

De VMware vRealize Orchestrator-client gebruiken

6 OKTOBER 2020

vRealize Orchestrator 8.2

U vindt de recentste technische documentatie op de website van VMware:

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

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

VMware Nederland B.V.
Key Office Papendorp
3e verdieping
Orteliuslaan 850
Utrecht
Nederland
Tel: +31 (0) 30-2849500
Fax: +31 (0) 30- 2849501
www.vmware.com/nl

Copyright © 2008-2020 VMware, Inc. Alle rechten voorbehouden. [Informatie over copyright en handelsmerken.](#)

Inhoud

1	De VMware vRealize Orchestrator-client gebruiken	5
2	De VMware vRealize Orchestrator-client	6
	Het gebruikersdashboard van de vRealize Orchestrator-client	7
	Organisatie van de inhoud in de vRealize Orchestrator Client	8
	Een map of submap in de vRealize Orchestrator-client maken	9
3	De vRealize Orchestrator Client instellen	12
	Rollen en groepen van de vRealize Orchestrator-client beheren	12
	Rollen toewijzen in de vRealize Orchestrator Client	14
	Groepen in de vRealize Orchestrator Client maken	14
	vRealize Orchestrator-clientrollen in vRealize Automation configureren	15
	Versiegeschiedenis van vRealize Orchestrator-objecten	16
	Een werkstroom terugzetten naar een eerdere versie	16
	Visuele vergelijking tussen werkstroomversies	17
	De inventaris van uw vRealize Orchestrator-inhoud opnieuw instellen op een eerdere status met Git	18
4	vRealize Orchestrator-toepassingsvoorbeelden	20
	Amazon Web Services in vRealize Orchestrator integreren met behulp van Python	20
	Eerste Python-script maken	21
	De Amazon Web Services-actie maken	22
	Foutopsporing voor de actie Amazon Web Services uitvoeren	23
	De Amazon Web Services-actie bijwerken	27
	Git-vertakkingen gebruiken om mijn vRealize Orchestrator-objectinventaris te beheren	27
	Uw GitLab-omgeving voorbereiden	28
	Een verbinding met een Git-opslagplaats configureren	29
	Wijzigingen naar een Git-opslagplaats pushen	30
	Modules van derden gebruiken om de vRealize Automation-project-API aan te roepen	33
	Een Python-script maken dat de vRealize Automation Project-API aanroept	33
	Een Node.js-script maken dat de vRealize Automation Project-API aanroept	35
	Een PowerShell-script maken dat de vRealize Automation Project-API aanroept	38
5	Werkstromen beheren	41
	Standaardwerkstromen in de vRealize Orchestrator-werkstroombibliotheek	42
	Werkstromen maken in de vRealize Orchestrator-client	42
	Werkstromen en acties vanuit de bovenliggende werkstroom bewerken	43
	Ontwerpfunctie voor vRealize Orchestrator-invoerformulier	43

	Het dialoogvenster met invoerparameters voor de werkstroom maken in de vRealize Orchestrator-client	44
	Eigenschappen voor invoerparameters in de vRealize Orchestrator-client	44
	Acties gebruiken om vRealize Orchestrator-werkstroominvoer te valideren	45
	Aanvragen voor gebruikersinteractie in de vRealize Orchestrator-client	47
	Werkstromen plannen op de vRealize Orchestrator-client	47
	Een geplande taak in de vRealize Orchestrator-client bewerken	48
6	Acties beheren	49
	Acties in de vRealize Orchestrator-client maken	49
	Acties uitvoeren en fouten opsporen	50
	Acties uitvoeren in de vRealize Orchestrator-client	51
	Acties voor foutopsporing in de vRealize Orchestrator-client	51
	Basisconcepten voor Python-, Node.js- en PowerShell-scripts	52
	Runtimelimieten voor Python, Node.js en PowerShell-scripts	54
7	Configuratie-elementen beheren	55
	Configuratie-elementen maken in de vRealize Orchestrator-client	55
8	Beleid beheren	57
	Beleidsregels maken en toepassen op de vRealize Orchestrator-client	57
	Beleidselementen in de vRealize Orchestrator-client	58
	Beleidsuitvoeringen in de vRealize Orchestrator-client beheren	59
9	Resource-elementen beheren	60
10	Pakketten beheren	61
	Een pakket maken in de vRealize Orchestrator-client	61
	Een pakket in de vRealize Orchestrator-client exporteren	62
	Een pakket importeren in de vRealize Orchestrator-client	63
11	Problemen oplossen in de vRealize Orchestrator-client	65
	Metrische gegevens in de vRealize Orchestrator-client	65
	Werkstromen in de vRealize Orchestrator-client profileren	65
	Het vRealize Orchestrator-systeemdashboord gebruiken	66
	Herhaling van werkstroomtoken in de vRealize Orchestrator-client gebruiken	67
	vRealize Orchestrator-werkstromen valideren	68
	Een werkstroom valideren en validatiefouten corrigeren in de vRealize Orchestrator-client	69
	Scripts voor foutopsporingswerkstroom in de vRealize Orchestrator-client	70
	Foutopsporing uitvoeren in werkstroom per schema-element	71

De VMware vRealize Orchestrator-client gebruiken

1

In *De VMware vRealize Orchestrator-client gebruiken* vindt u informatie over de functies voor werkstroomautomatisering van de vRealize Orchestrator Client.

Doelgroep

Deze informatie is bedoeld voor ervaren systeembeheerders die op zoek zijn naar een tool die hen kan helpen bij het uitvoeren en beheren van vRealize Orchestrator-werkstromen.

De VMware vRealize Orchestrator-client

2

Gebruik de vRealize Orchestrator Client om vRealize Orchestrator-services en -objecten te beheren.

U kunt vRealize Orchestrator-objecten maken en beheren met behulp van de vRealize Orchestrator Client. U vindt de vRealize Orchestrator Client op https://uw_orchestrator_FQDN/orchestration-ui.

REST API-communicatie

De vRealize Orchestrator Client wordt uitgevoerd op de vRealize Orchestrator Control Center-server. De client communiceert met de vRealize Orchestrator REST API via een REST-proxy.

Werkstroombeheer

Werkstromen in de vRealize Orchestrator Client maken, bewerken, plannen, uitvoeren en verwijderen.

Actiebeheer

Acties in de vRealize Orchestrator Client maken, bewerken en verwijderen. De actie-editor ondersteunt automatisch aanvullen voor algemene scriptelementen die zijn opgenomen in de vRealize OrchestratorAPI Explorer.

Beleidsbeheer

Beleidsregels in de vRealize Orchestrator Client maken, bewerken, uitvoeren en verwijderen.

Configuratiebeheer

Configuratie-elementen in de vRealize Orchestrator Client maken, uitvoeren en verwijderen.

Resourcebeheer

Resource-elementen in de vRealize Orchestrator Client exporteren, importeren en bijwerken.

Git-integratie

Maak een integratie met een Git-opslagplaats en gebruik de integratie om de ontwikkeling van werkstromen en andere vRealize Orchestrator te beheren in meerdere implementaties. Zie [Git-vertakkingen gebruiken om mijn vRealize Orchestrator-objectinventaris te beheren](#).

Metriekgegevens

Gebruik het vRealize Orchestrator Client-systeemdashbord en de profilingfunctie om nuttige metriekgegevens over uw vRealize Orchestrator-omgeving en -werkstromen te verzamelen.

Pakketbeheer

Met de vRealize Orchestrator Client kunt u pakketten met vRealize Orchestrator-objecten maken, verwijderen, exporteren en importeren.

Rechtenbeheer

Gebruikers met beheerdersrechten kunnen rollen toewijzen aan gebruikers in de vRealize Orchestrator Client en ze aan groepen toevoegen.

API Explorer

Bekijk de API-opdrachten die beschikbaar zijn in de vRealize Orchestrator Client.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Het gebruikersdashboard van de vRealize Orchestrator-client](#)
- [Organisatie van de inhoud in de vRealize Orchestrator Client](#)

Het gebruikersdashboard van de vRealize Orchestrator-client

Het vRealize Orchestrator Client-dashbord biedt een nuttige tool om vRealize Orchestrator Client-werkstromen te controleren, te beheren en problemen op te lossen.

Informatie over het vRealize Orchestrator Client-dashbord is verdeeld over vijf panelen.

Venster	Beschrijving
Werkstroomuitvoeringen	Biedt visuele gegevens over het aantal actieve, wachtende en mislukte werkstroomuitvoeringen.
Favoriete werkstromen	Geeft werkstromen weer die aan favorieten zijn toegevoegd.
Wachten op invoer	Geeft werkstroomuitvoeringen in behandeling weer waarvoor verdere gebruikersinteractie is vereist. Deze werkstromen worden ook weergegeven in het meldingenmenu in de rechterbovenhoek van de gebruikersinterface.

Venster	Beschrijving
Recente werkstroomuitvoeringen	Recente werkstroomuitvoeringen beheren. Toont de naam, de status, de begindatum en de einddatum van de werkstroomuitvoering.
Aandacht vereist	Toont mislukte werkstroomuitvoeringen en prestatietriecken van werkstroomuitvoeringen.

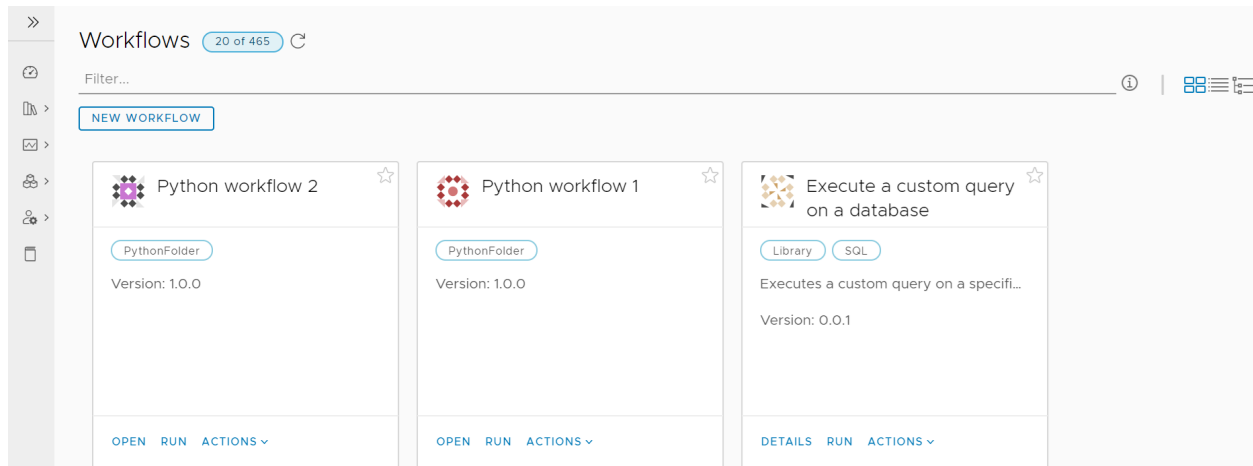
Organisatie van de inhoud in de vRealize Orchestrator Client

Beheer hoe uw vRealize Orchestrator-objectinventaris wordt weergegeven in de vRealize Orchestrator Client.

De vRealize Orchestrator Client ondersteunt drie verschillende weergavetypen voor objecten zoals werkstromen, acties, beleidsregels, resources en configuraties: kaartweergave, lijstweergave en structuurweergave. U kunt het huidige weergavetype wijzigen in de rechterbovenhoek van de pagina.

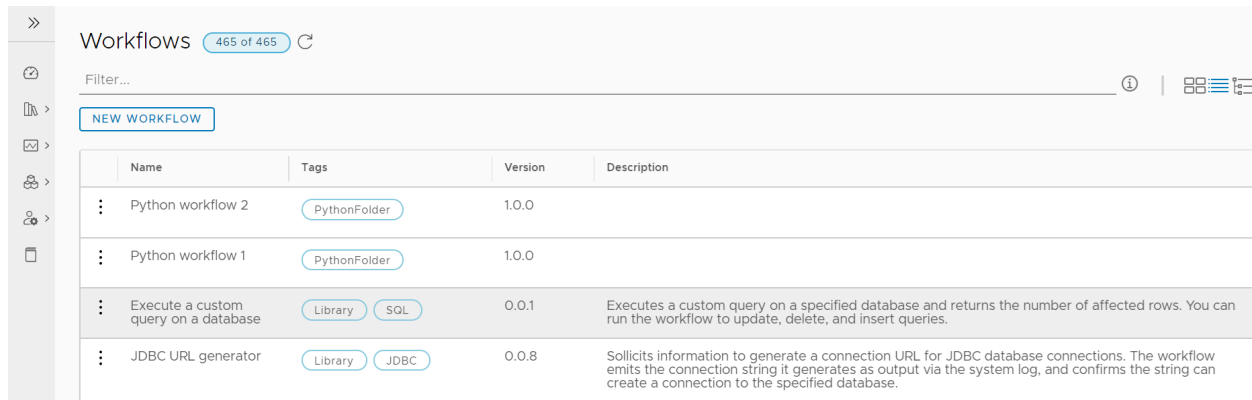
Kaartweergave

De kaartweergave is het standaardweergavetype dat wordt gebruikt in de vRealize Orchestrator Client. Informatie over het afzonderlijke inventarisobject, zoals een werkstroom, wordt weergegeven in een apart kaartelement.



Lijstweergave

Lijstweergave geeft informatie weer over uw vRealize Orchestrator-objecten, geordend als een lijst. Voor meer informatie over de acties die u kunt uitvoeren op het object, klikt u op het pictogram met drie verticale puntjes links van het object.



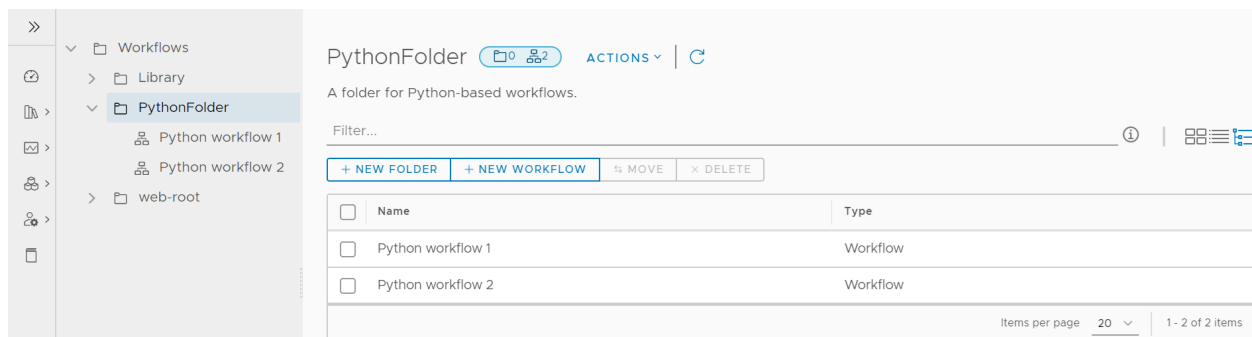
Structuurweergave

U kunt uw objectinventaris onder hiërarchische mappen in de structuurweergave ordenen. Elk vRealize Orchestrator-objecttype heeft een map op rootniveau. U kunt geen nieuwe objecten, zoals werkstromen, maken in de hoofdmap. U moet afzonderlijke mappen maken die zijn geordend onder de hoofdmap. Elke map bevat hulpmiddelen om u te helpen bij het beheren van de inhoud, zoals een inhoudsfilter.

Opmerking Elke map heeft een afzonderlijk inhoudsfilter. U kunt geen inhoud op meerdere mappen filteren.

Zie [Een map of submap in de vRealize Orchestrator-client maken](#) voor meer informatie over mappen.

Opmerking Wanneer u een object in de structuurweergave selecteert, wordt het in een alleen-lezen-modus geopend. Als u de objectinhoud, zoals werkstroomvariabelen of het werkstroomschema, wilt bewerken, klikt u op **Bewerken** in het bovenste optiemenu.



Een map of submap in de vRealize Orchestrator-client maken


Organiseer uw vRealize Orchestrator-objecten met een hiërarchische mapstructuur.

U kunt mappen en submappen maken om de volgende typen vRealize Orchestrator-objecten te organiseren:

- Werkstromen

- Acties
- Beleid
- Configuratie-elementen
- Resource-elementen

Procedure


- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Selecteer in het linkernavigatievenster een objectpagina, zoals **Werkstromen**.
- 3 Selecteer in de rechterbovenhoek het pictogram van de structuurweergave ()
- 4 (Optioneel) Als u een submap wilt maken, selecteert u een bovenliggende map in de boomstructuurweergave aan de linkerkant.
- 5 Klik op **Nieuwe map**.
- 6 Voer een naam en beschrijving in en klik op **Opslaan**.
- 7 Voeg objecten of submappen toe aan de nieuwe map.
- 8 (Optioneel) Als u de naam van de map wilt bewerken, selecteert u **Acties > Bewerken**.

