, il file `awsconfig.cfg` e la libreria `boto3`.

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

**Nota** È inoltre possibile creare il pacchetto ZIP utilizzando uno strumento dell'utilità ZIP, ad esempio 7-Zip.

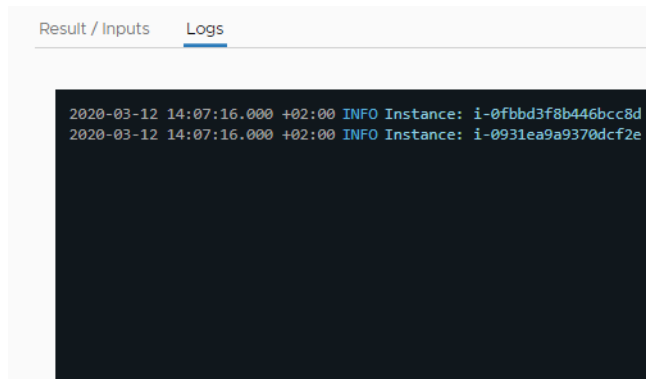
## Aggiornamento dell'azione di Amazon Web Services

Importare lo script Python aggiornato ed eseguire di nuovo l'azione.

## Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Azioni** e selezionare l'azione Amazon Web Services (AWS) originale.
- 3 (Facoltativo) Nella scheda **Generale**, modificare il numero di versione.
- 4 Rimuovere il vecchio pacchetto ZIP e fare clic su **Importa**.
- 5 Selezionare il pacchetto ZIP aggiornato.
- 6 Salvare l'azione e fare clic su **Esegui**.
- 7 Una volta completata l'esecuzione dell'azione, selezionare la scheda **Registri**.

Nei registri sono visualizzate le istanze di EC2 sottoposte a query dall'azione.



## Operazioni successive

Creare un workflow di vRealize Orchestrator che utilizzi l'azione AWS aggiornata come **Elemento azione**.

## Come utilizzare la diramazione Git per gestire l'inventario degli oggetti di vRealize Orchestrator

Utilizzare la diramazione per organizzare il modo in cui i contenuti di vRealize Orchestrator vengono gestiti nel repository Git.

Utilizzando Git, è possibile aumentare la flessibilità per gli sviluppatori di vRealize Orchestrator fornendo un repository centralizzato. È ad esempio possibile utilizzare Git per gestire lo sviluppo del workflow in più ambienti di vRealize Orchestrator.

---

**Nota** Per utilizzare Git per gestire l'inventario degli oggetti, la distribuzione di vRealize Orchestrator deve utilizzare una licenza di vRealize Automation. Per ulteriori informazioni, vedere *Abilitazione delle funzionalità di vRealize Orchestrator tramite licenze in Installazione e configurazione di vRealize Orchestrator*.

---

A partire da vRealize Orchestrator 8.1, è possibile eseguire il push e il pull di oggetti verso rami e da rami. È possibile utilizzare la diramazione per gestire lo sviluppo di gruppi specifici di oggetti di vRealize Orchestrator, prima che vengano uniti nuovamente al ramo principale.

In questo caso d'uso, si utilizza un progetto GitLab per gestire oggetti di vRealize Orchestrator che utilizzano il runtime Python. Questo caso d'uso rappresenta un esempio della funzionalità di Git in vRealize Orchestrator, ma non rappresenta i limiti dell'ambito della funzionalità.

---

**Nota** Se si ha più familiarità con GitHub, è possibile utilizzare un repository di GitHub per questo caso d'uso.

---

## Procedura

### 1 Preparazione dell'ambiente di GitLab

Creare un ramo Git per gli oggetti Python di vRealize Orchestrator.

### 2 Configurazione di una connessione a un repository Git

Un **amministratore** può configurare una connessione tra la distribuzione di vRealize Orchestrator e un progetto o un repository Git.

### 3 Push delle modifiche in un repository Git

Eseguire il push delle modifiche apportate agli oggetti di vRealize Orchestrator locali nel repository Git integrato. Per questo caso d'uso, viene eseguito il push delle modifiche apportate a un'azione di vRealize Orchestrator basata su Python in un ramo Git specifico.

## Preparazione dell'ambiente di GitLab

Creare un ramo Git per gli oggetti Python di vRealize Orchestrator.

## Prerequisiti

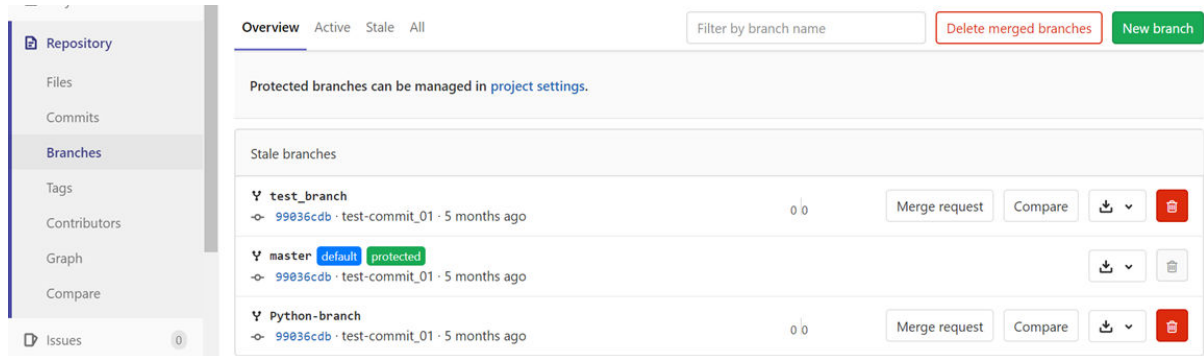
Creare un progetto GitLab per l'ambiente di vRealize Orchestrator. Vedere [Creazione di un progetto](#).

## Procedura

- 1 Accedere all'account GitLab.
- 2 Passare al progetto GitLab.
- 3 Nel riquadro di spostamento a sinistra, selezionare **Repository > Rami**.
- 4 Nella scheda **Panoramica**, fare clic su **Nuovo ramo**.
- 5 In **Nome ramo**, immettere **Python-branch**.
- 6 Lasciare l'opzione **Crea da** impostata su **master**.

## 7 Fare clic su **Crea ramo**.

È stato creato un ramo per gli oggetti di vRealize Orchestrator basati su Python.



## Configurazione di una connessione a un repository Git

Un **amministratore** può configurare una connessione tra la distribuzione di vRealize Orchestrator e un progetto o un repository Git.

Per utilizzare Git per la gestione dell'inventario degli oggetti di vRealize Orchestrator, è necessario configurare una connessione al repository Git utilizzando vRealize Orchestrator Client.

### Prerequisiti

- Verificare che l'ambiente di vRealize Orchestrator utilizzi una licenza di vRealize Automation.
- Generare un token di accesso per il progetto GitLab e copiarlo negli Appunti per utilizzarlo durante il processo di configurazione. Vedere [Creazione di un token di accesso personale](#).

**Nota** Per questo caso d'uso viene utilizzato un progetto GitLab. Se si ha familiarità con GitHub, è possibile utilizzare un repository GitHub. Per informazioni sulla generazione di un token GitHub, vedere [Creazione di un token di accesso personale per la riga di comando](#).

### Procedura

- 1 Accedere a vRealize Orchestrator Client come **amministratore**.
- 2 Passare a **Amministrazione > Repository Git**.
- 3 Fare clic su **Aggiungi repository**.
- 4 Immettere l'indirizzo URL del repository Git.

Ad esempio, <https://gitlab.com/myusername/my-vro-repo>.

**Nota** È inoltre possibile stabilire una connessione con il protocollo SSH.

- 5 Immettere il nome utente per il profilo Git.
- 6 Immettere il token di accesso del repository Git.
- 7 Per convalidare la connessione al repository Git, fare clic su **Convalida**.

- 8 (Facoltativo) Modificare il nome utilizzato per identificare il repository in vRealize Orchestrator Client.
- 9 (Facoltativo) Aggiungere una breve descrizione per il repository Git connesso.
- 10 Per attivare il repository Git connesso, fare clic su **Rendi attivo repository**.

---

**Nota** Solo un repository Git alla volta può essere attivo. È possibile modificare il repository Git attivo dalla pagina **Repository Git**.

---

- 11 Selezionare il ramo in cui si desidera eseguire il push delle modifiche. Per questo caso d'uso, viene utilizzato **Python-branch**. Vedere [Preparazione dell'ambiente di GitLab](#).

---

**Nota** È possibile modificare il ramo Git selezionato in qualsiasi momento dopo aver completato la configurazione Git iniziale.

---

- 12 Per completare il processo di configurazione, fare clic su **Salva**.

#### Operazioni successive

Tornare al menu **Repository Git** e verificare che lo stato del repository sia **Attivo**.

## Push delle modifiche in un repository Git

Eseguire il push delle modifiche apportate agli oggetti di vRealize Orchestrator locali nel repository Git integrato. Per questo caso d'uso, viene eseguito il push delle modifiche apportate a un'azione di vRealize Orchestrator basata su Python in un ramo Git specifico.

È possibile eseguire il push di un set di modifiche locali in un repository Git. Ogni set di modifiche può essere costituito da uno o più oggetti di vRealize Orchestrator modificati.

---

**Nota** Il processo di esecuzione del push e di rimozione dei set di modifiche in un repository Git non è limitato dalle autorizzazioni dei gruppi. Uno sviluppatore di workflow di un gruppo può quindi eseguire il push o la rimozione delle modifiche locali apportate da un altro sviluppatore.

---

#### Prerequisiti

- Verificare di aver creato un ramo Git. Vedere [Preparazione dell'ambiente di GitLab](#).
- Verificare di aver configurato una connessione con un repository Git. Vedere [Configurazione di una connessione a un repository Git](#).
- Verificare che l'integrazione Git sia impostata per eseguire il push delle modifiche al ramo Git **Python-branch**.
- Creare un oggetto di vRealize Orchestrator basato su Python. Ad esempio, vedere [Come integrare Amazon Web Services in vRealize Orchestrator tramite Python](#).

#### Procedura

- 1 Accedere a vRealize Orchestrator Client.

