# VMware vRealize Orchestrator のインストールおよび構成

2021 年 8 月 12 日

vRealize Orchestrator 8.5

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

**ヴイエムウェア株式会社**
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

# 目次

VMware vRealize Orchestrator のインストールと構成　6

1 VMware vRealize Orchestrator の概要　7
Orchestrator プラットフォームの主な機能　7
vRealize Orchestrator のユーザー ロール　9
vRealize Orchestrator のアーキテクチャ　10
vRealize Orchestrator Plug-in　11

2 vRealize Orchestrator システムの要件　12
デフォルトのアプライアンス コンポーネント　12
ハードウェア要件　13
スケーラビリティの上限　13
ネットワーク要件　13
ポートとエンドポイント　14
ブラウザのサポート　14
利用可能な言語に関するサポート　14

3 vRealize Orchestrator コンポーネントの設定　16
vCenter Server の設定　16
認証方法　16

4 vRealize Orchestrator のインストール　18
vRealize Orchestrator Appliance のダウンロードと展開　18
vRealize Orchestrator Appliance をパワーオンし、ホームページを開く　20
vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化　20

5 初期構成　22
スタンドアローン vRealize Orchestrator サーバの構成　22
vRealize Automation 認証でのスタンドアローン vRealize Orchestrator サーバの構成　22
vSphere 認証を使用したスタンドアローン vRealize Orchestrator サーバの構成　24
ライセンスを使用した vRealize Orchestrator 機能の有効化　25
vRealize Orchestrator データベース接続　26
証明書の管理　26
vRealize Orchestrator 証明書の管理　27
vRealize Orchestrator 用のカスタム TLS 証明書の生成　27
vRealize Orchestrator 用のカスタム TLS 証明書の設定　28
コントロール センターを使用した信頼されている証明書のインポート　31
vRealize Orchestrator プラグインの構成　31

vRealize Orchestrator プラグインを管理　31
vRealize Orchestrator プラグインのインストールまたはアップデート　32
プラグインの削除　33
vRealize Orchestrator の高可用性　33
スケーラビリティの上限　34
vRealize Orchestrator クラスタの設定　34
vRealize Orchestrator クラスタ ノードの削除　36
スタンドアローン vRealize Orchestrator 環境のスケールアウト　36
vRealize Orchestrator クラスタの監視　37
カスタマ エクスペリエンス改善プログラムの構成　38
VMware が受信する情報の種類　38
カスタマー エクスペリエンス向上プログラムへの参加または離脱　38

**6** vRealize Orchestrator API サービスの使用　40
REST API を使用した SSL 証明書の管理　40
REST API を使用した TLS 証明書の削除　40
REST API を使用した TLS 証明書のインポート　41
REST API を使用したキーストアの作成　42
REST API を使用したキーストアの削除　43
REST API を使用したキーの追加　43

**7** その他の設定オプション　44
認証の再構成　44
認証プロバイダの変更　44
認証パラメータの変更　45
ワークフローの実行のプロパティの設定　45
vRealize Orchestrator ログ ファイル　46
ログのパーシステンス　46
vRealize Orchestrator ログの設定　47
vRealize Log Insight とのログ統合の設定　47
vRealize Orchestrator での Syslog 統合の作成または上書き　48
vRealize Orchestrator での Syslog 統合の削除　49
Kerberos デバッグ ログの有効化　49
OpenTracing と Wavefront 拡張機能の有効化　50
OpenTracing 拡張機能の構成　51
Wavefront 拡張機能の構成　52
vRealize Orchestrator の時刻同期の有効化　53
vRealize Orchestrator の時刻同期を無効にする　54
vRealize Orchestrator Kubernetes CIDR の構成　54
vRealize Orchestrator の DNS 設定の更新　55

**8** 構成の使用事例とトラブルシューティング　57

vRealize Orchestrator サーバのビルド番号の確認　57

vSphere Web Client の vRealize Orchestrator プラグインの構成　57

実行中のワークフローのキャンセル　58

vRealize Orchestrator Server のデバッグの有効化　59

vRealize Orchestrator Appliance ディスクのサイズ変更　60

vRealize Orchestrator サーバのヒープ メモリ サイズの調整方法　61

Site Recovery Manager を使用した vRealize Orchestrator のディザスタ リカバリ　62

vSphere Replication のための仮想マシンの構成　63

保護グループの作成　63

リカバリ プランの作成　65

フォルダでのリカバリ プランの編成　66

リカバリ プランの編集　66

**9** システム プロパティの設定　68

ワークフローとアクションからサーバ ファイル システムにアクセスするための設定　68

vRealize Orchestrator システムへの書き込みアクセスを許可する、js-io-rights.conf ファイル内のルール　68

ワークフローとアクションからサーバ ファイル システムにアクセスするための設定　69

ワークフローとアクションからオペレーティング システム コマンドにアクセスするための設定　70

JavaScript から Java クラスにアクセスするための設定　71

カスタム タイムアウト プロパティの設定　72

vRealize Orchestrator SQL プラグインの JDBC コネクタの追加　73

**10** 次の手順　74

# VMware vRealize Orchestrator のインストールと構成

『VMware vRealize Orchestrator のインストールと構成』では、VMware® vRealize Orchestrator のインストールおよび構成に関する情報と手順が提供されています。

## 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。

# VMware vRealize Orchestrator の概要

1

VMware 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 製品およびサード パーティの技術を管理するための自動化された構成可能なプロセスを作成および実行できる拡張ワークフローのライブラリを提供します。

vRealize Orchestrator は、サービス デスク、変更管理システム、IT 資産管理システムなどの VMware およびサード パーティの両方のアプリケーションによる管理タスクおよび操作タスクを自動化します。

この章には、次のトピックが含まれています。


- Orchestrator プラットフォームの主な機能
- vRealize Orchestrator のユーザー ロール
- vRealize Orchestrator のアーキテクチャ
- vRealize Orchestrator Plug-in


## Orchestrator プラットフォームの主な機能

vRealize Orchestrator は、オーケストレーション ツールで必要な一般的な機能を提供するオーケストレーション プラットフォーム、サブシステムの制御を統合するプラグイン アーキテクチャ、およびワークフローのライブラリの 3 つの異なるレイヤーで構成されています。vRealize Orchestrator は、新しいプラグインやコンテンツを使用して拡張でき、REST API によって大規模なアーキテクチャに統合できるオープン プラットフォームです。

vRealize Orchestrator には、ワークフローの実行と管理に役立ついくつかの重要な機能が含まれています。

**永続性**

プロセス、ワークフローの状態、および vRealize Orchestrator の構成などの関連情報の保存に、本番レベルの PostgreSQL データベースが使用されます。

**集中管理**

vRealize Orchestrator にはプロセスを集中的に管理する方法があります。全バージョンの履歴を持つアプリケーション サーバベースのプラットフォームでは、同じストレージ場所にスクリプトおよびプロセス関連のプリミティブを保存できます。このようにして、バージョニングおよび適切な変更制御のないスクリプトがサーバに置かれないようにできます。

**チェックポイント処理**

ワークフローの各ステップがデータベースに保存されるため、サーバを再起動する必要がある場合に、データの損失を防ぎます。この機能は特に、長時間の処理において役立ちます。

**コントロール センター**

Web ベースのポータルであるコントロール センターは、ランタイム動作、ワークフロー監視、ワークフロー実行とシステム リソース間の相関のための集中管理インターフェイスを提供することによって、vRealize Orchestrator インスタンスの管理効率を高めます。

**バージョニング**

すべての vRealize Orchestrator プラットフォーム オブジェクトには、関連するバージョン履歴があります。バージョン履歴は、プロセスをプロジェクトのステージまたは場所に配布するときの基本的な変更管理に役立ちます。

**Git 統合**

vRealize Orchestrator Client は、Git リポジトリと統合することで、vRealize Orchestrator コンテンツのバージョンとソースの管理をさらに効率化することができます。Git を使用すると、複数の vRealize Orchestrator インスタンス間でワークフロー開発を管理できます。『VMware vRealize Orchestrator クライアントの使用』ガイドの「vRealize Orchestrator クライアントでの Git の使用」を参照してください。

**スクリプト エンジン**

Mozilla Rhino JavaScript エンジンには、vRealize Orchestrator Client プラットフォーム向けのビルディング ブロックを作成する方法が提供されています。このスクリプト エンジンは、基本バージョン管理、変数の型チェック、名前空間管理、および例外処理により強化されています。このエンジンは、次のビルディング ブロックで使用できます。

- アクション
- ワークフロー
- ポリシー

**ワークフロー エンジン**

ワークフロー エンジンを使用すると、ビジネス プロセスを自動化できます。この機能は、次のオブジェクトを使用して、ワークフローでの段階的なプロセスの自動化を作成します。

- vRealize Orchestrator Client が提供するワークフローおよびアクション
- ユーザーが作成するカスタム ビルディング ブロック
- プラグインが vRealize Orchestrator Client に追加するオブジェクト

ユーザー、他のワークフロー、スケジュール、またはポリシーがワークフローを開始できます。

**ポリシー エンジン**

ポリシー エンジンを使用すると、vRealize Orchestrator Client サーバまたはプラグイン テクノロジーでの変化する状態に反応して、イベントを監視および生成できます。ポリシーは、プラットフォームまたはプラグインからイベントを集計できるため、統合されたテクノロジーでの変化する状態の処理に役立ちます。

### vRealize Orchestrator Client

vRealize Orchestrator Client を使用してワークフローを作成、実行、編集、および監視します。また、vRealize Orchestrator Client を使用して、アクション、構成、ポリシー、およびリソース要素を管理することもできます。『vRealize Orchestrator クライアントの使用』を参照してください。

### 開発とリソース

vRealize Orchestrator のトップページから、vRealize Orchestrator で使用する、独自のプラグインを開発するためのリソースにすばやくアクセスできます。vRealize Orchestrator REST API を使用して vRealize Orchestrator サーバに要求を送信する方法の詳細も確認できます。

### セキュリティ

vRealize Orchestrator には、次の高度なセキュリティ機能があります。

- サーバ間でインポートおよびエクスポートされたコンテンツを署名して暗号化する公開鍵基盤 (PKI)。
- エクスポートされたコンテンツを表示、編集、および再配布する方法を制御するデジタル著作権管理 (DRM)。
- vRealize Orchestrator Client、vRealize Orchestrator サーバ間、および Web フロントエンドへの HTTPS アクセス間で暗号化された通信を提供する Transport Layer Security (TLS)。
- プロセスおよびこれらのプロセスによって操作されるオブジェクトへのアクセスを制御する、高度なアクセス権管理。

### 暗号化

vRealize Orchestrator は、文字列の暗号化用の 256 ビット暗号化キーを備えた FIPS 準拠の Advanced Encryption Standard (AES) を使用します。暗号化キーはランダムに生成され、クラスタの一部ではないアプリケーション全体にわたり一意となります。クラスタ内のすべてのノードが 1 つの暗号化キーを共有します。

## vRealize Orchestrator のユーザー ロール

vRealize Orchestrator には、グローバル ユーザー ロールの特定の責任に基づいたさまざまなツールおよびインターフェイスがあります。vRealize Orchestrator には、管理者グループに属してすべての権利を持つユーザー（管理者）、開発者（ワークフロー デザイナ）、トラブルシューティング ユーザー（閲覧者）、および限定的なアクセス権を持つユーザーがいます。

vRealize Orchestrator のユーザー ロールは、vRealize Orchestrator Client の [ロールの管理] メニューで管理します。vRealize Orchestrator Client でのユーザー ロールの構成の詳細については、VMware vRealize Orchestrator クライアントの使用ガイドの vRealize Orchestrator Client でのロールの割り当てを参照してください。

**注：** vRealize Automation で、または vRealize Automation ライセンスを使用して認証された vRealize Orchestrator 環境では、ユーザー ロールに vRealize Automation プラットフォームの ID およびアクセス管理サービスが割り当てられています。『VMware vRealize Orchestrator クライアントの使用』の「vRealize Automation での vRealize Orchestrator クライアントのロールの設定」を参照してください。

| ユーザー ロール | 説明 |
| --- | --- |
| 管理者 | このユーザーには、特定のグループによって作成されたコンテンツを含む、vRealize Orchestrator プラットフォームのすべての機能およびコンテンツへのフル アクセス権があります。管理者ユーザーの主な責任は次のとおりです。<br>■ vRealize Orchestrator のインストールと構成。<br>■ vRealize Orchestrator Client にユーザーを追加し、ロールを割り当て、グループを作成および削除する。『VMware vRealize Orchestrator クライアントの使用』ガイドの「vRealize Orchestrator クライアントでのグループの作成」を参照してください。<br>■ vRealize Orchestrator 環境の開発者用に Git リポジトリとの統合を作成する。『VMware vRealize Orchestrator クライアントの使用』の「Git リポジトリとの接続の設定」を参照してください。<br>■ ワークフローの検証やワークフロー スクリプトのデバッグなどの機能を使用して vRealize Orchestrator 環境をトラブルシューティングする。 |
| 閲覧者 | このユーザーには、すべてのグループおよびグループ コンテンツを含む、vRealize Orchestrator Client に対する読み取り専用アクセス権があります。このユーザーは、コンテンツを表示できますが、作成、編集、実行することはできません。また、ワークフローの実行、ワークフローの実行ログ、またはパッケージをエクスポートすることもできません。閲覧者がグループ権限によって制限されることはありません。<br>**注：** 閲覧者ロールは、vRealize Automation で認証された vRealize Orchestrator のインスタンスに対してのみサポートされます。デフォルトでは、このロールは vRealize Automation ロールにマッピングされていないため、明示的にユーザーに割り当てる必要があります。 |
| ワークフロー デザイナ | このユーザーは、オブジェクトを作成および編集することによって vRealize Orchestrator プラットフォームの機能を拡張します。ワークフロー デザイナは、vRealize Orchestrator Client の管理およびトラブルシューティング機能にはアクセスできません。ワークフロー デザイナの主な責任は次のとおりです。<br>■ ワークフロー、アクション、ポリシー、構成要素などの vRealize Orchestrator オブジェクトを作成、編集、実行、削除する。<br>■ ワークフローの実行をスケジュール設定する。『VMware vRealize Orchestrator クライアントの使用』ガイドの「vRealize Orchestrator クライアントでのワークフローのスケジューリング」を参照してください。<br>■ ワークフロー デザイナによって作成されたコンテンツを、割り当てられているグループに追加する。<br>■ vRealize Orchestrator コンテンツ インベントリへのローカルの変更を接続された Git リポジトリにプッシュする。『VMware vRealize Orchestrator クライアントの使用』の「Git リポジトリへの変更のプッシュ」を参照してください。 |
| 制限された権限を持つユーザー | ロールが割り当てられていないユーザーも vRealize Orchestrator Client にログインできますが、クライアント機能およびコンテンツへのアクセスは制限されます。グループに割り当てられたユーザーは、そのグループに含まれているコンテンツを表示および実行できます。 |

# vRealize Orchestrator のアーキテクチャ

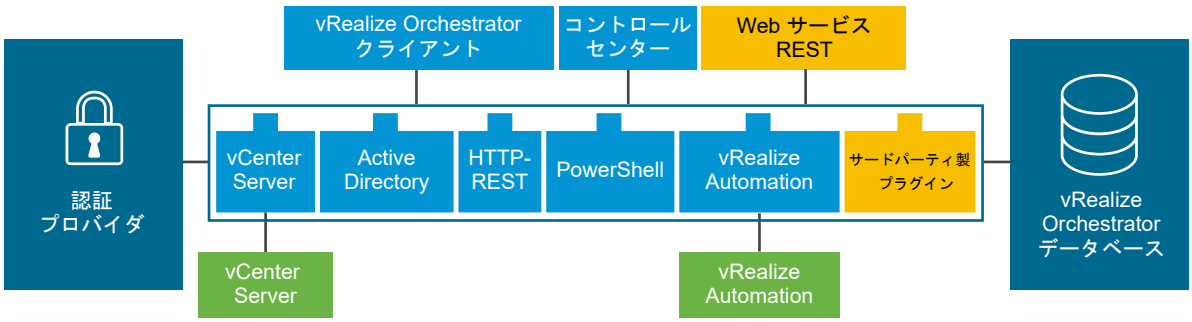
vRealize Orchestrator には、オーケストレーション プロセスを自動化するワークフローを作成および実行できる、ワークフロー ライブラリおよびワークフロー エンジンが含まれています。ワークフローは、vRealize Orchestrator が一連のプラグインを介してアクセスするさまざまなテクノロジーのオブジェクトに対して実行されます。

vRealize Orchestrator では、vCenter Server 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提供されており、プラグインで公開されているさまざまな環境においてタスクをオーケストレーションできます。

vRealize Orchestrator では、外部のサードパーティ製アプリケーションをオーケストレーション プラットフォームに接続するためのオープン アーキテクチャも提供されています。自身で定義したプラグイン テクノロジーのオブジェクトに対してワークフローを実行できます。vRealize Orchestrator は認証プロバイダに接続してユーザー アカウントを管理し、事前構成済み PostgreSQL データベースに接続して実行するワークフローからの情報を保存