Objecten en mappen in de vRealize Orchestrator Client verplaatsen

Organiseer uw vRealize Orchestrator-inhoud opnieuw door de inhoud naar een andere map te verplaatsen.

U kunt geen acties tussen actiemodules verplaatsen of objecten naar een rootmap verplaatsen. De rootmap bevat de hoofdobjectmappen en submappen, maar kan niet worden gebruikt om objecten op te slaan.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Selecteer in het linkernavigatievenster een objectpagina, zoals **Werkstromen**.
- 3 Selecteer in de rechterbovenhoek het pictogram van de structuurweergave ()
- 4 Vouw de structuurweergave uit en selecteer het object of de map om te verplaatsen.
- 5 Sleep het object of de map naar de nieuwe bovenliggende map.


Opmerking U kunt objecten ook direct vanuit de objecteditor naar nieuwe mappen verplaatsen. Klik op het tabblad **Samenvatting** op **Map selecteren** en selecteer de nieuwe bovenliggende map voor het object. Een andere verplaatsingsoptie is het selecteren van objecten uit de tabel op de mappagina. Deze optie is handig voor het uitvoeren van batchverplaatsingsbewerkingen die meerdere vRealize Orchestrator-objecten bevatten.

Een map of submap in de vRealize Orchestrator Client verwijderen

Verwijder verouderde mappen of submappen van uw vRealize Orchestrator Client.

U kunt de bijbehorende map op het hoofdniveau van elk vRealize Orchestrator-objecttype niet verwijderen.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Selecteer in het linkernavigatievenster een objectpagina, zoals **Werkstromen**.
- 3 Selecteer in de rechterbovenhoek het pictogram van de structuurweergave ().
- 4 Schakel het selectievakje in naast de map die u wilt verwijderen.

Opmerking Als u een submap wilt verwijderen, selecteert u de bovenliggende map in de structuurweergave en schakelt u het selectievakje in.

- 5 Klik op **Verwijderen**.
- 6 Als de geselecteerde map leeg is.
 - a Bevestig dat u de map wilt verwijderen.
 - b Klik op **Verwijderen**.
- 7 Als de geselecteerde map vRealize Orchestrator Client-objecten of submappen bevat.
 - a Bevestig dat u de map wilt verwijderen.
 - b Klik op **Verwijderen**.

U ontvangt het bericht Kan item 'your_folder_name' niet verwijderen: Map 'your_folder_name' is niet leeg.
 - c Als u de map en alle inhoud ervan wilt verwijderen, klikt u op **Verwijderen forceren**.
 - d Bevestig dat u de map wilt verwijderen en klik op **Verwijderen**.

Opmerking U kunt ook een batchverwijdering uitvoeren door meerdere objecten te selecteren in de tabel die in het mapmenu is opgenomen.

De vRealize Orchestrator Client instellen

3

Om de functionaliteit van de vRealize Orchestrator Client optimaal te benutten, moet u uw gebruikersrechten configureren en leren hoe u de versiegeschiedenis kunt gebruiken om uw objecten te beheren.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Rollen en groepen van de vRealize Orchestrator-client beheren](#)
- [Versiegeschiedenis van vRealize Orchestrator-objecten](#)

Rollen en groepen van de vRealize Orchestrator-client beheren

Als beheerder kunt u de vRealize Orchestrator Client gebruiken om gebruikersrollen en groepsrechten voor vRealize Orchestrator-functies en -inhoud in te stellen.

Nadat de vRealize Orchestrator-instantie is geverifieerd, kan de beheerder rechten instellen die de toegang tot functies en inhoud beheren. Rechten in de vRealize Orchestrator Client zijn onderverdeeld in rollenbeheer en groepsrechten. Met rollenbeheer kunt u bepalen welke vRealize Orchestrator Client-functies gebruikers kunnen bekijken en gebruiken. Met groepsrechten bepaalt u welke vRealize Orchestrator Client-inhoud gebruikers kunnen bekijken en gebruiken. Inhoudstoegang die wordt beheerd met groepsrechten omvat werkstromen, acties, beleidsregels, configuratie-elementen en resource-elementen. U kunt groepen gebruiken om gebruikers in algemene projecten te organiseren. U kunt bijvoorbeeld een groep maken die gebruikers bevat die werken aan het ontwikkelen van een aangepaste vRealize Orchestrator-invoegtoepassing.

Opmerking Toegang tot vooraf geconfigureerde vRealize Orchestrator-inhoud, zoals standaardwerkstromen en acties, wordt gedeeld door alle gebruikers, tenzij dit anders is ingesteld via groepsrechten.

Rollen

Het beheer van rollen aan de clientzijde is alleen beschikbaar voor vRealize Orchestrator-instanties die zijn geverifieerd met vSphere die een vRealize Automation-licentie gebruiken. Voor implementaties die gebruikmaken van een vRealize Automation-verificatie, moet u de functie Identiteits- en toegangsbeheer van vRealize Automation gebruiken. Zie [vRealize Orchestrator-clientrollen in vRealize Automation configureren](#).

Rol	Beschrijving
Beheerder	Heeft toegang tot alle vRealize Orchestrator Client-functies en -inhoud, inclusief de inhoud die door specifieke groepen is gemaakt. Verantwoordelijk voor het instellen van gebruikersrollen, het maken en verwijderen van groepen en het toevoegen van gebruikers aan groepen.
	Opmerking Tenantbeheerders uit de vRealize Automation-omgeving die wordt gebruikt om vRealize Orchestrator te verifiëren, hebben standaard beheerdersrechten .
Ontwerpfunctie voor werkstromen	Kan eigen vRealize Orchestrator Client-inhoud maken, uitvoeren, bewerken en verwijderen. Kan eigen inhoud toevoegen aan de eigen toegewezen groep. Heeft geen toegang tot de functies voor beheer en probleemoplossing van de vRealize Orchestrator Client.

Opmerking vRealize Automation gebruikers zonder een vooraf gedefinieerde rol kunnen zich nog steeds aanmelden bij de vRealize Orchestrator Client, maar hebben beperkte toegang tot clientfuncties. Als deze gebruikers deel uitmaken van een groep, kunnen ze inhoud bekijken en uitvoeren die aan die groep is gekoppeld.

Groepen

Groepsrechten in de vRealize Orchestrator Client kunnen worden gebruikt om meerdere gebruikers te verbinden die in een algemene vRealize Orchestrator werken, zoals het ontwikkelen van een aangepaste invoegtoepassing.

Groepsgebruikersrechten	Beschrijving
Uitvoeren en bewerken	Alleen beschikbaar voor vRealize Orchestrator-instanties die een vRealize Automation-licentie gebruiken. Kan vRealize Orchestrator-objecten maken, bewerken, toevoegen en uitvoeren voor gebruik in de groep.
Uitvoeren	Kan vRealize Orchestrator-objecten die in de groep zijn opgenomen, bekijken en uitvoeren.

Opmerking Groepsrechten zijn gekoppeld aan het systeem voor rollenbeheer in de vRealize Orchestrator Client. Gebruikers zonder een vooraf gedefinieerde rol kunnen bijvoorbeeld rechten voor **uitvoeren en bewerken** hebben, maar ze kunnen alleen hun eigen inhoud of groepsinhoud weergeven en uitvoeren, zonder de mogelijkheid om inhoud te maken, te bewerken en toe te voegen.

Rollen toewijzen in de vRealize Orchestrator Client

Als beheerder kunt u gebruikers toevoegen aan de vRealize Orchestrator Client en instellen welke functies ze kunnen bekijken en gebruiken.

Rollenbeheer regelt de toegang van gebruikers van de vRealize Orchestrator-aanbieder van identiteitsbeheer tot de functies van de vRealize Orchestrator Client. Rollenbeheer bestrijkt zowel de vRealize Orchestrator Client-gebruikersinterface als de API-functionaliteit.

Opmerking Het beheer van rollen aan de clientzijde is alleen beschikbaar voor vRealize Orchestrator-instanties die zijn geverifieerd met vSphere die een vRealize Automation-licentie gebruiken. Zie [vRealize Orchestrator-clientrollen in vRealize Automation configureren](#) voor informatie over het toewijzen van rollen aan vRealize Orchestrator die met vRealize Automation worden geverifieerd.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client als beheerder.
- 2 Navigeer naar **Beheer > Rollenbeheer**.
- 3 Klik op **Toevoegen**.
- 4 Zoek de gebruiker of groep die u wilt toevoegen aan de vRealize Orchestrator Client.
- 5 Selecteer de rol van de gebruiker. Zie [Rollen en groepen van de vRealize Orchestrator-client beheren](#) voor meer informatie over rollen.
- 6 Klik op **Opslaan**.

Groepen in de vRealize Orchestrator Client maken

Als beheerder kunt u groepen gebruiken om in te stellen wat vRealize Orchestrator-inhoudsgebruikers in de vRealize Orchestrator Client kunnen bekijken en openen.

U kunt de vRealize Orchestrator Client gebruiken om groepsrechten in te stellen voor vRealize Orchestrator-werkstromen, -acties, -beleidsregels, -configuratie-elementen, -bronelementen en -pakketten.

Opmerking Gebruikers van vRealize Orchestrator-instanties die met vSphere zijn geverifieerd, kunnen alleen groepsrechten voor **Uitvoeren** hebben.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client als beheerder.
- 2 Ga naar **Beheer > Groepen**.
- 3 Klik op **Nieuwe groep**.
- 4 Voeg op het tabblad **Samenvatting** een naam en beschrijving voor de groep toe.

- 5 Klik op het tabblad **Gebruikers** op **Toevoegen**.
 - a Zoek een gebruiker die u wilt toevoegen aan de groep.
 - b Wijs groepsrechten aan de gebruiker toe.
 - c Klik op **Toevoegen**.
- 6 Voeg op het tabblad **Items** vRealize Orchestrator-objecten toe aan de groep.

Opmerking U kunt ook een object toevoegen aan bestaande groepen wanneer dat object wordt gemaakt in de vRealize Orchestrator Client. Als u het object wilt toevoegen, selecteert u de groep in het vervolgkeuzemenu **Toegankelijk voor** op het tabblad **Samenvatting/Algemeen** van de objecteditor.

- 7 Klik op **Opslaan**.

vRealize Orchestrator-clientrollen in vRealize Automation configureren

U kunt servicerollen voor het vRealize Orchestrator Client toewijzen op de pagina **Identiteits- en toegangsbeheer** in vRealize Automation. Servicerollen kunnen worden toegewezen voor de ingesloten vRealize Orchestrator Client- en standalone vRealize Orchestrator-instanties die met vRealize Automation worden geverifieerd.

vRealize Orchestrator-servicerollen beheren welke functies van de ingesloten vRealize Orchestrator Client-gebruikers toegang hebben. Zie [Rollen en groepen van de vRealize Orchestrator-client beheren](#) voor meer informatie over vRealize Orchestrator-rollen.

Opmerking Standalone vRealize Orchestrator instanties die worden geverifieerd met vSphere en die een vRealize Automation-licentie gebruiken, kunnen rollen rechtstreeks in de vRealize Orchestrator Client toewijzen. Zie [Rollen toewijzen in de vRealize Orchestrator Client](#).

Voorwaarden

- Controleer of de juiste gebruikers en groepen zijn geïmporteerd uit een geldige vIDM-instantie.
- Voordat u een vRealize Orchestrator-servicerol aan uw gebruiker toewijst, controleert u of uw gebruiker een toegewezen organisatierol heeft in vRealize Automation. Zie *Gebruikers en groepen in vRealize Automation beheren* in *vRealize Automation beheren*.

Procedure

- 1 Selecteer de optie **Identiteits- en toegangsbeheer** in het vervolgkeuzemenu rechtsboven in de koptekst.
- 2 Zoek op het tabblad **Actieve gebruikers** naar het e-mailadres van de gebruiker die u aan vRealize Orchestrator wilt toewijzen.
- 3 Schakel het selectievakje naast de gebruiker in en klik op **Rollen bewerken**.

- 4 Klik op **Servicetoegang toevoegen**.
- 5 Selecteer **Orchestrator** in het vervolgkeuzemenu links.
- 6 Selecteer in het vervolgkeuzemenu rechts de rol die u aan de gebruiker wilt toewijzen.
- 7 Klik op **Opslaan**.

Versiegeschiedenis van vRealize Orchestrator-objecten

De vRealize Orchestrator Client handhaaft een versiegeschiedenisrecord voor elk vRealize Orchestrator-object. Met behulp van de versiegeschiedenis kunt u verschillende versies van vRealize Orchestrator-objecten vergelijken en terugkeren naar een vorige versie.

vRealize Orchestrator maakt een versiegeschiedenisrecord voor elk vRealize Orchestrator-object wanneer u het object opslaat. De volgende wijzigingen in vRealize Orchestrator-objecten maken een nieuw versiegeschiedenisrecord. De geschiedenisrecords voor vorige versies blijven behouden en kunnen worden gebruikt om wijzigingen in het object bij te houden en het object naar een vorige versie terug te zetten. Als u een object terugzet naar een vorige versie, wordt een nieuwe versiegeschiedenisrecord gemaakt.

De vRealize Orchestrator Client traceert de versiegeschiedenis van de volgende vRealize Orchestrator-objecten:

- Werkstromen
- Acties
- Pakketten
- Beleid
- Configuratie-elementen

U hebt toegang tot de versiegeschiedenis van een object via het tabblad **Versiegeschiedenis** van de pagina met de objecteditor. Als u probeert een object te bewerken op hetzelfde moment als een andere gebruiker, kan er een probleem met samenvoegen optreden. Om het samenvoegingsconflict op te lossen, klikt u op **Oplossen** rechts van het foutbericht. In het venster **Conflicten oplossen** hebt u drie opties:

- **Hun bestanden gebruiken.** Los het samenvoegingsconflict op door de wijzigingen te gebruiken die door de andere gebruiker zijn aangebracht.
- **Onze bestanden gebruiken.** Los het samenvoegingsconflict op door uw wijzigingen te gebruiken.
- **Oplossen.** Los het samenvoegingsconflict op door het weergegeven wijzigingsmodel te bewerken. Deze optie is niet beschikbaar als het opgegeven model ongeldig is.

Een werkstroom terugzetten naar een eerdere versie

U kunt een werkstroom terugzetten naar een eerder opgeslagen versie.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Navigeer naar **Bibliotheek > Werkstromen** en selecteer een werkstroom.
- 3 Selecteer het tabblad **Versiegeschiedenis**.
- 4 Als u een vergelijking tussen de versies wilt bekijken, selecteert u een werkstroomversie en selecteert u een andere versie in de vervolgkeuzemenu **Verschil met**.

In een venster worden de verschillen tussen de huidige werkstroomversie en de geselecteerde werkstroomversie weergegeven.

- 5 Als u de werkstroom naar een andere versie wilt terugzetten, klikt u op **Herstellen**.
De status van de werkstroom wordt teruggezet naar de status van de geselecteerde versie.

Opmerking U kunt ook een werkstroomversie herstellen vanuit de weergave van de grafische verschiltool. Zie [Visuele vergelijking tussen werkstroomversies](#).

Visuele vergelijking tussen werkstroomversies

Vergelijk wijzigingen tussen werkstroomversies met het hulpprogramma voor grafische verschillen.

De versiegeschiedenis van de vRealize Orchestrator geeft standaard verschillen weer tussen werkstroomversies in een YAML-formulier. U kunt ook een visuele vergelijking tussen verschillende werkstroomversies uitvoeren. U kunt wijzigingen bekijken in:

- De algemene werkstrooinformatie, zoals het versienummer en de werkstroombeschrijving.
- De variabelen die in de werkstroom worden gebruikt.
- De in- en uitvoerparameters van de werkstroom.
- Het werkstroomschema.

