

VMware vRealize Orchestrator プラグインの 使用

vRealize Orchestrator 7.5

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 © 2008-2018 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 10

1 Orchestrator プラグインの概要 11

Orchestrator のアーキテクチャ 11

Orchestrator Server とともにインストールされるプラグイン 12

Orchestrator API Explorer へのアクセス 15

2 Orchestrator プラグインの構成 16

Orchestrator プラグインの管理 16

プラグインのアンインストール 17

3 vCenter Server プラグインの使用 19

vCenter Server プラグインの構成 19

構成ワークフロー 20

vCenter Server インスタンスへの接続の構成 20

vCenter Server プラグインのスクリプト API 22

vCenter Server プラグイン インベントリの使用 22

クエリのパフォーマンスに関する考慮事項 22

vCenter Server プラグインを使用した XPath 式の使用 23

vCenter Server プラグインを使用した XPath 式の使用 23

vCenter Server プラグインのワークフロー ライブラリへのアクセス 24

vCenter Server プラグインのワークフロー ライブラリ 25

バッチ ワークフロー 28

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 28

構成ワークフロー 29

カスタム属性のワークフロー 29

データセンターのワークフロー 29

データストアとファイルのワークフロー 30

データセンター フォルダ管理のワークフロー 30

ホスト フォルダ管理のワークフロー 31

仮想マシン フォルダ管理のワークフロー 31

ゲスト操作ファイルのワークフロー 31

ゲスト操作プロセスのワークフロー 32

ホスト電力管理のワークフロー 32

基本的なホスト管理のワークフロー 32

ホスト登録管理のワークフロー 33

ネットワークのワークフロー 33

分散仮想ポート グループのワークフロー 33

分散仮想スイッチのワークフロー	34
標準仮想スイッチのワークフロー	34
Virtual SAN ネットワークのワークフロー	35
リソース プールのワークフロー	35
ストレージのワークフロー	35
ストレージ DRS のワークフロー	36
ストレージ VSAN のワークフロー	36
基本的な仮想マシン管理のワークフロー	37
クローンのワークフロー	38
リンク クローンのワークフロー	38
Linux カスタマイズ クローンのワークフロー	39
ツール クローンのワークフロー	39
Windows カスタマイズ クローンのワークフロー	39
デバイス管理のワークフロー	40
移動と移行のワークフロー	41
その他のワークフロー	41
電力管理のワークフロー	42
スナップショットのワークフロー	42
VMware Tools のワークフロー	43

4 vRealize Automation プラグインの使用 44

vRealize Automation 用 VMware vRealize Orchestrator プラグインの概要	44
vRealize Orchestrator の vRealize Automation プラグインにおける役割	45
vRealize Automation プラグインの構成	45
構成ワークフロー	45
vRealize Automation プラグインのワークフローの使用	48
操作制限の削除	48
vRealize Automation プラグイン インベントリの使用	51
vRealize Automation プラグインの管理ワークフローの使用	51
vRealize Automation プラグインのインフラストラクチャ管理ワークフローの使用	57
vRealize Automation プラグインの申請ワークフローの使用	60
vRealize Automation プラグインのサンプル ワークフローの使用	61
vRealize Automation プラグインの API へのアクセス	62
サンプル vRealize Automation プラグイン スクリプト	62
CRUD インフラストラクチャ管理サンプル スクリプト	62
vRealize Automation エンティティのサンプル スクリプトの検索	66
vRealize Automation がプロビジョニングしたリソースを取得するためのサンプル スクリプト	68
一般的なタスクのサンプル スクリプト	69

5 Configuration プラグインの使用 72

Configuration プラグインのワークフロー ライブラリへのアクセス	72
--	----

Configuration プラグインのワークフロー ライブラリ 72

6 Library プラグインの使用 74

Library プラグインのワークフロー 74

7 SQL プラグインの使用 76

SQL プラグインの構成 76

SQL プラグインの構成ワークフロー 76

データベースの追加 77

データベースへのテーブルの追加 78

データベースの更新 78

SQL サンプル ワークフローの実行 79

JDBC URL の生成 79

JDBC 接続のテスト 80

JDBC を使用したテーブルの作成 80

JDBC テーブルへの行の挿入 81

JDBC テーブルから行を選択 81

JDBC テーブルからエントリを削除 82

JDBC テーブルからすべてのエントリを削除 83

JDBC テーブルのドロップ 83

完全な JDBC サイクルの実行 84

SQL プラグインの標準ワークフローの使用 84

SQL プラグインのワークフロー ライブラリ 85

テーブルの CRUD ワークフローの生成 85

8 SSH プラグインの使用 87

SSH プラグインの構成 87

構成ワークフロー 88

SSH プラグイン サンプル ワークフローの実行 88

キー ペアの作成 89

キー ペアのパスフレーズの変更 90

Orchestrator パブリック キーの SSH ホストへの登録 90

SSH コマンドの実行 91

SSH ホストからのファイルのコピー 91

SSH ホストへのファイルのコピー 92

9 XML プラグインの使用 93

XML プラグイン サンプル ワークフローの実行 93

簡易 XML ドキュメントの作成 94

XML ドキュメントの要素の検索 94

XML ドキュメントの変更 95

[XML からのサンプルアドレス帳の作成](#) 96

10 Mail プラグインの使用 97

[デフォルトの SMTP 接続の定義](#) 97

[Mail プラグイン サンプル ワークフローの使用](#) 98

[Mail プラグイン サンプル ワークフローへのアクセス](#) 98

[Mail プラグイン サンプル ワークフロー](#) 98

11 Net プラグインの使用 100

12 Enumeration プラグインの使用 101

[タイム ゾーン コード](#) 101

13 Workflow Documentation プラグインの使用 105

[Workflow Documentation プラグインのワークフロー ライブラリ](#) 105

[ワークフロー ドキュメントの生成](#) 105

14 HTTP-REST プラグインの使用 107

[HTTP-REST プラグインの構成](#) 107

[構成ワークフロー](#) 107

[Kerberos 認証の構成](#) 108

[REST ホストの追加](#) 109

[REST 操作の追加](#) 111

[REST ホストへのスキーマの追加](#) 112

[REST 操作からの新しいワークフローの生成](#) 112

[REST 操作の呼び出し](#) 113

[REST 操作の呼び出し](#) 113

15 SOAP プラグインの使用 115

[SOAP プラグインの構成](#) 115

[構成ワークフロー](#) 115

[SOAP ホストの追加](#) 116

[Kerberos 認証の構成](#) 117

[SOAP 操作からの新しいワークフローの生成](#) 119

[カスタム生成したワークフローのテスト](#) 119

[SOAP 操作の呼び出し](#) 120

16 AMQP プラグインの使用 121

[AMQP プラグインの構成](#) 121

[構成ワークフロー](#) 121

[ブローカの追加](#) 121

キューへの登録	122
ブローカの更新	123
AMQP プラグインの標準ワークフローの使用	123
バインドの宣言	124
キューの宣言	124
Exchange の宣言	125
テキスト メッセージの送信	126
バインドの削除	127
17 SNMP プラグインの使用	128
SNMP デバイスの管理	128
デバイス管理ワークフロー	128
SNMP デバイスの登録	129
SNMP クエリの管理	130
クエリ管理ワークフロー	130
SNMP デバイスへのクエリの追加	130
SNMP トラップ ホストの管理	131
トラップ ホスト管理ワークフロー	131
SNMP トラップ ポートの設定	131
SNMP トラップの受信	132
SNMP デバイスでのトラップの待機	132
SNMP トラップ ポリシーの設定	133
SNMP トラップ ホスト ポリシーの構成	133
トラップ ポリシーの編集	134
汎用 SNMP 要求ワークフロー	135
18 Active Directory プラグインの使用	136
Active Directory プラグインの構成	136
Active Directory 構成ワークフロー	136
Active Directory プラグインのワークフロー ライブラリの使用	137
Active Directory プラグイン インベントリの使用	137
Active Directory プラグインのワークフロー ライブラリへのアクセス	137
Active Directory プラグインのワークフロー	137
19 Dynamic Types プラグインの使用	140
Dynamic Types 構成ワークフロー	140
20 PowerShell プラグインの使用	142
VMware vRealize OrchestratorPowerShell プラグインの概要	142
PowerShell プラグイン コンポーネント	143
WinRM の構成	145

Kerberos 認証の構成	148
PowerShell プラグインの構成	150
構成ワークフロー	150
PowerShell ホストの追加	150
PowerShell プラグイン インベントリの使用	151
PowerShell スクリプトの実行	152
PowerShell スクリプトの呼び出し	152
外部スクリプトの呼び出し	152
アクションの生成	153
PowerShell スクリプトからのアクションの生成	153
PowerShell コマンドレットのアクションの生成	154
アクション間の呼び出し結果の受け渡し	155
PowerShell プラグインによる PowerCLI の統合	155
コンバータ ワークフロー	156
サンプル ワークフロー	156
PowerShell プラグインの API へのアクセス	157
PowerShell の結果の使用	157
一般的な PowerShell タスクのスクリプトのサンプル	158
トラブルシューティング	160
Kerberos イベント ログの有効化	160
Kerberos データベースでサーバが見つからない	160
Kerberos チケットを取得できない	161
Kerberos 認証が時刻設定の違いにより失敗する	161
Kerberos 認証セッション モードが失敗する	162
レルムのキー配布センターに到達できない	162
デフォルトのレルムが見つからない	163

21 Multi-Node プラグインの使用 164

vRealize Orchestrator Multi-Node プラグインの概要	164
Multi-Node プラグインの構成	165
サーバ構成ワークフロー	165
Orchestrator サーバの追加	165
プロキシ ワークフローの使用	166
同期プロキシ ワークフロー	166
非同期プロキシ ワークフロー	167
リモート実行ワークフロー	167
Multi-Node プラグイン インベントリの使用	168
リモート管理ワークフロー	168
Multi-Node プラグインの API へのアクセス	169
マルチノード プラグインの使用事例	169
マルチプロキシ アクションの作成	170

[リモートおよびプロキシ ワークフローのメンテナンス](#) 171

[ローカル サーバからパッケージを展開](#) 171

22 vCloud Suite API (vAPI) プラグインの使用 173

[vCloud Suite API プラグインの構成](#) 173

[vCloud Suite API メタモデルのインポート](#) 173

[vCloud Suite API エンドポイントの追加](#) 174

[vCloud Suite API プラグインの API へのアクセス](#) 175

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。

Orchestrator プラグインの概要

1

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。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。

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、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。Orchestrator はオープン スタンドアロンを実装して外部システムとの統合を簡素化します。カスタム コンテンツの開発の詳細については、『VMware vRealize Orchestrator を使用した開発』を参照してください。

プラグインの標準セットは Orchestrator サーバとともに自動的にインストールされます。使用する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 Orchestrator のスクリプト エンジンを拡張します。また、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 Orchestrator 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。

- [Orchestrator のアーキテクチャ](#)
- [Orchestrator Server とともにインストールされるプラグイン](#)
- [Orchestrator API Explorer へのアクセス](#)

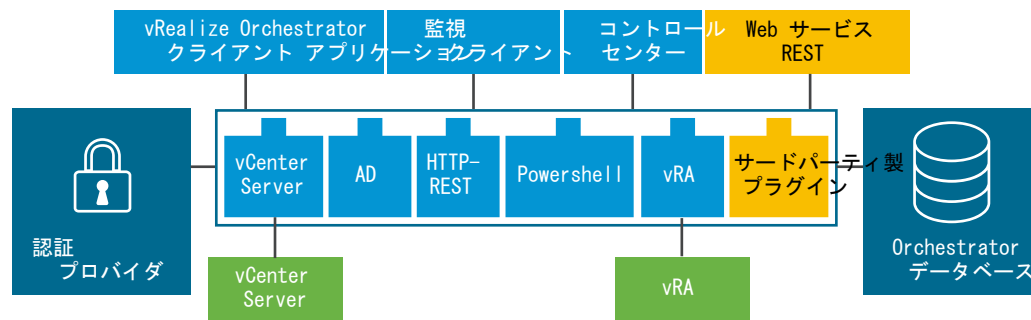
Orchestrator のアーキテクチャ

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、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します。

Orchestrator には、vCenter Server および vRealize Automation 向けのプラグインを含む、プラグインの標準のセットが用意されており、プラグインが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。

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。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、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。Orchestrator、Orchestrator で公開するオブジェクト、および Orchestrator ワークフローに、Orchestrator クライアント インターフェイスまたは Web サービスを介してアクセスできます。Orchestrator ワークフローおよびサービスの監視と構成は、監視クライアントとコントロール 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



Orchestrator Server とともにインストールされるプラグイン

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 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。設定が必要なプラグインについては、インターフェイスの個別のタブがあります。

表 1-1. Orchestrator とともにインストールされるプラグイン

プラグイン	目的	構成
vCenter Server	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。
構成	Orchestrator の認証、データベース接続、SSL 証明書などを構成するためのワークフローを提供します。	なし
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

表 1-1. Orchestrator とともにインストールされるプラグイン（続き）

プラグイン	目的	構成
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 を提供します。	なし
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、Orchestrator クライアント 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 プラグインの構成 を参照してください。
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、Orchestrator JavaScript API で ECMAScript for XML (E4X) 実装を使用できます。	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。
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 Net プラグインを組み合わせ、ワークフローに電子メールの包括的な送信機能と受信機能を提供します。	なし
ワークフロー ドキュメント	ワークフローまたはワークフロー カテゴリに関して PDF 形式で情報を生成するために使用できるワークフローを提供します。	なし
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。	なし
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。
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。
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。
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。	なし
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。
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。	19 章 Dynamic Types プラグインの使用 を参照してください。

表 1-1. Orchestrator とともにインストールされるプラグイン（続き）

プラグイン	目的	構成
Multi-Node	階層管理、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。	21 章 Multi-Node プラグインの使用 を参照してください。
PowerShell	PowerShell ホストを管理し、カスタム PowerShell 操作を実行できます。	20 章 PowerShell プラグインの使用 を参照してください。

プラグイン コンポーネント

各プラグインは、DAR ファイル パッケージ内にあります。DAR ファイルは、Orchestrator Appliance の /var/lib/vco/app-server/plugins に格納されています。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	DAR ファイル	ワークフロー カテゴリ	API モジュール
vCenter Server	o11nplugin-vsphere.dar	vCenter	VC
vRO 構成	o11nplugin-configurator.dar	構成	Configurator
ライブラリ	o11nplugin-library.dar	ロック Orchestrator トラブルシューティング	適用外。
SQL	o11nplugin-database.dar	JDBC SQL	SQL
SSH	o11nplugin-ssh.dar	SSH	SSH
XML	o11nplugin-xml.dar	XML	XML
メール	o11nplugin-mail.dar	メール	Mail
ネットワーク	o11nplugin-jakartacommonsnet.dar	なし	Net
ワークフロー ドキュメント	o11nplugin-wfdocs.dar	ワークフロー ドキュメント	ワークフロー ドキュメント
一般的な列挙値	o11nplugin-enums.dar	なし	Enums
動的タイプ	o11n-plugin-dynamictypes.dar	構成	DynamicTypes
HTTP-REST	o11nplugin-rest.dar	構成	REST
SOAP	o11n-plugin-soap.dar	構成	SOAP
AMQP	o11n-plugin-amqp.dar	構成	AMQP
SNMP	o11n-plugin-snmp.dar	デバイス管理 クエリ管理 トラップ ホスト管理	SNMP

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	DAR ファイル	ワークフロー カテゴリ	API モジュール
Active Directory	o11nplugin-ad.dar	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
Orchestrator	o11nplugin-multi-node.dar	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
PowerShell	o11nplugin-powershell.dar	構成 生成 テンプレート	PowerShell

Orchestrator API Explorer へのアクセス

Orchestrator に用意されている API Explorer を使用すると、Orchestrator API を検索したり、スクリプト化された要素で利用できる 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。

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。

手順

- 1 Orchestrator クライアントにログインします。
- 2 [ツール] - [API Explorer] の順に選択します。

結果

API Explorer が表示されます。API Explorer を使用して 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

API Explorer を使用してスクリプト化可能な要素のスクリプトを記述します。

Orchestrator プラグインの構成

2

デフォルトの Orchestrator プラグインは、ワークフローからのみ構成できます。

デフォルトの Orchestrator プラグインを構成するには、Orchestrator クライアントから特定のワークフローを使用する必要があります。

この章には、次のトピックが含まれています。

- [Orchestrator プラグインの管理](#)
- [プラグインのアンインストール](#)

Orchestrator プラグインの管理

コントロール センターの [プラグインを管理] ページで、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。

プラグインのログ レベルの変更

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。

新しいプラグインのインストール

Orchestrator プラグインでは、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。Orchestrator Appliance にはプレインストールされたプラグインが含まれており、カスタム プラグインをインストールすることもできます。

すべての Orchestrator プラグインは、コントロール センターからインストールできます。使用できるファイルの拡張子は、.vmoapp および .dar です。.vmoapp ファイルは .dar ファイルをいくつか含むことができ、アプリケーションとしてインストールできる一方で、.dar ファイルには一つのプラグインに関連するすべてのリソースが含まれています。

プラグインの無効化

プラグインの名前の横にある [有効] チェック ボックスをオフにすると、プラグインを無効にすることができます。

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。Orchestrator のプラグインのアンインストールについては、[プラグインのアンインストール](#)を参照してください。

