

VMware vRealize Orchestrator 8.x プラグイ ンの使用

2021年2月23日

vRealize Orchestrator 8.2

最新の技術ドキュメントは、VMware の Web サイト (<https://docs.vmware.com/jp/>)

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

VMware株式会社
105-0013 東京都港区浜松町 1-30-5
浜松町スクエア 13F
www.vmware.com/jp

Copyright © 2008-2021 VMware, Inc. All rights reserved. [著作権および商標情報](#)。

目次

VMware vRealize Orchestrator プラグインの使用 9

- 1 vRealize Orchestrator プラグインの概要 10**
 - vRealize Orchestrator のアーキテクチャ 10
 - vRealize Orchestrator サーバとともにインストールされるプラグイン 11
 - vRealize Orchestrator API Explorer へのアクセス 14
 - タイムゾーンコード 14

- 2 vRealize Orchestrator プラグインの構成 18**
 - vRealize Orchestrator プラグインを管理 18
 - vRealize Orchestrator プラグインのインストールまたはアップデート 19
 - プラグインの削除 19

- 3 Active Directory プラグインの使用 21**
 - Active Directory プラグインの構成 21
 - Active Directory プラグインのワークフロー ライブラリの使用 22
 - コンピュータ ワークフロー 22
 - 組織単位ワークフロー 22
 - ユーザー ワークフロー 23
 - ユーザー グループ ワークフロー 23
 - Active Directory プラグインのクライアント側のロード バランシング 24

- 4 AMQP プラグインの使用 25**
 - AMQP プラグインの構成 25
 - ブローカの追加 25
 - キューへの登録 26
 - ブローカの更新 27
 - AMQP プラグインのワークフロー ライブラリの使用 27
 - バインドの宣言 28
 - キューの宣言 28
 - Exchange の宣言 29
 - テキスト メッセージの送信 30
 - バインドの削除 31

- 5 Configuration プラグインの使用 32**

- 6 Dynamic Types プラグインの使用 34**
 - Dynamic Types 構成ワークフロー 34

7 HTTP-REST プラグインの使用 36

- HTTP-REST プラグインの構成 36
 - REST ホストの追加 37
 - REST 操作の追加 39
 - REST ホストへのスキーマの追加 39
- REST 操作からの新しいワークフローの生成 40
- REST 操作の呼び出し 40
 - REST 操作の呼び出し 41

8 Library プラグインの使用 42

9 Mail プラグインの使用 44

- Mail プラグイン サンプル ワークフローの使用 44
- デフォルトの SMTP 接続の定義 45

10 Multi-Node プラグインの使用 46

- vRealize Orchestrator Multi-Node プラグインの概要 46
- Multi-Node プラグインの構成 47
 - Orchestrator サーバの追加 47
- プロキシ ワークフローの使用 48
 - 同期プロキシ ワークフロー 48
 - 非同期プロキシ ワークフロー 49
 - リモート実行ワークフロー 50
- Multi-Node プラグイン インベントリの使用 50
 - リモート管理ワークフロー 51
- Multi-Node プラグインの API へのアクセス 51
- マルチノード プラグインの使用事例 52
 - マルチプロキシ アクションの作成 52
 - リモートおよびプロキシ ワークフローのメンテナンス 53
 - ローカル サーバからパッケージを展開 53

11 Net プラグインの使用 55

12 PowerShell プラグインの使用 56

- vRealize Orchestrator PowerShell プラグインの概要 56
- PowerShell プラグイン コンポーネント 57
- WinRM の構成 57
 - HTTP を使用するための WinRM の構成 58
 - HTTPS を使用するための WinRM の構成 59
- Kerberos 認証の構成 60
- PowerShell プラグインの構成 62

- 構成ワークフロー 62
 - PowerShell ホストの追加 62
- PowerShell プラグインの API へのアクセス 63
- PowerShell プラグイン インベントリの使用 64
- PowerShell スクリプトの実行 64
 - PowerShell スクリプトの呼び出し 64
 - 外部スクリプトの呼び出し 64
- アクションの生成 65
 - PowerShell スクリプトからのアクションの生成 65
 - PowerShell コマンドレットのアクションの生成 66
- アクション間の呼び出し結果の受け渡し 67
- PowerShell プラグインによる PowerCLI の統合 67
 - コンバータ ワークフロー 68
- PowerShell の結果の使用 68
- サンプル ワークフロー 69
- 一般的な PowerShell タスクのスクリプトのサンプル 69
- トラブルシューティング 71
 - Kerberos イベント ログの有効化 71
 - Kerberos データベースでサーバが見つからない 72
 - Kerberos チケットを取得できない 72
 - Kerberos 認証が時刻設定の違いにより失敗する 73
 - Kerberos 認証セッション モードが失敗する 73
 - レルムのキー配布センターに到達できない 74
 - デフォルトのレルムが見つからない 74

13 SNMP プラグインの使用 75

- SNMP デバイスの管理 75
 - デバイス管理のワークフロー 75
 - SNMP デバイスの登録 76
- SNMP クエリの管理 77
 - クエリ管理ワークフロー 77
 - SNMP デバイスへのクエリの追加 77
- SNMP トラップ ホストの管理 78
 - トラップ ホスト管理ワークフロー 78
 - vRealize Orchestrator Appliance への SNMP トラップ ポートの追加 78
 - SNMP トラップ ポートの設定 79
- SNMP トラップの受信 80
 - SNMP デバイスでのトラップの待機 80
 - SNMP トラップ ポリシーの設定 80
 - SNMP トラップ ホスト ポリシーの構成 81
 - トラップ ポリシーの編集 82

汎用 SNMP 要求ワークフロー 82

14 SOAP プラグインの使用 84

SOAP プラグインの構成 84

SOAP ホストの追加 85

SOAP 操作からの新しいワークフローの生成 86

カスタム生成したワークフローのテスト 87

SOAP 操作の呼び出し 87

15 SQL プラグインの使用 88

SQL プラグインの構成 88

データベースの追加 89

データベースへのテーブルの追加 90

データベースの更新 90

SQL サンプル ワークフローの実行 91

JDBC URL の生成 91

JDBC 接続のテスト 92

JDBC を使用したテーブルの作成 92

JDBC テーブルへの行の挿入 93

JDBC テーブルから行を選択 93

JDBC テーブルからエントリを削除 94

JDBC テーブルからすべてのエントリを削除 95

JDBC テーブルのドロップ 95

完全な JDBC サイクルの実行 96

SQL 操作の実行 96

テーブルの CRUD ワークフローの生成 96

16 SSH プラグインの使用 98

SSH プラグインの構成 98

SSH ホストの追加 98

SSH プラグイン サンプル ワークフローの実行 99

キー ペアの作成 100

キー ペアのパスフレーズの変更 100

vRealize Orchestrator パブリック キーの SSH ホストへの登録 101

SSH コマンドの実行 101

SSH ホストからのファイルのコピー 102

SSH ホストへのファイルのコピー 103

17 vCenter Server プラグインの使用 104

vCenter Server プラグインの構成 105

vCenter Server インスタンスへの接続の構成 105

vCenter Server プラグインのスクリプト API	107
vCenter Server プラグイン インベントリの使用	107
クエリのパフォーマンスに関する考慮事項	107
vCenter Server プラグインを使用した XPath 式の使用	108
vCenter Server プラグイン例を使用した XPath 式の使用	108
vCenter Server プラグインのワークフロー ライブラリ	109
バッチ ワークフロー	112
クラスタとコンピューティング リソースのワークフロー	112
構成ワークフロー	113
カスタム属性のワークフロー	114
データセンターのワークフロー	114
データストアとファイルのワークフロー	114
データセンター フォルダ管理のワークフロー	115
ホスト フォルダ管理のワークフロー	115
仮想マシン フォルダ管理のワークフロー	115
ゲスト操作ファイルのワークフロー	116
ゲスト操作プロセスのワークフロー	116
ホスト電力管理のワークフロー	116
基本的なホスト管理のワークフロー	117
ホスト登録管理のワークフロー	117
ネットワークのワークフロー	118
分散仮想ポート グループのワークフロー	118
分散仮想スイッチのワークフロー	118
標準仮想スイッチのワークフロー	119
Virtual SAN ネットワークのワークフロー	119
リソース プールのワークフロー	119
ストレージのワークフロー	120
ストレージ DRS のワークフロー	120
ストレージ VSAN のワークフロー	121
基本的な仮想マシン管理のワークフロー	121
クローンのワークフロー	122
リンク クローンのワークフロー	123
Linux カスタマイズ クローンのワークフロー	123
ツール クローンのワークフロー	123
Windows カスタマイズ クローンのワークフロー	124
デバイス管理のワークフロー	125
移動と移行のワークフロー	125
その他のワークフロー	126
電力管理のワークフロー	126
スナップショットのワークフロー	127
VMware Tools のワークフロー	127

18 vCloud Suite API (vAPI) プラグインの使用 129

vCloud Suite API プラグインの構成 129

vCloud Suite API メタモデルのインポート 129

vCloud Suite API エンドポイントの追加 130

vCloud Suite API プラグインの API へのアクセス 131

19 XML プラグインの使用 132

XML プラグイン サンプル ワークフローの実行 132

簡易 XML ドキュメントの作成 133

XML ドキュメントの要素の検索 133

XML ドキュメントの変更 134

XML からのサンプルアドレス帳の作成 135

VMware vRealize Orchestrator プラグインの使用

「VMware vRealize Orchestrator プラグインの使用」では、VMware[®] vRealize Orchestrator にインストールされているプラグインの標準セットの構成および使用に関する情報と手順について説明します。

対象者

この情報は、vSphere 管理者としての経験があり、仮想マシン テクノロジーおよびデータセンターの運用に詳しい方を対象としています。

vRealize Orchestrator プラグインの概要

1

vRealize Orchestrator プラグインを利用すると、外部のテクノロジーおよびアプリケーションにアクセスし、制御することができます。vRealize Orchestrator プラグインで外部テクノロジーを公開することで、オブジェクトと関数をワークフローに組み込み、その外部テクノロジーのオブジェクトに対してワークフローを実行できます。

プラグインを使用してアクセスする外部テクノロジーには、仮想化管理ツール、メール システム、データベース、ディレクトリ サービス、リモート制御インターフェイスなどがあります。

vRealize Orchestrator は事前インストールされたプラグインの標準セットを提供します。この標準セットは VMware vCenter Server API、E メール、認証などのテクノロジーを公開します。また、vRealize Orchestrator のオープン プラグイン アーキテクチャを使用して、他のアプリケーションにアクセスするためのプラグインを開発できます。vRealize Orchestrator はオープン スタンドアロッドを実装して外部システムとの統合を簡素化します。

プラグインの標準セットは vRealize Orchestrator サーバとともに自動的にインストールされます。使用前に一部のプラグイン（vCenter Server プラグインなど）の構成が必要になる場合があります。

プラグインは新しいオブジェクト タイプやメソッドを使用して vRealize Orchestrator のスクリプト エンジンを経験します。また、vRealize Orchestrator やプラグイン テクノロジーのイベントをトリガする外部システムの通知イベントを公開します。プラグインは vRealize Orchestrator Client クライアントの [インベントリ] タブからアクセス可能な JavaScript オブジェクトのインベントリを提供します。各プラグインにはワークフローとアクションのパッケージが含まれています。これらのワークフローとアクションをインベントリ内のオブジェクトに対して実行し、外部システムと統合した Orchestrator で一般的な操作を自動化できます。

この章には、次のトピックが含まれています。

- [vRealize Orchestrator のアーキテクチャ](#)
- [vRealize Orchestrator サーバとともにインストールされるプラグイン](#)
- [vRealize Orchestrator API Explorer へのアクセス](#)
- [タイム ゾーン コード](#)

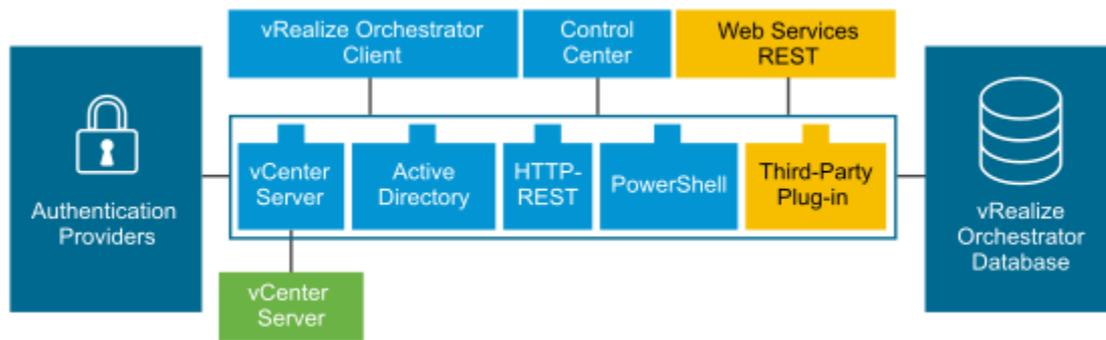
vRealize Orchestrator のアーキテクチャ

vRealize Orchestrator には、オーケストレーション プロセスを自動化するワークフローを作成および実行できる、ワークフロー ライブラリおよびワークフロー エンジンが含まれています。ワークフローは、vRealize Orchestrator が一連のプラグインを介してアクセスするさまざまなテクノロジーのオブジェクトに対して実行されます。

vRealize Orchestrator では、vCenter Server 向けのプラグインを含む、プラグインの標準のセットが提供されており、プラグインで公開されているさまざまな環境においてタスクをオーケストレーションできます。

vRealize Orchestrator では、外部のサードパーティ製アプリケーションをオーケストレーション プラットフォームに接続するためのオープン アーキテクチャも提供されています。自身で定義したプラグイン テクノロジーのオブジェクトに対してワークフローを実行できます。vRealize Orchestrator は認証プロバイダに接続してユーザーアカウントを管理し、事前構成済み PostgreSQL データベースに接続して実行するワークフローからの情報を保存します。vRealize Orchestrator、それによって公開されるオブジェクト、および vRealize Orchestrator ワークフローに、vRealize Orchestrator Client または Web サービスを介してアクセスできます。vRealize Orchestrator ワークフローおよびサービスの監視と構成は、vRealize Orchestrator Client とコントロール センターを介して行われます。

図 1-1. VMware vRealize Orchestrator のアーキテクチャ



vRealize Orchestrator サーバとともにインストールされるプラグイン

vRealize Orchestrator には、一連の標準プラグインが含まれています。各プラグインは外部製品の API を vRealize Orchestrator プラットフォームに公開します。プラグインは、インベントリ クラス、およびスクリプティング エンジン用の追加のオブジェクト タイプを提供します。また、外部システムからの通知イベントを発行します。さらに、統合された外部システムの一般的な使用例を自動化するために、ワークフローのライブラリも提供します。

インストールされているプラグインのリストは、コントロール センターの [プラグインを管理] ページで確認できます。

表 1-1. vRealize Orchestrator と一緒にインストールされたプラグイン

プラグイン	目的	構成
Active Directory	vRealize Orchestrator と Microsoft Active Directory 間の通信を確立します。	Active Directory プラグインの構成 を参照してください。
AMQP	ブローカーとも呼ばれる Advanced Message Queuing Protocol (AMQP) サーバと通信できます。	AMQP プラグインの構成 を参照してください。
構成	vRealize Orchestrator サーバのキーストアおよび信頼された証明書を構成および管理するためのワークフローを提供します。	なし
Dynamic Types	動的タイプを定義し、この動的タイプのオブジェクトを作成して使用できます。	6 章 Dynamic Types プラグインの使用 を参照してください。

表 1-1. vRealize Orchestrator と一緒にインストールされたプラグイン (続き)

プラグイン	目的	構成
Enumeration	他のプラグインがワークフローで使用できる一般的な列挙タイプを提供します。	タイムゾーンコード を参照してください。
HTTP-REST	vRealize Orchestrator と REST ホスト間の通信を介して、REST Web サービスの管理を有効にします。	HTTP-REST プラグインの構成 を参照してください。
ライブラリ	クライアント プロセスのカスタマイズおよび自動化の基本的な構成要素として機能するワークフローを提供します。ワークフローライブラリには、ライフサイクル管理、プロビジョニング、ディザスタリカバリ、ホットバックアップ、および他の標準的なシステム管理プロセスのテンプレートが含まれます。テンプレートのコピーおよび編集を行い、ニーズに応じてテンプレートを変更できます。	なし
メール	簡易メール転送プロトコル (SMTP) を使用してワークフローから電子メールを送信します。	EmailMessage オブジェクトで使用するデフォルト値を設定します。 デフォルトの SMTP 接続の定義 を参照してください。
Multi-Node	階層管理、vRealize Orchestrator インスタンスの管理、および Orchestrator アクティビティのスケールアウトのワークフローが含まれています。	10 章 Multi-Node プラグインの使用 を参照してください。
ネットワーク	Jakarta Apache Commons Net Library を使用します。Telnet、FTP、POP3、および IMAP プロトコルの実装を提供します。POP3 および IMAP プロトコルは、電子メールを読み取るために使用します。Mail プラグインと組み合わせて使用すると、Net プラグインはワークフローに E メール の包括的な送信機能と受信機能を提供します。	
PowerShell	PowerShell ホストを管理し、カスタム PowerShell 操作を実行できます。	12 章 PowerShell プラグインの使用 を参照してください。
SNMP	SNMP 対応システムおよびデバイスに接続して情報を受信できるように vRealize Orchestrator を有効にします。	
SOAP	vRealize Orchestrator と SOAP のホスト間の通信を確立することで、SOAP Web サービスを管理できます。	SOAP プラグインの構成 を参照してください。
SQL	Java Database Connectivity (JDBC) API を提供します。これは、Java プログラミング言語とさまざまなデータベースの間で利用されるデータベースに依存しない接続方法であり、業界標準です。このようなデータベースには、スプレッドシートまたはフラットファイルなど、SQL データベースおよび他の表形式データソースがあります。JDBC API は、ワークフローから SQL ベースのデータベースにアクセスするためのコールレベル API を提供しません。	
SSH	Secure Shell v2 (SSH-2) プロトコルの実装を提供します。ワークフローでパスワードと公開鍵ベースの認証を使用して、リモートコマンドおよびファイル転送セッションを実行できます。キーボード操作による認証をサポートしています。必要に応じて、SSH プラグインにより、vRealize Orchestrator Client インベントリ内を直接参照するリモートファイルシステムを提供できます。	SSH ホストの追加 を参照してください。

表 1-1. vRealize Orchestrator と一緒にインストールされたプラグイン (続き)

プラグイン	目的	構成
vCenter Server	vRealize Orchestrator を使用して自動化する管理プロセスに vCenter Server のすべてのオブジェクトおよび関数を組み込むことができるように、vCenter Server API へのアクセスを提供します。	vCenter Server プラグインの構成 を参照してください。
vCloud Suite API (vAPI)	すべての vAPI プロバイダによって公開される API サービスへのアクセスを提供します。	
XML	ワークフローに実装できる Document Object Model (DOM) XML パーサーです。また、vRealize Orchestrator JavaScript API で ECMAScript for XML (E4X) 実装を使用できます。	

プラグイン コンポーネント

ワークフロー カテゴリや API モジュールなど、各プラグインのコンポーネントは、異なる命名規則を使用します。

表 1-2. プラグイン コンポーネントの名前

構成 UI におけるプラグイン名	ワークフロー カテゴリ	API モジュール
Active Directory	コンピュータ 構成 組織単位 ユーザー ユーザー グループ	AD
AMQP	構成	AMQP
構成	構成	Configurator
動的タイプ	構成	DynamicTypes
一般的な列挙値	なし	Enums
HTTP-REST	構成	REST
ライブラリ	ロック Orchestrator タグ付け	適用外。
メール	メール	Mail
Orchestrator マルチノード	サーバ構成 リモート実行 リモート管理 タスク ワークフロー	VC0
ネットワーク	なし	Net
PowerShell	構成 生成 テンプレート	PowerShell

表 1-2. プラグイン コンポーネントの名前 (続き)

構成 UI におけるプラグイン名	ワークフロー カテゴリ	API モジュール
SNMP	デバイス管理 クエリ管理 トラップ ホスト管理	SNMP
SOAP	構成	SOAP
SQL	JDBC SQL	SQL
SSH	SSH	SSH
サポート	なし	Support
vAPI	VAPI	VAPI
vCenter Server	vCenter	VC
XML	XML	XML

vRealize Orchestrator API Explorer へのアクセス

vRealize Orchestrator API Explorer は、vRealize Orchestrator およびすべてのインストール済みプラグインによって公開される JavaScript オブジェクトの製品内リファレンス ガイドとして使用できます。

vRealize Orchestrator ドキュメントのホーム ページでは、vRealize Orchestrator プラグインのスクリプト API のオンライン バージョンを参照できます。

手順

- 1 vRealize Orchestrator クライアントへのログイン。
- 2 [API Explorer] に移動します。

結果

API Explorer が表示されます。API Explorer を使用して vRealize Orchestrator API のすべてのオブジェクトと関数を検索できます。

次のステップ

スクリプト化可能なアイテムやアクションのスクリプトを記述する場合は、vRealize Orchestrator API Explorer をリファレンス ガイドとして使用します。

タイムゾーンコード

ワークフローで一般的な列挙タイプを実装する場合、Enums:MSTimeZone 列挙の値として、タイムゾーン コードを使用できます。

タイムゾーンコード	タイムゾーン名	説明
000	日付変更線標準時	(GMT - 12:00) 国際日付変更線西側
001	サモア標準時	(GMT - 11:00) ミッドウェイ島、サモア
002	ハワイ標準時	(GMT - 10:00) ハワイ
003	アラスカ標準時	(GMT - 09:00) アラスカ
004	太平洋標準時	(GMT - 08:00) 太平洋標準時 (米国およびカナダ)、ティファナ
010	山地標準時	(GMT - 07:00) 山地標準時 (米国およびカナダ)
013	メキシコ標準時 2	(GMT - 07:00) チワワ、ラパス、マサトラン
015	米国山地標準時	(GMT - 07:00) アリゾナ
020	中部標準時	(GMT - 06:00) 中部標準時 (米国およびカナダ)
025	カナダ中部標準時	(GMT - 06:00) サスカチュワン
030	メキシコ標準時	(GMT - 06:00) グアダハラ、メキシコシティ、モンテレー
033	中央アメリカ標準時	(GMT - 06:00) 中央アメリカ
035	東部標準時	(GMT - 05:00) 東部標準時 (米国およびカナダ)
040	米国東部標準時	(GMT - 05:00) インディアナ (東部)
045	南アメリカ太平洋標準時	(GMT - 05:00) ボゴタ、リマ、キト
050	大西洋標準時	(GMT - 04:00) 大西洋標準時 (カナダ)
055	南アメリカ西部標準時	(GMT - 04:00) カラカス、ラパス
056	太平洋南アメリカ標準時	(GMT - 04:00) サンティアゴ
060	ニューファンドランド・ラブラドール標準時	(GMT - 03:30) ニューファンドランド、ラブラドール
065	東南アメリカ標準時	(GMT - 03:00) ブラジリア
070	南アメリカ東部標準時	(GMT - 03:00) ブエノスアイレス、ジョージタウン
073	グリーンランド標準時	(GMT - 03:00) グリーンランド
075	中央大西洋標準時	(GMT - 02:00) 中央大西洋
080	アゾレス標準時	(GMT - 01:00) アゾレス諸島
083	カーボベルデ標準時	(GMT - 01:00) カーボベルデ諸島
085	GMT 標準時	(GMT) グリニッジ標準時: ダブリン、エジンバラ、リスボン、ロンドン
090	グリニッジ標準時	(GMT) カサブランカ、モンロビア
095	中央ヨーロッパ標準時	(GMT + 01:00) ベオグラード、プラチスラバ、ブダペスト、リュブリャナ、プラハ
100	中央ヨーロッパ標準時	(GMT + 01:00) サラエボ、スコピエ、ワルシャワ、ザグレブ
105	ロマンス標準時	(GMT + 01:00) ブリュッセル、コペンハーゲン、マドリッド、パリ

タイムゾーンコード	タイムゾーン名	説明
110	西ヨーロッパ標準時	(GMT + 01:00) アムステルダム、ベルリン、ベルン、ローマ、ストックホルム、ウィーン
113	西中央アフリカ標準時	(GMT + 01:00) 西中央アフリカ
115	東ヨーロッパ標準時	(GMT + 02:00) ブカレスト
120	エジプト標準時	(GMT + 02:00) カイロ
125	FLE 標準時	(GMT + 02:00) ヘルシンキ、キエフ、リガ、ソフィア、タリン、ビリニユス
130	GTB 標準時	(GMT + 02:00) アテネ、イスタンブール、ミンスク
135	イスラエル標準時	(GMT + 02:00) エルサレム
140	南アフリカ標準時	(GMT + 02:00) ハラーレ、プレトリア
145	ロシア標準時	(GMT + 03:00) モスクワ、サンクトペテルブルク、ボルゴグラード
150	アラブ標準時	(GMT + 03:00) クウェート、リヤド
155	東アフリカ標準時	(GMT + 03:00) ナイロビ
158	アラビック標準時	(GMT + 03:00) バグダッド
160	イラン標準時	(GMT + 03:30) テヘラン
165	アラビア標準時	(GMT + 04:00) アブダビ、マスカット
170	コーカサス標準時	(GMT + 04:00) バクー、トビリシ、エレバン
175	アフガニスタン・イスラム共和国標準時	(GMT + 04:30) カブール
180	エカテリンブルク標準時	(GMT + 05:00) エカテリンブルク
185	西アジア標準時	(GMT + 05:00) イスラマバード、カラチ、タシケント
190	インド標準時	(GMT + 05:30) チェンナイ、コルカタ、ムンバイ、ニューデリー
193	ネパール標準時	(GMT + 05:45) カトマンズ
195	中央アジア標準時	(GMT + 06:00) アスタナ、ダッカ
200	スリランカ標準時	(GMT + 06:00) スリ ジャヤワルダナプラ
201	北中央アジア標準時	(GMT + 06:00) アルマトイ、ノボシビルスク
203	ミャンマー標準時	(GMT + 06:30) ヤンゴン (ラングーン)
205	南東アジア標準時	(GMT + 07:00) バンコク、ハノイ、ジャカルタ
207	北アジア標準時	(GMT + 07:00) クラスノヤルスク
210	中国標準時	(GMT + 08:00) 北京、重慶、香港 SAR、ウルムチ
215	シンガポール標準時	(GMT + 08:00) クアラルンプール、シンガポール
220	台北標準時	(GMT + 08:00) 台北
225	西オーストラリア標準時	(GMT + 08:00) パース
227	北アジア東部標準時	(GMT + 08:00) イルクーツク、ウランバートル
230	韓国標準時	(GMT + 09:00) ソウル

タイムゾーンコード	タイムゾーン名	説明
235	東京標準時	(GMT + 09:00) 大阪、札幌、東京
240	ヤクーツク標準時	(GMT + 09:00) ヤクーツク
245	オーストラリア中央標準時	(GMT + 09:30) ダーウィン
250	中央オーストラリア標準時	(GMT + 09:30) アデレード
255	オーストラリア東部標準時	(GMT + 10:00) キャンベラ、メルボルン、シドニー
260	東オーストラリア標準時	(GMT + 10:00) ブ리즈ベン
265	タスマニア標準時	(GMT + 10:00) ホバート
270	ウラジオストク標準時	(GMT + 10:00) ウラジオストク
275	西太平洋標準時	(GMT + 10:00) グアム、ポートモレスビー
280	中央太平洋標準時	(GMT + 11:00) マガダン、ソロモン諸島、ニューカレドニア
285	フィジー諸島標準時	(GMT + 12:00) フィジー諸島、カムチャツカ、マーシャル諸島
290	ニュージーランド標準時	(GMT + 12:00) オークランド、ウェリントン
300	トンガ標準時	(GMT + 13:00) ヌクアロファ

vRealize Orchestrator プラグインの構成

2

vRealize Orchestrator Appliance では、デフォルトのプラグインの事前インストール済みライブラリにアクセスできます。デフォルトの vRealize Orchestrator プラグインは、vRealize Orchestrator Client でプラグイン固有のワークフローを実行することによって構成されます。

デフォルトの vRealize Orchestrator プラグインには、構成ワークフローが含まれます。vRealize Orchestrator Client からこれらのワークフローを実行して、管理用のエンドポイントを登録できます。

構成ワークフローには、*Configuration* タグがあります。たとえば、AMQP ブローカおよびサブスクリプションの管理に使用されるワークフローにアクセスするには、ワークフロー ライブラリの検索テキスト ボックスに *AMQP* タグと *Configuration* タグを入力します。

この章には、次のトピックが含まれています。

- [vRealize Orchestrator プラグインを管理](#)
- [vRealize Orchestrator プラグインのインストールまたはアップデート](#)
- [プラグインの削除](#)

vRealize Orchestrator プラグインを管理

vRealize Orchestrator コントロール センターの [プラグインを管理] 画面で、vRealize Orchestrator にインストールされているすべてのプラグインのリストを表示して、基本的な管理アクションを実行できます。

プラグインのインストールまたはアップグレード

vRealize Orchestrator プラグインでは、vRealize Orchestrator サーバは他のソフトウェア製品と統合することができます。vRealize Orchestrator には、一連の事前にインストールされたデフォルト プラグインが付属しています。カスタムのプラグインをインストールして、vRealize Orchestrator プラットフォームの機能をさらに拡張することができます。

プラグインのインストールまたはアップグレードは、vRealize Orchestrator の [プラグインを管理] 画面から実行できます。使用できるファイル拡張子は、.vmoapp です。

vRealize Orchestrator プラグインのインストールまたはアップグレードの詳細については、[vRealize Orchestrator プラグインのインストールまたはアップデート](#)を参照してください。

