

# vRealize Automation から 7.4 への移行

2018 年 10 月 5 日

vRealize Automation 7.4



vmware®

VMware Web サイトで最新の技術ドキュメントをご確認いただけます。

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

VMware の Web サイトでは、最新の製品アップデートを提供しています。

本書に関するご意見、ご要望をお寄せください。フィードバック送信先：

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

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

ヴィ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](http://www.vmware.com/jp)

Copyright © 2008–2018 VMware, Inc. All rights reserved. [著作権および商標](#).

# 内容

更新情報 5

## 1 vRealize Automation の移行 6

## 2 vRealize Automation 環境のユーザー インターフェイス 8

## 3 移行の前提条件 12

最小環境に移行するための前提条件 12

高可用性環境に移行するための前提条件 15

## 4 移行前のタスク 19

vRealize Automation 6.2.x から 7.x への移行で導入される変更の確認 19

ソフトウェア エージェント パッチの適用 20

vSphere エージェントでの DoDeletes 設定の False への変更 20

vRealize Automation 6.x 移行前環境のテンプレートの確認 21

vRealize Automation 仮想マシンの移行準備 22

移行に必要な情報の収集 22

ソース vRealize Automation 環境からの暗号化キーの取得 23

ソース vRealize Automation 6.2.x 環境からテナントと IaaS 管理者のリスト作成 24

vRealize Automation の移行前の環境からターゲット環境への各テナントの追加 24

追加された各テナントの管理者の作成 26

最小環境への移行前に Active Directory リンクのユーザーとグループを同期する 27

高可用性環境への移行前に Active Directory リンクのユーザーとグループを同期する 28

ソース vRealize Automation 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行 30

ソース vRealize Automation IaaS Microsoft SQL データベースの手動によるクローン作成 31

ターゲット vRealize Automation 環境のスナップショット作成 31

## 5 移行手順 32

vRealize Automation 7.4 最小環境への vRealize Automation ソース データの移行 32

vRealize Automation 7.4 高可用性環境への vRealize Automation ソース データの移行 34

## 6 移行後のタスク 37

ソース vRealize Automation 6.2.x 環境からのテナントと IaaS 管理者の追加 38

接続テストの実行と移行後のエンドポイントの確認 38

ターゲット vRealize Automation 7.4 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行 39

移行後のロード バランサを高可用性環境に再構成する 40

外部 Orchestrator サーバから vRealize Automation 7.4 への移行 40

Migration Scenarios 41

Orchestrator 構成を Windows から仮想アプライアンスに移行 41

Windows 上の外部 vRealize Orchestrator 6.x から vRealize Automation 7.4 への移行 44

外部 vRealize Orchestrator 6.x 仮想アプライアンスから vRealize Automation 7.4 への移行 46

外部 vRealize Orchestrator 7.x から vRealize Automation 7.4 への移行 50

組み込み vRealize Orchestrator サーバの構成 53

vRealize Automation 証明書を信頼するように組み込みの vRealize Orchestrator を更新する 54

外部 Orchestrator および組み込み Orchestrator のコントロール センターの違い 55

ターゲットの vRealize Orchestrator の vRealize Automation エンドポイントの再構成 55

ターゲットの vRealize Orchestrator の vRealize Automation インフラストラクチャ エンドポイントの再構成 56

vRealize Orchestrator カスタマイズのインストール 57

ターゲット vRealize Automation に組み込まれた vRealize Orchestrator インフラストラクチャ エンドポイントの再構成 58

ターゲット vRealize Automation 環境の Azure エンドポイントの再構成 58

vRealize Automation 6.2.x Automation Application Services から 7.4 への移行 59

元のターゲット vRealize Automation IaaS Microsoft SQL データベースの削除 59

移行後にデータセンターの [場所] メニューの内容を更新する 60

ソフトウェア エージェントの TLS 1.2 へのアップグレード 60

ソース環境の仮想マシン テンプレートの更新 60

ソフトウェア エージェントのアップグレードが必要な仮想マシンの特定 61

vSphere 上のソフトウェア エージェントのアップグレード 63

Amazon Web Service または Azure 上のソフトウェア エージェントのアップグレード 65

移行後にプロパティ ディクショナリ設定を変更する 68

ターゲット vRealize Automation 7.4 環境の検証 68

## 7 移行に関するトラブルシューティング 70

PostgreSQL バージョンが原因のエラー 70

移行中に作成された展開が一部の仮想マシンに存在しない 70

移行ログの場所 71

移行後にサービス カタログに表示されるカタログ アイテムを申請できない 71

vRealize Automation の無効になった [データ収集] のラジオ ボタン 72

ソフトウェア エージェントのアップグレードのトラブルシューティング 73

## 更新情報

この『vRealize Automation から 7.4 への移行』は、製品のリリースごと、または必要に応じて更新されます。

『vRealize Automation から 7.4 への移行』の更新履歴については、次の表をご確認ください。

リビジョン	説明
2018 年 10 月 05 日	マイナー更新。
2018 年 5 月 3 日	<a href="#">「外部 Orchestrator サーバから vRealize Automation 7.4 への移行」</a> にガイダンスを含むセクションが追加されました。
2018 年 4 月 12 日	初期リリース。

# vRealize Automation の移行

移行を使用すると、既存の vRealize Automation 環境のサイドバイサイド アップグレードを実行できます。

移行は、現在の vRealize Automation の移行前の環境から vRealize Automation の最新バージョンのターゲット展開に、テナントと ID ストアを除くすべてのデータを移動します。さらに、移行によりすべてのデータが組み込みの vRealize Orchestrator 7.x から移行先の環境に移動します。

移行では、データを収集してそのデータをターゲット環境に安全にコピーするために必要な時間の間、vRealize Automation サービスを停止する以外に、移行前の環境を変更しません。ソース vRealize Automation データベースのサイズによって、移行は数分から数時間かかることがあります。

移行前の環境を最小インストールまたは高可用性展開に移行できます。

移行後にターゲット環境を本番環境にする場合は、移行前の環境を再開しないでください。移行後の移行前の環境への変更は、ターゲット環境と同期されません。

移行前の環境が vCloud Air または vCloud Director に統合されている場合、または物理的なエンドポイントがある場合は、移行を使用してアップグレードを実行する必要があります。移行により、これらのエンドポイントと、エンドポイントに関連付けられたすべてのものがターゲット環境から削除されます。また、移行により、ターゲット環境から 6.x VMware vRealize Application Services 統合も削除されます。