プラグインのアンインストール

プラグインはコントロール センターを使用して無効にできますが、このアクションによってプラグインが Orchestrator Appliance ファイル システムから削除されることはありません。プラグイン ファイルを削除するには、Orchestrator Appliance にログインし、プラグイン ファイルを手動で削除します。

手順

- 1 Orchestrator Appliance からプラグインを削除します。
 - a SSH を使用して Orchestrator Appliance に **root** としてログインします。
 - b テキスト エディタで `/etc/vco/app-server/plugins/_VSOPuginInstallationVersion.xml` ファイルを開きます。
 - c 削除するプラグインに対応するコード行を削除します。
 - d `/var/lib/vco/app-server/plugins` ディレクトリに移動します。
 - e 削除するプラグインが含まれている `.dar` アーカイブを削除します。

- 2 vRealize Orchestrator サービスを再起動します。

```
service vco-configurator restart && service vco-server restart
```

- 3 コントロール センターに **root** としてログインします。
- 4 [プラグインを管理] ページで、プラグインが削除されたことを確認します。
- 5 Orchestrator クライアントから、プラグインに関連するパッケージとフォルダを削除します。
 - a Orchestrator クライアントにログインします。
 - b 左上隅にあるドロップダウン メニューから、[設計]を選択します。
 - c [パッケージ] ビューをクリックします。
 - d 削除するパッケージを右クリックし、[コンテンツのあるエレメントを削除] を選択します。

注： 標準ライブラリ内のワークフローなど、読み取り専用の状態でロックされている Orchestrator 要素は削除されません。

- e 右上隅にある [ツール] メニューから、[ユーザー環境設定] を選択します。
[環境設定] コンテキスト メニューが開きます。
- f [全般] ページで、[許可される空でないフォルダを削除] チェック ボックスを選択します。
フォルダ全体を、そのサブフォルダとワークフローを含め、1 回のクリックで削除できるようになりました。
- g [ワークフロー] ビューをクリックします。
- h 削除するプラグインが含まれているフォルダを削除します。
- i [アクション] ビューをクリックします。
- j 削除するプラグインのアクション モジュールを削除します。

6 vRealize Orchestrator サービスを再起動します。

結果

プラグインに関連するカスタム ワークフロー、アクション、ポリシー、構成、設定、リソースがすべて削除されました。

vCenter Server プラグインの使用

3

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、ワークフローで利用できる 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。

vCenter Server プラグインには、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、Orchestrator 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 ライブラリへのアクセス](#)
- [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

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、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、Orchestrator クライアントから vCenter Server 構成ワークフローを実行します。

vSphere Web Client を使用して vSphere インベントリ内のオブジェクトを管理するには、vCenter Server と vSphere Web Client の両方が指している同一の vCenter Single Sign-On インスタンスと連携するように、Orchestrator サーバを構成する必要があります。また、Orchestrator を vCenter Server のエクステンションとして登録しておく必要があります。Orchestrator を vCenter Server のエクステンションとして登録する操作は、vCenter Server のエクステンションを管理する権限を持つユーザーを指定（ユーザー名とパスワードを入力）するときに行います。

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 vRealize Orchestrator エクステンションの一覧表示	vCenter Server のすべての vRealize 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	vSphere Web Client エクステンションを登録解除します。

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、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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [vCenter] - [構成] の順に展開し、[vCenter Server インスタンスの追加] ワークフローに移動します。
- 4 [vCenter Server インスタンスの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。

- 5 追加する vCenter Server インスタンスがインストールされたマシンの IP アドレスまたは DNS 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。

- 6 デフォルトのポート値 **443** は保持します。
- 7 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。
- 8 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、[次へ] をクリックします。
- 9 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。

- 10 vCenter Server システムへのユーザー アクセスの管理方法を選択します。

オプション	説明
一意のセッションの共有	<p>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際に使用する 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。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。通常、Orchestrator システム ワークフローを実行するために監視します。</p>
ユーザーごとのセッション	<p>vCenter Server との新しいセッションを作成します。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> <p>このオプションを選択するのは、vCenter Server が Active Directory ドメインに含まれている場合か、vCenter Server Single Sign-On が有効になっている場合のみにしてください。</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を持つ有効なユーザーである必要があります。</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。

- 11 (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。

- 12 (オプション) 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。

- 13 [送信] をクリックして、ワークフローを実行します。

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。[インベントリ] タブを使用して、認証要素を追加したり、vCenter Server オブジェクトに対してワークフローを実行したりすることができます。

[ユーザー環境設定] ツールの [インベントリ] タブから [インベントリでコンテキスト メニューを使用する] オプションを有効にすると、選択したインベントリ オブジェクトに対して実行可能なすべてのワークフローがポップアップメニューに表示されます。

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 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、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 ライブラリへのアクセス

vCenter Server プラグイン ワークフロー ライブラリの要素にアクセスするには、Orchestrator クライアントまたは vSphere Web Client を使用する必要があります。

前提条件

- vCenter Server インスタンスへの接続を構成します。
- ログインに使用しているユーザー アカウントに、vCenter Server ワークフローを実行するために必要な権限が付与されていることを確認します。

手順

- 1 Orchestrator クライアントで、左上隅にあるドロップダウン メニューから [設計] または [実行] を選択します。

- 2 Orchestrator クライアントの左側のペインにある [ワークフロー] ビューをクリックします。
- 3 階層リストで [ライブラリ] - [vCenter] の順に展開します。

次のステップ

ワークフロー ライブラリを確認します。

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。

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] - [バッチ] から [バッチ] ワークフローにアクセスできます。

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。

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] - [クラスタとコンピューティング リソース] から [クラスタとコンピューティング リソース] のワークフローにアクセスできます。

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。

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] - [カスタム属性] から [カスタム属性] 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
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。

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] - [データセンター] から [データセンター] 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
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。

ワークフロー名	説明
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。

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] - [データストアとファイル] から [データストアとファイル] 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
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。
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。

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] - [フォルダ管理] - [データセンター フォルダ] から [データ センター フォルダ管理] 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
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。

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] - [フォルダ管理] - [ホスト フォルダ] から [ホスト フォルダ管理] 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
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。

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] - [フォルダ管理] - [仮想マシン フォルダ] から [仮想マシン フォルダ管理] 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
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。

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] - [ゲスト操作] - [ファイル] から [ゲスト操作ファイル] 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
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。

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] - [ゲスト操作] - [プロセス] から [ゲスト操作プロセス] 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
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。
ゲスト内でのプログラムの実行	ゲスト OS でプログラムを起動します。
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。

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。

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] - [ホスト管理] - [電力] から [ホスト電力管理] ワークフローにアクセスできます。

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している場合は、ホストへの接続が失われ、操作の成功を示すタスクは返されません。

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している場合は、ホストへの接続が失われ、操作の成功を示すタスクは返されません。

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。

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] - [ホスト管理] - [基本] から [基本的なホスト管理] 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
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。

ワークフロー名	説明
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。

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] - [ホスト管理] - [登録] から [ホスト登録管理] 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
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。

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] - [ネットワーク] から [ネットワーク] 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
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。

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] - [ネットワーク] - [分散仮想ポート グループ] から [分散仮想ポート グループ] 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
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。

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] - [ネットワーク] - [分散仮想スイッチ] から [分散仮想スイッチ] 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
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。

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] - [ネットワーク] - [標準仮想スイッチ] から [標準仮想スイッチ] 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
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。

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] - [ネットワーク] - [VSAN] から [ネットワーク] 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
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。

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] - [リソース プール] から [リソース プール] 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
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。
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。

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] - [ストレージ] から [ストレージ] 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
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。

ワークフロー名	説明
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。

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] - [ストレージ] - [ストレージ DRS] から [ストレージ DRS] 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
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。
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。

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] - [ストレージ] - [VSAN] から [ネットワーク] 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
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。

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] - [仮想マシンの管理] - [基本] から [基本的な仮想マシン管理] 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
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。
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。

ワークフロー名	説明
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。

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] - [仮想マシンの管理] - [クローン] から [クローン] ワークフローにアクセスできます。

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。

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] - [仮想マシンの管理] - [クローン] - [リンク クローン] フォルダおよびそのサブフォルダから [リンク クローン] 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
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。
複数の NIC を持つ Linux マシンのリンク クローンの作成	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。
単一の NIC を持つ Linux マシンのリンク クローンの作成	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。
複数の NIC と認証情報を使用した Windows マシンのリンク クローンの作成	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。

ワークフロー名	説明
単一の NIC と認証情報を使用した Windows マシンのリンク クローンの作成	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。
カスタマイズなしでのリンク クローンの作成	仮想マシンの指定数のリンク クローンを作成します。

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。

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] - [仮想マシンの管理] - [クローン] - [Linux カスタマイズ] から [Linux カスタマイズ クローン] 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
複数の NIC を持つ Linux マシンのクローン作成	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。
単一の NIC を持つ Linux マシンのクローン作成	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。

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] - [仮想マシンの管理] - [クローン] - [ツール] から [ツール クローン] 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
ネットワークを変更する仮想イーサネット カードの取得	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。
複数の仮想イーサネット カード デバイス変更の取得	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。
認証情報による Sysprep 向け Windows カスタマイズ情報の取得	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 Sysprep 向け Windows カスタマイズ情報の取得	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。

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] - [仮想マシンの管理] - [クローン] - [Windows カスタマイズ] フォルダおよびそのサブフォルダから [Windows カスタマイズ クローン] 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
単一の NIC と認証情報による Windows マシンのカスタマイズ	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。
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 マシンのクローン作成	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 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 マシンの 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 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 マシンのクローン作成	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 マシンのクローン作成	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 マシンのクローン作成	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。

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。

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] - [仮想マシンの管理] - [デバイス管理] から [デバイス管理] 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
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	ESXi 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。

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] - [仮想マシンの管理] - [移動と移行] から [移動と移行] 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
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。
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 システム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。
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。
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。

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] - [仮想マシンの管理] - [その他] 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
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	<p>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。</p> <p>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。</p>

ワークフロー名	説明
名前と UUID による仮想マシンの取得	<p>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。</p> <p>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。</p>
仮想マシン UUID の取得	<p>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。</p> <p>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。</p>

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。

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] - [仮想マシンの管理] - [電力管理] から [電力管理] 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
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。

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] - [仮想マシンの管理] - [スナップショット] から [スナップショット] 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
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。

ワークフロー名	説明
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。

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] - [仮想マシンの管理] - [VMware Tools] から [VMware Tools] 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
VMware Tools 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。
時刻同期の有効化	VMware Tools により、仮想マシンと ESXi サーバの間の時刻同期を有効にします。
VMware Tools インストーラのマウント解除	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。
VMware Tools 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 VMware Tools をアップグレード	自動再起動を実行せずに、仮想マシンの VMware Tools をアップグレードします。

vRealize Automation プラグインの使用

4

vRealize Automation プラグインを使用すると、vRealize Automation から vRealize Orchestrator ワークフローを実行できます。

プラグインが提供するワークフローを使用すると、vRealize Automation からリソースを展開して管理できます。これらのワークフローのほかに、カスタム ワークフローを作成して実行できます。

この章には、次のトピックが含まれています。

- [vRealize Automation 用 VMware vRealize Orchestrator プラグインの概要](#)
- [vRealize Automation プラグインの構成](#)
- [vRealize Automation プラグインのワークフローの使用](#)
- [サンプル vRealize Automation プラグイン スクリプト](#)

vRealize Automation 用 VMware vRealize Orchestrator プラグインの概要

vRealize Automation 用の VMware vRealize Orchestrator プラグインは、vRealize Orchestrator と vRealize Automation 間の相互通信を可能にします。

vRealize Automation プラグインを使用して、以下の vRealize Automation 機能のワークフローを作成、実行できます。

- XaaS カスタム リソースおよびブループリントの管理
- カタログ アイテムおよびリソースの管理と要求
- 資格の構成
- ポリシー構成の承認
- 作業アイテムの相互通信
- vSphere および vCloud Director 仮想マシンのプロビジョニングとプロビジョニング後のアクション
- vRealize Automation IaaS モデル上での作成、読み込み、削除 (CRUD) の操作

vRealize Orchestrator の vRealize Automation プラグインにおける役割

Orchestrator クライアントを使用してワークフローを実行および作成し、プラグイン API にアクセスします。組み込みの vRealize Orchestrator インスタンスを、お使いの vRealize Automation インストールに使用するか、外部 vRealize Orchestrator サーバのいずれかに使用できます。

vRealize Orchestrator は、vRealize Automation プラグインを強力にします。vRealize Orchestrator は、開発およびプロセス自動化のプラットフォームであり、VMware クラウド スタックおよびサード パーティの技術を管理するための拡張ワークフローを提供します。

Orchestrator のオープンなプラグイン アーキテクチャにより、vRealize Orchestrator を各種の管理ソリューションと統合することができます。

vRealize Automation プラグインの構成

vRealize Automation ホストおよび IaaS ホストを追加して、プラグインを構成します。

構成ワークフロー

[構成] ワークフロー カテゴリ内のワークフローを使用して、vRealize Automation ホストを管理することができます。

vRealize Automation ホスト

Orchestrator クライアントの [ワークフロー] ビューに表示されるプラグイン ライブラリの [Configuration] サブディレクトリから、これらのワークフローにアクセスすることができます。

ワークフロー名	説明
vRA ホストの追加	vRealize Automation ホストをプラグイン インベントリに追加します。テナント管理タスクおよび管理タスクの場合は、[インベントリ] ビューを使用して、各テナントに対してワークフローを実行できます。テナントに対してプラグインの全機能を使用するには、テナントごとに専用の vRealize Automation ホストを作成します。
コンポーネント レジストリを使用した vRA ホストの追加	<p>[ユーザーごとのセッション] 接続を使用して、vRealize Automation ホストをプラグイン インベントリに追加します。vRealize Automation システム管理者の認証情報を使用して Orchestrator クライアントにログインしている必要があります。</p> <p>この機能を外部の vRealize Orchestrator サーバで使用するには、Orchestrator サーバを vRealize Automation コンポーネント レジストリに登録する必要があります。</p> <p>注： 外部 vRealize Orchestrator サーバをコンポーネント レジストリに登録するには、Orchestrator が vRealize Automation を認証プロバイダとして使用するよう設定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』を参照してください。</p>
vRA ホストの IaaS ホストの追加	選択した vRealize Automation ホストの IaaS ホストをプラグイン インベントリに追加します。
vRA ホストの削除	プラグイン インベントリから vRealize Automation ホストを削除します。

ワークフロー名	説明
vRA ホストの更新	プラグイン インベントリ内の vRealize Automation ホストを更新します。
vRA ホストの検証	vRealize Automation ホストとその接続を検証します。

注： vRealize Orchestrator サーバが vRealize Automation コンポーネント レジストリに登録されている場合は、vRealize Automation ホストが Default という名前で自動的に追加されます。Default ホストは、[ユーザーごとのセッション] 接続を使用して、デフォルト テナントに接続します。vRealize Automation のインストール時に組み込まれている Orchestrator サーバは、デフォルトで vRealize Automation コンポーネント レジストリに登録されます。

vRealize Automation IaaS ホスト

Orchestrator クライアントの [ワークフロー] ビューに表示されるプラグイン ライブラリの [Infrastructure Administration] - [Configuration][] サブディレクトリから、これらのワークフローにアクセスすることができます。

vRealize Automation のインストール時に組み込まれている vRealize Orchestrator サーバは、デフォルトで vRealize Automation コンポーネント レジストリに登録されます。

ワークフロー名	説明
IaaS ホストの追加	vRealize Automation IaaS ホストをプラグイン インベントリに追加します。このワークフローは、「vRA ホストの IaaS ホストの追加」と機能性は同じですが、vRealize Automation ホストが不要である点が異なります。
IaaS ホストの削除	プラグイン インベントリから vRealize Automation IaaS ホストを削除します。
IaaS ホストの更新	プラグイン インベントリ内の vRealize Automation IaaS ホストを更新します。
IaaS ホストの検証	vRealize Automation IaaS ホストとその接続を検証します。

vRealize Automation ホストの追加

vRealize Automation ホストを追加するワークフローを実行し、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 Orchestrator クライアントのドロップダウン メニューから、[実行] または [設計] を選択します。
- 2 [ワークフロー] ビューをクリックします。
- 3 [ライブラリ] - [vRealize Automation] - [構成] の順に展開します。
- 4 [vRA ホストの追加] ワークフローを右クリックして [ワークフローの開始] の順に選択します。
- 5 [ホスト名] テキスト ボックスに一意のホストの名前を入力します。
- 6 [ホスト URL] テキスト ボックスにホストの URL アドレスを入力します。

例： *https://hostname*。

- 7 (必須) [テナント] テキスト ボックスにテナントの名前を入力します。

テナント用のプラグインの全機能を使用するには、テナントごとに専用の vRealize Automation ホストを作成します。

- 8 ユーザーへの確認なしで自動的に SSL 証明書をインストールするかどうかを選択します。
- 9 (オプション) vRealize Orchestrator が vRealize Automation からの接続または応答を待機する時間を設定するには、[接続タイムアウト (秒)] および [操作タイムアウト (秒)] テキスト ボックスにタイムアウト間隔を入力します。
- 10 ホストへの接続の種類を [セッション モード] ドロップダウン メニューから選択します。