## 2 Modificare l'azione Python.

- Passare a **Libreria > Azioni** e selezionare l'azione Python.
- Apportare alcune piccole modifiche all'azione, ad esempio modificando la descrizione.
- Salvare l'azione.

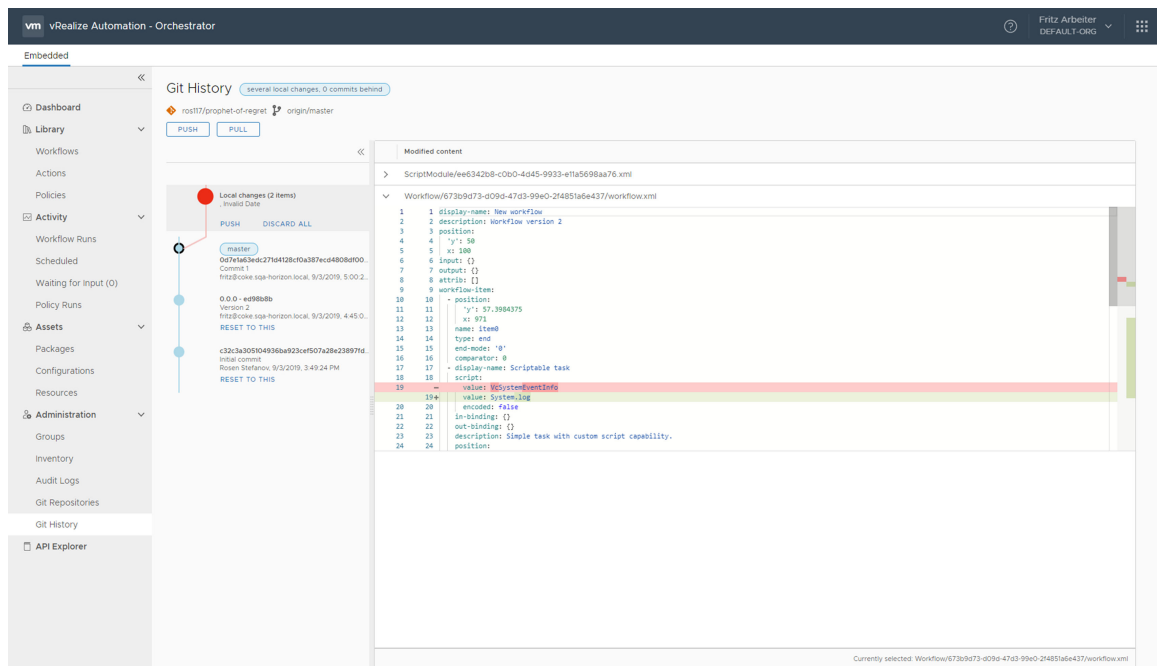
## 3 Eseguire il push delle modifiche nel repository Git.

**Nota** È inoltre possibile eseguire il push delle modifiche locali a livello dei singoli oggetti facendo clic sull'opzione **Versione** visualizzata nella parte inferiore dell'editor di oggetti.

- Passare a **Amministrazione > Cronologia Git**.

In **Cronologia Git** sono visualizzate le differenze correnti tra il ramo della versione locale e il ramo del repository Git selezionato. È possibile espandere la voce per qualsiasi oggetto di vRealize Orchestrator modificato per visualizzare le differenze tra le versioni.

**Nota** È possibile rimuovere un set di modifiche locale selezionando **Rimuovi tutto**.



- Fare clic su **Esegui push**.
- Immettere un titolo per il commit.
- (Facoltativo) Immettere una breve descrizione per il commit.
- Selezionare le modifiche apportate all'azione Python di cui si desidera eseguire il push nel repository Git.

## 4 Per completare il push del set di modifiche locali nel repository Git, fare clic su **Esegui push**.



### Operazioni successive

Dopo aver completato lo sviluppo nel ramo Git, unirlo al ramo principale. Vedere [Come creare una richiesta di unione](#).

## Come utilizzare moduli di terze parti per richiamare l'API del progetto di vRealize Automation

Questo caso d'uso di vRealize Orchestrator illustra come è possibile richiamare l'API del progetto di vRealize Automation utilizzando moduli di terze parti.

A partire da vRealize Orchestrator 8.1, è possibile utilizzare i seguenti runtime:

- Python 3.7
- Node.js 12
- PowerCLI 11/Powershell 6.2
- PowerCLI 12/Powershell 7

---

**Nota** Il runtime PowerCLI include PowerShell e i seguenti moduli: VMware.PowerCLI, PowerNSX, PowervRA.

---

Questo caso d'uso illustra come creare azioni di vRealize Orchestrator che utilizzino moduli di dipendenze di terze parti per connettersi all'API del progetto di vRealize Automation.

---

**Importante** Prima di iniziare a sviluppare lo script personalizzato, assicurarsi di conoscere i concetti di base dell'utilizzo degli script Python, Node.js e PowerShell in vRealize Orchestrator. Vedere [Concetti di base per gli script Python, Node.js e PowerShell](#).

---

## Creazione di uno script Python che richiami l'API del progetto di vRealize Automation

Creare uno script di esempio che utilizzi Python per richiamare l'API del progetto di vRealize Automation.

### Prerequisiti

Verificare di aver installato Python 3 e il programma di installazione del pacchetto PIP. Vedere la [pagina dei download di Python](#) e [Indice del pacchetto Python](#).

### Procedura

- 1 Sulla macchina locale, aprire una shell della riga di comando.
- 2 Creare una cartella vro-python-vra.

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

**3** Passare alla cartella vro-python-vra.

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

**4** Creare uno script Python denominato handler.py.

```
touch handler.py
```

Lo script handler.py deve definire una funzione che accetti due argomenti, il contesto dell'esecuzione del workflow di vRealize Orchestrator e gli input di vRealize Orchestrator associati.

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

**Nota** Se si utilizzano le librerie di registrazione standard, tutto ciò che viene registrato nell'azione che utilizza lo script viene visualizzato anche nel registro del workflow. Gli input e ciò che viene restituito dallo script devono avere parametri di input e tipi di restituzione corrispondenti configurati in vRealize Orchestrator Client. Ad esempio, l'input vRAUr1 nello script deve avere un parametro di input corrispondente denominato vRAUr1 in vRealize Orchestrator Client. Analogamente, se lo script restituisce un valore stringa, anche il tipo di restituzione configurato in vRealize Orchestrator Client deve essere stringa. Se l'azione restituisce un oggetto complesso, è possibile utilizzare il tipo di restituzione Properties o Composite Type.

**5** Installare il modulo di richiesta di Python.

**Importante** I moduli delle dipendenze di terze parti devono essere installati in una cartella di livello root nella cartella dello script vro-python-vra principale. Per questo caso d'uso, verrà creata una cartella lib per il modulo di richiesta.

**a** Creare una cartella lib.

```
mkdir lib
```

**b** Installare il modulo di richiesta.

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

**6** Aggiungere il modulo di richiesta allo script handler.py.

```
import requests

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

**7** Creare una richiesta GET per l'API del progetto di vRealize Automation.

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

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

**8** Definire i valori di token e vRAUrl .

- a Recuperare il token di accesso utilizzando l'API del servizio di identità di vRealize Automation. Vedere [Ottenere il token di accesso per l'API di vRealize Automation](#)
- b Per il valore di vRAUrl, definire lo script in modo che utilizzi un parametro di input di vRealize Orchestrator con lo stesso nome.

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

- c Aggiungere i nuovi valori al file 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()
```

**Nota** Poiché la risposta dell'API del progetto di vRealize Automation viene restituita in formato JSON, utilizzare il tipo di restituzione Properties o Composite Type per l'azione di vRealize Orchestrator.

**9** Creare un pacchetto ZIP che contenga il file handler.py e la cartella lib del modulo di richiesta.

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

**Operazioni successive**

Importare lo script di PowerShell in un'azione di vRealize Orchestrator. Vedere [Creazione di azioni in vRealize Orchestrator Client](#).

## Creazione di uno script Node.js che richiami l'API del progetto di vRealize Automation

Creare uno script di esempio che utilizzi Node.js per richiamare l'API del progetto di vRealize Automation.

### Prerequisiti

Scaricare e installare Node.js 12. Vedere [Download di Node.js](#).

### Procedura

- 1 Sulla macchina locale, aprire una shell della riga di comando.
- 2 Creare una cartella vro-node-vra.

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

- 3 Passare alla cartella vro-node-vra.

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

- 4 Creare uno script Node.js denominato handler.js.

```
touch handler.js
```

Lo script handler.js deve definire una funzione che accetti due argomenti, il contesto dell'esecuzione del workflow di vRealize Orchestrator e gli input di vRealize Orchestrator associati.

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

**Nota** Se si utilizzano le librerie di registrazione standard, tutto ciò che viene registrato nell'azione che utilizza lo script viene visualizzato anche nel registro del workflow. Gli input e ciò che viene restituito dallo script devono avere parametri di input e tipi di restituzione corrispondenti configurati in vRealize Orchestrator Client. Ad esempio, l'input vRAUr1 nello script deve avere un parametro di input corrispondente denominato vRAUr1 in vRealize Orchestrator Client. Analogamente, se lo script restituisce un valore stringa, anche il tipo di restituzione configurato in vRealize Orchestrator Client deve essere stringa. Se l'azione restituisce un oggetto complesso, è possibile utilizzare il tipo di restituzione Properties o Composite Type.

## 5 Installare il modulo di richiesta Node.js.

```
npm install request
```

**Importante** I moduli delle dipendenze di terze parti devono essere installati nella cartella di livello root `node_modules` nella cartella dello script `vro-node-vra` principale. Non spostare o rinominare questa cartella.

## 6 Aggiungere il modulo di richiesta allo script `handler.js`.

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

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

## 7 Creare una richiesta GET per l'API del progetto di 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** Definire i valori di token e vRAUrl .

- a Recuperare il token di accesso utilizzando l'API del servizio di identità di vRealize Automation. Vedere [Ottenere il token di accesso per l'API di vRealize Automation](#).
- b Per il valore di vRAUrl, definire lo script in modo che utilizzi un parametro di input di vRealize Orchestrator con lo stesso nome.