Voorwaarden

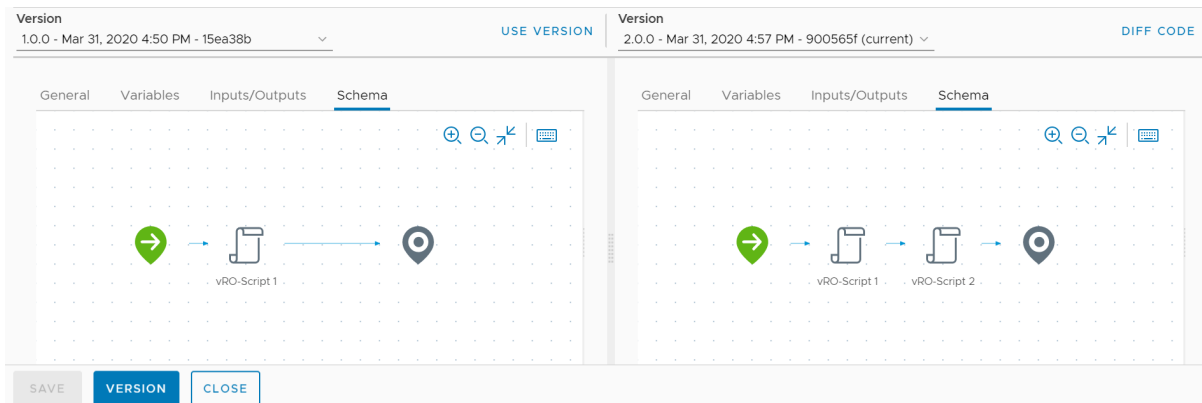
Maak een werkstroom.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Navigeer naar **Bibliotheek > Werkstromen** en selecteer een van uw werkstromen.
- 3 Bewerk de inhoud van de werkstroom.
U kunt bijvoorbeeld een extra element **Scriptbare taak** toevoegen op het tabblad **Schema**.
- 4 Klik op **Opslaan**.
- 5 Selecteer het tabblad **Versiegeschiedenis**.

6 Selecteer in de rechterbovenhoek **Visueel verschil**.

U kunt nu een visuele vergelijking uitvoeren tussen twee geselecteerde werkstroomversies. U kunt selecteren welke versies moeten worden vergeleken in het vervolgkeuzemenu **Versie**.



7 (Optioneel) U kunt een werkstroom naar een andere versie herstellen door **Versie gebruiken** te selecteren.

De inventaris van uw vRealize Orchestrator-inhoud opnieuw instellen op een eerdere status met Git

Door gebruik te maken van een oudere Git-commit, kunt u uw vRealize Orchestrator-inhoud opnieuw instellen op een eerdere status.

U kunt uw vRealize Orchestrator-inhoud opnieuw instellen op een eerdere status door een specifieke commit te selecteren.

Voorwaarden

- Configureer een verbinding met een GitHub- of GitLab-opslagplaats. Zie [Een verbinding met een Git-opslagplaats configureren](#).
- Push een lokale wijzigingsset naar de geconfigureerde Git-opslagplaats.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Navigeer naar **Beheer > Git-geschiedenis**.
- 3 Selecteer een wijzigingsset die u opnieuw wilt instellen en klik op **Opnieuw instellen**.
- 4 Bevestig dat u opnieuw wilt instellen op deze specifieke commit en klik op **OK**.

De inventaris van de vRealize Orchestrator-inhoud wordt opnieuw ingesteld op de status die is opgegeven in de commit. Relevante vRealize Orchestrator-inhoud wordt teruggezet naar een vorige versie. Als de inhoud niet bestaat wanneer de commit is gepusht, wordt deze uit de inventaris verwijderd.

Wat nu te doen

Als u de vRealize Orchestrator-inventaris wilt terugzetten naar de laatste status die in de Git-opslagplaats is opgeslagen, voert u een Pull-opdracht uit in het venster **Git-geschiedenis**.

vRealize Orchestrator-toepassingsvoorbeelden

4

Deze toepassingsvoorbeelden demonstreren een deel van de functionaliteit van het vRealize Orchestrator-platform.

In deze toepassingsvoorbeelden worden alleen de voorbeeldwaarden weergegeven. Uw eigen omgevingsstructuur en naamgevingsconventies kunnen variëren.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Amazon Web Services in vRealize Orchestrator integreren met behulp van Python](#)
- [Git-vertakkingen gebruiken om mijn vRealize Orchestrator-objectinventaris te beheren](#)
- [Modules van derden gebruiken om de vRealize Automation-project-API aan te roepen](#)

Amazon Web Services in vRealize Orchestrator integreren met behulp van Python

Dit vRealize Orchestrator-gebruiksscenario toont een voorbeeld van hoe u Python kunt gebruiken om de mogelijkheden van uw vRealize Orchestrator-implementatie uit te breiden.

Vanaf vRealize Orchestrator 8.1 kunt u drie nieuwe runtimes gebruiken voor gebruik in acties en werkstroomscripts.

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

Opmerking De PowerCLI-runtime bevat PowerShell en de volgende modules: VMware.PowerCLI, PowerNSX, PowervRA.

Belangrijk U kunt de nieuwe runtime alleen gebruiken als uw vRealize Orchestrator-implementatie een vRealize Automation-licentie gebruikt.

Dit gebruiksscenario laat zien hoe u een Python-script kunt maken waarmee EC2-instanties in Amazon Web Services (AWS) worden aangeroepen.

Belangrijk Voordat u begint met het ontwikkelen van uw aangepaste script, controleert u of u bekend bent met de basisconcepten voor het gebruik van Python-, Node.js- en PowerShell-scripts in vRealize Orchestrator. Zie [Basisconcepten voor Python-, Node.js- en PowerShell-scripts](#).

Procedure

1 Eerste Python-script maken

Maak op uw lokale machine uw Python-script en verpak het script en een boto3-bibliotheek als een ZIP-map.

2 De Amazon Web Services-actie maken

Maak een vRealize Orchestrator-actie die gebruikmaakt van uw Python-script.

3 Foutopsporing voor de actie Amazon Web Services uitvoeren

De oorspronkelijke versie van het Python-script heeft een opzettelijke ingebouwde fout, zodat u kunt leren hoe u fouten in uw script oplost.

4 De Amazon Web Services-actie bijwerken

Importeer het bijgewerkte Python-script en voer de actie opnieuw uit.

Eerste Python-script maken

Maak op uw lokale machine uw Python-script en verpak het script en een boto3-bibliotheek als een ZIP-map.

Voorwaarden

- Download en installeer Python 3. Zie de [pagina met Python-downloads](#).
- Download en installeer Visual Studio Code. Zie de [downloadpagina voor Visual Studio Code](#).
- Controleer of u de Python-extensie voor Visual Studio Code heeft geïnstalleerd. Zie de [Visual Studio Marketplace](#).

Procedure

- 1 Maak op uw lokale machine een map vro-python-aws en installeer de boto3 Python SDK op deze map.

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

- 2 Open een editor en maak het Python-hoofdsript. Voor dit gebruiksscenario gebruikt u 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)
```

Dit Python-script geeft alle actieve EC2-instanties in een bepaalde regio weer.

- 3 Sla het gemaakte script op als een `main.py`-bestand in de map `vro-python-aws`.
- 4 Meld u aan bij een opdrachtregelinterface.
- 5 Ga naar de map `vro-python-aws`.

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

- 6 Maak een ZIP-pakket dat het python-script bevat.

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

Opmerking U kunt ook het ZIP-pakket maken met behulp van een ZIP-hulpprogramma, zoals 7-Zip.

Resultaten

U heeft het Python-basiscript gemaakt en dit voorbereid voor import in uw vRealize Orchestrator-implementatie.

De Amazon Web Services-actie maken

Maak een vRealize Orchestrator-actie die gebruikmaakt van uw Python-script.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Bibliotheek > Acties**.
- 3 Klik op **Nieuwe actie**.
- 4 Voer op het tabblad **Algemeen** een naam, module en versienummer in voor de actie.
- 5 Selecteer op het tabblad **Script** de optie **Python 3.7** als de runtime en **Zip** als het scripttype.
- 6 Klik op **Importeren**.

- 7 Blader naar de map `vro-python-aws` en selecteer het ZIP-pakket met het Python-script.
- 8 Voer in het tekstvak **Vermeldingshandler** `main.handler` in.

Opmerking De vermeldingshandler van de actie is gebaseerd op het hoofdsript in het geïmporteerde ZIP-pakket. Omdat het hoofdsript zich in een bestand bevindt met de naam `main.py` en een functie met de naam **handler**, moet de vermeldingshandler **main.handler** zijn. Als u een andere naam heeft gegeven aan het hoofdsriptbestand, wijzigt u de waarde van de vermeldingshandler dienovereenkomstig.

- 9 Sla de actie op en klik op **Uitvoeren**.

Er treedt een fout op tijdens het uitvoeren van de actie.

- 10 Selecteer het tabblad **Logboeken**.

De logboeken van de actie-uitvoering geven een foutbericht `"botocore.exceptions.NoRegionError: U moet een regio opgeven."` weer. Dit is verwacht gedrag, omdat het eerste Python-script geen regio definieert.

Wat nu te doen

Voer foutopsporing uit op het Python-script. Zie [Foutopsporing voor de actie Amazon Web Services uitvoeren](#).

Foutopsporing voor de actie Amazon Web Services uitvoeren

De oorspronkelijke versie van het Python-script heeft een opzettelijke ingebouwde fout, zodat u kunt leren hoe u fouten in uw script oplost.

Voorwaarden

Meld u aan bij uw AWS-account (Amazon Web Services) en maak een IAM-gebruiker specifiek voor dit gebruiksscenario. Zie [Een IAM-gebruiker maken in uw AWS-account](#). De IAM-gebruiker moet de volgende rechten hebben:

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

```

Procedure

1 Bereid de vRealize Orchestrator Appliance voor.

Voorzichtig Voer geen foutopsporing uit in scripts in uw vRealize Orchestrator-productie-implementatie. Voer foutopsporing uit vanaf een vRealize Orchestrator-implementatie met één knooppunt die u gebruikt voor ontwikkeling en tests.

- a Meld u via SSH aan bij de vRealize Orchestrator Appliance-opdrachtregel als **root**.
- b Voer de opdracht `vracli dev tools` uit.
- c U wordt gevraagd om te bevestigen dat u wilt doorgaan. Voer **Ja** in om door te gaan of **Nee** om te annuleren.

Belangrijk Door de opdracht `vracli dev tools` uit te voeren, opent u de poorten die nodig zijn om fouten op te sporen in het Python-script. U moet de huidige SSH-sessie geopend laten tijdens de foutopsporing.

2 Start de configuratie van de foutopsporing.

- a Meld u aan bij de vRealize Orchestrator-client.
- b Open de actie AWS en klik op **Foutopsporing**.
De foutopsporing begint en de actie-uitvoering is opgeschort.
- c Selecteer het tabblad **Configuratie van foutopsporing**.
Het tabblad bevat een `.json`-configuratie die u op afstand kunt toevoegen aan uw IDE om fouten op te sporen in het Python-script.
- d Kopieer de configuratie-inhoud handmatig of klik op **Naar klembord kopiëren**.

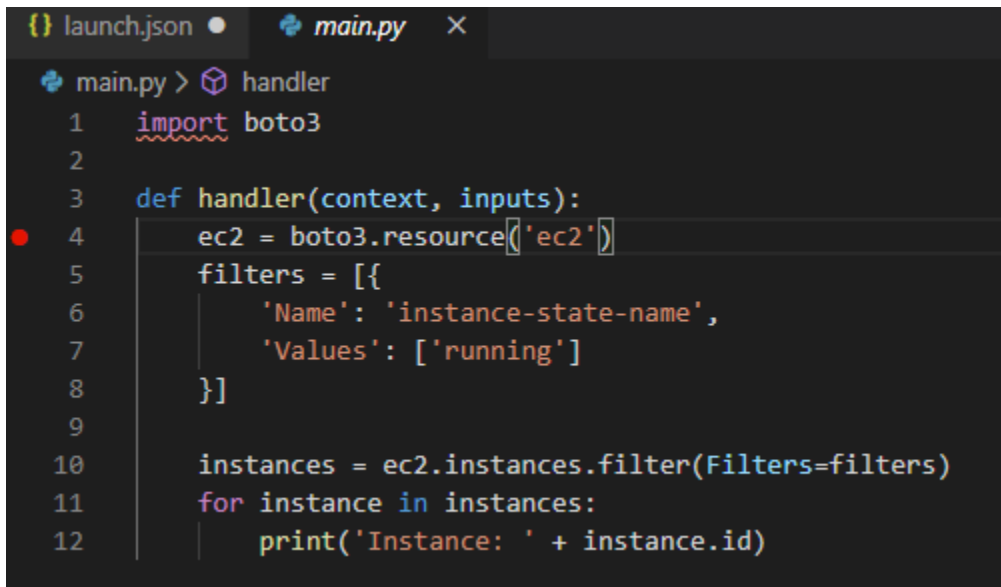
3 Voer foutopsporing uit op het Python-script.

- a Open Visual Studio Code.
- b Open de map `vro-python-aws`.
- c Selecteer in het bovenste navigatiedeelvenster **Uitvoeren > Configuraties openen**.
- d Selecteer **Python-bestand**.

- e Laat de kenmerken "version" en "configuration" op hun huidige posities staan en plak de inhoud van de .json-configuratie die is gekopieerd van de vRealize Orchestrator-client. Het gegenereerde launch.json-bestand moet er ongeveer als volgt uitzien:

```
{
  "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 Selecteer het main.py-scriptbestand en voeg een onderbrekingspunt toe aan de `ec2 = boto3.resource('ec2')`-regel.



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

- g Selecteer in het bovenste navigatiedeelvenster **Uitvoeren > Foutopsporing starten**.

- h Wanneer het foutopsporingsprogramma het onderbrekingspunt bereikt, voert u een Stap over-bewerking uit.

De foutopsporing geeft aan dat het Python-script geen specifieke regio en AWS-toegangssleutel bevat.

- i Ga terug naar de open vRealize Orchestrator Appliance-sessie en druk op **Enter** om de poorten te sluiten die u voor deze foutopsporingssessie heeft geopend.

4 Voeg de ontbrekende informatie toe aan het Python-script.

- a Maak in Visual Studio Code een bestand met de naam `awsconfig` dat de AWS-toegangssleutel van de IAM-gebruiker bevat en de AWS-regio die u wilt pinggen met het Python-script.

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

- b Sla `awsconfig` op als een configuratiebestand (`.cfg`) in de map `vro-python-aws`.
- c Open het bestand `main.py` en wijzig dit zodat de boto3-bibliotheek het bestand `awsconfig.cfg` kan gebruiken.

```
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 Maak een nieuw ZIP-pakket dat het bestand `main.py`, het bestand `awsconfig.cfg` en de boto3-bibliotheek bevat.

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

Opmerking U kunt ook het ZIP-pakket maken met behulp van een ZIP-hulpprogramma, zoals 7-Zip.

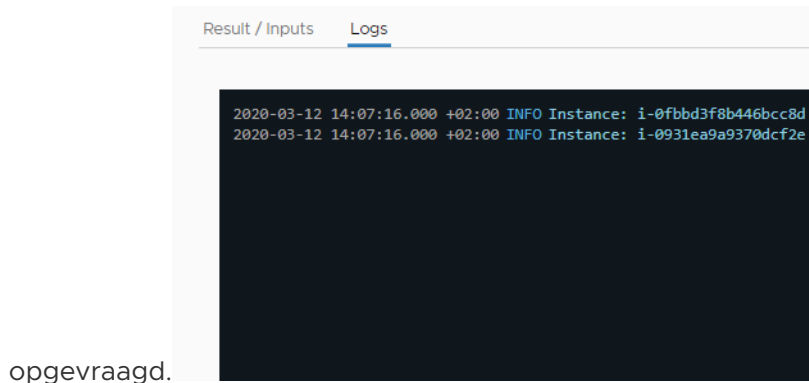
De Amazon Web Services-actie bijwerken

Importeer het bijgewerkte Python-script en voer de actie opnieuw uit.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Bibliotheek > Acties** en selecteer de oorspronkelijke Amazon Web Services-actie (AWS).
- 3 (Optioneel) Wijzig het versienummer op het tabblad **Algemeen**.
- 4 Verwijder het oude ZIP-pakket en klik op **Importeren**.
- 5 Selecteer het bijgewerkte ZIP-pakket.
- 6 Sla de actie op en klik op **Uitvoeren**.
- 7 Nadat de actie is uitgevoerd, selecteert u het tabblad **Logboeken**.

In de logboeken worden de EC2-instanties weergegeven die door de actie worden



Wat nu te doen

Maak een vRealize Orchestrator-werkstroom die de bijgewerkte AWS-actie gebruikt als **Actie-element**.

Git-vertakkingen gebruiken om mijn vRealize Orchestrator-objectinventaris te beheren

Gebruik vertakkingen om te beheren hoe u vRealize Orchestrator-inhoud in uw Git-opslagplaats beheert.