オプション	アクション
共有セッション	vRealize Automation ユーザーの認証情報を、[認証ユーザー名] と [認証パスワード] のテキスト ボックスに入力します。
ユーザー セッション単位	<p>現在ログインしているユーザーの認証情報を使用して接続します。vRealize Automation システム管理者の認証情報を使用して Orchestrator クライアントにログインしている必要があります。</p> <p>このオプションを外部の vRealize Orchestrator サーバで使用するには、Orchestrator サーバを vRealize Automation コンポーネント レジストリに登録する必要があります。</p> <p>注： 外部 vRealize Orchestrator サーバをコンポーネント レジストリに登録するには、Orchestrator が vRealize Automation を認証プロバイダとして使用するよう設定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』を参照してください。</p>

- 11 [送信] をクリックします。

次のステップ

vRealize Automation のインフラストラクチャ管理ホストを追加します。

laaS ホストの追加

vRealize Automation ホストの laaS ホストを追加するワークフローを実行し、接続パラメータを設定できます。

手順

- 1 Orchestrator クライアントのドロップダウン メニューから、[実行] または [設計] を選択します。
- 2 [ワークフロー] ビューをクリックします。
- 3 [ライブラリ] - [vRealize Automation] - [インフラストラクチャ管理] - [構成] の順に展開します。
- 4 [laaS ホストの追加] を右クリックして [ワークフローの開始] の順に選択します。
- 5 [vCAC ホスト] ドロップダウン メニューから、laaS ホストを構成する vRealize Automation ホストを選択します。
- 6 [ホスト名] テキスト ボックスに一意のホストの名前を入力します。
- 7 Model Manager がインストールされているマシンの URL を入力します。
例 : `https://model_manager_machine.com`
- 8 SSL 証明書をインストールするには、[はい] を選択します。
- 9 Model Manager マシンへのアクセスにプロキシを使用するには、[はい] を選択します。

このオプションを選択した場合は、次のページでプロキシ ホストとプロキシ ポートを入力する必要があります。

- 10 [次へ] をクリックします。
- 11 明示的プロキシを設定している場合は、プロキシ ホストの URL とポートを入力します。
- 12 [次へ] をクリックします。
- 13 独自のタイムアウト値を設定するには、[いいえ] をクリックします。
- 14 (オプション) vRealize Orchestrator が vRealize Automation からの接続または応答を待機する時間を設定するには、[接続タイムアウト (秒)] および [操作タイムアウト (秒)] テキスト ボックスにタイムアウト間隔を入力します。
- 15 [次へ] をクリックします。
- 16 ホストの認証タイプを選択します。

オプション	説明
SSO	vCenter Single Sign-On を使用する場合に選択します。
NTLM	Active Directory インフラストラクチャが NTLM 認証に依存している場合にのみ、このオプションを選択して NT LAN Manager (NTLM) プロトコルベースの認証を有効にします。 このオプションには、NTLM 認証情報と認証オプションが必要です。

- 17 NTLM を選択した場合は、[次へ] をクリックし、ワークステーション マシン名前と NetBIOS ドメイン名を入力します。
- 18 [送信] をクリックします。

vRealize Automation プラグインのワークフローの使用

vRealize Automation プラグインのワークフロー ライブラリには、カタログの操作、インフラストラクチャの管理、テナントやサービスの作成など、一般的なタスクに使用できるワークフローが含まれています。

カスタムの HTTP ヘッダー (vRealize Automation 固有の Tasks および Identity ヘッダーなど) を使用して、CRUD ワークフロー、プロビジョニング ワークフロー、およびプロビジョニング後のワークフローに適用できます。

操作制限の削除

作成、読み込み、更新、削除の操作は、バージョン 7.0 から制限されています。以前のバージョンのワークフローでこれらの操作を使用している場合、それらの操作はバージョン 7.0 以降では機能しません。ワークフローを更新してサポートされている操作に合わせるか、必要な操作を再度有効にすることができます。

操作を再度有効にするには、有効にしたい操作を `operations.properties` ファイルから削除してください。ファイルにある操作のリストについて詳しくは、[制限されている動作](#) を参照してください。

手順

- 1 vRealize Orchestrator のドロップダウン メニューから、[設計] を選択します。
- 2 [リソース] ビューをクリックします。
- 3 リソース階層で、[ライブラリ] - [VCAC] - [Util] を展開します。

- 4 バックアップを作成し、operations.properties ファイルを変更します。
 - a operations.properties をクリックして [ファイルに保存] を選択します。
 - b コピーをバックアップとして保存します。
 - c 新しいコピーを作成し、再度有効にする操作を削除します。
 - d 新しいファイルを保存します。
- 5 vRealize Orchestrator の既存のファイルを置き換えます。
 - a vRealize Orchestrator で、[Util] フォルダを右クリックして、[リソースのインポート] をクリックします。
 - b 新しいバージョンの operations.properties ファイルを参照して、[開く] をクリックします。
 - c [一斉置換] をクリックして、変更したバージョンを保存します。
- 6 vRealize Orchestrator サーバを再起動します。
- 7 operations.properties ファイルを選択して、[ビューア] タブをクリックします。
- 8 有効にした操作がファイルに存在しないことを確認してください。

結果

ファイルから削除した操作が、古いワークフローで機能するようになりました。

次のステップ

新しいワークフローを作成する場合は、制限されている操作を使用しないようにしてください。

制限されている動作

operations.properties ファイルのコンテンツには、制限されている動作が含まれています。操作を再度有効にするには、有効にしたい操作を operations.properties ファイルから削除します。

以下のテキストは、operations.properties ファイルのデフォルト バージョンです。操作を再度有効にするには、[操作制限の削除](#) を参照してください。

```
#Blueprints
operation.create=ManagementModelEntities.svc@VirtualMachineTemplates
operation.update=ManagementModelEntities.svc@VirtualMachineTemplates
operation.delete=ManagementModelEntities.svc@VirtualMachineTemplates
#Blueprint properties
operation.create=ManagementModelEntities.svc@VirtualMachineProperties
operation.read=ManagementModelEntities.svc@VirtualMachineProperties
operation.update=ManagementModelEntities.svc@VirtualMachineProperties
operation.delete=ManagementModelEntities.svc@VirtualMachineProperties
#Global profiles
operation.create=ManagementModelEntities.svc@GlobalProfiles
operation.read=ManagementModelEntities.svc@GlobalProfiles
operation.update=ManagementModelEntities.svc@GlobalProfiles
operation.delete=ManagementModelEntities.svc@GlobalProfiles
#Global profile properties
operation.create=ManagementModelEntities.svc@GlobalProfileProperties
```

```

operation.read=ManagementModelEntities.svc@GlobalProfileProperties
operation.update=ManagementModelEntities.svc@GlobalProfileProperties
operation.delete=ManagementModelEntities.svc@GlobalProfileProperties
#PropertySetXml
operation.create=ManagementModelEntities.svc@PropertySetXml
operation.read=ManagementModelEntities.svc@PropertySetXml
operation.update=ManagementModelEntities.svc@PropertySetXml
operation.delete=ManagementModelEntities.svc@PropertySetXml
#Property definitions
operation.create=ManagementModelEntities.svc@PropertyDefinitions
operation.read=ManagementModelEntities.svc@PropertyDefinitions
operation.update=ManagementModelEntities.svc@PropertyDefinitions
operation.delete=ManagementModelEntities.svc@PropertyDefinitions
#Property attributes
operation.create=ManagementModelEntities.svc@PropertyAttributes
operation.read=ManagementModelEntities.svc@PropertyAttributes
operation.update=ManagementModelEntities.svc@PropertyAttributes
operation.delete=ManagementModelEntities.svc@PropertyAttributes
#Property Attribute Types
operation.create=ManagementModelEntities.svc@PropertyAttributeTypes
operation.read=ManagementModelEntities.svc@PropertyAttributeTypes
operation.update=ManagementModelEntities.svc@PropertyAttributeTypes
operation.delete=ManagementModelEntities.svc@PropertyAttributeTypes
#Control layouts
operation.create=ManagementModelEntities.svc@ControlLayouts
operation.read=ManagementModelEntities.svc@ControlLayouts
operation.update=ManagementModelEntities.svc@ControlLayouts
operation.delete=ManagementModelEntities.svc@ControlLayouts
#Amazon Virtual Machine Templates
operation.create=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.read=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.update=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.delete=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
#Openstack Virtual Machine Templates
operation.create=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.read=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.update=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.delete=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
#Endpoint credentials
operation.create=ManagementModelEntities.svc@ConnectionCredentials
operation.update=ManagementModelEntities.svc@ConnectionCredentials
operation.delete=ManagementModelEntities.svc@ConnectionCredentials
#Management endpoints
operation.create=ManagementModelEntities.svc@ManagementEndpoints
operation.update=ManagementModelEntities.svc@ManagementEndpoints
operation.delete=ManagementModelEntities.svc@ManagementEndpoints
#Management endpoint properties
operation.create=ManagementModelEntities.svc@ManagementEndpointProperties
operation.read=ManagementModelEntities.svc@ManagementEndpointProperties
operation.update=ManagementModelEntities.svc@ManagementEndpointProperties
operation.delete=ManagementModelEntities.svc@ManagementEndpointProperties

```

vRealize Automation プラグイン インベントリの使用

[インベントリ] ビューを使用して、vRealize Automation オブジェクトに対してワークフローを実行できます。

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、[ツール] - [ユーザー環境設定] - [インベントリ] の順に移動し、[インベントリでコンテキスト メニューを使用する] チェック ボックスを選択します。オプションが有効になると、Orchestrator インベントリでオブジェクトを右クリックしたときに、そのオブジェクトに使用可能なすべてのワークフローが表示されます。

vRealize Automation プラグインの管理ワークフローの使用

管理ワークフローを使用すると、vRealize Automation サービス、テナント、承認ポリシー、資格、ビジネス グループ、カタログ アイテム、アドバンスト サービス コンポーネントを管理できます。

一部のワークフローには、vRealize Automation ホストの vCACCAFE:VCACHost 入力パラメータが含まれています。vRealize Automation ホスト接続の構成方法によって、ユーザーがワークフローを実行する際に、ロールがどのように適用されるかが決まります。

- 接続を共有セッションとして構成している場合、ワークフローを実行するために必要なロールが、共有セッション用のユーザー アカウントに割り当てられている必要があります。
- 接続を各ユーザーのセッションとして構成している場合、vRealize Automation ユーザー インターフェイスでユーザーにロールが割り当てられるように、ワークフローを実行する各ユーザーに必要なロールが割り当てられている必要があります。

これらのワークフローは、vRealize Orchestrator クライアントの [ワークフロー] ビューで表示される [ライブラリ] - [vRealize Automation] - [Administration] サブディレクトリ内で見つけることができます。

[Approval Policies] サブディレクトリ内のワークフローを使用して、承認ポリシーを作成、管理できます。

表 4-1. 承認ポリシー

ワークフロー	説明
承認ポリシーの有効化	承認ポリシーを有効にします。承認ポリシーは有効にすると読み取り専用になります。
承認レベルの追加	常に必須とする承認レベルを承認ポリシーに追加します。承認者として特定のユーザーとグループを選択する必要があります。
承認ポリシーのコピー	承認ポリシーをコピーします。
承認ポリシーの作成	承認レベルや承認者を指定せずに承認ポリシーのドラフトを作成します。承認レベルを作成してポリシーの承認者を指定するには、[承認レベルの追加] ワークフローを実行します。
承認ポリシーの無効化	承認ポリシーを無効にします。承認ポリシーに関連付けられているすべての既存の資格を削除することもできます。
承認ポリシーの削除	ドラフト状態にある承認ポリシーを削除します。アクティブな承認ポリシーは読み取り専用です。

[Business Groups] サブディレクトリ内のワークフローを使用して、ビジネス グループおよびビジネス グループのカスタム プロパティを作成、管理できます。

表 4-2. ビジネス グループ

ワークフロー	説明
カスタム プロパティの追加	カスタム プロパティをビジネス グループに追加します。
ビジネス グループの作成	ビジネス グループを作成します。
ビジネス グループの削除	ビジネス グループを削除します。
カスタム プロパティの削除	カスタム プロパティをビジネス グループから削除します。
ビジネス グループの更新	デフォルトのマシン プリフィックス、Active directory コンテナ、ユーザー ロールなど、ビジネス グループの詳細を更新します。
カスタム プロパティの更新	ビジネス グループのカスタム プロパティを更新します。

Administration サブディレクトリには vRealize Automation 7.0 より前のバージョンで使用できる [Business Groups (Deprecated)] サブディレクトリが含まれています。メイン フォルダにある同じ名前のワークフローを使用してください。

[Catalog Items] サブディレクトリ内のワークフローを使用して、カタログ アイテムを管理できます。

表 4-3. カatalog アイテム

ワークフロー	説明
カタログ アイテムの有効化	カタログ アイテムを有効にします。ユーザーがカタログ アイテムを申請するには、まずカタログ アイテムを有効にしてサービスに割り当てる必要があります。
サービスへのカタログ アイテムの割り当て	カタログ アイテムをサービスに割り当てます。ユーザーがカタログ アイテムを申請するには、まずカタログ アイテムを有効にしてサービスに割り当てる必要があります。
カタログ アイテムの無効化	ユーザーが申請できなくなるように、カタログ アイテムを無効にしてサービス カatalog から削除します。

[Composite Blueprint] サブディレクトリ内のワークフローを使用して、デザイン キャンバスで作成された複合ブループリントを管理できます。

表 4-4. 複合ブループリント

ワークフロー	説明
複合ブループリントの削除	未公開のブループリントをブループリントのデザイン リストから削除します。
複合ブループリントのインポート	YAML ファイルから複合ブループリントをインポートします。
複合ブループリントの公開	ドラフト状態にある複合ブループリントを公開します。
複合ブループリントの公開解除	公開済み複合ブループリントを公開解除します。

[Content] サブディレクトリ内のワークフローは廃止されました。インポートおよびエクスポート アクションを実行するには、Cloud Client を使用します。Cloud Client のダウンロードとドキュメントについては、<https://developercenter.vmware.com/tool/cloudclient> にアクセスしてください。

表 4-5. コンテンツ

ワークフロー	説明
コンテンツのエクスポート（非推奨）	インポートおよびエクスポート アクションを実行するには、Cloud Client を使用します。Cloud Client のダウンロードとドキュメントについては、 https://developercenter.vmware.com/tool/cloudclient にアクセスしてください。
コンテンツのインポート（非推奨）	インポートおよびエクスポート アクションを実行するには、Cloud Client を使用します。Cloud Client のダウンロードとドキュメントについては、 https://developercenter.vmware.com/tool/cloudclient にアクセスしてください。
コンテンツの転送（非推奨）	インポートおよびエクスポート アクションを実行するには、Cloud Client を使用します。Cloud Client のダウンロードとドキュメントについては、 https://developercenter.vmware.com/tool/cloudclient にアクセスしてください。
コンテンツの検証（非推奨）	インポートおよびエクスポート アクションを実行するには、Cloud Client を使用します。Cloud Client のダウンロードとドキュメントについては、 https://developercenter.vmware.com/tool/cloudclient にアクセスしてください。

[Entitlements] サブディレクトリ内のワークフローを使用して、資格を作成、管理できます。

表 4-6. 資格

ワークフロー	説明
資格の有効化	資格を有効にします。
資格へのカタログ アイテムの割り当て	1 つ以上のカタログ アイテムを資格に割り当てます。このワークフローを使用して承認ポリシーを割り当てすることもできます。
資格への即時アクションの割り当て	1 つ以上の即時アクションを資格に割り当てます。即時アクションで申請は行われません。
資格へのリソース アクションの割り当て	1 つ以上のリソース アクションを資格に割り当てます。このワークフローを使用して承認ポリシーを割り当てすることもできます。
資格へのサービスの割り当て	1 つ以上のサービスを資格に割り当てます。このワークフローを使用して承認ポリシーを割り当てすることもできます。
資格へのユーザーおよびグループの割り当て	1 人以上のユーザーまたは 1 つ以上のグループを資格に割り当てます。
資格の作成（非推奨）	資格を作成します。[サブテナントの資格の作成] を使用します。
サブテナントの資格の作成	資格を作成します。
資格の無効化	資格を無効にします。
ユーザーおよびグループの資格からの割り当て解除	資格のユーザー リストからユーザーおよびグループを削除します。

[Properties] サブディレクトリ内のワークフローを使用して、プロパティ定義とプロパティ グループを管理できます。vRealize Automation のプロパティとの競合を避けるため、すべてのカスタム プロパティ名で、会社名または機能名の後ろにドットを付けたプリフィックスを使用します。

表 4-7. プロパティ定義

ワークフロー	説明
プロパティ定義の作成	カスタム プロパティを作成します。
プロパティ定義の削除	カスタム プロパティを削除します。

プロパティ グループはプロパティ定義の集合です。

表 4-8. プロパティ グループ

ワークフロー	説明
グループへのプロパティの追加	定義済みカスタム プロパティをグループに追加します。
プロパティ グループの作成	定義済みカスタム プロパティの追加先となるプロパティ グループを作成します。
プロパティ グループの削除	プロパティ グループを削除します。
グループからのプロパティの削除	定義済みカスタム プロパティをプロパティ グループから削除します。
プロパティ グループの更新	プロパティ グループの名前または説明を変更します。
グループのプロパティの更新	プロパティ グループのプロパティの名前、値、動作を変更します。