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

- c Aggiungere i nuovi valori al file 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));
  });
}
```

**Nota** Poiché la risposta dell'API del progetto di vRealize Automation viene restituita in formato JSON, utilizzare il tipo di restituzione Properties o Composite Type per l'azione di vRealize Orchestrator.

- 9** Creare un pacchetto ZIP che contenga il file handler.js e la cartella node\_modules del modulo di richiesta.

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

**Operazioni successive**

Importare lo script Node.js in un'azione di vRealize Orchestrator. Vedere [Creazione di azioni in vRealize Orchestrator Client](#).

## Creazione di uno script di PowerShell che richiami l'API del progetto di vRealize Automation

Creare uno script di esempio che utilizzi PowerShell per richiamare l'API del progetto di vRealize Automation.

**Procedura**

- 1** Sulla macchina locale, aprire una shell della riga di comando.
- 2** Creare una cartella vro-powershell-vra.

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

**3** Passare alla cartella vro-powershell-vra.

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

**4** Creare uno script di PowerShell denominato handler.ps1.

```
touch handler.ps1
```

Lo script handler.ps1 deve definire una funzione che accetti due argomenti, il contesto dell'esecuzione del workflow di vRealize Orchestrator e gli input di vRealize Orchestrator associati.

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

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

**Nota** Se si utilizzano le librerie di registrazione standard, tutto ciò che viene registrato nell'azione che utilizza lo script viene visualizzato anche nel registro del workflow. Gli input e ciò che viene restituito dallo script devono avere parametri di input e tipi di restituzione corrispondenti configurati in vRealize Orchestrator Client. Ad esempio, l'input vRAUr1 nello script deve avere un parametro di input corrispondente denominato vRAUr1 in vRealize Orchestrator Client. Analogamente, se lo script restituisce un valore stringa, anche il tipo di restituzione configurato in vRealize Orchestrator Client deve essere stringa. Se l'azione restituisce un oggetto complesso, è possibile utilizzare il tipo di restituzione Properties o Composite Type.

**5** Installare il modulo di asserzione di PowerShell.

**Importante** I moduli delle dipendenze di terze parti devono essere installati in una cartella di livello root nella cartella dello script vro-powershell-vra principale. Per questo caso d'uso, verrà creata una cartella Modules per il modulo di asserzione.

**a** Creare una cartella Modules.

```
mkdir Modules
```

**b** Installare il modulo di asserzione.

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

**6** Aggiungere il modulo di asserzione allo script handler.ps1.

```
Import-Module Assert

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

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

- 7** Creare una richiesta GET per l'API del progetto di vRealize Automation che utilizza il cmdlet `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** Definire i valori di token e `vRAUrl`.
- Recuperare il token di accesso utilizzando l'API del servizio di identità di vRealize Automation. Vedere [Ottenere il token di accesso per l'API di vRealize Automation](#).
  - Aggiungere gli attributi del modulo di asserzione `Assert-NotNull` e `Assert-Type`.

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



- c Per il valore di `vRAUrl`, definire lo script in modo che utilizzi un parametro di input di vRealize Orchestrator con lo stesso nome.

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

- d Aggiungere i nuovi valori al file `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
}
```

**Nota** Poiché la risposta dell'API del progetto di vRealize Automation viene restituita in formato JSON, utilizzare il tipo di restituzione `Properties` o `Composite Type` per l'azione di vRealize Orchestrator.

- 9 Creare un pacchetto ZIP contenente il file `handler.ps1` e la cartella `Modules` del modulo di asserzione.

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

### Operazioni successive

Importare lo script di PowerShell in un'azione di vRealize Orchestrator. Vedere [Creazione di azioni in vRealize Orchestrator Client](#).

# Gestione dei workflow

# 5

Un workflow è una serie di azioni e decisioni eseguite in sequenza. vRealize Orchestrator include una libreria di workflow che eseguono attività di gestione comuni. vRealize Orchestrator include inoltre librerie delle singole azioni eseguite dai workflow.

I workflow combinano azioni, decisioni e risultati che, se eseguiti in un determinato ordine, completano un'attività o un processo specifici in un ambiente virtuale. I workflow eseguono attività come provisioning di macchine virtuali, backup, manutenzione regolare, invio di email, operazioni SSH, gestione dell'infrastruttura fisica e altre operazioni di utilità generale. I workflow accettano input in base alla rispettiva funzione. È possibile creare workflow che vengono eseguiti in base alle pianificazioni definite o che vengono eseguiti se si verificano determinati eventi previsti. Le informazioni possono essere fornite dall'utente, da altri utenti, da un altro workflow o azione o da un processo esterno, ad esempio una chiamata del servizio Web da un'applicazione. I workflow eseguono alcuni processi di convalida e filtro delle informazioni prima che vengano eseguiti.

I workflow possono chiamare altri workflow. È ad esempio possibile che sia presente un workflow che richiama un altro workflow per creare una nuova macchina virtuale.

È possibile creare workflow utilizzando l'IDE (Integrated Development Environment) dell'interfaccia di vRealize Orchestrator Client, che consente di accedere alla libreria dei workflow e offre la possibilità di eseguire workflow nel motore dei workflow. Il motore dei workflow può anche ricevere oggetti da librerie esterne collegate a vRealize Orchestrator. Questa funzionalità consente di personalizzare i processi o implementare le funzioni fornite da applicazioni di terze parti.

Questo capitolo include i seguenti argomenti:

- [Workflow standard nella libreria dei workflow di vRealize Orchestrator](#)
- [Creazione di workflow in vRealize Orchestrator Client](#)
- [Modifica di workflow e azioni dal workflow principale](#)
- [Designer del modulo di input di vRealize Orchestrator](#)
- [Richieste di interazione dell'utente in vRealize Orchestrator Client](#)
- [Pianificazione dei workflow in vRealize Orchestrator Client](#)

## Workflow standard nella libreria dei workflow di vRealize Orchestrator

vRealize Orchestrator fornisce una libreria standard di workflow che è possibile utilizzare per automatizzare le operazioni nell'infrastruttura virtuale. I workflow della libreria standard sono bloccati nello stato di sola lettura. Per personalizzare un workflow standard, è necessario duplicarlo. I workflow duplicati o i workflow personalizzati che vengono creati sono completamente modificabili.

Il contenuto della libreria dei workflow è accessibile tramite il menu **Libreria > Workflow** di vRealize Orchestrator Client basato su HTML5. I workflow standard e i workflow personalizzati del client vengono organizzati tramite tag. Ad esempio, è possibile accedere al workflow **Genera coppia di chiavi** immettendo **SSH** nella casella di ricerca della libreria dei workflow.

---

**Nota** Non è possibile aggiungere nuovi tag a un workflow standard, a meno che non lo si duplichi.

---

## Creazione di workflow in vRealize Orchestrator Client

È possibile utilizzare vRealize Orchestrator Client per creare e modificare workflow.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Selezionare **Libreria > Workflow**.
- 3 Fare clic su **Nuovo workflow**.
- 4 Immettere il nome del nuovo workflow e fare clic su **Crea**.
- 5 Utilizzare l'editor di workflow per configurare le variabili, gli input e gli output del workflow, la struttura dello schema e la presentazione del workflow.
- 6 Per completare la modifica del workflow, fare clic su **Salva**.

---

**Nota** È possibile tenere traccia delle modifiche apportate ai workflow nella scheda **Cronologia versioni**. Per ulteriori informazioni, vedere [Cronologia delle versioni di un oggetto di vRealize Orchestrator](#).

---

### Operazioni successive

È possibile utilizzare la funzionalità di riproduzione del token di vRealize Orchestrator per ottimizzare le prestazioni dei Workflow. Per ulteriori informazioni, vedere [Utilizzo della riproduzione del token del workflow in vRealize Orchestrator Client](#).

## Modifica di workflow e azioni dal workflow principale

Modificare i workflow e le azioni direttamente dal workflow principale in vRealize Orchestrator Client.

La modifica dei workflow e delle azioni secondari direttamente dal workflow principale può semplificare lo sviluppo del workflow.

### Prerequisiti

Creare un workflow che richiami un altro workflow, un'azione o entrambi.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Workflow** e selezionare il workflow.
- 3 Selezionare la scheda **Schema**.
- 4 In base al tipo di oggetto, fare doppio clic su **Elemento workflow** o **Elemento azione** nella tela del workflow.
- 5 Modificare l'oggetto.
- 6 Per completare la modifica del workflow o dell'azione secondari, fare clic su **Salva**.
- 7 Per tornare al workflow principale, chiudere l'editor di oggetti.

## Designer del modulo di input di vRealize Orchestrator

Se un workflow richiede parametri di input, apre una finestra di dialogo in cui gli utenti inseriscono i valori richiesti. È possibile organizzare il contenuto, il layout e la presentazione di questa finestra di dialogo con il designer del modulo di input.

Il designer del modulo di input si trova nella scheda **Modulo di input** dell'editor di workflow. Il designer è costituito da un menu di navigazione, una tela di progettazione e un menu delle proprietà. È possibile trascinare input ed elementi generici dal menu a sinistra alla tela di progettazione. Nella tela, è possibile impostare la posizione dei parametri di input, organizzarli in schede di input separate e configurare le proprietà dei parametri di input.

---

**Nota** Non è possibile utilizzare il contenuto della scheda **Variabili** dell'editor di workflow nel designer del modulo di input. È possibile utilizzare solo i parametri della scheda **Input/Output**.

---

### Elementi generici

È possibile aggiungere elementi generici, ad esempio menu a discesa e caselle di testo per la password, al designer del modulo di input. Gli elementi generici non corrispondono ai parametri di input effettivi, ma possono essere associati ai parametri di input.

## Creazione della finestra di dialogo dei parametri di input del workflow in vRealize Orchestrator Client

È possibile utilizzare il designer del modulo di input per creare e personalizzare la finestra di dialogo dei parametri di input del workflow.

### Prerequisiti

Verificare che il workflow disponga di un elenco definito di parametri di input.

### Procedura

**1** Accedere a vRealize Orchestrator Client.

**2** Passare a **Libreria > Workflow**.

**3** Selezionare il workflow personalizzato.

**4** Fare clic sulla scheda **Modulo di input**.

**5** (Facoltativo) Creare schede da utilizzare nella finestra di dialogo di input.

È possibile utilizzare schede per organizzare la struttura della finestra di dialogo.

**6** Selezionare i parametri di input.

**7** Modificare le proprietà dei parametri di input.

Per ulteriori informazioni sulle proprietà dei parametri di input, vedere [Proprietà dei parametri di input in vRealize Orchestrator Client](#).

**8** (Facoltativo) Aggiungere elementi generici alla tela e associarli ai parametri di input.

**9** (Facoltativo) Aggiungere la convalida esterna ai parametri di input. Per ulteriori informazioni, vedere [Utilizzo delle azioni per convalidare gli input dei workflow di vRealize Orchestrator](#).

**10** Fare clic su **Salva**.

### Risultati

È stato creato il layout della finestra di dialogo del workflow e sono state impostate le proprietà dei parametri di input.

## Proprietà dei parametri di input in vRealize Orchestrator Client

È possibile impostare le proprietà dei parametri per vincolare i parametri di input che gli utenti specificano quando eseguono i workflow di vRealize Orchestrator.

Con vRealize Orchestrator, è possibile definire le proprietà dei parametri utilizzate per quantificare i valori dei parametri di input utilizzati nei workflow. Le proprietà dei parametri definite impongono limiti ai tipi e ai valori dei parametri di input che gli utenti possono specificare nei workflow di vRealize Orchestrator.

Le proprietà dei parametri convalidano i parametri di input e modificano la presentazione delle caselle di testo che vengono visualizzate nella finestra di dialogo dei parametri di input. Alcune proprietà dei parametri possono creare dipendenze tra i parametri.

Proprietà del parametro	Descrizione
<b>Etichetta</b>	Imposta l'etichetta del parametro di input.
<b>Tipo di visualizzazione</b>	Imposta il tipo di visualizzazione della casella di testo dell'input.
<b>Visibilità</b>	Imposta la visibilità del parametro di input.
<b>Sola lettura</b>	Imposta la casella di testo dell'input come casella di sola lettura.
<b>Guida personalizzata</b>	Imposta la descrizione delle indicazioni del parametro di input.
<b>Valore predefinito</b>	Imposta il valore predefinito del parametro di input.
<b>Passaggio</b>	Utilizzato per gli input di tipo numero. Imposta di quanto il valore del parametro di input aumenta per ogni clic.
<b>Obbligatorio</b>	Specifica se il valore del parametro di input è obbligatorio o meno.
<b>Espressione regolare</b>	Convalida l'input utilizzando un'espressione regolare.
<b>Valore minimo</b>	Imposta il valore minimo o la lunghezza del parametro.
<b>Valore massimo</b>	Imposta il valore massimo o la lunghezza del parametro.
<b>Casella di testo corrispondenza</b>	Imposta il valore del parametro di input che deve corrispondere al valore di un altro parametro di input.
<b>Origine valore</b>	<p>Imposta l'origine del valore delle proprietà del parametro nelle schede <b>Aspetto</b>, <b>Valore</b> e <b>Vincoli</b>.</p> <p><b>Nota</b> È possibile importare il valore delle azioni esterne utilizzando <b>Origine esterna</b>. Il filtro delle azioni disponibili viene eseguito in base al tipo di parametro.</p>

## Utilizzo delle azioni per convalidare gli input dei workflow di vRealize Orchestrator

Utilizzare azioni esterne per convalidare gli input dei workflow personalizzati.

### Prerequisiti

Creare un workflow personalizzato con parametri di input. Per ulteriori informazioni, vedere [Creazione di workflow in vRealize Orchestrator Client](#).

È possibile utilizzare il designer del modulo di input per creare convalide esterne per gli input dei workflow. Le convalide esterne utilizzano script di azione che restituiscono un valore stringa quando il valore del parametro di input contiene un errore. Se il valore del parametro di input è valido, la convalida esterna non restituisce alcun valore.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Creare un'azione di convalida.
  - a Passare a **Libreria > Azioni**.
  - b Fare clic su **Nuova azione**.
  - c Immettere le informazioni richieste nella scheda **Riepilogo**.
  - d Immettere i parametri di input dell'azione di convalida.

---

**Nota** I nomi dei parametri di input dell'azione di convalida devono essere identici ai nomi dei parametri di input dei workflow che vengono convalidati.

---

- e Immettere lo script dell'azione di convalida nella scheda **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
```