---

**注:** 移行前に、vRealize Automation 仮想マシンを準備する追加のタスクを完了する必要があります。移行前に、ナレッジベースの記事 [KB51531](#) を確認してください。

---

vRealize Automation 6.2.x から最新バージョンに移行する場合は、次の問題が発生する可能性があります。

問題	解決方法
<p>vRealize Automation 6.2.x を最新バージョンに移行した後、以下のプロパティ定義を使用したカタログ アイテムはサービス カタログに表示されますが、申請できません。</p> <ul style="list-style-type: none"> <li>■ コントロール タイプ：チェック ボックスまたはリンク。</li> <li>■ 属性：関係、正規表現、またはプロパティのレイアウト。</li> </ul> <p>vRealize Automation 7.x では、プロパティ定義でこれらの要素が使用されなくなりました。</p>	<p>プロパティ定義を再作成するか、組み込みのコントロール タイプまたは属性ではなく vRealize Orchestrator スクリプト アクションを使用するようにプロパティ定義を設定する必要があります。詳細については、<a href="#">「移行後にサービス カタログに表示されるカタログ アイテムを申請できない」</a> を参照してください。</p>
<p>vRealize Automation 6.2.x のドロップ ダウン メニューで親子関係の定義に使用されていた正規表現は、バージョン 7.x ではサポートされていません。バージョン 6.2.x では、特定の親メニュー項目でのみ使用可能な 1 つ以上の子メニュー項目を定義するのに正規表現を使用できます。これらの子メニュー項目は、親メニュー項目を選択したときのみ表示されます。</p> <p>バージョン 7.x に移行すると、親ドロップ ダウン メニューでの選択とは無関係に、子ドロップダウン メニューにはすべての利用可能なメニュー項目が表示されます。以前に定義された動的な値が機能していないことを示すために、子ドロップダウン メニューの最初のメニュー項目には以下のメッセージが表示されます：警告: vRO ワークフローを使用して動的な値を定義してください。</p>	<p>移行後、以前の動的な値を復元するにはプロパティ定義を再作成する必要があります。親ドロップダウン メニューと子ドロップダウン メニューの間で親子関係を作成する方法の詳細については、<a href="#">How to use dynamic property definitions in vRA 7.2</a> を参照してください。</p>

# vRealize Automation 環境のユーザー インターフェイス

## 2

vRealize Automation 環境は、複数のインターフェイスで使用および管理します。

### ユーザー インターフェイス

これらの表は、vRealize Automation 環境を管理するために使用するインターフェイスを示しています。

表 2-1. vRealize Automation 管理コンソール

目的	アクセス	必要な認証情報
以下のシステム管理者のタスクには、vRealize Automation コンソールを使用します。 <ul style="list-style-type: none"><li>■ テナントを追加します。</li><li>■ vRealize Automation ユーザー インターフェイスをカスタマイズします。</li><li>■ メール サーバを構成します。</li><li>■ イベント ログを表示します。</li><li>■ vRealize Orchestrator を構成します。</li></ul>	<ol style="list-style-type: none"><li>1 ブラウザを起動し、仮想アプライアンスの完全修飾ドメイン名を使用して、vRealize Automation アプライアンスのスプラッシュ ページを開きます。  https://&lt;vra-virtual-hostname.domain.name&gt;</li><li>2 [vRealize Automation コンソール] をクリックします。  vRealize Automation コンソールを開くには、次の URL を使用することもできます : https://&lt;vra-virtual-hostname.domain.name&gt;/vca</li><li>3 ログインします。</li></ol>	システム管理者ロールを持つユーザーである必要があります。



表 2-2. vRealize Automation テナント コンソール。このインターフェイスは、サービスやリソースの作成および管理に使用するプライマリ ユーザー インターフェイスです。

目的	アクセス	必要な認証情報
<p>以下のタスクには、vRealize Automation を使用します。</p> <ul style="list-style-type: none"> <li>■ 新しい IT サービス ブループリントを申請します。</li> <li>■ クラウドおよび IT リソースを作成および管理します。</li> <li>■ カスタム グループを作成および管理します。</li> <li>■ ビジネス グループを作成、管理します。</li> <li>■ ユーザーにロールを割り当てます。</li> </ul>	<ol style="list-style-type: none"> <li>1 ブラウザを起動し、仮想アプライアンスの完全修飾ドメイン名とテナントの URL 名を使用して、テナントの URL を入力します。  https://&lt;vra-va-hostname.domain.name&gt;/vcac/org/&lt;tenant_URL_name&gt;</li> <li>2 ログインします。</li> </ol>	<p>以下の 1 つ以上のロールを持つユーザーである必要があります。</p> <ul style="list-style-type: none"> <li>■ アプリケーション アーキテクト</li> <li>■ 承認管理者</li> <li>■ カタログ管理者</li> <li>■ コンテナ管理者</li> <li>■ コンテナ アーキテクト</li> <li>■ 健全性サービス ユーザー</li> <li>■ インフラストラクチャ アーキテクト</li> <li>■ セキュアなエクスポートの利用者</li> <li>■ ソフトウェア アーキテクト</li> <li>■ テナント管理者</li> <li>■ XaaS アーキテクト</li> </ul>

表 2-3. vRealize Automation アプライアンス管理。このインターフェイスは、仮想アプライアンス管理インターフェイス (VAMI) と呼ばれます。