Door gebruik te maken van Git kunt u de flexibiliteit voor uw vRealize Orchestrator-ontwikkelaars verhogen door een centrale opslagplaats te bieden. U kunt Git bijvoorbeeld gebruiken om de werkstroomontwikkeling te beheren in meerdere vRealize Orchestrator-omgevingen.

Opmerking Om Git te gebruiken om uw objectinventaris te beheren, moet uw vRealize Orchestrator-implementatie een vRealize Automation-licentie gebruiken. Zie *vRealize Orchestrator-functies inschakelen met licenties* in *vRealize Orchestrator installeren en configureren* voor meer informatie.

Vanaf vRealize Orchestrator 8.1 kunt u nu objecten naar en van vertakkingen pushen en ophalen. U kunt vertakkingen gebruiken om de ontwikkeling van specifieke groepen van vRealize Orchestrator-objecten te beheren voordat ze worden samengevoegd in uw hoofdvertakking.

In dit geval gebruikt u een GitLab-project om vRealize Orchestrator-objecten te beheren die gebruikmaken van de Python-runtime. Dit gebruiksscenario vertegenwoordigt een voorbeeld van de Git-functionaliteit in vRealize Orchestrator en vertegenwoordigt niet de beperkingen van het functiebereik.

Opmerking Als u bekend bent met GitHub, kunt u ook een GitHub-opslagplaats gebruiken voor dit gebruiksscenario.

Procedure

1 Uw GitLab-omgeving voorbereiden

Maak een Git-vertakking voor uw vRealize Orchestrator Python-objecten.

2 Een verbinding met een Git-opslagplaats configureren

Als **beheerder** kunt u een verbinding tussen uw vRealize Orchestrator-implementatie en een Git-opslagplaats of -project configureren.

3 Wijzigingen naar een Git-opslagplaats pushen

Push uw wijzigingen in lokale vRealize Orchestrator-objecten naar uw geïntegreerde Git-opslagplaats. Voor dit toepassingsvoorbeeld worden wijzigingen in een Python-gebaseerde vRealize Orchestrator-actie naar een specifieke Git-tak gepusht.

Uw GitLab-omgeving voorbereiden

Maak een Git-vertakking voor uw vRealize Orchestrator Python-objecten.

Voorwaarden

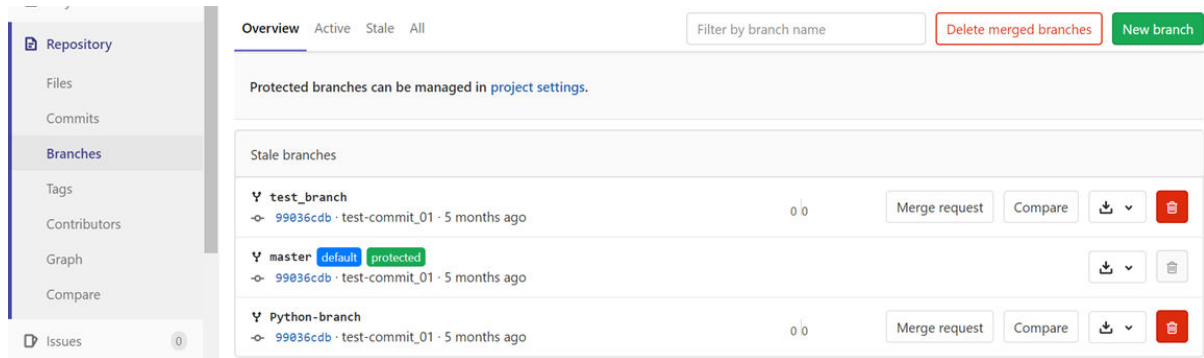
Maak een GitLab-project voor uw vRealize Orchestrator-omgeving. Zie [Een project maken](#).

Procedure

- 1 Meld u aan bij uw GitLab-account.
- 2 Ga naar uw GitLab-project.
- 3 Selecteer in het linkernavigatievenster de optie **Opslagplaats > Vertakkingen**.

- 4 Klik op het tabblad **Overzicht** op **Nieuwe vertakking**.
- 5 Voer onder **Vertakkingsnaam** **Python-vertakking** in.
- 6 Laat de optie **Maken van als hoofd**.
- 7 Klik op **Vertakking maken**.

U heeft een vertakking gemaakt voor uw op Python gebaseerde vRealize Orchestrator-objekten.



Een verbinding met een Git-opslagplaats configureren

Als **beheerder** kunt u een verbinding tussen uw vRealize Orchestrator-implementatie en een Git-opslagplaats of -project configureren.

Als u Git wilt gebruiken voor het beheer van uw vRealize Orchestrator-objectinventaris, moet u een verbinding met uw Git-opslagplaats configureren met behulp van de vRealize Orchestrator Client.

Voorwaarden

- Controleer of uw vRealize Orchestrator-omgeving een vRealize Automation-licentie gebruikt.
- Genereer een toegangstoken voor uw GitLab-project en kopieer het naar uw klembord voor gebruik tijdens het configuratieproces. Zie [Een persoonlijk toegangstoken maken](#).

Opmerking Voor dit gebruiksscenario gebruikt u een GitLab-project. Als u bekend bent met GitHub, kunt u ook een GitHub-opslagplaats gebruiken. Zie [Een persoonlijk toegangstoken maken voor de opdrachtregel](#) voor informatie over het genereren van een GitHub-token.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client als een **beheerder**.
- 2 Ga naar **Beheer > Git-opslagplaatsen**.
- 3 Klik op **Opslagplaats toevoegen**.

- 4 Voer het URL-adres van uw Git-opslagplaats in.

Bijvoorbeeld <https://gitlab.com/myusername/my-vro-repo>.

Opmerking U kunt ook een verbinding maken met het SSH-protocol.

- 5 Voer de gebruikersnaam in voor uw Git-profiel.
- 6 Voer het toegangstoken van uw Git-opslagplaats in.
- 7 Als u de verbinding met de Git-opslagplaats wilt valideren, klikt u op **Valideren**.
- 8 (Optioneel) Wijzig de naam die wordt gebruikt om de opslagplaats te identificeren in de vRealize Orchestrator Client.
- 9 (Optioneel) Voeg een korte beschrijving toe voor de gekoppelde Git-opslagplaats.
- 10 Als u de gekoppelde Git-opslagplaats wilt activeren, klikt u op **Actieve opslagplaats maken**.

Opmerking Er kan slechts één Git-opslagplaats actief zijn. U kunt de actieve Git-opslagplaats wijzigen op de pagina **Git-opslagplaatsen**.

- 11 Selecteer de tak waarnaar u uw wijzigingen wilt pushen. Voor dit gebruiksscenario gebruikt u **Python-tak**. Zie [Uw GitLab-omgeving voorbereiden](#).

Opmerking U kunt de geselecteerde Git-tak op elk gewenst moment wijzigen nadat u de eerste Git-configuratie heeft voltooid.

- 12 Klik op **Opslaan** om het configuratieproces te voltooien.

Wat nu te doen

Ga terug naar het menu **Git-opslagplaatsen** en controleer of de status van de opslagplaats **Actief** is.

Wijzigingen naar een Git-opslagplaats pushen

Push uw wijzigingen in lokale vRealize Orchestrator-objecten naar uw geïntegreerde Git-opslagplaats. Voor dit toepassingsvoorbeeld worden wijzigingen in een Python-gebaseerde vRealize Orchestrator-actie naar een specifieke Git-tak gepusht.

U kunt een lokale wijzigingsset naar een Git-opslagplaats pushen. Elke wijzigingsset kan bestaan uit een of meer aangepaste vRealize Orchestrator-objecten.

Opmerking Het proces van het pushen en annuleren van wijzigingen in een Git-opslagplaats wordt niet beperkt door groepsrechten. Daarom kan een werkstroomontwikkelaar van de ene groep lokale wijzigingen die door een andere ontwikkelaar zijn aangebracht, pushen of annuleren.

Voorwaarden

- Controleer of u een Git-tak hebt gemaakt. Zie [Uw GitLab-omgeving voorbereiden](#).

- Controleer of u een verbinding met een Git-opslagplaats hebt geconfigureerd. Zie [Een verbinding met een Git-opslagplaats configureren](#).
- Controleer of uw Git-integratie is ingesteld om wijzigingen naar de **Python-gebaseerde** Git-tak te pushen.
- Maak een Python-gebaseerd vRealize Orchestrator-object. Zie [Amazon Web Services in vRealize Orchestrator integreren met behulp van Python](#) voor een voorbeeld.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Bewerk uw Python-actie.
 - a Navigeer naar **Bibliotheek > Acties** en selecteer uw Python-actie.
 - b Maak kleine wijzigingen in de actie, zoals het wijzigen van de beschrijving.
 - c Sla de actie op.

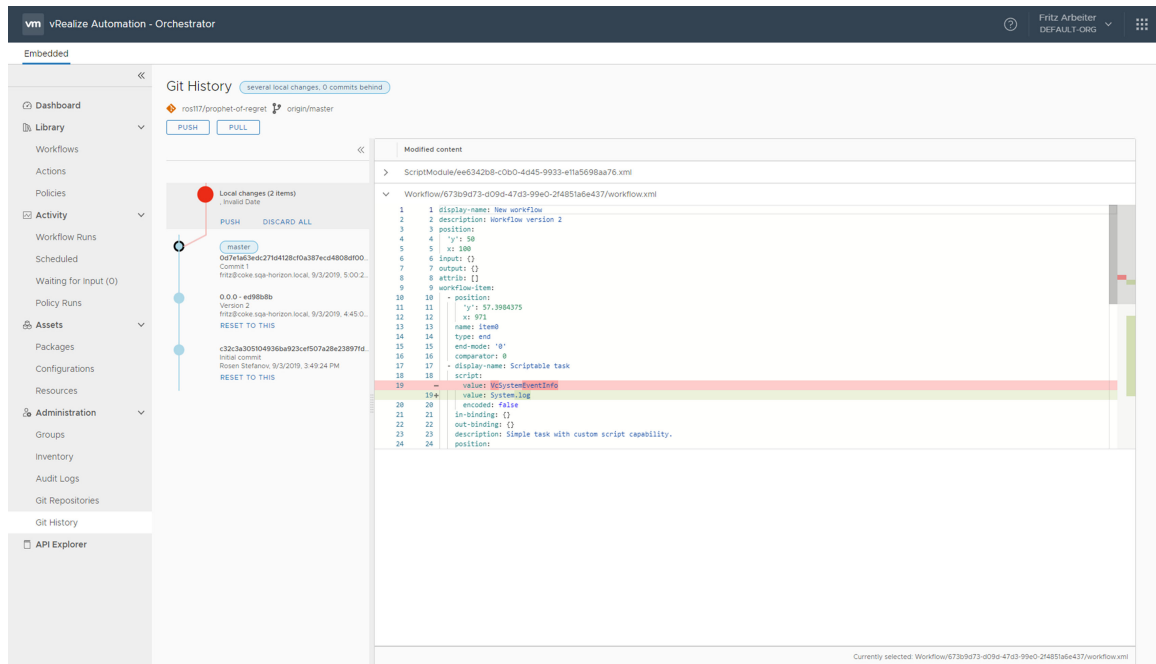
3 Push uw wijzigingen naar de Git-opslagplaats.

Opmerking U kunt lokale wijzigingen ook op objectniveau pushen door te klikken op de optie **Versie** die onderaan de objecteditor wordt weergegeven.

a Navigeer naar **Beheer > Git-geschiedenis**.

De **Git-geschiedenis** toont de huidige verschillen tussen de lokale versietak en de geselecteerde Git-opslagplaatstak. U kunt de vermelding voor een aangepast vRealize Orchestrator-object uitvouwen om de versieverschillen te bekijken.

Opmerking U kunt een lokale wijzigingsset annuleren door te klikken op **Alles annuleren**.



b Klik op **Pushen**.

c Voer een committitel in.

d (Optioneel) Voer een korte beschrijving in voor de commit.

e Selecteer de wijzigingen in uw Python-actie die u naar de Git-opslagplaats wilt pushen.

4 Klik op **Pushen** om het pushen van de lokale wijzigingsset naar de Git-opslagplaats te voltooien.

Wat nu te doen

Nadat u de ontwikkeling in uw Git-tak hebt voltooid, kunt u deze samenvoegen met de hoofdstructuur. Zie [Een samenvoegingsaanvraag maken](#).

Modules van derden gebruiken om de vRealize Automation-project-API aan te roepen

Dit vRealize Orchestrator toepassingsvoorbeeld toont hoe u de vRealize Automation-project-API kunt oproepen met behulp van modules van derden.

Vanaf vRealize Orchestrator 8.1 kunt u de volgende runtimes gebruiken:

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

Opmerking De PowerCLI-runtime bevat PowerShell en de volgende modules: VMware.PowerCLI, PowerNSX, PowervRA.

In dit toepassingsvoorbeeld leert u hoe u vRealize Orchestrator-acties maakt die gebruikmaken van afhankelijkheidsmodules van derden om verbinding te maken met de vRealize Automation-project-API.

Belangrijk Voordat u begint met het ontwikkelen van uw aangepaste script, controleert u of u bekend bent met de basisconcepten voor het gebruik van Python-, Node.js- en PowerShell-scripts in vRealize Orchestrator. Zie [Basisconcepten voor Python-, Node.js- en PowerShell-scripts](#).

Een Python-script maken dat de vRealize Automation Project-API aanroept

Maak een voorbeeldscript dat Python gebruikt om de vRealize Automation Project-API aan te roepen.

Voorwaarden

Controleer of u Python 3 en het installatieprogramma van het PIP-pakket heeft geïnstalleerd. Zie de pagina [Python-downloads](#) en [Python-pakketindex](#).

Procedure

- 1 Open een opdrachtregelshell op uw lokale machine.
- 2 Maak een map vro-python-vra.

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

- 3 Ga naar de map vro-python-vra.

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

4 Maak een Python-script met de naam `handler.py`.

```
touch handler.py
```

Het `handler.py`-script moet één functie definiëren die twee argumenten accepteert: de context van de vRealize Orchestrator-werkstroomuitvoering en de afhankelijke vRealize Orchestrator-invoer.

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

Opmerking Met behulp van standaardbibliotheken voor logboekregistratie wordt alles wat u vastlegt in de actie die uw script gebruikt, ook weergegeven in het werkstroomlogboek. De invoer en de retour van uw script moeten overeenkomende invoerparameters en retourtypen hebben die zijn geconfigureerd in de vRealize Orchestrator-client. De `vRAUrl`-invoer in uw script moet bijvoorbeeld een bijbehorende invoerparameter met de naam `vRAUrl` in de vRealize Orchestrator-client hebben. Op dezelfde wijze moet, als uw script een tekenreekswaarde retourneert, het retourtype dat is geconfigureerd in de vRealize Orchestrator-client ook een tekenreeks type zijn. Als uw actie een complex object retourneert, kunt u retourtype `Properties Of Composite Type` gebruiken.

5 Installeer de module voor Python-aanvragen.

Belangrijk De modules voor afhankelijkheden van derden moeten worden geïnstalleerd in een map op het hoofdniveau in uw primaire `vro-python-vra-scriptmap`. Voor dit gebruiksscenario maakt u een `lib`-map voor uw aanvragenmodule.

- a Maak een `lib`-map.

```
mkdir lib
```

- b Installeer de aanvragenmodule.

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

6 Voeg de aanvragenmodule toe aan het `handler.py`-script.

```
import requests

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

7 Maak een GET-aanvraag voor de vRealize Automation Project-API.

```
token = ''
vRAUrl = ''
```

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

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

8 Definieer de waarden token en vRAUrl .

- Haal het toegangstoken op met behulp van de API van de vRealize Automation-identiteitsservice. Zie [Uw toegangstoken voor de vRealize Automation API ophalen](#)
- Voor de waarde vRAUrl definieert u het script zodat het een vRealize Orchestrator-invoerparameter met dezelfde naam gebruikt.

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

- Voeg de nieuwe waarden toe aan het handler.py-bestand.

```
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()
```

Opmerking Omdat het antwoord van de vRealize Automation Project-API wordt geretourneerd in een JSON-indeling, gebruikt u een retourtype Properties of Composite Type voor uw vRealize Orchestrator-actie.

- Maak een ZIP-pakket dat het handler.py-bestand en de map lib van uw aanvragenmodule bevat.

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

Wat nu te doen

Importeer het PowerShell-script in een vRealize Orchestrator-actie. Zie [Acties in de vRealize Orchestrator-client maken](#).

Een Node.js-script maken dat de vRealize Automation Project-API aanroept

Maak een voorbeeldscript dat gebruikmaakt van Node.js om de vRealize Automation Project-API aan te roepen.

Voorwaarden

Download en installeer Node.js 12. Zie [Downloads voor Node.js](#).

Procedure

- 1 Open een opdrachtregelshell op uw lokale machine.

- 2 Maak een map vro-node-vra.

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

- 3 Ga naar de map vro-node-vra.

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

- 4 Maak een Node.js-script met de naam handler.js.