---

**Nota** Lo script precedente è un esempio semplice e non rappresenta l'ambito completo degli script di convalida che possono essere utilizzati.

---

- f Fare clic su **Salva**.
- 3 Applicare la convalida esterna.
  - a Passare a **Libreria > Workflow**.
  - b Selezionare il workflow personalizzato.
  - c Selezionare la scheda **Modulo di input**.
  - d Selezionare l'icona Appunti in alto a sinistra della schermata.
  - e Trascinare un elemento di convalida vRealize Orchestrator nella tela.
  - f Selezionare l'elemento di convalida, immettere un'etichetta di convalida e selezionare l'azione di convalida.
  - g (Facoltativo) Creare elementi di convalida aggiuntivi.
  - h Fare clic su **Salva**.

#### 4 Eseguire il workflow.

Se la convalida rileva un errore, restituisce una stringa. Se la convalida non rileva errori, non restituisce alcun valore e l'esecuzione del workflow continua.

#### Risultati

È stata creata una convalida esterna per il workflow di vRealize Orchestrator personalizzato.

## Richieste di interazione dell'utente in vRealize Orchestrator Client

Per il completamento dei workflow, è possibile che sia necessario ulteriore input dell'utente.

I workflow che richiedono ulteriore interazione dell'utente sospendono le operazioni finché i parametri di input richiesti non vengono forniti dall'utente. I workflow definiscono gli utenti che possono fornire le informazioni necessarie e inviano le richieste di interazione corrispondenti. I workflow in attesa di input dell'utente vengono visualizzati nel pannello **Esecuzioni dei workflow recenti** del dashboard di vRealize Orchestrator Client e nel menu di notifica in alto a destra.

## Pianificazione dei workflow in vRealize Orchestrator Client

È possibile utilizzare la pianificazione per automatizzare le esecuzioni dei workflow di vRealize Orchestrator.

Quando si pianificano le esecuzioni dei workflow, si imposta la data, l'ora e gli intervalli in cui viene eseguita l'attività pianificata.

#### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Selezionare il workflow nel menu **Libreria** e nel pannello del workflow, fare clic su **Pianifica**.
- 3 Configurare i parametri dell'attività pianificata nelle categorie **Generale**, **Pianificazione** e **Workflow**.

**Nota** La categoria di parametro **Workflow** è visibile solo per i workflow che richiedono parametri di input.

Parametro	Descrizione
<b>Nome</b>	Nome dell'attività pianificata.
<b>Descrizione</b>	Breve descrizione che indica lo scopo dell'attività pianificata.
<b>Inizio</b>	Data e ora della prima esecuzione pianificata del workflow.



Parametro	Descrizione
Inizia se è nel passato	Selezionare se avviare il workflow se l'ora pianificata è nel passato. <b>Sì</b> avvia immediatamente il workflow pianificato. <b>No</b> avvia il workflow alla successiva ricorrenza pianificata.
Pianifica	Impostare il criterio di ricorrenza e le voci del trigger di evento dell'attività pianificata.
Data di fine	Visibile solo se è stata selezionata l'opzione <b>Nessuna ricorrenza</b> . Impostare la data e l'ora di fine dell'attività pianificata.
Workflow	Immettere i parametri di input del workflow.

#### 4 Fare clic su **Crea**.

##### Risultati

È stata creata un'attività pianificata per il workflow. I workflow pianificati vengono visualizzati in **Attività > Pianificato**. È possibile eliminare le attività pianificate facendo clic su **Elimina** nel pannello della pianificazione.

## Modifica di un'attività pianificata in vRealize Orchestrator Client

Le attività pianificate possono essere modificate per modificare i parametri come la data, l'ora e la ricorrenza del workflow pianificato.

##### Prerequisiti

Creare un'attività di workflow pianificata.

##### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Selezionare l'attività pianificata da **Attività > Pianificato**.
- 3 Fare clic su **Modifica** nel pannello del workflow.
- 4 Modificare la pianificazione e fare clic su **Salva**.

**Nota** I parametri di input impostati durante la creazione dell'attività pianificata sono di sola lettura e non possono essere modificati. Per modificarli, creare una nuova attività pianificata per questo workflow.

# Gestione delle azioni

# 6

È possibile modificare i workflow di vRealize Orchestrator aggiungendo script di azioni.

vRealize Orchestrator Client offre librerie di azioni predefinite e un editor di azioni per gli script di azioni personalizzati. Le azioni rappresentano singole funzioni che vengono utilizzate come blocchi predefiniti nei workflow.

Le azioni sono funzioni JavaScript. Le azioni possono ricevere più parametri di input e avere un singolo valore restituito. Le azioni possono richiamare qualsiasi oggetto dell'API di vRealize Orchestrator o gli oggetti di qualsiasi API importata in vRealize Orchestrator utilizzando un plugin.

Quando un workflow è in esecuzione, un'azione riceve i parametri di input dalle variabili del workflow. Queste variabili possono essere i parametri di input iniziali del workflow o variabili che altri elementi nel workflow impostano quando vengono eseguiti.

L'editor di azioni include una funzionalità di completamento automatico per gli script e API Explorer che contiene i tipi di script disponibili e la relativa documentazione.

Questo capitolo include i seguenti argomenti:

- [Creazione di azioni in vRealize Orchestrator Client](#)
- [Esecuzione e debug delle azioni](#)
- [Concetti di base per gli script Python, Node.js e PowerShell](#)
- [Limiti di runtime per script Python, Node.js e PowerShell](#)

## Creazione di azioni in vRealize Orchestrator Client

È possibile utilizzare vRealize Orchestrator Client per creare, modificare ed eliminare gli script di azione.

A partire da vRealize Orchestrator 8.1, è possibile utilizzare i seguenti runtime:

- Python 3.7
- Node.js 12
- PowerCLI 11/Powershell 6.2

## ■ PowerCLI 12/Powershell 7

---

**Nota** Il runtime PowerCLI include PowerShell e i seguenti moduli: VMware.PowerCLI, PowerNSX, PowervRA.

---

### Prerequisiti

Prima di creare uno script Python, Node.js o PowerShell, assicurarsi di conoscere i concetti di base per lo sviluppo di script compatibili con vRealize Orchestrator che utilizzano questi runtime. Vedere [Concetti di base per gli script Python, Node.js e PowerShell](#).