目的	アクセス	必要な認証情報
<p>以下のタスクには vRealize Automation アプライアンス管理を使用します。</p> <ul style="list-style-type: none"> <li>登録されているサービスのステータスを表示します。</li> <li>システム情報を表示、およびアプライアンスを再起動またはシャットダウンします。</li> <li>カスタム エクスペリエンス改善プログラムへの参加を管理します。</li> <li>ネットワーク ステータスを表示します。</li> <li>更新ステータスを表示、およびアップデートをインストールします。</li> <li>管理設定を管理します。</li> <li>vRealize Automation ホスト設定を管理します。</li> <li>SSO の設定を管理します。</li> <li>製品ライセンスを管理します。</li> <li>vRealize Automation Postgres データベースを設定します。</li> <li>vRealize Automation メッセージングを設定します。</li> <li>vRealize Automation ログを設定します。</li> <li>IaaS コンポーネントをインストールします。</li> <li>既存の vRealize Automation 環境から移行します。</li> <li>IaaS コンポーネントの証明書を管理します。</li> <li>Xenon サービスを設定します。</li> </ul>	<ol style="list-style-type: none"> <li>ブラウザを起動し、仮想アプライアンスの完全修飾ドメイン名を使用して、vRealize Automation アプライアンスのブラッシュ ページを開きます。  https://&lt;vra-va-hostname.domain.name&gt;</li> <li>[vRealize Automation アプライアンス管理] をクリックします。  次の URL を使用して vRealize Automation アプライアンス管理を開くこともできます : https://&lt;vra-va-hostname.domain.name:5480&gt;</li> <li>ログインします。</li> </ol>	<ul style="list-style-type: none"> <li>ユーザー名 : root</li> <li>パスワード : vRealize Automation アプライアンスを展開したときに入力したパスワード。</li> </ul>

表 2-4. vRealize Orchestrator クライアント

目的	アクセス	必要な認証情報
<p>以下のタスクには、vRealize Orchestrator クライアントを使用します。</p> <ul style="list-style-type: none"> <li>アクションを作成します。</li> <li>ワークフローを作成します。</li> <li>ポリシーを管理します。</li> <li>パッケージをインストールします。</li> <li>ユーザーおよびユーザー グループの権限を管理します。</li> <li>URI オブジェクトにタグを追加します。</li> <li>インベントリを表示します。</li> </ul>	<ol style="list-style-type: none"> <li>ブラウザを起動し、仮想アプライアンスの完全修飾ドメイン名を使用して、vRealize Automation のブラッシュ ページを開きます。  https://&lt;vra-va-hostname.domain.name&gt;</li> <li>ローカル コンピュータに client.jnlp ファイルをダウンロードするには、[vRealize Orchestrator Client] をクリックします。</li> <li>client.jnlp ファイルを右クリックして [起動] を選択します。</li> <li>[続行しますか?] ダイアログボックスで、[続行] をクリックします。</li> <li>ログインします。</li> </ol>	<p>システム管理者ロールを持つユーザーであるか、または vRealize Orchestrator コントロール センターの認証プロバイダの設定で構成されている vcoadmins グループに属している必要があります。</p>

表 2-5. vRealize Orchestrator コントロール センター

目的	アクセス	必要な認証情報
vRealize Automation に組み込まれているデフォルトの vRealize Orchestrator インスタンスの設定を編集するには、vRealize Orchestrator コントロール センターを使用します。	<ol style="list-style-type: none"> <li>1 ブラウザを起動し、仮想アプライアンスの完全修飾ドメイン名を使用して、vRealize Automation アプライアンスのスプラッシュ ページを開きます。  https://&lt;vra-virtual-hostname.domain.name&gt;</li> <li>2 [vRealize Automation アプライアンス管理] をクリックします。  次の URL を使用して vRealize Automation アプライアンス管理を開くこともできます : https://&lt;vra-virtual-hostname.domain.name:5480&gt;</li> <li>3 ログインします。</li> <li>4 [vRA 設定] - [Orchestrator] の順にクリックします。</li> <li>5 [Orchestrator ユーザー インターフェイス] を選択します。</li> <li>6 [開始] をクリックします。</li> <li>7 Orchestrator ユーザー インターフェイスの URL をクリックします。</li> <li>8 ログインします。</li> </ol>	<p>ユーザー名</p> <ul style="list-style-type: none"> <li>■ ロールベースの認証が設定されていない場合は、<b>root</b> と入力します。</li> <li>■ ロールベースの認証で設定されている場合は、vRealize Automation ユーザー名を入力します。</li> </ul> <p>パスワード</p> <ul style="list-style-type: none"> <li>■ ロールベースの認証が設定されていない場合、vRealize Automation アプライアンスを展開したときに入力したパスワードを入力します。</li> <li>■ ロールベースの認証でユーザー名が設定されている場合は、ユーザー名に対するパスワードを入力します。</li> </ul>

表 2-6. Linux コマンド プロンプト

目的	アクセス	必要な認証情報
<p>vRealize Automation アプライアンス ホストなどのホストでは、以下のタスクには Linux コマンド プロンプトを使用します。</p> <ul style="list-style-type: none"> <li>■ サービスの開始または停止</li> <li>■ 構成ファイルの編集</li> <li>■ コマンドの実行</li> <li>■ データの取得</li> </ul>	<ol style="list-style-type: none"> <li>1 vRealize Automation アプライアンス ホストで、コマンド プロンプトを開きます。  ローカル コンピュータでコマンド プロンプトを開く方法の 1 つは、PuTTY などのアプリケーションを使用して、ホストでセッションを開始することです。</li> <li>2 ログインします。</li> </ol>	<ul style="list-style-type: none"> <li>■ ユーザー名 : root</li> <li>■ パスワード : vRealize Automation アプライアンスを展開したときに作成したパスワード。</li> </ul>

表 2-7. Windows コマンド プロンプト

目的	アクセス	必要な認証情報
laaS ホストなどのホスト上で、Windows コマンド プロンプトを使用してスクリプトを実行できます。	<ol style="list-style-type: none"> <li>1 laaS ホスト上で、Windows にログインします。  ローカル コンピュータからログインする方法の 1 つは、リモート デスクトップ セッションを開始することです。</li> <li>2 Windows コマンド プロンプトを開きます。  コマンド プロンプトを開く方法の 1 つは、ホスト上で [スタート] アイコンを右クリックし、[コマンド プロンプト] または [コマンド プロンプト (管理者)] を選択することです。</li> </ol>	<ul style="list-style-type: none"> <li>■ ユーザー名 : 管理者権限を持つユーザー。</li> <li>■ パスワード : ユーザーのパスワード。</li> </ul>