プラグインのログ レベルの変更

vRealize Orchestrator のログ レベルを変更する代わりに、特定のプラグインのみに合わせて変更することができます。

プラグインの無効化

プラグインの名前の横にある [プラグインを有効化] オプションの選択を解除すると、プラグインを無効にできます。このアクションでは、プラグイン ファイルは削除されません。vRealize Orchestrator のプラグインのアンインストールについては、[プラグインの削除](#)を参照してください。

vRealize Orchestrator プラグインのインストールまたはアップデート

vRealize Orchestrator コントロール センターを使用して、サードパーティ プラグインをインストールまたはアップデートできます。

前提条件

vRealize Orchestrator プラグインの `.vmoapp` ファイルをダウンロードします。

手順

- 1 コントロール センターに root としてログインします。
- 2 [プラグインを管理] 画面を選択します。
- 3 [参照] をクリックし、インストールまたはアップデートするプラグインの `.vmoapp` ファイルを選択します。
- 4 [アップロード] をクリックします。
- 5 プラグイン情報を確認します。該当する場合は、エンドユーザー使用許諾契約書に同意し、[インストール] をクリックします。

プラグインがインストールまたは更新され、vRealize Orchestrator サーバ サービスが再起動されます。

次のステップ

正しいプラグイン情報が [プラグインを管理] 画面に一覧表示されていることを確認します。

プラグインの削除

vRealize Orchestrator コントロール センターを介して、サードパーティ製プラグインを削除できます。

注： プラグインを削除すると、プラグイン パッケージが削除されます。

手順

- 1 コントロール センターに root としてログインします。
- 2 [プラグインの管理] を選択します。

- 3 削除するプラグインを検索して、削除アイコン () をクリックします。
- 4 プラグインを削除することを確認し、[削除] をクリックします。

結果

vRealize Orchestrator Appliance からプラグインが削除されました。

Active Directory プラグインの使用

3

VMware vRealize Orchestrator プラグイン (Microsoft Active Directory 用のプラグイン) を使用すると、vRealize Orchestrator と Microsoft Active Directory 間の通信が可能になります。また、このプラグインを使用すると、Active Directory プロセスを自動化する vRealize Orchestrator ワークフローを実行できます。

Active Directory プラグインには標準ワークフローのセットが含まれています。Active Directory 環境のタスクを自動化するプラグイン API を実装するためのカスタム ワークフローを作成することもできます。

この章には、次のトピックが含まれています。

- [Active Directory プラグインの構成](#)
- [Active Directory プラグインのワークフロー ライブラリの使用](#)
- [Active Directory プラグインのクライアント側のロード バランシング](#)

Active Directory プラグインの構成

Active Directory プラグインを使用して Microsoft Active Directory インスタンスに接続するには、Microsoft Active Directory インスタンスの接続パラメータを設定する必要があります。

Active Directory を構成するには、プラグインに組み込まれている構成ワークフローを実行します。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **active_directory** および **configuration** タグを入力します。

ワークフロー名	説明
Active Directory サーバの追加	新しい Active Directory ドメイン構成を追加します。
Active Directory プラグイン オプションの構成	Active Directory プラグインの検索制限オプションを構成します。
Active Directory サーバの更新	Active Directory サーバの既存の構成を変更します。
Active Directory サーバの削除	Active Directory サーバの構成を削除します。
Active Directory サーバの構成 (非推奨)	Active Directory サーバのデフォルト構成を作成または更新します。[Active Directory サーバの更新] を使用します。
構成のリセット (非推奨)	Active Directory サーバのデフォルト構成を削除します。[Active Directory サーバの削除] を使用します。

Active Directory プラグインのワークフロー ライブラリの使用

Active Directory プラグインのワークフロー ライブラリには、Microsoft Active Directory オブジェクトの管理に関連する自動プロセスの実行を可能にするワークフローが含まれています。

Active Directory プラグイン インベントリの使用

Active Directory プラグインは、接続されている Microsoft Active Directory インスタンスのすべてのオブジェクトを [インベントリ] ビューに公開します。

Active Directory インベントリ オブジェクトに使用できるワークフローを表示するには、vRealize Orchestrator クライアントで [管理] - [インベントリ] - [Active Directory] の順に移動します。

Active Directory プラグインのワークフロー

Active Directory プラグインには一連の標準ワークフローが含まれています。これらの標準ワークフローは、もっとも一般的な LDAP 機能に対応しています。ワークフローをビルディング ブロックとして使用すると、複雑なカスタム ソリューションを作成できます。標準ワークフローを結合すると、Active Directory 環境でマルチステップのプロセスを自動化できます。

Active Directory プラグインのコンピュータ ワークフロー

「コンピュータ」ワークフロー カテゴリには、Active Directory のコンピュータ管理に関連するワークフローが含まれています。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **active_directory** および **computer** タグを入力します。

ワークフロー名	説明
グループのコンピュータの作成	グループ内に Active Directory コンピュータを作成します。
組織単位のコンピュータの作成	組織単位内に Active Directory コンピュータを作成します。
コンピュータの削除	Active Directory インスタンスからコンピュータを削除します。
コンピュータとそのサブツリーの削除	Active Directory インスタンスからコンピュータを削除し、そのコンピュータのサブツリーに含まれるすべてのオブジェクトを削除します。
コンピュータの無効化	Active Directory インスタンスでコンピュータを無効にします。
コンピュータの有効化	Active Directory インスタンスでコンピュータを有効にします。

Active Directory プラグインの組織単位ワークフロー

「組織単位」ワークフロー カテゴリには、Active Directory の組織単位管理に関連するワークフローが含まれています。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **active_directory** および **organizational_unit** タグを入力します。

ワークフロー名	説明
組織単位の作成	既存の組織単位の中に組織単位を作成します。
組織単位の破棄	Active Directory インスタンスから組織単位を削除します。
組織単位の破棄とサブツリーの削除	Active Directory インスタンスから組織単位を削除し、その組織単位サブツリー内のすべてのインスタンスとオブジェクトを削除します。

Active Directory プラグインのユーザー ワークフロー

「ユーザー」ワークフロー カテゴリには、Active Directory のユーザー管理に関連するワークフローが含まれます。これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **active_directory** および **user** タグを入力します。

ワークフロー名	説明
ユーザー グループへのユーザーの追加	1人のユーザーをユーザー グループのメンバーとして追加します。
ユーザー パスワードの変更	ユーザーのパスワードを変更します。SSL 接続が必要です。また、パスワードは Active Directory の制限事項を満たしている必要があります。
グループのユーザーの作成	パスワードなしでユーザーを作成します。パスワードは次のログイン時に作成する必要があります。ユーザーによる空のパスワードの使用がドメイン ポリシーで許可されている必要があります。
組織単位のユーザーの作成	組織単位にユーザーを作成します。SSL 接続が無効になっている場合、そのユーザーのパスワードを作成することはできません。ユーザーによる空のパスワードの使用がドメイン ポリシーで許可されている必要があります。
パスワードを使用したグループのユーザーの作成	ユーザーを作成し、そのユーザーのパスワードを設定します。パスワードは次のログイン時に変更することができます。
パスワードを使用した組織単位のユーザーの作成	組織単位にユーザーを作成し、そのユーザーにパスワードを設定します。パスワードは次のログイン時に変更することができます。SSL 接続が無効になっている場合、パスワードを指定することはできません。
ユーザーの削除	Active Directory インスタンスからユーザーを削除します。
ユーザーの無効化	Active Directory インスタンスでユーザーを無効にします。
ユーザーの有効化	Active Directory インスタンスでユーザーを有効にします。
ユーザー グループからのユーザーの削除	ユーザーをユーザー グループから削除します。

Active Directory プラグインのユーザー グループ ワークフロー

「ユーザー グループ」ワークフロー カテゴリには、Active Directory のユーザー グループ管理に関連するワークフローが含まれています。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **active_directory** および **user_group** タグを入力します。

ワークフロー名	説明
グループ メンバーへのコンピュータの追加	1台以上のコンピュータをユーザー グループのメンバーとして追加します。
グループ メンバーへのグループの追加	1つ以上のユーザー グループをユーザー グループのメンバーとして追加します。
グループ メンバーへのユーザーの追加	1人以上のユーザーをユーザー グループのメンバーとして追加します。

ワークフロー名	説明
グループのユーザー グループの作成	既存のコンテナ（グループ）にユーザー グループを作成します。
グループのユーザー グループの作成および属性「Group name (pre-Windows 2000)」の設定	既存のコンテナ（組織単位）にユーザー グループを作成し、Group name (pre-Windows 2000) 属性を設定します。
組織単位のユーザー グループの作成	既存のコンテナ（組織単位）にユーザー グループを作成します。
ユーザー グループの削除	Active Directory インスタンスからユーザー グループを削除します。
グループ メンバーからのコンピュータの削除	1台以上のコンピュータをユーザー グループから削除します。
グループ メンバーからのグループの削除	1つ以上のユーザー グループをユーザー グループから削除します。
グループ メンバーからのユーザーの削除	1人以上のユーザーをユーザー グループから削除します。

Active Directory プラグインのクライアント側のロード バランシング

クライアント側のロード バランシングとフェイルオーバーを使用して、Active Directory プラグインの構成の安定性を向上させることができます。

[Active Directory サーバの追加] および [Active Directory サーバの更新] ワークフローを実行するときにクライアント側のロード バランシングを構成できます。クライアント側のロード バランシングは、ServerSet Java クラスを介して実行できます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **active_directory** および **configuration** タグを入力します。
- 3 [Active Directory サーバの追加] または [Active Directory サーバの更新] ワークフローを実行します。
- 4 [代替ホスト] タブを選択します。
- 5 ドロップダウン メニューから、[単一サーバ]、[ラウンドロビン DNS サーバ]、[ラウンドロビン]、または [フェイルオーバー] を選択します。

オプション	説明
単一サーバ	1台のサーバにのみ接続するサーバ セットの実装。
ラウンドロビン DNS サーバ	特定のホスト名が複数の IP アドレスに解決される可能性がある場合にサーバが処理するサーバ セット。このサーバ セットは、DNS サーバのセットアップを厳密に要求します。アドレスを選択するための順序のメカニズムは、ラウンドロビンです。
ラウンドロビン	複数のディレクトリ サーバ間で負荷を均等に分散するサーバ セット。あるサーバが使用できない場合、接続はセット内の次のサーバに移動します。
フェイルオーバー	サーバ接続が順番に確立されるサーバ セット。この実装では、個別のサーバ セット間で接続を確立できます。複雑な環境で高可用性を実現する場合に便利です。

- 6 ワークフロー実行の構成が完了したら、[実行] をクリックします。

AMQP プラグインの使用

4

AMQP プラグインを使用すると、ブローカとも呼ばれる Advanced Message Queuing Protocol (AMQP) サーバと通信できます。構成ワークフローを実行して AMQP ブローカとキュー登録をインベントリ オブジェクトとして定義し、定義したオブジェクトに対して AMQP 操作を実行できます。

このプラグインには AMQP ブローカの管理と AMQP 操作の呼び出しに関連する標準ワークフローのセットが含まれています。

この章には、次のトピックが含まれています。

- [AMQP プラグインの構成](#)
- [AMQP プラグインのワークフロー ライブラリの使用](#)

AMQP プラグインの構成

AMQP プラグインを構成するには、vRealize Orchestrator クライアントを使用する必要があります。

AMQP を構成するには、プラグインに組み込まれている構成ワークフローを実行します。「構成」ワークフロー カテゴリには、AMQP ブローカを管理できるワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **amqp** および **configuration** タグを入力します。

ワークフロー名	説明
ブローカの追加	AMQP ブローカを追加します。
ブローカの削除	AMQP ブローカを削除します。
サブスクリプションの削除	AMQP メッセージ サブスクリプションを削除します。
キューへの登録	サブスクリプション要素を作成します。
ブローカの更新	ブローカ プロパティを更新します。
ブローカの検証	接続の開始を試みることでブローカを検証します。

ブローカの追加

ワークフローを実行して、AMQP ブローカを追加することができます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** および **configuration** タグを入力します。
- 3 [ブローカの追加] ワークフローを見つけて、[実行] をクリックします。
- 4 [AMQP ブローカのプロパティ] タブで、ブローカの名前を入力します。
- 5 [AMQP 接続プロパティ] タブで、ブローカ接続に必要な情報を入力します。

オプション	アクション
ホスト	ホストのアドレスを入力します。
ポート	AMQP ブローカ サービスのポートを入力します。デフォルト ポートは 5672 です。
仮想ホスト	仮想ホストのアドレスを入力します。表示されるデフォルト値は / です。
SSL を使用	SSL 証明書を使用するかどうかを選択します。
すべての証明書を承諾	すべての SSL 証明書を検証せずに承諾するかどうかを選択します。
ユーザー名	ブローカのユーザー名を入力します。
パスワード	ブローカのパスワードを入力します。

- 6 [実行] をクリックします。

結果

ワークフローが正常に実行されたら、AMQP ブローカが [インベントリ] ビューに表示されます。

次のステップ

[ブローカの検証] ワークフローを実行できます。エラーが発生する場合は、「ブローカの更新」ワークフローを使用して、ブローカのプロパティを変更してから、再度検証を行います。

キューへの登録

ワークフローを実行して、新しいサブスクリプション要素を作成することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから AMQP ブローカの接続があることを確認します。
- AMQP ブローカで、サブスクリプションに含まれているすべてのキューが宣言されていることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** および **configuration** タグを入力します。
- 2 [キューに登録] ワークフローを見つけて、[実行] をクリックします。

- 3 [サブスクリプション] タブで、表示するキューの名前を入力します。
- 4 [AMQP ブローカ] タブで、サブスクリプションを追加するブローカを選択します。
- 5 [キュー] タブで、メッセージ サブスクリプションのすべてのキューを選択します。
- 6 [実行] をクリックします。

結果

ワークフローが正常に実行された後、ブローカの子が [インベントリ] ビューに表示されます。

次のステップ

ポリシーを作成できます。

ブローカの更新

ワークフローを実行して、ブローカのプロパティを更新することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから AMQP ブローカの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** および **configuration** タグを入力します。
- 2 [ブローカの更新] ワークフローを見つけて、[実行] をクリックします。
- 3 [AMQP ブローカ] タブで、更新するブローカを選択します。
ブローカの現在のプロパティは、[新しい AMQP 接続プロパティ] タブに表示されます。
- 4 [新しい AMQP 接続プロパティ] タブで、必要なプロパティを編集します。
- 5 [実行] をクリックします。

AMQP プラグインのワークフロー ライブラリの使用

[AMQP] ワークフロー カテゴリには、AMQP 操作の実行を可能にするワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **amqp** タグを入力します。

ワークフロー名	説明
バインド	特定のブローカのバインドを作成します。
キューの宣言	特定のブローカにキューを追加します。
Exchange の宣言	特定のブローカに Exchange を追加します。
キューの削除	特定のブローカからキューを削除します。

ワークフロー名	説明
Exchange の削除	特定のブローカから Exchange を削除します。
テキスト メッセージの受信	特定のブローカからテキスト メッセージを受信します。
テスト メッセージの送信	特定のブローカを使用してテキスト メッセージを送信します。
バインド解除	特定のブローカのバインドを解除します。

バインドの宣言

ワークフローを実行して、指定したブローカのバインドを作成することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから AMQP ブローカの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** タグを入力します。
- 2 [バインド] ワークフローを見つけて、[実行] をクリックします。
- 3 [AMQP ブローカ] タブで、バインドを作成するブローカを選択します。
- 4 [バインドのプロパティ] タブで、バインドに関する情報を指定します。

オプション	アクション
キュー名	キューの名前を入力します。
Exchange 名	Exchange の名前を入力します。
ルーティング キー	ルーティング キーを入力します。

- 5 [実行] をクリックします。

キューの宣言

ワークフローを実行して、指定したブローカにキューを追加することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから AMQP ブローカの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** タグを入力します。
- 2 [キューの宣言] ワークフローを見つけて、[実行] をクリックします。
- 3 [AMQP ブローカ] タブで、キューの追加先のブローカを選択します。

4 [キューのプロパティ] タブで、キューのプロパティを定義します。

- a [名前] テキスト ボックスで、表示するキューの名前を入力します。
- b キューの持続性の有無を選択します。

オプション	説明
はい	キューはブローカの再起動後に削除されます。
いいえ	キューはブローカの再起動後も保持されます。

- c 特定のキューに占有クライアントを設定するかどうかを選択します。

オプション	説明
はい	この特定のキューに1つのクライアントを設定します。
いいえ	この特定のキューに複数のクライアントを設定します。

- d サブスクリプションを有効化した後にキューを自動で削除するかどうかを選択します。

オプション	説明
はい	キューに接続されているクライアントが存在しなくなった時点で、キューを自動的に削除します。1つ以上のクライアントがキューをサブスクライブしている場合は、キューがそのまま保持されます。
いいえ	キューを削除しません。

5 [実行] をクリックします。

Exchange の宣言

ワークフローを実行して、特定のブローカの Exchange を追加することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから AMQP ブローカの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** タグを入力します。
- 2 [Exchange の宣言] ワークフローを見つけて、[実行] をクリックします。
- 3 [AMQP ブローカ] タブで、Exchange の追加先のブローカを選択します。

4 [Exchange のプロパティ] タブで、Exchange のプロパティを定義します。

- a [名前] テキスト ボックスで、表示するキューの名前を入力します。
- b Exchange のタイプを選択します。

オプション	説明
直接	メッセージのルーティング キーと、キューがこの Exchange にバインドされた場合に使用するルーティング基準の直接一致を行います。
ファンアウト	この Exchange に送信されたすべてのメッセージを、Exchange にバインドされたすべてのキューに転送します。この Exchange にバインドされたキューに引数は含まれません。
ヘッダー	この Exchange にバインドされるキューには、ヘッダーと値を含む引数のテーブルが付属します。x-match という名前の特殊引数により、マッチング アルゴリズムが決まります。
トピック	バインドで指定されたルーティング キーとルーティング パターン間でワイルドカード一致を実行します。

- c Exchange の持続性の有無を選択します。

オプション	説明
はい	Exchange はブローカの再起動後も残ります。
いいえ	Exchange はブローカの再起動後に削除されます。

- d Exchange をサブスクリプション有効化後に自動で削除するかどうかを選択します。

オプション	説明
はい	それ以上キューがバインドされていない場合は、Exchange を自動で削除します。Exchange は、少なくとも 1 つのキューがバインドされていれば残ります。
いいえ	Exchange を削除しません。

5 [実行] をクリックします。

テキスト メッセージの送信

特定のブローカを使用してテキスト メッセージを送信するワークフローを実行できます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから AMQP ブローカの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** タグを入力します。
- 2 [テキスト メッセージの送信] ワークフローを見つけ、[実行] をクリックします。
- 3 [AMQP ブローカ] タブで、メッセージの送信元となるブローカを選択します。

- 4 [Exchange] タブで、Exchange 名とルーティング キーを指定します。
- 5 [メッセージ] タブで、送信するメッセージを入力します。
- 6 [実行] をクリックします。

バインドの削除

ワークフローを実行して、特定のブローカのバインドを削除することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから AMQP ブローカの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **amqp** タグを入力します。
- 2 [バインド解除] ワークフローを見つけて、[実行] をクリックします。
- 3 [AMQP ブローカ] タブで、バインド解除するブローカを選択します。
- 4 [バインドのプロパティ] タブで、キューの名前、Exchange の名前、およびルーティング キーを入力します。
- 5 [実行] をクリックします。

Configuration プラグインの使用

5

コントロール センターを使用して vRealize Orchestrator を構成できるだけでなく、構成プラグインからワークフローを実行することにより、vRealize Orchestrator サーバ構成の設定を変更することができます。

Configuration プラグインを使用すると、vRealize Orchestrator サーバのキーストアと信頼性のある証明書を構成、管理できます。

SSL トラスト マネージャ ワークフロー

「SSL トラスト マネージャ」カテゴリには、SSL 証明書の削除およびインポートに使用できるワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **configuration** および **ssl_trust_manager** タグを入力します。

ワークフロー名	説明
信頼された証明書の削除	サーバのトラスト ストアから SSL 証明書を削除します。
URL から証明書をインポート	URL からサーバのトラスト ストアへ SSL 証明書をインポートします。
URL から認証プロキシ サーバを使用して証明書をインポート	認証プロキシ サーバ経由でアクセスできる URL から SSL 証明書をインポートします。
URL からプロキシ サーバを使用して証明書をインポート	プロキシ サーバ経由でアクセスできる URL から SSL 証明書をインポートします。
URL から証明書別名を持つ証明書をインポート	URL からサーバのトラスト ストアへ SSL 証明書をインポートします。
信頼された証明書をファイルからインポート	ファイルからサーバのトラスト ストアへ SSL 証明書をインポートします。

キーストア ワークフロー

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **configuration** および **keystores** タグを入力します。

ワークフロー名	説明
証明書の追加	証明書をキーストアに追加します。
キーの追加	キーを追加します。

ワークフロー名	説明
キーストアの作成	新しいキーストアを作成します。
キーストアの削除	キーストアを削除します。
証明書の削除	キーストアから証明書を削除します。
エントリの削除	エントリを削除します。
キーの削除	キーを削除します。

Dynamic Types プラグインの使用

6

vRealize Orchestrator Dynamic Types プラグインを使用すると、動的タイプを定義し、これらのタイプのオブジェクトを作成して、オブジェクト間の関係を設定できます。Dynamic Types プラグインを使用して、vRealize Orchestrator スクリプト API でサードパーティ オブジェクトをカスタム タイプとして公開できます。

動的タイプの定義には、プロパティの説明のほか、このタイプの動的オブジェクトを見つけるためのファインダ ワークフローとアクションのセットが含まれています。動的タイプのランタイム インスタンスは動的オブジェクトと呼ばれます。作成した動的オブジェクトに対してワークフローを実行し、さまざまな操作を実行できます。

個々の動的タイプは名前空間で定義する必要があります。名前空間はヘルパー動的オブジェクトで、動的タイプをコンテナごとにグループ化できるようにします。

- 1 Dynamic Types プラグインで「名前空間の定義」および「タイプの定義」ワークフローを実行し、新しい動的タイプとそのプロパティを定義します。この結果、新しい動的タイプのオブジェクトを見つけるためのファインダ ワークフローとインベントリ ワークフローのセットとともに、これらの他のオブジェクトとの関係が取得されます。
- 2 ファインダ ワークフローとインベントリ ワークフローを変更し、サードパーティの REST API からの入力を受信するようにします。
 - a HTTP-REST プラグインで「REST 操作の追加」ワークフローを使用して REST 操作を作成し、これらの操作を対応する REST API メソッドにマップします。
 - b ファインダ ワークフローとインベントリ ワークフローを変更し、これらの REST 操作を呼び出して出力を使用します。

この章には、次のトピックが含まれています。

- [Dynamic Types 構成ワークフロー](#)

Dynamic Types 構成ワークフロー

Dynamic Types プラグインの構成パッケージのワークフローでは、Dynamic Types の作成、XSD ファイルからのタイプの定義のエクスポートとインポート、ならびに作成した動的タイプ間の関係を定義できます。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **dynamic_types** タグを入力します。

ワークフロー名	説明
名前空間の定義	新しい名前空間を定義します。
関係の定義	タイプ間の新しい関係を定義します。
タイプの定義	指定した名前空間内に新しいタイプを定義します。
設定をパッケージとしてエクスポート	Dynamic Types 定義の設定をファイルベースの設定としてエクスポートします。エクスポートしたパッケージは他のサーバにインポートできます。
設定をパッケージからインポート	ファイルベースの設定をプラグイン設定にインポートします。
XSD からタイプの定義をインポート	タイプの定義を XSD ファイルからインポートします。
名前空間の削除	名前空間を削除します。
関係の削除	関係を削除します。
タイプの削除	タイプを削除します。
名前空間の更新	名前空間を更新します。
タイプの更新	タイプを更新します。

HTTP-REST プラグインの使用

7

HTTP-REST プラグインを使用すると、vRealize Orchestrator と REST ホスト間の通信を確立することで、REST Web サービスを管理できます。構成ワークフローを実行して REST サービスとその操作をインベントリ オブジェクトとして定義し、定義したオブジェクトに対して REST 操作を実行できます。

このプラグインには REST ホストの管理と REST 操作の呼び出しに関連する標準ワークフローのセットが含まれています。カスタム ワークフローを生成して REST 環境のタスクを自動化することもできます。

この章には、次のトピックが含まれています。

- [HTTP-REST プラグインの構成](#)
- [REST 操作からの新しいワークフローの生成](#)
- [REST 操作の呼び出し](#)

HTTP-REST プラグインの構成

HTTP-REST を構成するには、プラグインに組み込まれている構成ワークフローを実行します。「構成」ワークフロー カテゴリには、REST ホストの管理を支援するワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **http-rest** および **configuration** タグを入力します。

ワークフロー名	説明
REST ホストの追加	REST ホストをプラグイン インベントリに追加します。
文字列の Swagger 仕様を基に REST ホストを追加	文字列で指定された Swagger 仕様の Web リソースに基づいて、REST ホストを追加します。
URL から取得した Swagger 仕様を基に REST ホストを追加	特定の URL から取得できる Swagger 仕様に基づいて、REST ホストを追加します。
REST 操作の追加	REST ホストに操作を追加します。
REST ホストへのスキーマの追加	REST ホストに XSD スキーマを追加します。
REST ホストのクローン作成	REST ホストのクローンを作成します。
REST 操作のクローン作成	REST 操作のクローンを作成します。
プラグイン構成の再読み込み	プラグイン インベントリ内の REST ホストのリストを更新します。

ワークフロー名	説明
REST ホストの削除	プラグイン インベントリから REST ホストを削除します。
REST 操作の削除	REST ホストから操作を削除します。
REST ホストからのスキーマの削除	関連付けられているすべての XSD スキーマを REST ホストから削除します。
REST ホストの更新	プラグイン インベントリ内の REST ホストを更新します。
REST 操作の更新	REST ホストで操作を更新します。

REST ホストの追加

ワークフローを実行して REST ホストを追加し、ホストの接続パラメータを設定することができます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **http-rest** および **configuration** タグを入力します。
- 3 [REST ホストの追加] ワークフローを見つけて、[実行] をクリックします。
- 4 [ホストのプロパティ] タブで、新しいホストのプロパティを入力します。
 - a [名前] テキスト ボックスにホストの名前を入力します。
 - b [URL] テキスト ボックスにホストのアドレスを入力します。

注： Kerberos 認証では、完全修飾ドメイン名 (FQDN) のホスト アドレスが必要です。

- c [接続タイムアウト] テキスト ボックスに、接続がタイムアウトするまでの秒数を入力します。
- d [操作タイムアウト] テキスト ボックスに、操作がタイムアウトするまでの秒数を入力します。
- e [はい] を選択して、REST ホスト証明書を受け入れます。

vRealize Orchestrator サーバのトラスト ストアに証明書が追加されます。

- 5 [ホスト認証] タブで、認証タイプを選択します。

オプション	説明
なし	認証は要求されません。
OAuth 1.0	[OAuth 1.0] タブで、必要な認証パラメータを指定します。
OAuth 2.0	[OAuth 2.0] タブで、認証トークンを指定します。
基本	基本アクセス認証を指定します。 [ユーザー認証情報] タブで、セッション モードを選択します。 <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。

オプション	説明
ダイジェスト	<p>暗号化を使用するダイジェスト アクセス認証を指定します。</p> <p>[ユーザー認証情報] タブで、セッション モードを選択します。</p> <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。
NTLM	<p>Window セキュリティ サポート プロバイダ (SSPI) フレームワーク内の NT LAN Manager (NTLM) アクセス認証を指定します。</p> <p>[ユーザー認証情報] タブで、セッション モードを選択します。</p> <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。 <p>[NTLM] タブで、NTLM 設定を指定します。</p>
Kerberos	<p>Kerberos アクセス認証を指定します。</p> <p>[ユーザー認証情報] タブで、セッション モードを選択します。</p> <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。

- 6 (オプション) [プロキシ設定] タブで、プロキシ サーバを使用するかどうかを選択します。
 - a プロキシ サーバのアドレスとポートを入力します。
 - b プロキシの認証タイプを選択します。

オプション	説明
なし	認証は要求されません。
基本	<p>基本アクセス認証を指定します。</p> <p>[プロキシ認証情報] タブで、セッション モードを選択します。</p> <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。

- 7 [SSL] タブで、ターゲットのホスト名をサーバ証明書に保存されている名前と一致させるかどうかを選択します。
- 8 (オプション) サーバに対する認証に使用するキーストア エントリを選択します。キーストア エントリは PrivateKeyEntry タイプである必要があります。
- 9 [実行] をクリックします。

結果

ワークフローが正常に実行されたら、REST ホストが [インベントリ] ビューに表示されます。

次のステップ

REST ホストに操作と XSD スキーマを追加でき、[インベントリ] ビューからワークフローを実行できます。

REST 操作の追加

ワークフローを実行して、プラグイン インベントリから REST ホストに操作を追加することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから REST ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **http-rest** および **configuration** タグを入力します。
- 2 [REST 操作の追加] ワークフローを見つけて、[実行] をクリックします。
- 3 操作を追加する先の親ホストを選択します。
- 4 [名前] テキスト ボックスに操作の名前を入力します。
- 5 [テンプレート URL] テキスト ボックスに、URL の操作部分のみを入力します。
操作の実行時に指定されるパラメータ用のプレースホルダを含めることができます。
以下に、URL 構文の例を示します。