[Services] サブディレクトリ内のワークフローを使用して、サービスを管理できます。

表 4-9. サービス

ワークフロー	説明
サービスの有効化	サービスを有効にします。
サービスへのカタログ アイテムの割り当て	1 つ以上のカタログ アイテムをサービスに割り当てます。
サービスのコピー	サービスをコピーします。
サービスの作成	サービスを作成します。
サービスの無効化	サービスを無効にします。
サービスの削除	サービスを削除します。

[Tenants] サブディレクトリ内のワークフローを使用して、テナントを作成、管理できます。

ID ストア ワークフローは廃止されました。代替ワークフローは vRealize Automation の Directories Management API の変更に対応します。

表 4-10. テナント

ワークフロー	説明
管理者の追加	1 人以上のテナント管理者およびインフラストラクチャ管理者をテナントに追加します。
テナントへの ID ストアの追加	ID ストアを vRealize Automation ホストのテナントに追加します。 このワークフローを実行できるのは、テナントを構成するシステム管理者に限られます。
テナントへの ID ストアの追加（非推奨）	「テナントへの ID ストアの追加」ワークフローを使用します。

表 4-10. テナント（続き）

ワークフロー	説明
vCAC ホストへの ID ストアの追加	ID ストアを vRealize Automation ホストとして構成されているテナントに追加します。このワークフローを実行できるのは、テナントに ID ストアを構成するテナント管理者に限られます。
vCAC ホストへの ID ストアの追加（非推奨）	「vCAC ホストへの ID ストアの追加」ワークフローを使用します。
テナントの作成	テナントを作成します。システム管理者の認証情報が追加されている vRealize Automation ホストを選択する必要があります。
テナントからの ID ストアの削除	ID ストアを vRealize Automation ホストのテナントから削除します。このワークフローを実行できるのは、テナントを構成するシステム管理者に限られます。
vCAC ホストからの ID ストアの削除	vRealize Automation ホストとして構成されているテナントから ID ストアを削除します。このワークフローを実行できるのは、テナントの ID ストアを構成するテナント管理者に限られます。
テナントの削除	テナントを削除します。
管理者の削除	1 人以上のテナント管理者およびインフラストラクチャ管理者をテナントから削除します。
テナントの ID ストアの更新	vRealize Automation ホストのテナントが使用している既存の ID ストアを更新します。このワークフローを実行できるのは、テナントを構成するシステム管理者に限られます。
テナントの ID ストアの更新（非推奨）	「テナントの ID ストアの更新」ワークフローを使用します。
vCAC ホストの ID ストアの更新	vRealize Automation ホストとして構成されているテナントの ID ストアを更新します。このワークフローを実行できるのは、テナントの ID ストアを構成するテナント管理者に限られます。
vCAC ホストの ID ストアの更新（非推奨）	「vCAC ホストの ID ストアの更新」ワークフローを使用します。
テナントの更新	既存のテナントの名前、説明、連絡先のメール アドレスを更新します。

[Workflow Subscriptions] サブディレクトリ内のワークフローを使用して、イベント ワークフロー サブスクリプションを管理できます。

表 4-11. ワークフロー サブスクリプション

ワークフロー	説明
ワークフロー サブスクリプションの削除	未公開のワークフロー サブスクリプションを削除します。このワークフローはシステム ワークフロー サブスクリプションとテナント ワークフロー サブスクリプションに適用されます。
システム ワークフロー サブスクリプションのエクスポート	システム ワークフロー サブスクリプションをエクスポートして、vRealize Orchestrator リソース要素として JSON 形式で保存します。 システム ワークフロー サブスクリプションは、システム イベントと全テナントのイベントに対処する特別なワークフロー サブスクリプションです。
テナント ワークフロー サブスクリプションのエクスポート	テナント ワークフロー サブスクリプションをエクスポートして、リソース要素として JSON 形式で保存します。 テナント固有のワークフローを実行する特別なワークフロー サブスクリプションです。

表 4-11. ワークフロー サブスクリプション (続き)

ワークフロー	説明
システム ワークフロー サブスクリプションのインポート	JSON ファイルからシステム ワークフロー サブスクリプションをインポートします。システム ワークフロー サブスクリプションはシステム イベントと全テナントのイベントに応じてトリガされます。
テナント ワークフロー サブスクリプションのインポート	エクスポート済みワークフロー サブスクリプションを JSON ファイルからインポートします。これらはテナント固有のワークフロー サブスクリプションです。
ワークフロー サブスクリプションの公開	ドラフトまたは未公開の状態にあるワークフロー サブスクリプションを公開します。このワークフローはシステム ワークフロー サブスクリプションとテナント ワークフロー サブスクリプションに適用されます。
システム ワークフロー サブスクリプションの登録	タイムアウト値、優先度の値を含むシステム ワークフロー サブスクリプションを作成します。
テナント ワークフロー サブスクリプションの登録	タイムアウト値、優先度の値を含め、テナント固有のワークフロー サブスクリプションを作成します。
ワークフロー サブスクリプションの公開解除	公開済みワークフロー サブスクリプションを公開解除します。このワークフローはシステム ワークフロー サブスクリプションとテナント ワークフロー サブスクリプションに適用されます。
ワークフロー サブスクリプションの更新	名前、説明、vRealize Orchestrator ワークフロー、サブスクリプション条件、タイムアウト値、ステータス値、優先度の値を変更します。イベント トピックやブロック状態を更新することはできません。

[XaaS Custom Resources] サブディレクトリ内のワークフローを使用して、XaaS カスタム リソースを作成、削除できます。

表 4-12. XaaS カスタム リソース

ワークフロー	説明
カスタム リソースの作成	カスタム リソースを作成します。
カスタム リソースの削除	カスタム リソースを削除します。

[XaaS Resource Actions] サブディレクトリ内のワークフローを使用して、XaaS リソース アクションを作成、管理できます。

表 4-13. XaaS リソース アクション

ワークフロー	説明
リソース アクションのクローン作成	既存のリソース アクションのコピーを作成します。
リソース アクションの作成	リソース アクションを作成します。
リソース アクションの削除	リソース アクションを削除します。
リソース アクションの公開	リソース アクションを公開します。
リソース アクションの公開解除	リソース アクションを公開解除します。

[XaaS Resource Mappings] サブディレクトリ内のワークフローを使用して、XaaS 以外のリソースへの XaaS マッピングを作成、管理できます。

表 4-14. XaaS リソース マッピング

ワークフロー	説明
リソース マッピングの作成	カタログ リソース タイプを vRealize Orchestrator タイプにマップします。
リソース マッピングの削除	リソース マッピングを削除します。
ターゲット基準の設定	リソース マッピングの可用性を決定する条件を指定します。

[XaaS Server Configuration] サブディレクトリ内のワークフローを使用して、ターゲットの Orchestrator インスタンスを管理できます。

表 4-15. XaaS サーバの構成

ワークフロー	説明
Orchestrator サーバ構成の更新	ポート、ホスト、ユーザー名、パスワードを含め、サーバ設定を変更します。
Orchestrator サーバ構成の検証	vRealize Orchestrator 設定が有効であることを確認します。ワークフローは構成が有効である場合は TRUE を、構成が有効でない場合は FALSE を返します。

[XaaS Service Blueprints] サブディレクトリ内のワークフローを使用して、XaaS ブループリントを作成、管理できます。

表 4-16. XaaS ブループリント

ワークフロー	説明
サービス ブループリントのクローン作成	サービス ブループリントのコピーを作成します。
サービス ブループリントの作成	サービス ブループリントを作成します。
サービス ブループリントの削除	サービス ブループリントを削除します。
サービス ブループリントの公開	サービス ブループリントを公開します。
サービス ブループリントの公開解除	サービス ブループリントを公開解除します。

vRealize Automation プラグインのインフラストラクチャ管理ワークフローの使用

インフラストラクチャ管理ワークフローを使用して基本操作を実行できます。拡張性パッケージを使用して vRealize Automation をカスタマイズすると、vRealize Orchestrator ワークフローをプロビジョニング プロセスの一環として呼び出したり、カスタム操作メニューを使用して呼び出したりできます。

インフラストラクチャ管理ワークフローは、Orchestrator クライアントの [ワークフロー] ビューで表示されるプラグイン ライブラリの [Infrastructure Administration] サブディレクトリ内で見つけることができます。

インフラストラクチャ管理ワークフローを使用すると、仮想マシンをプロビジョニングして、作成、読み取り、更新、削除といった基本操作を実行できます。

表 4-17. インフラストラクチャ管理

ワークフロー名	説明
仮想マシンの状態変更の待機	<p>仮想マシン セットの状態が変更されるのを待機します。すべての仮想マシンが正常な状態にある場合、トリガが呼び出されてワークフローが正常に終了します。指定したいいずれかの仮想マシンがエラー状態にあるか、存在しない場合、ワークフローは失敗します。次のいずれかを入力して、正常な状態にあるかどうかを指定する必要があります。</p> <ul style="list-style-type: none"> ■ Requested ■ AwaitingApproval ■ RegisterMachine ■ BuildingMachine ■ AddingDisks ■ MachineProvisioned ■ MachineActivated ■ InstallTools (VMware のみ) ■ On ■ Off ■ TurningOn ■ TurningOff ■ ShuttingDown ■ Suspending ■ Resetting ■ Rebooting ■ Expired ■ DeactivateMachine ■ UnprovisionMachine ■ Disposing ■ Finalized
IaaS モデル エンティティの作成	指定した vRealize Automation モデルのエンティティを作成して保持します。
IaaS モデル エンティティの削除	指定した vRealize Automation モデル エンティティを削除します。
プロビジョニング後のアクションの呼び出し（非推奨）	「リソース アクションの申請」ワークフローを使用します。
ブループリントからの仮想マシンのプロビジョニング（vRealize Automation 7.0 で廃止済み）	「カタログ アイテムの申請」または「プロビジョニング申請を使用したカタログ アイテムの申請」に置き換えられました。
カスタム フィルタを使用した IaaS エンティティの読み取り	カスタム フィルタを使用して vRealize Automation エンティティのリストを読み取ります。フィルタを指定しない場合は、すべてのエンティティが返ります。
システム クエリを使用した IaaS エンティティの読み取り	OData システム フィルタを使用して vRealize Automation エンティティのリストを読み取ります。システム フィルタは OData URI 規則に対して適用されます。
IaaS モデル エンティティの読み取り	ID を使用して vRealize Automation モデル エンティティを読み取ります。
IaaS モデル エンティティの更新	ID を使用して vRealize Automation モデル エンティティを更新します。

[Extensibility] サブディレクトリ内のワークフローを使用して vRealize Automation をカスタマイズし、vRealize Orchestrator ワークフローをプロビジョニング プロセスの一環として呼び出すか、カスタム操作メニューを使用して呼び出します。

サブディレクトリには、IaaS の認証情報、エンドポイント、エンタープライズ グループ、マシン プリフィックスなどのエンティティを管理するためのワークフローも含まれています。

表 4-18. 拡張性

ワークフロー名	説明
vCO カスタマイズのインストール	カスタマイズされた状態変更ワークフローやメニュー操作ワークフローを含め、Orchestrator カスタマイズをインストールします。
vCO カスタマイズのアンインストール	カスタマイズされた状態変更ワークフローやメニュー操作ワークフローを含め、Orchestrator カスタマイズをアンインストールします。
IaaS 仮想マシンの予約の変更	予約、ビジネス グループなど、管理対象仮想マシンの属性を変更します。
IaaS 仮想マシンのインポート（非推奨）	Cloud Client を使用します。Cloud Client のダウンロードとドキュメントについては、 https://developercenter.vmware.com/tool/cloudclient にアクセスしてください。
vCenter 仮想マシンのインポート（非推奨）	Cloud Client を使用します。Cloud Client のダウンロードとドキュメントについては、 https://developercenter.vmware.com/tool/cloudclient にアクセスしてください。
仮想マシンの登録解除（vRealize Automation 7.0 で廃止済み）	代替ワークフローは提供されていません。
ブループリントおよびその仮想マシンへのメニュー操作の割り当て（非推奨）	仮想マシンのメニュー操作を追加または更新します。 非推奨ではない代替ワークフローには、「資格内のリソースアクションの割り当て」や、「複合ブループリントのインポート」が含まれます。
仮想マシンへのメニュー操作の割り当て（非推奨）	ID を使用して vRealize Automation モデル エンティティを更新します。 非推奨ではない代替ワークフローには、「資格内のリソースアクションの割り当て」や、「複合ブループリントのインポート」が含まれます。
ブループリントおよびその仮想マシンへの状態変更ワークフローの割り当て（非推奨）	vRealize Automation のイベント ブローカ サブスクリプションによって置き換えられました。
メニュー操作のカスタマイズ(vRealize Automation 7.0 で廃止済み)	代替ワークフローは提供されていません。
ブループリントおよびその仮想マシンからのメニュー操作の削除(vRealize Automation 7.0 で廃止済み)	代替ワークフローは提供されていません。
ブループリントおよびその仮想マシンからの状態変更ワークフローの削除	ブループリントおよびその仮想マシンから状態変更ワークフローを削除します。

vRealize Automation IaaS モデル エンティティの作成

ワークフローを実行して、単純または複雑な vRealize Automation IaaS エンティティ（仮想マシンからユーザーへの参照など）を作成することができます。

手順

- 1 Orchestrator クライアントのドロップダウン メニューから、[実行] または [設計] を選択します。
- 2 [ワークフロー] ビューをクリックします。

- 3 [ライブラリ] - [vRealize Automation] - [インフラストラクチャ管理] の順に展開します。
- 4 [IaaS モデル エンティティの作成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 vRealize Automation ホストの順に選択します。
- 6 [モデル名] テキスト ボックスにモデルの名前を入力します。
- 7 [エンティティ セット名] テキスト ボックスにエンティティ セットの名前を入力します。
スクリプトまたは REST API を使用して、[単純なプロパティ]、[複雑なプロパティへのリンク]、および [HTTP ヘッダー] プロパティを設定します。
- 8 [送信] をクリックして、ワークフローを実行します。

vRealize Automation IaaS モデル エンティティの読み込み

ワークフローを実行して、vRealize Automation IaaS モデル エンティティを読み込むことができます。

手順

- 1 Orchestrator クライアントのドロップダウン メニューから、[実行] または [設計] を選択します。
- 2 [ワークフロー] ビューをクリックします。
- 3 [ライブラリ] - [vRealize Automation] - [インフラストラクチャ管理] の順に展開します。
- 4 [IaaS モデル エンティティの読み込み] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 vRealize Automation ホストの順に選択します。
- 6 [モデル名] テキスト ボックスにモデルの名前を入力します。
- 7 [エンティティ セット名] テキスト ボックスにエンティティ セットの名前を入力します。
スクリプトまたは REST API を使用して、[HTTP ヘッダー] プロパティを設定します。
- 8 [送信] をクリックして、ワークフローを実行します。

vRealize Automation プラグインの申請ワークフローの使用

申請ワークフローを使用して、カタログ アイテムやリソース アクションを申請したり、作業アイテムを完了またはキャンセルしたりできます。

作業アイテムはユーザー入力やアクションを必要とします。たとえば、ワークフロー操作、承認アクション、回収要請への応答が必要です。

vRealize Orchestrator クライアントの [ワークフロー] ビューで表示されるプラグイン ライブラリの [Requests] サブディレクトリから、これらのワークフローにアクセスすることができます。

ワークフロー	説明
作業アイテムのキャンセル	アクティブな作業アイテムをキャンセルします。このワークフローを使用できるのは、システム管理者に限られます。
作業アイテムの完了	ユーザー入力に基づいて作業アイテムを終了します。