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Azioni**.
- 3 Fare clic su **Nuova azione**.
- 4 Nella scheda **Generale**, immettere il nome e il nome del modulo dell'azione.

---

**Nota** Il nome e il nome del modulo devono essere univoci per ogni azione. Il nome dell'azione deve essere una funzione JavaScript valida. Il nome dell'azione deve essere costituito da una sola parola e può contenere solo lettere e numeri, nonché il simbolo del dollaro ("\$\$") e il carattere di sottolineatura ("\_"). Il nome del modulo deve essere composto da parole separate dal carattere punto (".").

---

- 5 (Facoltativo) Creare una descrizione, il numero di versione, i tag e le autorizzazioni di gruppo per l'azione.
- 6 Nella scheda **Script**, aggiungere input azione, selezionare il tipo di restituzione dell'output e scrivere lo script.

---

**Nota** Selezionando **Zip** nel menu a discesa **Tipo**, è possibile importare un'origine di script esterna e, se applicabile, i relativi moduli delle dipendenze.

---

- 7 Per completare la modifica dell'azione, fare clic su **Salva**.

Un messaggio indica che l'azione è stata salvata.

### Operazioni successive

Per visualizzare un esempio di caso d'uso che illustri come è possibile utilizzare le azioni di vRealize Orchestrator, vedere [Come integrare Amazon Web Services in vRealize Orchestrator tramite Python](#).

## Esecuzione e debug delle azioni

È possibile migliorare le azioni eseguendole ed eseguendone il debug direttamente dall'editor di azioni.

A partire da vRealize Orchestrator 8.1, è possibile eseguire le azioni ed eseguirne il debug direttamente dall'editor di azioni di vRealize Orchestrator Client. Con questa funzionalità, è possibile garantire che le azioni vengano eseguite come previsto quando sono integrate nei workflow.

## Esecuzione di azioni in vRealize Orchestrator Client

Un designer di workflow desidera eseguire le azioni prima di integrarle in un workflow.

### Prerequisiti

Creare un'azione. Vedere [Creazione di azioni in vRealize Orchestrator Client](#).

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Azioni** e selezionare l'azione che si desidera eseguire.
- 3 Fare clic su **Esegui**.
- 4 Immettere i parametri di input richiesti e fare clic su **Esegui**.

Al termine dell'esecuzione dell'azione, fare clic sulla scheda **Risultati/Input**. Se durante l'esecuzione dell'azione si verifica un errore, viene visualizzato in questa scheda con il colore rosso. È possibile visualizzare i dettagli dell'esecuzione dell'azione dall'elemento **Risultati azione**.

---

**Nota** I risultati dell'esecuzione dell'azione non vengono salvati.

---

## Azioni di debug in vRealize Orchestrator Client

Un designer del workflow può eseguire il debug delle azioni inserendo punti di interruzione nello script.

vRealize Orchestrator include uno strumento di debug integrato che è possibile utilizzare per eseguire il debug dello script e delle proprietà di input dell'azione. Il processo di debug può essere avviato nell'editor di azioni inserendo punti di interruzione nelle righe dello script dell'azione.

---

**Nota** Lo strumento di debug integrato funziona solo con azioni che utilizzano il runtime JavaScript predefinito. Per un esempio di come eseguire il debug degli script di azioni che utilizzano runtime diversi, vedere [Debug di un'azione di Amazon Web Services](#).

---

### Prerequisiti

Creare un'azione. Vedere [Creazione di azioni in vRealize Orchestrator Client](#).

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Azioni** e selezionare l'azione di cui si desidera eseguire il debug.

**3** Nell'editor di azioni, aggiungere punti di interruzione alle righe dello script dell'azione di cui si desidera eseguire il debug.

**4** Fare clic su **Debug**.

**5** Immettere i parametri di input dell'azione e fare clic su **Esegui**.

Viene avviata l'esecuzione dell'azione in modalità di debug.

**6** Quando l'esecuzione dell'azione viene sospesa dopo il raggiungimento di un punto di interruzione, selezionare una delle seguenti opzioni:

Opzione	Descrizione
<b>Continua</b>	Riprende l'esecuzione dell'azione finché non viene raggiunto un altro punto di interruzione o non termina l'esecuzione dell'azione.
<b>Passaggio in</b>	Esegue l'istruzione della funzione dell'azione corrente. Se il debugger non può andare ulteriormente in profondità nella riga della funzione corrente, esegue un'operazione <b>Passaggio su</b> .
<b>Passaggio su</b>	Il debugger continua nella riga successiva della funzione corrente.
<b>Step return</b>	Il debugger passa alla riga che verrà eseguita durante la restituzione della funzione corrente.

**7** (Facoltativo) Nella scheda **Debugger**, aggiungere espressioni.

**8** (Facoltativo) Nella scheda **Debugger**, modificare il valore delle variabili.

## Concetti di base per gli script Python, Node.js e PowerShell

Quando si crea uno script da utilizzare in vRealize Orchestrator, è necessario verificare che lo script abbia la struttura e la formattazione corrette.

### Runtime supportati

A partire da vRealize Orchestrator 8.1, è possibile utilizzare i seguenti runtime:

- Python 3.7
- Node.js 12
- PowerCLI 11/Powershell 6.2
- PowerCLI 12/Powershell 7

**Nota** Il runtime PowerCLI include PowerShell e i seguenti moduli: VMware.PowerCLI, PowerNSX, PowervRA.

È possibile aggiungere qualsiasi codice sorgente personalizzato ai nuovi runtime, ma per accettare il contesto e gli input e restituire un risultato da e verso il motore vRealize Orchestrator, è necessario seguire il formato funzionale corretto.

## Consigli per lo scripting

Per semplificare le attività di scripting, è possibile aggiungere elementi **Attività gestibile tramite script** allo schema del workflow. Per le attività di scripting più complesse, è possibile utilizzare azioni di vRealize Orchestrator.

L'utilizzo delle azioni offre due vantaggi specifici:

- Le azioni possono essere create, aggiornate, importate ed esportate indipendentemente dai workflow.
- Le azioni sono oggetti autonomi che possono essere eseguiti e sottoposti a debug nel proprio ambiente. Ciò può rendere più agevole il processo di sviluppo. Vedere [Esecuzione e debug delle azioni](#).

## Requisiti della funzione dello script

Il nome predefinito della funzione dello script è **handler**. La funzione accetta due argomenti, ovvero context e input. Context è un oggetto mappa contenente informazioni di sistema. Ad esempio, vroURL può contenere l'URL dell'istanza di vRealize Orchestrator che si desidera richiamare, mentre executionId contiene l'ID token di un'esecuzione di workflow.

Un input è un oggetto mappa contenente tutti gli input forniti alle azioni. Ad esempio, se si definisce un input nell'azione denominata myInput, è possibile accedervi dall'argomento input, ovvero inputs.myInput o inputs["myInput"], in base al runtime. Qualsiasi elemento restituito dalla funzione è il risultato dell'azione. Pertanto, il tipo restituito dell'azione deve corrispondere al tipo di contenuto restituito dallo script in vRealize Orchestrator. Se si restituisce un numero primitivo, il tipo restituito dall'azione deve essere un tipo numero. Se si restituisce una stringa, il tipo restituito dall'azione deve essere un tipo stringa. Se si restituisce un oggetto complesso, il tipo restituito deve essere mappato a Properties o Composite Type. Gli stessi principi si applicano anche agli array.

Tipi di parametri di input e output supportati per i runtime Python, Node.js e PowerShell:

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

## Definizione del gestore delle voci

Per impostazione predefinita, il valore del gestore delle voci è `handler.handler`. Questo valore indica che il motore vRealize Orchestrator cerca un file di livello superiore nel pacchetto ZIP denominato `handler.py`, `handler.js` o `handler.ps1`, che include una funzione denominata `handler`. Qualsiasi differenza tra i nomi della funzione e del file del gestore deve essere riportata nel valore del gestore delle voci. Ad esempio, se il gestore principale è denominato `index.js` e la funzione è denominata `callMe`, è necessario impostare il valore per il gestore delle voci su `index.callMe`.

## Debug degli script di runtime in un IDE esterno

vRealize Orchestrator supporta il debug degli script Python e Node.js in un IDE esterno. Non è possibile eseguire il debug degli script PowerShell in un IDE esterno.

## Limiti di runtime per script Python, Node.js e PowerShell

Alcuni script Python, Node.js o PowerShell possono richiedere la modifica dei valori di memoria e timeout in vRealize Orchestrator Client.

vRealize Orchestrator Client utilizza un set di valori di memoria e timeout predefiniti per gli script di azione Python, Node.js e PowerShell:

- Memoria: 64 MB
- Timeout: 180 secondi

Se lo script di azione supera uno o entrambi i valori predefiniti, l'esecuzione dell'azione non riesce. È ad esempio possibile che lo script di azione utilizzi più moduli di dipendenze di terze parti. In questo scenario, il limite di memoria predefinito di 64 MB potrebbe non essere sufficiente.

Per evitare che le esecuzioni di azioni non riescano a causa di risorse insufficienti, modificare i valori di memoria e timeout dall'editor di azioni.

---

**Nota** È inoltre possibile suddividere lo script in più elementi di attività di script, che possono essere aggiunti ai workflow.

---

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Azioni** e selezionare l'azione.
- 3 Selezionare la scheda **Script**.
- 4 In **Limiti runtime**, modificare i valori di memoria e timeout.
- 5 Fare clic su **Salva**.
- 6 Per verificare i nuovi limiti di runtime, fare clic su **Debug**.

# Gestione degli elementi di configurazione

# 7

Un elemento di configurazione è un elenco di variabili che è possibile utilizzare per configurare le costanti in un'intera distribuzione del server di vRealize Orchestrator.

È possibile utilizzare gli elementi di configurazione per rendere le variabili disponibili per tutti i workflow, le azioni e i criteri in esecuzione sul server di vRealize Orchestrator.