`/customer/{id}/orders?date={date}`

- 6 操作で使用する HTTP メソッドを選択します。
[POST] または [PUT] を選択した場合は、メソッドにコンテンツタイプの要求ヘッダーを指定できます。
- 7 [実行] をクリックします。

次のステップ

操作に対してワークフローを実行するには、[インベントリ] ビューを使用します。

REST ホストへのスキーマの追加

ワークフローを実行して、プラグイン インベントリから REST ホストに XSD スキーマを追加することができます。

XSD スキーマは、Web サービスとの間で入出力コンテンツとして使用される XML ドキュメントを記述します。このようなスキーマをホストに関連付けることによって、REST 操作からワークフローを生成する際に入力として求められる XML 要素を指定できます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから REST ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **http-rest** および **configuration** タグを入力します。

- 2 [REST ホストにスキーマを追加] ワークフローを見つけて、[実行] をクリックします。
- 3 [ホスト] タブで、XSD スキーマを追加する先のホストを選択します。
- 4 [XSD スキーマの詳細] タブで、URL からスキーマをロードするかどうかを選択します。

オプション	アクション
はい	スキーマの URL を入力します。
いいえ	スキーマのコンテンツを指定します。

- 5 [実行] をクリックします。

REST 操作からの新しいワークフローの生成

REST 操作からカスタム ワークフローを生成することができます。

カスタム生成したワークフローをハイレベル ワークフローに統合することができます。ワークフロー開発の詳細については、『vRealize Orchestrator を使用したワークフローの開発』ガイドを参照してください。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから REST ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **http-rest** タグを入力します。
- 2 [REST 操作からの新しいワークフローの生成] ワークフローを見つけて、[実行] をクリックします。
- 3 使用可能な操作のリストから REST 操作を選択します。
操作に入力がある場合や、ホストに XSD スキーマが追加される場合は、要求される入力タイプを指定します。
- 4 [名前] テキスト ボックスに生成するワークフローの名前を入力します。
- 5 新しいワークフローを生成するワークフロー フォルダの名前を選択します。
ワークフロー ライブラリの既存のフォルダを選択することができます。
- 6 [実行] をクリックします。

REST 操作の呼び出し

REST 操作の直接呼び出し

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから REST ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **http-rest** および **configuration** タグを入力します。
- 2 [REST 操作の呼び出し] ワークフローを見つけて、[実行] をクリックします。
- 3 [操作] タブで、使用可能な操作のリストから REST 操作を選択します。
- 4 操作に必要な入力パラメータを指定します。
- 5 [実行] をクリックします。

REST 操作の呼び出し

REST 要求を行うには、構成済みの REST 操作を呼び出すか、または構成済みの REST 操作をテンプレートとして使用し実行時にパラメータを置き換えて REST 操作を動的に呼び出します。

REST 操作を呼び出すには、いくつかの方法があります。

- REST ホストを構成し REST 操作を関連付けるには、[REST ホストの追加] および [REST 操作の追加] ワークフローを実行します。登録された REST ホストおよび REST 操作はパーシステントとなり、[インベントリ] ビューや [リソース] ビューに表示されます。
- 事前に REST ホストを設定したり、[REST 操作の呼び出し] ワークフローを [ライブラリ] - [ワークフロー] から実行したりすることなく、動的な REST 操作を呼び出します。このワークフローでは、REST ホストベースの URL と操作パラメータを指定できます。データはパーシステントではなく、[インベントリ] ビューや [リソース] ビューには表示されません。
- [動的パラメータによる REST ホストの呼び出し] および [動的パラメータによる REST 操作の呼び出し] ワークフローを [ライブラリ] - [ワークフロー] から実行して、REST ホストを構成し、REST 操作を関連付けて、構成済みの REST ホストと REST 操作をテンプレートとして今後使用できるようにします。ワークフロー実行時、構成済みの REST ホストや REST 操作のパラメータを置き換えることができます。元の REST ホストや REST 操作に影響はありません。

Library プラグインの使用

8

Library プラグインのワークフローをクライアント プロセスのカスタマイズおよび自動化のテンプレートとして使用して、vRealize Orchestrator のトラブルシューティングを実行できます。Library プラグインは、[ロック]、[Orchestrator]、[タグ付け] ワークフロー カテゴリに属するワークフローを提供します。

ロック ワークフロー

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **locking** タグを入力します。

ワークフロー名	説明
すべてのロックの表示	すべてのロックを表示します。
ロック テスト	ロックを作成するテスト ワークフロー。
ロック テスト (x 5)	5 つのロックを作成するテスト ワークフロー。
すべてのロックの解除	すべてのロックを解除します。

タスク ワークフロー

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **tasks** タグを入力します。

ワークフロー名	説明
循環タスクの作成	循環タスクを作成して新しく作成したタスクを返します。
タスクの作成	ワークフローを実行する日時をスケジュール設定してタスクとして実行します。

Orchestrator ワークフロー

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **orchestrator** および **workflows** タグを入力します。

ワークフロー名	説明
待機状態にあるワークフローの実行の状態を再表示	指定したリモート サーバで待機状態にあるすべてのワークフローの実行を処理し、リモート ワークフローの実行に応じてワークフローの状態をアップデートします。このワークフローは、たとえば vRealize Orchestrator サーバ間の接続が切断されたときなどの、ワークフローの実行間でデータが失われた場合に使用できます。
ワークフローの順次開始	ワークフローを複数回にわたって実行し、1つのインスタンスから次のインスタンスへと順次処理します。ワークフローのパラメータは配列で指定します。また、実行を開始するワークフローの各インスタンスに対して、プロパティ リスト（ワークフロー入力あたり1つのプロパティ）も指定します。配列内のプロパティ数によってワークフローの実行数が決まります。
ワークフローの同時開始	ワークフローを複数回にわたって異なるパラメータを使用して実行します。ワークフローのパラメータは配列で指定します。また、実行を開始するワークフローの各インスタンスに対して、プロパティ リスト（ワークフロー入力あたり1つのプロパティ）も指定します。配列内のプロパティ数によってワークフローの実行数が決まります。

タグ付けワークフロー

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **tagging** タグを入力します。

ワークフロー名	説明
タグによるオブジェクトの検索	オブジェクトに割り当てられているタグに基づいてオブジェクトを検索します。タグの名前と値を指定すると、これらのタグが適用されているオブジェクトのリストをワークフローが返します。
ワークフロー タグの一覧表示	入力パラメータとして指定したワークフローに割り当てられているタグを一覧表示します。
ワークフローのタグ付け	ワークフローにタグを割り当てます。タグ付けするワークフロー、タグ名、値を指定する必要があります。
タグ付け例	ワークフローのタグ付けをデモンストレーションします。
ワークフローのタグの解除	ワークフローからタグを削除します。タグを解除するワークフローと、ワークフローから削除するタグを指定する必要があります。

Mail プラグインの使用

9

Mail プラグインを使用すると、ワークフローからメール メッセージを送信できます。Mail プラグインは簡易メール 転送プロトコル (SMTP) を使用します。たとえば、ワークフローの実行が完了した場合、またはユーザー操作が必要になる場合に、指定のアドレスに E メールを送信するワークフローを作成できます。

この章には、次のトピックが含まれています。

- [Mail プラグイン サンプル ワークフローの使用](#)
- [デフォルトの SMTP 接続の定義](#)

Mail プラグイン サンプル ワークフローの使用

カスタム ワークフローから Mail プラグインのサンプル ワークフローを呼び出して、メール機能をカスタム ワークフローに実装することができます。サンプル ワークフローを実行して、vRealize Orchestrator と SMTP サーバ間の操作をテストできます。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **mail** タグを入力します。

注： ワークフローにアクセスする前に、ログイン時に使用したユーザー アカウントに Mail ワークフローの実行に必要な権限があることを確認します。

ワークフロー名	説明
メールの設定	SMTP サーバへの接続、SMTP 認証アカウント、ならびに送信者のアドレスと表示名を設定します。
メッセージの取得	POP3 プロトコルを使用して、指定のメール アカウントのメッセージを取得します。
メッセージの取得 (メールクライアント経由)	MailClient クラスで提供される新しいスクリプティング API を使用して、特定のメール アカウントのメッセージを削除せずに取得します。
通知を送信	指定のメール アドレスに指定の内容の E メールを送信します。オプションのパラメータが指定されていない場合、ワークフローはメールの設定ワークフローで設定されたデフォルトの値を使用します。
メーリングリストに通知を送信	特定のメール アドレス リスト、CC リストおよび BCC リストに指定の内容の E メールを送信します。オプションのパラメータが指定されていない場合、ワークフローはメールの設定ワークフローで設定されたデフォルトの値を使用します。

デフォルトの SMTP 接続の定義

Mail プラグインは、vRealize Orchestrator サーバとともにインストールされ、メール通知の送受信に使用されます。SMTP サーバに対してメール通知の送受信を認証できるデフォルトのメール アカウントを設定できます。

注： vRealize Orchestrator でメールを設定するときは、ロード バランサを回避してください。そうしないと、SMTP_HOST_UNREACHABLE エラーが発生することがあります。

手順

- 1 vRealize Orchestrator クライアントへのログイン。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **mail** タグを入力します。
- 3 [メールの設定] ワークフローを見つけて、[実行] をクリックします。
- 4 [ホスト] タブで、SMTP サーバの IP アドレスまたはドメイン名、および SMTP 構成と一致するポート番号を入力します。

デフォルトの SMTP ポートは 25 です。
- 5 [認証情報] タブで、認証のためのユーザー名とパスワードを入力します。

有効な E メール アカウントと、関連付けられているパスワードを入力します。vRealize Orchestrator は E メール アカウントを使用して E メールを送信します。
- 6 [E メールの内容] タブで、送信者のメール アドレスと名前を入力します。

vRealize Orchestrator によって送信されるすべての E メールに表示される送信者の情報を入力します。
- 7 [実行] をクリックします。

Multi-Node プラグインの使用

10

Multi-Node プラグイン ワークフロー ライブラリには、階層オーケストレーション、vRealize Orchestrator インスタンスの管理、および vRealize Orchestrator アクティビティのスケールアウトを行うためのワークフローが含まれています。

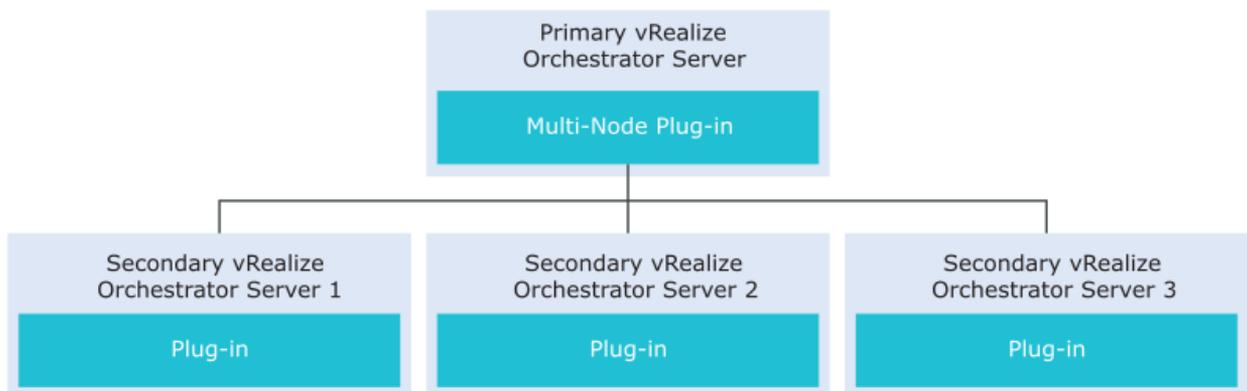
この章には、次のトピックが含まれています。

- [vRealize Orchestrator Multi-Node プラグインの概要](#)
- [Multi-Node プラグインの構成](#)
- [プロキシ ワークフローの使用](#)
- [Multi-Node プラグイン インベントリの使用](#)
- [Multi-Node プラグインの API へのアクセス](#)
- [マルチノード プラグインの使用事例](#)

vRealize Orchestrator Multi-Node プラグインの概要

Multi-Node プラグインは、vRealize Orchestrator サーバ間にプライマリ/セカンダリ関係を作成します。この関係は、パッケージ管理とワークフローの実行にも拡大されます。

図 10-1. マルチノード プラグイン スキーマ



プラグインには、階層オーケストレーション、vRealize Orchestrator インスタンスの管理、および vRealize Orchestrator アクティビティのスケールアウトのワークフローが含まれています。

Multi-Node プラグインの構成

Multi-Node プラグインを構成するには、vRealize Orchestrator クライアントを使用する必要があります。

「サーバ構成」ワークフロー カテゴリには、vRealize Orchestrator サーバの接続を可能にするワークフローが含まれています。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **orchestrator** および **servers_configuration** タグを入力します。

ワークフロー名	説明
Orchestrator サーバの追加	vRealize Orchestrator サーバをプラグイン インベントリに追加します。
Orchestrator サーバの削除	vRealize Orchestrator サーバをプラグイン インベントリから削除し、そのサーバのために作成されたすべてのプロキシを削除します。
Orchestrator サーバの更新	詳細を変更して、プラグイン インベントリから vRealize Orchestrator サーバを更新します。

Orchestrator サーバの追加

ワークフローを実行して、新しい vRealize Orchestrator サーバへの接続を確立することができます。

前提条件

マスターおよびスレーブの vRealize Orchestrator サーバのバージョンが同じであることを検証します。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **orchestrator** および **servers_configuration** タグを入力します。
- 3 [Orchestrator サーバの追加] ワークフローを見つけて、[実行] をクリックします。
- 4 [サーバの詳細] タブで、リモート vRealize Orchestrator サーバのホスト アドレスとポートを指定します。
 - a 証明書が警告なしで受け入れられ、信頼済みストアに追加されるかを選択します。
 - b リモート vRealize Orchestrator サーバのプロキシ ワークフローを生成するかを選択します。
- 5 [接続設定] タブで、接続設定を指定します。
 - a [接続タイムアウト] テキスト ボックスに、vRealize Orchestrator から リモート サーバへの接続がタイムアウトするまでの秒数を入力します。
 - b [ソケット タイムアウト] テキスト ボックスに、要求の正常な処理がタイムアウトするまでの秒数を入力します。
 - c [再試行タイムアウト] テキストボックスに、接続がない場合に、プロキシ ワークフローがリモート vRealize Orchestrator サーバからの通知を受信するまでの秒数を入力します。

6 [接続モード] タブで、接続が共有されるかを選択します。

オプション	説明
いいえ	リモートの vRealize Orchestrator サーバへの接続に、ログインしたユーザーの認証情報が使用されます。
はい	すべてのユーザーが同じ認証情報を使用して、リモートの Orchestrator サーバにアクセスできます。共有接続用の認証情報を指定します。

7 [実行] をクリックします。

プロキシ ワークフローの使用

プロキシ ワークフローを使用すると、ローカルの vRealize Orchestrator サーバと、リモートの vRealize Orchestrator サーバ上のワークフロー間の通信を管理できます。

Multi-Node プラグインを使用すると、リモート ワークフローと通信するローカル ワークフローを生成できます。これらのローカル ワークフローは、プロキシ ワークフローと呼ばれます。プロキシ ワークフローは Multi-Node プラグインのインベントリから入力パラメータを取得します。プロキシ ワークフローを実行すると、リモート ワークフローに必要なタイプにパラメータが変換されます。リモート ワークフローの実行が完了すると、出力パラメータは変換されて vRealize Orchestrator サーバ上でローカル表現に戻ります。

同期プロキシ ワークフロー

同期タイプのプロキシ ワークフローでは、リモート ワークフローの API と操作コントラクトが保持されます。

すべての同期プロキシ ワークフローのスキーマは同じですが、含まれるスクリプトは異なります。



同期プロキシ ワークフローが完了するのは、リモート ワークフローが完了して出力パラメータが指定された後になります。

リモート ワークフローの結果を待機する間、ローカル ワークフローはサーバ リソースを消費しません。

実行が正常に完了すると、プロキシ ワークフローの出力パラメータには、リモート ワークフロー トークンがローカル表現で指定されます。ブール値、数値、文字列など、単純なタイプの出カパラメータは、vRealize Orchestrator サーバ上の他のワークフローで直接使用することができます。

非同期プロキシ ワークフロー

非同期プロキシ ワークフローを使用して、リモート ワークフローの実行を最適化することができます。

すべての非同期プロキシ ワークフローのスキーマは同じですが、含まれるスクリプトは異なります。



非同期プロキシ ワークフローは結果を即時に返します。返される結果は、リモート ワークフロー トークン オブジェクトのローカル ラッパーです。プロキシ ワークフローはこのトークンを使用して、実行の状態をチェックしたり、リモート ワークフローの実行完了時に出力パラメータを取得したりします。ブール値、数値、文字列など、単純なタイプの出力パラメータは、vRealize Orchestrator サーバ上の他のワークフローで直接使用することができます。

リモート実行ワークフロー

「リモート実行」ワークフロー カテゴリには、プロキシ ワークフローを管理できるワークフローが含まれています。

リモート実行標準ワークフロー

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **orchestrator** および **remote_execution** タグを入力します。

ワークフロー名	説明
複数のプロキシ アクションの作成	プロキシ アクションを複数作成して複数のサーバでワークフローを実行します。
プロキシ ワークフローの作成	リモートの Orchestrator サーバでのワークフローの開始に使用できるプロキシ ワークフローを作成します。
フォルダからプロキシ ワークフローを作成	リモートの Orchestrator サーバ上のフォルダ内のすべてのワークフローにプロキシ ワークフローを作成します。

サーバ プロキシ

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **orchestrator**、**remote_execution** および **server_proxies** タグを入力します。

ワークフロー名	説明
Orchestrator サーバのプロキシ ワークフローを作成	リモート サーバの構造をミラーリングして、ローカル Orchestrator サーバにプロキシ ワークフローを作成します。
Orchestrator サーバのプロキシ ワークフローを削除	ローカルの Orchestrator サーバのプロキシ ワークフローを削除し、生成されたすべてのワークフローを消去します。
Orchestrator サーバのプロキシ ワークフローを更新	リモート サーバからローカルの Orchestrator サーバにすべてのプロキシ ワークフローを再生成します。

Multi-Node プラグイン インベントリの使用

Multi-Node プラグインは、接続されている vRealize Orchestrator サーバのすべてのインベントリを [インベントリ] ビューにミラーします。

単一のリモート サーバのインベントリは、システム オブジェクトとプラグイン オブジェクトという 2 つの主要パーツで構成されています。これらのオブジェクトはどちらもリモート オブジェクトをローカルで使用可能なタイプに変換するラッパーです。

システム オブジェクト

システム オブジェクトは [システム] と呼ばれるトップレベルのグループにあります。これらには構成、パッケージ、ワークフロー、アクション、関連フォルダが含まれています。リモート システム オブジェクトには個々のラッパー タイプがあります。

プラグイン オブジェクト

プラグイン オブジェクトはリモートの vRealize Orchestrator サーバに接続されているすべてのプラグインのインベントリをミラーします。リモート プラグイン オブジェクトはすべて単一のローカル タイプ [VCO:RemotePluginObject] にラップされます。

リモート管理ワークフロー

「リモート管理」ワークフロー カテゴリには、リモートの vRealize Orchestrator インスタンス上でパッケージおよびワークフローの管理を可能にするワークフローが含まれています。

リモート管理パッケージ

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **orchestrator**、**remote_management** および **packages** タグを入力します。

ワークフロー名	説明
パッケージの削除	リモートの vRealize Orchestrator サーバからパッケージとその内容を削除します。
名前によるパッケージの削除	リモートの vRealize Orchestrator サーバ上で名前に基づいてパッケージとその内容を削除します。
ローカル サーバからパッケージを展開	ローカルの vRealize Orchestrator サーバからリモートの Orchestrator サーバにパッケージを展開します。
リモート サーバからパッケージを展開	リモートの vRealize Orchestrator サーバから複数のリモート Orchestrator サーバにパッケージを展開します。
ローカル サーバからパッケージを展開	ローカルの vRealize Orchestrator サーバからリモートの vRealize Orchestrator サーバにパッケージを展開します。

リモート管理ワークフロー

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **orchestrator**、**remote_management** および **workflows** タグを入力します。

ワークフロー名	説明
リモート ワークフローの削除	リモートの vRealize Orchestrator サーバからワークフローを削除します。
終了したワークフローの実行をすべて削除	リモート ワークフローから終了したワークフローの実行をすべて削除します。
ローカル サーバからワークフローを展開	ローカルの vRealize Orchestrator サーバから複数のリモート Orchestrator サーバにワークフローを展開します。
リモート サーバからワークフローを展開	リモートの vRealize Orchestrator サーバから複数のリモート Orchestrator サーバにワークフローを展開します。

Multi-Node プラグインの API へのアクセス

Orchestrator に用意されている API Explorer を使用すると、Multi-Node プラグインの API を検索したり、スク립ト化された要素で使用できる JavaScript オブジェクトのドキュメントを参照したりすることができます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 vRealize Orchestrator クライアントまたはワークフロー、ポリシー、アクション エディタの [スクリプティング] タブから API Explorer にアクセスします。
 - vRealize Orchestrator クライアントから API Explorer にアクセスするには、vRealize Orchestrator クライアントのナビゲーション ペインで [API Explorer] をクリックします。
 - ワークフロー、ポリシー、アクション エディタの [スクリプティング] タブから API Explorer にアクセスするには、左側にある [API の検索] をクリックします。
- 3 Multi-Node プラグインの API オブジェクトの階層リストを展開するには、左側のペインで [VCO] モジュールをダブルクリックします。

次のステップ

API 要素からコードをコピーしてスクリプト処理ボックスに貼り付けることができます。API スクリプティングの詳細については、『VMware vRealize Orchestrator における開発』を参照してください。

マルチノード プラグインの使用事例

マルチノード プラグインの使用事例には、リモートおよびプロキシのワークフローについての情報のほかに、ローカルの vRealize Orchestrator サーバからのパッケージのインポート、複数のプロキシ アクションの使用などが含まれます。

マルチプロキシ アクションの作成

「マルチプロキシ アクションの作成」ワークフローを実行して、複数のサーバに対してワークフローを実行することができます。

リモートの vRealize Orchestrator サーバ上で後からワークフローを実行するためのアクションを作成できます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **orchestrator** および **remote_execution** タグを入力します。
- 3 [マルチプロキシ アクションの作成] ワークフローを見つけて、[実行] をクリックします。
- 4 [アクションの詳細] タブにアクション名とモジュールを入力します。

アクション名には英数字のみを使用でき、文字間にスペースを挿入することはできません。

同じ名前のアクションがすでに存在している場合でも、新しいアクションが作成されます。

- 5 [ワークフローの詳細] タブで、ワークフローがローカルまたはリモートのいずれであるかを選択します。

オプション	説明
はい	このアクションに使用するリモート ワークフローを選択します。
いいえ	このアクションに使用するローカル ワークフローを選択します。

- 6 [実行] をクリックします。

結果

生成されたアクションは、ソース ワークフローと同じパラメータを受け入れますが、オブジェクトが複数選択されている場合は、それらのパラメータを1つの配列に変換します。配列内の値にはインデックスが付けられます。

リモートおよびプロキシ ワークフローのメンテナンス

リモートおよびプロキシのワークフローが変更される場合は、プロキシを更新するか、必要でない場合は削除します。メンテナンス用に、Multi-Node プラグインはプロキシおよびリモート ワークフローの情報を更新または削除するワークフローを提供しています。

vRealize Orchestrator クライアントでプロキシ ワークフローを管理するためのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **orchestrator**、**remote_execution**、および **servers_proxies** タグを入力します。

ワークフロー名	説明
Orchestrator サーバのプロキシ ワークフローを更新	リモート サーバからローカルの vRealize Orchestrator サーバにすべてのプロキシ ワークフローを再生成します。
Orchestrator サーバのプロキシ ワークフローを削除	ローカルの vRealize Orchestrator サーバのプロキシ ワークフローを削除し、生成されたすべてのワークフローを消去します。

ワークフローにアクセスして、vRealize Orchestrator クライアントでプロキシワークフローの追加メンテナンスを行うには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **orchestrator**、**remote_management**、および **workflows** タグを入力します。

ワークフロー名	説明
終了したワークフローの実行をすべて削除	リモート ワークフローから終了したワークフローの実行をすべて削除します。
リモート ワークフローの削除	リモートの vRealize Orchestrator サーバからワークフローを削除します。
ローカル サーバからワークフローを展開	ローカルの vRealize Orchestrator サーバから複数のリモート vRealize Orchestrator サーバにワークフローを展開します。

ローカル サーバからパッケージを展開

ワークフローを実行して、ローカルの vRealize Orchestrator サーバからリモートの vRealize Orchestrator サーバにパッケージを展開することができます。

この例では、ローカル サーバからリモート サーバの配列にパッケージを展開します。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **orchestrator** および **remote_management** タグを入力します。
- 3 [ローカル サーバからパッケージを展開] ワークフローを見つけて、[実行] をクリックします。
- 4 展開するパッケージをローカル ストレージから選択します。
- 5 パッケージの展開先となるリモート サーバを選択します。
- 6 リモート サーバのパッケージを上書きするか選択します。

オプション	説明
はい	リモート サーバ上のパッケージを置き換え、パッケージに含まれる要素のバージョンを破棄します。
いいえ	サーバのバージョン確認およびパッケージの展開を実行します。確認が正常に完了した後、パッケージが展開されます。

- 7 [実行] をクリックします。

結果

ワークフローの実行後、ログ ビューおよびプラグインのインベントリにステータス情報が表示されます。

Net プラグインの使用

11

Net プラグインを使用すると、Telnet、FTP、POP3、IMAP プロトコルをワークフローに実装できます。POP3 と IMAP を実装すると、E メールダウンロードと読み取りが可能になります。Mail プラグインと組み合わせて使用すると、Net プラグインは E メール包括的な送信機能と受信機能をワークフローに提供します。

PowerShell プラグインの使用

12

PowerShell プラグインのワークフロー ライブラリには、PowerShell ホストの管理とカスタム PowerShell 操作の実行を可能にするワークフローが含まれています。

vRealize Orchestrator クライアントの [インベントリ] ビューを使用して、使用可能な PowerShell リソースを管理できます。プラグインのスクリプト API を使用すると、カスタム ワークフローを作成できます。

この章には、次のトピックが含まれています。

- [vRealize Orchestrator PowerShell プラグインの概要](#)
- [PowerShell プラグインの構成](#)
- [PowerShell プラグインの API へのアクセス](#)
- [PowerShell プラグイン インベントリの使用](#)
- [PowerShell スクリプトの実行](#)
- [アクションの生成](#)
- [アクション間の呼び出し結果の受け渡し](#)
- [PowerShell プラグインによる PowerCLI の統合](#)
- [PowerShell の結果の使用](#)
- [サンプル ワークフロー](#)
- [一般的な PowerShell タスクのスクリプトのサンプル](#)
- [トラブルシューティング](#)

vRealize Orchestrator PowerShell プラグインの概要

PowerShell プラグインにより、vRealize Orchestrator および Windows PowerShell 間でやりとりができません。

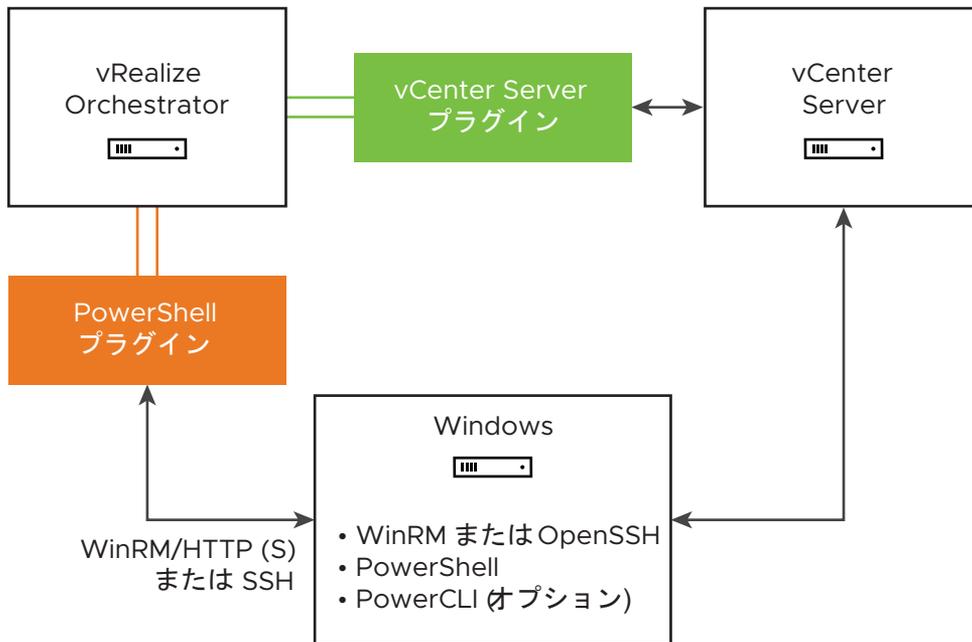
プラグインを使用して vRealize Orchestrator アクションおよびワークフローから PowerShell スクリプトおよびコマンドレットを呼び出したり、結果を使用したりします。このプラグインには標準ワークフローのセットが含まれています。プラグイン API を実装するカスタム ワークフローを作成することもできます。

PowerShell プラグイン コンポーネント

PowerShell プラグインは、適切に機能するためにさまざまなコンポーネントに依存しています。

vRealize Orchestrator および Windows PowerShell では、プラグインのプラットフォームを提供しています。このプラグインによって、製品間のやりとりが可能になります。PowerShell プラグインも vCenter Server や vSphere PowerCLI などの他のコンポーネントと連携できます。