します。vRealize Orchestrator、それによって公開されるオブジェクト、および vRealize Orchestrator ワークフローに、vRealize Orchestrator Client または Web サービスを介してアクセスできます。vRealize Orchestrator ワークフローおよびサービスの監視と構成は、vRealize Orchestrator Client とコントロール センターを介して行われます。

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




# vRealize Orchestrator Plug-in

プラグインでは、vRealize Orchestrator を使用して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、それらを制御できます。vRealize Orchestrator プラグインで外部テクノロジーを公開することで、外部テクノロジーのオブジェクトと関数にアクセスするワークフローにオブジェクトおよび関数を組み込むことができます。

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

vRealize Orchestrator には、VMware vCenter Server API および E メール機能などのテクノロジーをワークフローに組み込むために使用できる一連の標準プラグインが用意されています。プラグインを使用することで、新しい IT サービスの提供を自動化したり、既存のインフラストラクチャおよびアプリケーション サービスの機能を適用したりできます。また、vRealize Orchestrator 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。

VMware が開発する vRealize Orchestrator プラグインは、`.vmoapp` ファイルとして配布されます。

vRealize Orchestrator プラグインの詳細については、『VMware vRealize Orchestrator プラグインの使用』を参照してください。

サードパーティの vRealize Orchestrator プラグインの詳細については、VMware Marketplace を参照してください。

# vRealize Orchestrator システムの要件

2

システムは vRealize Orchestrator が適切に機能するうえで必要な技術要件を満たしている必要があります。

vCenter Server、vSphere Web Client、vRealize Automation、および他の VMware ソリューションのサポートされているバージョンのリストについては、VMware 製品の相互運用性マトリックス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。

- vRealize Orchestrator Appliance のコンポーネント
- vRealize Orchestrator Appliance のハードウェア要件
- vRealize Orchestrator スケーラビリティの上限
- vRealize Orchestrator のネットワーク要件
- vRealize Orchestrator ポートとエンドポイント
- vRealize Orchestrator でサポートされるブラウザ
- 国際化およびローカライズのサポートのレベル

## vRealize Orchestrator Appliance のコンポーネント

vRealize Orchestrator Appliance は、コンテナで実行される、Photon ベースの仮想アプライアンスです。

vRealize Orchestrator Appliance には、次のコンポーネントが含まれています。

- インフラストラクチャ レベルの Kubernetes レイヤー。
- 事前構成済み PostgreSQL データベース。
- コア vRealize Orchestrator サービス：サーバ サービス、コントロール センター サービス、およびオーケストレーション ユーザー インターフェイス サービス。

vRealize Orchestrator Appliance のデフォルトのデータベース構成は動作準備済みです。

**注：** vRealize Orchestrator Appliance を本番環境で使用するには、vRealize Automation または vSphere を使用して認証するように vRealize Orchestrator サーバを設定する必要があります。スタンドアローン vRealize Orchestrator サーバの構成を参照してください。

# vRealize Orchestrator Appliance のハードウェア要件

vRealize Orchestrator Appliance は、コンテナで実行される、事前構成された Photon ベースの仮想マシンです。アプライアンスを導入する前に、システムがハードウェアの最小要件を満たしていることを確認します。

vRealize Orchestrator Appliance には、次のハードウェア要件があります。

- 4 個の CPU
- 12 GB のメモリ
- 200 GB のハード ディスク

vRealize Orchestrator サーバには 8 GB 以上の空きメモリが必要なため、デフォルトのメモリ サイズを減らさないでください。

# vRealize Orchestrator スケーラビリティの上限

スケーラビリティの上限の表は、vRealize Orchestrator 8.x 展開での推奨される上限を示しています。

| コンポーネント | 拡張のターゲット | 詳細情報 |
|---|---|---|
| 仮想マシン | 35,000 | |
| vCenter Server 接続 | 10 | vCenter Server の設定を参照してください。 |
| クラスタ内のアクティブ ノード | 3 | vRealize Orchestrator クラスタの設定を参照してください。 |
| 同時実行中のワークフロー | ノードあたり 300 | ワークフローの実行のプロパティの設定を参照してください。 |
| キューに入れられた実行中のワークフロー | ノードあたり 10,000 | |
| 保持されるワークフローの実行 | ノードあたり 100 | |
| ログ イベントの有効期間（日） | 15 | |

# vRealize Orchestrator のネットワーク要件

各 vRealize Orchestrator ノードにはネットワークのセットアップが必要です。

vRealize Orchestrator のネットワーク要件は次のとおりです。

- 単一の固定 IPv4 およびネットワーク アドレス
- アクセス可能な DNS サーバが手動で設定されていること
- DNS サーバを介して正引きと逆引きの両方で解決できる有効な完全修飾ドメイン名 (FQDN) が手動で設定されていること

**注：** インストール後の IP アドレスの変更またはホスト名の変更はサポートされていません。変更すると、セットアップが破損し、復元できません。

# vRealize Orchestrator ポートとエンドポイント

vRealize Orchestrator Kubernetes サービスには、2 台のエンドポイントと複数の主なネットワーク ポートが含まれています。

## vRealize Orchestrator のネットワーク ポート

vRealize Orchestrator には、ポート 443 経由でアクセスできます。443 ポートは、インストール中に生成される自己署名証明書で保護されており、ユーザーが置き換えることはできません。外部のロード バランサを使用している場合、ポート 443 で負荷が分散されるように設定する必要があります。

すべての vRealize Orchestrator ポートを表示するには、ポートとプロトコル ツールを参照してください。

## vRealize Orchestrator のエンドポイント

vRealize Orchestrator クライアントおよびコントロール センターのサービスには、次のエンドポイントでアクセスできます。

| サービス | エンドポイント |
| --- | --- |
| vRealize Orchestrator クライアント | `https://your_orchestrator_FQDN/orchestration-ui` |
| コントロール センター | `https://your_orchestrator_FQDN/vco-controlcenter` |

# vRealize Orchestrator でサポートされるブラウザ

ブラウザで vRealize Orchestrator がサポートされていることを確認します。

vRealize Orchestrator Client およびコントロール センターにアクセスするには、次のブラウザのいずれかを使用する必要があります。

- Microsoft Edge
- Mozilla Firefox
- Google Chrome

# 国際化およびローカライズのサポートのレベル

vRealize Orchestrator コントロール センターおよび vRealize Orchestrator Client には、英語以外のオペレーティング システム、英語以外のデータ形式、およびコントロール センターとクライアント ユーザー インターフェイスでの複数言語のサポートが含まれています。

vRealize Orchestrator コントロール センターおよび vRealize Orchestrator Client では、英語以外のオペレーティング システムの使用、英語以外の入力および出力、および日付、時刻、数字などのデータにおける英語以外の形式もサポートされています。

vRealize Orchestrator および vRealize Orchestrator Client のユーザー インターフェイスは、次の言語にローカライズされています。

- スペイン語
- フランス語
- ドイツ語
- 繁体字中国語
- 簡体字中国語
- 韓国語
- 日本語
- イタリア語
- オランダ語
- ブラジルポルトガル語
- ロシア語

# vRealize Orchestrator コンポーネントの設定

3

vRealize Orchestrator Appliance をダウンロードして展開すると、vRealize Orchestrator サーバが事前構成されます。展開後、サービスは自動的に開始します。

vRealize Orchestrator 設定の可用性とスケーラビリティを強化するには、以下のガイドラインに従います。

- 認証プロバイダをインストールして構成し、このプロバイダと連携して使用できるように vRealize Orchestrator を構成します。『スタンドアローン vRealize Orchestrator サーバの構成』を参照してください。
- クラスタ化された vRealize Orchestrator 環境では、ロード バランシング サーバをインストールして設定することにより、vRealize Orchestrator サーバ間でワークロードを分散します。

この章には、次のトピックが含まれています。


- vCenter Server の設定
- 認証方法


## vCenter Server の設定

vRealize Orchestrator の設定で vCenter Server インスタンスの数を大きくすると、vRealize Orchestrator によって管理されるセッションが増えます。アクティブなセッションの数が多すぎて vCenter Server の接続が 10 件を超えると、vRealize Orchestrator の操作がタイムアウトになる可能性があります。

vCenter Server でサポートされるバージョンのリストについては、VMware 製品の相互運用性マトリックスを参照してください。

**注：** ネットワークのバンド幅と遅延が適切な場合は、vRealize Orchestrator セットアップ中に複数の仮想マシンで複数の vCenter Server インスタンスを実行できます。LAN を使用して vRealize Orchestrator と vCenter Server の間の通信を向上する場合、100 Mb が必須です。

## 認証方法

ユーザー権限を認証し、管理するには、vRealize Orchestrator で vRealize Automation または vSphere サーバ インスタンスのいずれかに接続する必要があります。

vRealize Orchestrator Appliance をダウンロードおよび展開するときは、vRealize Automation または vSphere 認証を使用してサーバを設定する必要があります。『スタンドアローン vRealize Orchestrator サーバの構成』を参照してください。

**注：** vRealize Automation での vRealize Orchestrator 8.x 認証は vRealize Automation 8.x でのみサポートされます。

# vRealize Orchestrator のインストール

4

vRealize Orchestrator は、サーバ コンポーネントとクライアント コンポーネントから構成されています。

vRealize Orchestrator を使用するには、vRealize Orchestrator Appliance を展開し、vRealize Orchestrator サーバを構成する必要があります。

デフォルトの vRealize Orchestrator 設定を変更するには、vRealize Orchestrator コントロール センターを使用します。

この章には、次のトピックが含まれています。


- vRealize Orchestrator Appliance のダウンロードと展開


## vRealize Orchestrator Appliance のダウンロードと展開

vRealize Orchestrator のコンテンツおよびサービスにアクセスするには、まず vRealize Orchestrator Appliance をダウンロードして展開する必要があります。

**前提条件**

- vCenter Server インスタンスが実行されていることを確認します。vCenter Server のバージョンは 6.0 以降である必要があります。
- vRealize Orchestrator Appliance を展開する先のホストが、ハードウェアの最小要件を満たしていることを確認します。vRealize Orchestrator Appliance のハードウェア要件を参照してください。
- システムが分離されておりインターネットにアクセスできない場合は、VMware Web サイトからアプライアンスの `.ova` ファイルをダウンロードする必要があります。

**手順**

1. 管理者として vSphere Web Client にログインします。
2. 仮想マシンの有効な親オブジェクトであるインベントリ オブジェクト（データセンター、フォルダ、クラスタ、リソース プール、ホストなど）を選択します。
3. [アクション] - [OVF テンプレートのデプロイ] の順に選択します。
4. `.ova` ファイルのファイル パスまたは URL を入力し、[次へ] をクリックします。
5. vRealize Orchestrator Appliance の名前と場所を入力し、[次へ] をクリックします。

6 アプライアンスの実行対象としてホスト、クラスタ、リソース プール、または vApp を選択し、[次へ] をクリックします。

7 展開の詳細を確認し、[次へ] をクリックします。

8 使用許諾契約書の条項に同意して、[次へ] をクリックします。

9 vRealize Orchestrator Appliance に使用するストレージ フォーマットを選択します。

| 形式 | 説明 |
| --- | --- |
| **シック プロビジョニング (Lazy Zeroed)** | 仮想ディスクをデフォルトのシック形式で作成します。仮想ディスクの作成時に、仮想ディスクに必要な容量が割り当てられます。物理デバイスにデータが残っている場合は、そのデータは仮想ディスク作成中に消去されませんが、後で仮想マシンから初めて書き込むときにオン デマンドで消去できます。 |
| **シック プロビジョニング (Eager Zeroed)** | フォルト トレランスなどのクラスタ機能をサポートします。仮想ディスクの作成時に、仮想ディスクに必要な容量が割り当てられます。物理デバイスにデータが残っている場合は、そのデータは仮想ディスクの作成時に消去されます。この形式でのディスクの作成には、他の形式でのディスク作成時と比較すると、はるかに長い時間を要することがあります。 |
| **シン プロビジョニング形式** | ハード ディスク容量を節約します。シン ディスクの場合、選択したディスク サイズの値に応じて、ディスクに必要な容量と同じデータストア容量をプロビジョニングします。シン ディスクは最初は小さく、初期処理に必要なデータストア容量のみを使用します。 |

10 [次へ] をクリックします。

11 ネットワークを設定し、root パスワードを入力します。

vRealize Orchestrator Appliance のネットワークを設定する場合は、IPv4 プロトコルを使用する必要があります。DHCP と固定ネットワークの両方の設定で、vRealize Orchestrator Appliance の完全修飾ドメイン名 (FQDN) を追加する必要があります。

展開した vRealize Orchestrator Appliance のシェルに表示されるホスト名が *photon-machine* の場合、上記のネットワーク構成要件は満たされていません。

12 (オプション) SSH アクセスを有効にするなど、vRealize Orchestrator Appliance の追加のネットワーク設定を行います。

**注：** Kubernetes ネットワークを構成する場合、内部クラスタ CIDR と内部サービス CIDR の値は、少なくとも 1024 台のホストを許可する必要があります。この要件のため、ネットワーク マスク値は 22 以下である必要があります。22 より大きいネットワーク マスク値は無効です。Kubernetes ネットワーク プロパティのデフォルト値は次のとおりです。

| Kubernetes network property | Default value | Property description |
| --- | --- | --- |
| Kubernetes 内部クラスタ CIDR | 10.244.0.0/22 | Kubernetes クラスタ内で実行されているポッドに使用される CIDR。 |
| Kubernetes 内部サービス CIDR | 10.244.4.0/22 | Kubernetes クラスタ内の Kubernetes サービスに使用される CIDR。 |

**注：** 展開後に Kubernetes CIDR ネットワーク プロパティを変更することもできます。vRealize Orchestrator Kubernetes CIDR の構成を参照してください。

13 （オプション）vRealize Orchestrator Appliance で FIPS モードを有効にするには、[FIPS モード] を **厳密** に設定します。

**注：** FIPS 140-2 の有効化は、新しい vRealize Orchestrator 環境でのみサポートされます。環境で FIPS モードを有効にする場合は、インストール中に実行する必要があります。

14 [次へ] をクリックします。

15 [設定の確認] 画面の内容を確認して [終了] をクリックします。

#### 結果

vRealize Orchestrator Appliance が正常に展開されました。

#### 次のステップ

vRealize Orchestrator Appliance コマンドラインに root としてログインし、DNS 正引き参照や逆引き参照が実行可能なことを確認します。

- DNS 正引き参照を実行するには、`nslookup` *your_orchestrator_FQDN* コマンドを実行します。このコマンドによって vRealize Orchestrator Appliance IP アドレスが返されます。
- DNS 逆引き参照を実行するには、`nslookup` *your_orchestrator_IP* コマンドを実行します。このコマンドによって vRealize Orchestrator Appliance FQDN が返されます。

**注：** 展開中に SSH を有効にしていない場合は、vSphere Web Client の仮想マシン コンソールから DNS 参照を実行することもできます。

## vRealize Orchestrator Appliance をパワーオンし、ホームページを開く

スタンドアローンの vRealize Orchestrator Appliance を使用するには、最初にパワーオンする必要があります。

#### 手順

1 vSphere Web Client に管理者としてログインします。

2 vRealize Orchestrator Appliance を右クリックし、[電源] - [パワーオン] の順に選択します。

3 Web ブラウザで、OVA の展開時に設定した vRealize Orchestrator Appliance 仮想マシンのホスト アドレスに移動します。

https://*your_orchestrator_FQDN*/vco

## vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化

SSH による vRealize Orchestrator Appliance へのアクセスを有効または無効にすることができます。

#### 前提条件

- vRealize Orchestrator Appliance をダウンロードして展開します。
- vRealize Orchestrator Appliance が稼動し、実行されていることを確認します。

**手順**

1. vRealize Orchestrator Appliance コマンド ラインに root としてログインします。
2. SSH アクセスを有効にするには、`/usr/bin/toggle-ssh enable` コマンドを実行します。
3. SSH アクセスを無効にするには、`/usr/bin/toggle-ssh disable` コマンドを実行します。

# 初期構成

5

vRealize Orchestrator を使用してタスクの自動化とシステムおよびアプリケーションの管理を開始する前に、vRealize Orchestrator コントロール センターを使用して外部の認証プロバイダを設定する必要があります。また、vRealize Orchestrator コントロール センターを使用して、ライセンスおよび証明書の情報の管理、プラグインのインストール、vRealize Orchestrator クラスタの状態の監視などの追加の設定タスクを実行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。


- スタンドアローン vRealize Orchestrator サーバの構成
- ライセンスを使用した vRealize Orchestrator 機能の有効化
- vRealize Orchestrator データベース接続
- 証明書の管理
- vRealize Orchestrator プラグインの構成
- vRealize Orchestrator の高可用性
- カスタマ エクスペリエンス改善プログラムの構成