```
touch handler.js
```

Het handler.js-script moet één functie definiëren die twee argumenten accepteert: de context van de vRealize Orchestrator-werkstroomuitvoering en de afhankelijke vRealize Orchestrator-invoer.

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

Opmerking Met behulp van standaardbibliotheken voor logboekregistratie wordt alles wat u vastlegt in de actie die uw script gebruikt, ook weergegeven in het werkstroomlogboek. De invoer en de retour van uw script moeten overeenkomende invoerparameters en retourtypen hebben die zijn geconfigureerd in de vRealize Orchestrator-client. De vRAURL-invoer in uw script moet bijvoorbeeld een bijbehorende invoerparameter met de naam vRAURL in de vRealize Orchestrator-client hebben. Op dezelfde wijze moet, als uw script een tekenreekswaarde retourneert, het retourtype dat is geconfigureerd in de vRealize Orchestrator-client ook een tekenreeks type zijn. Als uw actie een complex object retourneert, kunt u retourtype Properties of Composite Type gebruiken.

- 5 Installeer de module voor Node.js-aanvragen.

```
npm install request
```

Belangrijk De modules voor afhankelijkheden van derden moeten worden geïnstalleerd in de map node_modules op het hoofdniveau in uw primaire vro-node-vra-scriptmap. Verplaats deze map niet en wijzig de naam ervan niet.

6 Voeg de aanvragenmodule toe aan het handler.js-script.

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

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

7 Maak een GET-aanvraag voor de vRealize Automation Project-API.

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

8 Definieer de waarden token en vRAUrl .

- Haal het toegangstoken op met behulp van de API van de vRealize Automation-identiteitsservice. Zie [Uw toegangstoken voor de vRealize Automation API ophalen](#).
- Voor de waarde vRAUrl definieert u het script zodat het een vRealize Orchestrator-invoerparameter met dezelfde naam gebruikt.

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

- Voeg de nieuwe waarden toe aan het handler.js-bestand.

```
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));
  });
}
```

Opmerking Omdat het antwoord van de vRealize Automation Project-API wordt geretourneerd in een JSON-indeling, gebruikt u een retourtype Properties of Composite Type voor uw vRealize Orchestrator-actie.

9 Maak een ZIP-pakket dat het handler.js-bestand en de map node_modules van uw aanvragenmodule bevat.

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

Wat nu te doen

Importeer het Node.js-script in een vRealize Orchestrator-actie. Zie [Acties in de vRealize Orchestrator-client maken](#).

Een PowerShell-script maken dat de vRealize Automation Project-API aanroept

Maak een voorbeeldscript dat PowerShell gebruikt om de vRealize Automation Project-API aan te roepen.

Procedure

- 1 Open een opdrachtregelshell op uw lokale machine.
- 2 Maak een map vro-powershell-vra.

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

- 3 Ga naar de map vro-powershell-vra.

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

- 4 Maak een PowerShell-script met de naam handler.ps1.

```
touch handler.ps1
```

Het handler.ps1-script moet één functie definiëren die twee argumenten accepteert: de context van de vRealize Orchestrator-werkstroomuitvoering en de afhankelijke vRealize Orchestrator-invoer.

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

Opmerking Met behulp van standaardbibliotheken voor logboekregistratie wordt alles wat u vastlegt in de actie die uw script gebruikt, ook weergegeven in het werkstroomlogboek. De invoer en de retour van uw script moeten overeenkomende invoerparameters en retourtypen hebben die zijn geconfigureerd in de vRealize Orchestrator-client. De vRAUrl-invoer in uw script moet bijvoorbeeld een bijbehorende invoerparameter met de naam vRAUrl in de vRealize Orchestrator-client hebben. Op dezelfde wijze moet, als uw script een tekenreekswaarde retourneert, het retourtype dat is geconfigureerd in de vRealize Orchestrator-client ook een tekenreeks type zijn. Als uw actie een complex object retourneert, kunt u retourtype Properties of Composite Type gebruiken.

5 Installeer de PowerShell-bevestigingsmodule.

Belangrijk De modules voor afhankelijkheden van derden moeten worden geïnstalleerd in een map op het hoofdniveau in uw primaire vro-powershell-vra-scriptmap. Voor dit gebruiksscenario maakt u een map Modules voor uw bevestigingsmodule.

- a Maak een map Modules.

```
mkdir Modules
```

- b Installeer de bevestigingsmodule.

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

6 Voeg de bevestigingsmodule toe aan het handler.ps1-script.

```
Import-Module Assert

function Handler {
    Param($context, $inputs)

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

7 Maak een GET-aanvraag voor de vRealize Automation Project-API die de Invoke-RestMethod-cmdlet gebruikt.

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

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

8 Definieer de waarden token en vrUrl .

- a Haal het toegangstoken op met behulp van de API van de vRealize Automation-identiteitsservice. Zie [Uw toegangstoken voor de vRealize Automation API ophalen](#).
- b Voeg de kenmerken Assert-NotNull en Assert-Type van de bevestigingsmodule toe.

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

- c Voor de waarde `vRAUrl` definieert u het script zodat het een vRealize Orchestrator-invoerparameter met dezelfde naam gebruikt.

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

- d Voeg de nieuwe waarden toe aan het `handler.ps1`-bestand.

```
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
}
```

Opmerking Omdat het antwoord van de vRealize Automation Project-API wordt geretourneerd in een JSON-indeling, gebruikt u een retourtype `Properties of Composite Type` voor uw vRealize Orchestrator-actie.

- 9 Maak een ZIP-pakket dat het `handler.ps1`-bestand en de map `Modules` van uw bevestigingsmodule bevat.

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

Wat nu te doen

Importeer het PowerShell-script in een vRealize Orchestrator-actie. Zie [Acties in de vRealize Orchestrator-client maken](#).

Werkstromen beheren

5

Een werkstroom bestaat uit een reeks acties en beslissingen die u sequentieel uitvoert. vRealize Orchestrator biedt een bibliotheek met werkstromen die algemene beheertaken uitvoeren. vRealize Orchestrator biedt ook bibliotheken met de afzonderlijke acties die de werkstromen uitvoeren.

Werkstromen combineren acties, beslissingen en resultaten die, wanneer ze in een bepaalde volgorde worden uitgevoerd, een specifieke taak of een specifiek proces voltooien in een virtuele omgeving. Werkstromen voeren taken uit zoals het inrichten van virtual machines, het maken van back-ups, het uitvoeren van regelmatig onderhoud, het verzenden van e-mails, het uitvoeren van SSH-bewerkingen, het beheren van de fysieke infrastructuur en andere algemene bewerkingen van hulpprogramma's. Afhankelijk van hun functie accepteren werkstromen de invoer van parameters. U kunt werkstromen maken die worden uitgevoerd volgens gedefinieerde schema's of die worden uitgevoerd als bepaalde verwachte gebeurtenissen optreden. Informatie kan door u worden verstrekt, door andere gebruikers, door een andere werkstroom of actie, of door een extern proces, zoals de aanroep van een webservice door een toepassing. Werkstromen valideren en filteren informatie voordat ze worden uitgevoerd.

Werkstromen kunnen andere werkstromen aanroepen. U kunt bijvoorbeeld een werkstroom hebben die een andere werkstroom aanroept om een nieuwe virtual machine te maken.

U kunt werkstromen maken met de Integrated Development Environment (IDE) van de vRealize Orchestrator Client-interface, die toegang biedt tot de werkstroombibliotheek en de mogelijkheid om werkstromen uit te voeren op de werkstroomengine. De werkstroomengine kan ook objecten uit externe bibliotheken gebruiken die u aan vRealize Orchestrator koppelt. Met deze functie kunt u processen aanpassen of functies implementeren die door toepassingen van derden worden aangeboden.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Standaardwerkstromen in de vRealize Orchestrator-werkstroombibliotheek](#)
- [Werkstromen maken in de vRealize Orchestrator-client](#)
- [Werkstromen en acties vanuit de bovenliggende werkstroom bewerken](#)
- [Ontwerpfunctie voor vRealize Orchestrator-invoerformulier](#)
- [Aanvragen voor gebruikersinteractie in de vRealize Orchestrator-client](#)

■ [Werkstromen plannen op de vRealize Orchestrator-client](#)

Standaardwerkstromen in de vRealize Orchestrator-werkstroombibliotheek

vRealize Orchestrator biedt een standaardbibliotheek met werkstromen die u kunt gebruiken om bewerkingen in uw virtuele infrastructuur te automatiseren. De werkstromen in de standaardbibliotheek worden vergrendeld met de alleen-lezen status. Als u een standaardwerkstroom wilt aanpassen, moet u die werkstroom dupliceren. Dubbele werkstromen of aangepaste werkstromen die u maakt, kunnen volledig worden bewerkt.

De inhoud van de werkstroombibliotheek is toegankelijk via het menu **Bibliotheek > Werkstromen** van de HTML5-gebaseerde vRealize Orchestrator Client. Zowel standaard- als aangepaste werkstromen in de client worden geordend met behulp van tags. U krijgt bijvoorbeeld toegang tot de werkstroom **Sleutelpaar genereren** door **SSH** in te voeren in het zoekvak van de werkstroombibliotheek.

Opmerking U kunt geen nieuwe tags toevoegen aan standaardwerkstromen, tenzij u de werkstroom dupliceert.

Werkstromen maken in de vRealize Orchestrator-client

U kunt de vRealize Orchestrator Client gebruiken om werkstromen te maken en te bewerken.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Selecteer **Bibliotheek > Werkstromen**.
- 3 Klik op **Nieuwe werkstroom**.
- 4 Voer de naam van de nieuwe werkstroom in en klik op **Maken**.
- 5 Gebruik de werkstroomeditor om de variabelen, de werkstroominvoer en -uitvoer, de schemastructuur en de presentatie van de werkstroom te configureren.
- 6 Klik op **Opslaan** om het bewerken van de werkstroom te voltooien.

Opmerking U kunt wijzigingen in werkstromen bijhouden op het tabblad **Versiegeschiedenis**. Zie [Versiegeschiedenis van vRealize Orchestrator-objecten](#) voor meer informatie.

Wat nu te doen

U kunt de functie voor vRealize Orchestrator-tokenherhaling gebruiken om de prestaties van uw werkstromen te optimaliseren. Zie [Herhaling van werkstroomtoken in de vRealize Orchestrator-client gebruiken](#) voor meer informatie.

Werkstromen en acties vanuit de bovenliggende werkstroom bewerken

Bewerk werkstromen en acties direct vanuit de bovenliggende werkstroom in de vRealize Orchestrator Client.

Het bewerken van onderliggende werkstromen en acties direct vanuit de bovenliggende werkstroom kan de werkstroomontwikkeling helpen stroomlijnen.

Voorwaarden

Maak een werkstroom die een andere werkstroom, actie of beide aanroept.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Bibliotheek > Werkstromen** en selecteer uw werkstroom.
- 3 Selecteer het tabblad **Schema**.
- 4 Afhankelijk van het objecttype dubbelklikt u op het **Werkstroomelement** of **Actie-element** in het werkstroomcanvas.
- 5 Bewerk het object.
- 6 Klik op **Opslaan** om het bewerken van de onderliggende werkstroom of actie te voltooien.
- 7 Om terug te keren naar de bovenliggende werkstroom, sluit u de objecteditor.

Ontwerpfunctie voor vRealize Orchestrator-invoerformulier

Als voor een werkstroom invoerparameters zijn vereist, wordt een dialoogvenster geopend waarin gebruikers de vereiste waarden invoeren. U kunt de inhoud, lay-out en presentatie van dit dialoogvenster indelen met de ontwerpfunctie voor invoerformulieren.

De ontwerpfunctie voor invoerformulieren vindt u op het tabblad **Invoerformulier** van de werkstroomeditor. Deze ontwerpfunctie bestaat uit een navigatiemenu, ontwerpcanvas en eigenschappenmenu. U kunt invoer en generieke elementen vanuit het linkermenu naar het ontwerpcanvas slepen. Op het canvas kunt u de positie van de invoerparameters instellen, deze in afzonderlijke invoertabbladen indelen en de eigenschappen voor invoerparameters configureren.

Opmerking U kunt geen inhoud van het tabblad **Variabelen** van de werkstroomeditor gebruiken in de ontwerpfunctie voor invoerformulieren. U kunt alleen parameters gebruiken van het tabblad **Invoer/Uitvoer**.

Algemene elementen

U kunt algemene elementen, zoals vervolgkeuzelijsten en tekstvakken voor wachtwoorden, toevoegen aan de ontwerpfunctie voor invoerformulieren. Algemene elementen komen niet

overeen met werkelijke invoerparameters, maar kunnen aan invoerparameters worden gebonden.

Het dialoogvenster met invoerparameters voor de werkstroom maken in de vRealize Orchestrator-client

U kunt de ontwerpfunctie voor invoerformulieren gebruiken om het dialoogvenster met invoerparameters voor de werkstroom te maken en aan te passen.

Voorwaarden

Controleer of de werkstroom een gedefinieerde lijst met invoerparameters heeft.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Bibliotheek > Werkstromen**.
- 3 Selecteer uw aangepaste werkstroom.
- 4 Klik op het tabblad **Invoerformulier**.
- 5 (Optioneel) Maak tabbladen voor gebruik in het invoerdialoogvenster.
U kunt tabbladen gebruiken om de structuur van uw dialoogvenster in te delen.
- 6 Selecteer uw invoerparameters.
- 7 Bewerk de eigenschappen van de invoerparameters.
Zie [Eigenschappen voor invoerparameters in de vRealize Orchestrator-client](#) voor meer informatie over eigenschappen voor invoerparameters.
- 8 (Optioneel) Voeg algemene elementen toe aan het canvas en bind deze aan invoerparameters.
- 9 (Optioneel) Voeg externe validatie toe aan de invoerparameters. Zie [Acties gebruiken om vRealize Orchestrator-werkstroominvoer te valideren](#) voor meer informatie.
- 10 Klik op **Opslaan**.

Resultaten

U hebt de lay-out van het werkstroombdialoogvenster gemaakt en de eigenschappen voor de invoerparameters ingesteld.

Eigenschappen voor invoerparameters in de vRealize Orchestrator-client

U kunt parametereigenschappen instellen om de invoerparameters te beperken die gebruikers opgeven wanneer ze vRealize Orchestrator-werkstromen uitvoeren.

Met vRealize Orchestrator kunt u de parametereigenschappen definiëren die worden gebruikt om de waarden van invoerparameters te kwantificeren die in werkstromen worden gebruikt. De parametereigenschappen die u definieert, stellen limieten in voor de typen en waarden van de invoerparameters die gebruikers kunnen opgeven in vRealize Orchestrator-werkstromen.

Parametereigenschappen valideren de invoerparameters en wijzigen de presentatie van de tekstvakken die worden weergegeven in het dialoogvenster met invoerparameters. Sommige parametereigenschappen kunnen afhankelijkheden tussen de parameters maken.

Parametereigenschap	Beschrijving
Label	Stel het label van de invoerparameter in.
Weergavetype	Stel het weergavetype van het invoertekstvak in.
Zichtbaarheid	Stel de zichtbaarheid van de invoerparameter in.
Alleen-lezen	Stel het invoertekstvak in als alleen-lezen.
Aangepaste help	Stel de wegwijzerbeschrijving van de invoerparameter in.
Standaardwaarde	Stel de standaardwaarde van de invoerparameter in.
Stap	Wordt gebruikt voor numerieke invoertypen. Stel in met hoeveel de waarde van de invoerparameter per klik moet worden verhoogd.
Vereist	Bepaalt of de waarde van de invoerparameter verplicht is of niet.
Reguliere expressie	Valideert de invoer met behulp van een reguliere expressie.
Minimumwaarde	Stel de minimumwaarde of lengte van de parameter in.
Maximumwaarde	Stel de maximumwaarde of -lengte van de parameter in.
Overeenkomst met tekstvak	Stel de waarde van de invoerparameter zo in dat deze overeenkomt met de waarde van een andere invoerparameter.
Waardebron	<p>Stel de waardebron van de parametereigenschappen op de tabbladen Vormgeving, Waarde en Beperkingen in.</p> <p>Opmerking U kunt de waarde van externe acties importeren met behulp van Externe bron. Het filteren van beschikbare acties wordt uitgevoerd op basis van het parametertype.</p>

Acties gebruiken om vRealize Orchestrator-werkstroominvoer te valideren

Gebruik externe acties om de invoer van uw aangepaste werkstromen te valideren.