## 移行の前提条件

移行の前提条件はターゲット環境によって異なります。

最小環境または高可用性環境に移行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。

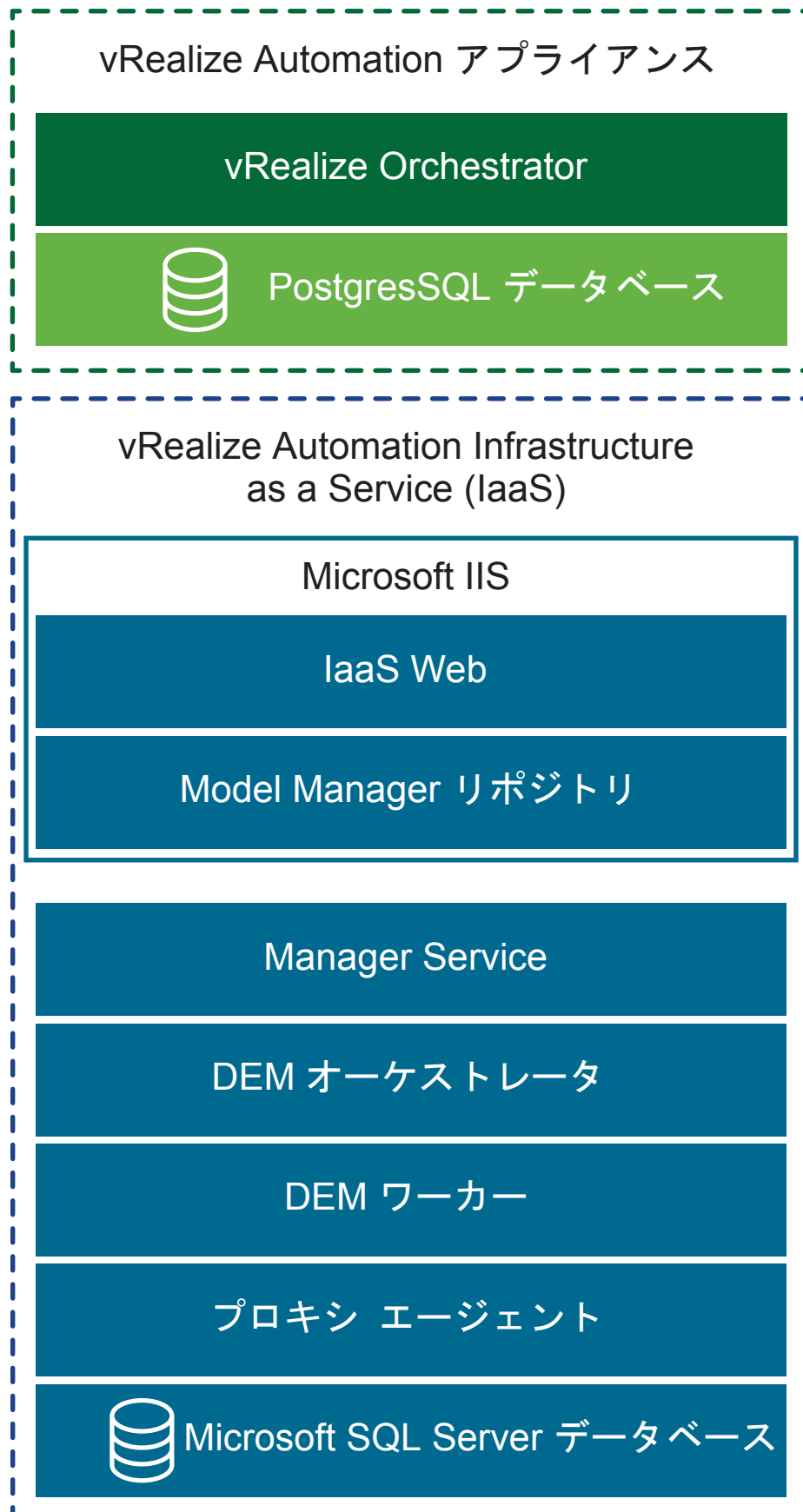
- [最小環境に移行するための前提条件](#)
- [高可用性環境に移行するための前提条件](#)

### 最小環境に移行するための前提条件

最小環境への移行が確実に成功するように、次の前提条件を確認します。

最小インストールには、1 台の vRealize Automation アプライアンス、および IaaS コンポーネントをホストする 1 台の Windows サーバが含まれます。最小インストールでは、vRealize Automation SQL Server データベースを IaaS コンポーネントと同じ IaaS Windows サーバに置くことも、別の Windows サーバに置くこともできます。

図 3-1. vRealize Automation 最小インストール



## 前提条件

- vRealize Automation の新しいターゲット環境があることを確認します。
- 次の要件に基づいて、適切なプロキシ エージェントをターゲット環境にインストールします。
  - ターゲット プロキシ エージェント名は、vSphere 用、Hyper-V 用、Citrix XenServer 用、およびテスト用の各プロキシ エージェントのソース プロキシ エージェント名と一致する必要があります。

---

**注:** 次の手順を実行してエージェント名を取得します。

- 1 IaaS ホストで、**管理者**権限を持つローカル ユーザーとして Windows にログインします。
  - 2 Windows Explorer を使用して、エージェントのインストール ディレクトリに移動します。
  - 3 **VRMAgent.exe.config** ファイルを開きます。
  - 4 ServiceConfiguration タグで、agentName 属性の値を特定します。
- 

- ナレッジベースの記事 [KB51531](#) を確認します。
- ターゲット プロキシ エージェント エンドポイント名は、vSphere 用、Hyper-V 用、Citrix XenServer 用、およびテスト用の各プロキシ エージェントのソース プロキシ エージェント エンドポイント名と一致する必要があります。
- ターゲット環境で vSphere 用、Hyper-V 用、Citrix XenServer 用、またはテスト用の各プロキシ エージェントのエンドポイントを作成しないでください。
- ターゲット vRealize Automation アプライアンスの vRealize Automation コンポーネントのバージョン番号を確認します。
  - a ターゲット vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**root** としてターゲット vRealize Automation アプライアンス管理にログインします。
  - b [vRA 設定] - [クラスタ] の順に選択します。
  - c 三角形をクリックして、[ホスト/ノード名] のレコードを展開します。

vRealize Automation IaaS コンポーネントのバージョン番号が一致していることを確認します。
- vRealize Automation ターゲット IaaS データベース用のターゲット Microsoft SQL Server のバージョンが 2012、2014、または 2016 であることを確認します。
- ポート 22 がソースとターゲットの vRealize Automation 環境間で開いていることを確認します。ポート 22 は、ソースとターゲットの仮想アプライアンス間のセキュア シェル (SSH) 接続を確立するために必要です。
- エンドポイント vCenter に、移行を完了するための十分なリソースがあることを確認します。
- Cafe と IaaS コンポーネントの間で、ターゲット vRealize Automation 環境のシステム時刻が同期していることを確認します。
- ターゲット環境の IaaS サーバ ノードに Java SE Runtime Environment (JRE) 8 Update 161 (64 ビット) 以降がインストールされていることを確認します。JRE をインストールしたら、JAVA\_HOME 環境変数が、各 IaaS ノードにインストールされた Java バージョンをポイントしていることを確認します。必要に応じて、パスを変更します。