Se si crea un pacchetto contenente un workflow, un'azione o un criterio che utilizza una variabile di un elemento di configurazione, vRealize Orchestrator include automaticamente l'elemento di configurazione nel pacchetto. Se si importa un pacchetto contenente un elemento di configurazione in un altro server di vRealize Orchestrator, è possibile importare anche i valori delle variabili degli elementi di configurazione. Ad esempio, se si crea un workflow che richiede valori delle variabili che dipendono dal server di vRealize Orchestrator su cui viene eseguito, l'impostazione di tali variabili in un elemento di configurazione consente di esportare tale workflow, in modo che un altro server di vRealize Orchestrator possa utilizzarlo. Gli elementi di configurazione consentono pertanto di scambiare più facilmente workflow, azioni e criteri tra i server.

---

**Nota** Non è possibile importare i valori di una variabile di un elemento di configurazione esportato da vRealize Orchestrator 5.1 o versioni precedenti.

---

Questo capitolo include i seguenti argomenti:

- [Creazione di elementi di configurazione in vRealize Orchestrator Client](#)

## Creazione di elementi di configurazione in vRealize Orchestrator Client

Con gli elementi di configurazione, è possibile impostare variabili comuni in un server vRealize Orchestrator. Tutti gli elementi in esecuzione nel server possono utilizzare le variabili impostate in un elemento di configurazione.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Asset > Configurazioni**.



- 3 Selezionare **Nuova configurazione**.
- 4 Immettere il nome dell'elemento di configurazione.
- 5 Selezionare la scheda **Variabili**.
- 6 Per creare una variabile locale, fare clic su **Nuovo**.
  - a Immettere il nome della variabile.
  - b Selezionare il tipo della variabile.

---

**Nota** Per creare un array di variabili di configurazione, selezionare la casella di controllo **Array**.

---

- c (Facoltativo) Immettere un valore per la variabile di configurazione.
  - d Fare clic su **Salva**.
- 7 Per completare la creazione dell'elemento di configurazione, fare clic su **Salva**.

#### Operazioni successive

È possibile utilizzare l'elemento di configurazione per fornire variabili a workflow, azioni o criteri.

# Gestione dei criteri

# 8

I criteri sono trigger di eventi che monitorano l'attività del sistema. I criteri rispondono agli eventi predefiniti emessi dalle modifiche dello stato o delle prestazioni di oggetti specifici di vRealize Orchestrator.

I criteri sono una serie di regole, misuratori, soglie e filtri di eventi che eseguono determinati workflow o script quando si verificano eventi predefiniti specifici in vRealize Orchestrator o nelle tecnologie a cui vRealize Orchestrator accede tramite i plug-in. vRealize Orchestrator valuta costantemente le regole dei criteri mentre il criterio è in esecuzione. Ad esempio, è possibile implementare misuratori e soglie dei criteri che monitorino il comportamento degli oggetti di vCenter Server di tipo VC:HostSystem e VC:VirtualMachine.

Questo capitolo include i seguenti argomenti:

- [Creazione e applicazione di criteri in vRealize Orchestrator Client](#)
- [Elementi criterio in vRealize Orchestrator Client](#)
- [Gestione delle esecuzioni dei criteri in vRealize Orchestrator Client](#)

## Creazione e applicazione di criteri in vRealize Orchestrator Client

È possibile utilizzare i criteri per monitorare l'attività del sistema di vRealize Orchestrator per eventi specifici.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Criteri**.
- 3 Selezionare **Nuovo criterio**.  
Viene creato un criterio vuoto.
- 4 Immettere un nome e un numero di versione per il criterio.
- 5 Selezionare la scheda **Variabili**.

**6** Per creare una variabile locale, fare clic su **Nuovo**.

- a Immettere il nome della variabile.
- b Selezionare il tipo della variabile.

---

**Nota** Per creare un array di variabili del criterio, selezionare la casella di controllo **Array**.

---

- c Immettere il valore della variabile.

---

**Nota** Per importare il valore della variabile di un elemento di configurazione, è possibile utilizzare **Associa a configurazione**.

---

- d Fare clic su **Salva**.

**7** Nella scheda **Definizione**, aggiungere elementi criterio e impostare i gestori di eventi.

Per ulteriori informazioni sugli elementi criterio, vedere [Elementi criterio in vRealize Orchestrator Client](#).

**8** Fare clic su **Salva**.

Il criterio è stato configurato.

**Operazioni successive**

Per avviare un criterio, selezionare il criterio e fare clic su **Esegui**. Immettere il nome dell'esecuzione del criterio e, se richiesto, i parametri di input necessari.

Per visualizzare lo stato del criterio, passare a **Attività > Esecuzioni criteri**.

## Elementi criterio in vRealize Orchestrator Client

È possibile utilizzare gli elementi criterio per eseguire workflow o script di vRealize Orchestrator predefiniti quando si verifica un evento.

È possibile aggiungere un elemento criterio per attivare esecuzioni di workflow o script in risposta a eventi attivati dagli oggetti. Con l'elemento evento periodico, è possibile pianificare esecuzioni di script o workflow. Con l'elemento root, è possibile impostare il comportamento di avvio o arresto dei criteri. Gli elementi criterio possono includere gestori di eventi che definiscono quando è necessario eseguire gli elementi criterio.

---

**Nota** I gestori di eventi che attivano gli elementi criterio possono essere workflow o script di azione. Se si aggiunge un workflow e uno script a un gestore eventi, il criterio ignora il trigger dello script e utilizza solo il trigger del workflow.

---

Gestore eventi	Descrizione
<b>OnInit</b>	L'elemento criterio viene attivato ogni volta che si avvia il criterio.
<b>OnExit</b>	L'elemento criterio viene attivato ogni volta che si interrompe il criterio.
<b>OnExecute</b>	Utilizzato dall'elemento evento periodico. Attiva l'elemento criterio durante il tempo specificato nell'elemento evento periodico.

**Nota** Le tecnologie collegate al database di vRealize Orchestrator possono disporre di gestori eventi univoci. Ad esempio, tramite il plug-in SNMP, è possibile utilizzare il gestore eventi **OnTrap** durante la creazione di elementi criterio basati su SNMP.

Gli elementi criterio sono configurati nella scheda **Definizione** della finestra di modifica del criterio.

## Gestione delle esecuzioni dei criteri in vRealize Orchestrator Client

È possibile utilizzare vRealize Orchestrator Client per gestire la priorità dei criteri e il comportamento di avvio del server dei criteri per quando il server di vRealize Orchestrator viene riavviato.

### Prerequisiti

Creare e avviare un criterio. Per ulteriori informazioni, vedere [Creazione e applicazione di criteri in vRealize Orchestrator Client](#).

### Procedura

- 1 Accedere a vRealize Orchestrator Client come amministratore.
- 2 Passare a **Attività > Esecuzioni criteri**.
- 3 Fare clic sull'esecuzione del criterio che si desidera gestire.
- 4 Fare clic su **Arresta**.  
Lo stato del criterio diventa **Arrestato**.
- 5 Nella scheda **Generale**, impostare la priorità dei criteri e il comportamento di avvio del server.
- 6 Per riavviare il criterio, fare clic su **Esegui**.  
Lo stato del criterio diventa **In esecuzione**.

# Gestione degli elementi delle risorse

## 9

I workflow possono utilizzare oggetti creati indipendentemente da vRealize Orchestrator come attributi. Per utilizzare gli oggetti esterni come attributi nei workflow, importarli nel server come elementi delle risorse.

Gli oggetti che i workflow di vRealize Orchestrator possono utilizzare come elementi delle risorse includono file di immagine, script, modelli XML, file HTML e così via. Ogni workflow eseguito nel server di vRealize Orchestrator può utilizzare tutti gli elementi delle risorse importati in vRealize Orchestrator.

Dopo aver importato un oggetto in vRealize Orchestrator come elemento di risorsa, è possibile apportare modifiche all'oggetto in un'unica posizione e propagare automaticamente tali modifiche a tutti i workflow che utilizzano questo elemento di risorsa.

Le dimensioni massime consentite per un elemento di risorsa sono 16 MB.

È possibile importare, esportare, ripristinare, aggiornare ed eliminare un elemento di risorsa.

# Gestione dei pacchetti

# 10

Utilizzare vRealize Orchestrator Client per creare, esportare e importare pacchetti. I pacchetti possono essere utilizzati per esportare gli oggetti del workflow da utilizzare in altre istanze di vRealize Orchestrator.

I pacchetti possono contenere workflow, azioni, criteri, elementi di configurazione o elementi di risorse.

Quando si aggiunge un elemento a un pacchetto, vRealize Orchestrator verifica la presenza di dipendenze e aggiunge tutti gli elementi dipendenti al pacchetto. Ad esempio, se si aggiunge un workflow che utilizza azioni o altri workflow, vRealize Orchestrator aggiunge tali azioni e workflow al pacchetto.

Quando si importa un pacchetto, il server confronta le versioni dei diversi elementi del suo contenuto con gli elementi locali corrispondenti. Il confronto mostra le differenze delle versioni tra gli elementi locali e quelli importati. L'utente può decidere se importare il pacchetto oppure selezionare elementi specifici da importare.

Per la maggior parte degli oggetti creati in vRealize Orchestrator Client, a parte gli elementi delle risorse, i pacchetti sono l'unico modo per esportare e importare tali oggetti.

I pacchetti utilizzano Digital Rights Management per controllare il modo in cui il server ricevente può utilizzare il contenuto del pacchetto. vRealize Orchestrator firma i pacchetti e li crittografa per la protezione dei dati. I pacchetti possono tenere traccia degli utenti che esportano e ridistribuiscono gli elementi utilizzando certificati X509.

## Creazione di un pacchetto in vRealize Orchestrator Client

È possibile esportare e importare workflow, criteri, azioni, riferimenti ai plug-in, elementi di risorse ed elementi di configurazione nei pacchetti. Tutti gli elementi dipendenti correlati agli oggetti del pacchetto vengono aggiunti automaticamente al pacchetto per garantire la compatibilità tra le versioni. Per eliminare elementi dipendenti, è innanzitutto necessario rimuovere l'oggetto del pacchetto correlato.