Voorwaarden

Maak een aangepaste werkstroom met invoerparameters. Zie [Werkstromen maken in de vRealize Orchestrator-client](#) voor meer informatie.

U kunt de ontwerpfunctie voor invoerformulieren gebruiken om externe validaties voor uw werkstroominvoer te maken. Externe validaties gebruiken actiescripts die een tekenreekswaarde retourneren wanneer de waarde van de invoerparameter een fout bevat. Als de waarde van de invoerparameter geldig is, retourneert de externe validatie niets.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Maak een validatieactie.
 - a Ga naar **Bibliotheek > Acties**.
 - b Klik op **Nieuwe actie**.
 - c Voer de vereiste informatie in op het tabblad **Samenvatting**.
 - d Voer de invoerparameters voor de validatieactie in.

Opmerking De namen van de invoerparameters voor de validatieactie moeten identiek zijn aan de namen van de invoerparameters voor de werkstroom die worden gevalideerd.

- e Voer het script van de validatieactie in op het tabblad **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
```

Opmerking Het voorgaande script is een eenvoudig voorbeeld en vertegenwoordigt niet het volledige bereik van de validatiescripts die kunnen worden gebruikt.

- f Klik op **Opslaan**.
- 3 Pas externe validatie toe.
 - a Ga naar **Bibliotheek > Werkstromen**.
 - b Selecteer uw aangepaste werkstroom.
 - c Selecteer het tabblad **Invoerformulier**.
 - d Selecteer het klembordpictogram linksboven op het scherm.
 - e Sleep een vRealize Orchestrator-validatie-element naar het canvas.
 - f Selecteer het validatie-element, voer een validatielabel in en selecteer de validatieactie.
 - g (Optioneel) Maak aanvullende validatie-elementen.
 - h Klik op **Opslaan**.

4 Voer de werkstroom uit.

Als er een fout optreedt tijdens de validatie, wordt een tekenreeks geretourneerd. Als de validatie is gelukt, wordt er niets geretourneerd en wordt de uitvoering van de werkstroom voortgezet.

Resultaten

U hebt een externe validatie voor uw aangepaste vRealize Orchestrator-werkstroom gemaakt.

Aanvragen voor gebruikersinteractie in de vRealize Orchestrator-client

Werkstromen kunnen aanvullende gebruikersinvoer aanvragen voordat ze kunnen worden voltooid.

Werkstromen die verdere gebruikersinteractie vereisen, schorten de werking op tot de aangevraagde invoerparameters worden opgegeven door de gebruiker. Werkstromen definiëren welke gebruikers de aangevraagde informatie kunnen opgeven en dienovereenkomstig aanvragen voor interactie moeten verzenden. Werkstromen die wachten op gebruikersinvoer worden weergegeven in het paneel **Recente werkstroomuitvoeringen** van het vRealize Orchestrator Client-dashboard en het meldingsmenu aan de rechterkant.

Werkstromen plannen op de vRealize Orchestrator-client

U kunt planning gebruiken om uitvoeringen van uw vRealize Orchestrator-werkstroom te automatiseren.

Wanneer u werkstroomuitvoeringen plant, stelt u de datum, tijd en intervallen in waarop de geplande taak moet worden uitgevoerd.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Selecteer uw werkstroom in het menu **Bibliotheek** en klik in het werkstroompaneel op **Plannen**.
- 3 Configureer de parameters voor geplande taken in de categorieën **Algemeen**, **Planning** en **Werkstroom**.

Opmerking De parametercategorie **Werkstroom** is alleen zichtbaar voor werkstromen waarvoor invoerparameters zijn vereist.

Parameter	Beschrijving
Naam	De naam van de geplande taak.
Beschrijving	Een korte beschrijving van het doel van de geplande taak.

Parameter	Beschrijving
Start	De datum en tijd van de eerste geplande uitvoering van de werkstroom.
Starten indien in het verleden	Selecteer of de werkstroom moet worden gestart als de geplande tijd in het verleden ligt. Met Ja start de geplande werkstroom onmiddellijk. Met Nee start de werkstroom bij het volgende geplande terugkeerpatroon.
Plannen	Stel de velden voor het terugkeerpatroon en de gebeurtenistrigger van de geplande taak in.
Einddatum	Alleen zichtbaar als Geen terugkeerpatroon is geselecteerd. Stel de datum en tijd in waarop de geplande taak eindigt.
Werkstroom	Voer de invoerparameters van de werkstroom in.

4 Klik op **Maken**.

Resultaten

U hebt een geplande taak voor de werkstroom gemaakt. Geplande werkstromen worden weergegeven onder **Activiteit > Gepland**. U kunt geplande taken verwijderen door op **Verwijderen** te klikken in het paneel Plannen.

Een geplande taak in de vRealize Orchestrator-client bewerken

Geplande taken kunnen worden bewerkt om parameters zoals datum, tijd en herhaling van de geplande werkstroom te wijzigen.

Voorwaarden

Maak een geplande werkstroomtaak.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Selecteer uw geplande taak in **Activiteit > Gepland**.
- 3 Klik op **Bewerken** in het werkstroompaneel.
- 4 Bewerk het schema en klik op **Opslaan**.

Opmerking Invoerparameters die zijn ingesteld bij het maken van de geplande taak, zijn alleen-lezen en kunnen niet worden bewerkt. Om deze parameters te wijzigen, maakt u een nieuwe geplande taak voor deze werkstroom.

Acties beheren

6

U kunt uw vRealize Orchestrator-werkstromen aanpassen door actiescripts toe te voegen.

De vRealize Orchestrator Client biedt bibliotheken met vooraf gedefinieerde acties en een actie-editor voor aangepaste actiescripts. Acties vertegenwoordigen individuele functies die u als bouwstenen in werkstromen gebruikt.

Acties zijn JavaScript-functies. Acties kunnen meerdere invoerparameters hebben en één retourwaarde krijgen. Acties kunnen elk object in de vRealize Orchestrator API aanroepen of objecten in elke API die u in vRealize Orchestrator importeert met behulp van een invoegtoepassing.

Wanneer een werkstroom wordt uitgevoerd, krijgt een actie invoerparameters uit de variabelen van de werkstroom. Deze variabelen kunnen de initiële invoerparameters van de werkstroom zijn of variabelen die door andere elementen in de werkstroom worden ingesteld wanneer deze worden uitgevoerd.

De actie-editor bevat een functie voor automatisch aanvullen voor scripts en een API Explorer met de beschikbare scripttypen en hun documentatie.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Acties in de vRealize Orchestrator-client maken](#)
- [Acties uitvoeren en fouten opsporen](#)
- [Basisconcepten voor Python-, Node.js- en PowerShell-scripts](#)
- [Runtimelimieten voor Python, Node.js en PowerShell-scripts](#)

Acties in de vRealize Orchestrator-client maken

U kunt de vRealize Orchestrator Client gebruiken om actiescripts te maken, bewerken en verwijderen.

Vanaf vRealize Orchestrator 8.1 kunt u de volgende runtimes gebruiken:

- Python 3.7
- Node.js 12

- PowerCLI 11/Powershell 6.2
- PowerCLI 12/Powershell 7

Opmerking De PowerCLI-runtime bevat PowerShell en de volgende modules: VMware.PowerCLI, PowerNSX, PowervRA.

Voorwaarden

Voordat u een Python-, Node.js- of PowerShell-script maakt, controleert u of u bekend bent met de basisconcepten voor het ontwikkelen van vRealize Orchestrator-compatibele scripts die deze runtimes gebruiken. Zie [Basisconcepten voor Python-, Node.js- en PowerShell-scripts](#).

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Bibliotheek > Acties**.
- 3 Klik op **Nieuwe actie**.
- 4 Voer op het tabblad **Algemeen** de naam en de modulenaam van de actie in.

Opmerking De naam en modulenaam moeten uniek zijn voor elke actie. De naam van de actie moet een geldige JavaScript-functie zijn. De naam van de actie moet een enkel woord zijn dat alleen letters, cijfers en de symbolen voor dollar ("\$\$") en onderstrepingstekens ("_") mag bevatten. De modulenaam moet bestaan uit woorden die door een punt (".") worden gescheiden.

- 5 (Optioneel) Maak een beschrijving, versienummer, tags en groepsrechten voor de actie.
- 6 Voeg op het tabblad **Script** actie-invoer toe, selecteer het retourtype van de uitvoer en schrijf het script.

Opmerking Door **Zip** te selecteren in het vervolgkeuzemenu **Type**, kunt u een externe scriptbron en, indien van toepassing, de afhankelijkheidsmodules ervan importeren.

- 7 Als u het bewerken van de actie wilt voltooien, klikt u op **Opslaan**.

In een bericht wordt gemeld dat de actie is opgeslagen.

Wat nu te doen

Zie [Amazon Web Services in vRealize Orchestrator integreren met behulp van Python](#) voor een voorbeeldscenario voor het gebruik van vRealize Orchestrator-acties.

Acties uitvoeren en fouten opsporen

U kunt uw acties verbeteren door deze rechtstreeks in de actie-editor uit te voeren en fouten op te sporen.

Vanaf vRealize Orchestrator 8.1 kunt u acties rechtstreeks in de actie-editor van de vRealize Orchestrator-client uitvoeren en fouten opsporen. Met deze functie kunt u garanderen dat uw acties als verwacht worden uitgevoerd wanneer ze in uw werkstromen worden geïntegreerd.

Acties uitvoeren in de vRealize Orchestrator-client

Als werkstroomontwerper wilt u uw acties uitvoeren voordat u ze integreert in een werkstroom.

Voorwaarden

Maak een actie. Zie [Acties in de vRealize Orchestrator-client maken](#).

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Navigeer naar **Bibliotheek > Acties** en selecteer de actie die u wilt uitvoeren.
- 3 Klik op **Uitvoeren**.
- 4 Voer de vereiste invoerparameters in en klik op **Uitvoeren**.

Nadat de uitvoering van de actie is voltooid, klikt u op het tabblad **Resultaten/Invoer**. Als tijdens de uitvoering van een actie een fout is opgetreden, wordt deze op dit tabblad in een rode kleur weergegeven. U kunt de details van de uitvoering van de actie bekijken via het element **Actieresultaten**.

Opmerking De resultaten van de uitvoering van de actie worden niet opgeslagen.

Acties voor foutopsporing in de vRealize Orchestrator-client

Als werkstroomontwerper kunt u foutopsporing uitvoeren voor acties door onderbrekingspunten in uw script in te voegen.

vRealize Orchestrator bevat een ingebouwd foutopsporingsprogramma dat u kunt gebruiken om fouten in de script- en invoereigenschappen van uw actie op te sporen. De foutopsporing kan worden gestart in de actie-editor door onderbrekingspunten in te voegen in de scriptregels van uw actie.

Opmerking Het ingebouwde foutopsporingsprogramma werkt alleen met acties die gebruikmaken van de standaard JavaScript-runtime. Zie [Foutopsporing voor de actie Amazon Web Services uitvoeren](#) voor een voorbeeld van de manier waarop u foutopsporing kunt uitvoeren op actiescripts die verschillende runtimes gebruiken.

Voorwaarden

Maak een actie. Zie [Acties in de vRealize Orchestrator-client maken](#).

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Bibliotheek > Acties** en selecteer de actie waarvoor u foutopsporing wilt uitvoeren.

- 3 Voeg in de actie-editor onderbrekingspunten toe aan de regels van het actiescript waarvoor u foutopsporing wilt uitvoeren.
- 4 Klik op **Fouten opsporen**
- 5 Voer de invoerparameters van uw actie in en klik op **Uitvoeren**.
Er wordt een actie gestart in de foutopsporingsmodus.
- 6 Wanneer de actie-uitvoering is onderbroken nadat een onderbrekingspunt is bereikt, selecteert u een van de volgende opties:

Optie	Beschrijving
Doorgaan	Hervat de uitvoering van de actie totdat een andere onderbrekingspunt is bereikt of de uitvoering van de actie is voltooid.
Stap naar	Stap in de huidige actiefunctie. Als het foutopsporingsprogramma niet dieper in de huidige regel van de functie kan gaan, wordt een Stap over -bewerking uitgevoerd.
Stap over	Het foutopsporingsprogramma gaat door naar de volgende regel van de huidige functie.
Stap afsluiten	Het foutopsporingsprogramma gaat naar de regel die zal worden uitgevoerd wanneer de huidige functie wordt geretourneerd.

- 7 (Optioneel) Voeg expressies toe op het tabblad **Foutopsporingsprogramma**.
- 8 (Optioneel) Bewerk de waarde van uw variabelen op het tabblad **Foutopsporingsprogramma**.

Basisconcepten voor Python-, Node.js- en PowerShell-scripts

Wanneer u uw script maakt voor gebruik in vRealize Orchestrator, moet u controleren of uw script de juiste structuur en opmaak heeft.

Ondersteunde runtimes

Vanaf vRealize Orchestrator 8.1 kunt u de volgende runtimes gebruiken:

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

Opmerking De PowerCLI-runtime bevat PowerShell en de volgende modules: VMware.PowerCLI, PowerNSX, PowervRA.

U kunt elke aangepaste broncode aan de nieuwe runtimes toevoegen, maar om context en invoer te accepteren en een resultaat te retourneren van en naar de vRealize Orchestrator-engine, moet u de juiste functionele indeling volgen.

Aanbeveling voor scriptverwerking

Voor eenvoudigere scripttaken kunt u elementen **Scriptbare taak** toevoegen aan uw werkstroomschema. U kunt vRealize Orchestrator-acties gebruiken voor meer complexe scripttaken.

Het gebruik van acties biedt twee specifieke voordelen:

- Acties kunnen onafhankelijk van werkstromen worden gemaakt, bijgewerkt, geïmporteerd en geëxporteerd.
- Acties zijn standalone objecten die kunnen worden uitgevoerd en waarvoor fouten kunnen worden opgespoord in hun eigen omgeving. Dit kan leiden tot een vloeiender ontwikkelingsproces. Zie [Acties uitvoeren en fouten opsporen](#).

Vereisten voor scriptfuncties

De standaardnaam voor uw scriptfunctie is **handler**. De functie accepteert twee argumenten, context en invoer. Context is een toewijzingsobject met systeeminformatie. `vroURL` kan bijvoorbeeld de URL van de vRealize Orchestrator-instantie bevatten die u wilt aanroepen, terwijl `executionId` de token-id van een werkstroomuitvoering bevat.

Een invoer is een toewijzingsobject met alle invoer die aan de acties is geleverd. Als u bijvoorbeeld een invoer definieert in uw actie met de naam `myInput`, kunt u deze openen via het argument `invoer`, zoals `inputs.myInput` of `inputs["myInput"]`, afhankelijk van uw runtime. Alles wat u retourneert van de functie, is het resultaat van de actie. Daarom moet het retourtype van uw actie overeenkomen met het type inhoud dat het script retourneert in vRealize Orchestrator. Als u een primitief getal retourneert, moet het retourtype van de actie een nummertype zijn. Als u een tekenreeks retourneert, moet het retourtype van de actie een tekenreekstype zijn. Als u een complex object retourneert, moet het retourtype worden toegewezen aan ofwel `Properties` of `Composite Type`. Deze principes zijn ook van toepassing op arrays.

Ondersteunde typen in- en uitvoerparameters voor Python-, Node.js- en PowerShell-runtimes:

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

De vermeldingshandler definiëren

Standaard is de waarde van de vermeldingshandler `handler.handler`. Deze waarde betekent dat de vRealize Orchestrator-engine zoekt naar een bestand op het hoogste niveau in uw ZIP-pakket met de naam `handler.py`, `handler.js` of `handler.ps1`, dat een functie bevat genaamd `handler`. Eventuele verschillen met de namen van het functie- en handlerbestand moeten worden weergegeven in de waarde van de vermeldingshandler. Als uw hoofdhandler bijvoorbeeld `index.js` heet en uw functie `callMe` heet, moet u de waarde voor de vermeldingshandler instellen op `index.callMe`.

Runtimescripts voor foutopsporing in een externe IDE

vRealize Orchestrator ondersteunt Python- en Node.js-scripts voor foutopsporing in een externe IDE. U kunt geen foutopsporing uitvoeren op PowerShell-scripts in een externe IDE.

Runtimelimieten voor Python, Node.js en PowerShell-scripts

In sommige Python-, Node.js- of PowerShell-scripts moet u mogelijk de geheugen- en time-outwaarden in de vRealize Orchestrator Client wijzigen.