図 12-1. コンポーネントの関係



プラグインは OpenSSH および WinRM 通信プロトコルを介して Windows PowerShell とやりとりできます。[WinRM の構成](#)を参照してください。

オプションで、PowerShell プラグインを vSphere PowerCLI および vCenter Server と統合できます。[PowerShell プラグインによる PowerCLI の統合](#)を参照してください。

注： ローカル ホスト上のすべてのコンポーネントをインストールできます。PowerShell プラグインの用途、機能、および通信プロトコルの要件は、vRealize Orchestrator および Windows PowerShell が同じマシンにインストールされている場合は変更されません。

WinRM の構成

PowerShell プラグインと Windows PowerShell の間の接続を確立するには、サポートされている通信プロトコルのいずれかを使用するように WinRM を構成する必要があります。

PowerShell プラグインでは、Windows リモート管理 (WinRM) 2.0 が管理プロトコルとしてサポートされています。

次の認証方法がサポートされます。

認証方法	詳細
基本	ユーザー名とパスワードを必要とする、セキュアでない認証メカニズム。
Kerberos	チケットを使用してクライアントの ID とサーバを検証するセキュアな認証プロトコル。

注： PowerShell プラグインでは WinRM でのユーザー認証情報の委任はサポートされておらず、CredSSP は認証方法としてサポートされていません。

HTTP 経由の WinRM

PowerShell プラグインでは、HTTP プロトコルを介した WinRM ホストとの通信がサポートされています。WinRM では通信が認証されますが、データ転送は暗号化されず、ネットワーク上でプレーン テキスト形式で送信されます。IPSec が通信を行うマシン間で構成されている場合は、HTTP プロトコルを使用する必要があります。

基本認証を使用するには、サービスおよびクライアントの WinRM 構成で AllowUnencrypted プロパティを **true** に設定する必要があります。HTTP 構成の例については、[HTTP を使用するための WinRM の構成](#)を参照してください。

HTTPS 経由の WinRM

PowerShell プラグインでは、HTTPS プロトコルを介した WinRM ホストとの通信がサポートされています。よりセキュアな通信手段として HTTPS プロトコルを使用することができます。

HTTPS プロトコルを使用するには、サーバ認証用に証明書を生成し、WinRM ホストにインストールする必要があります。HTTPS 構成の例については、[HTTPS を使用するための WinRM の構成](#)を参照してください。

HTTP を使用するための WinRM の構成

WinRM ホストを構成して、HTTP プロトコルを介した PowerShell プラグインとの通信を有効にできます。

WinRM ホスト マシン上でコマンドを実行して、WinRM 構成を変更する必要があります。同一のマシンを WinRM サービスと WinRM クライアントの両方として使用できます。

重要： WinRM を構成して HTTP を使用するとき次のいずれかの手順をスキップすると、ホストは追加されずログに次のようなエラー メッセージが表示される場合があります。

```
Caused by: org.dom4j.DocumentException: Error on line -1 of document : Premature end of file.
Nested exception: Premature end of file.
at org.dom4j.io.SAXReader.read(SAXReader.java:482)
at org.dom4j.DocumentHelper.parseText(DocumentHelper.java:278)
at
com.xebialabs.overthere.cifs.winrm.connector.JdkHttpConnector.sendMessage(JdkHttpConnector.java:117)
```

手順

- 1 次のコマンドを実行して WinRM 構成値をデフォルトに設定します。

```
c:\> winrm quickconfig
```

- 2 (オプション) 次のコマンドを実行してリスナーが実行されているかどうかを確認し、デフォルトのポートを確認します。

```
c:\> winrm e winrm/config/listener
```

デフォルトのポート番号は、5985 (HTTP) および 5986 (HTTPS) です。

- 3 WinRM サービスで基本認証を有効にします。

- a 次のコマンドを実行して基本認証が有効になっているかどうかを確認します。

```
c:\> winrm get winrm/config/service
```

- b 次のコマンドを実行して基本認証を有効にします。

```
c:\> winrm set winrm/config/service/auth @{Basic="true"}
```

- 4 次のコマンドを実行して WinRM サービスで暗号化されていないデータの転送を許可します。

```
c:\> winrm set winrm/config/service @{AllowUnencrypted="true"}
```

- 5 WinRM サービスのチャンネル バインディング トークンのセキュリティ強化レベルが **strict** に設定されている場合は、**relaxed** にその値を変更します。

```
c:\> winrm set winrm/config/service/auth @{CbtHardeningLevel="relaxed"}
```

- 6 WinRM クライアントで基本認証を有効にします。

- a 次のコマンドを実行して基本認証が有効になっているかどうかを確認します。

```
c:\> winrm get winrm/config/client
```

- b 次のコマンドを実行して基本認証を有効にします。

```
c:\> winrm set winrm/config/client/auth @{Basic="true"}
```

- 7 次のコマンドを実行して WinRM クライアントで暗号化されていないデータの転送を許可します。

```
c:\> winrm set winrm/config/client @{AllowUnencrypted="true"}
```

- 8 WinRM ホスト マシンが外部ドメイン内にある場合は、次のコマンドを実行して信頼できるホストを指定します。

```
c:\> winrm set winrm/config/client @{TrustedHosts="host1, host2, host3"}
```

- 9 次のコマンドを実行して WinRM サービスへの接続をテストします。

```
c:\> winrm identify -r:http://winrm_server:5985 -auth:basic -u:user_name -p:password -encoding:utf-8
```

HTTPS を使用するための WinRM の構成

WinRM ホストを構成して、HTTPS プロトコルを介した PowerShell プラグインとの通信を有効にできます。

WinRM ホストで HTTPS プロトコルを介した通信を有効にするには、証明書が必要です。証明書を取得するか、生成します。たとえば、.NET Framework SDK に含まれる証明書作成ツール (makecert.exe) を使用して、自己署名証明書を生成できます。

前提条件

- WinRM を構成して HTTP プロトコルを使用します。詳細については、[HTTP を使用するための WinRM の構成](#)を参照してください。
- WinRM ホストで Microsoft 管理コンソール (mmc.exe) にアクセスできることを確認します。

手順

- 1 自己署名の証明書を生成します。

次のコマンドラインには、WinRM ホスト上で `makecert.exe` を使用して証明書を作成するための構文の例が含まれています。

```
makecert.exe -r -pe -n "CN=host_name-3,0=organization_name" -e mm/dd/yyyy -eku
1.3.6.1.5.5.7.3.1 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA
SChannel Cryptographic Provider" -sy 12 certificate_name.cer
```

- 2 生成された証明書を追加するには、Microsoft 管理コンソールを使用します。

- a `mmc.exe` を実行します。
- b [ファイル] - [スナップインの追加と削除] を選択します。
- c 利用できるスナップインのリストから [証明書] を選択し、[追加] をクリックします。
- d [コンピュータ アカウント] を選択して [次へ] をクリックします。
- e [完了] をクリックします。
- f [コンソール ルート] - [証明書 (ローカル コンピューター)] - [個人] - [証明書]、および [コンソール ルート] - [証明書 (ローカル コンピューター)] - [信頼されたルート証明機関] - [証明書] で証明書がインストールされていることを確認します。

証明書が [信頼されたルート認証局] および [個人] フォルダにインストールされていない場合は、手動でインストールする必要があります。

- 3 適切なサムプリントとホスト名を使用して HTTPS リスナーを作成します。

次のコマンドラインには、HTTPS リスナーを作成するための構文例が含まれています。

```
winrm create winrm/config/Listener?Address=*&Transport=HTTPS
@{Hostname="host_name";CertificateThumbprint="certificate_thumbprint"}
```

注： 証明書サムプリントのスペースは省略します。

- 4 接続をテストします。

次のコマンドラインには、接続をテストするための構文例が含まれています。

```
winrs -r:https://host_name:port_number -u:user_name -p:password hostname
```

Kerberos 認証の構成

PowerShell ホストを追加および管理する場合は、Kerberos 認証を使用できます。

Kerberos 認証では、ドメイン ユーザーは WinRM を経由して、PowerShell が有効なリモート マシン上でコマンドを実行できます。

手順

- 1 PowerShell ホストで WinRM を構成します。

```
winrm quickconfig
winrm set winrm/config/service/auth @{Kerberos="true"}
winrm set winrm/config/service @{AllowUnencrypted="true"}
winrm set winrm/config/winrs @{MaxMemoryPerShellMB="2048"}
```

- 2 /data/vco/usr/lib/vco/app-server/conf/ で krb5.conf ファイルを作成または編集します。

krb5.conf ファイルの構造は次のとおりです。

```
[libdefaults]
default_realm = YOURDOMAIN.COM
[realms]
YOURDOMAIN.COM = {
kdc = dc.yourdomain.com
default_domain = yourdomain.com
}
[domain_realm]
.yourdomain.com=YOURDOMAIN.COM
yourdomain.com=YOURDOMAIN.COM
```

krb5.conf には、特定の構成パラメータとその値を含める必要があります。

Kerberos 構成タグ	詳細
default_realm	クライアントが Active Directory サーバの認証に使用するデフォルトの Kerberos レルム。 注： 大文字にする必要があります。
kdc	キー配布センター (KDC) として機能し、Kerberos チケットを発行するドメイン コントローラ。
default_domain	完全修飾ドメイン名を生成するために使用されるデフォルト ドメイン。 注： このタグは、Kerberos 4 の互換性のために使用されます。

注： デフォルトでは、Java Kerberos の構成に UDP プロトコルを使用します。TCP プロトコルのみを使用するには、値 **1** を使用した udp_preference_limit パラメータを指定する必要があります。

注： Kerberos 認証では、完全修飾ドメイン名 (FQDN) のホスト アドレスが必要です。

重要： krb5.conf ファイルを追加または変更するときには、vRealize Orchestrator サーバ サービスを再起動する必要があります。

- 3 次のコマンドを実行して、権限を変更します。

```
chmod 644 krb5.conf
```

4 vRealize Orchestrator ポッドを再展開します。

```
kubectl -n prelude get pods
```

次のようなエントリを探します。

```
vco-app-<ID>
```

5 ポッドを破棄します。

```
kubectl -n prelude delete pod vco-app-<ID>
```

破棄したポッドを置き換えるために、新しいポッドが自動的に展開されます。

次のステップ

vRealize Orchestrator クライアントで、[PowerShell ホストの追加] ワークフローを実行します。

PowerShell プラグインの構成

PowerShell プラグインを構成するには、vRealize Orchestrator クライアントを使用する必要があります。

構成ワークフロー

「構成」ワークフロー カテゴリには、PowerShell ホストを管理できるワークフローが含まれています。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **powershell** および **configuration** タグを入力します。

ワークフロー名	説明
PowerShell ホストの追加	PowerShell ホストをプラグイン インベントリに追加します。
PowerShell ホストの削除	プラグイン インベントリから PowerShell ホストを削除します。
PowerShell ホストの更新	プラグイン インベントリ内の指定された PowerShell ホストを更新します。
PowerShell ホストの検証	指定された PowerShell ホストの構成を検証します。

PowerShell ホストの追加

ワークフローを実行して、PowerShell ホストの追加およびホストの接続パラメータの構成を行います。リモートまたはローカルの PowerShell ホストへの接続を設定できます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **powershell** および **configuration** タグを入力します。
- 3 PowerShell ホストの追加ワークフローを見つけて、[実行] をクリックします。
- 4 [名前] テキスト ボックスにホストの名前を入力します。

- 5 [ホスト/IP アドレス] テキスト ボックスにホストのアドレスを入力します。

注： Kerberos 認証では、完全修飾ドメイン名 (FQDN) のホスト アドレスが必要です。

- 6 (オプション) [ポート] テキスト ボックスにホストのポートを入力します。

ポート 5985 (HTTP)、または 5986 (HTTPS) を使用します。

- 7 [ホスト タイプ] タブで、プラグインの接続先となる PowerShell のホスト タイプを指定します。

- a 転送プロトコルを選択します。

注： HTTPS 転送プロトコルを使用すると、リモート PowerShell ホストの証明書が Orchestrator キーストアにインポートされます。

- b 認証タイプを選択します。

重要： Kerberos 認証を使用するには、WinRM サービスで有効にする必要があります。

- 8 [認証情報を使用] タブで PowerShell ホストへの接続に使用するセッション モードのタイプを選択します。

オプション	説明
共有セッション	プラグインは共有の認証情報を使用して、リモート ホストに接続します。共有セッション用の PowerShell ホスト認証情報を指定する必要があります。
ユーザーごとのセッション	Orchestrator クライアントは、ログインしているユーザーから認証情報を取得します。 user@domain という形式で Orchestrator にログインして、[ユーザーごとのセッション] モードを使用する必要があります。

- 9 [詳細オプション] タブの [シェル コード ページ] ドロップダウン メニューから、PowerShell が使用するエンコードのタイプを選択します。

- 10 [実行] をクリックします。

結果

ワークフローが正常に実行されると、PowerShell ホストが [インベントリ] ビューに表示されます。

PowerShell プラグインの API へのアクセス

Orchestrator の API Explorer を使用すると、PowerShell プラグイン API の検索や、スクリプト化された要素で使用できる JavaScript オブジェクトのドキュメントの参照を実行できます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 vRealize Orchestrator クライアントまたはワークフロー、ポリシー、アクション エディタの [スクリプティング] タブから API Explorer にアクセスします。
 - vRealize Orchestrator クライアントから API Explorer にアクセスするには、vRealize Orchestrator クライアントのナビゲーション ペインで [API Explorer] をクリックします。

- ワークフロー、ポリシー、アクション エディタの [スクリプティング] タブから API Explorer にアクセスするには、左側にある [API の検索] をクリックします。
- 3 PowerShell プラグインの API オブジェクトの階層リストを展開するには、左側のペインで [PowerShell] モジュールをダブルクリックします。

次のステップ

API 要素からコードをコピーしてスクリプト処理ボックスに貼り付けることができます。API スクリプティングの詳細については、『VMware vRealize Orchestrator における開発』を参照してください。

PowerShell プラグイン インベントリの使用

PowerShell プラグインは、接続されている PowerShell ホストのすべてのオブジェクトを [インベントリ] ビューに公開します。

プラグインのインベントリ内では、PowerShell ホストおよびそのスナップインとコマンドレットを監視できます。各リモート ホストにはスナップインを、各スナップインにはコマンドレットを追加できます。

PowerShell スクリプトの実行

PowerShell ホスト上で、外部スクリプトまたはカスタム スクリプトを呼び出すワークフローを実行できます。

PowerShell スクリプトの呼び出し

プラグイン インベントリのホスト上にある既存またはカスタムの PowerShell スクリプトを実行することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから PowerShell ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **powershell** タグを入力します。
- 2 PowerShell スクリプトの呼び出しワークフローを見つけて、[実行] をクリックします。
- 3 [PowerShell ホスト] タブで、スクリプトを実行する PowerShell ホストを選択します。
- 4 [スクリプト] タブの [スクリプト] テキストボックスで、実行する PowerShell スクリプトを入力または貼り付けます。
- 5 [実行] をクリックします。

外部スクリプトの呼び出し

プラグイン インベントリのホスト上にある外部 PowerShell スクリプトを実行できます。

外部 PowerShell スクリプトは、.ps1 ファイルに含まれています。実行する .ps1 ファイルは、PowerShell ホストに保存する必要があります。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから PowerShell ホストへの接続があることを確認します。
- スクリプトが参照する可能性のある .ps1 ファイルにアクセス権があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **powershell** タグを入力します。
- 2 [外部スクリプトの呼び出し] ワークフローを見つけて、[実行] をクリックします。
- 3 [PowerShell ホスト] タブで、スクリプトを実行する PowerShell ホストを選択します。
- 4 [外部スクリプト] タブの [名前] テキストボックスに、実行する外部 ps1 スクリプトのファイル名を入力します。

注： .ps1 ファイルがデフォルトのフォルダに入っていない場合は、ファイルの絶対パスを入力する必要があります。スクリプト パスの指定には、環境変数を使用できます。たとえば、**\$env:HOME\PATH\test1.ps1** のように指定できます。

- 5 [引数] テキストボックスで、スクリプトの引数を入力します。
有効な構文は PowerShell コンソールで使用されるものと同じです。
- 6 [実行] をクリックします。

アクションの生成

ワークフローを実行して、PowerShell スクリプトまたは PowerShell コマンドレットに基づくアクションを生成できます。生成されたワークフローは、カスタム ワークフローのビルディング ブロックとして使用することができます。

PowerShell スクリプトからのアクションの生成

ワークフローを実行して、指定した PowerShell スクリプトからアクションを生成できます。生成したアクションを実行するサンプル ワークフローを任意で作成することもできます。

生成したアクションのスクリプトをプレースホルダを使用してカスタマイズできます。生成したアクションでは、ワークフローによって string タイプの対応するアクション パラメータがプレースホルダごとに作成されます。アクションを実行する際、実際の値をアクション パラメータに指定して、プレースホルダを置き換えることができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから PowerShell ホストへの接続があることを確認します。

手順

- 1 vRealize Orchestrator クライアントで [ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **powershell** および **generate** タグを入力します。

- 2 [PowerShell スクリプトからのアクションの生成] ワークフローを見つけて、[実行] をクリックします。
- 3 [スクリプト] タブで [スクリプト] テキスト ボックスに、アクションを生成する PowerShell スクリプトを入力または貼り付けます。

注： ユーザー入力のプレースホルダとして **{#ParamName#}** を使用できます。プレースホルダのタイプが string である場合、二重引用符を使用してプレースホルダの値をアクションに渡す必要があります。

次のスクリプトは、生成したアクション パラメータをスクリプト パラメータにリンクする方法の例を示しています。

```
param($name={#ParamName#})
echo $name;
```

- 4 [アクション] タブの [名前] テキストボックスに、生成するアクションの名前を入力して、アクションを生成する既存のモジュールを選択します。
- 5 [ワークフロー] タブで、ワークフローを生成するかどうかを選択します。

オプション	説明
はい	生成したアクションを実行するサンプル ワークフローを生成します。ワークフローの生成先のフォルダを選択する必要があります。 注： 生成したワークフローの名前は、呼び出しスクリプトの事前定義済み文字列と、生成したアクションの名前で構成されます。
いいえ	サンプル ワークフローは生成されません。

- 6 [送信] をクリックして、ワークフローを実行します。

次のステップ

生成したアクションをカスタム ワークフローに統合できます。

PowerShell コマンドレットのアクションの生成

ワークフローを実行して、指定した PowerShell コマンドレットとパラメータ セットのアクションを生成できます。このアクションでは、vRealize Orchestrator の PowerShell 機能を使用できます。生成したアクションを実行するサンプル ワークフローを任意で作成することもできます。

PowerShell スクリプト エンジンを使用して、多数のデータ タイプ セットを使用できます。使用できるデータ タイプには、プリミティブ タイプ (Integer、Boolean、Char、および .NET アセンブリで使用可能なタイプなど) やユーザー定義タイプが含まれます。PowerShell コマンドレットの定義に基づいてアクションを生成する場合、入力および出力コマンドレット パラメータは、vRealize Orchestrator プラットフォームがサポートするタイプで表示されます。PowerShell プラグインはタイプのマッピングを定義します。一般的に、プリミティブ タイプは対応する vRealize Orchestrator タイプにマップされ、複雑なタイプは PowerShellRemotePSObject オブジェクトによって表示されます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。

- [インベントリ] ビューから PowerShell ホストへの接続があることを確認します。

手順

- 1 vRealize Orchestrator クライアントで [ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **powershell** および **generate** タグを入力します。
- 2 [PowerShell コマンドレットからのアクションの生成] ワークフローを見つけて、[実行] をクリックします。
- 3 [コマンドレット] タブで、生成したアクションを使用する際に実行する PowerShell コマンドレットを選択します。
- 4 コマンドレットのパラメータ セットを選択します。

パラメータ セットの定義値が [パラメータ セットの定義] テキスト ボックスに表示されます。

注： [パラメータ セットの定義] テキストボックス内の文字列を編集して、パラメータ セットの定義値を変更することはできません。パラメータ セットに含まれているパラメータ情報の文字列は確認できます。

- 5 [アクション] タブの [名前] テキストボックスに、生成するアクションの名前を入力して、アクションを生成する既存のモジュールを選択します。
- 6 [ワークフロー] タブで、ワークフローを生成するかどうかを選択します。

オプション	説明
はい	生成したアクションを実行するサンプル ワークフローを生成します。ワークフローの生成先のフォルダを選択する必要があります。 注： 生成したワークフローの名前は、実行コマンドレットの事前定義済み文字列と、生成したアクションの名前で構成されます。
いいえ	サンプル ワークフローは生成されません。

- 7 [実行] をクリックします。

次のステップ

生成したアクションをカスタム ワークフローに統合できます。

アクション間の呼び出し結果の受け渡し

PowerShell プラグインでは、1つの PowerShell スクリプト呼び出しから別の呼び出しへの結果をパラメータとして受け渡すことができます。結果を正しく受け渡すには、両方の呼び出しが同一のセッションで行われている必要があります。

PowerShell プラグインによる PowerCLI の統合

VMware vSphere PowerCLI といったサードパーティのスナップインの機能を、PowerShell プラグインで使うことができます。

サードパーティのスナップイン機能を使用するには、PowerShell ホストでスナップインが使用可能になっている必要があります。現在のセッションにスナップインを読み込むには、AddPsSnapin アクションも呼び出す必要があります。PowerCLI を使用するには、スナップインの名前を VMware.VimAutomation.Core に設定する必要があります。

PowerShell プラグインでは、サードパーティのスナップイン用に事前生成されたアクションは提供されません。サードパーティのスナップイン用にアクションを生成するには、「PowerShell コマンドレットにアクションを作成」ワークフローを実行します。[PowerShell コマンドレットのアクションの生成](#) を参照してください。

com.vmware.library.powershell.converter パッケージには、VC:<SomeObjectType> オブジェクトから対応する PowerCLI オブジェクトに変換可能な基本ビルディング ブロックが含まれています。この機能により、vCenter Server プラグインからのワークフローが PowerShell プラグインからのワークフローとやりとりできるようになり、2 つのプラグイン間でのパラメータの受け渡しが可能になります。

コンバータ ワークフロー

「コンバータ」ワークフロー カテゴリのサンプル ワークフローを使用して、PowerShell プラグインと PowerCLI 間の統合をテストすることができます。統合をテストするには、PowerShell ホストに PowerCLI をインストールする必要があります。

「コンバータ」サンプル ワークフローは、プラグインで使用できる変換機能をデモンストレーションします。

注： PowerShell プラグインでは、PowerCLI と vCenter Server プラグインで使用できる一部のタイプがサポートされていません。サポート対象外のタイプは例外を返します。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **powershell** および **converter** タグを入力します。

ワークフロー名	説明
PSObject から vCO オブジェクトへの変換	PowerShellRemotePSObject を VC:<SomeObjectType> に変換します。
PSObject と vCO オブジェクトの双方向変換	PowerShellRemotePSObject から VC:<SomeObjectType> への変換、および逆方向の変換を行います。
vCO オブジェクトから PSObject への変換	VC:<SomeObjectType> を PowerShellRemotePSObject に変換します。

PowerShell の結果の使用

PowerShell プラグイン API のオブジェクトを使用して、Windows PowerShell が返す結果と連携できます。

PowerShellInvocationResult クラスのメソッドを使用して、実行するスクリプトに関する情報を取得できます。

メソッド	説明
getErrors()	スクリプトの起動中に PowerShell エンジンによって報告されるエラーのリストを返します。
getInvocationState()	スクリプトのステータス。可能な値は Completed または Failed です。
getHostOutput()	PowerShell コンソールに表示されるスクリプトの出力。
getResults()	PowerShell エンジンによって返されるオブジェクト。返されるオブジェクトのタイプは PowershellRemotePSObject です。

PowerShellRemotePSObject は PowerShell エンジンが返すオブジェクトのリモート表現です。

PowerShellRemotePSObject には XML シリアル化された結果が含まれており、getXml() メソッドを呼び出すことによりアクセスできます。

PowerShell プラグインは XML 結果をラップするオブジェクト モデルを提供するほか、特定のオブジェクト プロパティへの簡単なアクセスも提供します。getRootObject() メソッドはオブジェクト モデルへのアクセスを提供します。一般的に、getRootObject() メソッドは次のルールを使用して、PowerShell タイプを vRealize Orchestrator で使用できるタイプにマップします。

- 返されたオブジェクトがプリミティブな PowerShell タイプである場合、オブジェクトは対応する Orchestrator のプリミティブなタイプにマッピングされます。
- 返されたオブジェクトのタイプが collection である場合、オブジェクトは ArrayList と表示されます。
- 返されたオブジェクトのタイプが dictionary である場合、オブジェクトは Hashtable と表示されます。
- 返されたオブジェクトのタイプが complex である場合、オブジェクトは PSObject と表示されます。

サンプル ワークフロー

「サンプル」ワークフロー カテゴリには、基本的な使用ケースのテストを可能にするワークフローが含まれています。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **powershell** および **samples** タグを入力します。

ワークフロー名	説明
API 経由のスクリプト呼び出し	使用可能なスクリプト API 経由で PowerShell スクリプトを呼び出す方法を示します。
ディレクトリ内容のリスト表示	PowerShell ホスト ファイル システム上のディレクトリの内容を一覧表示します。
パイプライン実行サンプル	複数の cmdlets をパイプライン実行する方法を示します。
仮想マシンの状態の切り替え	仮想マシンの電源状態を切り替えます。

一般的な PowerShell タスクのスクリプトのサンプル

JavaScript サンプルを切り取り、貼り付け、および編集することで、一般的な PowerShell タスクのスクリプトを記述できます。

スクリプティングの詳細については、『vRealize Orchestrator 開発者ガイド』を参照してください。

例：API 経由での PowerShell スクリプトの実行

JavaScript を使用して、プラグイン API 経由で PowerShell スクリプトを実行できます。

このサンプル スクリプトは、次のアクションを実行します。

- PowerShell ホストのセッションを開きます。
- 実行するスクリプトを指定します。
- 呼び出し結果を確認します。

- セッションを閉じます。

```

var sess;
try {
    //Open session to PowerShell host
    var sess = host.openSession()
    //Set executed script
    var result = sess.invokeScript('dir')

    //Check for errors
    if (result.invocationState == 'Failed'){
        throw "PowerShellInvocationError: Errors found while executing script \n" +
result.getErrors();
    }
    //Show result
    System.log( result.getHostOutput() );
} catch (ex){
    System.error (ex)
} finally {
    if (sess) {
        //Close session
        host.closeSession( sess.getSessionId() );
    }
}

```

例：結果の使用

JavaScript を使用して、PowerShell スクリプトの実行結果を処理することができます。

このサンプル スクリプトは、次のアクションを実行します。

- 呼び出し状態を確認します。
- 結果から値を抽出します。
- RemotePSObject タイプを確認します。

```

var sess = host.openSession()
sess.addCommandFromString("dir " + directory)
var invResult = sess.invokePipeline();
//Show result
System.log( invResult.getHostOutput() );

//Check for errors
if (invResult.invocationState == 'Failed'){
System.error(invResult.getErrors());
} else {
    //Get PowerShellRemotePSObject
    var psObject = invResult.getResults();
    var directories = psObject.getRootObject();

    var isList = directories instanceof Array
    if ( isList ){
        for (idx in directories){
            var item = directories[idx];

```

```

        if ( item instanceof('System.IO.FileInfo') ){//Check type of object
            System.log( item.getProperty('FullName') );//Extract value from result
        }
    }
} else {
    System.log( directories.getProperty('FullName') );//Extract value from result
}
}

host.closeSession( sess.getSessionId());

```

例：カスタムの資格情報と連携する

JavaScript を使用して、PowerShell ホストをカスタム認証情報と連携させることができます。

```

var sess;
try {
    sess = host.openSessionAs(userName, password);

    var invResult = sess.invokeScript('$env:username');

    //Check for errors
    if (invResult.invocationState == 'Failed'){
        System.error(invResult.getErrors());
    } else {
        //Show result
        System.log( invResult.getHostOutput() );
    }
} catch (ex){
    System.error (ex)
} finally {
    if (sess) {
        host.closeSession( sess.getSessionId());
    }
}
}

```

トラブルシューティング

PowerShell プラグインの使用時に問題が発生した場合は、トラブルシューティングのトピックを参照し、回避策があれば問題を解決できます。

Kerberos イベント ログの有効化

トラブルシューティングの目的で、キー配布センター (KDC) マシンで Kerberos イベント ログを有効にする必要がある場合があります。

前提条件

Windows レジストリをバックアップします。

手順

- 1 キー配布センター (KDC) として機能するドメイン コントローラにログインします。

- 2 管理者として、レジストリ エディタを実行します。
- 3 [レジストリ] ウィンドウで、HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters を展開します。
- 4 **LogLevel** レジストリ キーの値がない場合は、右クリックして作成します。
 - a [パラメータ] を右クリックし、[新規] - [DWORD (32 ビット) 値] の順に選択して、**LogLevel** と入力します。
 - b [パラメータ] を選択し、右側のペインで **LogLevel** をダブルクリックして、[値のデータ:] テキスト ボックスに **1** と入力します。

新しい設定が有効になります。Windows Server 2003 以降の場合は再起動する必要はありません。

結果

System Windows イベント ログに、Kerberos エラー イベントのエントリが記録されます。

次のステップ

Kerberos イベント ログを無効にするには、**LogLevel** レジストリ キーの値を削除するか、値を **0** に変更します。

Kerberos データベースでサーバが見つからない

Kerberos 認証で PowerShell サーバを追加した後、正しく追加されていないためにサーバが見つからない場合があります。

問題

サーバに接続しようとする、Kerberos データベースでそのサーバが見つからない。

```
No valid credentials provided (Mechanism level: No valid credentials provided (Mechanism level: Server not found in Kerberos database (7)))
```