ワークフロー	説明
カタログ アイテムの申請	ワークフローを実行しているユーザーのカタログ アイテムを申請します。 複合ブループリントを申請するワークフローが必要な場合は、「プロビジョニング申請を使用したカタログ アイテムの申請」ワークフローを使用します。
ユーザーの代わりにカタログ アイテムを申請	ユーザーの代わりにカタログ アイテムの申請を送信します。このワークフローを使用できるのは、ユーザーおよびそのユーザーの代わりに申請を送信するユーザーの両方が、カタログ アイテムの使用資格を持っている場合に限られます。
プロビジョニング申請を使用したカタログ アイテムの申請	ワークフローを実行しているユーザーのカタログ アイテムとして複合ブループリントを申請します。申請への入力のカスタマイズしている場合は、ワークフローをカスタマイズする必要があります。複合ブループリントではこのワークフローを使用します。
リソース アクションの申請	ワークフローを実行しているユーザーによって所有されているカタログ アイテムに対するリソース アクションを申請します。
ユーザーの代わりにリソース アクションを申請	ユーザーの代わりにリソース アクションの申請を送信します。このワークフローを使用できるのは、ユーザーおよびそのユーザーの代わりに申請を送信するユーザーの両方が、リソース アクションの使用資格を持っている場合に限られます。
申請テンプレートを使用したリソース アクションの申請	<p>複雑なパラメータが含まれているリソース アクションを申請します。ベスト プラクティスとして、ワークフローを複製してからアクションに応じてカスタマイズすることが推奨されます。このワークフローを使用して複雑なパラメータを渡したり、申請フォームに表示する必要のないパラメータを非表示にしたりできます。このワークフローの主要な利点の 1 つとして、IaaS の仮想マシンの再構成アクションをカスタマイズできることが挙げられます。</p> <p>仮想マシンで再構成操作を作成するには、ワークフローのコピーを作成してからスクリプトを変更する必要があります。vRealize Orchestrator に表示されるパラメータを構成し、Cafe.Shim.VirtualMachine.Reconfigure.Requestor パラメータを設定します。このパラメータはログ作成に使用され、空白にすることはできません。次の例を参照してください。</p> <pre> var requestTemplate = vCACCAFERequestsHelper.getRequestForResourceAction(operation) var jsonData = vCACCAFERequestsHelper.getResourceActionRequestData(requestTemplate); var json = JSON.parse(jsonData); //Change cpu example json.cpu = 2; //This is a property needed for the Reconfigure IaaS operation: json["Cafe.Shim.VirtualMachine.Reconfigure.Requestor"] = 1; vCACCAFERequestsHelper.setResourceActionRequestData(requestTemplate, JSON.stringify(json)); request = System.getModule("com.vmware.library.vcaccafe.request").requestResourceActionWithRequestTemplate(operation, requestTemplate); </pre>
カタログ アイテム申請の待機	カタログ アイテム申請の終了を待機します。
リソース アクション申請の待機	リソース アクション申請の終了を待機します。
作業アイテムの待機	作業アイテムの終了を待機します。

vRealize Automation プラグインのサンプル ワークフローの使用

サンプル ワークフローは例として使用したり、独自のカスタム ワークフローを作成する際の開始点として使用したりできます。

これらのワークフローは、vRealize Orchestrator クライアントの [ワークフロー] ビューで表示されるプラグインライブラリの [Sample] サブディレクトリで見つけることができます。

ワークフロー名	説明
権限の作成	認証クライアントと通信するためのサンプル スクリプトと、vRealize Automation で権限を作成するための権限サービスを提供します。
テナントの作成	デフォルト テナントと同じ vRealize Automation ホストと Active Directory 構成を使用してテナントを作成します。このワークフローを実行するには、システム管理者の認証情報を使用して追加した vRealize Automation ホストを選択します。Active Directory 設定はワークフローを実行する前に変更できます。
カタログ アイテムの一覧表示	選択したテナントのカタログ アイテムのリストを返します。
カタログ アイテムのプロビジョニング申請の JSON 出力	カタログ アイテムのデフォルトの申請フォームを取得して、JSON 形式でコンソール ログに追加します。このデータを使用してプロビジョニング申請をカスタマイズできます。また、この情報を使用して [プロビジョニング申請を使用したカタログ アイテムの申請] ワークフローを変更できます。

vRealize Automation 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、vRealize Automation 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利用できる 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。

最新版の vRealize Automation API ドキュメントについては、<https://www.vmware.com/support/pubs/vcac-pubs.html> を参照してください。

手順

- 1 Orchestrator クライアントに管理者としてログインします。
- 2 [ツール] - [API Explorer] を選択します。
- 3 左側のペインで [vCAC] モジュールと [VCACCAFE] モジュールをダブルクリックして、vRealize Automation プラグインの API オブジェクトの階層リストを展開します。

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。

開発のベスト プラクティスの詳細については、『[vRealize Orchestrator ドキュメント](#)』を参照してください。

サンプル vRealize Automation プラグイン スクリプト

JavaScript サンプルを切り取り、貼り付け、および編集することで、vRealize Automation タスク自動化用のオリジナル カスタム スクリプトを開発することができます。

CRUD インフラストラクチャ管理サンプル スクリプト

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

vRealize Orchestrator でのスクリプト記述の詳細については、『VMware vRealize Orchestrator を使用した開発』を参照してください。

例：vRealize Automation モデル エンティティの作成

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。

- 1 モデル名およびエンティティの設定名を定義します。
- 2 ホスト プリフィックスのプロパティを定義します。
- 3 ホスト プリフィックスのエンティティを保存します。
- 4 プロビジョニング グループのプロパティを定義します。
- 5 プロビジョニング グループをリンクとして定義します。
- 6 プロビジョニング グループのエンティティをホスト名のプリフィックスを付けて保存します。

表 4-19. 入力変数

変数	タイプ
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'HostNamePrefixes';
var links = null;
var headers = null;
//Create properties for prefix entity
var prefixInputProperties = {
    MachinePrefix:'test-prefix',
    NextMachineNo:1,
    MachineNumberLength:3
};
//Save the prefix
var prefixEntity = vCACEntityManager
    .createModelEntity(host.id, modelName, entitySetName, prefixInputProperties, links, headers);
entitySetName = 'ProvisioningGroups';
//Create properties for the provisioning group entity
inputProperties = {
    GroupName:'TestGroupName',
    GroupDescription:'This group was generated with a vCO workflow',
    AdministratorEmail:'test@test.com',
    AdContainer:'AD',
    IsTestGroup:false,
    Flags:2,
    GroupType:1};
//Add a reference to the newly created prefix entity
links = {
    HostNamePrefix:prefixEntity
};
//Save the provisioning group
var entity = vCACEntityManager.createModelEntity(host.id, modelName, entitySetName, inputProperties,
links, headers);
```

例：vRealize Automation モデル エンティティの更新

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。

- 1 指定のエンティティからホスト ID を取得します。
- 2 指定のエンティティからモデル名を取得します。
- 3 指定のエンティティからエンティティの設定名を取得します。
- 4 指定のエンティティからエンティティ ID を取得します。
- 5 更新するプロパティを定義します。
- 6 エンティティを更新するアクションを開始します。

表 4-20. 入力変数

変数	Type
entity	vCAC:Entity
updatedDescription	文字列

```
var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
var entityIdString = entity.keyString;
var links = null;
var headers = null;
var updateProperties = new Properties();
updateProperties.put("UserNameDescription", updatedDescription);
//Update the user description
System.getModule("com.vmware.library.vcac")
    .updateVCACEntity(hostId, modelName, entitySetName, entityIdString, updateProperties, links,
        headers);
```

例：vRealize Automation モデル エンティティの読み込み

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。

- 1 モデル名およびエンティティの設定名を定義します。
- 2 プロパティ オブジェクトにブループリント ID を定義します。
- 3 エンティティを読み込みます。

表 4-21. 入力変数

変数	Type
host	vCAC:VcacHost
blueprintID	文字列

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var links = null;
var headers = null;
```



```
//Create properties for the prefix entity
var blueprintId = {
    VirtualMachineTemplateID:blueprintId,
};
//Read the blueprint
var entity = vCACEntityManager
    .readModelEntity(host.id, modelName, entitySetName, blueprintId, headers);
```

例：vRealize Automation モデル エンティティの削除

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。

- 1 指定のエンティティからホスト ID を取得します。
- 2 指定のエンティティからモデル名を取得します。
- 3 指定のエンティティからエンティティの設定名を取得します。
- 4 指定のエンティティからエンティティ ID を取得します。
- 5 エンティティを削除するアクションを開始します。

表 4-22. 入力変数

変数	タイプ
entity	vCAC:Entity

```
var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
var entityKeyString = entity.keyString;
var headers = null;
//Delete the entity
System.getModule("com.vmware.library.vcac")
    .deleteVCACEntity(hostId, modelName, entitySetName, entityKeyString, headers);
```

例：カスタム フィルタを使用した vRealize Automation エンティティの読み取り

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。

- 1 モデル名およびエンティティの設定名を定義します。
- 2 エンティティのフィルタとして使用するプロパティを定義します。
- 3 エンティティのリストを読み込みます。

表 4-23. 入力変数

変数	Type
host	vCAC:VcacHost
templateName	文字列

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var headers = null;
```

```
//Create properties for prefix entity
var properties = {
    VirtualMachineTemplateName:templateName,
};
//Read a list of entities
var entities = vCACEntityManager
    .readModelEntitiesByCustomFilter(host.id, modelName, entitySetName, properties, headers);
```

例：システム クエリを使用した vRealize Automation エンティティの読み取り

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。

- 1 モデル名およびエンティティの設定名を定義します。
- 2 エンティティのフィルタとして使用するシステム クエリを指定し、マシン状態とコンポーネント フラグでフィルタされたすべての仮想マシンの上位 10 位の結果を選択します。
- 3 エンティティのリストを読み込みます。

表 4-24. 入力変数

変数	タイプ
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachines';
var filter = "VirtualMachineState eq 'Off' and IsComponent eq true";
var orderBy = 'VirtualMachineName asc';
var top = 10; {
var skip = 0;,
var headers = null;
var select = null;
var entities = vCACEntityManager
    readModelEntitiesBySystemQuery(host.id, modelName, entitySetName, filter, orderBy, select, top,
    skip, headers);
```

vRealize Automation エンティティのサンプル スクリプトの検索

JavaScript サンプルを切り取り、貼り付け、および編集することで、vCACCAFEEntitiesFinder スクリプト作成ユーティリティ オブジェクトで vRealize Automation エンティティを検索するためのスクリプトを記述できます。

vRealize Orchestrator でのスクリプト記述の詳細については、『VMware vRealize Orchestrator を使用した開発』を参照してください。

例：名前フィルタによるカタログリソースの検索

表 4-25. 入力変数

変数	タイプ
host	vCACCAFE:VcacHost

次のサンプルのいずれかを使用できます。

- このサンプル スクリプトは、*name_of_the_resource* のクエリに名前および説明で一致するターゲット ホストのすべてのカタログ リソースを取得します。

```
var items = vCACCAFEEntitiesFinder.findCatalogResources(host, "name_of_the_resource");
```

- 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。
 - a コンシューマ リソース サービスを取得し、vCACCAFEPageOdataRequest オブジェクトのインスタンスを Pageable パラメータとして渡して get メソッドを開始します。
 - b vCACCAFEPageOdataRequest オブジェクトを、*name_of_the_resource* 文字列に一致する name 属性の 1 つのフィルタとして OData クエリを提供して作成します。

```
var service = host.createCatalogClient().getCatalogConsumerResourceService();

var filter = new Array();
filter[0] = vCACCAFEFilterParam.equal("name", vCACCAFEFilterParam.string("name_of_the_resource"));
var query = vCACCAFEQuery.query().addFilter(filter);

var items = service.getResourcesList(new vCACCAFEPageOdataRequest(query));
```

例：所有者フィルタによるカタログリソースの検索

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。

- 1 コンシューマ リソース サービスを取得し、vCACCAFEPageOdataRequest オブジェクトのインスタンスを Pageable パラメータとして渡して get メソッドを開始します。
- 2 vCACCAFEPageOdataRequest オブジェクトを、*user@domain.com* 文字列に一致する owner/ref 属性の 1 つのフィルタとして OData クエリを提供して作成します。

owners/ref 属性はカタログ リソースの内部構造およびフィールドに基づいた構成です。

vCACCAFECatalogResource エンティティは owners 属性を持ち、これは vCACCAFECatalogPrincipal エンティティの集合です。vCACCAFECatalogPrincipal エンティティはユーザーのプリンシパル ID を表す文字列である ref プロパティを持ちます。

```
var filter = new Array();
filter[0] = vCACCAFEFilterParam.substringOf("owners/ref",
vCACCAFEFilterParam.string("user@domain.com"));
var query = vCACCAFEQuery.query().addFilter(filter);

var items = service.getResourcesList(new vCACCAFEPageOdataRequest(query));
```

例：名前および所有者フィルタによるカタログリソースの検索

このサンプル スクリプトでは、前述の 2 つのサンプルの OData クエリを vCACCAFEFilterParam.and(array of conditions) 論理演算子を使用して一つの条件に結合します。

```
var conditions = new Array();
conditions[0] = vCACCAFEFilterParam.equal("name",
vCACCAFEFilterParam.string("name_of_the_resource_here"));
```

```

conditions[1] = vCACCAFEFilterParam.substringOf("owners/ref",
vCACCAFEFilterParam.string("user@domain.com"));

var filter = new Array();
filter[0] = vCACCAFEFilterParam.and(conditions);
var query = vCACCAFE0dataQuery.query().addFilter(filter);

var items = service.getResourcesList(new vCACCAFEPage0dataRequest(query));

```

その他の条件は、vCACCAFEFilterParam.group(array of parameters)、vCACCAFEFilterParam.not(parameter)、vCACCAFEFilterParam.startsWith(id, string)、vCACCAFEFilterParam.endsWith(id, string)、vCACCAFEFilterParam.greaterThan(id, number)、vCACCAFEFilterParam.lessThan(id, number) など異なる論理演算子を使用して定義できます。

vRealize Automation がプロビジョニングしたリソースを取得するためのサンプル スクリプト

JavaScript のサンプルを切り取り、貼り付け、および編集することで、vRealize Automation がプロビジョニングしたリソースの実際のエンティティを取得するスクリプトを作成できます。

vRealize Automation では CatalogResource タイプが、プロビジョニングされたリソースを表します。このタイプには ProviderBinding タイプの属性があり、以下の属性でカタログ リソースとそのプロバイダ間の関係を表します。

- bindingId - プロバイダに固有のエンティティの ID を表します。
- providerRef - vRealize Automation コンポーネント レジストリに登録されたサービスに直接対応するカタログ プロバイダを識別します。

vRealize Orchestrator でのスクリプト記述の詳細については、『VMware vRealize Orchestrator を使用した開発』を参照してください。

例：vRealize Automation カatalog リソースとしてプロビジョニングされた仮想マシンの取得

このサンプルでは、vRealize Automation ホストとその IaaS ホストを入力パラメータとして使用し、プロビジョニングされたリソース ID の場合は、対応する IaaS 仮想マシンを返します。スクリプト コードは、iaas-service プロバイダがプロビジョニングした Virtual Machine タイプのカタログ リソースのみを取得します。

表 4-26. 入力変数

変数	タイプ
vcacHost	vCACCAFE:VCACHost
iaasHost	vCAC:VCACHost

```

// Id of the catalog resource (or vCACCAFECatalogResource_instance.getId())
var resourceId = "c222629c-6f90-4458-8c92-8ece0ba06173";

var resource = vCACCAFEEntitiesFinder.getCatalogResource(vcacHost, resourceId);

var resourceType = resource.getResourceTypeRef().getLabel();
System.log("resource type: " + resourceType);

```

```

var providerBinding = resource.getProviderBinding();

var bindingId = providerBinding.getBindingId();
System.log("provider binding id: " + bindingId);

var provider = providerBinding.getProviderRef();
System.log("provider id: " + provider.getId());
System.log("provider name: " + provider.getLabel());

if ((resourceType == "Virtual Machine") && (provider.getLabel() == "iaas-service")) {
    System.log("It is an IaaS VM!");

    // IaaS virtual machine
    var vm = Server.findForType("vCAC:VirtualMachine", bindingId);
    System.log("IaaS VM id: " + vm.virtualMachineID);
    System.log("IaaS VM name: " + vm.displayName);

    // IaaS Entity
    var entity = System.getModule("com.vmware.library.vcac").getVirtualMachineEntityFromId(iaasHost,
bindingId);
    System.log("IaaS entity id: " + entity.keyString);
}

```

一般的なタスクのサンプル スクリプト

JavaScript サンプルの切り取り、貼り付け、編集、またはそれらをサンプルとして使用することで、一般的な vRealize Automation タスクで使用する独自のスクリプト開発への理解を深めることができます。

vRealize Orchestrator でのスクリプト記述の詳細については、『VMware vRealize Orchestrator を使用した開発』を参照してください。

例：vRealize Automation アドバンスド サービス ブループリントの作成

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。

- 1 vRealize Orchestrator ワークフローを使用して、サービス ブループリントを構築します。
- 2 ワークフローに基づいてサービス ブループリントのコンテンツを生成します。
- 3 サービス ブループリントのエンティティを作成します。
- 4 サービス ブループリントを発行します。

表 4-27. 入力変数

変数	タイプ
host	vCACCAFE:VCACHost

```

//ID of the workflow used to create the service blueprint
var workflowId = "44e42047-2fa0-4e4a-ba0c-12086540b28b";

var name = "MyBlueprint"
var description = "Blueprint description";
var workflowClient = host.createAdvancedDesignerClient().getAdvancedDesignerWorkflowService();

```

```
//Generate a service blueprint based on the workflow ID
var blueprint = workflowClient.generateServiceBlueprintByWorkflowId(workflowId);
blueprint.setTenant(host.tenant);
blueprint.setName(name);
blueprint.setDescription(description);

//Create the service blueprint
var blueprintService =
host.createAdvancedDesignerClient().getAdvancedDesignerServiceBlueprintService();
var uri = blueprintService.createServiceBlueprint(host.tenant , blueprint);

//Publish the service blueprint
var createdBlueprint = blueprintService.getServiceBlueprintByUri(uri);
blueprintService.updateServiceBlueprintStatus(host.tenant, createdBlueprint.getId(),
vCACCAFEDesignerPublishStatus.PUBLISHED);
```

例：vRealize Automation 承認ポリシーの作成

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。

- 1 承認ポリシー タイプを取得します。
- 2 承認を受ける必要のあるユーザーとグループを指定します。
- 3 承認レベルを設定します。
- 4 プロビジョニング前の承認段階を定義します。
- 5 プロビジョニング後の承認段階を定義します。
- 6 名前、説明、タイプといった承認ポリシーの仕様を定義します。
- 7 承認ポリシーを作成します。
- 8 承認ポリシーを公開します。一度公開されると、承認ポリシーは読み取り専用になります。