Per la maggior parte degli oggetti creati in vRealize Orchestrator Client, a parte gli elementi delle risorse, i pacchetti sono l'unico modo per esportare e importare tali oggetti.

### Prerequisiti

Verificare che il server vRealize Orchestrator contenga oggetti come workflow, azioni e criteri, che è possibile aggiungere a un pacchetto.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Asset > Pacchetti**.
- 3 Fare clic su **Nuovo pacchetto**.
- 4 Nella scheda **Generale** immettere un nome e una descrizione per il pacchetto.

---

**Nota** Non è possibile utilizzare caratteri speciali quando si attribuisce un nome ai pacchetti in vRealize Orchestrator Client.

---

- 5 Nella scheda **Contenuto**, fare clic su **Aggiungi**.
- 6 Selezionare gli oggetti che si desidera aggiungere al pacchetto e fare clic su **Aggiungi**.

---

**Nota** Gli elementi dipendenti vengono aggiunti automaticamente al pacchetto, ma non vengono visualizzati nella scheda **Contenuto** durante la creazione del pacchetto. Per visualizzare gli elementi dipendenti, selezionare la scheda **Contenuto** dopo la creazione del pacchetto.

---

- 7 Per completare la creazione del pacchetto, fare clic su **Crea**.

## Esportazione di un pacchetto in vRealize Orchestrator Client

È possibile utilizzare vRealize Orchestrator Client per esportare pacchetti in un altro ambiente di vRealize Orchestrator.

### Prerequisiti

Creare un pacchetto contenente gli oggetti di vRealize Orchestrator che si desidera esportare. Per ulteriori informazioni, vedere [Creazione di un pacchetto in vRealize Orchestrator Client](#).

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Asset > Pacchetti**.
- 3 Fare clic su **Esporta** nel pacchetto.

#### 4 (Facoltativo) Selezionare opzioni di esportazione aggiuntive.

Opzione	Descrizione
<b>Aggiungi valori dell'attributo di configurazione al pacchetto</b>	Esporta i valori degli attributi degli elementi di configurazione.
<b>Aggiungi valori dell'attributo SecureString di configurazione al pacchetto</b>	Esporta i valori dell'attributo di configurazione SecureString.
<b>Aggiungi tag globali al pacchetto</b>	Esporta i tag globali.

#### 5 Impostare i diritti di accesso per gli utenti che importano il pacchetto.

Opzione	Descrizione
<b>Visualizza contenuti</b>	L'utente può visualizzare i contenuti del pacchetto.
<b>Aggiungi al pacchetto</b>	L'utente può aggiungere i contenuti dal pacchetto importato ad altri pacchetti.
<b>Modifica contenuti</b>	L'utente può modificare i contenuti del pacchetto.

#### 6 Fare clic su **OK**.

**Nota** I file con estensione `.package` vengono salvati in una cartella predefinita nella macchina locale. Per impostare una cartella personalizzata, è possibile modificare le impostazioni di storage nel browser.

#### Risultati

Il pacchetto è stato esportato. Ora è possibile utilizzare gli oggetti esportati in un altro ambiente di vRealize Orchestrator.

## Importazione di un pacchetto in vRealize Orchestrator Client

È possibile utilizzare vRealize Orchestrator Client per importare pacchetti di workflow. Importando i pacchetti, è possibile riutilizzare gli oggetti di un server vRealize Orchestrator su un altro server.

#### Prerequisiti

- Eseguire il backup di tutti gli oggetti standard di vRealize Orchestrator che sono stati modificati.
- Nel server remoto, creare ed esportare un pacchetto con gli oggetti che si desidera importare.

#### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Asset > Pacchetti**.



- 3** Fare clic su **Importa**, individuare il file `.package` che si desidera importare e fare clic su **Apri**.
- 4** Rivedere le informazioni del pacchetto importate.
  - a La scheda **Generale** include informazioni sul pacchetto importato, come il nome, la descrizione, il numero di elementi contenuti e le informazioni sul certificato.

Prima di poter importare il file, potrebbe essere necessario indicare che si considera attendibile il certificato dell'editore dell'istanza di vRealize Orchestrator di origine.
  - b Nella scheda **Elementi pacchetto** sono elencati gli oggetti inclusi nel file di importazione. Se la versione di un oggetto nel pacchetto è successiva alla versione nel server, il sistema seleziona tale versione dell'oggetto per l'importazione. Le versioni precedenti degli elementi di vRealize Orchestrator devono essere selezionate manualmente.
  - c Deselezionare **Importa valori attributo di configurazione** se non si desidera importare i valori degli attributi degli elementi di configurazione dal pacchetto.
  - d Nel menu a discesa, selezionare se si desidera importare i tag.
- 5** Fare clic su **Importa**.

# Risoluzione dei problemi in vRealize Orchestrator Client

# 11

È possibile monitorare l'istanza di vRealize Orchestrator e risolverne i problemi utilizzando le metriche, la riproduzione del token, la convalida e il debug.

Questo capitolo include i seguenti argomenti:

- [Dati metrici in vRealize Orchestrator Client](#)
- [Utilizzo della riproduzione del token del workflow in vRealize Orchestrator Client](#)
- [Convalida dei workflow di vRealize Orchestrator](#)
- [Debug degli script dei workflow in vRealize Orchestrator Client](#)
- [Workflow di debug in base all'elemento dello schema](#)

## Dati metrici in vRealize Orchestrator Client

Gli amministratori di vRealize Orchestrator possono utilizzare la profilatura del workflow e le metriche del dashboard di sistema per risolvere i problemi relativi al sistema e ai workflow di vRealize Orchestrator.

La funzionalità di profilatura raccoglie i dati metrici sulle esecuzioni dei workflow. La profilatura del workflow è abilitata per impostazione predefinita. È possibile disabilitare la profilatura automatica in **Centro di controllo > Proprietà estensione > profiler-8.2.0**.

L'altra origine per i dati metrici in vRealize Orchestrator Client è il dashboard di sistema, che fornisce le metriche a livello di sistema. Per ulteriori informazioni, vedere [Utilizzo del dashboard del sistema vRealize Orchestrator](#).

## Profilatura dei workflow in vRealize Orchestrator Client

È possibile profilare le esecuzioni dei workflow per risolvere i problemi e ottimizzare l'ambiente di vRealize Orchestrator.

È possibile utilizzare la funzionalità di profilatura di vRealize Orchestrator Client per raccogliere dati metrici utili sulle esecuzioni dei workflow. Questi dati possono essere utilizzati per ottimizzare le prestazioni dei workflow. Per impostazione predefinita, le esecuzioni dei workflow vengono profilate automaticamente. È possibile disabilitare la profilatura automatica dalla pagina **Proprietà estensione** del Centro di controllo di vRealize Orchestrator ed eseguire il profiler manualmente. Per eseguire un'operazione di profilatura manuale, individuare il workflow nella libreria e selezionare **Azioni > Profilo**.

#### Prerequisiti

Eseguire un workflow.

#### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Attività > Esecuzioni workflow**.
- 3 Selezionare un'esecuzione di workflow.

Nello schema dell'esecuzione del workflow, è possibile visualizzare i dati relativi ai singoli elementi del workflow. I dati includono la durata totale dell'esecuzione, la durata massima e il numero di esecuzioni di elementi. È possibile filtrare queste informazioni dal menu a discesa in alto a destra della pagina.

- 4 Selezionare la scheda **Prestazioni**.

Questa scheda include dati metrici su workflow, tempi CPU dell'esecuzione del workflow, durata dell'esecuzione, dimensioni del token e dati degli elementi del workflow.

---

**Nota** Se l'esecuzione del workflow viene sospesa, ad esempio quando il workflow è in attesa di ulteriore input, la metrica dei tempi CPU acquisisce solo il thread di runtime che si è verificato prima del completamento.

---

#### Operazioni successive

Utilizzare i dati raccolti tramite la profilatura per ottimizzare il workflow.

## Utilizzo del dashboard del sistema vRealize Orchestrator

Un amministratore può utilizzare il dashboard del sistema vRealize Orchestrator Client per raccogliere dati metrici utili sui nodi dell'ambiente di vRealize Orchestrator.

È possibile accedere al dashboard del sistema facendo clic sulla scheda **Sistema** nella parte superiore della pagina del dashboard di vRealize Orchestrator Client. I dati forniti includono:

- Stato del nodo
- Proprietà del nodo
- Impostazioni del cluster. È possibile visualizzare le impostazioni del cluster solo dal dashboard del sistema. Per modificare queste impostazioni, passare alla pagina **Gestione cluster di Orchestrator** del Centro di controllo di vRealize Orchestrator.

- Informazioni thread
- Memoria heap
- Memoria non heap
- Utilizzo del file system
- Dati di autenticazione
- Pool di connessioni del database di Orchestrator
- Argomenti dell'input del processo

Questi dati possono essere utilizzati per monitorare lo stato dei singoli nodi dell'ambiente di vRealize Orchestrator e risolvere i problemi. Per passare da un nodo all'altro, fare clic sulla scheda associata a un nodo nella parte superiore del dashboard del sistema.

## Utilizzo della riproduzione del token del workflow in vRealize Orchestrator Client

È possibile utilizzare la funzionalità di riproduzione del token per visualizzare le transizioni tra gli elementi nelle esecuzioni dei workflow.

La funzionalità di riproduzione del token registra le informazioni contestuali per ogni transizione tra gli elementi del workflow. Per ogni elemento del workflow, la riproduzione del token registra quando l'esecuzione del workflow è iniziata, quando è finita e quali variabili sono state modificate al termine dell'esecuzione dell'elemento del workflow. La riproduzione del token fa anche riferimento ai messaggi del registro degli script generati per ogni elemento del workflow.

---

**Nota** I dati relativi alle transizioni degli elementi del workflow vengono archiviati nel database PostgreSQL di vRealize Orchestrator. Questi dati vengono rimossi dal database quando l'esecuzione del workflow viene eliminata.

---

### Prerequisiti

- Abilitare la funzionalità di riproduzione del token dal Centro di controllo.
  - a Accedere al Centro di controllo come **root**.
  - b Selezionare **Proprietà estensione**.
  - c Fare clic su **tokenreplay-8.2.0**.
  - d Per abilitare la funzionalità di riproduzione del token, fare clic su **Abilita**.
  - e Fare clic su **Salva**.