## スタンドアローン vRealize Orchestrator サーバの構成

vRealize Orchestrator Appliance は事前構成された Photon ベースの仮想マシンですが、vRealize Orchestrator コントロール センターおよび vRealize Orchestrator Client のすべての機能にアクセスする前に、認証プロバイダを設定しておく必要があります。

### vRealize Automation 認証でのスタンドアローン vRealize Orchestrator サーバの構成

vRealize Orchestrator Appliance を使用できるようにするには、ホストの設定および認証プロバイダを構成する必要があります。vRealize Automation で認証するように vRealize Orchestrator を構成できます。vRealize Automation 8.x で vRealize Automation 認証を使用します。

**前提条件**

- vRealize Orchestrator Appliance の最新バージョンをダウンロードして導入します。vRealize Orchestrator Appliance のダウンロードと展開を参照してください。

- vRealize Automation 8.x をインストールおよび構成し、vRealize Automation サーバが実行されていることを確認します。vRealize Automation のドキュメントを参照してください。

  **重要：** vRealize Automation 認証プロバイダの製品バージョンは、vRealize Orchestrator 環境の製品バージョンと一致する必要があります。たとえば、vRealize Orchestrator 8.5 環境を認証するには、vRealize Automation 8.5 環境を使用する必要があります。

クラスタを作成するには、次の操作を実行します。

- vRealize Orchestrator の複数のインスタンス間でトラフィックを分散するロード バランサを設定します。VMware vRealize Orchestrator 8.x のロード バランシング ガイドを参照してください。

**手順**

1 コントロール センターにアクセスして、構成ウィザードを開始します。

  a https://*your_orchestrator_FQDN*/vco-controlcenter に移動します。

  b OVA の展開時に入力したパスワードを使用して root としてログインします。

2 認証プロバイダを構成します。

  a [認証プロバイダを設定] 画面で、[認証モード] ドロップダウン メニューから [vRealize Automation] を選択します。

  b [ホスト アドレス] テキスト ボックスに vRealize Automation のホスト アドレスを入力し、[接続] をクリックします。

  vRealize Automation ホスト アドレスの形式は、https://*your_vra_hostname* にする必要があります。

  c [証明書を承諾] をクリックします。

  d vRealize Orchestrator を構成する vRealize Automation 組織の所有者の認証情報を入力します。[登録] をクリックします。

  e [変更を保存] をクリックします。

  設定が正常に保存されたことを示すメッセージが表示されます。

**結果**

これで、vRealize Orchestrator サーバの構成が正常に完了しました。

**次のステップ**

- [CSP] が [ライセンス] 画面で構成されたライセンス プロバイダであることを確認します。
- [設定を検証] ページで、ノードが正しく構成されていることを確認します。

  **注：** 認証プロバイダの構成に従って、vRealize Orchestrator サーバは 2 分後に自動的に再起動します。認証直後に構成を確認すると、無効な構成ステータスが返される可能性があります。

# vSphere 認証を使用したスタンドアローン vRealize Orchestrator サーバの構成

vSphere 認証モードを使用して、vRealize Orchestrator サーバを vCenter Single Sign-On サーバに登録します。vCenter Single Sign-On 認証は vCenter Server 6.0 以降で使用します。

**前提条件**

- vRealize Orchestrator Appliance の最新バージョンをダウンロードして導入します。vRealize Orchestrator Appliance のダウンロードと展開を参照してください。
- vCenter Single Sign-On の実行中に vCenter Server をインストールし、構成します。vSphere のドキュメントを参照してください。

クラスタを作成するには、次の操作を実行します。

- vRealize Orchestrator の複数のインスタンス間でトラフィックを分散するロード バランサを設定します。VMware vRealize Orchestrator 8.x のロード バランシング ガイドを参照してください。

**手順**

1. コントロール センターにアクセスして、構成ウィザードを開始します。
   a. https://*your_orchestrator_FQDN*/vco-controlcenter に移動します。
   b. OVA の展開時に入力したパスワードを使用して root としてログインします。
2. 認証プロバイダを設定します。
   a. [認証プロバイダを設定] ページで、[認証モード] ドロップダウン メニューから [vSphere] を選択します。
   b. [ホスト アドレス] テキスト ボックスに、vCenter Single Sign-On を含む Platform Services Controller インスタンスの完全修飾ドメイン名または IP アドレスを入力し、[接続] をクリックします。

      **注：** 外部の Platform Services Controller を使用する場合、またはロード バランサの背後で複数の Platform Services Controller インスタンスを使用する場合は、vCenter Single Sign-On ドメインを共有するすべての Platform Services Controller の証明書を手動でインポートする必要があります。

      **注：** 構成済みの vRealize Orchestrator 環境と異なる vSphere Client を統合するには、vRealize Orchestrator に登録されているのと同じ Platform Services Controller を使用するように vSphere を構成する必要があります。高可用性 vRealize Orchestrator 環境では、vRealize Orchestrator ロード バランサ サーバの背後に Platform Services Controller (PCS) インスタンスをレプリケートする必要があります。
   c. 認証プロバイダの証明書情報を確認して、[証明書を承諾] をクリックします。
   d. vCenter Single Sign-On ドメインのローカル管理者アカウントの認証情報を入力します。[登録] をクリックします。

      デフォルトでは、このアカウントは **`administrator@vsphere.local`**、デフォルト テナント名は **`vsphere.local`** です。

e [管理グループ] テキスト ボックスに管理者グループの名前を入力し、[検索] をクリックします。

例：**vsphere.local\vcoadmins**

f 使用する管理グループを選択します。

g [変更を保存] をクリックします。

設定が正常に保存されたことを示すメッセージが表示されます。

**結果**

これで、vRealize Orchestrator サーバの構成が正常に完了しました。

**次のステップ**

- [CIS] が [ライセンス] ページで構成されたライセンス プロバイダであることを確認します。
- [設定を検証] ページで、ノードが正しく構成されていることを確認します。

**注：** 認証プロバイダの構成に従って、vRealize Orchestrator サーバは 2 分後に自動的に再起動します。認証直後に構成を確認すると、無効な構成ステータスが返される可能性があります。

# ライセンスを使用した vRealize Orchestrator 機能の有効化

特定の vRealize Orchestrator 機能へのアクセスは、vRealize Orchestrator 環境に適用されているライセンスに基づいて行われます。

認証後、認証プロバイダに基づいて、vRealize Orchestrator インスタンスにライセンスが割り当てられます。ライセンスは、次の vRealize Orchestrator 機能へのアクセスを制御します。

- Git 統合
- ロールの管理
- 複数言語のサポート（Python、Node.js、PowerShell）

vRealize Orchestrator サーバのライセンスは、コントロール センターの [ライセンス] ページから手動で変更できます。

**注：** ライセンス タイプに関係なく、同じライセンスを適用できる vRealize Orchestrator 環境の数に制限はありません。vRealize Automation ライセンスの場合、vRealize Automation 環境を展開および構成された状態にする必要はありません。

| 認証 | ライセンス | Git 統合 | ロールの管理 | 複数言語のサポート |
|---|---|---|---|---|
| vSphere | vSphere<br>vCloud Suite Standard | いいえ | いいえ | いいえ |
| vSphere | vRealize Automation<br>vRealize Suite Advanced または Enterprise<br>vCloud Suite Advanced または Enterprise | はい | はい | はい |
| vRealize Automation | vRealize Automation<br>vRealize Suite Advanced または Enterprise<br>vCloud Suite Advanced または Enterprise | はい | ロールは、vRealize Orchestrator の認証に使用される vRealize Automation インスタンスで管理されます。 | はい |

**注：** vRealize Suite Standard ライセンスは vRealize Automation を含んでいないため、vRealize Orchestrator 機能へのアクセスをサポートしません。

# vRealize Orchestrator データベース接続

vRealize Orchestrator サーバには、データを保存するためのデータベースが必要です。

展開された vRealize Orchestrator Appliance には、vRealize Orchestrator サーバでデータの格納に使用される事前構成済み PostgreSQL データベースが含まれています。

ユーザーは PostgreSQL データベースにアクセスできません。

# 証明書の管理

証明書は特定のサーバ用に発行され、そのサーバのパブリック キーに関する情報を含んでいるため、これを使用すると、vRealize Orchestrator で作成されたすべての要素に署名し、その正当性を保証できます。クライアントは、サーバから要素（通常はパッケージ）を受信すると、そのアイデンティティを検証してその署名を信頼するかどうかを判断します。

- vRealize Orchestrator 証明書の管理

  vRealize Orchestrator コントロール センターの [証明書] 画面で、または vRealize Orchestrator Client で *ssl_trust_manager* タグ付きワークフローを使用することにより、vRealize Orchestrator 証明書を管理できます。

# vRealize Orchestrator 証明書の管理

vRealize Orchestrator コントロール センターの [証明書] 画面で、または vRealize Orchestrator Client で *ssl_trust_manager* タグ付きワークフローを使用することにより、vRealize Orchestrator 証明書を管理できます。

## Orchestrator トラスト ストアへの証明書のインポート

vRealize Orchestrator コントロール センターは安全な接続を使用して vCenter Server、リレーショナル データベース管理システム (RDBMS)、LDAP、Single Sign-On など、各種サーバと通信します。必要な TLS 証明書は、URL からインポートするか、PEM でエンコードされたファイルからインポートすることができます。TLS 接続を使用してサーバ インスタンスに接続するたびに、[証明書] ページの [信頼された証明書] タブから対応する証明書をインポートし、対応する TLS 証明書をインポートする必要があります。

vRealize Orchestrator の TLS 証明書は、URL アドレスからロードするか、PEM でエンコードされたファイルからロードできます。

| オプション | 説明 |
| --- | --- |
| [URL またはプロキシ URL からインポート] | リモート サーバの URL：<br>**`https://your_server_IP_address`** または **`your_server_IP_address:port`** |
| [ファイルからインポート] | PEM でエンコードされた証明書ファイルのパス。<br>**注：** また、vRealize Orchestrator Client で [信頼されている証明書をファイルからインポート] ワークフローを実行することにより、信頼されている証明書をインポートすることもできます。このワークフローによってインポートするファイルは、DER でエンコードされている必要があります。 |

証明書のインポートの詳細については、コントロール センターを使用した信頼されている証明書のインポートを参照してください。

## パッケージ署名証明書

vRealize Orchestrator サーバからエクスポートされたパッケージはデジタル署名されています。パッケージの署名に使用する新しい証明書は、インポート、エクスポート、生成できます。パッケージ署名証明書のフォームはデジタル ID です。これは暗号化された通信や Orchestrator パッケージの署名に使用されます。

vRealize Orchestrator Appliance には、アプライアンスのネットワーク設定に基づいて自動的に生成されるパッケージ署名証明書が含まれています。アプライアンスのネットワーク設定が変更された場合は、新しいパッケージ署名証明書を手動で生成する必要があります。新しいパッケージ署名証明書を生成すると、その後でエクスポートされるすべてのパッケージは、新しい証明書を使用して署名されます。

## vRealize Orchestrator 用のカスタム TLS 証明書の生成

vRealize Orchestrator Appliance を使用して、環境の新しい TLS 証明書を生成したり、既存のカスタム証明書を設定したりできます。

vRealize Orchestrator Appliance には、アプライアンスのネットワーク設定に基づいて自動的に生成される Trusted Layer Security (TLS) が含まれています。アプライアンスのネットワーク設定が変更された場合は、新しい証明書を手動で生成する必要があります。証明書チェーンを作成すると、暗号化された通信を確保して、パッケージに署名を追加できます。ただし、受信者は、自己署名付きパッケージがサードパーティからではなく署名者のサーバから実際に発行されているのかどうかを確認することができません。サーバの ID を証明するには、認証局 (CA) が署名した証明書を使用します。

vRealize Orchestrator は環境ごとに一意のサーバ証明書を生成します。プライベート キーは、vRealize Orchestrator データベースの `vmo_keystore` テーブルに保存されます。

**注：** 既存のカスタム TLS 証明書を使用するよう vRealize Orchestrator Appliance を設定するには、vRealize Orchestrator 用のカスタム TLS 証明書の設定を参照してください。

#### 前提条件

vRealize Orchestrator Appliance の SSH アクセスが有効になっていることを確認します。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。

#### 手順

1. SSH を使用して、vRealize Orchestrator Appliance コマンド ラインに root としてログインします。
2. `vracli certificate ingress --generate auto --set stdin` コマンドを実行します。
3. vRealize Orchestrator Appliance にカスタム証明書を適用するには、展開スクリプトを実行します。
   a. `/opt/scripts/` ディレクトリに移動します。

   ```
   cd /opt/scripts/
   ```

   b. `./deploy.sh` スクリプトを実行します。

   **重要：** 展開スクリプトは、中断しないでください。スクリプトの実行が完了すると、次のメッセージが表示されます。

   ```
   Prelude が正常に展開されました。アクセスするには、your_orchestrator_address に移動してください
   ```

#### 次のステップ

新しい証明書チェーンが適用されていることを確認するには、`vracli certificate ingress --list` コマンドを実行します。

## vRealize Orchestrator 用のカスタム TLS 証明書の設定

vRealize Orchestrator Appliance のカスタム TLS 証明書を設定します。

vRealize Orchestrator Appliance には、アプライアンスのネットワーク設定に基づいて自動的に生成される Trusted Layer Security (TLS) が含まれています。

既存のカスタム TLS 証明書を使用するよう vRealize Orchestrator Appliance を設定できます。関連する PEM ファイルをローカル マシンから vRealize Orchestrator Appliance にインポートすることで、証明書を設定できます。また、vRealize Orchestrator Appliance に証明書チェーンを直接コピーして、カスタム TLS 証明書を設定することもできます。どちらの手順でも、vRealize Orchestrator 展開で新しい TLS 証明書を使用する前に、`/deploy.sh` スクリプトを実行する必要があります。

新しいカスタム TLS 証明書の生成の詳細については、vRealize Orchestrator 用のカスタム TLS 証明書の生成を参照してください。

**前提条件**

- vRealize Orchestrator Appliance の SSH アクセスが有効になっていることを確認します。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。
- TLS 証明書を含む PEM ファイルに、次のコンポーネントが設定した順番で含まれていることを確認します。

  a　証明書のプライベート キー。

  b　プライマリ証明書。

  c　認証局 (CA) の中間証明書または証明書（該当する場合）。

  d　ルート CA 証明書。

  たとえば、TLS 証明書を次のような構造にすることができます。

```
 -----BEGIN RSA PRIVATE KEY-----
<Private Key>
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
<Primary TLS certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Intermediate certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA certificate>
-----END CERTIFICATE-----
```

**手順**

1 PEM ファイルを vRealize Orchestrator Appliance にインポートして、証明書を設定します。

a SSH シェルから Secure Copy (SCP) コマンドを実行して、ローカル マシンから証明書 PEM をインポートします。

Linux の場合は、ターミナル SCP コマンドを使用できます。

```
scp ~/PEM_local_filepath/your_cert_file.PEM root@orchestrator_FQDN_or_IP:/
PEM_orchestrator_filepath/your_cert_file.PEM
```

Windows の場合は、PuTTY クライアント PSCP コマンドを使用できます。

```
pscp C:\PEM_local_filepath\your_cert_file.PEM root@<orchestrator_FQDN_or_IP>:/
PEM_orchestrator_filepath/your_cert_file.PEM
```

b SSH を使用して、vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

c `vracli certificate ingress --set` *your_cert_file.PEM* コマンドを実行します。

2 （オプション） 証明書チェーンをアプライアンスに直接コピーして、証明書を設定します。

a SSH を使用して、vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

b `vracli certificate ingress --set stdin` コマンドを実行します。

c 証明書チェーンをコピーし、貼り付けて、Ctrl + D キーを押します。

3 新しい TLS 証明書の適用を完了するには、展開スクリプトを実行します。

a `/opt/scripts/` ディレクトリに移動します。

```
cd /opt/scripts/
```

b `./deploy.sh` スクリプトを実行します。

**重要：** 展開スクリプトは、中断しないでください。スクリプトの実行が完了すると、次のメッセージが表示されます。

```
Prelude が正常に展開されました。アクセスするには、https://your_orchestrator_FQDN に移動してください
```

**結果**

vRealize Orchestrator Appliance のカスタム TLS 証明書が設定されました。

**次のステップ**

新しい証明書チェーンが適用されていることを確認するには、`vracli certificate ingress --list` コマンドを実行します。

### コントロール センターを使用した信頼されている証明書のインポート

他のサーバと安全に通信するには、vRealize Orchestrator サーバがその ID を確認できる必要があります。この目的で、リモート エンティティの TLS 証明書を、vRealize Orchestrator トラスト ストアにインポートすることが必要になる場合があります。特定の URL への接続を確立してトラスト ストアにインポートする方法、または PEM エンコード ファイルとしてトラスト ストアに直接インポートする方法のいずれかが使用できます。