- IaaS の各ノードに PowerShell 3.0 以降がインストールされていることを確認します。
- ソースおよびターゲットの vRealize Automation 環境が稼動していることを確認します。
- ソース vRealize Automation 環境で、ユーザー アクティビティやプロビジョニング アクティビティが発生していないことを確認します。
- オペレーティング システムと通信する可能性のあるターゲット vRealize Automation 環境の IaaS ノードで、ウイルス対策またはセキュリティ ソフトウェアが実行されていること、およびコンポーネントが適切に設定、または無効にされていることを確認します。
- Windows インストールの更新が保留中であるため、IaaS Web サービスおよび Model Manager を再起動する必要がないことを確認します。更新が保留中であるため、移行で World Wide Web 発行サービスが開始または終了されない場合があります。

次のステップ

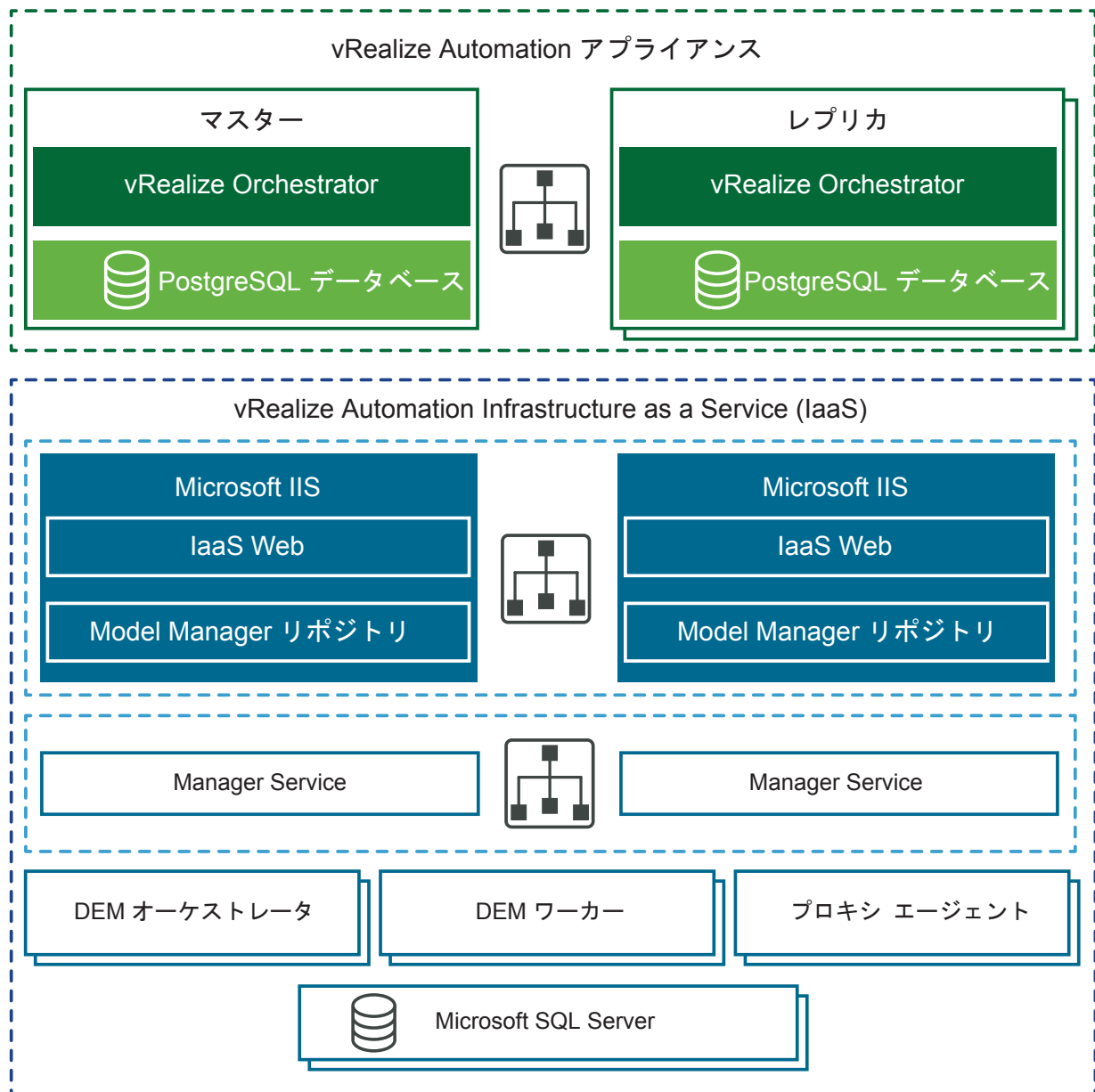
[章 4 「移行前のタスク」](#)。

## 高可用性環境に移行するための前提条件

次の前提条件を確認し、高可用性環境に問題なく確実に移行できるようにします。

高可用性環境には、さまざまなサイズを使用できます。基本的な分散展開では、各 Windows サーバ上で IaaS コンポーネントをホストするだけで vRealize Automation のパフォーマンスを向上させることができます。多くの場合、高可用性環境では、より多くの容量を確保するために、冗長アプライアンス、冗長サーバ、ロード バランシングが使用されます。大規模な分散展開では、優れたスケーラビリティ、高可用性、ディザスタ リカバリが可能になります。