De vRealize Orchestrator Client gebruikt een set standaardgeheugen- en time-outwaarden voor Python-, Node.js- en PowerShell-actiescripts:

- Geheugen: 64 MB
- Time-out: 180 seconden

Als uw actiescript een of beide van deze standaardwaarden overschrijdt, mislukt de uitvoering van de actie. Uw actiescript kan bijvoorbeeld meerdere afhankelijkheidsmodules van derden gebruiken. In een dergelijk scenario is de standaardgeheugenlimiet van 64 MB mogelijk niet voldoende.

Om te voorkomen dat de uitvoering van acties mislukt vanwege onvoldoende bronnen, wijzigt u de geheugen- en time-outwaarden van de actie-editor.

Opmerking U kunt ook overwegen uw script te verdelen in meerdere scriptbare taakelementen die aan uw werkstromen kunnen worden toegevoegd.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Navigeer naar **Bibliotheek > Acties** en selecteer uw actie.
- 3 Selecteer het tabblad **Script**.
- 4 Wijzig de geheugen- en time-outwaarden onder **Runtimelimieten**.
- 5 Klik op **Opslaan**.
- 6 Als u de nieuwe runtimelimieten wilt testen, klikt u op **Fouten opsporen**.

Configuratie-elementen beheren

7

Een configuratie-element is een lijst met variabelen die u kunt gebruiken om constanten te configureren in een volledige vRealize Orchestrator-serverimplementatie.

U kunt configuratie-elementen gebruiken om variabelen beschikbaar te maken voor alle werkstromen, acties en beleidsregels die op de vRealize Orchestrator-server worden uitgevoerd.

Als u een pakket maakt met een werkstroom, actie of beleid dat een variabele gebruikt van een configuratie-element, wordt in vRealize Orchestrator automatisch het configuratie-element in het pakket opgenomen. Als u een pakket met een configuratie-element in een andere vRealize Orchestrator-server importeert, kunt u ook de waarden voor configuratie-elementvariabelen importeren. Als u bijvoorbeeld een werkstroom maakt waarvoor waarden voor variabelen zijn vereist die afhankelijk zijn van de vRealize Orchestrator-server waarop deze wordt uitgevoerd, kunt u die variabelen in een configuratie-element instellen zodat u die werkstroom kunt exporteren, zodat een andere vRealize Orchestrator-server deze kan gebruiken. Configuratie-elementen stellen u daarom in staat om werkstromen, acties en beleidsregels tussen servers gemakkelijker uit te wisselen.

Opmerking U kunt geen waarden voor een configuratie-elementvariabele importeren uit een configuratie-element dat vanaf vRealize Orchestrator 5.1 of lager is geëxporteerd.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Configuratie-elementen maken in de vRealize Orchestrator-client](#)

Configuratie-elementen maken in de vRealize Orchestrator-client

Met configuratie-elementen kunt u algemene variabelen instellen op een vRealize Orchestrator-server. Alle elementen die op de server worden uitgevoerd, kunnen de variabelen gebruiken die u in een configuratie-element heeft ingesteld.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Assets > Configuraties**.

- 3 Selecteer **Nieuwe configuratie**.
- 4 Voer de naam van het configuratie-element in.
- 5 Selecteer het tabblad **Variabelen**.
- 6 Als u een lokale variabele wilt maken, klikt u op **Nieuw**.
 - a Voer de naam van de variabele in.
 - b Selecteer het type variabele.

Opmerking Als u een array met configuratievariabelen wilt maken, schakelt u het selectievakje **Array** in.

- c (Optioneel) Voer een waarde in voor de configuratievariabele.
 - d Klik op **Opslaan**.
- 7 Klik op **Opslaan** om het maken van een configuratie-element te voltooien.

Wat nu te doen

U kunt het configuratie-element gebruiken om variabelen aan werkstromen, acties of beleid toe te voegen.

Beleid beheren

8

Beleidsregels zijn gebeurtenistriggers waarmee de activiteit van het systeem wordt gecontroleerd. Beleidsregels reageren op vooraf gedefinieerde gebeurtenissen die worden uitgegeven door wijzigingen in de status of prestaties van specifieke vRealize Orchestrator-objecten.

Beleidsregels zijn een reeks regels, meters, drempelwaarden en gebeurtenisfilters die bepaalde werkstromen of scripts uitvoeren wanneer specifieke vooraf gedefinieerde gebeurtenissen plaatsvinden in vRealize Orchestrator of in de technologieën waartoe vRealize Orchestrator via invoegtoepassingen toegang krijgt. vRealize Orchestrator evalueert voortdurend de beleidsregels wanneer het beleid wordt uitgevoerd. U kunt bijvoorbeeld beleidsmaten en drempelwaarden implementeren die het gedrag van vCenter Server-objecten van de typen VC:HostSystem en VC:VirtualMachine bewaken.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Beleidsregels maken en toepassen op de vRealize Orchestrator-client](#)
- [Beleidselementen in de vRealize Orchestrator-client](#)
- [Beleidsuitvoeringen in de vRealize Orchestrator-client beheren](#)

Beleidsregels maken en toepassen op de vRealize Orchestrator-client

U kunt beleidsregels gebruiken om de activiteiten van het vRealize Orchestrator-systeem voor specifieke gebeurtenissen te bewaken.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Bibliotheek > Beleid**.
- 3 Selecteer **Nieuw beleid**.
U heeft een leeg beleid gecreëerd.
- 4 Voer een beleidsnaam en versienummer in.

5 Selecteer het tabblad **Variabelen**.

6 Als u een lokale variabele wilt maken, klikt u op **Nieuw**.

- a Voer de naam van de variabele in.
- b Selecteer het type variabele.

Opmerking Als u een array met beleidsvariabelen wilt maken, schakelt u het selectievakje **Array** in.

- c Voer de waarde van de variabele in.

Opmerking Als u de waarde van een configuratie-elementvariabele wilt importeren, kunt u **Binden aan configuratie** gebruiken.

- d Klik op **Opslaan**.

7 Op het tabblad **Definitie** kunt u beleidselementen toevoegen en gebeurtenishandlers instellen.

Zie [Beleidselementen in de vRealize Orchestrator-client](#) voor meer informatie over beleidselementen.

8 Klik op **Opslaan**.

U heeft het beleid geconfigureerd.

Wat nu te doen

Om een beleid te starten, selecteert u het beleid en klikt u op **Uitvoeren**. Voer de naam van de beleidsuitvoering in en, indien u hierom wordt gevraagd, de vereiste invoerparameters.

Om de beleidsstatus weer te geven, gaat u naar **Activiteit > Beleidsuitvoeringen**.

Beleidselementen in de vRealize Orchestrator-client

U kunt beleidselementen gebruiken om vooraf gedefinieerde vRealize Orchestrator-werkstromen of -scripts uit te voeren wanneer een gebeurtenis optreedt.

U kunt een beleidselement toevoegen om werkstroom- of scriptuitvoeringen te activeren als reactie op gebeurtenissen die door objecten worden geactiveerd. Met het periodieke gebeurteniselement kunt u werkstroom- of scriptuitvoeringen plannen. Met het rootelement kunt u het start- of stopgedrag van beleidsregels instellen. Beleidselementen kunnen gebeurtenishandlers hebben die bepalen wanneer beleidselementen moeten worden uitgevoerd.

Opmerking Gebeurtenishandlers die beleidselementen activeren, kunnen werkstromen of actiescripts zijn. Als u zowel een werkstroom als een script toevoegt aan een gebeurtenishandler, negeert het beleid de scripttrigger en wordt alleen de werkstroomtrigger gebruikt.

Gebeurtenishandler	Beschrijving
OnInit	Het beleidselement wordt geactiveerd telkens wanneer u het beleid start.
OnExit	Het beleidselement wordt geactiveerd telkens wanneer u het beleid stopt.
OnExecute	Wordt gebruikt door het periodieke gebeurteniselement. Activeert het beleidselement gedurende de tijd die is opgegeven in het periodieke gebeurteniselement.

Opmerking Technologieën die zijn aangesloten op de vRealize Orchestrator-database kunnen unieke gebeurtenishandlers hebben. Via de SNMP-invoegtoepassing kunt u bijvoorbeeld de gebeurtenishandler **OnTrap** gebruiken bij het maken van op SNMP gebaseerde beleidselementen.

Beleidselementen worden geconfigureerd op het tabblad **Definitie** in het venster voor beleidsbewerking.

Beleidsuitvoeringen in de vRealize Orchestrator-client beheren

U kunt de vRealize Orchestrator Client gebruiken om de beleidsprioriteit en het opstartgedrag voor servers van beleidsregels te beheren voor wanneer de vRealize Orchestrator-server opnieuw wordt opgestart.

Voorwaarden

Een beleid maken en uitvoeren. Zie [Beleidsregels maken en toepassen op de vRealize Orchestrator-client](#) voor meer informatie.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client als beheerder.
- 2 Navigeer naar **Activiteit > Beleidsuitvoeringen**.
- 3 Klik op de beleidsuitvoering die u wilt beheren.
- 4 Klik op **Stoppen**.
De beleidsstatus verandert in **Gestopt**.
- 5 Stel op het tabblad **Algemeen** de beleidsprioriteit en het opstartgedrag van de server in.
- 6 Klik op **Uitvoeren** om het beleid opnieuw te starten.
De beleidsstatus wordt gewijzigd in **In uitvoering**.

Resource-elementen beheren

9

Werkstromen kunnen objecten gebruiken die u onafhankelijk van vRealize Orchestrator maakt als kenmerken. Als u externe objecten als kenmerken in werkstromen wilt gebruiken, importeert u deze als resource-elementen in de server.

Objecten die vRealize Orchestrator-werkstromen als resource-elementen kunnen gebruiken, zijn afbeeldingsbestanden, scripts, XML-sjablonen, HTML-bestanden, enzovoort. Alle werkstromen die op de vRealize Orchestrator-server worden uitgevoerd, kunnen alle resource-elementen gebruiken die u in vRealize Orchestrator importeert.

Nadat u een object als resource-element in vRealize Orchestrator hebt geïmporteerd, kunt u op één locatie wijzigingen in het object aanbrengen en deze wijzigingen automatisch doorvoeren in alle werkstromen die dit resource-element gebruiken.

De maximale grootte van een resource-element is 16 MB.

U kunt een resource-element importeren, exporteren, herstellen, bijwerken en verwijderen.

Pakketten beheren

10

Gebruik de vRealize Orchestrator Client om pakketten te maken, te exporteren en te importeren. Pakketten kunnen worden gebruikt om werkstroomobjecten te exporteren voor gebruik op andere vRealize Orchestrator-instanties.

Pakketten kunnen werkstromen, acties, beleidsregels, configuratie-elementen of resource-elementen bevatten.

Wanneer u een element aan een pakket toevoegt, wordt door vRealize Orchestrator op afhankelijkheden gecontroleerd en worden afhankelijke elementen aan het pakket toegevoegd. Als u bijvoorbeeld een werkstroom toevoegt die acties of andere werkstromen gebruikt, voegt vRealize Orchestrator deze acties en werkstromen toe aan het pakket.

Wanneer u een pakket importeert, vergelijkt de server de versies van de verschillende elementen van de inhoud met de overeenkomende lokale elementen. De vergelijking toont de verschillen in versies tussen de lokale en geïmporteerde elementen. De gebruiker kan beslissen of het pakket moet worden geïmporteerd, of specifieke elementen selecteren om te importeren.

Voor de meeste objecten die in de vRealize Orchestrator Client zijn gemaakt, zijn naast bronelementen pakketten de enige manier om deze objecten te exporteren en te importeren.

Pakketten maken gebruik van Digital Rights Management om te bepalen hoe de ontvangende server de inhoud van het pakket kan gebruiken. vRealize Orchestrator ondertekent pakketten en versleutelt de pakketten voor gegevensbescherming. Pakketten kunnen bijhouden welke gebruikers elementen exporteren en opnieuw verdelen met x509-certificaten.

Een pakket maken in de vRealize Orchestrator-client

U kunt werkstromen, beleidsregels, acties, invoegtoepassingsverwijzingen, resource-elementen en configuratie-elementen in pakketten exporteren en importeren. Alle afhankelijke elementen die betrekking hebben op pakketobjecten, worden automatisch aan het pakket toegevoegd om compatibiliteit tussen versies te garanderen. Als u afhankelijke elementen wilt verwijderen, moet u eerst het gerelateerde pakketobject verwijderen.

Voor de meeste objecten die in de vRealize Orchestrator Client zijn gemaakt, zijn naast bronelementen pakketten de enige manier om deze objecten te exporteren en te importeren.

Voorwaarden

Controleer of de vRealize Orchestrator-server objecten zoals werkstromen, acties en beleidsregels bevat die u aan een pakket kunt toevoegen.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Assets > Pakketten**.
- 3 Klik op **Nieuw pakket**.
- 4 Voer op het tabblad **Algemeen** een naam en beschrijving voor het pakket in.

Opmerking U kunt geen speciale tekens gebruiken wanneer u een naam geeft aan pakketten in de vRealize Orchestrator Client.

- 5 Klik op het tabblad **Inhoud** op **Toevoegen**.
- 6 Selecteer de objecten die u aan het pakket wilt toevoegen en klik op **Toevoegen**.

Opmerking Afhankelijke elementen worden automatisch aan het pakket toegevoegd, maar worden niet weergegeven op het tabblad **Inhoud** op het moment dat het pakket wordt gemaakt. Als u afhankelijke elementen wilt weergeven, selecteert u het tabblad **Inhoud** nadat u het pakket heeft gemaakt.

- 7 Om het maken van het pakket te voltooien, klikt u op **Maken**.

Een pakket in de vRealize Orchestrator-client exporteren

U kunt de vRealize Orchestrator Client gebruiken om pakketten naar een andere vRealize Orchestrator-omgeving te exporteren.

Voorwaarden

Maak een pakket met de vRealize Orchestrator-objecten die u wilt exporteren. Zie [Een pakket maken in de vRealize Orchestrator-client](#) voor meer informatie.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Assets > Pakketten**.
- 3 Klik op **Exporteren** op het pakket.

4 (Optioneel) Selecteer aanvullende exportopties.

Optie	Beschrijving
Configuratiekenmerkwaarden aan pakket toevoegen	Exporteer de kenmerkwaarden van de configuratie-elementen.
SecureString-kenmerkwaarden van configuratie toevoegen aan pakket	Exporteer de waarden van de SecureString-configuratiekenmerken.
Algemene tags toevoegen aan pakket	Exporteer de algemene tags.

5 Stel de toegangsrechten in voor gebruikers die het pakket importeren.

Optie	Beschrijving
Inhoud weergeven	De gebruiker kan de inhoud van het pakket weergeven.
Toevoegen aan pakket	De gebruiker kan inhoud van het geïmporteerde pakket toevoegen aan andere pakketten.
Inhoud bewerken	De gebruiker kan de inhoud van het pakket bewerken.

6 Klik op **OK**.

Opmerking Bestanden met de extensie `.package` worden opgeslagen in een standaardmap op uw lokale machine. Als u een aangepaste map wilt instellen, kunt u de opslaginstellingen in uw browser wijzigen.

Resultaten

U heeft het pakket geëxporteerd. U kunt nu de geëxporteerde objecten in een andere vRealize Orchestrator-omgeving gebruiken.

Een pakket importeren in de vRealize Orchestrator-client

U kunt de vRealize Orchestrator Client gebruiken om werkstroompakketten te importeren. Door pakketten te importeren, kunt u objecten van de ene vRealize Orchestrator-server opnieuw gebruiken op een andere server.

Voorwaarden

- Maak een back-up van de vRealize Orchestrator-standaardobjecten die u heeft aangepast.
- Maak en exporteer een pakket met de objecten die u wilt importeren op de externe server.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Ga naar **Assets > Pakketten**.
- 3 Klik op **Importeren**, blader naar het bestand `.package` dat u wilt importeren en klik op **Openen**.

4 Controleer de geïmporteerde pakketinformatie.

- a Op het tabblad **Algemeen** vindt u informatie over het geïmporteerde pakket, zoals de naam, beschrijving, het aantal ingesloten items en certificaatinformatie.

Mogelijk wordt u gevraagd om aan te geven dat u het uitgevercertificaat van de vRealize Orchestrator-broninstantie vertrouwt voordat u het bestand kunt importeren.