**手順**

1 コントロール センターに **root** としてログインします。

2 [証明書] ページに移動します。

3 信頼されている証明書 を選択して、[インポート] をクリックします。

4 ファイルから証明書をインポートするには、[PEM エンコード ファイルからインポート] を選択します。

5 証明書ファイルを参照して、[インポート] をクリックします。

6 URL アドレスから証明書をインポートするには、[URL からインポート] からインポートを選択します。

7 証明書が保存されている URL アドレスを入力し、[インポート] をクリックします。

**結果**

リモート サーバ証明書は vRealize Orchestrator トラスト ストアに正常にインポートされました。

# vRealize Orchestrator プラグインの構成

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

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

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

## vRealize Orchestrator プラグインを管理

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

### プラグインのインストールまたはアップグレード

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

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

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

### プラグインのログ レベルの変更

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

### プラグインの無効化

プラグインの名前の横にある [プラグインを有効化] オプションの選択を解除すると、プラグインを無効にできます。

このアクションでは、プラグイン ファイルは削除されません。vRealize Orchestrator のプラグインのアンインストールについては、プラグインの削除を参照してください。

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

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

#### 前提条件

プラグインの *.dar* ファイルまたは *.vmoapp* ファイルをダウンロードします。

**注：** vRealize Orchestrator プラグインの推奨されるファイル形式は *.vmoapp* です。

#### 手順

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

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

#### 次のステップ

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

## プラグインの削除

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

**注：** vRealize Orchestrator 8.0 以降では、プラグイン パッケージを vRealize Orchestrator Client から手動で削除することはありません。

**手順**

1 コントロール センターに root としてログインします。

2 [プラグインの管理] を選択します。

3 削除するプラグインを検索して、削除アイコン ( ) をクリックします。

4 プラグインを削除することを確認し、[削除] をクリックします。

**結果**

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

# vRealize Orchestrator の高可用性

vRealize Orchestrator サービスの可用性を高めるには、共有データベースを使用してクラスタ内で複数の vRealize Orchestrator サーバ インスタンスを起動します。vRealize Orchestrator は、クラスタの一部として動作するように構成されるまで、単一のインスタンスとして動作します。

サーバおよびプラグインの構成が同じである複数の vRealize Orchestrator サーバ インスタンスは、クラスタ内で一緒に動作して 1 つのデータベースを共有できます。

すべての vRealize Orchestrator サーバ インスタンスはハートビートを交換して相互に通信します。個々のハートビートは、ノードによってクラスタの共有データベースに一定の間隔で書き込まれるタイムスタンプです。ネットワーク問題、データベース サーバが応答しない問題、またはオーバーロードの問題が発生すると、vRealize Orchestrator クラスタ ノードは応答を停止する場合があります。アクティブな vRealize Orchestrator サーバ インスタンスが、フェイルオーバーのタイムアウト期間内にハートビートを送信しなかった場合は、応答していないとみなされます。フェイルオーバーのタイムアウト期間は、ハートビート間隔をフェイルオーバー ハートビートの数で乗じた値と等しくなります。これによって信頼性の低いノードを特定できます。また、フェイルオーバーのタイムアウト期間は、使用可能なリソースや本番環境の負荷に応じてカスタマイズすることができます。

vRealize Orchestrator ノードは、データベース接続が失われると、データベース接続がリストアされるまでスタンバイ モードになります。クラスタ内の他のノードがアクティブな操作を制御し、中断されているすべてのワークフローを未完了の操作（スクリプト化可能なタスク、ワークフローの呼び出しなど）から再開します。

vRealize Orchestrator コントロール センターの [Orchestrator クラスタ管理] 画面から、vRealize Orchestrator クラスタの状態を監視できます。このページを使用して、クラスタのハートビート、フェイルオーバー ハートビートの数、アクティブな vRealize Orchestrator ノードの数を設定することもできます。

# vRealize Orchestrator スケーラビリティの上限

スケーラビリティの上限の表は、vRealize Orchestrator 8.x 展開での推奨される上限を示しています。

| コンポーネント | 拡張のターゲット | 詳細情報 |
|---|---|---|
| 仮想マシン | 35,000 | |
| vCenter Server 接続 | 10 | vCenter Server の設定を参照してください。 |
| クラスタ内のアクティブ ノード | 3 | vRealize Orchestrator クラスタの設定を参照してください。 |
| 同時実行中のワークフロー | ノードあたり 300 | ワークフローの実行のプロパティの設定を参照してください。 |
| キューに入れられた実行中のワークフロー | ノードあたり 10,000 | |
| 保持されるワークフローの実行 | ノードあたり 100 | |
| ログ イベントの有効期間（日） | 15 | |

# vRealize Orchestrator クラスタの設定

3 台のノードを展開し、これらをクラスタとして接続することで、新しい vRealize Orchestrator 環境が高可用性で実行されるように設定できます。

vRealize Orchestrator クラスタは、共通の PostgreSQL データベースを共有する 3 つの vRealize Orchestrator インスタンスで構成されています。設定された vRealize Orchestrator クラスタのデータベースは、非同期モードでのみ実行できます。

vRealize Orchestrator クラスタを作成するには、クラスタのプライマリ ノードとなる vRealize Orchestrator インスタンスを 1 つ選択する必要があります。プライマリ ノードを設定したら、セカンダリ ノードをプライマリ ノードに参加させます。

作成された vRealize Orchestrator クラスタには、自動フェイルオーバーが事前に設定されています。

**注：** 自動フェイルオーバーが失敗すると、データベースのデータが失われる可能性があります。

**前提条件**

- 3 つのスタンドアローン vRealize Orchestrator インスタンスをダウンロードして展開します。vRealize Orchestrator Appliance のダウンロードと展開を参照してください。

  **注：** クラスタ化された vRealize Orchestrator 環境の作成に使用できるノードの推奨数は 3 です。

- すべての vRealize Orchestrator ノードで SSH アクセスが有効になっていることを確認します。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。
- ロード バランサ サーバを設定します。VMware vRealize Orchestrator 8.x のロード バランシング ガイドを参照してください。

**手順**

1 プライマリ ノードを設定します。

a SSH を使用してプライマリ ノードの vRealize Orchestrator Appliance に root としてログインします。

b クラスタのロード バランサ サーバを設定するには、`vracli load-balancer set` *load_balancer_FQDN* コマンドを実行します。

c プライマリ ノードのコントロール センターにログインして、[ホストの設定] を選択します。

d [変更] をクリックして、接続されているロード バランサ サーバのホスト アドレスを設定します。

e 認証プロバイダを構成します。スタンドアローン vRealize Orchestrator サーバの構成を参照してください。

2 セカンダリ ノードをプライマリノードに参加させます。

a SSH を使用してセカンダリ ノードの vRealize Orchestrator Appliance に root としてログインします。

b セカンダリ ノードをプライマリ ノードに参加させるには、`vracli cluster join` *primary_node_hostname_or_IP* コマンドを実行します。

c プライマリ ノードの root パスワードを入力します。

d 他のセカンダリ ノードにもこの手順を繰り返します。

3 （オプション） プライマリ ノードでカスタム証明書が使用されている場合は、アプライアンスで証明書を設定するか、新しい証明書を生成する必要があります。vRealize Orchestrator 用のカスタム TLS 証明書の生成を参照してください。

**注：** 証明書チェーンを含むファイルは、PEM でエンコードされている必要があります。

4 クラスタの展開を完了します。

a SSH を使用してプライマリ ノードの vRealize Orchestrator Appliance に root としてログインします。

b すべてのノードが準備完了状態になっていることを確認するには、 `kubectl -n prelude get nodes` コマンドを実行します。

c `/opt/scripts/deploy.sh` スクリプトを実行し、展開が完了するまで待機します。

**結果**

vRealize Orchestrator クラスタが作成されました。クラスタを作成した後に vRealize Orchestrator 環境にアクセスするには、ロード バランサ サーバの FQDN アドレスからアクセスする必要があります。

**注：** クラスタのコントロール センターにアクセスするには、ロード バランサの root パスワードを使用する必要があるため、クラスタ ノードの設定は、異なる root パスワードが指定されている場合でも編集することができません。このノードの設定を編集するには、ロード バランサからノードを削除し、コントロール センターの設定を編集してから、ノードをロード バランサに再度追加します。

### 次のステップ

vRealize Orchestrator クラスタの状態を監視するには、コントロール センターにログインし、[Orchestrator クラスタ管理] ページを選択します。vRealize Orchestrator クラスタの監視を参照してください。

## vRealize Orchestrator クラスタ ノードの削除

vRealize Orchestrator を削除して、クラスタのキャパシティを削減することができます。

vRealize Orchestrator クラスタから削除したノードは機能しなくなります。このノードを再度使用する場合は、その vRealize Orchestrator Appliance を vCenter Server から削除して、再展開する必要があります。『vRealize Orchestrator Appliance のダウンロードと展開』を参照してください。

### 前提条件

vRealize Orchestrator クラスタを作成します。『vRealize Orchestrator クラスタの設定』を参照してください。

### 手順

1. 削除するノードの vRealize Orchestrator Appliance コマンド ラインに root としてログインします。
2. vRealize Orchestrator からノードを削除するには、`vracli cluster leave` コマンドを実行します。
3. 他のノードの 1 台の vRealize Orchestrator Appliance コマンドラインに root としてログインします。
4. `kubectl -n prelude get nodes` コマンドを実行して、削除されたノードがクラスタに含まれなくなったことを確認します。

## スタンドアローン vRealize Orchestrator 環境のスケールアウト

構成した vRealize Orchestrator 環境をスケール アウトすることで、可用性とスケーラビリティを向上させることができます。

### 前提条件

- vRealize Orchestrator インスタンスをダウンロード、展開、および構成します。『vRealize Orchestrator Appliance のダウンロードと展開と『スタンドアローン vRealize Orchestrator サーバの構成』を参照してください。
- 2 つの追加の vRealize Orchestrator インスタンスをダウンロードして展開します。vRealize Orchestrator Appliance のダウンロードと展開を参照してください。
- ロード バランサ サーバを設定します。VMware vRealize Orchestrator 8.x のロード バランシング ガイドを参照してください。

### 手順

1. プライマリ ノードを設定します。
   a. 設定した vRealize Orchestrator 環境のコントロール センターに root としてログインします。
   b. [認証プロバイダを設定] を選択して、認証プロバイダを登録解除します。
   c. [ホストの設定] を選択して、ロード バランサ サーバのホスト名を入力します。

d [認証プロバイダを設定] を選択して、認証プロバイダを再度登録します。

e 設定したインスタンスの vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

f vRealize Orchestrator インスタンスのすべてのサービスを停止するには、`/opt/scripts/deploy.sh --onlyClean` コマンドを実行します。

g ロード バランサを設定するには、`vracli load-balancer set`*load_balancer_FQDN* を実行します。

h （オプション）vRealize Orchestrator インスタンスでカスタム証明書が使用されている場合は、`vracli certificate ingress --set` *your_cert_file* コマンドを実行します。

**注：** 証明書チェーンを含むファイルは、PEM でエンコードされている必要があります。

2 設定したインスタンスにセカンダリ ノードを参加させます。

a セカンダリ ノードの vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

b 設定したインスタンスにセカンダリ ノードを参加させるには、`vracli cluster join`*primary_node_hostname_or_IP* コマンドを実行します。

c 他のセカンダリノードに手順を繰り返します。

3 スケール アウト プロセスを終了します。

a 設定したインスタンスの vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

b `/opt/scripts/deploy.sh` を実行し、スクリプトが終了するまで待機します。

**結果**

vRealize Orchestrator 環境がスケール アウトされました。

## vRealize Orchestrator クラスタの監視

vRealize Orchestrator コントロール センターを使用して、既存の vRealize Orchestrator クラスタを監視できます。

コントロール センターの [Orchestrator クラスタ管理] 画面から、クラスタに参加している vRealize Orchestrator インスタンスの構成の同期状態を監視できます。

| 設定同期状態 | 説明 |
| --- | --- |
| 実行時間 | vRealize Orchestrator サービスが使用可能になり、要求を受け入れることができます。 |
| スタンバイ | vRealize Orchestrator サービスは、次のことが原因で要求を処理できません。<br>■ ノードは、高可用性 (HA) クラスタの一部であり、プライマリ ノードに障害が発生するまでスタンバイ モードのままになります。<br>■ データベース、認証プロバイダ、および vRealize Orchestrator インスタンスのライセンスへの有効な接続など、構成の前提条件をサービスでは確認できません。 |
| サービスの健全性ステータスの取得に失敗しました。 | vRealize Orchestrator サーバ サービスが停止しているか、またはネットワークの問題が発生しているため、接続できません。 |
| 保留中の再起動 | コントロール センターによって構成変更が検出され、vRealize Orchestrator サーバが自動的に再起動します。 |

# カスタマ エクスペリエンス改善プログラムの構成

カスタマー エクスペリエンス向上プログラム (CEIP) への参加を選択すると、VMware 製品およびサービスの品質、信頼性、および機能の向上に役立つ匿名の情報が VMware に送信されます。

## VMware が受信する情報の種類

カスタマー エクスペリエンス向上プログラム (CEIP) では、VMware 製品とサービスの改善や問題点の修正に役立つ情報を VMware に提供します。

CEIP を通して収集されるデータおよび VMware のその使用目的に関する詳細は、Trust & Assurance センター (http://www.vmware.com/trustvmware/ceip.html) に記載されています。この製品の CEIP への参加または参加終了については、カスタマー エクスペリエンス向上プログラムへの参加または離脱を参照してください。

## カスタマー エクスペリエンス向上プログラムへの参加または離脱

vRealize Orchestrator Appliance コマンド ラインからカスタマー エクスペリエンス向上プログラムに参加します。

**手順**

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

2 カスタマー エクスペリエンス向上プログラムに参加するには、`vracli ceip on` コマンドを実行します。

3 カスタマー エクスペリエンス向上プログラムの情報を確認し、`vracli ceip on --acknowledge-ceip` コマンドを実行します。

4 vRealize Orchestrator サービスを再起動します。

   a サーバ サービスを再起動するには、`kubectl -n prelude exec -it` *your_vro_pod* `-c vco-server-app /bin/bash` コマンドを実行します。

   b サービスを停止するには、`kill 1` コマンドを実行します。

c コントロール センター サービスを再起動するには、`kubectl -n prelude exec -it` *your_vro_pod*`-c vco-controlcenter-app /bin/bash` コマンドを実行します。

d サービスを停止するには、`kill 1` コマンドを実行します。

5 カスタマー エクスペリエンス向上プログラムから離脱するには、`vracli ceip off` コマンドを実行します。

6 手順を繰り返してサービスを再起動します。

# vRealize Orchestrator API サービスの使用

6

コントロール センターを使用して vRealize Orchestrator を構成できるだけでなく、vRealize Orchestrator REST API、コントロール センターの REST API、またはコマンド ライン ユーティリティを使用して、アプライアンスに保存されている vRealize Orchestrator サーバの構成を変更することもできます。

構成プラグインは、デフォルトでは vRealize Orchestrator パッケージに含まれています。構成プラグイン ワークフローには、vRealize Orchestrator ワークフロー ライブラリまたは vRealize Orchestrator REST API からアクセスできます。これらのワークフローを使用すると、vRealize Orchestrator サーバの信頼された証明書およびキーストアの設定を変更できます。使用可能なすべての vRealize Orchestrator REST API サービス呼び出しについては、https://*your_orchestrator_FQDN*/vco/api/docs にある vRealize Orchestrator Server API のドキュメントを参照してください。


- REST API を使用した TLS 証明書とキーストアの管理

  コントロール センターを使用した TLS 証明書の管理のほか、構成プラグインまたは REST API を使用したワークフローの実行時の信頼された証明書とキーストアの管理も可能です。


## REST API を使用した TLS 証明書とキーストアの管理

コントロール センターを使用した TLS 証明書の管理のほか、構成プラグインまたは REST API を使用したワークフローの実行時の信頼された証明書とキーストアの管理も可能です。

構成プラグインには、TLS 証明書とキーストアをインポートおよび削除するためのワークフローが含まれています。これらのワークフローにアクセスするには、vRealize Orchestrator Client で [ライブラリ] - [ワークフロー] - [SSL トラスト マネージャ] および[ライブラリ] - [ワークフロー] - [キーストア] の順に選択します。これらのワークフローは、vRealize Orchestrator REST API を使用して実行することもできます。

コントロール センターの REST API を使用すると、vRealize Orchestrator サーバを設定するためのリソースにアクセスできます。サードパーティ システムでコントロール センターの REST API を使用して、vRealize Orchestrator の設定を自動化することができます。コントロール センターの REST API のルート エンドポイントは、https://*your_orchestrator_FQDN*/vco/api です。コントロール センターの REST API に実行できるすべてのサービス呼び出しについては、https://*your_orchestrator_FQDN*/vco-controlcenter/docs にある『vRealize Orchestrator コントロール センター API』ドキュメントを参照してください。