図 3-2. vRealize Automation 高可用性環境





## 前提条件

- 高可用性向けに構成されたマスターおよびレプリカ仮想アプライアンスでターゲット vRealize Automation が新しくインストールされていることを確認します。リファレンス アーキテクチャ で、「vRealize Automation 高可用性構成の考慮事項」を参照してください。
- すべての vRealize Automation 仮想アプライアンスが root ユーザーの同一のパスワードを使用していることを確認します。
- 次の要件に基づいて、適切なプロキシ エージェントをターゲット環境にインストールします。
  - ターゲット プロキシ エージェント名は、vSphere 用、Hyper-V 用、Citrix XenServer 用、およびテスト用の各プロキシ エージェントのソース プロキシ エージェント名と一致する必要があります。

---

**注:** 次の手順を実行してエージェント名を取得します。

- 1 IaaS ホストで、**管理者**権限を持つローカル ユーザーとして Windows にログインします。
  - 2 Windows Explorer を使用して、エージェントのインストール ディレクトリに移動します。
  - 3 **VRMAgent.exe.config** ファイルを開きます。
  - 4 ServiceConfiguration タグで、agentName 属性の値を特定します。
- 

- ターゲット プロキシ エージェント エンドポイント名は、vSphere 用、Hyper-V 用、Citrix XenServer 用、およびテスト用の各プロキシ エージェントのソース プロキシ エージェント エンドポイント名と一致する必要があります。
- ターゲット環境で vSphere 用、Hyper-V 用、Citrix XenServer 用、またはテスト用の各プロキシ エージェントのエンドポイントを作成しないでください。
- ターゲット vRealize Automation アプライアンスの vRealize Automation コンポーネントのバージョン番号を確認します。
  - a ターゲット vRealize Automation 7.3 環境で、ブラウザを起動し、**https:// vra-va-hostname.domain.name:5480** にある vRealize Automation アプライアンス管理コンソールにアクセスします。
  - b ユーザー名 root と、アプライアンスの展開時に入力したパスワードを使用してログインします。
  - c [vRA 設定] - [クラスタ] の順に選択します。
  - d [ホスト/ノード名] のレコードを展開してコンポーネントを表示するために、展開ボタンをクリックします。

すべての仮想アプライアンス ノード間で vRealize Automation コンポーネントのバージョン番号が一致していることを確認します。

すべての IaaS ノード間で vRealize Automation IaaS コンポーネントのバージョン番号が一致していることを確認します。
- ナレッジベースの記事 [KB51531](#) を確認します。
- トラフィックをマスター ノードのみに転送するために、次の手順を実行します。
  - a すべての冗長ノードを無効にします。

- b ロード バランサのドキュメントのとおり、次のアイテムの健全性モニタを削除します。
- vRealize Automation 仮想アプライアンス
  - IaaS Web サイト
  - IaaS Manager Service
- vRealize Automation ターゲット IaaS データベース用のターゲット Microsoft SQL Server のバージョンが 2012、2014、または 2016 であることを確認します。
  - ポート 22 がソースとターゲットの vRealize Automation 環境間で開いていることを確認します。ポート 22 は、ソースとターゲットの仮想アプライアンス間のセキュア シェル (SSH) 接続を確立するために必要です。
  - エンドポイント vCenter に、移行を完了するための十分なリソースがあることを確認します。
  - ロード バランサのタイムアウト設定を、デフォルト設定から 10 分以上に変更したことを確認します。
  - Cafe と IaaS コンポーネントの間で、ターゲット vRealize Automation 環境のシステム時刻が同期していることを確認します。
  - ターゲット環境の IaaS Web サービス ノードと Model Manager ノードに適切な Java Runtime Environment があることを確認します。Java SE Runtime Environment (JRE) 8 Update 161 (64 ビット) 以降がインストールされている必要があります。JAVA\_HOME システム変数が IaaS の各ノードにインストールされている Java のバージョンを指していることを確認します。必要に応じて、パスを変更します。
  - 各 IaaS ノードに PowerShell 3.0 以降がインストールされていることを確認します。
  - ソースおよびターゲットの vRealize Automation 環境が稼動していることを確認します。
  - ソース vRealize Automation 環境で、ユーザー アクティビティやプロビジョニング アクティビティが発生していないことを確認します。
  - オペレーティング システムと通信する可能性のあるターゲット vRealize Automation 環境の IaaS ノードで、ウイルス対策またはセキュリティ ソフトウェアが実行されていること、およびコンポーネントが適切に設定、または無効にされていることを確認します。
  - Windows インストールの更新が保留中であるため、IaaS Web サービスおよび Model Manager を再起動する必要がないことを確認します。更新が保留中であるため、移行で World Wide Web 発行サービスが開始または終了されない場合があります。

#### 次のステップ

章 4 「移行前のタスク」。

## 移行前のタスク

移行する前に、いくつかの移行前のタスクを実行する必要があります。

ソース vRealize Automation 環境のデータからターゲット vRealize Automation 環境に移行する前に実行する移行前タスクは、移行前の環境に応じて異なります。

この章には、次のトピックが含まれています。

- [vRealize Automation 6.2.x から 7.x への移行で導入される変更の確認](#)
- [ソフトウェア エージェント パッチの適用](#)
- [vSphere エージェントでの DoDeletes 設定の False への変更](#)
- [vRealize Automation 6.x 移行前環境のテンプレートの確認](#)
- [vRealize Automation 仮想マシンの移行準備](#)
- [移行に必要な情報の収集](#)
- [ソース vRealize Automation 環境からの暗号化キーの取得](#)
- [ソース vRealize Automation 6.2.x 環境からテナントと IaaS 管理者のリスト作成](#)
- [vRealize Automation の移行前の環境からターゲット環境への各テナントの追加](#)
- [追加された各テナントの管理者の作成](#)
- [最小環境への移行前に Active Directory リンクのユーザーとグループを同期する](#)
- [高可用性環境への移行前に Active Directory リンクのユーザーとグループを同期する](#)
- [ソース vRealize Automation 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行](#)
- [ソース vRealize Automation IaaS Microsoft SQL データベースの手動によるクローン作成](#)
- [ターゲット vRealize Automation 環境のスナップショット作成](#)