原因

このエラーは、いくつかの構成の誤りが原因であると考えられます。

- PowerShell ホストがドメインの一部に含まれていない。
- レルムへのホストのマッピングが適切でない。
- PowerShell ホストのサービス プリンシパル名が正しく組み込まれていない。

注： Kerberos 認証は、接続先が IP アドレスの場合は機能しません。

解決方法

Kerberos 認証を使用して PowerShell ホストを追加するには、DNS または NetBIOS の接続先を入力します。

Kerberos チケットを取得できない

誤った認証情報を指定すると、プラグインで Kerberos チケットの取得に失敗します。

問題

ホストをプラグイン インベントリに追加しようとする失敗し、次のエラー メッセージが表示されます。

```
Pre-authentication information was invalid (24)
```

原因

誤った認証情報が指定されています。

解決方法

正しい認証情報を指定してください。

Kerberos 認証が時刻設定の違いにより失敗する

Kerberos 構成を使用する環境で時刻の設定が一貫していないと、認証に失敗する場合があります。

問題

Kerberos を使用してホストの初期認証またはリソースへのアクセスを試行すると失敗し、以下のエラーメッセージが表示されます。

```
Clock Skew
```

原因

その環境にあるコンピュータのシステム時刻がドメイン コントローラやその他のコンピュータとの間で 5 分以上異なる場合、Kerberos 認証は失敗します。

解決方法

その環境のシステム時刻を同期してください。

Kerberos 認証セッション モードが失敗する

共有セッションまたはユーザーごとのセッションで Kerberos 認証を使用する場合、PowerShell ホストを追加すると失敗することがあります。

問題

共有セッションまたはユーザーごとのセッションでプラグインのインベントリに PowerShell ホストを追加しようとすると、ワークフローが失敗して以下のエラーが表示されます。

```
Null realm name (601) - default realm not specified (Dynamic Script Module name :  
addPowerShellHost#16)
```

原因

デフォルトのレルムが Kerberos 構成ファイル `krb5.conf` で指定されておらず、またユーザー名の一部としても指定されていません。

解決方法

デフォルトのレルムを Kerberos 構成ファイルで指定するか、Kerberos 認証時にユーザー名にレルムを含めてください。

レルムのキー配布センターに到達できない

krb5.conf ファイルにスペルミスが存在すると、ホストを追加する際にエラーが発生することがあります。

問題

ホストを追加する際、Kerberos 認証で *yourrealm* のキー配布センター (KDC) に到達することができません。

```
Cannot get kdc for realm YOURREALM.COM
```

原因

krb5.conf ファイルの libdefaults および realms セクションにスペルミスが存在する可能性があります。

解決方法

krb5.conf ファイルの libdefaults および realms セクションのスペルが正しいことを確認してください。

デフォルトのレルムが見つからない

Kerberos 構成ファイルが適切な形式でないか、正しくエンコードされていないと、Kerberos 認証を必要とする vRealize Orchestrator ワークフローが失敗することがあります。

問題

Kerberos 認証では、デフォルトのレルムを識別できません。

```
Cannot locate default realm
```

原因

vRealize Orchestrator Appliance にアップロードした Kerberos の構成ファイル *krb5.conf* が、UNIX 以外のオペレーティング システムで編集されています。その結果、フォーマットおよびエンコードが正しくありません。

解決方法

vRealize Orchestrator アプライアンスで *krb5.conf* ファイルを読み取るには、ファイルの形式を UNIX、文字エンコーディングを ANSI および UTF-8 に設定する必要があります。

SNMP プラグインの使用

13

SNMP プラグインを使用すると、vRealize Orchestrator から SNMP 対応システムおよびデバイスに接続して情報を受信できます。ワークフローを実行して SNMP デバイスをインベントリ オブジェクトとして定義し、定義したオブジェクトに対して SNMP 操作を実行できます。

プラグインを使用すると、ルーター、スイッチ、ネットワーク プリンタ、UPS デバイスなど、SNMP デバイスに接続することができます。また、SNMP プロトコルを介して vCenter Server からイベントを受信することもできます。

SNMP プラグインは SNMP デバイスと通信する方法を 2 種類提供します。

- 特定の SNMP 変数の値を問い合わせる。
- デバイスから生成されたイベント (SNMP トラップ) をリッスンして、登録されている SNMP マネージャにブッシュする。

このプラグインには SNMP デバイス、クエリ、トラップ ホストの管理と、SNMP 操作の実行に関連する標準ワークフローのセットが含まれています。カスタム ワークフローを生成して SNMP 環境のタスクを自動化することもできます。

この章には、次のトピックが含まれています。

- [SNMP デバイスの管理](#)
- [SNMP クエリの管理](#)
- [SNMP トラップ ホストの管理](#)
- [SNMP トラップの受信](#)
- [汎用 SNMP 要求ワークフロー](#)

SNMP デバイスの管理

ワークフローを実行して SNMP デバイスを vRealize Orchestrator に登録したり、既存のデバイスの設定を編集したり、デバイスの登録を解除したりすることができます。

デバイス管理のワークフロー

「デバイス管理」ワークフロー カテゴリには、SNMP デバイスを管理できるワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **snmp** および **device_management** タグを入力します。

ワークフロー名	説明
SNMP デバイスの編集	登録済みの SNMP デバイスの構成を編集します。
SNMP デバイスの登録	SNMP 対応デバイスをプラグイン インベントリに登録します。
SNMP デバイスの登録解除	SNMP デバイスをプラグイン インベントリから登録解除します。

SNMP デバイスの登録

SNMP デバイスを登録するワークフローを実行し、オプションとして高度な接続パラメータを設定することができます。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **snmp** と **device_management** タグを入力します。
- 3 [SNMP デバイスの登録] ワークフローを見つけ、[実行] をクリックします。
- 4 [デバイスのアドレス] テキストボックスに、SNMP デバイスの IP アドレスまたは DNS 名を入力します。

注： さらに信頼性の高い接続を確立するには、IP アドレスを使用します。

- 5 (オプション) [名前] テキスト ボックスに、[インベントリ] ビューに表示するデバイスの名前を入力します。
テキスト ボックスを空欄のままにしておくと、デバイスのアドレスを使用して、名前が自動生成されます。
- 6 (オプション) 詳細な接続パラメータを設定するには、チェックボックスを選択し、[詳細] タブに移動します。
 - a [ポート] テキストボックスでは、接続ポートを指定します。
デフォルト ポートは 161 です。
 - b [バージョン] ドロップダウン メニューから、使用する SNMP バージョンを選択し、資格情報を付与します。
SNMPv3 のサポートは、MD5 認証の AuthPriv セキュリティ レベルに制限されます。DES パスフレーズは、MD5 パスワードと同じです。

注： SNMPv3 のサポートは廃止されました。

- c [コミュニティ名] テキストボックスに、デバイスの名前を入力します。デフォルトの名前は、[public] です。
- 7 [実行] をクリックします。

結果

ワークフローが正しく実行されると、SNMP デバイスが [インベントリ] ビューに表示されます。

次のステップ

SNMP デバイスにクエリを追加して、[インベントリ] ビューからワークフローを実行できます。

SNMP クエリの管理

クエリを登録済みの SNMP デバイスに追加することも、既存のクエリを実行、コピー、および編集することもできます。また、SNMP デバイスからクエリを削除できます。SNMP クエリを、より複雑なワークフローのビルディング ブロックとして使用できます。

クエリ管理ワークフロー

[クエリ管理] ワークフロー カテゴリには、SNMP クエリを管理できるワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **snmp** および **query_management** タグを入力します。

ワークフロー名	説明
SNMP デバイスへのクエリの追加	SNMP デバイスにクエリを追加します。
SNMP クエリのコピー	あるデバイスから別のデバイスへ SNMP クエリをコピーします。
SNMP クエリの編集	既存の SNMP クエリを編集します。
SNMP デバイスからクエリを削除	SNMP クエリをデバイスから削除します。
SNMP クエリの実行	SNMP デバイスに対してクエリを実行します。

SNMP デバイスへのクエリの追加

ワークフローを実行して、プラグイン インベントリから SNMP デバイスへクエリを追加することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから SNMP デバイスへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **snmp** および **query_management** タグを入力します。
- 2 [SNMP デバイスにクエリを追加]ワークフローを見つけて、[実行] をクリックします。
- 3 クエリを追加する先のデバイスを選択します。
- 4 [タイプ] ドロップダウン メニューからクエリ タイプを選択します。
- 5 [OID] テキスト ボックスに、クエリの対象とする変数のオブジェクト ID を入力します。

以下に、OID 値の例を示します。

- **1.3.6.1.2.1.1.5.0**
- **.1.3.6.1.2.1.1.5.0**

■ iso.3.6.1.2.1.1.5.0

注： プラグインは、数値の OID 値、または iso で始まり数値が続く OID 値のみをサポートします。

- 6 (オプション) [名前] テキスト ボックスにクエリの名前を入力します。

テキスト ボックスを空欄のままにしておくと、タイプと OID パラメータを使用して、名前が自動生成されます。

- 7 [実行] をクリックします。

次のステップ

クエリに対してワークフローを実行するには、[インベントリ] ビューを使用します。

SNMP トラップ ホストの管理

vRealize Orchestrator は、SNMP リスナーとして機能します。SNMP トラップ ホストを開始および停止したり、vRealize Orchestrator が SNMP トラップをリスンするポートを変更したりすることができます。

SNMP プラグインは SNMPv1 および SNMPv2c トラップをサポートします。

注： SNMPv3 のサポートは廃止されました。

重要： クラスタ化された vRealize Orchestrator の展開を使用する場合は、クラスタ内のすべてのノードに SNMP トラップを送信する必要があります。

トラップ ホスト管理ワークフロー

[トラップ ホスト管理] ワークフロー カテゴリには、SNMP トラップ ホストを管理できるワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **snmp** および **trap_host_management** タグを入力します。

ワークフロー名	説明
SNMP トラップ ポートの設定	SNMP トラップをリスンする vRealize Orchestrator のポートを設定します。
トラップ ホストの開始	vRealize Orchestrator で SNMP トラップのリスンを開始します。
トラップ ホストの停止	vRealize Orchestrator で SNMP トラップのリスンを停止します。

vRealize Orchestrator Appliance への SNMP トラップ ポートの追加

SNMP トラップ ポートの設定を実行する前に、ポートを vRealize Orchestrator Appliance に追加する必要があります。

手順

- 1 vRealize Orchestrator Appliance のコマンド ラインに root としてログインします。

- SNMP トラップ ポートの変数を作成します。

```
SNMP_PORT=port_value
```

注： SNMP トラップ用に開いているポートは 81 ~ 32767 です。

- ユーザー データグラム プロトコル (UDP) を介した SNMP トラップ ポート用の Kubernetes NodePort サービスを作成します。

```
kubectl -n prelude expose deployment vro-app --type=NodePort --name=vro-snmp-$SNMP_PORT --port=$SNMP_PORT --target-port=$SNMP_PORT --protocol=UDP
```

- ポートを NodePort サービスに追加します。

```
kubectl -n prelude patch svc vro-snmp-$SNMP_PORT --type='json' -p '[{"op":"replace","path":"/spec/ports/0/nodePort","value":'$SNMP_PORT'}]'
```

- NodePort サービスにパッチを適用して、送信元の IP アドレスが vRealize Orchestrator ポッドに接続できるようにします。

```
kubectl -n prelude patch svc vro-snmp-$SNMP_PORT -p '{"spec":{"externalTrafficPolicy":"Local"}}'
```

- 設定された SNMP ポートのファイアウォール ルールを作成します。

```
iptables -A INPUT -p udp -m udp --dport $SNMP_PORT -j ACCEPT
```

SNMP トラップ ポートの設定

ワークフローを実行して、vRealize Orchestrator が SNMP トラップをリスンするポートを設定することができます。

SNMP トラップのデフォルト ポートは 162 です。Linux システムでは、スーパーユーザー権限でのみ、1024 以下のポートを開くことができます。SNMP トラップ用に開いているポートは 81 ~ 32767 です。

前提条件

- SNMP トラップポートの構成用に vRealize Orchestrator Appliance を準備します。 [vRealize Orchestrator Appliance への SNMP トラップ ポートの追加](#) を参照してください。
- vRealize Orchestrator Client で、[管理] - [インベントリ] を選択し、SNMP デバイスが接続されていることを確認します。

手順

- 管理者として vRealize Orchestrator Client にログインします。
- [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **snmp** および **trap_host_management** タグを入力します。
- [SNMP トラップ ポートの設定] ワークフローを実行します。
- SNMP トラップをリスンする vRealize Orchestrator のポートを入力します。

- 5 [実行] をクリックします。

結果

ワークフローは、トラップ ホストを停止し、新しいポートを設定して、トラップ ホストをもう一度開始します。

注： vRealize Orchestrator 環境をアップグレードする場合は、vRealize Orchestrator Appliance を実行し、[SNMP トラップ ポートの設定] ワークフローを再度実行する必要があります。

SNMP トラップの受信

SNMP プラグインは、単一トラップ メッセージを待機するワークフローまたはトラップを継続的に処理するポリシーを持つワークフローを実行して、SNMP トラップを受信することができます。このプラグインは SNMPv1 および SNMPv2c トラップをサポートします。

SNMP デバイスでのトラップの待機

指定したデバイスから SNMP トラップの受信を待機するワークフローを実行できます。

このワークフローではトリガが発生し、これによりワークフローの実行が停止して、続行する前に SNMP トラップを待機します。トラップが受信されるとワークフローは実行を再開します。このワークフローはより複雑なワークフローの一環として使用することも、特定のニーズに応じてカスタマイズまたは拡張するサンプルとして使用することもできます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから SNMP デバイスへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] に移動し、ワークフローの検索ボックスに **snmp** タグを入力します。
- 2 [SNMP デバイスでのトラップの待機] ワークフローを見つけて、[実行] をクリックします。
- 3 SNMP トラップを待機するデバイスを選択します。
- 4 (オプション) [OID] テキスト ボックスで、特定のトラップのオブジェクトの識別子を入力します。

注： テキスト ボックスを空のままにすると、指定した SNMP デバイスからトラップが受信された後、ワークフローの実行が再開されます。

- 5 [実行] をクリックします。

SNMP トラップ ポリシーの設定

プラグイン インベントリにすでに登録されている SNMP デバイスからトラップを継続的にリッスンするようにポリシーを設定できます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。

- [インベントリ] ビューから SNMP デバイスへの接続があることを確認します。

手順

- 1 vRealize Orchestrator クライアントで、[ライブラリ] - [ポリシー] の順に移動します。
- 2 [SNMP トラップ ポリシー] テンプレートを開き、[実行] をクリックします。
- 3 作成するポリシーの名前を入力します。
- 4 (オプション) ポリシーの説明を入力します。
- 5 ポリシーを設定する SNMP デバイスを選択します。
- 6 [実行] をクリックして、ポリシーを作成します。

vRealize Orchestrator クライアントが、[実行] パースペクティブに切り替わります。

- 7 [ポリシー] ビューで、作成したポリシーを右クリックして [ポリシーの開始] を選択します。

結果

トラップ ポリシーは SNMP トラップのリッスンを開始します。

次のステップ

SNMP トラップ ポリシーを編集できます。

SNMP トラップ ホスト ポリシーの構成

SNMP トラップ ホストを使用して、ポリシーが登録済みの SNMP デバイスとして追加されないホストから SNMP トラップをリッスンします。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから SNMP デバイスへの接続があることを確認します。

手順

- 1 vRealize Orchestrator クライアントで、[ライブラリ] - [ポリシー] の順に移動します。
- 2 [SNMP トラップ ホスト] ポリシー テンプレートを開き、[実行] をクリックします。
- 3 作成するポリシーの名前を入力します。
- 4 (オプション) ポリシーの説明を入力します。
- 5 インベントリ ツリーから Trap Host (Online) を選択します。
- 6 [実行] をクリックして、ポリシーを作成します。

Orchestrator クライアントが、[実行] パースペクティブに切り替わります。

- 7 このポリシーを右クリックして [編集] を選択します。

- 8 **【スクリプト作成】** タブで [ホスト] - [OnTrapAll] の順に展開します。
 - a ワークフローまたはポリシーに関連付けるスクリプトを選択します。
- 9 **【保存して閉じる】** をクリックして、編集済みの設定を適用します。
- 10 **【ポリシー】** ビューで、上記の手順で編集したポリシーを右クリックして **【ポリシーの開始】** を選択します。

次のステップ

SNMP トラップ ホスト ポリシーを編集できます。

トラップ ポリシーの編集

特定の使用事例に合わせて、トラップ ポリシーを編集してカスタマイズすることができます。トラップ ポリシーを編集する場合、優先度や起動設定を変更したり、スクリプトやポリシー関連の権限をカスタマイズすることができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- **【インベントリ】** ビューから SNMP デバイスへの接続があることを確認します。

手順

- 1 vRealize Orchestrator クライアントで、**【ライブラリ】** - **【ポリシー】** の順に移動します。
- 2 編集するポリシーが実行中の場合、ポリシーを右クリックして **【ポリシーの停止】** を選択します。
- 3 このポリシーを右クリックして **【編集】** を選択します。
- 4 **【全般】** タブから、起動設定、優先順位、ポリシーの説明を編集します。
- 5 (オプション) 複雑なシナリオの統合の場合は、**【スクリプティング】** タブから、特定のワークフローまたはスクリプティング コードをポリシーと関連付けます。

トラップの受け取り時にカスタム ワークフローがトリガするようにポリシーを設定できます。

- 6 (オプション) アクセス権を変更する場合は、**【権限】** タブから変更を行います。
ユーザーまたはグループに、ポリシーを編集する権限を与えずに、ポリシーを開始する権限を付与することができます。
- 7 **【保存して閉じる】** をクリックして、編集済みの設定を適用します。
- 8 **【ポリシー】** ビューで、上記の手順で編集したポリシーを右クリックして **【ポリシーの開始】** を選択します。

汎用 SNMP 要求ワークフロー

SNMP ワークフロー カテゴリには、クエリを生成せずに基本的な SNMP 要求の実行を可能にするワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、**【ライブラリ】** - **【ワークフロー】** の順に移動して、ワークフローの検索ボックスに **snmp** タグを入力します。

ワークフロー名	説明
バルク SNMP 値の取得	SNMP デバイスに対して GET BULK クエリを実行します。
次の SNMP 値の取得	SNMP デバイスに対して GET NEXT クエリを実行します。
SNMP 値の取得	SNMP デバイスに対して GET クエリを実行します。
SNMP トラップの送信	指定のアドレスに SNMP トラップを送信します。
すべてのデバイスでのトラップの待機	vRealize Orchestrator にトラップを送信するすべてのホストからの SNMP トラップの受信を待機します。
SNMP デバイスでのトラップの待機	指定したデバイスからの SNMP トラップの受信を待機します。

SOAP プラグインの使用

14

SOAP プラグインを使用すると、vRealize Orchestrator と SOAP ホスト間の通信を確立することで、SOAP Web サービスを管理できます。構成ワークフローを実行して SOAP サービスをインベントリ オブジェクトとして定義し、定義したオブジェクトに対して SOAP 操作を実行できます。

このプラグインには SOAP ホストの管理と SOAP 操作の呼び出しに関連する標準ワークフローのセットが含まれています。カスタム ワークフローを生成して SOAP 環境のタスクを自動化することもできます。

この章には、次のトピックが含まれています。

- SOAP プラグインの構成
- SOAP 操作からの新しいワークフローの生成
- SOAP 操作の呼び出し

SOAP プラグインの構成

SOAP プラグインを構成するには、vRealize Orchestrator クライアントを使用する必要があります。

「構成」ワークフロー カテゴリには、SOAP ホストを管理できるワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **soap** および **configuration** タグを入力します。

ワークフロー名	説明
SOAP ホストの追加	プラグイン インベントリに SOAP ホストを追加します。
プラグイン構成の再読み込み	プラグイン インベントリ内の SOAP ホストのリストを更新します。
SOAP ホストの削除	プラグイン インベントリから SOAP ホストを削除します。 注意: インベントリからホストを削除すると、そのホストで生成されたすべてのワークフローの処理が停止します。
SOAP ホストの更新	プラグイン インベントリ内の SOAP ホストを更新します。
エンドポイント URL で SOAP ホストを更新	優先されるエンドポイント アドレスを使用して SOAP ホストを更新します。新しいエンドポイント アドレスは、WSDL 内に定義されたエンドポイント アドレスの代わりに、SOAP メッセージの送受信に使用されます。

SOAP ホストの追加

ワークフローを実行して、SOAP ホストを追加し、ホストの接続パラメータを設定することができます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **soap** および **configuration** タグを入力します。
- 3 [SOAP ホストの追加] ワークフローを見つけて、[実行] をクリックします。
- 4 [SOAP ホスト] タブで、ホストを作成するためのプロパティを指定します。
 - a [名前] テキスト ボックスにホストの名前を入力します。
 - b WSDL コンテンツをテキスト形式で指定するかどうかを選択します。

オプション	アクション
はい	[WSDL コンテンツ] テキスト ボックスにテキストをコピーします。
いいえ	[WSDL URI] テキスト ボックスに正しいパスを入力します。

- c [接続タイムアウト] テキスト ボックスに、vRealize Orchestrator から SOAP ホストへの接続がタイムアウトするまでの秒数を入力します。
 - d [要求タイムアウト] テキスト ボックスに、SOAP 要求の正常な処理がタイムアウトするまでの秒数を入力します。
- 5 [プロキシ] タブで、プロキシを使用するかどうかを選択します。

オプション	アクション
はい	プロキシ アドレスとプロキシ ポートを指定します。
いいえ	次の手順に進みます。

- 6 [ホスト認証] タブで、認証タイプを選択します。

オプション	説明
なし	認証は要求されません。
基本	基本アクセス認証を指定します。 [ユーザー認証情報] タブで、セッション モードを選択します。 <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、vRealize Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。
ダイジェスト	暗号化を使用するダイジェスト アクセス認証を指定します。 [ユーザー認証情報] タブで、セッション モードを選択します。 <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、vRealize Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。

オプション	説明
NTLM	<p>Window セキュリティ サポート プロバイダ (SSPI) フレームワーク内の NT LAN Manager (NTLM) アクセス認証を指定します。</p> <p>[ユーザー認証情報] タブで、セッション モードを選択します。</p> <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、vRealize Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。 <p>[NTLM 設定] タブで、NTLM ドメインと NTLM ワークステーションを入力します。</p>
ネゴシエーション	<p>Kerberos アクセス認証を指定します。</p> <p>[ユーザー認証情報] タブで、セッション モードを選択します。</p> <ul style="list-style-type: none"> ■ [共有セッション] を選択した場合は、共有セッションの認証情報を指定します。 ■ [ユーザーごとのセッション] を選択した場合は、vRealize Orchestrator クライアントが、ログインしているユーザーから認証情報を取得します。 <p>[ネゴシエーション] タブで、Kerberos サービスの SPN を指定します。</p>

7 [実行] をクリックします。

結果

ワークフローが正常に実行されたら、SOAP ホストが [インベントリ] ビューに表示されます。

次のステップ

SOAP ホスト オブジェクトを探索したり、オブジェクトに対してワークフローを実行したりするには、[インベントリ] ビューを使用します。

SOAP 操作からの新しいワークフローの生成

SOAP 操作からカスタム ワークフローを生成することができます。

カスタム生成したワークフローをハイレベル ワークフローに統合することができます。ワークフロー開発の詳細については、『vRealize Orchestrator を使用したワークフローの開発』ガイドを参照してください。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから SOAP ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **soap** タグを入力します。
- 2 [SOAP 操作からの新しいワークフローの生成] ワークフローを見つけて、[実行] をクリックします。
- 3 使用可能な操作のリストから SOAP 操作を選択します。
- 4 [名前] テキスト ボックスに生成するワークフローの名前を入力します。
- 5 新しいワークフローを生成するワークフロー フォルダの名前を選択します。
ワークフロー ライブラリの既存のフォルダを選択することができます。

6 [実行] をクリックします。

次のステップ

生成したワークフローをテストできます。

カスタム生成したワークフローのテスト

SOAP 操作で生成したカスタム ワークフローを実行して、その操作の出力パラメータを取得することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから SOAP ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動します。
- 2 カスタム ワークフローを見つけて、[実行] をクリックします。
- 3 SOAP 操作に必要な入力パラメータを指定します。
- 4 [実行] をクリックします。
- 5 (オプション) [ログ] タブで、使用可能な出力パラメータのリストを確認します。

SOAP 操作の呼び出し

新しいワークフローを生成することなく、SOAP 操作を直接呼び出すことができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューから SOAP ホストへの接続があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **soap** タグを入力します。
- 2 [SOAP 操作の呼び出し] ワークフローを見つけて、[実行] をクリックします。
- 3 使用可能な操作のリストから SOAP 操作を選択します。
- 4 SOAP 操作に必要な入力パラメータを指定します。
- 5 [実行] をクリックします。
- 6 (オプション) [ログ] タブで、使用可能な出力パラメータのリストを確認します。

SQL プラグインの使用

15

SQL プラグインが提供する API を使用して、SQL データベースとその他の表形式データ ソース（スプレッドシート、フラット ファイルなど）への接続を実装できます。

JDBC に基づく SQL プラグイン API は、SQL ベースのデータベースにアクセスするためのコール レベル API を提供します。また、SQL プラグインは、ワークフローで API を使用方法を示すサンプル ワークフローも提供します。

この章には、次のトピックが含まれています。

- [SQL プラグインの構成](#)
- [SQL サンプル ワークフローの実行](#)
- [SQL 操作の実行](#)

SQL プラグインの構成

SQL プラグインに組み込まれているワークフローを使用し、vRealize Orchestrator クライアントから実行して、データベースを追加、更新、または削除するように SQL プラグインを構成することができます。

SQL プラグインの「構成」ワークフロー カテゴリには、データベースとデータベース テーブルを管理できるワークフローが含まれています。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **sql** および **configuration** タグを入力します。

ワークフロー名	説明
データベースの追加	データベース オブジェクトを SQL プラグイン インベントリに追加します。
データベースへのテーブルの追加	データベース テーブルを SQL プラグイン インベントリ内のデータベースに追加します。
データベースの削除	SQL プラグイン インベントリからデータベース オブジェクトを削除します。
データベースからのテーブルの削除	SQL プラグイン インベントリ内のデータベースからデータベース テーブルを削除します。
データベースの更新	SQL プラグイン インベントリ内のデータベース オブジェクトの構成を更新します。
データベースの検証	SQL プラグイン インベントリ内のデータベースを検証します。

データベースの追加

ワークフローを実行して、vRealize Orchestrator サーバにデータベースを追加し、ホスト接続パラメータを設定することができます。

安全な接続を必要とするデータベースを追加するときには、データベースの SSL 証明書をインポートする必要があります。コントロールセンターの [信頼された証明書] タブで SSL 証明書をインポートする必要があります。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **sql** および **configuration** タグを入力します。
- 3 [データベースの追加] ワークフローを見つけて、[実行] をクリックします。
- 4 [名前] テキスト ボックスにデータベースの名前を入力します。
- 5 データベースのタイプを選択します。
- 6 [接続 URL] テキスト ボックスに、データベースのアドレスを入力します。

データベースタイプ	構文
Oracle	<code>jdbc:oracle:thin:@database_url:port_number:SID</code>
Microsoft SQL (SQL 認証を使用)	<code>jdbc:jtds:sqlserver://database_url:port_number/database_name</code>
Microsoft SQL (Windows アカウント認証を使用)	<code>jdbc:jtds:sqlserver://database_url:port_number/database_name;useNTLMv2=true;domain=domain_name</code>
PostgreSQL	<code>jdbc:postgresql://database_url:port_number/database_name</code>
MySQL	<code>jdbc:mysql://database_url:port_number/database_name</code>

- 7 [ユーザー認証情報] タブで、プラグインがデータベースへの接続に使用するセッション モードを選択します。

オプション	説明
共有セッション	プラグインはデータベースへの接続時に、共有の認証情報を使用します。共有セッション用のデータベース認証情報を指定する必要があります。
ユーザーごとのセッション	vRealize Orchestrator クライアントは、ログインしているユーザーから認証情報を取得します。 注： [ユーザーごとのセッション] モードを使用するには、ユーザー名のみを使用して認証する必要があります。認証に <code>domain\user</code> または <code>user@domain</code> を使用しないでください。

- 8 [実行] をクリックします。

結果

ワークフローが正常に実行されたら、データベースとそれに属するすべてのテーブルが、[インベントリ] ビューに表示されます。

データベースへのテーブルの追加

ワークフローを実行して、SQL プラグイン インベントリ内のデータベースにテーブルを追加することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューからデータベースへの接続が確立していることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **sql** および **configuration** タグを入力します。
- 2 [データベースにテーブルを追加] ワークフローを見つけて、[実行] をクリックします。
- 3 テーブルを追加するデータベースを選択します。
- 4 追加するテーブルを選択します。
- 5 [実行] をクリックします。

結果

ワークフローが正常に実行されたら、追加されたデータベース テーブルが vRealize Orchestrator クライアントの [インベントリ] ビューに表示されます。

データベースの更新

ワークフローを実行して、SQL プラグイン インベントリ内のデータベースの構成を更新することができます。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **sql** および **configuration** タグを入力します。
- 3 [データベースの更新] ワークフローを見つけて、[実行] をクリックします。
- 4 更新するデータベースを選択します。
- 5 [名前] テキスト ボックスで、データベースの新しい名前を入力します。
入力した名前が付けられたデータベースが [インベントリ] ビューに表示されます。
- 6 データベースのタイプを選択します。
- 7 [接続 URL] テキスト ボックスで、データベースの新しいアドレスを入力します。