### REST API を使用した TLS 証明書の削除

TLS 証明書を削除するには、構成プラグインの「信頼されている証明書の削除」ワークフローを実行するか、REST API を使用します。

**手順**

1 「信頼されている証明書の削除」ワークフローのワークフロー サービスの URL で GET 要求を作成します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Delete trusted
certificate
```

2 「信頼されている証明書の削除」ワークフローの定義の URL で GET 要求を作成して、ワークフローの定義を取得します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/8a70a326-
ffd7-4fef-97e0-2002ac49f5bd
```

3 「信頼されている証明書の削除」ワークフローの実行オブジェクトが存在する URL で POST 要求を作成します。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/8a70a326-
ffd7-4fef-97e0-2002ac49f5bd/executions/
```

4 要求本文の実行コンテキスト要素内の「信頼されている証明書の削除」ワークフローの入力パラメータとして、削除する証明書の名前を指定します。

## REST API を使用した TLS 証明書のインポート

TLS 証明書をインポートするには、構成プラグインからワークフローを実行するか、REST API を使用します。

信頼された証明書をファイルまたは URL からインポートできます。コントロール センターを使用した信頼されている証明書のインポートを参照してください。

**手順**

1 ワークフロー サービスの URL で以下の GET 要求を作成します。

| オプション | 説明 |
| --- | --- |
| **信頼された証明書をファイルからインポート** | 信頼された証明書をファイルからインポートします。 |
| **信頼された証明書を URL からインポート** | 信頼された証明書を URL アドレスからインポートします。 |
| **プロキシ サーバを使用して信頼された証明書を URL からインポート** | プロキシ サーバを使用して、信頼された証明書を URL アドレスからインポートします。 |
| **証明書別名を持つ信頼された証明書を URL からインポート** | 証明書別名を持つ信頼された証明書を URL アドレスからインポートします。 |

信頼された証明書をファイルからインポートするには、以下の GET 要求を作成します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Import
                trusted certificate from a file
```

2 ワークフローの定義の URL で GET 要求を作成して、ワークフローの定義を取得します。

「信頼された証明書をファイルからインポート」ワークフローの定義を取得するには、以下の GET 要求を作成します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5
```

3 ワークフローの実行オブジェクトが存在する URL で POST 要求を作成します。

「信頼された証明書をファイルからインポート」ワークフローの場合は、以下の POST 要求を作成します。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5/executions
```

4 要求本文内の実行コンテキスト要素で、ワークフローの入力パラメータの値を指定します。

| パラメータ | 説明 |
| --- | --- |
| **cer** | TLS 証明書をインポートする元の CER ファイル。<br>このパラメータは、「信頼された証明書をファイルからインポート」ワークフローの場合に適用可能です。 |
| **url** | TLS 証明書をインポートする元の URL。非 HTTPS サービスの場合、サポートされる形式は *IP_address_or_DNS_name:port* です。<br>このパラメータは、「信頼された証明書を URL からインポート」ワークフローの場合に適用可能です。 |

## REST API を使用したキーストアの作成

キーストアを作成するには、構成プラグインの「キーストアの作成」ワークフローを実行するか、REST API を使用します。

**手順**

1 「キーストアの作成」ワークフローのワークフロー サービスの URL で以下の GET 要求を作成します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Create a keystore
```

2 「キーストアの作成」ワークフローの定義の URL で以下の GET 要求を作成して、ワークフローの定義を取得します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/
```

3 「キーストアの作成」ワークフローの実行オブジェクトが存在する URL で以下の POST 要求を作成します。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/executions/
```

4 要求本文の実行コンテキスト要素で「キーストアの作成」ワークフローの入力パラメータとして、作成するキーストアの名前を指定します。

## REST API を使用したキーストアの削除

キーストアを削除するには、構成プラグインの「キーストアの削除」ワークフローを実行するか、REST API を使用します。

**手順**

1 「キーストアの削除」ワークフローのワークフロー サービスの URL で以下の GET 要求を作成します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Delete a keystore
```

2 「キーストアの削除」ワークフローの定義の URL で以下の GET 要求を作成して、ワークフローの定義を取得します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/
7a3389eb-1fab-4d77-860b-81b66bb45b86/
```

3 「キーストアの削除」ワークフローの実行オブジェクトが存在する URL で以下の POST 要求を作成します。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/
7a3389eb-1fab-4d77-860b-81b66bb45b86/executions/
```

4 要求本文の実行コンテキスト要素で「キーストアの削除」ワークフローの入力パラメータとして、削除するキーストアの名前を指定します。

## REST API を使用したキーの追加

キーを追加するには、構成プラグインの「キーの追加」ワークフローを実行するか、REST API を使用します。

**手順**

1 「キーの追加」ワークフローのワークフロー サービスの URL で以下の GET 要求を作成します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Add key
```

2 「キーの追加」ワークフローの定義の URL で以下の GET 要求を作成して、定義を取得します。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/
```

3 「キーの追加」ワークフローの実行オブジェクトが存在する URL で以下の POST 要求を作成します。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/executions/
```

4 要求本文の実行コンテキスト要素で「キーの追加」ワークフローの入力パラメータとしてキーストア、キー エイリアス、PEM エンコード キー、証明書チェーン、およびキー パスワードを指定します。

# その他の設定オプション 7

コントロール センターを使用して、vRealize Orchestrator のデフォルトの動作を変更できます。

この章には、次のトピックが含まれています。


- 認証の再構成
- ワークフローの実行のプロパティの設定
- vRealize Orchestrator ログ ファイル
- OpenTracing と Wavefront 拡張機能の有効化
- vRealize Orchestrator の時刻同期の有効化
- vRealize Orchestrator の時刻同期を無効にする
- vRealize Orchestrator Kubernetes CIDR の構成
- vRealize Orchestrator の DNS 設定の更新


## 認証の再構成

コントロール センターの初期構成時に認証方法を設定すると、その後はいつでも、認証プロバイダや構成済みのパラメータを変更できます。

### 認証プロバイダの変更

認証モードまたは認証プロバイダの接続設定を変更するには、まず既存の認証プロバイダを登録解除する必要があります。

**手順**

1 コントロール センターに **root** としてログインします。

2 [認証プロバイダを設定] ページで、ホスト アドレスのテキスト ボックスの横にある [登録解除] ボタンをクリックして、使用中の認証プロバイダを登録解除します。

**結果**

これで、認証プロバイダを正常に登録解除できました。

#### 次のステップ

コントロール センターの認証を再設定します。『スタンドアローン vRealize Orchestrator サーバの構成』を参照してください。

## 認証パラメータの変更

vSphere をコントロール センターの認証プロバイダとして使用する場合は、vRealize Orchestrator 管理者グループのデフォルト テナントを変更できます。

#### 前提条件

vRealize Orchestrator 環境の認証プロバイダとして vSphere を設定します。『vSphere 認証を使用したスタンドアローン vRealize Orchestrator サーバの構成』を参照してください。

**注：** vRealize Automation の認証に、これらのパラメータは含まれません。

#### 手順

1. コントロール センターに root としてログインします。
2. [認証プロバイダを設定] を選択します。
3. [デフォルト テナント] テキスト ボックスの横にある [変更] ボタンをクリックします。
4. テナントの名前を置き換えます。
5. [管理グループ] テキスト ボックスの横にある [変更] ボタンをクリックします。

   **注：** 管理者グループを再設定しないと、このグループは空のままになりコントロール センターにアクセスできなくなります。
6. 管理者グループの名前を入力し、[検索] をクリックします。
7. 管理者グループを選択します。
8. 管理者グループを変更します。
9. 認証パラメータの編集を終了するには、[変更を保存] をクリックします。

# ワークフローの実行のプロパティの設定

デフォルトではノードあたり最大 300 個のワークフローを実行できます。実行中のワークフローが最大数に達した場合は、最大 10,000 個のワークフローをキューに入れることができます。

vRealize Orchestrator ノードで 300 個を超える同時ワークフローを実行する必要がある場合、保留中のワークフローの実行はキューに入れられます。アクティブなワークフローの実行が完了すると、キュー内にある次のワークフローの実行が開始されます。キュー内のワークフローが最大数に達すると、保留中のいずれかのワークフローの実行が開始されるまで、次のワークフローの実行は失敗します。

コントロール センターの [詳細オプション] ページで、ワークフローの実行プロパティを構成できます。

| オプション | 説明 |
| --- | --- |
| [セーフ モードを有効化] | セーフ モードが有効になっている場合、すべての実行中のワークフローはキャンセルされ、次回の vRealize Orchestrator ノード起動時に再開されません。 |
| [同時実行中のワークフローの数] | 同時に実行されるワークフローの数。デフォルトは、ノードあたり 300 個のワークフローです。 |
| [キュー内の実行中のワークフローの最大数] | vRealize Orchestrator サーバが使用不可になるまでに受け付け可能なワークフローの実行要求の数。デフォルトは、ノードあたり 10,000 個のワークフローです。 |
| [ワークフローごとに保持される実行の最大数] | ワークフローごとに履歴として保持される完了済みワークフロー実行の最大数。最大数を超えると最も古いワークフローの実行が削除されます。デフォルトは、ワークフローあたり 100 個の実行です。 |
| [ログ イベントの有効期間 (日)] | ログ イベントがデータベースから消去されるまでの日数。デフォルトは、15 日です。 |

# vRealize Orchestrator ログ ファイル

VMware テクニカル サポートでは、通常、サポート要求が送信されると診断情報を要求します。診断情報には、製品固有のログや、製品を実行しているホストの構成ファイルが含まれます。

vRealize Orchestrator Appliance ログは、`/data/vco/usr/lib/vco/app-server/logs/` ディレクトリに格納されます。アプライアンスのコマンド ラインにログインし、`vracli log-bundle` コマンドを実行することにより、vRealize Orchestrator Appliance 環境のログをエクスポートします。生成されたログ バンドルは、vRealize Orchestrator Appliance のルート フォルダに保存されます。

## ログのパーシステンス

ワークフロー、ポリシー、アクションなど、任意の種類の vRealize Orchestrator スクリプトで情報をログに記録できます。この情報にはタイプとレベルがあります。タイプは、パーシステントまたは非パーシステントのいずれかです。レベルは `DEBUG`、`INFO`、`WARN`、`ERROR`、`TRACE`、および `FATAL` のいずれかです。

**表 7-1. パーシステント ログおよび非パーシステント ログの作成**

| ログ レベル | パーシステント タイプ | 非パーシステント タイプ |
| --- | --- | --- |
| `DEBUG` | Server.debug("short text", "long text") | System.debug("text") |
| `INFO` | Server.log("short text", "long text") | System.log("text") |
| `WARN` | Server.warn("short text", "long text") | System.warn("text") |
| `ERROR` | Server.error("short text", "long text") | System.error("text") |

### パーシステント ログ

パーシステント ログ（サーバ ログ）は、過去のワークフロー実行ログを追跡し、vRealize Orchestrator データベースに保存されます。

### 非パーシステント ログ

非パーシステント ログ（システム ログ）を使用してスクリプトを作成した場合は、実行中のすべての vRealize Orchestrator アプリケーションに対し、このログについて vRealize Orchestrator サーバから通知がされますが、この情報はデータベースには保存されません。アプリケーションを再起動すると、ログ情報は失われます。非パーシステント ログは、デバッグの目的およびライブ情報のために使用されます。システム ログを表示するには、vRealize Orchestrator Client で完了したワークフローの実行を選択し、[ログ] タブを選択する必要があります。

## vRealize Orchestrator ログの設定

コントロール センターの [ログを設定] ページで、必要なサーバ ログとスクリプト ログのレベルを設定できますどちらかのログが 1 日に複数回生成されると、問題の原因を特定するのが困難になります。

サーバ ログとスクリプト ログのデフォルトのログ レベルは `INFO` です。ログ レベルを変更すると、サーバによってログに記録されるすべての新しいメッセージと、データベースへのアクティブな接続の数に影響が及びます。ログの詳細レベルは降順で低くなります。

**注意：** `DEBUG` または `ALL` のログ レベルは、問題をデバッグする場合にのみ設定してください。パフォーマンスを大幅に低下させる可能性があるため、これらのログ レベルは本番環境に設定しないでください。

### vRealize Orchestrator ログの生成

vRealize Orchestrator Appliance コマンド ラインに root としてログインし、`vracli log-bundle` コマンドを実行することにより、環境のログをエクスポートできます。生成されたログ バンドルは、アプライアンスのルート フォルダに保存されます。

。

**注：** クラスタ内に vRealize Orchestrator インスタンスが複数ある場合は、ログ バンドルにクラスタのすべての vRealize Orchestrator インスタンスのログが含まれます。

## vRealize Log Insight とのログ統合の設定

vRealize Log Insight サーバがログ情報を送信するように vRealize Orchestrator を設定することが可能です。

vRealize Orchestrator Appliance コマンド ラインを使用して、vRealize Log Insight サーバへのログ統合を設定できます。

**注：** リモート Syslog サーバとのログ統合を設定する方法については、vRealize Orchestrator での Syslog 統合の作成または上書きを参照してください。

#### 前提条件

- vRealize Log Insight サーバを設定します。「vRealize Log Insight のドキュメント」を参照してください。
- vRealize Log Insight バージョンが 4.7.1 以降であることを確認します。

#### 手順

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

2 vRealize Log Insight とのログ統合を設定するには、`vracli vrli set`*vRLI_FQDN* コマンドを実行します。

> **注：** vRealize Orchestrator インスタンスで自己署名証明書が使用されている場合は、オプションの `-k` または `--insecure` 引数を含めることによって SSL 認証を無効にできます。

**次のステップ**

vRealize Log Insight 設定オプションの詳細については、`vracli vrli -h` コマンドを実行してください。

## vRealize Orchestrator での Syslog 統合の作成または上書き

ログ情報を 1 台以上のリモート Syslog サーバに送信するように vRealize Orchestrator を設定できます。

`vracli remote-syslog set` コマンドは、Syslog 統合の作成、または既存の統合の上書きに使用されます。

vRealize Orchestrator リモート Syslog 統合では、次の 3 つの接続タイプがサポートされます。

- UDP 経由。
- TLS を使用しない TCP 経由。

> **注：** TLS を使用せずに Syslog 統合を作成するには、`--disable-ssl` フラグを `vracli remote-syslog set` コマンドに追加します。

- TLS を使用した TCP 経由。

vRealize Log Insight とのログ統合の設定については、vRealize Log Insight とのログ統合の設定を参照してください。

**前提条件**

1 台以上のリモート Syslog サーバを設定します。

**手順**

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

2 Syslog サーバへの統合を作成するには、`vracli remote-syslog set` コマンドを実行します。

```
vracli remote-syslog set -id name_of_integration protocol_type://
syslog_URL_or_FQDN:syslog_port
```

> **注：** `vracli remote-syslog set` コマンドにポートを入力しない場合、ポートの値はデフォルトの 514 になります。

> **注：** Syslog 設定に証明書を追加できます。証明書ファイルを追加するには、`--ca-file` フラグを使用します。証明書をプレーンテキストとして追加するには、`--ca-cert` フラグを使用します。

3 （オプション） 既存の Syslog 統合を上書きするには、`vracli remote-syslog set` を実行して、`-id` フラグの値を上書きする統合の名前に設定します。

> **注：** デフォルトでは、vRealize Orchestrator Appliance は、Syslog 統合への上書きの確認を要求します。確認の要求をスキップするには、`-f` または `--force` フラグを `vracli remote-syslog set` コマンドに追加します。

#### 次のステップ

アプライアンスで現在の Syslog 統合を確認するには、`vracli remote-syslog` コマンドを実行します。

### vRealize Orchestrator での Syslog 統合の削除

`vracli remote-syslog unset` コマンドを実行して、vRealize Orchestrator Appliance から Syslog 統合を削除できます。

#### 前提条件

vRealize Orchestrator Appliance で、1 つ以上の Syslog 統合を作成します。vRealize Orchestrator での Syslog 統合の作成または上書きを参照してください。

#### 手順

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

2 vRealize Orchestrator Appliance から Syslog 統合を削除します。

a 特定の Syslog 統合を削除するには、`vracli remote-syslog unset -id` *Integration_name* コマンドを実行します。

b vRealize Orchestrator Appliance のすべての Syslog 統合を削除するには、`-id` フラグを付けずに `vracli remote-syslog unset` コマンドを実行します。

> **注：** デフォルトでは、vRealize Orchestrator Appliance は、すべての Syslog 統合の削除の確認を要求します。確認の要求をスキップするには、`-f` または `--force` フラグを `vracli remote-syslog unset` コマンドに追加します。

## Kerberos デバッグ ログの有効化

プラグインで使用される Kerberos 構成ファイルを変更することで、vRealize Orchestrator プラグインの問題のトラブルシューティングを行うことができます。

Kerberos 構成ファイルは、vRealize Orchestrator Appliance の `/data/vco/usr/lib/vco/app-server/conf/` ディレクトリにあります。

#### 手順

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

2 `kubectl -n prelude edit deployment vco-app` コマンドを実行します。

3 展開ファイル内で `-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf'` という文字列を検索して、編集します。