### vRealize Automation 6.2.x から 7.x への移行で導入される変更の確認

vRealize Automation 7 以降では、アップグレード プロセスの間、およびプロセス後にさまざまな機能が変更されます。vRealize Automation 6.2.x 環境を最新バージョンにアップグレードする前に、これらの変更を確認する必要があります。

vRealize Automation 6.2.x と 7.x の違いについては、『Upgrading vRealize Automation 6.2.5 to 7.4』の「Considerations About Upgrading to This vRealize Automation Version」を参照してください。

---

**注:** vRealize Production Test Upgrade Assist Tool は、vRealize Automation 6.2.x 環境を分析してアップグレードに関する問題を引き起こす可能性がある機能構成を特定し、使用環境のアップグレード準備ができていることを確認します。このツールおよび関連ドキュメントをダウンロードするには、[VMware vRealize Production Test Tool](#) の製品のダウンロード ページに移動します。

---

vRealize Automation 6.2.x を最新バージョンに移行した後、以下のプロパティ定義を使用したカタログ アイテムはサービス カタログに表示されますが、申請できません。

- コントロール タイプ: チェック ボックスまたはリンク。
- 属性: 関係、正規表現、またはプロパティのレイアウト。

vRealize Automation 7.x では、プロパティ定義でこれらの要素が使用されなくなりました。プロパティ定義を再作成するか、組み込みのコントロール タイプまたは属性以外の vRealize Orchestrator スクリプト アクションを使用するようにプロパティ定義を構成する必要があります。詳細については、「[移行後にサービス カタログに表示されるカタログ アイテムを申請できない](#)」を参照してください。

## ソフトウェア エージェント パッチの適用

vRealize Automation 7.1 または 7.3 から 7.4 へ移行する前に、ソース アプライアンスにホット フィックスを適用して、ソフトウェア エージェントを TLS 1.2 にアップグレードできるようにする必要があります。

トランスポート レイヤー セキュリティ (TLS) プロトコルは、ブラウザと vRealize Automation 間のデータの整合性を確保します。このホット フィックスにより、移行前の環境のソフトウェア エージェントを TLS 1.2 にアップグレードできます。このアップグレードでは最高レベルのセキュリティが確保され、vRealize Automation 7.1 または 7.3 では必須です。各バージョンには独自のホット フィックスがあります。

### 前提条件

移行元となる実行中の vRealize Automation 7.1 または 7.3 の環境。

### 手順

- ◆ 7.4 に移行する前に、移行元の vRealize Automation 7.1 または 7.3 アプライアンスにこのホット フィックスを適用します。[ナレッジベースの記事 KB52897](#) を参照してください。

### 次のステップ

[\[vSphere エージェントでの DoDeletes 設定の False への変更\]](#)。

## vSphere エージェントでの DoDeletes 設定の False への変更

vRealize Automation 6.2.x 環境から移行する場合、移行前にターゲットの vSphere エージェントで DoDeletes 値を **true** から **false** に変更する必要があります。

### 前提条件

移行の前提条件を完成させます。

## 手順

- 1 DoDeletes 値を **false** に変更します。

これにより、ソース環境から仮想マシンが削除されるのを回避します。ソース環境とターゲット環境は並行して実行されます。本番環境での移行が検証された後、リースの不一致が発生する可能性があります。

- 2 本番環境での移行が検証され、ソース環境がシャットダウンされた後、DoDeletes 値を **true** に設定します。

## 次のステップ

[\[vRealize Automation 仮想マシンの移行準備\]](#)。

# vRealize Automation 6.x 移行前環境のテンプレートの確認

vRealize Automation 6.x から 7.4 に移行する前に、すべてのテンプレートに少なくとも 4 MB の最小メモリ設定があることを確認するために仮想マシン テンプレートを確認する必要があります。

vRealize Automation 6.x 移行前環境にメモリが 4 MB 未満の仮想マシン テンプレートがあると、移行は失敗します。6.x の移行前環境のブループリントに 4 MB 未満のメモリがあるかどうかを判断するには、この手順を完了します。

## 前提条件

vRealize Automation 6.x から 7.4 に移行しています。

## 手順

- 1 SSH を使用して、プライマリ vRealize Automation アプライアンスに **root** としてログインします。

vRealize Orchestrator が外部の場合、Orchestrator ホスト マシンにログインします。

- 2 プライマリ ホスト上の PostgreSQL データ フォルダ (/var/vmware/vpostgres/current/pgdata/) にディレクトリを変更します。

- 3 次のスクリプトを実行して、すべてのブループリントで 4 MB 未満に指定されているメモリがあるかどうかを確認します。

```
select * from [vCAC].[dbo].[VirtualMachineTemplate] where IsHidden = 0 and MemoryMB < 4;
```

vCAC はデータベース名です。

- 4 スクリプトで 4 MB 未満に指定されているメモリがあるブループリントが検出された場合は、次のスクリプトを実行してメモリを 4 MB 以上に更新します。

```
update [vCAC].[dbo].[VirtualMachineTemplate] set MemoryMB = 4 where IsHidden = 0 and MemoryMB < 4;
```

vCAC はデータベース名です。

## 次のステップ

[\[vRealize Automation 仮想マシンの移行準備\]](#) .

## vRealize Automation 仮想マシンの移行準備

vRealize Automation 6.2.x 仮想マシンの移行に関する既知の問題により、移行後に問題が発生する場合があります。

ナレッジベースの記事 [KB000051531](#) を 移行前に参照して、関連する修正を環境で実行する必要があります。

### 次のステップ

[「移行に必要な情報の収集」](#)。

## 移行に必要な情報の収集

以下のテーブルを使用して、移行前の環境からターゲット環境への移行に必要な情報を記録します。

### 前提条件

移行の前提条件の検証を完了します。

- [「最小環境に移行するための前提条件」](#)。
- [「高可用性環境に移行するための前提条件」](#)。

表 4-1. ソース vRealize Automation アプライアンス

オプション	説明	値
ホスト名	ソース vRealize Automation アプライアンス管理にログインします。[システム] タブでホスト名を見つけます。ホスト名は、完全修飾ドメイン名 (FQDN) を使用する必要があります。	
root ユーザー名	root	
root パスワード	ソース vRealize Automation アプライアンスを展開したときに入力した root パスワード。	
移行パッケージの場所	移行パッケージが作成された、移行元の vRealize Automation 6.2.x または 7.x アプライアンス上の既存のディレクトリへのパス。ディレクトリは、vRealize Automation データベースのサイズの 2 倍の容量が使用できる必要があります。デフォルトの場所は、 <code>/storage</code> です。	