- b Op het tabblad **Pakketelementen** worden de objecten vermeld die in het importbestand zijn opgenomen. Als de versie van een object in het pakket hoger is dan de versie op de server, selecteert het systeem die objectversie voor het importeren. Eerdere versies van vRealize Orchestrator-elementen moeten handmatig worden geselecteerd.
- c Schakel **Configuratiekenmerkwaarden importeren** uit als u de kenmerkwaarden van de configuratie-elementen niet uit het pakket wilt importeren.
- d Selecteer in het vervolgkeuzemenu of u tags wilt importeren.

5 Klik op **Importeren**.

Problemen oplossen in de vRealize Orchestrator-client

11

U kunt problemen met uw vRealize Orchestrator-instantie oplossen en de instantie bewaken met behulp van statistieken, tokenherhaling, validatie en foutopsporing.

Dit hoofdstuk omvat de volgende onderwerpen:

- [Metrische gegevens in de vRealize Orchestrator-client](#)
- [Herhaling van werkstroomtoken in de vRealize Orchestrator-client gebruiken](#)
- [vRealize Orchestrator-werkstromen valideren](#)
- [Scripts voor foutopsporingswerkstroom in de vRealize Orchestrator-client](#)
- [Foutopsporing uitvoeren in werkstromen per schema-element](#)

Metrische gegevens in de vRealize Orchestrator-client

vRealize Orchestrator-beheerders kunnen gebruikmaken van werkstroomprofiling en de metrieke van het systeemdashboard om problemen met het vRealize Orchestrator-systeem en werkstromen op te lossen.

De profilingfunctie verzamelt metriekgegevens over werkstroomuitvoeringen. Profiling van werkstromen is standaard ingeschakeld. U kunt automatisch profiling uitschakelen in **Control Center > Extensie-eigenschappen > profiler-8.2.0**.

De andere bron voor metriekgegevens in de vRealize Orchestrator Client is het systeemdashboard, dat metrieke op systeemniveau biedt. Zie [Het vRealize Orchestrator-systeemdashboard gebruiken](#) voor meer informatie.

Werkstromen in de vRealize Orchestrator-client profileren

U kunt uw werkstroomuitvoeringen profileren om problemen met uw vRealize Orchestrator-omgeving op te lossen en deze te optimaliseren.

U kunt de profilingfunctie van de vRealize Orchestrator Client gebruiken om nuttige metriekgegevens over uw werkstroomuitvoeringen te verzamelen. Deze gegevens kunnen worden gebruikt om de prestaties van uw werkstromen te optimaliseren. Werkstroomuitvoeringen worden standaard automatisch geprofileerd. U kunt automatische profiling uitschakelen op de pagina **Extensie-eigenschappen** van het vRealize Orchestrator Control Center en de profiler handmatig uitvoeren. Om een handmatige profiling uit te voeren, zoekt u uw werkstroom in de bibliotheek en selecteert u **Acties > Profiel**.

Voorwaarden

Voer een werkstroom uit.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client.
- 2 Navigeer naar **Activiteit > Werkstroomuitvoeringen**.
- 3 Selecteer een werkstroomuitvoering.

Op het schema voor de werkstroomuitvoering kunt u gegevens over de afzonderlijke werkstroomitems zien. De gegevens omvatten de totale uitvoeringsduur, de maximale duur en het aantal itemuitvoeringen. U kunt deze informatie in de vervolgkeuzelijst rechtsboven op de pagina filteren.

- 4 Selecteer het tabblad **Prestaties**.

Op dit tabblad vindt u metriekgegevens over CPU-tijden van werkstroomuitvoeringen, uitvoeringsduur, de tokengrootte en de gegevens van het werkstroomitem.

Opmerking Als de werkstroomuitvoering is opgeheven, bijvoorbeeld wanneer de werkstroom wacht op verdere invoer, legt de metriek voor CPU-tijden alleen de runtimethread vast die heeft plaatsgevonden vóór de voltooiing.

Wat nu te doen

Gebruik de gegevens die zijn verzameld bij de profiling om uw werkstroom te optimaliseren.

Het vRealize Orchestrator-systeemdashbord gebruiken

Als beheerder kunt u het vRealize Orchestrator Client-systeemdashbord gebruiken om nuttige metriekgegevens te verzamelen over de knooppunten van uw vRealize Orchestrator-omgeving.

U kunt het systeemdashbord openen door te klikken op het tabblad **Systeem**, bovenaan de pagina vRealize Orchestrator Client-dashbord. De opgegeven gegevens omvatten:

- Knooppuntstatus
- Knooppunteigenschappen
- Clusterinstellingen. U kunt alleen de clusterinstellingen van het systeemdashbord weergeven. Als u deze instellingen wilt wijzigen, gaat u naar de pagina **Orchestrator-clusterbeheer** van het vRealize Orchestrator Control Center.

- Informatie over threads
- Heapgeheugen
- Niet-heapgeheugen
- Bestandssysteemgebruik
- Verificatiegegevens
- Verbindingspool voor Orchestrator-database
- Invoerargumenten voor proces

Deze gegevens kunnen worden gebruikt om de status van individuele knooppunten van uw vRealize Orchestrator-omgeving te controleren en problemen op te lossen. Als u wilt navigeren tussen afzonderlijke knooppunten, klikt u op het tabblad dat is gekoppeld aan een knooppunt bovenaan het systeemdashboard.

Herhaling van werkstroomtoken in de vRealize Orchestrator-client gebruiken

U kunt de functie voor herhaling van tokens gebruiken om de overgangen tussen items in werkstroomuitvoeringen weer te geven.

De functie voor herhaling van tokens registreert contextuele informatie voor elke overgang tussen werkstroomitems. Voor elk werkstroomitem wordt herhaling van tokens geregistreerd wanneer de werkstroomuitvoering is gestart of beëindigd en welke variabelen aan het einde van de uitvoering van het werkstroomitem zijn gewijzigd. Herhaling van tokens verwijst ook naar de logboekberichten voor de gegenereerde scripts voor elk werkstroomitem.

Opmerking Gegevens over overgangen van werkstroomitems worden opgeslagen in de vRealize Orchestrator PostgreSQL-database. Deze gegevens worden uit de database verwijderd wanneer de werkstroomuitvoering wordt verwijderd.

Voorwaarden

- Schakel de functie voor herhaling van tokens inschakelen via het Control Center.
 - a Meld u aan bij het Control Center als **root**.
 - b Selecteer **Extensie-eigenschappen**.
 - c Klik op **tokenreplay-8.2.0**.
 - d Klik op **Inschakelen** om de functie voor herhaling van tokens in te schakelen.
 - e Klik op **Opslaan**.

Opmerking Het kan tot 5 minuten duren voordat de vRealize Orchestrator-server de extensie heeft vernieuwd.

- Voer een werkstroom uit.

Opmerking Herhaling van tokens wordt standaard niet automatisch uitgevoerd voor alle werkstroomuitvoeringen op uw vRealize Orchestrator-server. U kunt herhaling van tokens voor elke werkstroom afzonderlijk uitvoeren of de extensie voor herhaling van tokens inschakelen voor alle werkstromen op de pagina **Extensie-eigenschappen** van het Control Center.

Procedure

- 1 (Optioneel) Schakel herhaling van tokens in voor alle werkstroomuitvoeringen op uw vRealize Orchestrator-server.

Opmerking Als u afzonderlijke herhalingen van tokens wilt uitvoeren zonder de functie in het Control Center in te schakelen, klikt u op **Uitvoeren met herhaling** op de pagina met de werkstroomeditor.

- a Meld u aan bij het Control Center als **root**.
- b Selecteer **Extensie-eigenschappen**.
- c Klik op **tokenreplay-8.2.0**.
- d Als u de functie voor herhaling van tokens opnieuw wilt inschakelen voor alle werkstromen, controleert u of **Herhaling voor alle werkstroomuitvoeringen opnemen** is ingeschakeld.
- e Klik op **Opslaan**.

Opmerking Het kan tot 5 minuten duren voordat de vRealize Orchestrator-server de extensie heeft vernieuwd.

- 2 Meld u aan bij de vRealize Orchestrator-client als beheerder.
- 3 Navigeer naar **Activiteit > Werkstroomuitvoeringen**.
- 4 Selecteer een werkstroomuitvoering.
- 5 Selecteer een werkstroomuitvoeringsitem in het linkermenu.

De tabbladen **Variabele** en **Logboeken** geven nu specifieke informatie weer voor dat werkstroomitem.

vRealize Orchestrator-werkstromen valideren

vRealize Orchestrator biedt een tool voor werkstroomvalidatie. Het valideren van een werkstroom helpt bij het identificeren van fouten in de werkstroom en controleert of de gegevens van het ene element correct naar het volgende stromen.

Standaard voert vRealize Orchestrator altijd een werkstroomvalidatie uit wanneer u een werkstroom uitvoert.

Wanneer u een werkstroom valideert, maakt de validatietool een lijst met fouten of waarschuwingen. Als u op een fout in de lijst klikt, wordt het werkstroomelement gemarkeerd dat de fout bevat.

Als u de validatietool uitvoert in de werkstroomeditor, biedt de tool voorgestelde snelle correcties voor de fouten die worden gedetecteerd. Voor sommige snelle correcties zijn aanvullende informatie of invoerparameters vereist. Andere snelle correcties lossen de fout voor u op.

De werkstroomvalidatie controleert de gegevensbindingen en verbindingen tussen elementen. Werkstroomvalidatie controleert niet de gegevensverwerking die elk element in de werkstroom uitvoert. Hierdoor kan een geldige werkstroom verkeerd worden uitgevoerd en foutieve resultaten veroorzaken als een functie in een schema-element onjuist is.

Een werkstroom valideren en validatiefouten corrigeren in de vRealize Orchestrator-client

U moet een werkstroom valideren voordat u deze kunt uitvoeren. U kunt alleen validatiefouten corrigeren als u de werkstroom hebt geopend voor bewerken.

Voorwaarden

Controleer of u een volledige werkstroom hebt om te valideren, met schema-elementen die zijn gekoppeld en bindingen die zijn gedefinieerd.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client als beheerder.
- 2 Navigeer naar **Bibliotheek > Werkstromen** en selecteer de werkstroom die u wilt valideren.
- 3 Klik op **Bewerken**.
- 4 Klik op **Valideren** in het bovenste menu.

Als de werkstroom geldig is, wordt er een bevestigingsbericht weergegeven. Als de werkstroom ongeldig is, wordt een lijst met fouten weergegeven.

- 5 Voor een ongeldige werkstroom klikt u op een foutmelding en neemt u de nodige stappen om het probleem op te lossen.

De validatietool markeert het schema-element waarin de fout optreedt door er een rood pictogram aan toe te voegen. Waar mogelijk wordt in de validatietool een actie voor snel corrigeren weergegeven.

- Als u akkoord gaat met de voorgestelde snelle correctie, klikt u erop om die actie uit te voeren.
- Als u het niet eens bent met de voorgestelde actie voor snel correctie, sluit u het dialoogvenster Werkstroomvalidatie en corrigeert u het schema-element handmatig.

Belangrijk Controleer altijd of de correctie die vRealize Orchestrator voorstelt, geschikt is.

Het is bijvoorbeeld mogelijk dat de voorgestelde actie een ongebruikt kenmerk verwijdt, wanneer dat kenmerk mogelijk niet correct is gebonden.

- 6 Herhaal de voorgaande stappen totdat u alle validatiefouten hebt verholpen.

Resultaten

U hebt een werkstroom gevalideerd en de validatiefouten gecorrigeerd.

Wat nu te doen

U kunt de werkstroom uitvoeren.

Scripts voor foutopsporingswerkstroom in de vRealize Orchestrator-client

U kunt fouten in werkstroomuitvoeringen opsporen door onderbrekingspunten in het script van werkstroomitems in te voegen.

Wanneer een onderbrekingspunt is bereikt, hebt u verschillende opties om het foutopsporingsproces voort te zetten. Wanneer u een fout opspoot in een element van het werkstroomschema, kunt u de algemene informatie over de uitvoering van de werkstroom bekijken, de werkstroomvariabelen aanpassen, expressies toevoegen om te bekijken en logboekberichten bekijken.

Opmerking Voer alle foutopsporing voor scripts uit in een niet-productieomgeving.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator-client als beheerder.
- 2 Selecteer een werkstroom in de bibliotheek.
- 3 Open het werkstroomschema, selecteer een werkstroomelement en klik op het tabblad **Scriptverwerking**.
- 4 Als u een onderbrekingspunt wilt invoegen, klikt u op de rode cirkel links van het regelnummer.

Opmerking U kunt alleen onderbrekingspunten in werkstroomelementen met scriptverwerking invoegen.

- 5 Als u de werkstroom in de foutopsporingsmodus wilt uitvoeren, klikt u op **Fouten opsporen**. Als de werkstroom invoerparameters vereist, moet u deze opgeven.

- 6 Wanneer de uitvoering van de werkstroom wordt onderbroken na het bereiken van een onderbrekingspunt, selecteert u een van de beschikbare opties.

Optie	Beschrijving
Doorgaan	Hervat de uitvoering van de werkstroom totdat een andere onderbrekingspunt is bereikt of de uitvoering van de werkstroom is voltooid.
Stap naar	U kunt deze optie gebruiken om toegang te krijgen tot een werkstroomelement. U kunt geen toegang krijgen tot een genest werkstroomelement wanneer u fouten opspoor in een werkstroom in de werkstroomeditor.
Stap over	Slaat het huidige element in het schema over en onderbreekt de uitvoering van de werkstroom voor het volgende element.

Opmerking U kunt het foutopsporingsprogramma opdracht geven om het huidige onderbrekingspunt te negeren door op het onderbrekingspunt te klikken. Hierdoor wordt het onderbrekingspuntsymbool gewijzigd in een groen driehoekje.

- 7 (Optioneel) Voeg op het tabblad **Foutopsporingsprogramma** expressies in om te bekijken. U kunt expressies gebruiken om de voltooiing van specifieke variabelen te volgen.
- 8 (Optioneel) Wijzig de waarden van variabelen op het tabblad **Foutopsporingsprogramma**.

Foutopsporing uitvoeren in werkstromen per schema-element

Als werkstroomontwerper kunt u foutopsporing uitvoeren op afzonderlijke schema-elementen.

Procedure

- 1 Meld u aan bij de vRealize Orchestrator Client.
- 2 Ga naar **Bibliotheek > Werkstromen** en selecteer uw werkstroom.
- 3 Selecteer het tabblad **Schema**.
- 4 Selecteer het werkstroomelement waarvoor u foutopsporing wilt uitvoeren en klik op de knop voor foutopsporing linksboven in het element.

Opmerking Door een onderbrekingspunt toe te voegen aan een schema-element **Werkstroomelement**, kunt u rechtstreeks vanuit de bovenliggende werkstroom foutopsporing uitvoeren op de onderliggende werkstromen. Wanneer het foutopsporingsprogramma het schema-element **Werkstroomelement** bereikt, opent het de schemaweergave van de onderliggende werkstroom.

- 5 Herhaal dit voor alle andere schema-elementen waarvoor u foutopsporing wilt uitvoeren.
- 6 Klik op **Fouten opsporen**

7 Voer de gevraagde invoerparameterwaarden in en klik op **Uitvoeren**.

De uitvoering van de werkstroom begint en wordt onderbroken wanneer het foutopsporingsprogramma een schema-element met een onderbrekingspunt bereikt.

8 Bij een onderbrekingspunt selecteert u een van de volgende opties:

Optie	Beschrijving
Doorgaan	Hervat de uitvoering van de werkstroom totdat een andere onderbrekingspunt is bereikt of de uitvoering van de werkstroom is voltooid.
Stap naar	Stap in de huidige werkstroomfunctie. Als het foutopsporingsprogramma niet dieper in de huidige regel van de functie kan gaan, wordt een Stap over -bewerking uitgevoerd.
Stap over	Het foutopsporingsprogramma gaat door naar de volgende regel van de huidige functie.
Stap afsluiten	Het foutopsporingsprogramma gaat naar de regel die zal worden uitgevoerd wanneer de huidige functie wordt geretourneerd.

9 (Optioneel) Bewerk de waarde van uw werkstroomvariabelen op het tabblad **Variabelen**.

The screenshot shows the VMware vRealize Orchestrator debug console. At the top, there's a toolbar with buttons like 'Debugging', 'Suspended', 'CANCEL RUN', 'ALL RUNS', 'RUN AGAIN', 'DEBUG', and 'EXPORT'. Below this is a workflow diagram with a task named 'My Orchestrator Task'. To the right, the 'Scripting' tab is selected, showing a JavaScript code snippet. At the bottom, the 'Variables' tab is active, displaying a table of variables and their values.

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