```
-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf
-Dsun.security.krb5.debug=true'
```

4 変更内容を保存し、ファイル エディタを終了します。

5 `kubectl -n prelude get pods` コマンドを実行します。

すべてのポッドが実行されるまで待機します。

6 Kerberos デバッグ ログが有効になっていることを確認します。

```
kubectl -n prelude log {vco_app_name} -c vco-server-app | grep krb5
```

ログに同様なメッセージが含まれていることを確認します。

```
kubectl -n prelude log vco-app-5c965f9b9d-v8srd -c vco-server-app | grep krb5
12:23:05,417 INFO O11N:75 - Sysprop: java.security.krb5.conf = /usr/lib/vco/app-server/
conf/krb5.conf
12:23:05,421 INFO O11N:75 - Sysprop: sun.security.krb5.debug = true
2019-10-22 12:23:38.521+0000 [Thread-19] INFO {} [O11N] Sysprop: java.security.krb5.conf
= /usr/lib/vco/app-server/conf/krb5.conf
2019-10-22 12:23:38.525+0000 [Thread-19] INFO {} [O11N] Sysprop: sun.security.krb5.debug =
true
Java config name: /usr/lib/vco/app-server/conf/krb5.conf
EType: sun.security.krb5.internal.crypto.Aes256CtsHmacSha1EType
```

# OpenTracing と Wavefront 拡張機能の有効化

vRealize Orchestrator の Opentracing および Wavefront 拡張機能には、vRealize Orchestrator 環境に関するデータを収集するためのツールが提供されています。このデータを使用して、vRealize Orchestrator システムおよびワークフローのトラブルシューティングを行うことができます。

Opentracing および Wavefront 拡張機能を使用するように vRealize Orchestrator を設定するには、vRealize Orchestrator Appliance でそれらを有効にする必要があります。

**前提条件**

- vRealize Orchestrator Appliance SSH サービスが有効であることを確認します。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。
- 以前のバージョンの Opentracing または Wavefront 拡張機能を有効にしている場合は、現在のバージョンを有効にする前に削除する必要があります。たとえば、以前に Wavefront 拡張機能のバージョン 8.1.0 を有効にしていた場合は、`rm /data/vco/usr/lib/vco/app-server/extensions/wavefront-8.1.0.jar` コマンドを実行する必要があります。

**手順**

1 SSH を使用して vRealize Orchestrator Appliance に root としてログインします。

2 使用可能なすべての拡張機能を一覧表示するには、`ls /data/vco/usr/lib/vco/app-server/extensions/` コマンドを実行します。

3 次のコマンドを実行して Opentracing 拡張機能を有効にします。

```
mv /data/vco/usr/lib/vco/app-server/extensions/opentracing-8.5.0.jar.inactive /
data/vco/usr/lib/vco/app-server/extensions/opentracing-8.5.0.jar
```

4 次のコマンドを実行して Wavefront 拡張機能を有効にします。

```
mv /data/vco/usr/lib/vco/app-server/extensions/wavefront-8.5.0.jar.inactive /
data/vco/usr/lib/vco/app-server/extensions/wavefront-8.5.0.jar
```

5 コントロール センターにログインし、拡張機能が [拡張機能のプロパティ] ページに表示されていることを確認します。

#### 次のステップ

[拡張機能のプロパティ] ページで Opentracing および Wavefront と vRealize Orchestrator の統合を設定します。『OpenTracing 拡張機能の構成と『Wavefront 拡張機能の構成』を参照してください。

## OpenTracing 拡張機能の構成

Opentracing 拡張機能は、ワークフローの実行に関するデータを Jaeger サーバに送信します。データには、ワークフローのステータス、入出力パラメータ、ワークフローの実行を開始したユーザー、およびワークフロー ID データが含まれます。

#### 前提条件

- vRealize Orchestrator Appliance で Opentracing が有効になっていることを確認します。『OpenTracing と Wavefront 拡張機能の有効化』を参照してください。
- OpenTracing 拡張機能で使用する Jaeger サーバを展開します。詳細については、Jaeger スタート ガイド ドキュメントを参照してください。

#### 手順

1 コントロール センターに root としてログインします。

2 [拡張機能のプロパティ] ページを選択します。

3 Opentracing 拡張機能を選択します。

4 Jaeger サーバ ホストのアドレスおよびポートを入力します。

> **注：** サーバ アドレスを入力する前に、2 つのスラッシュ (//) を挿入します。

5 [保存] をクリックします。

#### 結果

vRealize Orchestrator の Opentracing 拡張機能が設定されました。

**次のステップ**

- OpenTracing 拡張機能によって収集されたデータを含む Jaeger ユーザー インターフェイスにアクセスするには、構成時に入力したホストのアドレスにアクセスします。
- [サービス] オプションで、[ワークフロー] を選択します。
- 表示するデータを指定するには、[タグ] オプションを使用します。たとえば、失敗したワークフローに関するデータを表示するには、**`status=failed`** と入力します。

## Wavefront 拡張機能の構成

Wavefront 拡張機能を使用して、vRealize Orchestrator システムとワークフローに関するメトリック データを収集します。

**前提条件**

1 vRealize Orchestrator Appliance で Wavefront が有効になっていることを確認します。『OpenTracing と Wavefront 拡張機能の有効化』を参照してください。

2 Wavefront 証明書をインポートします。

   a vRealize Orchestrator コントロール センターに root としてログインします。

   b [証明書] ページを選択します。

   c [インポート] ドロップダウン メニューをクリックして、[URL からのインポート] を選択します。

   d Wavefront URL を入力し、[インポート] をクリックします。

3 Wavefront プロキシを設定します。詳細については、Wavefront プロキシの構成を参照してください。

**手順**

1 vRealize Orchestrator コントロール センターに root としてログインします。

2 [拡張機能のプロパティ] ページを選択します。

3 Wavefront 拡張機能を選択します。

4 Wavefront のプロパティを設定します。

| オプション | 説明 |
| --- | --- |
| **プロキシ** | Wavefront プロキシのアドレス。 |
| **ホスト** | 任意。Wavefront ホストのアドレス。 |
| **トークン** | 任意。Wavefront API トークン。Wavefront API トークンの生成の詳細については、API トークンの生成を参照してください。 |
| **プリフィックス** | Wavefront に送信されたメトリックごとにプリフィックス ラベルを追加します。プリフィックス ラベルは、ドット記号で区切ります。 |

5 （オプション）[次の開始時にデフォルトのダッシュボードを送信する] を選択します。

6 [保存] をクリックします。

**結果**

vRealize Orchestrator の Wavefront 拡張機能が設定されました。

**次のステップ**

- Wavefront によって収集されたメトリックにアクセスするには、構成時に入力したアドレスでダッシュボードにアクセスします。
- vRealize Orchestrator 環境内の特定のイベントに関する通知を取得するには、Wavefront アラートを使用します。詳細については、Wavefront アラートのドキュメントを参照してください。

# vRealize Orchestrator の時刻同期の有効化

vRealize Orchestrator Appliance コマンド ラインを使用して、vRealize Orchestrator 環境で時刻同期を有効にすることができます。

NTP (Network Time Protocol) 通信プロトコルを使用して、スタンドアローンまたはクラスタ化された vRealize Orchestrator 環境に対して時刻同期を構成できます。vRealize Orchestrator は、相互に排他的な 2 つの NTP 構成をサポートしています。

| NTP 構成 | 説明 |
| --- | --- |
| ESXi | この構成は、vRealize Orchestrator Appliance をホストしている ESXi サーバが NTP サーバと同期している場合に使用できます。クラスタ化された環境を使用している場合は、すべての ESXi ホストを NTP サーバと同期する必要があります。ESXi 向けに NTP を構成する方法については、「Configuring Network Time Protocol (NTP) on an ESXi host using the vSphere Web Client」を参照してください。<br>**注：** NTP サーバと同期されていない ESXi ホストに vRealize Orchestrator 環境を移行すると、時刻のずれが生じる場合があります。 |
| systemd | この構成では systemd-timesyncd デーモンを使用して vRealize Orchestrator 環境の時刻を同期します。<br>**注：** デフォルトでは、systemd-timesyncd デーモンは有効になっていますが、NTP サーバなしで構成されています。動的 IP アドレス構成が使用されている vRealize Orchestrator Appliance では、DHCP プロトコルが受信する任意の NTP サーバを使用できます。 |

**手順**

1. vRealize Orchestrator Appliance コマンド ラインに root としてログインします。
2. ESXi で NTP モードを有効にします。
   a `vracli ntp esxi` コマンドを実行します。
   b （オプション） NTP 構成のステータスを確認するには、`vracli ntp status` コマンドを実行します。

3 systemd で NTP モードを有効にします。

a `vracli ntp systemd --set` *FQDN_or_IP_of_systemd_server* コマンドを実行します。

> **注：** 複数の systemd NTP サーバを追加するには、ネットワーク アドレスをカンマで区切ります。各ネットワーク アドレスは一重引用符で囲む必要があります。たとえば、`vracli ntp systemd --set`*'ntp_address_1'*、*'ntp_address_2'* のようになります。

b （オプション） NTP 構成のステータスを確認するには、`vracli ntp status` コマンドを実行します。

#### 結果

vRealize Orchestrator 環境の時刻同期を有効にしました。

#### 次のステップ

NTP サーバと vRealize Orchestrator 環境の間に 10 分を超える時間差があると、NTP 構成は失敗する可能性があります。この問題を解決するには、vRealize Orchestrator Appliance を再起動します。

## vRealize Orchestrator の時刻同期を無効にする

vRealize Orchestrator 環境で NTP (Network Time Protocol) 時刻同期を無効にするには、vRealize Orchestrator Appliance コマンド ラインを使用します。

`vracli ntp reset` コマンドを実行して、vRealize Orchestrator Appliance の NTP 構成をデフォルトの状態にリセットすることもできます。

#### 前提条件

ESXi または systemd との時刻同期が構成されていることを確認します。vRealize Orchestrator の時刻同期の有効化を参照してください。

#### 手順

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

2 ESXi または systemd との時刻同期を無効にするには、`vracli ntp disable` コマンドを実行します。

3 （オプション） NTP 構成のステータスを確認するには、`vracli ntp status` コマンドを実行します。

## vRealize Orchestrator Kubernetes CIDR の構成

展開後に Kubernetes Classless Inter-domain Routing (CIDR) サブネット マスクを変更できます。

vRealize Orchestrator Appliance は、Kubernetes クラスタを構成して実行します。このクラスタ内のポッドとサービスは、それぞれ内部クラスタ CIDR と内部サービス CIDR で表される個別の IPv4 サブネットに展開されます。OVF の展開時に設定されるサブネット マスクのデフォルト値は次のとおりです。

| Kubernetes network property | Default value | Property description |
|---|---|---|
| cluster-cidr | 10.244.0.0/22 | Kubernetes クラスタ内で実行されているポッドに使用される CIDR。 |
| service-cidr | 10.244.4.0/22 | Kubernetes クラスタ内の Kubernetes サービスに使用される CIDR。 |

デフォルトの CIDR ネットワーク アドレスは、使用している可能性のある外部プライベート ネットワークと競合する可能性があります。このようなシナリオでは、vRealize Orchestrator Appliance の展開中または展開後に、これらの CIDR 値の構成を変更できます。

**注：** アプライアンスの展開中に CIDR 構成を変更する方法については、vRealize Orchestrator Appliance のダウンロードと展開を参照してください。

**前提条件**

- CIDR アドレスの値が少なくとも 1024 台のホストをサポートしていることを確認します。
- 内部クラスタ CIDR と内部サービス CIDR は、同じサブネット値を共有できません。
- 一方のサブネットの CIDR 値には、もう一方のサブネットに追加する値を含めることはできません。

  **注：** たとえば、`cluster-cidr` の値を **`10.244.4.0/22 10.244.4.0/24`** にすることはできません。これには `service-cidr` プロパティのサブネット値も含まれるためです。各サブネット値は個別に追加する必要があります。

**手順**

1. vRealize Orchestrator Appliance に root としてログインします。
2. `vracli upgrade exec -y --prepare --profile k8s-subnets` コマンドを実行します。
3. 仮想マシン (VM) のスナップショットを作成して、vRealize Orchestrator 環境をバックアップします。仮想マシンのスナップショットの作成を参照してください。

   **注意：** vRealize Orchestrator 8.x は、現在メモリ スナップショットをサポートしていません。vRealize Orchestrator 環境のスナップショットを作成する前に、[仮想マシンのメモリのスナップショット作成] オプションが無効になっていることを確認します。
4. `vracli network k8s-subnets` コマンドを実行して、クラスタ CIDR およびサービス CIDR サブネットの値を変更します。

   ```
   vracli network k8s-subnets --cluster-cidr <CIDR_value> --service-cidr <CIDR_value>
   ```
5. CIDR 構成プロセスを完了するには、`vracli upgrade exec` コマンドを実行します。

# vRealize Orchestrator の DNS 設定の更新

管理者は、`vracli network dns` コマンドを使用して、vRealize Orchestrator 環境の DNS 設定を更新できます。

**前提条件**

vRealize Orchestrator Appliance SSH サービスが有効であることを確認します。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。

**手順**

1 SSH を使用して、vRealize Orchestrator Appliance コマンドラインに root としてログインします。

> **注：** クラスタ化環境では、クラスタ内の任意のノードのアプライアンスにログインします。

2 新しい DNS サーバを vRealize Orchestrator 環境に設定するには、`vracli network dns set` コマンドを実行します。

```
vracli network dns set --servers DNS1,DNS2
```

3 `vracli network dns status` コマンドを実行して、新しい DNS サーバがすべての vRealize Orchestrator ノードに適切に適用されていることを確認します。

4 環境内の vRealize Orchestrator サービスを停止するには、次の一連のコマンドを実行します。

```
/opt/scripts/svc-stop.sh
sleep 120
/opt/scripts/deploy.sh --onlyClean
```

5 vRealize Orchestrator ノードを再起動し、ノードが完全に起動するまで待機します。

6 SSH を使用して各 vRealize Orchestrator ノードのコマンドラインにログインし、新しい DNS サーバが `/etc/resolve.conf` ファイルにリストされていることを確認します。

7 vRealize Orchestrator サービスを開始するには、環境内のいずれかのノードで `/opt/scripts/deploy.sh` スクリプトを実行します。

**結果**

vRealize Orchestrator DNS 設定は、指定されたとおりに変更されます。

# 構成の使用事例とトラブルシューティング

8

構成の使用事例には、vRealize Orchestrator サーバの特定の設定要件を満たすために実行できるタスク フローと、問題を理解して解決するためのトラブルシューティングのトピックが含まれます。

この章には、次のトピックが含まれています。


- vRealize Orchestrator サーバのビルド番号の確認
- vSphere Web Client の vRealize Orchestrator プラグインの構成
- 実行中のワークフローのキャンセル
- vRealize Orchestrator Server のデバッグの有効化
- vRealize Orchestrator Appliance ディスクのサイズ変更
- vRealize Orchestrator サーバのヒープ メモリ サイズの調整方法
- Site Recovery Manager を使用した vRealize Orchestrator のディザスタ リカバリ


## vRealize Orchestrator サーバのビルド番号の確認

特定のシナリオでは、vRealize Orchestrator 展開環境のサーバのビルド番号を確認する必要があります。

vRealize Orchestrator サーバのビルド番号を確認するには、https://*your_orchestrator_FQDN*/vco/api/about に移動します。サーバのビルド番号は、`<ns2:build-number>` タグに表示されます。

サーバのビルド番号の確認は、VMware のサポートに提出したサポート リクエスト (SR) に追加情報を提供するなどの使用事例に役立ちます。また、サーバのビルド番号を確認して、最新バージョンの vRealize Orchestrator へのアップグレードが成功したか確認することもできます。

## vSphere Web Client の vRealize Orchestrator プラグインの構成

vSphere Web Client に対して vRealize Orchestrator プラグインを使用するには、vRealize Orchestrator を vCenter Server の拡張機能として登録する必要があります。

vRealize Orchestrator サーバを vCenter Single Sign-On に登録し、vCenter Server と連携するように構成したら、vRealize Orchestrator を vCenter Server の拡張機能として登録する必要があります。

**前提条件**

- SSH アクセスが vRealize Orchestrator Appliance に対して有効であることを確認します。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。
- 管理対象 vCenter Server の認証に使用されるものと同じ Platform Services Controller に、vSphere 認証を使用して vRealize Orchestrator を登録する必要があります。
- `vco-plugin.zip` を vRealize Orchestrator Appliance にコピーします。
  a VMware Technology Network から `vco-plugin.zip` ファイルをダウンロードします。
  b SSH クライアントを開きます。

  **注：** Linux または MacOS 環境では、ターミナル コマンドライン インターフェイスを使用できます。Windows 環境では、PuTTY クライアントを使用できます。

  c `vco-plugin.zip` ファイルをコピーするには、Secure Copy コマンドを実行します。