---

**Nota** Il server vRealize Orchestrator può impiegare fino a 5 minuti per aggiornare l'estensione.

---

- Eseguire un workflow.

---

**Nota** Per impostazione predefinita, la riproduzione del token non viene eseguita automaticamente per tutte le esecuzioni dei workflow nel server vRealize Orchestrator. È possibile eseguire la riproduzione del token per ogni workflow singolarmente oppure abilitare l'estensione di riproduzione del token per tutti i workflow dalla pagina **Proprietà estensione** del Centro di controllo.

---

#### Procedura

- 1 (Facoltativo) Abilitare la riproduzione del token per tutte le esecuzioni dei workflow nel server vRealize Orchestrator.

---

**Nota** Per eseguire singole riproduzioni del token senza abilitare la funzionalità dal Centro di controllo, fare clic su **Esegui con riproduzione** nella pagina dell'editor di workflow.

---

- a Accedere al Centro di controllo come **root**.
- b Selezionare **Proprietà estensione**.
- c Fare clic su **tokenreplay-8.2.0**.
- d Per abilitare la funzionalità di riproduzione del token per tutti i workflow, verificare che sia abilitata la funzionalità **Registra riproduzione per tutte le esecuzioni di workflow**.
- e Fare clic su **Salva**.

---

**Nota** Il server vRealize Orchestrator può impiegare fino a 5 minuti per aggiornare l'estensione.

---

- 2 Accedere a vRealize Orchestrator Client come amministratore.
- 3 Passare a **Attività > Esecuzioni workflow**.
- 4 Selezionare un'esecuzione di workflow.
- 5 Selezionare un elemento dell'esecuzione del workflow nel menu a sinistra.

Nelle schede **Variabile** e **Registri** vengono ora visualizzate informazioni specifiche per tale elemento del workflow.

## Convalida dei workflow di vRealize Orchestrator

vRealize Orchestrator fornisce uno strumento di convalida dei workflow. La convalida di un workflow consente di identificare gli errori del workflow e verifica che i dati passino correttamente da un elemento all'altro.

Per impostazione predefinita, quando si esegue un workflow, vRealize Orchestrator esegue sempre una convalida del workflow.

Quando si convalida un workflow, lo strumento di convalida crea un elenco di eventuali errori o avvisi. Se si fa clic su un errore nell'elenco, viene evidenziato l'elemento del workflow che contiene l'errore.

Se si esegue lo strumento di convalida nell'editor di workflow, lo strumento fornisce correzioni rapide suggerite per gli errori rilevati. Alcune correzioni rapide richiedono ulteriori informazioni o parametri di input. Altre correzioni rapide risolvono automaticamente l'errore.

La convalida del workflow controlla le associazioni dei dati e le connessioni tra gli elementi. La convalida del workflow non verifica l'elaborazione dei dati eseguita da ogni elemento del workflow. Di conseguenza, è possibile che un workflow valido non venga eseguito correttamente e generi risultati errati se una funzione in un elemento dello schema non è corretta.

## Convalida di un workflow e correzione degli errori di convalida in vRealize Orchestrator Client

Prima di poter eseguire un workflow, è necessario convalidarlo. È possibile correggere gli errori di convalida solo se si apre il workflow per la modifica.

### Prerequisiti

Verificare di disporre di un workflow completo da convalidare, con gli elementi dello schema collegati e le associazioni definite.

### Procedura

- 1 Accedere a vRealize Orchestrator Client come amministratore.
- 2 Passare a **Libreria > Workflow** e selezionare il workflow che si desidera convalidare.
- 3 Fare clic su **Modifica**.
- 4 Fare clic su **Convalida** nel menu in alto.

Se il workflow è valido, viene visualizzato un messaggio di conferma. Se il workflow non è valido, viene visualizzato un elenco di errori.

- 5 Per un workflow non valido, fare clic su un messaggio di errore ed eseguire i passaggi appropriati per risolvere il problema.

Lo strumento di convalida evidenzia l'elemento dello schema in cui si verifica l'errore aggiungendovi un'icona rossa. Se possibile, lo strumento di convalida visualizza un'azione di correzione rapida.

- Se si accetta l'azione di correzione rapida proposta, fare clic su tale azione per eseguirla.
- Se non si accetta l'azione di correzione rapida proposta, chiudere la finestra di dialogo di convalida del workflow e correggere manualmente l'elemento dello schema.

---

**Importante** Verificare sempre che la correzione proposta da vRealize Orchestrator sia appropriata.

---

Ad esempio, l'azione proposta potrebbe essere quella di eliminare un attributo non utilizzato, mentre in realtà tale attributo potrebbe non essere associato correttamente.

- 6 Ripetere i passaggi precedenti finché non sono stati eliminati tutti gli errori di convalida.

#### Risultati

È stato convalidato un workflow e sono stati corretti gli errori di convalida.

#### Operazioni successive

È possibile eseguire il workflow.

## Debug degli script dei workflow in vRealize Orchestrator Client

È possibile eseguire il debug delle esecuzioni dei workflow inserendo punti di interruzione nello script degli elementi dei workflow.

Quando viene raggiunto un punto di interruzione, sono disponibili diverse opzioni per continuare il processo di debug. Quando si esegue il debug di un elemento dallo schema del workflow, è possibile visualizzare le informazioni generali sull'esecuzione del workflow, modificare le variabili del workflow, aggiungere espressioni da controllare e visualizzare messaggi dei registri.

---

**Nota** Eseguire il debug di tutti gli script in un ambiente non di produzione.

---

#### Procedura

- 1 Accedere a vRealize Orchestrator Client come amministratore.
- 2 Selezionare un workflow nella libreria.
- 3 Aprire lo schema del workflow, selezionare un elemento del workflow e fare clic sulla scheda **Scripting**.
- 4 Per inserire un punto di interruzione, fare clic sul cerchio rosso a sinistra del numero di riga.

---

**Nota** È possibile inserire punti di interruzione solo negli elementi del workflow con scripting.

---

- 5 Per eseguire il workflow in modalità di debug, fare clic su **Debug**.  
Se il workflow richiede parametri di input, è necessario specificarli.

- 6 Quando l'esecuzione del workflow viene sospesa dopo il raggiungimento di un punto di interruzione, selezionare una delle opzioni disponibili.

Opzione	Descrizione
<b>Continua</b>	Riprende l'esecuzione del workflow finché non viene raggiunto un altro punto di interruzione o non termina l'esecuzione del workflow.
<b>Passaggio in</b>	È possibile utilizzare questa opzione per passare a un elemento del workflow. Non è possibile passare a un elemento del workflow nidificato quando si esegue il debug di un workflow nell'editor di workflow.
<b>Passaggio su</b>	Ignora l'elemento corrente nello schema e sospende l'esecuzione del workflow nell'elemento successivo.

**Nota** È possibile indicare al debugger di ignorare il punto di interruzione corrente facendo clic sul punto di interruzione. Il simbolo del punto di interruzione diventa un triangolo verde.

- 7 (Facoltativo) Nella scheda **Debugger**, inserire le espressioni da controllare.  
È possibile utilizzare le espressioni per seguire il completamento di variabili specifiche.
- 8 (Facoltativo) Nella scheda **Debugger**, modificare i valori delle variabili.

## Workflow di debug in base all'elemento dello schema

Un designer del workflow può eseguire il debug di singoli elementi dello schema.

### Procedura

- 1 Accedere a vRealize Orchestrator Client.
- 2 Passare a **Libreria > Workflow** e selezionare il workflow.
- 3 Selezionare la scheda **Schema**.
- 4 Selezionare l'elemento del workflow di cui si desidera eseguire il debug e fare clic sul pulsante Debug in alto a sinistra dell'elemento.

**Nota** Aggiungendo un punto di interruzione a un elemento dello schema **Elemento workflow**, è possibile eseguire il debug dei workflow secondari direttamente dal workflow principale. Quando il debugger raggiunge l'elemento dello schema **Elemento workflow**, apre la visualizzazione dello schema del workflow secondario.

- 5 Ripetere la procedura per tutti gli altri elementi dello schema di cui si desidera eseguire il debug.
- 6 Fare clic su **Debug**.
- 7 Immettere i valori dei parametri di input richiesti e fare clic su **Esegui**.

L'esecuzione del workflow inizia e viene sospesa quando il debugger raggiunge un elemento dello schema con un punto di interruzione.



## 8 Quando si è in un punto di interruzione, selezionare una delle seguenti opzioni:

Opzione	Descrizione
<b>Continua</b>	Riprende l'esecuzione del workflow finché non viene raggiunto un altro punto di interruzione o non termina l'esecuzione del workflow.
<b>Passaggio in</b>	Esegue l'istruzione della funzione del flusso di lavoro corrente. Se il debugger non può andare ulteriormente in profondità nella riga della funzione corrente, esegue un'operazione <b>Passaggio su</b> .
<b>Passaggio su</b>	Il debugger continua nella riga successiva della funzione corrente.
<b>Ritorno passaggio</b>	Il debugger passa alla riga che verrà eseguita durante la restituzione della funzione corrente.

## 9 (Facoltativo) Nella scheda **Variabili**, modificare il valore delle variabili del workflow.

The screenshot displays the VMware vRealize Orchestrator Client interface during a debugging session. The top bar shows the workflow name 'debug.by.schema test' and its status 'Debugging'. Below this, a visual workflow diagram is shown on the left, with a green arrow indicating the current execution point. The right pane is divided into two sections: 'General' and 'Scripting'. The 'Scripting' section shows a JavaScript script for 'Item1 (My Orchestrator Task)'. The script includes a loop to iterate over an array 'VMs' and a function call 'workflowToLaunch'. The bottom section of the interface is the 'Debugger' tab, which contains a 'Watch expressions' table on the left and a 'Variables' table on the right. The 'Watch expressions' table has columns for 'Expression' and 'Value', with a button 'CLICK TO ADD AN EXPRESSION'. The 'Variables' table lists several variables and their current values:

Variable	Value
i	Not set
VM	test
workflowToLaunch	Not set
workflowParameters	Not set
wfToken	Not set
VM	test
workflowToRun	test