- 8 [ユーザー認証情報] タブで、プラグインがデータベースへの接続に使用するセッション モードを選択します。

オプション	説明
共有セッション	プラグインはデータベースへの接続時に、共有の認証情報を使用します。共有セッション用のデータベース認証情報を指定する必要があります。
ユーザーごとのセッション	vRealize Orchestrator クライアントは、ログインしているユーザーから認証情報を取得します。 注： [ユーザーごとのセッション] モードを使用するには、ユーザー名のみを使用して認証する必要があります。認証に <code>domain\user</code> または <code>user@domain</code> を使用しないでください。

- 9 [実行] をクリックします。

SQL サンプル ワークフローの実行

SQL プラグインのワークフローを実行して、JDBC URL の生成、JDBC 接続のテスト、および JDBC テーブルの行の管理といった JDBC 操作を実行できます。SQL プラグインのワークフローを実行して、データベースやデータベース テーブルを管理したり、SQL 操作を実行したりできます。

JDBC URL の生成

vRealize Orchestrator クライアントからワークフローを実行して、JDBC 接続 URL を生成できます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc** タグを入力します。
- 3 [JDBC URL ジェネレータ] ワークフローを見つけて、[実行] をクリックします。
- 4 [全般] タブで、URL を生成するデータベースのタイプを選択します。

注： Microsoft データベースを使用している場合、[Microsoft] タブを選択して、データベース インスタンス名とユーザー ドメイン名を入力します。

- 5 必要な情報を入力して、データベースの URL を生成します。
 - a データベース サーバ名または IP アドレスを入力します。
 - b データベース名を入力します。
 - c (オプション) データベースのポート番号を入力します。
ポート番号を指定しない場合、ワークフローではデフォルトのポート番号が使用されます。
 - d ユーザー名を入力してデータベースにアクセスします。
 - e パスワードを入力してデータベースにアクセスします。

- 6 [実行] をクリックします。

JDBC 接続のテスト

vRealize Orchestrator クライアントからワークフローを実行して、データベースとの接続をテストすることができます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- 3 [JDBC 接続の例] ワークフローを見つけて、[実行] をクリックします。
- 4 必要な情報を入力して、データベース接続をテストします。
 - a ユーザー名を入力してデータベースにアクセスします。
 - b テスト対象の URL を入力します。
 - c パスワードを入力してデータベースにアクセスします。
- 5 [実行] をクリックします。

JDBC を使用したテーブルの作成

vRealize Orchestrator クライアントからワークフローを実行して、データベースを作成することができます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- 3 [JDBC テーブルの作成] サンプル ワークフローを見つけて、[実行] をクリックします。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - a パスワードを入力してデータベースにアクセスします。
 - b データベース接続 URL を入力します。
 - c ユーザー名を入力してデータベースにアクセスします。

- SQL 作成ステートメントを入力します。

構文の例は次のとおりです。

```
CREATE TABLE "table_name"
("column1" "data_type_for_column1",
"column2" "data_type_for_column2")
```

- [実行] をクリックします。

JDBC テーブルへの行の挿入

vRealize Orchestrator クライアントからワークフローを実行して、JDBC テーブルへの行の挿入をテストすることができます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- vRealize Orchestrator クライアントへのログイン
- [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- [JDBC テーブルへの挿入] ワークフローを見つけて、[実行] をクリックします。
- [データベース接続] タブで、必要な情報を入力します。
 - データベース接続 URL を入力します。
 - ユーザー名を入力してデータベースにアクセスします。
 - パスワードを入力してデータベースにアクセスします。
- [SQL ステートメント] タブで、次の例のような SQL 挿入ステートメントを入力します。

```
INSERT INTO "table_name" ("column1", "column2")
VALUES ("value1", "value2")
```

- [挿入する値] タブで値を入力して、行に挿入する値を入力します。
- [実行] をクリックします。

JDBC テーブルから行を選択

vRealize Orchestrator クライアントからワークフローを実行して、JDBC テーブルから行を選択することができます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- 3 [JDBC テーブルの選択] ワークフローを見つけて、[実行] をクリックします。
- 4 [データベース接続] タブで、必要な情報を入力します。
 - a データベース接続 URL を入力します。
 - b ユーザー名を入力してデータベースにアクセスします。
 - c パスワードを入力してデータベースにアクセスします。
- 5 [SQL ステートメント] タブで、次の例のような SQL 選択ステートメントを入力します。
構文の例は次のとおりです。

```
SELECT * FROM "table_name"
```

- 6 [実行] をクリックします。

JDBC テーブルからエントリを削除

vRealize Orchestrator クライアントからワークフローを実行して、JDBC テーブルからの 1 つのエントリの削除をテストすることができます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- 2 [JDBC テーブルからエントリを削除] サンプル ワークフローを見つけて、[実行] をクリックします。
- 3 必要な情報を入力します。
 - a 削除するユーザー エントリの名を入力します。
 - b ユーザー名を入力してデータベースにアクセスします。
 - c JDBC 接続 URL を入力します。
 - d 削除するユーザー エントリの姓を入力します。
 - e パスワードを入力してデータベースにアクセスします。
- 4 次の構文例のような SQL 削除ステートメントを入力します。

```
DELETE FROM "table_name" where ("column1" = ?, "column2" = ?)
```

- 5 [実行] をクリックします。

JDBC テーブルからすべてのエントリを削除

vRealize Orchestrator クライアントからワークフローを実行して、JDBC テーブルからすべてのエントリを削除することができます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- 2 [JDBC テーブルからすべて削除] サンプル ワークフローを見つけて、[実行] をクリックします。
- 3 必要な情報を入力します。
 - a データベース接続 URL を入力します。
 - b ユーザー名を入力してデータベースにアクセスします。
 - c パスワードを入力してデータベースにアクセスします。
- 4 次の構文例のような SQL 削除ステートメントを入力します。

```
DELETE FROM "table_name"
```

- 5 [実行] をクリックします。

JDBC テーブルのドロップ

vRealize Orchestrator クライアントからワークフローを実行して、JDBC テーブルのドロップをテストすることができます。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- 2 [JDBC ドロップ テーブル サンプル] ワークフローを見つけて、[実行] をクリックします。
- 3 必要な情報を入力します。
 - a パスワードを入力してデータベースにアクセスします。
 - b データベース接続 URL を入力します。
 - c ユーザー名を入力してデータベースにアクセスします。

- 4 次の構文例のような SQL ドロップ ステートメントを入力します。

```
DROP TABLE "table_name"
```

- 5 [実行] をクリックします。

完全な JDBC サイクルの実行

vRealize Orchestrator クライアントからワークフローを実行して、1つの完全なサイクルですべての JDBC サンプル ワークフローをテストします。

前提条件

ログイン時に使用したユーザー アカウントに JDBC ワークフローの実行に必要な権限があることを確認します。

手順

- [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **jdbc_examples** タグを入力します。
- [JDBC 完全サイクル] サンプル ワークフローを見つけて、[実行] をクリックします。
- 必要な情報を入力します。
 - データベース接続 URL を入力します。
 - ユーザー名を入力してデータベースにアクセスします。
 - パスワードを入力してデータベースにアクセスします。
- データベースのエントリに使用する値を入力します。
- [実行] をクリックします。

SQL 操作の実行

SQL ワークフローを使用して SQL 操作を実行できます。

vRealize Orchestrator クライアントで SQL 操作ワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **sql** タグを入力します。

ワークフロー名	説明
データベースに対するカスタム クエリの実行	指定したデータベースに対してカスタム クエリを実行し、影響を受ける行の数を返します。ワークフローを実行して、クエリを更新、削除、挿入、書き込むことができます。
テーブルの CRUD ワークフローの生成	特定のテーブルの作成、読み取り、更新、削除ワークフローを生成します。
データベースからのカスタム クエリの読み取り	特定のデータベースに対してカスタム クエリを実行し、その結果をプロパティの配列に返します。ワークフローを実行して、クエリの選択、読み込みができます。

テーブルの CRUD ワークフローの生成

ワークフローを実行して、特定のテーブルを作成、読み取り、更新、および削除ワークフローを生成することができます。

前提条件

- 管理者として vRealize Orchestrator クライアントにログインしていることを確認します。
- [インベントリ] ビューからデータベースへの接続が確立していることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **sql** タグを入力します。
- 2 [テーブルの CRUD ワークフローの生成] ワークフローを見つけて、[実行] をクリックします。
- 3 ワークフローを生成するテーブルを選択します。
- 4 ワークフローを生成するワークフロー フォルダを選択します。
- 5 既存のワークフローを上書きするかどうかを選択します。

オプション	説明
はい	生成したワークフローで同じ名前の既存のワークフローを上書きします。
いいえ	フォルダ内に同じ名前のワークフローが存在する場合は、新しいワークフローは生成されません。

- 6 (オプション) [読み取り専用列を選択] タブで、入力しない列を選択します。
選択した列は、生成した CRUD ワークフローで編集することはできません。
- 7 [実行] をクリックします。

結果

ワークフローが正常に実行されたら、選択したワークフロー フォルダに CRUD ワークフローが表示されます。

次のステップ

生成されたワークフローは、選択したデータベース テーブルで実行できます。

SSH プラグインの使用

16

SSH プラグイン ワークフローを使用すると、SSH をサポートするリモート ホストに対して SSH コマンドを実行し、vRealize Orchestrator サーバとリモート ホスト間で安全な接続を介してファイルを転送できます。

この章には、次のトピックが含まれています。

- SSH プラグインの構成
- SSH プラグイン サンプル ワークフローの実行

SSH プラグインの構成

SSH プラグインの構成ワークフローを実行して、vRealize Orchestrator と SSH ホスト間の接続を管理できます。

vRealize Orchestrator クライアントでこれらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **ssh** および **configuration** タグを入力します。

ワークフロー名	説明
SSH ホストへのルート フォルダの追加	SSH ホストへの既存の接続にルート フォルダを追加します。
SSH ホストの追加	SSH ホストへの接続を既存の構成に追加します。
SSH ホストからのルート フォルダの削除	SSH ホストへの既存の接続からルート フォルダを削除します。
SSH ホストの削除	SSH ホストへの既存の接続を既存の構成から削除します。
SSH ホストの更新	SSH ホストへの既存の接続を更新します。

SSH ホストの追加

SSH プラグインを設定すると、接続を確実に暗号化することができます。

手順

- 1 vRealize Orchestrator クライアントへのログイン。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **ssh** および **configuration** タグを入力します。
- 3 [SSH ホストの追加] ワークフローを見つけて、[実行] をクリックします。

- 4 [全般情報] タブの [ホスト名] テキストボックスに、vRealize Orchestrator から SSH を使用してアクセスするホストの名前を入力します。
- 5 ターゲット ポートを入力します。デフォルトの SSH ポートは 22 です。
ホストが SSH 接続のリストに追加されます。
- 6 (オプション) サーバ上でのエントリ パスを構成します。
 - a [新規ルート フォルダ] をクリックします。
 - b 新しいパスを入力し、[値の挿入] をクリックします。
- 7 [認証] タブで、SSH コマンドを実行するために必要な権限を持つユーザーのユーザー名を入力します。
- 8 認証タイプを選択します。

オプション	アクション
はい	パスワード認証を使用するには、パスワードを入力します。
いいえ	キー認証を使用するには、プライベート キーへのパスとプライベート キーのパスフレーズを入力します。

- 9 [実行] をクリックします。

結果

vRealize Orchestrator クライアントの [インベントリ] ビューで SSH ホストが使用できるようになりました。

SSH プラグイン サンプル ワークフローの実行

vRealize Orchestrator クライアントから SSH プラグインのサンプル ワークフローを実行して、vRealize Orchestrator サーバと SSH ホストとの接続をテストすることができます。

■ キー ペアの作成

vRealize Orchestrator クライアントからワークフローを実行して、キー ペアを生成できます。キー ペアを使用することで、パスワードなしで SSH ホストに接続することができます。

■ キー ペアのパスフレーズの変更

vRealize Orchestrator クライアントからワークフローを実行して、最後に生成したキー ペアのパスフレーズを変更することができます。

■ vRealize Orchestrator パブリック キーの SSH ホストへの登録

パスワードの代わりにパブリック キーを使用できます。vRealize Orchestrator パブリック キーを SSH ホストへ登録するには、vRealize Orchestrator クライアントからワークフローを実行します。

■ SSH コマンドの実行

vRealize Orchestrator クライアントからワークフローを実行して、リモートの SSH サーバ上で SSH コマンドを実行できます。

■ SSH ホストからのファイルのコピー

vRealize Orchestrator クライアントでワークフローを実行して、SSH ホストから vRealize Orchestrator サーバにファイルをコピーすることができます。

■ SSH ホストへのファイルのコピー

vRealize Orchestrator クライアントでワークフローを実行して、vRealize Orchestrator サーバから SSH ホストにファイルをコピーすることができます。

キー ペアの作成

vRealize Orchestrator クライアントからワークフローを実行して、キー ペアを生成できます。キー ペアを使用することで、パスワードなしで SSH ホストに接続することができます。

キー ペアはパブリック キーとプライベート キーで構成されています。vRealize Orchestrator では、プライベート キーを使用して SSH ホストのパブリック キーへ接続することができます。パスフレーズを使用して、セキュリティを向上させることができます。

注意： すべての vRealize Orchestrator ユーザーは、正しい権限のセットを持っていればプライベート キーの読み取り、使用、および上書きが可能です。

前提条件

ログイン時に使用したユーザー アカウントに SSH ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **ssh** タグを入力します。
- 3 [キーペアの生成] ワークフローを見つけて、[実行] をクリックします。
- 4 必要な情報を入力します。
 - a キー タイプを入力します。
 - b キー サイズを入力します。
 - c (オプション) パスフレーズを入力します。

注： パスフレーズは後で変更できます。

- d (オプション) コメントを入力します。
- 5 [実行] をクリックします。

キー ペアが存在する場合は、新しいキー ペアで上書きされます。

キー ペアのパスフレーズの変更

vRealize Orchestrator クライアントからワークフローを実行して、最後に生成したキー ペアのパスフレーズを変更することができます。

前提条件

ログイン時に使用したユーザー アカウントに SSH ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **ssh** タグを入力します。
- 3 [キー ペアのパスフレーズの変更] ワークフローを見つけて、[実行] をクリックします。
- 4 [パスフレーズの変更] タブで、キー ペアのパスフレーズをリセットします。
 - a 現在のパスフレーズを入力します。
 - b 新しいパスフレーズを入力します。
- 5 [実行] をクリックします。

vRealize Orchestrator パブリック キーの SSH ホストへの登録

パスワードの代わりにパブリック キーを使用できます。vRealize Orchestrator パブリック キーを SSH ホストへ登録するには、vRealize Orchestrator クライアントからワークフローを実行します。

前提条件

ログイン時に使用したユーザー アカウントに SSH ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **ssh** タグを入力します。
- 3 [ホストに vCO パブリック キーを登録] ワークフローを見つけて、[実行] をクリックします。
- 4 [ホストに VS-O を登録] タブで、SSH ホスト名を指定し、そのホストにログインするためのユーザー名とパスワードを設定します。

注： SSH ホストに登録される認証情報を指定する必要があります。

- 5 [実行] をクリックします。

結果

登録ユーザーとして SSH ホストに接続する場合は、パスワード認証の代わりにパブリック キー認証を使用できません。

SSH コマンドの実行

vRealize Orchestrator クライアントからワークフローを実行して、リモートの SSH サーバ上で SSH コマンドを実行できます。

前提条件

ログイン時に使用したユーザー アカウントに SSH ワークフローの実行に必要な権限があることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **ssh** タグを入力します。
- 3 [SSH コマンドの実行] ワークフローを見つけて、[実行] をクリックします。
- 4 [ホストの選択] タブで、SSH ホスト名または IP アドレスを入力します。
- 5 [コマンド] タブで、実行する SSH コマンドを入力します。

注： デフォルトの SSH コマンドは **uptime** です。これは、サーバがアクティブな時間の長さと、その時間のユーザーの負荷を表示します。

- 6 [エンコーディング] タブで、エンコード方法を指定します。
デフォルトのシステム エンコーディングを使用するには、このフィールドを空白のままにします。
- 7 [認証] タブで、ユーザー名を入力します。
- 8 (オプション) パスワード認証を使用するには、このチェックボックスをオンにします。

注： デフォルト オプションでは、キー ファイル認証を使用します。

- 9 認証方法にパスワードが必要な場合は、パスワードを入力します。それ以外の場合は、プライベート キーのパスを入力してから、そのプライベート キーのパスフレーズを入力します。
- 10 [実行] をクリックします。

SSH ホストからのファイルのコピー

vRealize Orchestrator クライアントでワークフローを実行して、SSH ホストから vRealize Orchestrator サーバにファイルをコピーすることができます。

SSH プラグインは、SFTP を実装する Java JCraft ライブラリを使用します。「SCP get コマンド」ワークフローは、SFTP を使用してファイルを転送します。

前提条件

ログイン時に使用したユーザー アカウントに SSH ワークフローの実行に必要な権限があることを確認します。

注： vRealize Orchestrator は、フォルダに書き込みをするための書き込み権限を明示的に保持している必要があります。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **ssh** タグを入力します。
- 3 [SCP get コマンド] ワークフローを見つけて、[実行] をクリックします。

- 4 [ホスト] タブで、ソース ホスト情報を入力します。
 - a SSH ホスト名または IP アドレスを入力します。
 - b SSH ユーザー名とパスワードを入力します。
- 5 [ファイル] タブで、ファイル情報を入力します。
 - a リモート SSH ホストから取得するファイルのパスを入力します。
 - b ファイルをコピーする先の vRealize Orchestrator サーバ上のディレクトリのパスを入力します。
- 6 [実行] をクリックします。

SSH ホストへのファイルのコピー

vRealize Orchestrator クライアントでワークフローを実行して、vRealize Orchestrator サーバから SSH ホストにファイルをコピーすることができます。

SSH プラグインは、SFTP を実装する Java JCraft ライブラリを使用します。「SCP put コマンド」ワークフローは、SFTP を使用してファイルを転送します。

前提条件

ログイン時に使用したユーザー アカウントに SSH ワークフローの実行に必要な権限があることを確認します。

手順

- 1 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **ssh** タグを入力します。
- 2 [SCP put コマンド] ワークフローを見つけて、[実行] をクリックします。
- 3 [ホスト] タブで、ソース ホスト情報を入力します。
 - a SSH ホスト名または IP アドレスを入力します。
 - b SSH ユーザー名とパスワードを入力します。
- 4 [ファイル] タブで、ファイル情報を入力します。
 - a ローカルの Orchestrator サーバからリモートの SSH ホストへコピーするファイルのパスを入力します。
 - b ファイルをコピーする先のリモート SSH ホスト上のディレクトリのパスを入力します。
- 5 [実行] をクリックします。

vCenter Server プラグインの使用

17

vCenter Server プラグインを使用すると、複数の vCenter Server インスタンスを管理できます。vCenter Server プラグイン API を使用するワークフローを作成して、vCenter Server 環境のタスクを自動化できます。

vCenter Server プラグインは、vCenter Server API を、ワークフローで使用できる JavaScript にマップします。また、ワークフローに含めることができる個々の vCenter Server タスクを実行するアクションも提供します。

vCenter Server プラグインは vCenter Server 操作を自動化する標準ワークフローのライブラリを提供します。たとえば、仮想マシンを作成、クローン作成、移行、削除するワークフローを実行できます。

注： 多くの vCenter Server プラグイン ワークフローは、vCenter Server とのみ通信します。ただし、一部のゲスト操作ワークフローでは、vCenter Server で管理されている ESXi ホストとの通信が必要です。これらのワークフローを実行する前に、vRealize Orchestrator コントロール センターを使用して ESXi ホストの証明書をインポートする必要があります。詳細については、『VMware vRealize Orchestrator のインストールおよび構成』の「vRealize Orchestrator の証明書の管理」を参照してください。

vCenter Server プラグインには、vRealize Orchestrator のスクリプト API に、スクリプト オブジェクトとしてポリシー ベース管理 (PBM) API およびストレージ監視サービス (SMS) API が含まれています。ストレージ ポリシー ベース管理のポリシーおよびコンポーネントは、vRealize Orchestrator Client の [インベントリ] タブに表示されます。

この章には、次のトピックが含まれています。

- [vCenter Server プラグインの構成](#)
- [vCenter Server プラグインのスクリプト API](#)
- [vCenter Server プラグイン インベントリの使用](#)
- [クエリのパフォーマンスに関する考慮事項](#)
- [vCenter Server プラグインを使用した XPath 式の使用](#)
- [vCenter Server プラグインのワークフロー ライブラリ](#)

vCenter Server プラグインの構成

vRealize Orchestrator を使用して vSphere インベントリでオブジェクトを管理する前に、また、オブジェクトに対してワークフローを実行するときに、vCenter Server プラグインを構成し、vRealize Orchestrator と統合対象の vCenter Server インスタンス間の接続パラメータを定義する必要があります。

vCenter Server プラグインを構成するには、vRealize Orchestrator Client から vCenter Server 構成ワークフローを実行します。構成ワークフローを参照してください。

vCenter Server インスタンスへの接続の構成

vCenter Server インスタンスへの接続を構成するには、vRealize Orchestrator クライアントで vCenter Server 構成ワークフローを実行します。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **vcenter** および **configuration** タグを入力します。
- 3 [vCenter Server インスタンスの追加] ワークフローを見つけて、[実行] をクリックします。
- 4 [vCenter Server インスタンスのプロパティの設定] タブで、追加する vCenter Server インスタンスがインストールされているマシンの IP アドレスまたはホスト名を入力します。

注： 入力するホスト名では大文字と小文字が区別されます。

- 5 デフォルトのポート値 **443** は保持します。
- 6 vCenter Server インスタンスへの接続に使用する SDK のデフォルトの場所は保持します。
- 7 vRealize Orchestrator を使用して vCenter Server インスタンスを管理するかどうかを選択します。
- 8 追加する vCenter Server インスタンスの証明書の警告を無視するかどうかを選択します。

証明書の警告を無視する場合、vCenter Server インスタンスの証明書が警告なしで受け入れられ、信頼されているストアに追加されます。

- 9 [接続プロパティの設定] タブで、vCenter Server システムでユーザーアクセスを管理するためにユーザーごとのセッションを使用するかどうかを選択します。

オプション	説明
一意のセッションの共有	<p>vRealize Orchestrator がオーケストレーションする vCenter Server インスタンスと同じ PSC を認証プロバイダとして使用しない場合に選択します。</p> <p>vRealize Orchestrator で vCenter Server への接続を 1 つのみ作成できるようにします。</p> <p>[ユーザー名] および [パスワード] テキスト ボックスに vCenter Server ホストへの接続を確立するために使用する vRealize Orchestrator の認証情報を入力します。</p> <p>選択するユーザーは、vCenter Server エクステンションを管理する権限およびカスタム定義された権限のセットを持つ、有効なユーザーである必要があります。通常、vRealize Orchestrator システム ワークフローを実行するために、vRealize Orchestrator はこれらの認証情報を使用して VirtualCenter Web サービスを監視します。</p>
ユーザーごとのセッション	<p>vRealize Orchestrator と vCenter Server が同じ PSC を認証プロバイダとして使用する場合に選択します。</p> <p>vCenter Server とのセッションを作成します。</p> <p>注： このアクションは CPU、メモリ、帯域幅を急速に消費します。</p>

選択するユーザー アカウントは、統計などのデータを収集するポリシー エンジンにも使用されます。選択するユーザーに十分な権限がない場合、ポリシー エンジンは vCenter Server インベントリの必要な場所にアクセスできないため、必要なデータを収集できません。

- 10 (オプション) ユーザー ドメインを入力します。

共有セッションを使用する場合にのみ、ユーザー ドメイン名を指定する必要があります。

注： ユーザーごとにセッションが選択されている場合は、このテキスト ボックスに入力します。

- 11 (オプション) [追加のエンドポイント] タブで、vSphere ストレージ管理エンドポイントの URL を入力します。

ポリシー ベース管理 (PBM) エンドポイント、ストレージ監視サービス (SMS) エンドポイント、またはその両方を構成できます。

- a PBM エンドポイントの URL を入力します。デフォルトの vSphere 6.5 PBM エンドポイント URL は、**https://vSphereHostName:443/pbm** です。
- b SMS エンドポイントの URL を入力します。デフォルトの vSphere 6.5 SMS エンドポイント URL は、**https://vSphereHostName:443/sms/sdk** です。

- 12 [実行] をクリックします。

結果

ワークフローが正常に実行された後、ワークフローに属している vCenter Server インスタンスとすべての vSphere オブジェクトが [インベントリ] ビューに表示されます。

vCenter Server プラグインのスクリプト API

vCenter Server スクリプト API には、vRealize Orchestrator と vCenter Server 間の通信を可能にするクラスと、クラスの属性、メソッド、コンストラクタが含まれています。API を使用してカスタム ワークフローを作成できます。

使用可能な API オブジェクトのリストについては、<https://www.vmware.com/support/orchestrator/doc/vro-vsphere65-api/index.html> を参照してください。

vCenter Server プラグイン インベントリの使用

vCenter Server プラグインは、接続されている vCenter Server インスタンスのすべてのオブジェクトをインベントリ ビューに公開します。

vCenter Server インベントリ オブジェクトに使用できるワークフローを表示するには、vRealize Orchestrator クライアントで [管理] - [インベントリ] - [vSphere vCenter プラグイン] の順に移動します。

クエリのパフォーマンスに関する考慮事項

vRealize Orchestrator の vCenter Server プラグインを使用して、特定のオブジェクトの vCenter Server インベントリをクエリできます。

メソッドのクエリ

クエリを実行するには、vCSearchIndex 管理対象オブジェクト、または `getAllDatastores()`、`getAllVirtualMachines()`、`findAllForType()` などのプラグイン インベントリに含まれているオブジェクト ファインダ メソッドのいずれかを使用できます。

パフォーマンス

デフォルトでは、どちらの方法も検索クエリでメソッドのパラメータに対する引数として一連のプロパティを指定しない限り、それらのプロパティのいずれも指定せずにクエリ対象のオブジェクトを返します。

注: `getAll...` および `findAll...` のファインダ オブジェクトでは常にクエリ式を使用して、Orchestrator クライアントが返されたオブジェクトの大規模なフィルタリングを実行しないようにします。これは、Orchestrator サーバの全体的なパフォーマンスに影響する可能性があります。

vCenter Server インベントリをクエリするには、2 つのタイプの式を使用できます。

式のタイプ	説明
名前の式	クエリ パラメータの引数として名前を指定できます。 注： vCenter Server プラグイン インベントリに表示されるプラグイン オブジェクトの名前に従って、指定した名前引数でフィルタリングします。
XPath 式	XPath のクエリ言語に基づいて式を使用できます。詳細については、 vCenter Server プラグインを使用した XPath 式の使用 を参照してください。

カスタム プロパティ、このオブジェクトへの各参照、ワークフロー、またはアクションを使用して vCenter Server のインベントリ オブジェクトを呼び出す場合は、主なパフォーマンスのオーバーヘッドを生成する vCenter Server にクエリを送信します。パフォーマンスを最適化し、ワークフローの実行中にオブジェクトのシリアル化/非シリアル化を何度も実行しないようにするには、オブジェクトをワークフロー属性や入力または出力パラメータとして保存するのではなく、共有リソースを使用して保存することをお勧めします。このような共有リソースには、構成要素またはリソース要素を含めることができます。

vCenter Server プラグインを使用した XPath 式の使用

vCenter Server プラグインのファインダ メソッドを使用して vCenter Server インベントリ オブジェクトのクエリを実行できます。XPath 式を使用して検索パラメータを定義できます。

vCenter Server プラグインには `getAllDatastores()`、`getAllResourcePools()`、`findAllForType()` など、オブジェクトのファインダ メソッド セットが含まれています。これらのメソッドを使用して vRealize Orchestrator サーバに接続している vCenter Server インスタンスのインベントリにアクセスし、ID や名前などのプロパティを指定して、オブジェクトを検索できます。

パフォーマンス上の理由により、検索クエリで一連のプロパティを指定しない限り、ファインダ メソッドではクエリ対象のオブジェクトのプロパティを返しません。

Orchestrator ドキュメントのホーム ページでは、vCenter Server プラグインの Scripting API のオンラインバージョンを参照できます。

重要： XPath 式に基づくクエリは、vRealize Orchestrator のパフォーマンスに影響を及ぼすことがあります。ファインダ メソッドは vCenter Server 側に存在する指定したタイプのオブジェクトをすべて返すほか、vCenter Server プラグイン側にクエリ フィルタを適用するためです。

vCenter Server プラグイン例を使用した XPath 式の使用

ファインダ メソッドを呼び出す際は、XPath クエリ言語に基づいた式を使用できます。検索では XPath 式に一致するすべてのインベントリ オブジェクトが返されます。すべてのプロパティがクエリの対象になる場合は、これらのプロパティを文字列アレイ形式の検索スクリプトに含めます。

次の JavaScript の例は VcPlugin スクリプト オブジェクトと XPath 式を使用して、vCenter Server の管理対象オブジェクトに含まれていて、かつ名前に **ds** という文字列が使用されているすべてのデータベース オブジェクトの名前を返しています。

```
var datastores = VcPlugin.getAllDatastores(null, "xpath:name[contains(.,'ds')]");
for each (datastore in datastores){
    System.log(datastore.name);
}
```