```
For Linux/MacOS: scp ~/<zip_download_dir>/vco-plugin.zip
root@<orchestrator_FQDN_or_IP>:/data/vco/usr/lib/vco/downloads/vco-plugin.zip
```

```
For Windows: pscp C:\<zip_download_dir>\vco-plugin.zip root@<orchestrator_FQDN_or_IP>:/
data/vco/usr/lib/vco/downloads/vco-plugin.zip
```

**手順**

1. vRealize Orchestrator Client にログインします。
2. [ライブラリ] - [ワークフロー] の順に移動します。
3. [vCenter Orchestrator を vCenter Server エクステンションとして登録] ワークフローを見つけ、[実行] をクリックします。
4. vRealize Orchestrator を登録する vCenter Server インスタンスを選択します。
5. https://*your_orchestrator_FQDN* を入力するか、要求を vRealize Orchestrator サーバ ノードにリダイレクトするロード バランサのサービス URL を入力します。
6. [実行] をクリックします。

# 実行中のワークフローのキャンセル

vRealize Orchestrator コントロール センターを使用すると、適切に終了しなかったワークフローをキャンセルできます。

**手順**

1. コントロール センターに **root** としてログインします。
2. [トラブルシューティング] をクリックします。

3 実行中のワークフローをキャンセルします。

| オプション | 説明 |
| --- | --- |
| **すべてのワークフローの実行をキャンセル** | ワークフロー ID を入力し、そのワークフローのすべてのトークンをキャンセルします。 |
| **ID を使用してワークフローの実行をキャンセル** | キャンセルするすべてのトークン ID を入力します。ID はコンマで区切ります。 |
| **すべての実行中のワークフローをキャンセル** | サーバ上で実行中のすべてのワークフローをキャンセルします。 |

**注：** 実行中のスレッドをすぐにキャンセルできる確実な方法がないため、ID を使用してワークフローをキャンセルする操作は正常に完了しない可能性があります。

**結果**

次回のサーバ起動時に、ワークフローはキャンセルされた状態に設定されます。

# vRealize Orchestrator Server のデバッグの有効化

プラグインの開発時に問題をデバッグするには、vRealize Orchestrator サーバをデバッグ モードで起動します。

**前提条件**

ローカル マシンに Kubernetes コマンドライン ツールをインストールして構成します。kubectl のインストールおよびセットアップを参照してください。

**手順**

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

2 `kubectl -n prelude edit deployment vco-app` コマンドを実行します。

3 `vco-server-app` コンテナにデバッグ環境変数を追加して、環境の `YAML` ファイルを編集します。変数は、`vco-server-app` コンテナの `env` セクションの下に追加する必要があります。

```
containers:
      - command:
      ...
        env:
        - name: DEBUG_PORT
          value: "your_desired_debug_port"
      ...
    name: vco-server-app
      ...
```

**注：** デバッグ環境変数を `env` セクションに追加する場合は、前の例で説明した YAML のインデント フォーマットに従う必要があります。

4 この変更を環境ファイルに保存します。

環境ファイルの編集が成功すると、`deployment.extensions/vco-app edited` メッセージが表示されます。

5 `vracli dev kubeconfig` コマンドを実行して、Kubernetes 構成ファイルを生成します。

Kubeconfig は開発者環境であるため、続行するか確認するプロンプトが表示されます。続行するには **yes**、停止するには **no** を入力します。

6 生成された構成ファイルの内容を、`apiVersion: v1` から `client-key-data` の内容まで含めてコピーします。

7 生成された Kubernetes 構成ファイルをローカル マシンに保存します。

8 vRealize Orchestrator Appliance からログアウトします。

9 ローカル マシンでデバッグ モードの設定を完了します。

a コマンドライン シェルを開きます。

b *KUBECONFIG* 環境変数を、保存した設定ファイルに割り当てます。

**注：** これは、Linux 環境の場合の例です。

```
export KUBECONFIG=/file/path/fileName
```

c サービスが実行中であることを確認するには、`kubectl cluster-info` コマンドを実行します。

d デバッグ モードの設定を完了するには、次の Kubernetes API 要求を実行します。

**注：** *localhost_debug_port* 変数の値は、統合開発環境 (IDE) のリモート デバッグ構成で設定されているポートです。*vro_debug_port* 変数の値は、この手順の手順 3 で生成されます。

```
kubectl port-forward pod/vco_app_pod_ID localhost_debug_port:vro_debug_port
```

**重要：** デバッグ ツールを設定する場合は、ポート転送コマンドを実行したローカル マシンの DNS および IP アドレスの設定を指定します。

#### 結果

vRealize Orchestrator Appliance にサーバ デバッグが設定されました。

# vRealize Orchestrator Appliance ディスクのサイズ変更

vRealize Orchestrator Appliance のディスク サイズを変更するには、vSphere で vRealize Orchestrator Appliance 仮想マシンのディスク サイズ設定を編集します。

#### 前提条件

vRealize Orchestrator Appliance SSH サービスが有効であることを確認します。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。

**手順**

1 vRealize Orchestrator Appliance で現在使用可能なディスク容量を確認します。

> **注：** vRealize Orchestrator Appliance ディスクには、少なくとも 20% の空きディスク容量が必要です。

a SSH を使用して、vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

b `vracli disk-mgr` コマンドを実行します。

2 vSphere で vRealize Orchestrator Appliance 仮想マシンのディスクのサイズを変更します。

a 管理者として vSphere Client にログインします。

b 仮想マシンを右クリックして [設定の編集] を選択します。

c [仮想ハードウェア] タブで、[ハード ディスク] を展開してディスクの設定を表示および変更し、[OK] をクリックします。

vSphere 仮想マシンのディスク サイズの変更の詳細については、『vSphere 仮想マシン管理』の「仮想ディスク構成の変更」を参照してください。

3 Photon OS の自動サイズ変更をトリガします。

a SSH を使用して、vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

b `vracli disk-mgr resize` コマンドを実行します。

> **注：** ディスク サイズの変更手順の進行状況は、`/var/log/vmware/prelude/disk_resize.log` で追跡できます。

vRealize Orchestrator Appliance ディスクのサイズが変更されました。

4 `disk-mgr` コマンドを実行して、ディスク サイズの変更手順が成功したことを確認します。

```
vracli disk-mgr
```

**次のステップ**

ディスク サイズの変更手順に関する問題をトラブルシューティングするには、KB 79925 を参照してください。

# vRealize Orchestrator サーバのヒープ メモリ サイズの調整方法

`values.yaml` ファイルを編集して、vRealize Orchestrator サーバのヒープ メモリ サイズを調整できます。

vRealize Orchestrator サーバのヒープ メモリ サイズは、オーケストレーション環境で変化するワークロードを管理できるように調整することが可能です。たとえば、複数の vCenter Server を管理する場合、vRealize Orchestrator 環境のヒープ メモリの増設が必要になることがあります。

**前提条件**

- vRealize Orchestrator Appliance への SSH アクセスを有効にします。vRealize Orchestrator Appliance への SSH アクセスの有効化または無効化を参照してください。

- vRealize Orchestrator が展開されている仮想マシンの RAM を、次に適切な増分まで拡張します。vSphere における仮想マシンの RAM の拡張の詳細については、『vSphere 仮想マシン管理』で「メモリ構成の変更」を参照してください。

**手順**

1. SSH を使用して、vRealize Orchestrator Appliance コマンド ラインに root としてログインします。
2. `/opt/charts/vco/` ディレクトリに移動します。
3. 任意のエディタを使用して、`values.yaml` ファイルを編集します。

   ```
   vi values.yaml
   ```

4. `serverMemoryLimit`、 `serverMemoryRequest`、および `serverJvmHeapMax` パラメータを変更します。
   a. `serverJvmHeapMax` パラメータを編集して、ヒープ メモリ値を設定します。
   b. `serverMemoryLimit` および `serverMemoryRequest` パラメータの値を更新します。

   **注意：** `serverMemoryLimit` パラメータ値は、`serverJvmHeapMax` パラメータで設定された値より 2 ギガバイト大きくなければなりません。`serverMemoryRequest` パラメータ値は、`serverJvmHeapMax` パラメータで設定された値より 1 ギガバイト大きくなければなりません。以下に、メモリ構成の例を示します。

   ```
   serverMemoryLimit: 8G
   serverMemoryRequest: 7G
   serverJvmHeapMax: 6G
   ```

   **注：** クラスタ化環境では、クラスタのすべてのノードで上記の手順を実行します。
5. `values.yaml` ファイルへの変更内容を保存し、`/opt/scripts` ディレクトリに移動します。
6. `deploy.sh` コマンドを実行します。

**結果**

これで vRealize Orchestrator サーバのヒープ メモリ サイズが変更されました。

# Site Recovery Manager を使用した vRealize Orchestrator のディザスタ リカバリ

vRealize Orchestrator を保護するように Site Recovery Manager を構成する必要があります。Site Recovery Manager の一般的な構成タスクを実行して保護します。

## 環境の準備

Site Recovery Manager の構成を開始する前に、以下の前提条件を満たしておく必要があります。

- 保護されたサイトとリカバリ サイトに vSphere 6.0 以降がインストールされていることを確認します。

- ご使用の Site Recovery Manager が 8.1 以降であることを確認します。
- vRealize Orchestrator が構成されていることを確認します。

## vSphere Replication のための仮想マシンの構成

Site Recovery Manager を使用するためには、vSphere Replication またはアレイ ベースのレプリケーションに対応するように仮想マシンを構成する必要があります。

必要な仮想マシンで vSphere Replication を有効にするには、次の手順を実行します。

**手順**

1. vSphere Web Client で vSphere Replication を有効にする仮想マシンを選択し、[アクション] - [vSphere Replication のすべてのアクション] - [レプリケーションの構成] の順にクリックします。
2. [レプリケーション タイプ] ウィンドウで [vCenter Server にレプリケート] を選択し、[次へ] をクリックします。
3. [ターゲット サイト] ウィンドウでリカバリ サイトの vCenter を選択し、[次へ] をクリックします。
4. [レプリケーション サーバ] ウィンドウで vSphere Replication サーバを選択し、[次へ] をクリックします。
5. [ターゲットの場所] ウィンドウで [編集] をクリックし、レプリケートしたファイルを保存する先のターゲット データストアを選択し、[次へ] をクリックします。
6. [レプリケーション オプション] ウィンドウでデフォルト設定をそのまま保持し、[次へ] をクリックします。
7. [リカバリ設定] ウィンドウで、[リカバリ ポイント目標 (RPO)] と [ポイント イン タイム インスタンス] の時間を入力し、[次へ] をクリックします。
8. [設定内容の確認] ウィンドウで設定を確認し、[完了] をクリックします。
9. vSphere Replication を有効にする必要があるすべての仮想マシンについて、上記の手順を繰り返します。

## 保護グループの作成

Site Recovery Manager が仮想マシンを保護できるようにするには、保護グループを作成します。

フォルダの保護グループを整備することができます。[保護グループ] タブには保護グループの名前が表示されますが、保護グループがどのフォルダに配置されたかは表示されません。異なるフォルダに同じ名前の保護グループが 2 つある場合、それらを区別するのが難しくなる可能性があります。そのため、保護グループ名は、すべてのフォルダで一意にするようにしてください。全ユーザーがすべてのフォルダを参照する権限を保有していない環境では、必ず保護グループの名前は一意になるようにしてください。フォルダに保護グループを置かないようにしてください。

保護グループを作成する場合、この処理が想定どおりに完了するまで待機します。Site Recovery Manager が保護グループを作成すること、およびグループ内の仮想マシンが正常に保護されていることを確認します。

**前提条件**

次のいずれかのタスクを実行したことを確認します。

- アレイ ベースのレプリケーションを構成するデータストアに仮想マシンを含めてあること。

- ストレージ ポリシー保護グループの前提条件の要件を満たし、『Site Recovery Manager 管理』ガイドの「ストレージ ポリシー保護グループの制限事項」を確認したこと。
- 仮想マシンで vSphere Replication を構成したこと
- 上記のいずれかの組み合わせ、またはすべてを実行したこと

**手順**

1. vSphere Client または vSphere Web Client で、[サイト リカバリ] - [サイト リカバリを開く] の順にクリックします。
2. [サイト リカバリ] ホーム タブで、サイト ペアを選択し、[詳細表示] をクリックします。
3. [保護グループ] タブを選択し、[新規] をクリックして保護グループを作成します。
4. [名前と方向] 画面で、保護グループの名前と説明を入力し、方向を選択して、[次へ] をクリックします。
5. [保護グループ タイプ] 画面で保護グループのタイプを選択し、[次へ] をクリックします。

| オプション | アクション |
|---|---|
| **アレイ ベースのレプリケーションの保護グループの作成** | [データストア グループ (アレイ ベースのレプリケーション)] を選択し、アレイのペアを選択します。 |
| **vSphere Replication 保護グループの作成** | [個別の仮想マシン (vSphere Replication)] を選択します。 |
| **ストレージ ポリシーの保護グループの作成** | [ストレージ ポリシー（アレイ ベースのレプリケーション)] を選択します。 |

6. 保護グループに追加するデータストア グループ、仮想マシン、またはストレージ ポリシーを選択します。

| オプション | アクション |
|---|---|
| **アレイ ベースのレプリケーションの保護グループ** | データストア グループを選択し、[次へ] をクリックします。<br>データストア グループを選択すると、そのグループに含まれる仮想マシンが [仮想マシン] テーブルに表示されます。 |
| **vSphere Replication 保護グループ** | リストから仮想マシンを選択し、[次へ] をクリックします。<br>vSphere Replication 用に構成し、マシンと保護グループにまだ含まれていない仮想マシンのみがリストに表示されます。 |
| **ストレージ ポリシーの保護グループ** | リストからストレージ ポリシーを選択し、[次へ] をクリックします。 |

7. [リカバリ プラン] 画面で、必要に応じて保護グループをリカバリ プランに追加できます。

| オプション | アクション |
|---|---|
| **既存のリカバリ プランに追加** | 保護グループを既存のリカバリ プランに追加します。 |
| **新規のリカバリ プランに追加** | 保護グループを新しいリカバリ プランに追加します。このオプションを選択する場合は、リカバリ プラン名を入力する必要があります。 |
| **今はリカバリ プランに追加しない。** | 保護グループをリカバリ プランに追加しない場合は、このオプションを選択します。 |

8 設定を確認し、[完了] をクリックします。

保護グループ作成の進捗状況は、[保護グループ] タブで監視できます。

- アレイ ベースのレプリケーションおよび vSphere Replication の保護グループの場合、保護された仮想マシンに Site Recovery Manager がインベントリ マッピングを正常に適用すると、保護グループの保護ステータスは *OK* になります。
- ストレージ ポリシーの保護グループの場合、Site Recovery Manager が、ストレージ ポリシーに関連付けられているすべての仮想マシンを正常に保護すると、保護グループの保護ステータスは *OK* になります。
- アレイ ベースのレプリケーションおよび vSphere Replication の保護グループの場合、インベントリ マッピングを設定しなかったか Site Recovery Manager がそれを適用できなかったときは、保護グループの保護ステータスは*未構成*になります。
- ストレージ ポリシーの保護グループの場合、Site Recovery Manager が、ストレージ ポリシーに関連付けられているすべての仮想マシンを保護できないと、保護グループの保護ステータスは*未構成*になります。

**次のステップ**

アレイ ベースのレプリケーションおよび vSphere Replication の保護グループの場合、保護グループの保護ステータスが*未構成*である場合は、仮想マシンにインベントリ マッピングを適用します。

- サイト全体にインベントリ マッピングを適用する場合、または設定済みのインベントリ マッピングが有効かどうかを確認する場合は、Site Recovery Manager 管理ガイドのインベントリ マッピングの構成を参照してください。これらのマッピングをすべての仮想マシンに適用するには、Site Recovery Manager 管理ガイドの保護グループのすべてのメンバーにインベントリ マッピングを適用を参照してください。
- インベントリ マッピングを保護グループ内の各仮想マシンに個別に適用するには、 Site Recovery Manager 管理ガイドの保護グループの個々の仮想マシンのインベントリ マッピングの構成を参照してください。

ストレージ ポリシーの保護グループの場合、保護グループの保護ステータスが*未構成*であるときは、ストレージ ポリシー保護グループの前提条件の要件を満たしていることを確認し、Site Recovery Manager 管理ガイドのストレージ ポリシー保護グループの制限事項を確認してください。

## リカバリ プランの作成

リカバリ プランを作成して、Site Recovery Manager による仮想マシンのリカバリ方法を確立します。

**手順**

1 vSphere Client または vSphere Web Client で、[サイト リカバリ] - [サイト リカバリを開く] の順に選択します。

2 [サイト リカバリ] ホーム タブで、サイト ペアを選択し、[詳細表示] をクリックします。

3 [リカバリ プラン] タブを選択し、[新規] をクリックしてリカバリ プランを作成します。

4 プランの名前、説明、方向を入力し、フォルダを選択して、[次へ] をクリックします。

5 メニューからグループ タイプを選択します。

| オプション | 説明 |
| --- | --- |
| **個別の仮想マシンまたはデータストア グループの保護グループ** | アレイ ベースのレプリケーションと vSphere Replication 保護グループが含まれているリカバリ プランを作成するには、このオプションを選択します。 |
| **ストレージ ポリシーの保護グループ** | ストレージ ポリシーの保護グループが含まれているリカバリ プランを作成するには、このオプションを選択します。<br>拡張ストレージを使用している場合は、このオプションを選択します。 |

6 このプランで復旧する保護グループを 1 つ以上選択し、[次へ] をクリックします。

7 [テスト ネットワーク] ドロップダウン メニューから、リカバリ テスト中に使用するネットワークを選択して、[次へ] をクリックします。

サイトレベルのマッピングがない場合は、デフォルトのオプション [サイトレベル マッピングの使用] により、隔離されたテスト ネットワークが作成されます。

8 概要情報を確認し、[終了] をクリックして、リカバリ プランを作成します。

## フォルダでのリカバリ プランの編成

さまざまなユーザーやグループからのリカバリ プランへのアクセスを制御するために、リカバリ プランをフォルダに編成できます。

リカバリ プランが多数ある場合は、リカバリ プランをフォルダに編成すると便利です。リカバリ プランへのアクセスを制限するには、リカバリ プランをフォルダに配置し、ユーザーまたはグループごとに異なる権限をフォルダに割り当てます。フォルダに権限を割り当てる方法については、Site Recovery Manager 管理ガイドの Site Recovery Manager のロールと権限の割り当てを参照してください。

**手順**

1 [サイト リカバリ] ホーム タブで、サイト ペアを選択し、[詳細表示] をクリックします。

2 [リカバリ プラン] タブをクリックし、左側のペインで [リカバリ プラン] を右クリックして、[新規フォルダ] をクリックします。

3 作成するフォルダの名前を入力し、[追加] をクリックします。

4 新規または既存のリカバリ プランをフォルダに追加します。

| オプション | 説明 |
| --- | --- |
| **新しいリカバリ プランの作成** | フォルダを右クリックし、[新しいリカバリ プラン] を選択します。 |
| **既存のリカバリ プランの追加** | インベントリ ツリーでリカバリ プランを右クリックし、[移動] をクリックします。目的のフォルダを選択し、[移動] をクリックします。 |

## リカバリ プランの編集

リカバリ プランを編集して、そのリカバリ プランの作成時に指定したプロパティを変更することができます。リカバリ プランは、保護サイトまたはリカバリ サイトから編集できます。

手順

1 vSphere Client または vSphere Web Client で、[サイト リカバリ] - [サイト リカバリを開く] の順にクリックします。

2 [サイト リカバリ] ホーム タブで、サイト ペアを選択し、[詳細表示] をクリックします。

3 [リカバリ プラン] タブをクリックし、リカバリ プランを右クリックして、[編集] をクリックします。

4 （オプション） プランの名前や説明を変更し、[次へ] をクリックします。

リカバリ プランの方向および場所は変更できません。

5 （オプション） プランに追加するかプランから削除する保護グループを 1 つ以上選択または選択解除し、[次へ] をクリックします。

6 （オプション） ドロップダウン メニューからリカバリ サイト上の別のテスト ネットワークを選択し、[次へ] をクリックします。

7 概要情報を確認して [終了] をクリックし、指定した変更をリカバリ プランに加えます。

プランの更新は [最近のタスク] ビューで監視できます。

# システム プロパティの設定

9

システム プロパティを設定することにより、Orchestrator のデフォルトの動作を変更することができます。

この章には、次のトピックが含まれています。

- ワークフローとアクションからサーバ ファイル システムにアクセスするための設定
- ワークフローとアクションからオペレーティング システム コマンドにアクセスするための設定
- JavaScript から Java クラスにアクセスするための設定
- カスタム タイムアウト プロパティの設定
- vRealize Orchestrator SQL プラグインの JDBC コネクタの追加

## ワークフローとアクションからサーバ ファイル システムにアクセスするための設定

vRealize Orchestrator では、ワークフローとアクションから特定のファイル システム ディレクトリへのアクセスが制限されます。`js-io-rights.conf` 構成ファイルを変更することにより、アクセス範囲を拡大して、サーバ ファイル システムの他の場所にアクセスすることができます。

### vRealize Orchestrator システムへの書き込みアクセスを許可する、js-io-rights.conf ファイル内のルール

`js-io-rights.conf` ファイルには、サーバ ファイル システム内の定義されたディレクトリに対する書き込みアクセスを許可するルールが含まれています。

#### js-io-rights.conf ファイルの必須の内容

`js-io-rights.conf` ファイルの各行には、次の情報を含める必要があります。

- 権限を許可するのか、それとも拒否するのかを示すプラス (`+`) またはマイナス (`-`) の記号
- 権限に対する読み取り (`r`)、書き込み (`w`)、および実行 (`x`) のレベル

- 権限を適用するパス

  **注：** `js-io-rights.conf` ファイルのルート フォルダは常に `/var/run/vco` です。vRealize Orchestrator Appliance ファイル システムでは、このフォルダは `/data/vco/var/run/vco` に配置されています。vRealize Orchestrator ファイル システムへのアクセス権を持つすべてのコンテンツは、このルート フォルダでマッピングされている必要があります。

### js-io-rights.conf ファイルのデフォルトの内容

Orchestrator Appliance の `js-io-rights.conf` 構成ファイルのデフォルトの内容は、次のとおりです。