表 4-28. 入力変数

変数	タイプ
host	vCACCAFE:VCACHost

```
// Get the type of approval policy by ID
var typeService = host.createApprovalClient().getApprovalApprovalPolicyTypeService();
var type = typeService.getApprovalPolicyType("com.vmware.cafe.catalog.request");

// Set the user and group required to complete the approval
var user = new vCACCAFEApprovalPrincipal();
user.setValue("user@domain.com");
user.setType(vCACCAFEApprovalPrincipalType.USER);

var group = new vCACCAFEApprovalPrincipal();
group.setValue("group@domain.com");
group.setType(vCACCAFEApprovalPrincipalType.GROUP);
```

```

// Set the level of the approval
var level = new vCACCAFEApprovalLevel();
level.setName("IT Approval Level");
level.setDescription("IT Approval Level description");
level.setApprovalMode(vCACCAFEApprovalMode.ALL);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers", user);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers", group);
level.setLevelNumber(1);

// Set pre-provisioning phase type and the phase of the approval
var phase1Type = new vCACCAFEApprovalPhaseType();
phase1Type.setId("com.vmware.cafe.catalog.request.pre");
phase1Type.setName("Pre-Provisioning type");
phase1Type.setDescription("Pre-Provisioning type description");
phase1Type.setPhaseOrder(1);

var phase1 = new vCACCAFEPhase();
phase1.setName("Pre-Provisioning");
phase1.setDescription("Pre provisioning phase");
phase1.setPhasetype(phase1Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase1, "getLevels", level);

// Set post-provisioning phase type and the phase of the approval
var phase2Type = new vCACCAFEApprovalPhaseType();
phase2Type.setId("com.vmware.cafe.catalog.request.post");
phase2Type.setName("Post-Provisioning type");
phase2Type.setDescription("Post-Provisioning type description");
phase2Type.setPhaseOrder(1);

var phase2 = new vCACCAFEPhase();
phase2.setName("Post-Provisioning");
phase2.setDescription("Post provisioning phase");
phase2.setPhasetype(phase2Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase2, "getLevels", level);

// Create the approval policy specifications
var spec = new vCACCAFEApprovalPolicy();
spec.setName("New Policy");
spec.setDescription("New Policy description");
spec.setPolicyType(type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase1);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase2);

// Create the approval policy
var approvalPolicyService = host.createApprovalClient().getApprovalApprovalPolicyService();
var approvalPolicy = approvalPolicyService.createPolicy(spec);

// Publish the approval policy
approvalPolicy.setState(vCACCAFEApprovalPolicyState.PUBLISHED);
approvalPolicy = approvalPolicyService.update(approvalPolicy);
System.log("New approval policy id: " + approvalPolicy.getId());

```

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 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、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、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。

この章には、次のトピックが含まれています。

- [Configuration プラグインのワークフロー ライブラリへのアクセス](#)
- [Configuration プラグインのワークフロー ライブラリ](#)

Configuration プラグインのワークフロー ライブラリへのアクセス

Configuration プラグイン ワークフロー ライブラリの要素にアクセスするには、Orchestrator クライアントを使用する必要があります。

手順

- 1 Orchestrator クライアントのドロップダウン メニューから、[実行] または [設計] を選択します。
- 2 [ワークフロー] ビューをクリックします。
- 3 階層リストで [ライブラリ] - [構成] の順に展開します。

次のステップ

ワークフロー ライブラリを確認します。

Configuration プラグインのワークフロー ライブラリ

Configuration プラグインのワークフロー ライブラリには、vRealize Orchestrator の構成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。

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。

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [構成] - [SSL トラスト マネージャ ワークフロー] 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
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。
ファイルから証明書をインポート	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

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [構成] - [キーストア] 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
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。
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。

Library プラグインの使用

6

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、Orchestrator のトラブルシューティングを実行できます。

この章には、次のトピックが含まれています。

■ Library プラグインのワークフロー

Library プラグインのワークフロー

Library プラグインは、[ロック]、[Orchestrator]、[トラブルシューティング] 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

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] - [ロック] 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
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。

Orchestrator タスク ワークフロー

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] - [Orchestrator] - [タスク] 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
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

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] - [Orchestrator] - [ワークフロー] 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
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 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

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] - [タグ付け] 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
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。
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。
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、指定した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。

SQL プラグインの使用

7

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 プラグインの標準ワークフローの使用](#)

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、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

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。

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [SQL] - [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
データベースの追加	Database プラグイン インベントリにデータベース オブジェクトを追加します。
データベースへのテーブルの追加	Database プラグイン インベントリ内のデータベースにデータベース テーブルを追加します。
データベースの削除	Database 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	Database 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	Database 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	Database 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、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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [SQL] - [構成] の順に展開し、[データベースの追加] ワークフローに移動します。
- 4 [データベースの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [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。
- 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>

- 8 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	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> を使用する必要はありません。

- 9 [送信] をクリックして、ワークフローを実行します。

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、Database 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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SQL] - [構成] の順に展開し、[データベースへのテーブルの追加] ワークフローに移動します。
- 3 [データベースへのテーブルの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 テーブルを追加するデータベースを選択します。
- 5 追加するテーブルを選択します。
- 6 [送信] をクリックして、ワークフローを実行します。

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 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、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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 [ワークフロー] の階層リストで、[ライブラリ] - [SQL] - [構成] の順に展開し、[データベースの更新] ワークフローに移動します。
- 4 [データベースの更新] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 更新するデータベースを選択します。
- 6 [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。
指定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。
- 7 データベースのタイプを選択します。
- 8 [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。

9 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	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> を使用する必要はありません。

10 [送信] をクリックして、ワークフローを実行します。

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。また、SQL 操作の実行と同様に、SQL プラグイン ワークフローを実行して、データベースおよびデータベース テーブルを管理することもできます。

JDBC URL の生成

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] の順に展開し、[JDBC URL ジェネレータ] ワークフローに移動します。
- 3 [JDBC URL ジェネレータ] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 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、[次へ] をクリック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指定する必要があります。

- 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 [送信] をクリックして、ワークフローを実行します。

JDBC 接続のテスト

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] の順に展開し、[JDBC 接続] サンプル ワークフローに移動します。
- 3 [JDBC 接続] サンプル ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 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 [送信] をクリックして、ワークフローを実行します。

JDBC を使用したテーブルの作成

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] を展開し、[JDBC テーブルの作成] サンプル ワークフローに移動します。
- 3 [JDBC テーブルの作成] サンプル ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 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。

5 SQL create 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")
```

6 [送信] をクリックして、ワークフローを実行します。

JDBC テーブルへの行の挿入

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

- 1 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] の順に展開し、[JDBC テーブルへの挿入] サンプル ワークフローに移動します。
- 3 [JDBC テーブルへの挿入] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - 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 挿入ステートメントを入力し、[次へ] をクリックします。

構文の例は次のとおりです。

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

6 行に挿入する値を入力します。

7 [送信] をクリックして、ワークフローを実行します。

JDBC テーブルから行を選択

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] を展開し、[JDBC テーブルの選択] サンプル ワークフローに移動します。
- 3 [JDBC テーブルの選択] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - 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 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 [送信] をクリックして、ワークフローを実行します。

JDBC テーブルからエントリを削除

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] の順に展開し、[JDBC テーブルからエントリを削除] サンプル ワークフローに移動します。
- 3 [JDBC テーブルからエントリを削除] サンプル ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - a 削除するユーザーエントリの最初の名前を入力します。
 - 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最後の名前を入力します。
 - 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。

5 SQL の削除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。

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

6 [送信] をクリックして、ワークフローを実行します。

JDBC テーブルからすべてのエントリを削除

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] の順に展開し、[JDBC テーブルからすべて削除] サンプル ワークフローに移動します。
- 3 [JDBC テーブルからすべて削除] サンプル ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - 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 の削除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。

```
DELETE FROM "table_name"
```

6 [送信] をクリックして、ワークフローを実行します。

JDBC テーブルのドロップ

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。

- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] の順に展開し、[JDBC ドロップ テーブル] サンプル ワークフローに移動します。
- 3 [JDBC ドロップ テーブル] サンプル ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 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。
- 5 SQL ドロップ 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。

```
DROP TABLE "table_name"
```

- 6 [送信] をクリックして、ワークフローを実行します。

完全な JDBC サイクルの実行

Orchestrator クライアントからワークフローを実行して、一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

- 1 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [JDBC] - [JDBC サンプル] を展開し、[JDBC 完全サイクル] サンプル ワークフローに移動します。
- 3 [JDBC 完全サイクル] サンプル ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - 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 データベースのエントリに使用する値を入力します。
- 6 [送信] をクリックして、ワークフローを実行します。

SQL プラグインの標準ワークフローの使用

SQL ワークフローを使用して SQL 操作を実行できます。

SQL プラグインのワークフロー ライブラリ

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。

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] - [SQL] - [構成] からデータベース構成ワークフローにアクセスできます。

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	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	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	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	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。

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] - [SQL] から SQL 操作ワークフローにアクセスできます。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SQL] の順に展開し、[テーブルの CRUD ワークフローの生成] ワークフローに移動します。
- 3 [テーブルの CRUD ワークフローの生成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 ワークフローを生成するテーブルを選択します。
- 5 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。

6 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。

7 (オプション) 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。

8 [送信] をクリックして、ワークフローを実行します。**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

8

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、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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフロー階層リストで、[ライブラリ] - [SSH] の順に展開し、「SSH ホストの追加」ワークフローに移動します。
- 4 「メールの設定」ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [ホスト名] テキスト ボックスに、Orchestrator 経由で SSH を使用してアクセスするホストの名前を入力します。
- 6 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。
- 7 (オプション) サーバ上でのエントリ パスを設定します。
 - 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。
- 8 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。

9 認証タイプを選択します。

オプション	アクション
はい	パスワードを入力して、パスワード認証を使用します。
いいえ	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て、キー認証を使用します。

10 [送信] をクリックして、ワークフローを実行します。

結果

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。

構成ワークフロー

SSH プラグインの「構成」カテゴリには、Orchestrator と SSH ホスト間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] - [SSH] - [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
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 ホストへの新しい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 プラグイン サンプル ワークフローの実行

Orchestrator クライアントから SSH プラグインを実行して、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

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

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。

■ 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。Orchestrator パブリック キーを SSH ホストへ登録するには、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

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

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 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

Orchestrator クライアントでワークフローを実行して、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

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。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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SSH] の順に展開し、[キー ペアの生成] ワークフローに移動します。
- 3 [キー ペアの生成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 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 [送信] をクリックして、ワークフローを実行します。

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

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SSH] の順に展開し、[キー ペアのパスフレーズの変更] ワークフローに移動します。
- 3 [キー ペアのパスフレーズの変更] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 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 [送信] をクリックして、ワークフローを実行します。

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。Orchestrator パブリック キーを SSH ホストへ登録するには、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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SSH] を展開し、[ホストに vCO パブリック キーを登録] ワークフローに移動します。
- 3 [ホストに vCO パブリック キーを登録] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 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 [送信] をクリックして、ワークフローを実行します。

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す。

SSH コマンドの実行

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SSH] を展開し、[SSH コマンドの実行] ワークフローに移動します。
- 3 [SSH コマンドの実行] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 SSH ホスト名または IP アドレスを入力して、[次へ] をクリックします。
- 5 実行する SSH コマンドを入力し、[次へ] をクリックします。

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 [はい] を選択してパスワード認証を使用し、[次へ] をクリックします。

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。

- 7 ユーザー名を入力し、[次へ] をクリックします。
- 8 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。
- 9 [送信] をクリックして、ワークフローを実行します。

SSH ホストからのファイルのコピー

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 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： 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。

- 2 ワークフローの階層リストで、[ライブラリ] - [SSH] を展開し、[SCP get コマンド] ワークフローに移動します。
- 3 [SCP get コマンド] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - a SSH ホスト名または IP アドレスを入力します。
 - b SSH 認証情報を入力します。
- 5 ファイル情報を入力します。
 - a ファイルをコピーする先の Orchestrator サーバ上のディレクトリのパスを入力します。
 - b 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。
- 6 [送信] をクリックして、ワークフローを実行します。

SSH ホストへのファイルのコピー

Orchestrator クライアントでワークフローを実行して、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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SSH] を展開し、[SCP put コマンド] ワークフローに移動します。
- 3 [SCP put コマンド] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 必要な情報を入力し、[次へ] をクリックします。
 - a SSH ホスト名または IP アドレスを入力します。
 - b SSH 認証情報を入力します。
- 5 ファイルの情報を入力します。
 - 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。
- 6 [送信] をクリックして、ワークフローを実行します。

XML プラグインの使用

9

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、『VMware vRealize Orchestrator を使用した開発』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

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](#)

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

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

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

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

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフロー階層リストで、[ライブラリ] - [XML] - [サンプル XML (簡易)] を開き、[簡易 XML ドキュメントの作成] ワークフローに移動します。
- 3 [簡易 XML ドキュメントの作成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 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 [送信] をクリックして、ワークフローを実行します。

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

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフロー階層リストで、[ライブラリ] - [XML] - [サンプル XML (簡易)] を開き、[ドキュメント ワークフローの要素を検索] ワークフローに移動します。
- 3 [ドキュメント ワークフローの要素を検索] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 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 [送信] をクリックして、ワークフローを実行します。

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、Orchestrator クライアントで完了している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

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフロー階層リストで、[ライブラリ] - [XML] - [サンプル XML (簡易)] を開き、[XML ドキュメントの変更] ワークフローに移動します。
- 3 [XML ドキュメントの変更] ワークフローを右クリックして、[ワークフローの開始] を選択します。
- 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 [送信] をクリックして、ワークフローを実行します。

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

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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフロー階層リストで、[ライブラリ] - [XML] - [サンプル XML (アドレス帳)] を開き、[アドレス帳の全テスト] ワークフローに移動します。
- 3 [アドレス帳の全テスト] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 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 [送信] をクリックして、ワークフローを実行します。

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。

Mail プラグインの使用

10

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。

- [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](#)

デフォルトの SMTP 接続の定義

Mail プラグインは、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： 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。
SMTP_HOST_UNREACHABLE エラーが発生することがあります。

手順

- 1 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [Mail] を展開し、「メールの設定」ワークフローに移動します。
- 4 「メールの設定」ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 必要な情報を入力します。

テキスト ボックス	説明
SMTP ホスト	SMTP サーバの IP アドレスまたはドメイン名を入力します。
SMTP ポート	SMTP 設定に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。
ユーザー名	有効なメール アカウントを入力します。 これは、Orchestrator がメールを送信するために使用するメール アカウントです。

テキスト ボックス	説明
パスワード	ユーザー名に関連付けられているパスワードを入力します。
送信元の名前とアドレス	Orchestrator によって送信されるすべてのメールに表示される送信者の情報を入力します。

6 [送信] をクリックして、ワークフローを実行します。

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、Orchestrator と SMTP サーバ間の操作をテストできます。

■ Mail プラグイン サンプル ワークフローへのアクセス

Mail プラグイン サンプル ワークフローにアクセスするには、Orchestrator クライアントを使用します。

■ Mail プラグイン サンプル ワークフロー

サンプルの「Mail プラグイン」ワークフローを統合して、カスタム ワークフローを拡張することができます。

Mail プラグイン サンプル ワークフローへのアクセス

Mail プラグイン サンプル ワークフローにアクセスするには、Orchestrator クライアントを使用します。

前提条件

ログイン時に使用したユーザー アカウントに 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。

手順

1 Orchestrator クライアントの [ワークフロー] ビューをクリックします。

2 階層リストで [ライブラリ] - [メール] の順に展開します。

次のステップ

サンプル ワークフローを確認し、実行します。

Mail プラグイン サンプル ワークフロー

サンプルの「Mail プラグイン」ワークフローを統合して、カスタム ワークフローを拡張することができます。

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] - [Mail] から Mail ワークフローにアクセスできます。

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。

ワークフロー名	説明
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。

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。

Enumeration プラグインの使用

12

Enumeration プラグインを使用して、ワークフローに一般的な列挙タイプを実装できます。

この章には、次のトピックが含まれています。

■ タイムゾーンコード

タイムゾーンコード

タイムゾーンコードは `Enums:MSTimeZone` 列挙の値として使用することができます。

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)

タイムゾーンコード	タイムゾーン名	説明
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	アソレス標準時	(GMT - 01:00) アソレス諸島
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、ブラチスラバ、ブダペスト、リュブリャナ、ブラハ
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
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

タイムゾーンコード	タイムゾーン名	説明
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
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) ブリスベン
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

タイムゾーンコード	タイムゾーン名	説明
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

Workflow Documentation プラグインの使用

13

Workflow Documentation プラグインを使用すると、特定のワークフローまたはワークフロー カテゴリに関する PDF ドキュメントを生成できます。

この章には、次のトピックが含まれています。

- Workflow Documentation プラグインのワークフロー ライブラリ
- ワークフロー ドキュメントの生成

Workflow Documentation プラグインのワークフロー ライブラリ

Workflow Documentation プラグインのワークフローを使用すると、特定のワークフローまたはワークフロー カテゴリに関する PDF ドキュメントを生成できます。

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] - [ワークフロー ドキュメント] 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
ワークフローのドキュメントの取得	選択したワークフローに関する情報を生成します。
ワークフロー カテゴリのドキュメントの取得	選択したワークフロー カテゴリに関する情報を生成します。

ワークフロー ドキュメントの生成

ワークフローまたはワークフロー フォルダに関する文書は、いつでも選択して PDF 形式でエクスポートできます。

エクスポートしたドキュメントには、選択したワークフローまたはフォルダ内のワークフローに関する詳細情報が記載されています。各ワークフローに関する情報には、ワークフローの名前、バージョン履歴、属性、パラメータ表示、ワークフローのスキーマ、ワークフローのアクションなどが含まれます。ドキュメントにはさらに、使用されたアクションのソース コードも含まれています。

手順

- 1 Orchestrator クライアントのドロップダウン メニューから、[実行] または [設計] を選択します。
- 2 [ワークフロー] ビューをクリックします。
- 3 ドキュメントを生成するワークフローまたはワークフロー フォルダのある場所に移動し、右クリックします。

4 [ドキュメントの生成] を選択します。

5 PDF ファイルを保存するフォルダを参照して指定し、ファイル名を入力し、[保存] をクリックします。

結果

選択したワークフローまたはフォルダ内のワークフローに関する情報を含む PDF ファイルが、システムに保存されます。

HTTP-REST プラグインの使用

14

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 プラグインを構成するには、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、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。

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [HTTP-REST] - [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
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。

ワークフロー名	説明
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。
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。

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 WinRM サービスで Kerberos 認証を有効にします。

- a 次のコマンドを実行して Kerberos 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 Kerberos 認証を有効にします。

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

- 2 WinRM クライアントで Kerberos 認証を有効にします。

- a 次のコマンドを実行して Kerberos 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 Kerberos 認証を有効にします。

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

- 3 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:Kerberos -u:user_name -p:password -encoding:utf-8
```