Server スクリプト オブジェクトと findAllForType ファインダ メソッドを使用して、同じ XPath 式を呼び出すことができます。

```
var datastores = Server.findAllForType("VC:Datastore", "xpath:name[contains(.,'ds')]");
for each (datastore in datastores){
    System.log(datastore.name);
}
```

次のスクリプトの例は、ID が **1** の数字で始まるすべてのホスト システム オブジェクトの名前を返しています。

```
var hosts = VcPlugin.getAllHostSystems(null, "xpath:id[starts-with(.,'1')]");
for each (host in hosts){
    System.log(host.name);
}
```

次のスクリプトは、名前に **DC** という文字列が大文字または小文字で含まれているすべてのデータセンター オブジェクトの名前と ID を返しています。スクリプトには、**tag** プロパティも取得します。

```
var datacenters = VcPlugin.getAllDatacenters(['tag'], "xpath:name[contains(translate(., 'DC', 'dc'), 'dc')]");
for each (datacenter in datacenters){
    System.log(datacenter.name + " " + datacenter.id);
}
```

vCenter Server プラグインのワークフロー ライブラリ

vCenter Server プラグインのワークフロー ライブラリには、vCenter Server の管理に関連する自動プロセスの実行に使用できるワークフローが含まれています。

■ バッチ ワークフロー

バッチ ワークフローは構成要素を入力したり、選択した vCenter Server オブジェクトに対してワークフローを実行したりします。

■ クラスタとコンピューティング リソースのワークフロー

クラスタとコンピューティング リソースのワークフローでは、クラスタを作成、名前変更、削除できます。また、クラスタで高可用性、Distributed Resource Scheduler、vCloud Distributed Storage を有効または無効にすることもできます。

■ 構成ワークフロー

vCenter Server プラグインの「構成」ワークフロー カテゴリには、vCenter Server インスタンスへの接続を管理するためのワークフローが含まれています。

■ カスタム属性のワークフロー

カスタム属性のワークフローでは、仮想マシンにカスタム属性を追加したり、仮想マシンのカスタム属性を取得したりすることができます。

■ データセンターのワークフロー

データセンターのワークフローでは、データセンターの作成、削除、再ロード、名前の変更、再スキャンが行えます。

■ データストアとファイルのワークフロー

データストアとファイルのワークフローでは、ファイル リストを削除したり、データストア内の未使用ファイルを検索したりすることができます。

■ データセンター フォルダ管理のワークフロー

データセンター フォルダ管理のワークフローでは、データセンター フォルダの作成、削除、名前の変更が行えます。

■ ホスト フォルダ管理のワークフロー

ホスト フォルダ管理のワークフローでは、ホスト フォルダの作成、削除、名前の変更が行えます。

■ 仮想マシン フォルダ管理のワークフロー

仮想マシン フォルダ管理のワークフローでは、仮想マシン フォルダの作成、削除、名前の変更が行えます。

■ ゲスト操作ファイルのワークフロー

ゲスト操作ファイルのワークフローを使用して、ゲスト OS のファイルを管理できます。

■ ゲスト操作プロセスのワークフロー

ゲスト操作プロセスのワークフローを使用して、ゲスト OS の情報を取得し、ゲスト OS でのプロセスの実行を制御できます。

■ ホスト電力管理のワークフロー

ホスト電力管理のワークフローでは、ホストを再起動したり、シャットダウンしたりすることができます。

■ 基本的なホスト管理のワークフロー

基本的なホスト管理のワークフローを使用すると、ホストをメンテナンス モードにしたり、ホストでメンテナンス モードを終了したりすることができます。また、ホストをフォルダやクラスタに移動して、ホストからデータを再ロードすることもできます。

■ ホスト登録管理のワークフロー

ホスト登録管理のワークフローでは、クラスタにホストを追加したり、クラスタとホストの接続を切断または再接続したりすることができます。

■ ネットワークのワークフロー

ネットワークのワークフローでは、分散仮想スイッチにポート グループを追加したり、ポート グループで分散仮想スイッチを作成したりすることができます。

■ 分散仮想ポート グループのワークフロー

分散仮想ポート グループのワークフローでは、ポート グループをアップデートまたは削除したり、そのポート グループを再構成したりすることができます。

■ 分散仮想スイッチのワークフロー

分散仮想スイッチのワークフローでは、分散仮想スイッチの作成、アップデート、削除をしたり、プライベート VLAN の作成、削除、アップデートをしたりすることができます。

■ 標準仮想スイッチのワークフロー

標準仮想スイッチのワークフローでは、標準仮想スイッチの作成、アップデート、削除をしたり、標準仮想スイッチ内のポート グループの作成、削除、アップデートをしたりすることができます。

■ Virtual SAN ネットワークのワークフロー

Virtual SAN ネットワークのワークフローでは、Virtual SAN ネットワーク トラフィックを構成できます。

■ リソース プールのワークフロー

リソース プールのワークフローでは、リソース プールの作成、名前の変更、再構成または削除を行ったり、リソース プール情報を取得したりすることができます。

■ ストレージのワークフロー

ストレージのワークフローでは、ストレージ関連の操作を実行できます。

■ ストレージ DRS のワークフロー

ストレージ DRS のワークフローでは、データストア クラスタの作成と構成、クラスタからのデータストアの削除、クラスタへのストレージの追加など、ストレージ関連の操作を行います。

■ ストレージ VSAN のワークフロー

ストレージ VSAN のワークフローでは、Virtual SAN クラスタ内の SSD 以外のディスクとディスク グループを管理できます。

■ 基本的な仮想マシン管理のワークフロー

基本的な仮想マシン管理のワークフローでは、たとえば仮想マシンの作成、名前の変更、削除、仮想ハードウェアのアップグレードなど、仮想マシンでの基本的な操作を行うことができます。

■ クローンのワークフロー

クローンのワークフローでは、仮想マシンのプロパティをカスタマイズしても、またはしなくても、仮想マシンのクローンを作成できます。

■ リンク クローンのワークフロー

リンク クローンのワークフローでは、リンク クローンからの仮想マシンのリストアやリンク クローンの作成など、リンク クローンの処理を実行できます。

■ Linux カスタマイズ クローンのワークフロー

Linux カスタマイズのワークフローでは、Linux 仮想マシンのクローンを作成したり、ゲスト OS のカスタマイズを行ったりできます。

■ ツール クローンのワークフロー

ツール クローンのワークフローでは、仮想マシンの OS に関するカスタマイズ情報や、仮想デバイスのアップデートに必要な情報などを取得できます。

■ Windows カスタマイズ クローンのワークフロー

Windows カスタマイズ クローンのワークフローでは、Windows 仮想マシンのクローンを作成したり、ゲスト OS のカスタマイズを行ったりすることができます。

■ デバイス管理のワークフロー

デバイス管理のワークフローでは、仮想マシンまたはホストのデータストアに接続されているデバイスを管理することができます。

■ 移動と移行のワークフロー

移動と移行のワークフローでは、仮想マシンを移行することができます。

■ その他のワークフロー

その他のカテゴリのワークフローでは、Fault Tolerance (FT) を有効および無効にしたり、仮想マシン情報を取得したり、親なしの仮想マシンを検索したりすることができます。

■ 電力管理のワークフロー

電力管理のワークフローでは、仮想マシンのパワーオンおよびパワーオフを行ったり、仮想マシンのゲスト OS を再起動したり、仮想マシンをサスペンドしたりすることができます。

■ スナップショットのワークフロー

スナップショットのワークフローでは、スナップショット関連の操作を実行できます。

■ VMware Tools のワークフロー

VMware Tools のワークフローでは、VMware Tools 関連のタスクを仮想マシンで実行できます。

バッチ ワークフロー

バッチ ワークフローは構成要素を入力したり、選択した vCenter Server オブジェクトに対してワークフローを実行したりします。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **batch** タグを入力します。

ワークフロー名	説明
バッチ構成要素の入力	<p>[選択したオブジェクトに対するワークフローの実行] ワークフローで使用する構成要素を入力します。次のタスクを実行します。</p> <ul style="list-style-type: none"> ■ BatchObject および BatchAction 構成要素をリセットします。 ■ 入力パラメータを1つのみ持つすべてのワークフローの BatchObject 構成要素を入力します。 ■ 入力パラメータを持たないか1つのみ持ち、returnType の配列を持つすべてのアクションの BatchAction 構成要素を入力します。
選択したオブジェクトに対するワークフローの実行	<p>選択した vCenter Server オブジェクトに対し、1つのアクションを入力として取得してワークフローを実行します。これはワークフローの実行対象であるオブジェクトのリストを取得するアクションです。選択したワークフローを実行しないでオブジェクトを返すには、ワークフローをシミュレーション モードで実行します。</p>

クラスタとコンピューティング リソースのワークフロー

クラスタとコンピューティング リソースのワークフローでは、クラスタを作成、名前変更、削除できます。また、クラスタで高可用性、Distributed Resource Scheduler、vCloud Distributed Storage を有効または無効にすることもできます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **cluster_and_compute_resource** タグを入力します。

ワークフロー名	説明
クラスタへの DRS 仮想マシン グループの追加	DRS 仮想マシン グループをクラスタに追加します。
DRS グループへの仮想マシンの追加	仮想マシン リストを既存の DRS 仮想マシン グループに追加します。
クラスタの作成	クラスタをホスト フォルダに作成します。
クラスタの削除	クラスタを削除します。
クラスタでの DRS の無効化	クラスタで DRS を無効にします。
クラスタでの HA の無効化	クラスタで高可用性を無効にします。
クラスタでの vCloud Distributed Storage の無効化	クラスタで vCloud Distributed Storage を無効にします。
クラスタでの DRS の有効化	クラスタで DRS を有効にします。
クラスタでの HA の有効化	クラスタで高可用性を有効にします。
クラスタでの vCloud Distributed Storage の有効化	クラスタで vCloud Distributed Storage を有効にします。
クラスタからの DRS 仮想マシン グループの削除	クラスタから DRS 仮想マシン グループを削除します。
DRS グループからの仮想マシンの削除	クラスタ DRS グループから仮想マシンを削除します。
クラスタ名の変更	クラスタ名を変更します。

構成ワークフロー

vCenter Server プラグインの「構成」ワークフロー カテゴリには、vCenter Server インスタンスへの接続を管理するためのワークフローが含まれています。

Orchestrator クライアントの [ワークフロー] ビューにある [ライブラリ] - [vCenter] - [構成] からこれらのワークフローにアクセスできます。

ワークフロー名	説明
vCenter Server インスタンスの追加	Orchestrator が新しい vCenter Server インスタンスに接続するように構成し、vSphere インフラストラクチャ内のオブジェクトに対してワークフローを実行できるようにします。
vCenter Server の Orchestrator エクステンションの一覧表示	vCenter Server のすべての Orchestrator エクステンションを一覧表示します。
Orchestrator を vCenter Server のエクステンションとして登録	Orchestrator インスタンスを vCenter Server エクステンションとして登録します。
vCenter Server インスタンスの削除	vCenter Server インスタンスを Orchestrator インベントリから削除します。この vCenter Server インスタンスは統合できなくなります。
vCenter Server インスタンスの更新	vCenter Server インスタンスへの接続を更新します。たとえば、vCenter Server システムの IP アドレスが変更された場合、vCenter Server インスタンスへの接続パラメータを更新することにより、Orchestrator を使用して vSphere インベントリを管理できるようにする必要があります。
vCenter Server エクステンションの登録解除	vCenter Server エクステンションを登録解除します。

カスタム属性のワークフロー

カスタム属性のワークフローでは、仮想マシンにカスタム属性を追加したり、仮想マシンのカスタム属性を取得したりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **custom_attributes** タグを入力します。

ワークフロー名	説明
仮想マシンへのカスタム属性の追加	仮想マシンにカスタム属性を追加します。
複数の仮想マシンへのカスタム属性の追加	選択した複数の仮想マシンにカスタム属性を追加します。
カスタム属性の取得	vCenter Server 内の仮想マシンのカスタム属性を取得します。

データセンターのワークフロー

データセンターのワークフローでは、データセンターの作成、削除、再ロード、名前の変更、再スキャンが行えます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **datacenter** タグを入力します。

ワークフロー名	説明
データセンターの作成	データセンターをデータセンター フォルダに作成します。
データセンターの削除	データセンターを削除します。
データセンターの再ロード	vCenter Server でデータをデータセンターから強制的に再ロードします。
データセンター名の変更	データセンター名を変更してタスクの完了を待機します。
データセンター HBA の再スキャン	データセンター内のホストをスキャンし、新しいストレージを検出するためにホスト バス アダプタの再スキャンを開始します。

データストアとファイルのワークフロー

データストアとファイルのワークフローでは、ファイル リストを削除したり、データストア内の未使用ファイルを検索したりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **datastore_and_files** タグを入力します。

ワークフロー名	説明
すべてのファイルの削除	ファイルのリストを削除します。
すべての未使用データストア ファイルの削除	vCenter Server 環境内のすべてのデータストアを検索し、すべての未使用ファイルを削除します。
未使用データストア ファイルのエクスポート	すべてのデータストアを検索し、すべての未使用ファイルをリストした XML 記述子ファイルを作成します。
データストア内の未使用ファイルの検索	Orchestrator に登録されているどの vCenter Server インスタンスとも関連付けられていない、すべての未使用ディスク (*.vmdk)、仮想マシン (*.vmx)、テンプレート (*.vmtx) ファイルを vCenter Server 環境から検索します。

ワークフロー名	説明
仮想マシンからのすべての構成、テンプレート、ディスク ファイルの取得	すべてのデータストアについて、すべての仮想マシン記述子ファイルのリストと、すべての仮想マシン ディスク ファイルのリストを作成します。
すべてのデータストア ファイルのロギング	すべてのデータストアで検出された、すべての仮想マシン構成ファイルとすべての仮想マシン ファイルのログを作成します。
未使用データストア ファイルのロギング	仮想マシンに登録されている未使用ファイルを vCenter Server 環境から検索し、これらのファイルのログをテキスト ファイルとしてエクスポートします。
データストアへのファイルのアップロード	特定のデータストア上にある既存のフォルダにファイルをアップロードします。アップロードしたファイルにより、アップロード先のフォルダに存在する同じ名前を持つ既存のファイルが上書きされます。

データセンター フォルダ管理のワークフロー

データセンター フォルダ管理のワークフローでは、データセンター フォルダの作成、削除、名前の変更が行えます。これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **datacenter_folder** タグを入力します。

ワークフロー名	説明
データセンター フォルダの作成	データセンター フォルダを作成します。
データセンター フォルダの削除	データセンター フォルダを削除してタスクの完了を待機します。
データセンター フォルダ名の変更	データセンター フォルダ名を変更してタスクの完了を待機します。

ホスト フォルダ管理のワークフロー

ホスト フォルダ管理のワークフローでは、ホスト フォルダの作成、削除、名前の変更が行えます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **host_folder** タグを入力します。

ワークフロー名	説明
ホスト フォルダの作成	ホスト フォルダを作成します。
ホスト フォルダの削除	ホスト フォルダを削除してタスクの完了を待機します。
ホスト フォルダ名の変更	ホスト フォルダ名を変更してタスクの完了を待機します。

仮想マシン フォルダ管理のワークフロー

仮想マシン フォルダ管理のワークフローでは、仮想マシン フォルダの作成、削除、名前の変更が行えます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **vm_folder** タグを入力します。

ワークフロー名	説明
仮想マシン フォルダの作成	仮想マシン フォルダを作成します。
仮想マシン フォルダの削除	仮想マシン フォルダを削除してタスクの完了を待機します。
仮想マシン フォルダ名の変更	仮想マシン フォルダ名を変更してタスクの完了を待機します。

ゲスト操作ファイルのワークフロー

ゲスト操作ファイルのワークフローを使用して、ゲスト OS のファイルを管理できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**guest_operations** および **files** タグを入力します。

ワークフロー名	説明
ゲスト内のディレクトリのチェック	ゲスト仮想マシンにディレクトリが存在することを確認します。
ゲスト内のファイルのチェック	ゲスト仮想マシンにファイルが存在することを確認します。
ゲストから Orchestrator へのファイルのコピー	指定したファイルをゲスト ファイル システムから Orchestrator サーバにコピーします。
Orchestrator からゲストへのファイルのコピー	指定したファイルを Orchestrator サーバからゲスト ファイル システムにコピーします。
ゲスト内のディレクトリの作成	ゲスト仮想マシン内にディレクトリを作成します。
ゲスト内の一時ディレクトリの作成	ゲスト仮想マシン内に一時ディレクトリを作成します。
ゲスト内の一時ファイルの作成	ゲスト仮想マシン内に一時ファイルを作成します。
ゲスト内のディレクトリの削除	ゲスト仮想マシンからディレクトリを削除します。
ゲスト内のファイルの削除	ゲスト仮想マシンからファイルを削除します。
ゲスト内のパスの一覧表示	ゲスト仮想マシン内のパスを表示します。
ゲスト内のディレクトリの移動	ゲスト仮想マシン内のディレクトリを移動します。
ゲスト内のファイルの移動	ゲスト仮想マシン内のファイルを移動します。

ゲスト操作プロセスのワークフロー

ゲスト操作プロセスのワークフローを使用して、ゲスト OS の情報を取得し、ゲスト OS でのプロセスの実行を制御できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**guest_operations** および **processes** タグを入力します。

ワークフロー名	説明
ゲストからの環境変数の取得	ゲストからの環境変数のリストを返します。対話形式のセッションでは、現在ログインしているユーザーの変数を返します。
ゲストからのプロセスの取得	ゲスト OS で実行されているプロセスと、API によって開始され、最近完了したプロセスのリストを返します。
ゲスト内でのプロセスの強制終了	ゲスト OS でプロセスを終了します。
ゲスト内でのプログラムの実行	ゲスト OS でプログラムを起動します。

ホスト電力管理のワークフロー

ホスト電力管理のワークフローでは、ホストを再起動したり、シャットダウンしたりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**host_management** および **power** タグを入力します。

ワークフロー名	説明
ホストの再起動	ホストを再起動します。Orchestrator クライアントがホストに直接接続されている場合、ホストとの接続が切断され、返されたタスクの成功を示すメッセージが表示されません。
ホストのシャットダウン	ホストをシャットダウンします。Orchestrator クライアントがホストに直接接続されている場合、ホストとの接続が切断され、返されたタスクの成功を示すメッセージが表示されません。

基本的なホスト管理のワークフロー

基本的なホスト管理のワークフローを使用すると、ホストをメンテナンス モードにしたり、ホストでメンテナンス モードを終了したりすることができます。また、ホストをフォルダやクラスタに移動して、ホストからデータを再ロードすることもできます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**host_management** および **basic** タグを入力します。

ワークフロー名	説明
メンテナンス モードへの切り替え	ホストをメンテナンス モードにします。タスクはキャンセル可能です。
メンテナンス モードの終了	メンテナンス モードを終了します。タスクはキャンセル可能です。
クラスタへのホストの移動	既存のホストをクラスタに移動します。ホストは同一データセンターの一部である必要があります。クラスタの一部であるホストは、メンテナンス モードである必要があります。
フォルダへのホストの移動	ホストをスタンドアロン ホストとしてフォルダに移動します。ホストは同一データセンター内の ClusterComputeResource の一部である必要があり、メンテナンス モードである必要があります。
ホストの再ロード	vCenter Server でデータをホストから強制的に再ロードします。

ホスト登録管理のワークフロー

ホスト登録管理のワークフローでは、クラスタにホストを追加したり、クラスタとホストの接続を切断または再接続したりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**host_management** および **registration** タグを入力します。

ワークフロー名	説明
クラスタへのホストの追加	ホストをクラスタに追加します。ホストの SSL 証明書を認証できない場合、このワークフローは失敗します。
スタンドアロン ホストの追加	ホストをスタンドアロン ホストとして登録します。
ホストの切断	vCenter Server インスタンスからホストを切断します。
ホストの再接続	ホストの情報を提供するだけで、切断されたホストを再接続します。
すべての情報によるホストの再接続	ホストに関するすべての情報を提供することで、切断されたホストを再接続します。
ホストの削除	ホストを削除して vCenter Server インスタンスから登録解除します。クラスタの一部であるホストは、削除前にメンテナンス モードにする必要があります。

ネットワークのワークフロー

ネットワークのワークフローでは、分散仮想スイッチにポート グループを追加したり、ポート グループで分散仮想スイッチを作成したりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **networking** タグを入力します。

ワークフロー名	説明
分散仮想スイッチへのポート グループの追加	指定した分散仮想スイッチに新しい分散仮想ポート グループを追加します。
分散仮想スイッチへのホスト システムの追加	分散仮想スイッチにホストを追加します。
ポート グループによる分散仮想スイッチの作成	分散仮想ポート グループを使用して新しい分散仮想スイッチを作成します。

分散仮想ポート グループのワークフロー

分散仮想ポート グループのワークフローでは、ポート グループをアップデートまたは削除したり、そのポート グループを再構成したりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**networking** および **distributed_virtual_port_group** タグを入力します。

ワークフロー名	説明
分散仮想ポート グループへの仮想マシン NIC 番号の接続	指定した仮想マシン NIC 番号のネットワーク接続を再構成し、指定した分散仮想ポート グループに接続します。NIC 番号を指定しない場合は、ゼロが使用されます。
分散仮想ポート グループの削除	指定した分散仮想ポート グループを削除します。
チーミング オプションの設定	分散仮想ポート グループのチーミング オプションを管理するためのインターフェイスを提供します。
分散仮想ポート グループのアップデート	指定した分散仮想ポート グループの構成をアップデートします。

分散仮想スイッチのワークフロー

分散仮想スイッチのワークフローでは、分散仮想スイッチの作成、アップデート、削除をしたり、プライベート VLAN の作成、削除、アップデートをしたりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**networking** および **distributed_virtual_switch** タグを入力します。

ワークフロー名	説明
分散仮想スイッチの作成	指定した名前とアップリンク ポート名を持つ分散仮想スイッチを指定のネットワーク フォルダに作成します。少なくとも 1 つのアップリンク ポート名を指定する必要があります。
プライベート VLAN の作成	指定した分散仮想スイッチに VLAN を作成します。
分散仮想スイッチの削除	分散仮想スイッチとすべての関連要素を削除します。
プライベート VLAN の削除	指定した分散仮想スイッチから VLAN を削除します。セカンダリ VLAN が存在する場合は、セカンダリ VLAN を先に削除する必要があります。
分散仮想スイッチのアップデート	分散仮想スイッチのプロパティをアップデートします。
プライベート VLAN のアップデート	指定した分散仮想スイッチの VLAN をアップデートします。

標準仮想スイッチのワークフロー

標準仮想スイッチのワークフローでは、標準仮想スイッチの作成、アップデート、削除をしたり、標準仮想スイッチ内のポート グループの作成、削除、アップデートをしたりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**networking** および **standard_virtual_switch** タグを入力します。

ワークフロー名	説明
標準仮想スイッチへのポート グループの追加	標準仮想スイッチにポート グループを追加します。
標準仮想スイッチの作成	標準仮想スイッチを作成します。
標準仮想スイッチからのポート グループの削除	標準仮想スイッチからポート グループを削除します。
標準仮想スイッチの削除	ホスト ネットワーク構成から標準仮想スイッチを削除します。
すべての標準仮想スイッチの取得	すべての標準仮想スイッチをホストから取得します。
標準仮想スイッチ内のポート グループのアップデート	標準仮想スイッチ内のポート グループのプロパティをアップデートします。
標準仮想スイッチのアップデート	標準仮想スイッチのプロパティをアップデートします。
標準仮想スイッチ内のポート グループの VNIC のアップデート	標準仮想スイッチ内のポート グループに関連付けられている仮想 NIC をアップデートします。

Virtual SAN ネットワークのワークフロー

Virtual SAN ネットワークのワークフローでは、Virtual SAN ネットワーク トラフィックを構成できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**networking**、および **vsan** タグを入力します。

ワークフロー名	説明
クラスタの VSAN トラフィック ネットワークの設定	クラスタの Virtual SAN トラフィック ネットワークを設定します。
ホストの VSAN トラフィック ネットワークの設定	ホストの Virtual SAN トラフィック ネットワークを設定します。

リソース プールのワークフロー

リソース プールのワークフローでは、リソース プールの作成、名前の変更、再構成または削除を行ったり、リソース プール情報を取得したりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **resource_pool** タグを入力します。

ワークフロー名	説明
リソース プールの作成	デフォルトの CPU とメモリ割り当て値のリソース プールを作成します。クラスタにリソース プールを作成するには、クラスタで VMware DRS が有効化されている必要があります。
指定した値によるリソース プールの作成	指定した CPU とメモリ割り当て値のリソース プールを作成します。クラスタにリソース プールを作成するには、クラスタで VMware DRS が有効化されている必要があります。
リソース プールの削除	リソース プールを削除し、タスクの完了を待機します。
リソース プール情報の取得	指定したリソース プールの CPU とメモリの情報を返します。

ワークフロー名	説明
リソース プールの再構成	指定したリソース プールの CPU とメモリ割り当ての設定を再構成します。
リソース プール名の変更	リソース プール名を変更してタスクの完了を待機します。

ストレージのワークフロー

ストレージのワークフローでは、ストレージ関連の操作を実行できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **storage** タグを入力します。

ワークフロー名	説明
iSCSI/FC/ローカル SCSI へのデータストアの追加	ファイバ チャネル、iSCSI、またはローカル SCSI ディスクにデータストアを作成します。新しいデータストアを作成できるのは、既存の VMFS が現在使用していないディスクのみです。新しいデータストアは、指定したディスクの最大使用可能領域を割り当てます。
NFS へのデータストアの追加	NFS サーバにデータストアを追加します。
iSCSI ターゲットの追加	iSCSI ターゲットを vCenter Server ホストに追加します。ターゲットのタイプは Send または Static です。
すべての使用可能ディスクの VMFS の作成	指定したホストのすべての使用可能ディスクの VMFS ボリュームを作成します。
データストアの削除	vCenter Server ホストからデータストアを削除します。
iSCSI ターゲットの削除	すでに構成されている iSCSI ターゲットを削除します。ターゲットのタイプは Send または Static です。
iSCSI アダプタの無効化	指定したホストのソフトウェア iSCSI アダプタを無効にします。
すべてのデータストアとディスクの表示	指定したホストの既存のデータストアと使用可能ディスクを表示します。
iSCSI アダプタの有効化	iSCSI アダプタを有効にします。
すべてのストレージ アダプタの一覧表示	指定したホストのすべてのストレージ アダプタを一覧表示します。

ストレージ DRS のワークフロー

ストレージ DRS のワークフローでは、データストア クラスタの作成と構成、クラスタからのデータストアの削除、クラスタへのストレージの追加など、ストレージ関連の操作を行います。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter** および **storage_drs** タグを入力します。

ワークフロー名	説明
クラスタへのデータストアの追加	データストア クラスタにデータストアを追加します。データストアは、データストア クラスタに含まれるすべてのホストに接続できる必要があります。データストア クラスタに存在するデータストアの接続タイプは、同一である必要があります。
仮想マシンごとのストレージ DRS 構成の変更	各仮想マシンのストレージ DRS の設定を構成します。
データストア クラスタの構成	データストア クラスタの自動化およびランタイム ルールの設定値を構成します。
シンプル データストア クラスタの作成	デフォルト構成のシンプルなデータストア クラスタを作成します。新しいデータストア クラスタには、データストアは含まれません。

ワークフロー名	説明
ストレージ DRS スケジュール設定タスクの作成	データストア クラスタを再構成するためのスケジュール設定タスクを作成します。設定できるのは、自動化およびランタイム ルールのみです。
仮想マシン非アフィニティ ルールの作成	非アフィニティ ルールを作成して、特定の仮想マシンのすべての仮想ディスクは別々のデータストアに存在する必要があることを示します。
VMDK 非アフィニティ ルールの作成	仮想マシンの VMDK 非アフィニティ ルールを作成して、どの仮想ディスクを別のデータストアに保存する必要があるかを示します。ルールは、選択した仮想マシンの仮想ディスクに適用されます。
データストア クラスタの削除	データストア クラスタを削除します。データストア クラスタを削除すると、そのクラスタのすべての設定とアラームも vCenter Server システムから削除されます。
クラスタからのデータストアの削除	データストア クラスタからデータストアを削除し、データストアをデータストア フォルダに配置します。
ストレージ DRS スケジュール設定タスクの削除	ストレージ DRS のスケジュール設定タスクを削除します。
仮想マシンの非アフィニティ ルールの削除	指定したデータストア クラスタの仮想マシン非アフィニティ ルールを削除します。
VMDK 非アフィニティ ルールの削除	指定したデータストア クラスタの VMDK 非アフィニティ ルールを削除します。

ストレージ VSAN のワークフロー

ストレージ VSAN のワークフローでは、Virtual SAN クラスタ内の SSD 以外のディスクとディスク グループを管理できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**storage** および **vsan** タグを入力します。

ワークフロー名	説明
ディスク グループへのディスクの追加	SSD 以外のディスクを Virtual SAN ディスク グループに追加します。
ディスク グループへのディスクの配布要求	Virtual SAN システムで使用するディスクを要求し、自動的にディスク グループを作成して、ディスクを既存のディスク グループに配布します。
ディスク グループの作成	Virtual SAN ディスク グループを作成します。
ホスト、ディスク グループ、ディスクの一覧表示	クラスタ内のすべてのホストと、Virtual SAN システムで使用されているか使用可能なディスク グループとディスクを一覧表示します。
ディスク グループの削除	Virtual SAN ディスク グループを削除します。
ディスク グループからのディスクの削除	SSD 以外のディスクを Virtual SAN ディスク グループから削除します。