```
-rwx /
+rwx /var/run/vco
+rx /etc/vco
-rwx /etc/vco/app-server/security/
+rx /var/log/vco/
```

デフォルトの `js-io-rights.conf` 構成ファイルの先頭の 2 行は、次のアクセス権を許可します。

**`-rwx /`**

ファイル システムへのすべてのアクセスは拒否されます。

**`+rwx /var/run/vco`**

`/var/run/vco` ディレクトリでの読み取り、書き込み、実行のアクセスは許可されます。

### js-io-rights.conf ファイル内のルール

vRealize Orchestrator は、`js-i/o-rights.conf` ファイルに記述されている順序でアクセス権を解決します。各行は前の行をオーバーライドできます。

**重要：** ファイル システムのすべての部分へのアクセスを許可するには、`js-i/o-rights.conf` ファイルで `+rwx /` を設定します。ただし、これを行うとセキュリティ リスクが増加します。

## ワークフローとアクションからサーバ ファイル システムにアクセスするための設定

ワークフローと vRealize Orchestrator API がアクセスできるサーバ ファイル システムの場所を変更するには、`js-io-rights.conf` 構成ファイルを変更します。`js-io-rights.conf` ファイルは、ワークフローが vRealize Orchestrator サーバ ファイル システムへのアクセスを試みる際に作成されます。

**手順**

1. vRealize Orchestrator Appliance コマンド ラインに root としてログインします。
2. `/data/vco/var/run/vco/` ディレクトリに移動します。
3. `js-io-rights.conf` 構成ファイルをテキスト エディタで開きます。

4 必要な行を `js-io-rights.conf` ファイルに追加して、ファイル システム領域へのアクセスを許可または拒否します。

たとえば、次の行により `/data/vco/var/run/vco/noexec` ディレクトリでの実行権限は拒否されます。

```
-x /data/vco/var/run/vco/noexec
```

`/data/vco/var/run/vco/noexec` は実行権限を保持しますが、`/data/vco/var/run/vco/noexec/bar` は保持しません。どちらのディレクトリも読み取りと書き込みは引き続き可能です。

**結果**

ワークフローと vRealize Orchestrator API のファイル システムへのアクセス権が変更されました。

# ワークフローとアクションからオペレーティング システム コマンドにアクセスするための設定

vRealize Orchestrator API では、vRealize Orchestrator サーバ ホストのオペレーティング システムでコマンドを実行するための `Command` というスクリプト クラスが提供されています。サーバ ホストに対する不正アクセスを防ぐため、デフォルトで、vRealize Orchestrator アプリケーションには `Command` クラスを実行するための権限が設定されていません。ホストのオペレーティング システム上でコマンドを実行するための権限が vRealize Orchestrator アプリケーションに必要な場合は、`Command` スクリプト クラスを有効にします。

`Command` クラスを使用する権限を付与するには、vRealize Orchestrator 構成のシステム プロパティを設定します。

**手順**

1 コントロール センターに **root** としてログインします。

2 [システム プロパティ] をクリックします。

3 [新規] をクリックします。

4 [キー] テキスト ボックスに **com.vmware.js.allow-local-process** と入力します。

5 [値] テキスト ボックスに **true** と入力します。

6 [説明] テキスト ボックスにシステム プロパティの説明を入力します。

7 [追加] をクリックします。

8 ポップアップ メニューから [変更を保存] をクリックします。

正常に保存されたことを示すメッセージが表示されます。

9 vRealize Orchestrator サーバが再起動するまで待ちます。

**結果**

vRealize Orchestrator サーバ ホストのオペレーティング システムでローカル コマンドを実行する権限が vRealize Orchestrator アプリケーションに付与されました。

**注：** `com.vmware.js.allow-local-process` システム プロパティを `true` に設定すると、`Command` スクリプト クラスでファイル システムのすべての場所での書き込みが許可されます。このプロパティは、`js-io-rights.conf` ファイルで `Command` スクリプト クラスにのみ設定しているファイル システム アクセス権限より優先されます。`Command` を除くすべてのスクリプト クラスでは、`js-io-rights.conf` ファイルで設定しているファイル システム アクセス権限は引き続き適用されます。

# JavaScript から Java クラスにアクセスするための設定

デフォルトでは、vRealize Orchestrator の JavaScript からアクセスできる Java クラスのセットは制限されています。JavaScript からさまざまな Java クラスにアクセスするには、vRealize Orchestrator システム プロパティを設定する必要があります。

JavaScript エンジンに Java 仮想マシン（JVM）へのフル アクセスを許可すると、セキュリティ上の問題が発生することがあります。不正な形式のスクリプトや悪意のあるスクリプトが、vRealize Orchestrator サーバを実行するユーザーがアクセスできるすべてのシステム コンポーネントへのアクセス権を取得する可能性があります。そのため、デフォルトでは、vRealize Orchestrator JavaScript エンジンは `java.util.*` パッケージ内のクラスにのみアクセスできます。

JavaScript が `java.util.*` パッケージの外部にあるクラスにアクセスする必要がある場合は、JavaScript からアクセスできるようにする Java パッケージのリストを構成ファイル内に記載できます。その後、このファイルを指すように `com.vmware.scripting.rhino-class-shutter-file` システム プロパティを設定します。

**手順**

1 JavaScript からアクセスできるようにする Java パッケージのリストを格納するためのテキスト構成ファイルを作成します。

たとえば、JavaScript から `java.net` パッケージ内のすべてのクラスと `java.lang.Object` クラスにアクセスできるようにするには、このファイルに次の内容を追加します。

```
java.net.*
java.lang.Object
```

2 構成ファイルの名前を入力します。

3 構成ファイルを `/data/vco/usr/lib/vco` のサブディレクトリに保存します。

**注：** 構成ファイルは他のディレクトリに保存することはできません。

4 コントロール センターに **root** としてログインします。

5 [システム プロパティ] をクリックします。

6 [新規] をクリックします。

7 [キー] テキスト ボックスに **com.vmware.scripting.rhino-class-shutter-file** と入力します。

8 [値] テキスト ボックスに、`vco/usr/lib/vco/`*your_configuration_file_subdirectory* と入力します。

9 [説明] テキスト ボックスにシステム プロパティの説明を入力します。

10 [追加] をクリックします。

11 ポップアップ メニューから [変更を保存] をクリックします。

正常に保存されたことを示すメッセージが表示されます。

12 vRealize Orchestrator サーバが再起動するまで待ちます。

**結果**

JavaScript エンジンが指定された Java クラスにアクセスできるようになりました。

# カスタム タイムアウト プロパティの設定

vCenter Server が過負荷になっている場合は、vRealize Orchestrator サーバに応答を返すのに、デフォルトで設定されている 20,000 ミリ秒よりも長い時間がかかります。この状況を回避するには、vRealize Orchestrator 構成ファイルを変更して、デフォルトのタイムアウト期間を延長する必要があります。

特定の操作が完了する前にデフォルトのタイムアウト期間が経過した場合は、vRealize Orchestrator サーバ ログにエラーが記録されます。

```
Operation 'getPropertyContent' total time : '5742228' for 1823 calls, mean
time : '3149.0', min time : '0', max time : '32313' Timeout, unable to get
property 'info' com.vmware.vmo.plugin.vi4.model.TimeoutException
```

**手順**

1 コントロール センターに **root** としてログインします。

2 [システム プロパティ] をクリックします。

3 [新規] をクリックします。

4 [キー] テキスト ボックスに、**com.vmware.vmo.plugin.vi4.waitUpdatesTimeout** と入力します。

5 [値] テキスト ボックスに、新しいタイムアウト期間をミリ秒単位で入力します。

6 （オプション） [説明] テキスト ボックスにシステム プロパティの説明を入力します。

7 [追加] をクリックします。

8 ポップアップ メニューから [変更を保存] をクリックします。

正常に保存されたことを示すメッセージが表示されます。

9 Orchestrator サーバを再起動します。

**結果**

設定した値により、デフォルトのタイムアウト設定である 20,000 ミリ秒はオーバーライドされます。

# vRealize Orchestrator SQL プラグインの JDBC コネクタの追加

この例では、vRealize Orchestrator SQL プラグイン用の MySQL コネクタを追加する方法を示します。

**手順**

1 MySQL connector.jar ファイルを vRealize Orchestrator Appliance に追加します。

a SSH を使用して、vRealize Orchestrator Appliance コマンド ラインに root としてログインします。

b `/data/vco/var/run/vco` ディレクトリに移動します。

```
cd /data/vco/var/run/vco
```

c `plugins/SQL/lib/` ディレクトリを作成します。

```
mkdir -p plugins/SQL/lib/
```

d SCP (Secure Copy) コマンドを使用して、MySQL connector.jar ファイルをローカル マシンから `/data/vco/var/run/vco/plugins/SQL/lib/` ディレクトリにコピーします。

```
scp ~/local_machine_dir/your_mysql_connector.jar root@orchestrator_FQDN_or_IP:/
data/vco/var/run/vco/plugins/SQL/lib/
```

**注：** また、PSCP などの別の方法を使用して、connector.jar ファイルを vRealize Orchestrator Appliance にコピーすることもできます。

2 新しい MySQL プロパティをコントロール センターに追加します。

a コントロール センターに root としてログインします。

b [システム プロパティ] を選択します。

c [新規] をクリックします。

d [キー] で、**o11n.plugin.SQL.classpath** と入力します。

e [値] で、**/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar**. と入力します。

**注：** 値テキスト ボックスに複数の JDBC コネクタを含めることができます。各 JDBC コネクタはセミコロン（「;」）で区切られています。例：

```
/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar;/var/run/vco/plugins/SQL/lib/
your_mssql_connector.jar;/var/run/vco/plugins/SQL/lib/your_other_connector.jar
```

f （オプション） MySQL システム プロパティの説明を入力します。

g [追加] をクリックし、vRealize Orchestrator サーバが再起動するまで待機します。

**注：** JDBC connector.jar ファイルを別のディレクトリに保存したり、別の値を `o11n.plugin.SQL.classpath` プロパティに設定したりしないでください。これを行うと、vRealize Orchestrator 環境で JDBC コネクタが使用できなくなります。

# 次の手順

10

vRealize Orchestrator のインストールと構成の完了後、vRealize Orchestrator を使用することで、仮想環境の管理に関連した、繰り返しの頻度が高いプロセスを自動化することができます。

- vRealize Orchestrator Client にログインし、vCenter Server インベントリ オブジェクト、または vRealize Orchestrator がプラグイン経由でアクセスするその他のオブジェクトに関するワークフローを実行して、スケジューリングします。「VMware vRealize Orchestrator クライアントの使用」を参照してください。
- 標準 vRealize Orchestrator ワークフローを複製および変更し、必要なアクションとワークフローを記述して vCenter Server の操作を自動化します。
- vRealize Orchestrator プラットフォームの機能を拡張するには、プラグインを開発します。
- リモートの Git リポジトリの統合を使用して、複数の vRealize Orchestrator インスタンス間で vRealize Orchestrator インベントリを管理します。『VMware vRealize Orchestrator クライアントの使用』を参照してください。
- vSphere Web Client を使用して、vSphere インベントリ オブジェクトに関するワークフローを実行します。