- 4 krb5.conf ファイルを作成し、次の場所に保存します。

オペレーティング システム	パス
Windows	C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\
Linux	/usr/java/jre-vmware/lib/security/ 外部 vRealize Orchestrator 用です。 /etc/krb5.conf vRealize Automation に組み込まれた vRealize Orchestrator 用です。

krb5.conf ファイルの構造は次のとおりです。

```
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
  kdc = kdc.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alm。 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、Orchestrator サーバ サービスを再起動する必要があります。

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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [HTTP-REST] - [構成] の順に展開し、[REST ホストの追加] ワークフローに移動します。
- 4 [REST ホストの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [名前] テキスト ボックスにホストの名前を入力します。

6 [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。

7 [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。

8 [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。

9 [はい] を選択して、REST ホスト証明書を受け入れます。

Orchestrator サーバのトラスト ストアに証明書が追加されます。

10 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	必要な認証パラメータを入力します。
OAuth 2.0	認証トークンを指定します。
基本	<p>基本アクセス認証を指定します。</p> <p>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。
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>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>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 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>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。

11 プロキシを使用するには、プロキシ サーバのアドレスとポートを入力します。

a (オプション) 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>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。

12 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。

13 (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。

14 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [HTTP-REST] - [構成] の順に展開し、[REST 操作の追加] ワークフローに移動します。
- 3 [REST 操作の追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 操作を追加する先のホストを選択します。
- 5 [名前] テキスト ボックスに操作の名前を入力します。
- 6 [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}

7 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。

8 [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

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [HTTP-REST] - [構成] の順に展開し、[REST ホストへのスキーマの追加] ワークフローに移動します。
- 3 [REST ホストへのスキーマの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 XSD スキーマを追加する先のホストを選択します。
- 5 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。

6 [送信] をクリックして、ワークフローを実行します。

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 開発者ガイド』を参照してください。

前提条件

- 管理者として 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [HTTP-REST] の順に展開し、[REST 操作からの新しいワークフローの生成] ワークフローに移動します。
- 3 [REST 操作からの新しいワークフローの生成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 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。
- 5 [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。
- 6 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。
- 7 [送信] をクリックして、ワークフローを実行します。

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] - [HTTP-REST サンプル] 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] - [HTTP-REST サンプル] 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。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [HTTP-REST] の順に展開し、[REST 操作の呼び出し] ワークフローに移動します。
- 3 [REST 操作の呼び出し] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 使用可能な操作のリストから REST 操作を選択します。
- 5 操作に必要な入力パラメータを指定します。
- 6 [送信] をクリックして、ワークフローを実行します。

SOAP プラグインの使用

15

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、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、SOAP ホストを管理できる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] ビューで [ライブラリ] - [SOAP] - [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
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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [SOAP] - [構成] の順に展開し、[SOAP ホストの追加] ワークフローに移動します。
- 4 [SOAP ホストの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [名前] テキスト ボックスにホストの名前を入力します。
- 6 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。

- 7 [接続タイムアウト] テキスト ボックスに、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。
- 8 [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。
- 9 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。

- 10 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>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。
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>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。

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>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 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>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。

11 [送信] をクリックして、ワークフローを実行します。

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。

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 が有効なりモート マシン上でコマンドを実行できます。

手順

- WinRM サービスで Kerberos 認証を有効にします。
 - 次のコマンドを実行して Kerberos 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
```
 - 次のコマンドを実行して Kerberos 認証を有効にします。

```
c:\> winrm set winrm/config/service/auth @{Kerberos="true"}
```
- WinRM クライアントで Kerberos 認証を有効にします。
 - 次のコマンドを実行して Kerberos 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
```
 - 次のコマンドを実行して Kerberos 認証を有効にします。

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

- 3 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:Kerberos -u:user_name -p:password -encoding:utf-8
```

- 4 krb5.conf ファイルを作成し、次の場所に保存します。

オペレーティング システム	パス
Windows	C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\
Linux	/Usr/java/jre-vmware/lib/security/ 外部 vRealize Orchestrator 用です。 /Etc/krb5.confvRealize Automation に組み込まれた vRealize Orchestrator 用です。

krb5.conf ファイルの構造は次のとおりです。

```
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
kdc = kdc.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alm。 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、Orchestrator サーバ サービスを再起動する必要があります。

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 開発者ガイド』を参照してください。

前提条件

- 管理者として 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SOAP] の順に展開し、[SOAP 操作からの新しいワークフローの生成] ワークフローに移動します。
- 3 [SOAP 操作からの新しいワークフローの生成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 使用可能な操作のリストから SOAP 操作を選択します。
- 5 [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。
- 6 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。
- 7 [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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの場所に移動します。
- 3 カスタム ワークフローを右クリックし、[ワークフローの開始] 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 [送信] をクリックして、ワークフローを実行します。

- 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。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SOAP] の順に展開し、[SOAP 操作の呼び出し] ワークフローに移動します。
- 3 [SOAP 操作の呼び出し] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 使用可能な操作のリストから SOAP 操作を選択します。
- 5 SOAP 操作に必要な入力パラメータを指定します。
- 6 [送信] をクリックして、ワークフローを実行します。
- 7 (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。

AMQP プラグインの使用

16

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 プラグインの標準ワークフローの使用](#)

AMQP プラグインの構成

AMQP プラグインを構成するには、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、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。

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [AMQP] - [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
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	新しい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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [AMQP] - [構成] の順に展開し、[ブローカの追加] ワークフローに移動します。
- 4 [ブローカの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 「ブローカの追加」ワークフローに必要な情報を指定します。

オプション	アクション
名前	ブローカの名前を入力します。
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 を使用する	SSL 証明書を使用するかどうかなを選択します。
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なを選択します。
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 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [AMQP] - [構成] の順に展開し、[キューへの登録] ワークフローに移動します。

- 3 [キューへの登録] ワークフローを右クリックして [ワークフローの開始] を選択します。
- 4 [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。
- 5 サブスクリプションの追加先のブローカを選択します。
- 6 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。
- 7 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [AMQP] の順に展開し、[ブローカの更新] ワークフローに移動します。
- 3 [ブローカの更新] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 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が表示されます。
- 5 対象のプロパティを編集します。
- 6 [送信] をクリックして、ワークフローを実行します。

AMQP プラグインの標準ワークフローの使用

「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。

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] - [AMQP] 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
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。

ワークフロー名	説明
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。
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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [AMQP] の順に展開し、[バインド] ワークフローに移動します。
- 3 [バインド] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 バインドを作成するブローカを選択します。
- 5 バインドの情報を指定します。

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。

- 6 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。

- 2 ワークフローの階層リストで、[ライブラリ] - [AMQP] の順に展開し、[キューの宣言] ワークフローに移動します。
- 3 [キューの宣言] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 キューを追加する先のブローカを選択します。
- 5 [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。
- 6 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。

- 7 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。

- 8 キュー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	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	キューを自動で削除しません。

- 9 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [AMQP] の順に展開し、[Exchange の宣言] ワークフローに移動します。
- 3 [Exchange の宣言] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 Exchange の追加先のブローカを選択します。

5 [名前] テキスト ボックスで、Exchange の名前を入力します。

6 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。

7 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。

8 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 を自動で削除しません。

9 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [AMQP] の順に展開し、[テキスト メッセージの送信] ワークフローに移動します。
- 3 [テキスト メッセージの送信] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 メッセージを送るブローカを選択します。

- 5 [Exchange の名前] テキストボックスで Exchange の名前を設定します。
- 6 [ルーティング キー] テキスト ボックスでルーティング キーを設定します。
- 7 [コンテンツ] テキスト ボックスで送信するメッセージを入力します。
- 8 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [AMQP] の順に展開し、[バインド解除] ワークフローに移動します。
- 3 [バインド解除] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 バインドを解除する対象のブローカを選択します。
- 5 バインドの情報を指定します。

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

- 6 [送信] をクリックして、ワークフローを実行します。

SNMP プラグインの使用

17

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 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。

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] - [SNMP] - [デバイス管理] 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
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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [SNMP] - [デバイス管理] を展開し、[SNMP デバイスの登録] ワークフローに移動します。
- 4 [SNMP デバイスの登録] ワークフローを右クリックして、[ワークフローの開始] を選択します。
- 5 [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 アドレスを使用してください。

- 6 (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。
- 7 (オプション) 高度な接続パラメータを設定するには、[はい] を選択します。
 - 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。

- 8 [送信] をクリックして、ワークフローを実行します。

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。

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] - [SNMP] - [クエリ管理] 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
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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SNMP] - [クエリ管理] の順に展開し、[SNMP デバイスへのクエリの追加] ワークフローに移動します。
- 3 [SNMP デバイスへのクエリの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 クエリを追加する先のデバイスを選択します。
- 5 [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。
- 6 [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。

7 (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。

8 [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

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、Orchestrator が SNMP トラップをリッ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。

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。

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] - [SNMP] - [トラップ管理ホスト] 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
SNMP トラップ ポートの設定	SNMP トラップをリッスンする Orchestrator のポートを設定します。
トラップ ホストの開始	Orchestrator で SNMP トラップのリッスンを開始します。
トラップ ホストの停止	Orchestrator で SNMP トラップのリッスンを停止します。

SNMP トラップ ポートの設定

ワークフローを実行して、Orchestrator が SNMP トラップをリッ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。ただし、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。

注： 互換性を高めるため、SNMP プラグインでは SNMP トラップをリッスンするポートがデフォルトで 4000 に設定されています。

前提条件

- 管理者として 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SNMP] - [トラップ ホスト管理] を展開し、[SNMP トラップ ポートの設定] ワークフローに移動します。
- 3 [SNMP トラップ ポートの設定] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 [ポート] テキスト ボックスで、Orchestrator が SNMP トラップをリスンするポート番号を入力します。
- 5 [送信] をクリックして、ワークフローを実行します。

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。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [SNMP] の順に展開し、[SNMP デバイスでのトラップの待機] ワークフローに移動します。
- 3 [SNMP デバイスでのトラップの待機] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 SNMP トラップを待機するデバイスを選択します。

- 5 (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。

- 6 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントのドロップダウン メニューから、[管理者] を選択します。
- 2 [ポリシー テンプレート] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [SNMP] を展開し、[SNMP トラップ ポリシー テンプレート] に移動します。
- 4 [SNMP トラップ ポリシー テンプレート] を右クリックして、[ポリシーの適用] を選択します。
- 5 [ポリシーの名前] テキスト ボックスに、作成するポリシーの名前を入力します。
- 6 (オプション) [ポリシーの説明] テキストボックスに、ポリシーの説明を入力します。
- 7 ポリシーを設定する SNMP デバイスを選択します。
- 8 [送信] 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。

- 9 [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 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 Orchestrator クライアントのドロップダウン メニューから、[管理者] を選択します。
- 2 [ポリシー テンプレート] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [SNMP] の順に展開し、[SNMP トラップ ホストのポリシー テンプレート] に移動します。
- 4 [SNMP トラップ ホストのポリシー テンプレート] を右クリックして、[ポリシーの適用] を選択します。
- 5 [ポリシーの名前] テキスト ボックスに、作成するポリシーの名前を入力します。
- 6 (オプション) [ポリシーの説明] テキストボックスに、ポリシーの説明を入力します。
- 7 インベントリ ツリーから Trap Host (Online) を選択します。
- 8 [送信] 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。
- 9 このポリシーを右クリックして [編集] を選択します。
- 10 **[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。
- 11 [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。
- 12 [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 ポリシーの編集では、スクリプティングやポリシー関連の権限と同様に、優先度や起動設定を変更できます。

前提条件

- 管理者として 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 Orchestrator クライアントの [ポリシー] ビューをクリックします。
- 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。

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] - [SNMP] 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
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	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。

Active Directory プラグインの使用

18

Active Directory プラグイン (Microsoft Active Directory 用の VMware vRealize Orchestrator 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 Orchestrator ワークフローを実行できます。

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。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。

Active Directory 構成ワークフロー

Active Directory プラグインの「構成」ワークフロー カテゴリには、Active Directory サーバを構成し SSL 証明書を管理できる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] - [Microsoft] - [Active Directory] - [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
Active Directory サーバの追加	Microsoft Active Directory サーバを構成します。
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。
Active Directory サーバの削除	Active Directory サーバの構成を削除します。

ワークフロー名	説明
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 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 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。

ワークフローはオブジェクト タイプに応じてカテゴリごとにグループ化されます。ワークフロー ライブラリにある標準ワークフローはカスタム ワークフローに統合することができます。

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。[インベントリ] ビューを使用して、認証要素を追加したり、Microsoft Active Directory オブジェクトに対してワークフローを実行したりできます。

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、[ツール] - [ユーザー環境設定] - [インベントリ] の順に移動し、[インベントリでコンテキスト メニューを使用する] チェック ボックスを選択します。オプションが有効になると、Orchestrator インベントリでオブジェクトを右クリックしたときに、そのオブジェクトに使用可能なすべてのワークフローが表示されます。

Active Directory プラグインのワークフロー ライブラリへのアクセス

Active Directory プラグイン ワークフロー ライブラリの要素にアクセスするには、Orchestrator クライアントを使用する必要があります。

手順

- 1 Orchestrator クライアントに管理者としてログインします。
- 2 [ワークフロー] ビューをクリックします。
- 3 階層リストで [ライブラリ] - [Microsoft] - [Active Directory] の順に展開し、選択項目を展開します。

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。

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。

[ライブラリ] - [Microsoft] - [Active Directory] - [コンピュータ] 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
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。

組織単位ワークフロー

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

[ライブラリ] - [Microsoft] - [Active Directory] - [組織単位]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
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。

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 ユーザー管理に関連するワークフローが含まれています。

[ライブラリ] - [Microsoft] - [Active Directory] - [ユーザー] 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
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	パスワードを指定しない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変更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、パスワードを指定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。

ワークフロー名	説明
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。

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 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。

[ライブラリ] - [Microsoft] - [Active Directory] - [ユーザー グループ] 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
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。
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。

Dynamic Types プラグインの使用

19

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。

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。

Dynamic Types プラグインと HTTP-REST プラグインを一緒に使用し、サードパーティの REST API サービスを Orchestrator に統合して、サードパーティのオブジェクトを Orchestrator タイプとして公開することができます。

- 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、これらの操作に対応する 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。

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] - [Dynamic Types] - [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
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。

PowerShell プラグインの使用

20

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。

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。

- [VMware vRealize OrchestratorPowerShell プラグインの概要](#)
- [PowerShell プラグインの構成](#)
- [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](#)
- [サンプル ワークフロー](#)
- [PowerShell プラグインの API へのアクセス](#)
- [PowerShell の結果の使用](#)
- [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](#)

VMware vRealize OrchestratorPowerShell 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す。

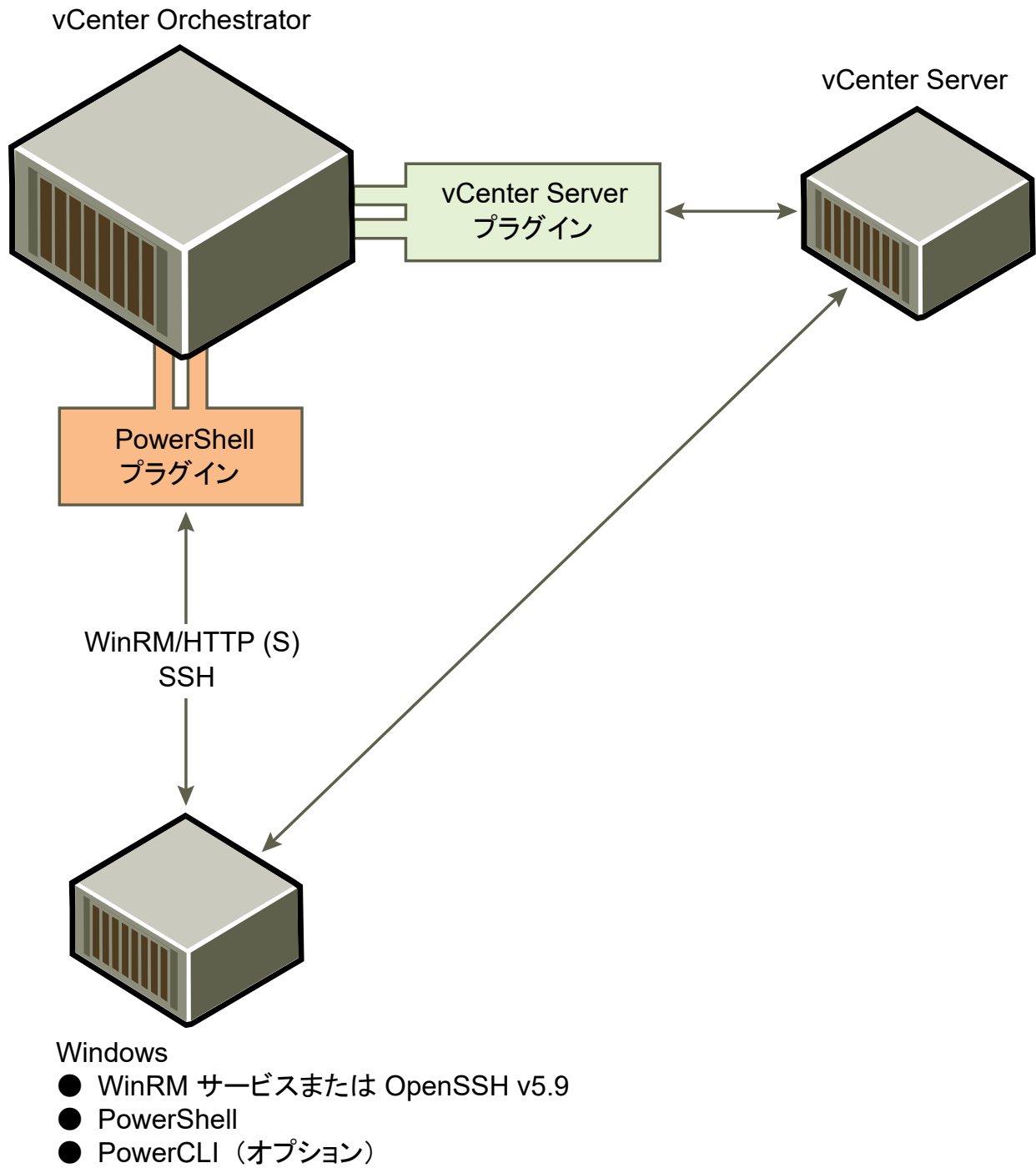
プラグインを使用して Orchestrator アクションおよびワークフローから PowerShell スクリプトおよびコマンドレットを呼び出したり、結果を使用したりします。このプラグインには標準ワークフローのセットが含まれています。プラグイン API を実装するカスタム ワークフローを作成することもできます。

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 20-1. 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。

vRealize Orchestrator の PowerShell プラグインにおける役割

Orchestrator 構成インターフェイスを使用して、PowerShell プラグインをインストールする必要があります。
 Orchestrator クライアントを使用してワークフローを実行および作成し、プラグイン API にアクセスします。

PowerShell プラグインは vRealize Orchestrator で機能します。Orchestrator は、拡張可能なワークフローのライブラリを提供して VMware vCenter インフラストラクチャおよびその他のテクノロジーを管理する、開発およびプロセス自動化プラットフォームです。

Orchestrator のオープンなプラグイン アーキテクチャにより、Orchestrator を各種の管理ソリューションと統合することができます。PowerShell は、プラグインを使用して Orchestrator と統合することができる、管理ソリューションの 1 つの例です。

プラグインの Windows PowerShell とのやりとり

プラグインを使用すると、Windows PowerShell ホストとやりとりして PowerShell スクリプトの呼び出しなどのタスクを実行する Orchestrator ワークフローを実行できます。

Windows PowerShell は、タスクベースのコマンドライン シェルであり、システム管理用に設計されているスクリプト言語です。

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 適切なサン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： 証明書サン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 WinRM サービスで Kerberos 認証を有効にします。

- a 次のコマンドを実行して Kerberos 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 Kerberos 認証を有効にします。

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

2 WinRM クライアントで Kerberos 認証を有効にします。

- a 次のコマンドを実行して Kerberos 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 Kerberos 認証を有効にします。

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

3 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:Kerberos -u:user_name -p:password -encoding:utf-8
```