基本的な仮想マシン管理のワークフロー

基本的な仮想マシン管理のワークフローでは、たとえば仮想マシンの作成、名前の変更、削除、仮想ハードウェアのアップグレードなど、仮想マシンでの基本的な操作を行うことができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management** および **basic** タグを入力します。

ワークフロー名	説明
カスタム仮想マシンの作成	指定した構成オプションと追加デバイスを持つ仮想マシンを作成します。
シンプル dvPortGroup 仮想マシンの作成	シンプルな仮想マシンを作成します。使用ネットワークは分散仮想ポート グループです。

ワークフロー名	説明
シンプル仮想マシンの作成	最も一般的なデバイスと構成オプションを持つ仮想マシンを作成します。
仮想マシンの削除	インベントリとデータストアから仮想マシンを削除します。
名前による仮想マシンの取得	登録されているすべての vCenter Server インスタンスから指定した表現と一致する仮想マシンを検索し、そのリストを返します。
テンプレートとしてマーキング	既存の仮想マシンをテンプレートに変換し、起動させないようにします。テンプレートは、仮想マシンの作成に使用できます。
仮想マシンとしてマーキング	既存のテンプレートを仮想マシンに変換し、起動できるようにします。
フォルダへの仮想マシンの移動	指定した仮想マシン フォルダに仮想マシンを移動します。
リソース プールへの仮想マシンの移動	仮想マシンをリソース プールに移動します。ターゲット リソース プールが同じクラスタに存在しない場合は、移行または再配置のワークフローを使用する必要があります。
フォルダへの仮想マシンの移動	指定した仮想マシン フォルダに複数の仮想マシンを移動します。
リソース プールへの仮想マシンの移動	複数の仮想マシンをリソース プールに移動します。
仮想マシンの登録	仮想マシンを登録します。仮想マシン ファイルが既存のデータストアに配置されていて、かつ未登録の状態である必要があります。
仮想マシンの再ロード	vCenter Server で強制的に仮想マシンを再ロードします。
仮想マシン名の変更	データストアではなく、vCenter Server システムまたはホストに存在する仮想マシンの名前を変更します。
仮想マシン パフォーマンスの設定	共有、最小値、最大値、ネットワークのシェーピング、ディスク アクセスなど、仮想マシンのパフォーマンス設定を変更します。
仮想マシンの登録解除	既存の仮想マシンをインベントリから削除します。
仮想マシン ハードウェアのアップグレード (必要に応じて強制)	仮想マシンのハードウェアを、ホストがサポートする最新版にアップグレードします。このワークフローは、VMware Tools が古くなっている場合でも、強制的にアップグレードを継続します。VMware Tools が古くなっているときにアップグレードを強制的に継続すると、ゲスト ネットワークの設定はデフォルト設定に戻されます。このような事態を回避するには、ワークフローの実行前に VMware Tools をアップグレードします。
仮想マシンのアップグレード	仮想ハードウェアを、ホストがサポートする最新版にアップグレードします。入力パラメータにより、VMware Tools が古くなっている場合でもアップグレードを強制できます。
タスクの待機と仮想マシンの質問に対する回答	vCenter Server タスクの完了、または仮想マシンによる質問を待機します。仮想マシンが回答を必要とする場合は、ユーザー入力を受け付け、質問に回答します。

クローンのワークフロー

クローンのワークフローでは、仮想マシンのプロパティをカスタマイズしても、またはしなくても、仮想マシンのクローンを作成できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management**、および **clone** タグを入力します。

ワークフロー名	説明
プロパティによる仮想マシンのクローン作成	プロパティを入力パラメータとして使用して、仮想マシンをクローン作成します。
カスタマイズなしの仮想マシンのクローン作成	仮想マシン UUID 以外は何も変更せずに仮想マシンをクローン作成します。
プロパティによる仮想マシンのカスタマイズ	プロパティを入力パラメータとして使用して、仮想マシンをカスタマイズします。

リンク クローンのワークフロー

リンク クローンのワークフローでは、リンク クローンからの仮想マシンのリストアやリンク クローンの作成など、リンク クローンの処理を実行できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management** および **linked_clone** タグを入力します。

ワークフロー名	説明
リンク クローン、複数の NIC がある Linux	Linux 仮想マシンのリンク クローンを作成し、ゲスト OS のカスタマイズと最大で 4 つの仮想ネットワーク カードの構成を行います。
リンク クローン、1 つの NIC がある Linux	Linux 仮想マシンのリンク クローンを作成し、ゲスト OS のカスタマイズと 1 つの仮想ネットワーク カードの構成を行います。
リンク クローン、カスタマイズなし	仮想マシンの指定数のリンク クローンを作成します。
リンク クローン、複数の NIC と認証情報がある Windows	Windows 仮想マシンのリンク クローンを作成し、ゲスト OS のカスタマイズを行います。最大で 4 つの仮想ネットワーク カードとローカル管理ユーザー アカウントを構成します。
リンク クローン、単一の NIC と認証情報がある Windows	Windows 仮想マシンのリンク クローンを作成し、ゲスト OS のカスタマイズを行います。1 つの仮想ネットワーク カードとローカル管理者のユーザー アカウントを構成します。
リンク クローンからの仮想マシンのリストア	リンク クローンのセットアップから仮想マシンを削除します。
リンク クローン向けの仮想マシンのセットアップ	リンク クローンとなる仮想マシンを準備します。

Linux カスタマイズ クローンのワークフロー

Linux カスタマイズのワークフローでは、Linux 仮想マシンのクローンを作成したり、ゲスト OS のカスタマイズを行ったりできます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management**、**clone** および **linux_customization** タグを入力します。

ワークフロー名	説明
クローン、複数の NIC がある Linux	Linux 仮想マシンのクローンを作成し、ゲスト OS のカスタマイズと最大で 4 つの仮想ネットワーク カードの構成を行います。
クローン、単一の NIC がある Linux	Linux 仮想マシンのクローンを作成し、ゲスト OS のカスタマイズと 1 つの仮想ネットワーク カードの構成を行います。

ツール クローンのワークフロー

ツール クローンのワークフローでは、仮想マシンの OS に関するカスタマイズ情報や、仮想デバイスのアップデートに必要な情報などを取得できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、**vcenter**、**virtual_machine_management**、**clone** および **tools** タグをワークフローの検索ボックスに入力します。

ワークフロー名	説明
ネットワークを変更するための VirtualEthernetCard の取得	仮想デバイスをアップデートするための新しいイーサネット カードを返します。指定した仮想デバイスのデバイス キーと新しいネットワークのみが含まれます。
Linux カスタマイズ情報の取得	Linux カスタマイズの準備情報を返します。
複数の VirtualEthernetVard デバイスの変更の取得	VirtualEthernetCard オブジェクトに対する追加および削除操作のための一連の VirtualDeviceConfigSpec オブジェクトを返します。
NIC 設定マップの取得	VimAdapterMapping を使用して、仮想ネットワーク カードの設定マップを返します。仮想マシンのクローン作成および再構成を行うワークフローの NIC 情報を変更します。他のクローン ワークフローはこのワークフローを呼び出します。
認証情報による Windows カスタマイズ、Sysprep の取得	認証情報を使用して、Microsoft Sysprep プロセスに関するカスタマイズ情報を返します。Windows 仮想マシンのクローンを作成するワークフローは、このワークフローを使用します。
Unattended.txt による Windows カスタマイズ情報、Sysprep の取得	Unattended.txt ファイルを使用して、Microsoft Sysprep プロセスに関するカスタマイズ情報を返します。Windows 仮想マシンのクローンを作成するワークフローは、このワークフローを使用します。
Sysprep 向け Windows カスタマイズ情報の取得	Microsoft Sysprep プロセスに関するカスタマイズ情報を返します。Windows 仮想マシンのクローンを作成するワークフローは、このワークフローを使用します。

Windows カスタマイズ クローンのワークフロー

Windows カスタマイズ クローンのワークフローでは、Windows 仮想マシンのクローンを作成したり、ゲスト OS のカスタマイズを行ったりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management**、**clone** および **windows_customization** タグを入力します。

ワークフロー名	説明
単一の NIC と認証情報によるシンプロビジョニングされた Windows のクローン作成	Windows 仮想マシンのクローンを作成し、ゲスト OS のカスタマイズを行います。仮想ディスクのシンプロビジョニング ポリシーを指定し、1つのネットワーク カードとローカル管理者ユーザー アカウントを構成します。vCenter Server システムで Sysprep ツールを使用できる必要があります。
単一の NIC と認証情報による Windows Sysprep のクローン作成	Windows 仮想マシンのクローンを作成し、ゲスト OS のカスタマイズを行います。1つの仮想ネットワーク カードとローカル管理者のユーザー アカウントを構成します。vCenter Server システムで Sysprep ツールを使用できる必要があります。
複数の NIC と認証情報による Windows のクローン作成	Windows 仮想マシンのクローンを作成し、ゲスト OS のカスタマイズを行います。ローカル管理ユーザー アカウントと最大で 4 つの仮想ネットワーク カードを構成します。vCenter Server システムで Sysprep ツールを使用できる必要があります。
単一の NIC による Windows のクローン作成	Windows 仮想マシンのクローンを作成し、ゲスト OS のカスタマイズと1つの仮想ネットワーク カードの構成を行います。vCenter Server システムで Sysprep ツールを使用できる必要があります。
単一の NIC と認証情報による Windows のクローン作成	Windows 仮想マシンのクローンを作成し、ゲスト OS のカスタマイズを行います。1つの仮想ネットワーク カードとローカル管理者のユーザー アカウントを構成します。vCenter Server システムで Sysprep ツールを使用できる必要があります。
単一の NIC と認証情報による Windows のカスタマイズ	ゲスト OS のカスタマイズを行い、Windows 仮想マシンの1つの仮想ネットワーク カードとローカル管理ユーザー アカウントを構成します。

デバイス管理のワークフロー

デバイス管理のワークフローでは、仮想マシンまたはホストのデータストアに接続されているデバイスを管理することができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management** および **device_management** タグを入力します。

ワークフロー名	説明
CD-ROM の追加	仮想マシンに仮想 CD-ROM を追加します。仮想マシンに IDE コントローラがない場合は、ワークフローによって作成されます。
ディスクの追加	仮想マシンに仮想ディスクを追加します。
RAM の変更	仮想マシンの RAM 容量を変更します。
シン プロビジョニングするディスクの変換	仮想マシンのシック プロビジョニングされたディスクをシン プロビジョニング ディスクに変換します。
独立ディスクの変換	ディスクから独立フラグを削除することで、仮想マシンの独立ディスクをすべて通常ディスクに変換します。
実行中仮想マシンからのすべての分離可能デバイスの切断	実行中の仮想マシンからフロッピー ディスク、CD-ROM ドライブ、パラレル ポート、シリアル ポートを切断します。
CD-ROM のマウント	仮想マシンの CD-ROM をマウントします。仮想マシンに IDE コントローラまたは CD-ROM ドライブがない場合は、ワークフローによって作成されます。
フロッピー ディスク ドライブのマウント	ESX データストアからフロッピー ディスク ドライブの FLP ファイルをマウントします。

移動と移行のワークフロー

移動と移行のワークフローでは、仮想マシンを移行することができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management** および **move_and_migrate** タグを入力します。

ワークフロー名	説明
Storage vMotion による仮想マシンの一括移行	Storage vMotion を使用して、単一の仮想マシン、選択した複数の仮想マシン、または使用可能なすべての仮想マシンを移行します。
vMotion による仮想マシンの一括移行	vMotion、Storage vMotion、またはその両方を使用して、単一の仮想マシン、選択した複数の仮想マシン、または使用可能なすべての仮想マシンを移行します。 注： vCenter Server では、パワーオン状態の仮想マシンと同じパスで Storage vMotion と vMotion を使用することはできません。同じパスで Storage vMotion と vMotion を使用するには、仮想マシンをパワーオフする必要があります。
vMotion による仮想マシンの移行	vSphere API からの MigrateVM_Task 操作を使用して、あるホストから別のホストに仮想マシンを移行します。
別の vCenter Server への仮想マシンの移動	仮想マシンのリストを別の vCenter Server システムに移動します。
複数の仮想マシンの迅速移行	パワーオン状態にある場合は仮想マシンをサスペンドし、同じストレージを使用してそれを別のホストに移行します。

ワークフロー名	説明
仮想マシンの迅速な移行	パワーオン状態にある場合は仮想マシンをサスペンドし、同じストレージを使用してそれを別のホストに移行します。
仮想マシンディスクの再配置	vSphere API からの ReLocateVM_Task 操作を使用して、仮想マシンがパワーオフ状態にある間に仮想マシンディスクを別のホストまたはデータストアに再配置します。

その他のワークフロー

その他のカテゴリのワークフローでは、Fault Tolerance (FT) を有効および無効にしたり、仮想マシン情報を取得したり、親なしの仮想マシンを検索したりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management**、および **others** タグを入力します。

ワークフロー名	説明
FT の無効化	指定した仮想マシンの Fault Tolerance を無効にします。
FT の有効化	指定した仮想マシンの Fault Tolerance を有効にします。
仮想マシン情報の抽出	指定した仮想マシンの仮想マシン フォルダ、ホスト システム、リソース プール、コンピューティング リソース、データストア、ハード ドライブ サイズ、CPU とメモリ、ネットワーク、IP アドレスを返します。VMware Tools が必要になる場合があります。
親なし状態の仮想マシンの検出	Orchestrator インベントリ内で親なし状態にあるすべての仮想マシンを一覧表示します。Orchestrator インベントリ内のどの仮想マシンとも関連性を持たない、Orchestrator インベントリ内のすべてのデータストアの VMDK および VMTX ファイルを一覧表示します。オプションとして、リストを E メールで送信します。
名前と BIOS UUID による仮想マシンの取得	名前で仮想マシンを検索した後、特定のユニバーサル意識別子 (UUID) を使用して結果をフィルタし、一意の仮想マシンを特定します。 注: このワークフローが必要となるのは、特定の DynamicOps と vRealize Orchestrator 仮想マシン間で送受信を行うため、VC:VirtualMachine タイプの入力パラメータを持つ vRealize Orchestrator ワークフローを DynamicOps が呼び出す場合です。
名前と UUID による仮想マシンの取得	名前で仮想マシンを検索した後、特定のユニバーサル意識別子 (UUID) を使用して結果をフィルタし、一意の仮想マシンを特定します。 注: このワークフローが必要となるのは、特定の DynamicOps と vRealize Orchestrator 仮想マシン間で送受信を行うため、VC:VirtualMachine タイプの入力パラメータを持つ vRealize Orchestrator ワークフローを DynamicOps が呼び出す場合です。
仮想マシンの UUID の取得	名前で仮想マシンを検索した後、特定のユニバーサル意識別子 (UUID) を使用して結果をフィルタし、一意の仮想マシンを特定します。 注: このワークフローが必要となるのは、特定の DynamicOps と vRealize Orchestrator 仮想マシン間で送受信を行うため、VC:VirtualMachine タイプの入力パラメータを持つ vRealize Orchestrator ワークフローを DynamicOps が呼び出す場合です。

電力管理のワークフロー

電力管理のワークフローでは、仮想マシンのパワーオンおよびパワーオフを行ったり、仮想マシンのゲスト OS を再起動したり、仮想マシンをサスペンドしたりすることができます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management** および **power_management** タグを入力します。

ワークフロー名	説明
仮想マシンのパワーオフと待機	仮想マシンをパワーオフし、プロセスの完了を待機します。
ゲスト OS の再起動	仮想マシンのゲスト OS を再起動します。読み取り専用仮想マシンのリセットは行いません。VMware Tools を実行している必要があります。
仮想マシンのリセットと待機	仮想マシンをリセットし、プロセスの完了を待機します。
仮想マシンの再開と待機	サスペンドされた仮想マシンを再開し、プロセスの完了を待機します。
ゲスト OS のスタンバイ モードへの設定	ゲスト OS をスタンバイ モードに設定します。VMware Tools を実行している必要があります。
仮想マシンのシャットダウンと削除	仮想マシンをシャットダウンし、インベントリおよびディスクから削除します。
ゲスト OS のシャットダウンと待機	ゲスト OS をシャットダウンし、プロセスの完了を待機します。
仮想マシンの起動と待機	仮想マシンを起動し、VMware Tools の起動を待機します。
仮想マシンのサスペンドと待機	仮想マシンをサスペンドし、プロセスの完了を待機します。

スナップショットのワークフロー

スナップショットのワークフローでは、スナップショット関連の操作を実行できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management** および **snapshot** タグを入力します。

ワークフロー名	説明
スナップショットの作成	スナップショットを作成します。
リソース プール内のすべての仮想マシンのスナップショットの作成	リソース プール内の各仮想マシンのスナップショットを作成します。
すべてのスナップショットの削除	既存のすべてのスナップショットを以前のスナップショットに戻さずに削除します。
余分なスナップショットの削除	指定数を超える数のスナップショットを持つ仮想マシンを検索し、オプションとして最も古いスナップショットを削除します。結果を E メールで送信します。
古いスナップショットの削除	指定日数より古いすべてのスナップショットを取得し、削除するスナップショットの選択をユーザーに求めます。
指定サイズのスナップショットの削除	指定サイズを超える大きさのすべてのスナップショットを取得し、削除の確認をユーザーに求めます。
現在のスナップショットまで戻る	現在のスナップショットまで戻ります。
スナップショットへの復帰と待機	特定のスナップショットまで戻ります。スナップショットの削除は行いません。

VMware Tools のワークフロー

VMware Tools のワークフローでは、VMware Tools 関連のタスクを仮想マシンで実行できます。

これらのワークフローにアクセスするには、[ライブラリ] - [ワークフロー] の順に移動して、ワークフローの検索ボックスに **vcenter**、**virtual_machine_management** および **vmware_tools** タグを入力します。

ワークフロー名	説明
ツール インストーラのマウント	VMware Tools インストーラを仮想 CD-ROM にマウントします。
コンソール画面解像度の設定	コンソール ウィンドウの解像度を設定します。仮想マシンがパワーオン状態である必要があります。

ワークフロー名	説明
時刻同期の有効化	VMware Tools により、仮想マシンと ESX サーバの間の時刻同期を有効にします。
ツール インストーラのアンマウント	VMware Tools CD-ROM をマウント解除します。
再起動せずに Windows 仮想マシンのツールを更新する	Windows 仮想マシンの VMware Tools を、マシンを再起動せずに更新します。
ツールのアップグレード	仮想マシンの VMware Tools をアップグレードします。
次の再起動時にツールをアップグレード	廃止：[再起動せずに Windows 仮想マシンのツールを更新する] ワークフローを使用

vCloud Suite API (vAPI) プラグインの使用

18

vCloud Suite API プラグインを使用すると、vCloud Suite API プロバイダが公開する API を使用できるようになります。vCloud Suite API は、vCloud Suite Endpoint を使用して vCenter Server に要求を発行することにより、仮想環境内のリソースにアクセスするためのサービス指向アーキテクチャを提供します。

このプラグインには標準ワークフローとサンプル ワークフローのセットが含まれています。仮想環境のタスクを自動化するプラグインを実装するためのカスタム ワークフローを作成することもできます。vCloud Suite API の詳細については、『VMware vCloud Suite SDKs Programming Guide』を参照してください。

この章には、次のトピックが含まれています。

- [vCloud Suite API プラグインの構成](#)
- [vCloud Suite API プラグインの API へのアクセス](#)

vCloud Suite API プラグインの構成

vCloud Suite API を構成するには、プラグインに組み込まれている構成ワークフローを実行します。

vCloud Suite API メタモデルのインポート

vCloud Suite API プラグインは、vCloud Suite API プロバイダのメタデータ サービスにクエリを出すことで、vCloud Suite API サービスを見つけます。メタデータ サービスを表示していない vCloud Suite API プロバイダはサポートされません。

vCloud Suite API メタモデルをインポートし、後でエンドポイントを追加する必要があります。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **vapi** タグを入力します。
- 3 [vAPI メタモデルのインポート] ワークフローを見つけて、[実行] をクリックします。
- 4 [vAPI エンドポイントの URL] テキスト ボックスに、vCloud Suite API エンドポイントの URL を入力します。

5 セキュア プロトコル接続を使用するかどうかを選択します。

オプション	説明
いいえ	セキュア プロトコル接続を使用せずに vCloud Suite API メタモデルをインポートします。
はい	セキュア プロトコル接続を使用して vCloud Suite API メタモデルをインポートするには、次のうちいずれかの操作を行います。 <ol style="list-style-type: none"> 証明書の警告を無視するかどうかを選択し、vCloud Suite エンドポイントを自動的に受け入れます。 vCloud Suite エンドポイントへの認証に使用するユーザー認証情報を指定します。

6 同じ認証情報を使用して vAPI エンドポイントを追加するかどうかを選択します。

7 [実行] をクリックします。

次のステップ

[vCloud Suite API エンドポイントの追加](#)

vCloud Suite API エンドポイントの追加

vCloud Suite API エンドポイントを追加します。

前提条件

vCloud Suite API メタモデルをインポートします。

手順

- vRealize Orchestrator クライアントに管理者としてログインします。
- [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **vapi** タグを入力します。
- [VAPI エンドポイントの追加] ワークフローを見つけて、[実行] をクリックします。
- [vAPI エンドポイントの URL] テキスト ボックスに、vCloud Suite API エンドポイントの URL を入力します。
- セキュア プロトコル接続を使用するかどうかを選択します。

オプション	説明
いいえ	セキュア プロトコル接続を使用せずに vCloud Suite API メタモデルをインポートします。
はい	セキュア プロトコル接続を使用して vCloud Suite API メタモデルをインポートするには、次のうちいずれかの操作を行います。 <ol style="list-style-type: none"> 証明書の警告を無視するかどうかを選択し、vCloud Suite エンドポイントを自動的に受け入れます。 vCloud Suite エンドポイントへの認証に使用するユーザー認証情報を指定します。

6 [実行] をクリックします。

vCloud Suite API プラグインの API へのアクセス

vRealize Orchestrator に用意されている API Explorer を使用すると、vCloud Suite API プラグインの API を検索したり、スクリプト化された要素で使用できる JavaScript オブジェクトのドキュメントを参照したりすることができます。

手順

- 1 vRealize Orchestrator クライアントに管理者としてログインします。
- 2 vRealize Orchestrator クライアントまたはワークフロー、ポリシー、アクション エディタの [スクリプティング] タブから API Explorer にアクセスします。
 - vRealize Orchestrator クライアントから API Explorer にアクセスするには、vRealize Orchestrator クライアントのナビゲーション ペインで [API Explorer] をクリックします。
 - ワークフロー、ポリシー、アクション エディタの [スクリプティング] タブから API Explorer にアクセスするには、左側にある [API の検索] をクリックします。
- 3 vCloud Suite API プラグインの API オブジェクトの階層リストを展開するには、左側のペインで [VAPI] モジュールをダブルクリックします。

次のステップ

API 要素からコードをコピーしてスクリプト処理ボックスに貼り付けることができます。API スクリプティングの詳細については、『VMware vRealize Orchestrator における開発』を参照してください。

XML プラグインの使用

19

XML プラグインを使用すると、XML ドキュメントを作成、変更するワークフローを実行できます。

XML プラグインは、Orchestrator の JavaScript API にドキュメント オブジェクト モデル (DOM) XML パーサーの実装を追加します。また、ワークフローから XML ドキュメントを作成して変更する方法を示すサンプル ワークフローもいくつか提供します。

または、Orchestrator の JavaScript API で ECMAScript for XML (E4X) 実装を使用し、JavaScript で直接 XML ドキュメントを処理することができます。E4X スクリプトの例については、『vRealize Orchestrator を使用したワークフロー開発』を参照してください。

E4X の詳細については、ECMA-357 標準を管理する組織の Web サイトをご覧ください。

この章には、次のトピックが含まれています。

- [XML プラグイン サンプル ワークフローの実行](#)

XML プラグイン サンプル ワークフローの実行

vRealize Orchestrator クライアントから XML プラグインのサンプル ワークフローを実行して、テスト用に XML ドキュメントを作成および変更することができます。

ワークフローではファイルの作成、読み込み、変更などを行うため、作業ディレクトリに対する十分なアクセス権を持っている必要があります。

Orchestrator には、サーバ システムのルートに、orchestrator という名前のフォルダに対する読み取り、書き込みおよび実行の権限があります。ワークフローにはこのフォルダに対する読み取り、書き込みおよび実行の権限がありますが、サーバ システムにフォルダを作成する必要があります。Orchestrator Appliance を使用する場合、フォルダの名前は vco で、/var/run/vco に配置します。

ワークフローや JavaScript からサーバ ファイル システムのアクセス権設定を変更することで、その他のフォルダへのアクセスを許可することができます。「VMware vRealize Orchestrator のインストールと構成」、「ワークフローやアクションからのサーバ ファイル システムのアクセス権の設定」を参照してください。

- [簡易 XML ドキュメントの作成](#)

vRealize Orchestrator クライアントからワークフローを実行して、テスト用の簡易な XML ドキュメントを作成することができます。

■ XML ドキュメントの要素の検索

vRealize Orchestrator クライアントからワークフローを実行して、簡易 XML ドキュメントの作成ワークフローで作成された要素を XML 内で検索します。

■ XML ドキュメントの変更

vRealize Orchestrator クライアントからワークフローを実行して、簡易 XML ドキュメントのワークフローで作成した XML を変更します。

■ XML からのサンプルアドレス帳の作成

vRealize Orchestrator クライアントからワークフローを実行して、テスト用にアドレス帳を作成することができます。

簡易 XML ドキュメントの作成

vRealize Orchestrator クライアントからワークフローを実行して、テスト用の簡易な XML ドキュメントを作成することができます。

前提条件

- ログイン時に使用したユーザー アカウントに XML ワークフローの実行に必要な権限があることを確認します。
- Orchestrator サーバ システムのルートに `C:/orchestrator` フォルダを作成してあること、または他のフォルダへのアクセス権を設定してあることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに `xml` および `samples_xml_(simple)` タグを入力します。
- 3 [簡易 XML ドキュメントの作成] ワークフローを見つけて、[実行] をクリックします。
- 4 作成する XML ドキュメントのファイルパスを入力します。
たとえば、`c:/orchestrator/filename.xml` のように入力します。
- 5 [実行] をクリックします。

結果

ワークフローによって、ユーザーのリストが含まれた XML ドキュメントが作成されます。各エントリの属性は、`user ID` および `name` です。

XML ドキュメントの要素の検索

vRealize Orchestrator クライアントからワークフローを実行して、簡易 XML ドキュメントの作成ワークフローで作成された要素を XML 内で検索します。

前提条件

- ログイン時に使用したユーザー アカウントに XML ワークフローの実行に必要な権限があることを確認します。

- Orchestrator サーバ システムのルートに C:/orchestrator フォルダを作成してあること、または他のフォルダへのアクセス権を設定してあることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **xml** および **samples_xml_(simple)** タグを入力します。
- 3 [ドキュメントの要素を検索] ワークフローを見つけて、[実行] をクリックします。
- 4 XML ドキュメントのファイルパスを入力します。
たとえば、c:/orchestrator/*filename.xml* のように入力します。
- 5 [実行] をクリックします。
ワークフローは要素を検索し、システム ログに結果を表示します。

次のステップ

結果を表示するには、vRealize Orchestrator クライアントで完了したワークフローの実行を選択し、[スキーマ] タブの [ログ] をクリックします。

XML ドキュメントの変更

vRealize Orchestrator クライアントからワークフローを実行して、簡易 XML ドキュメントのワークフローで作成した XML を変更します。

前提条件

- ログイン時に使用したユーザー アカウントに XML ワークフローの実行に必要な権限があることを確認します。
- Orchestrator サーバ システムのルートに C:/orchestrator フォルダを作成してあること、または他のフォルダへのアクセス権を設定してあることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **xml** および **samples_xml_(simple)** タグを入力します。
- 3 [XML ドキュメントの変更] ワークフローを見つけて、[実行] をクリックします。

- 4 入力および出力ファイルパスを指定します。
 - a 変更する XML ドキュメントのファイルパスを入力します。
たとえば、`c:/orchestrator/filename.xml` のように入力します。
 - b 変更済み XML ドキュメントのファイルパスを入力します。
たとえば、`c:/orchestrator/filename.xml` のように入力します。

注： 両方のフィールドに同じファイルパスを入力した場合、ワークフローは元のファイルを変更済みファイルで上書きします。存在しないファイルの出力ファイルパスを指定した場合、ワークフローは変更済みファイルを新規に作成します。

- 5 [実行] をクリックします。

結果

ワークフローは要素を検索し、要素が見つかった場所のエントリを修正します。

XML からのサンプルアドレス帳の作成

vRealize Orchestrator クライアントからワークフローを実行して、テスト用にアドレス帳を作成することができます。

前提条件

- ログイン時に使用したユーザー アカウントに XML ワークフローの実行に必要な権限があることを確認します。
- Orchestrator サーバ システムのルートに `C:/orchestrator` フォルダを作成してあること、または他のフォルダへのアクセス権を設定してあることを確認します。

手順

- 1 vRealize Orchestrator クライアントへのログイン
- 2 [ライブラリ] - [ワークフロー] の順に移動し、ワークフローの検索ボックスに **xml** および **samples_xml_ (address_book)** タグを入力します。
- 3 [アドレス帳の全テスト] ワークフローを見つけて、[実行] をクリックします。
- 4 アドレス帳のフォルダのパスを入力します。
たとえば、`c:/orchestrator/foldername` のように入力します。
フォルダが存在しない場合は、自動的に作成されます。
- 5 [実行] をクリックします。

結果

ワークフローによって、スタイルシートが付加された DTD、XML、および CSS ファイルが作成され、ファイルが指定のフォルダに保存されます。