4 krb5.conf ファイルを作成し、次の場所に保存します。

オペレーティング システム	パス
Windows	C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\
Linux	/usr/java/jre-vmware/lib/security/ 外部 vRealize Orchestrator 用です。 /etc/krb5.conf vRealize Automation に組み込まれた vRealize Orchestrator 用です。

krb5.conf ファイルの構造は次のとおりです。

```
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
kdc = kdc.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alm。 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、Orchestrator サーバ サービスを再起動する必要があります。

PowerShell プラグインの構成

PowerShell プラグインを構成するには、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。

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [PowerShell] - [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
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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [PowerShell] - [構成] の順に展開し、[PowerShell ホストの追加] ワークフローに移動します。

- 4 [PowerShell ホストの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [名前] テキスト ボックスにホストの名前を入力します。
- 6 [ホスト / IP] 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。

- 7 (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。

- 8 プラグインの接続先である PowerShell ホストのタイプを選択します。

- 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。

- 9 プラグインで 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。 <i>user@domain</i> 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。

- 10 [Shell のコード ページ] ドロップダウン メニューから PowerShell で使用されるエンコードのタイプを選択します。
- 11 [送信] をクリックして、ワークフローを実行します。

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 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。[インベントリ] ビューを使用して、認証要素を追加したり、PowerShell オブジェクトに対してワークフローを実行したりできます。

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。

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、[ツール] - [ユーザー環境設定] - [インベントリ] の順に移動し、[インベントリでコンテキスト メニューを使用する] チェック ボックスを選択します。オプションが有効になると、Orchestrator インベントリでオブジェクトを右クリックしたときに、そのオブジェクトに使用可能なすべてのワークフローが表示されます。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [PowerShell] の順に展開し、[PowerShell スクリプトの呼び出し] ワークフローに移動します。
- 3 [PowerShell スクリプトの呼び出し] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 スクリプトを実行する PowerShell ホストを選択します。
- 5 [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します。
- 6 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。

- 2 ワークフローの階層リストで、[ライブラリ] - [PowerShell] の順に展開し、[外部スクリプトの呼び出し] ワークフローに移動します。
- 3 [外部スクリプトの呼び出し] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 スクリプトを実行する PowerShell ホストを選択します。
- 5 [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。

- 6 [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。
- 7 [送信] をクリックして、ワークフローを実行します。

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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [PowerShell] - [生成] の順に展開し、[PowerShell スクリプトからのアクションの生成] ワークフローに移動します。
- 3 [PowerShell スクリプトからのアクションの生成] ワークフローを右クリックし、[ワークフローの開始] を選択します。

- 4 [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します。

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;
```

- 5 [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ます。
- 6 アクションの生成先とする既存のモジュールを選択します。
- 7 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。

- 8 [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、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、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 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 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 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 2 ワークフローの階層リストで、[ライブラリ] - [PowerShell] - [生成] の順に展開し、[PowerShell コマンドレットのアクションの生成] ワークフローに移動します。
- 3 [PowerShell コマンドレットのアクションの生成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 4 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。
- 5 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。

- 6 [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ます。
- 7 アクションの生成先とする既存のモジュールを選択します。
- 8 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。

- 9 [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。

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、プラグインで使え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 プラグインで使え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。

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [PowerShell] - [サンプル] - [コンバータ] 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
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。

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。

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [PowerShell] - [サンプル] コンバータ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
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。

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利用できる 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 Orchestrator クライアントに管理者としてログインします。
- 2 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。
 - Orchestrator クライアントから API Explorer にアクセスするには、Orchestrator クライアントのツールバーで [ツール] - [API Explorer] の順にクリックします。
 - 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 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。

方法	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 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。

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 認証でサーバを追加した後、正しく追加されていないと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、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 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

Orchestrator Appliance 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。

Multi-Node プラグインの使用

21

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、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。

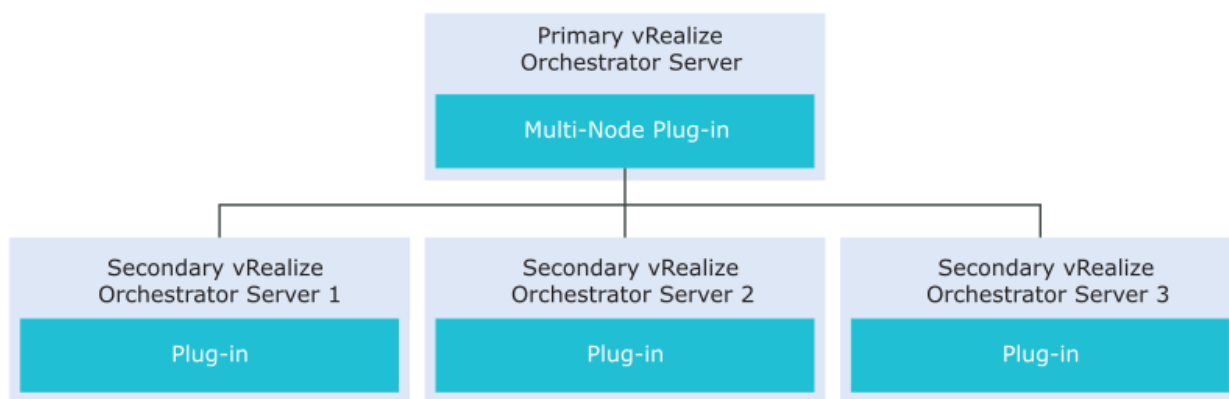
この章には、次のトピックが含まれています。

- [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 21-1. 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、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

「サーバ構成」ワークフロー カテゴリには、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。

Orchestrator クライアントの [ワークフロー] ビューで、[ライブラリ] - [Orchestrator] - [サーバ構成] 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
Orchestrator サーバの追加	プラグイン インベントリに Orchestrator サーバを追加します。
Orchestrator サーバの削除	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 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

プライマリおよびセカンダリの 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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [Orchestrator] - [サーバ構成] の順に展開し、[Orchestrator サーバの追加] ワークフローに移動します。
- 4 [Orchestrator サーバの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 新しいサーバの詳細情報を指定します。
- 6 接続を共有するかどうかを選択します。

オプション	説明
いいえ	リモートの 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 [送信] をクリックして、ワークフローを実行します。

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 Orchestrator サーバと、リモートの 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プライマリ 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、単純なタイプの出力パラメータは、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、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

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] - [Orchestrator] - [リモート実行] からこれらのワークフローにアクセスしてプロキシ ワークフローを作成できます。

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

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [Orchestrator] - [リモート実行] - [サーバ プロキシ] からこれらのワークフローにアクセスしてサーバ プロキシを管理できます。

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。[インベントリ] ビューを使用して、認証要素を追加したり、リモートの Orchestrator に対してワークフローを実行したりできます。

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、[ツール] - [ユーザー環境設定] - [インベントリ] の順に移動し、[インベントリでコンテキスト メニューを使用する] チェック ボックスを選択します。オプションが有効になると、Orchestrator インベントリでオブジェクトを右クリックしたときに、そのオブジェクトに使用可能なすべてのワークフローが表示されます。

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 [System] と呼ばれる上位のグループに属しています。システム オブジェクトには構成、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 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 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。

パッケージ

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [Orchestrator] - [リモート管理] - [パッケージ] からこれらのワークフローにアクセスしてリモート パッケージを管理できます。

ワークフロー名	説明
パッケージの削除	リモートの 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 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。

ワークフロー名	説明
ローカル サーバからパッケージを展開	ローカルの 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 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 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。

ワークフロー

Orchestrator クライアントの [ワークフロー] ビューで [ライブラリ] - [Orchestrator] - [リモート管理] - [ワークフロー] からこれらのワークフローにアクセスしてリモート ワークフローを管理できます。

ワークフロー名	
リモート ワークフローの削除	リモートの 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 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 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 Orchestrator クライアントに管理者としてログインします。
- 2 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립ティング] タブから API Explorer にアクセスします。
 - Orchestrator クライアントから API Explorer にアクセスするには、Orchestrator クライアントのツールバーで [ツール] - [API Explorer] の順にクリックします。
 - ワークフロー、ポリシー、アクション エディタの [スク립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립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 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

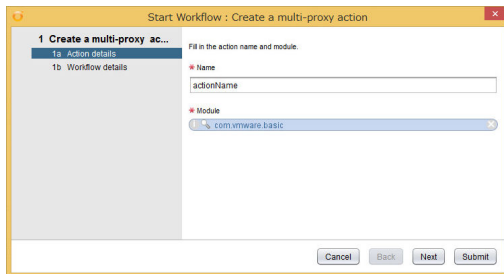
「マルチプロキシ アクションの作成」ワークフローを実行して、複数のサーバに対してワークフローを実行することができます。

リモートの 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

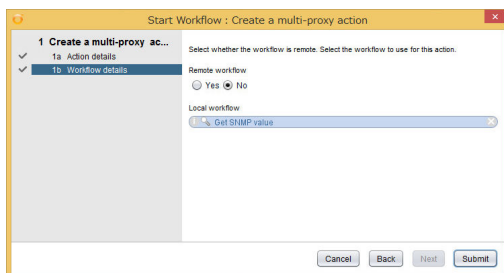
- 1 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [Orchestrator] - [リモート実行] を展開し、[マルチプロキシ アクションの作成] ワークフローに移動します。
- 4 [マルチプロキシ アクションの作成] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [アクション名] テキスト ボックスにアクションの名前を入力します。

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。

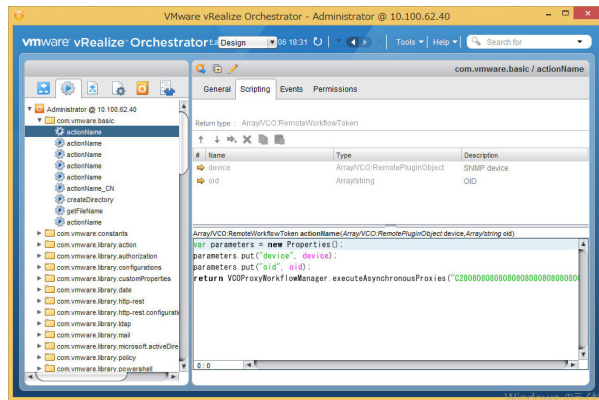
- 6 アクションを追加する先のモジュールを選択します。
- 7 ワークフローがローカルであるかリモートであるかを選択します。



- 8 このアクションに使用するワークフローを選択します。
- 9 [送信] をクリックして、ワークフローを実行します。

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。

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] - [Orchestrator] - [リモート実行] - [サーバ プロキシ] 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
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。
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 クライアントの [ワークフロー] ビューにある [ライブラリ] - [Orchestrator] - [リモート管理] - [ワークフロー] からこれらのワークフローにアクセスしてさらにメンテナンスすることができます。

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 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 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。

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 Orchestrator サーバからリモートの 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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。

- 3 ワークフローの階層リストで、[ライブラリ] - [Orchestrator] - [リモート管理] の順に展開し、[ローカル サーバからパッケージを展開] ワークフローに移動します。
- 4 [ローカル サーバからパッケージを展開] をクリックしてから、[ワークフローの開始] を選択します。
- 5 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。
- 6 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。
- 7 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。

- 8 [送信] をクリックして、ワークフローを実行します。

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。

vCloud Suite API (vAPI) プラグインの使用

22

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 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [VAPI] の順に展開し、[vAPI メタモデルのインポート] ワークフローに移動します。
- 4 [vAPI メタモデルのインポート] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [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。

6 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	<p>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。</p> <p>a 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。</p> <p>b 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。</p>

7 [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

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

- 1 Orchestrator クライアントに管理者としてログインします。
- 2 Orchestrator クライアントの [ワークフロー] ビューをクリックします。
- 3 ワークフローの階層リストで、[ライブラリ] - [VAPI] の順に展開し、[vAPI エンドポイントの追加] ワークフローに移動します。
- 4 [vAPI エンドポイントの追加] ワークフローを右クリックし、[ワークフローの開始] を選択します。
- 5 [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。
- 6 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	<p>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。</p> <p>a 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。</p> <p>b 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。</p>

7 [送信] をクリックして、ワークフローを実行します。

vCloud Suite API 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、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利用できる 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 Orchestrator クライアントに管理者としてログインします。
- 2 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。
 - Orchestrator クライアントから API Explorer にアクセスするには、Orchestrator クライアントのツールバーで [ツール] - [API Explorer] の順にクリックします。
 - 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。