表 4-2. ターゲット vRealize Automation アプライアンス

オプション	説明	値
root ユーザー名	root	
root パスワード	ターゲット vRealize Automation アプライアンスを展開したときに入力した root パスワード。	
デフォルトのテナント	vsphere.local	
管理者ユーザー名	管理者	
管理者パスワード	ターゲット vRealize Automation 環境を展開したときに入力した administrator@vsphere.local ユーザーのパスワード。	

表 4-3. ターゲット IaaS データベース

オプション	説明	値
データベース サーバ	クローン作成されたデータベースが配置される Microsoft SQL Server インスタンスの場所。名前付きインスタンスでデフォルト以外のポートが使用されている場合は、「SERVER,PORT\INSTANCE-NAME」という形式で指定します。	
クローン作成されたデータベース名	移行のためにクローン作成されたソース vRealize Automation 6.2.x/7.x IaaS Microsoft SQL データベースの名前。	
認証モード	Windows または SQL Server のいずれかを選択します。[SQL Server] を選択する場合は、ログイン名とパスワードを入力する必要があります。	
ログイン名	クローン作成された IaaS Microsoft SQL データベースの db_owner ロールを持つ SQL Server ユーザーのログイン名。	
パスワード	SQL Server ユーザーのパスワード。	
元の暗号化キー	移行前の環境から取得する元の暗号化キー。 <a href="#">「ソース vRealize Automation 環境からの暗号化キーの取得」</a> を参照してください。	
新しいパスフレーズ	新しい暗号化キーを生成するために使用する一連の単語。このパスフレーズは、ターゲット vRealize Automation 環境に新しい IaaS コンポーネントをインストールするときに使用します。	

#### 次のステップ

[「ソース vRealize Automation 環境からの暗号化キーの取得」](#)。

## ソース vRealize Automation 環境からの暗号化キーの取得

移行手順の一環として、ソース vRealize Automation 環境の暗号化キーを入力する必要があります。

#### 前提条件

移行前の環境内のアクティブな Manager Service ホスト仮想マシンに対して、管理者権限があることを確認します。

#### 手順

- 1 移行前の環境内のアクティブな Manager Service をホストする仮想マシン上で、管理者としてコマンド プロンプトを開き、次のコマンドを実行します。

```
"C:\Program Files
(x86)\VMware\VCAC\Server\ConfigTool\EncryptionKeyTool\DynamicOps.Tools.EncryptionKeyTool.exe" key-read -c "C:\Program Files
(x86)\VMware\VCAC\Server\ManagerService.exe.config" -v
```

インストール ディレクトリがデフォルトの場所 (C:\Program Files (x86)\VMware\VCAC) にない場合は、パスを編集して、実際のインストール ディレクトリを表示します。

- 2 コマンドを実行して表示されるキーを保存します。

キーは次の例のような長い文字列です。

NRH+f/BlnCB6yvasLS3sxespgdkcFWAEuyV0g4lfryg=。

#### 次のステップ

- vRealize Automation 6.2.x 環境から移行する場合は、[「vRealize Automation の移行前の環境からターゲット環境への各テナントの追加」](#)。
- vRealize Automation 7.x 環境から移行する場合は、[「ソース vRealize Automation 6.2.x 環境からテナントと IaaS 管理者のリスト作成」](#)。

## ソース vRealize Automation 6.2.x 環境からテナントと IaaS 管理者のリスト作成

vRealize Automation 6.2.x 環境を移行する前に、テナントと各テナントの IaaS 管理者のリストを作成する必要があります。

ソース vRealize Automation コンソールで、テナントごとに次の手順を実行します。

---

**注:** vRealize Automation 7.x 環境から移行する場合は、この手順を実行する必要はありません。

---

#### 前提条件

ソース vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**管理者**としてソース vRealize Automation コンソールにログインします。

---

**注:** 高可用性環境の場合は、ソース仮想アプライアンスのロード バランサの完全修飾ドメイン名 (<https://<vra-vault-hostname.domain.name>/vcac>) を使用して、コンソールを開きます。

---

#### 手順

- 1 [管理] - [テナント] を選択します。
- 2 テナント名をクリックします。
- 3 [管理者] をクリックします。
- 4 各テナントと IaaS 管理者のユーザー名のリストを作成します。
- 5 [キャンセル] をクリックします。

#### 次のステップ

[「vRealize Automation の移行前の環境からターゲット環境への各テナントの追加」](#)。

## vRealize Automation の移行前の環境からターゲット環境への各テナントの追加

ターゲット環境にテナントを追加するには、移行前の環境での各テナントの名前を使用する必要があります。



移行が正常に実行されるためには、移行前の環境の各テナントがターゲット環境に作成されることが必要です。また、移行前の環境のテナント URL 名を使用して追加する各テナントに、固有のアクセス URL を使用することも必要です。移行前の環境に未使用のテナントがあり、それを移行しない場合は、移行する前に移行元の環境から削除します。

**注:** 移行の検証により、少なくとも、前提条件の要求どおりに移行元で設定されているのと同じだけのテナントが、移行先のシステムにあることが確認されます。テナント名ではなくテナント URL 名に基づいて、大文字と小文字を区別してテナントが比較されます。

移行前の環境の各テナントに対してこの手順を実行します。


- vRealize Automation 6.2.x 環境から移行する場合は、移行前の環境からターゲット環境の VMware Identity Manager に既存の SSO2 テナントおよび ID ストアを移行します。
- vRealize Automation 7.x 環境から移行する場合は、移行前の環境からターゲット環境の VMware Identity Manager に既存の VMware Identity Manager テナントおよび ID ストアを移行します。

#### 前提条件

- [「移行に必要な情報の収集」](#)。
- ターゲット vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**管理者**としてターゲット vRealize Automation コンソールにログインします。

**注:** 高可用性環境の場合は、ターゲット仮想アプライアンス ロード バランサの完全修飾ドメイン名 `https://<vra-vb-lb-hostname.domain.name>/vcac` を使用してコンソールを開きます。

#### 手順

- 1 [管理] - [テナント] を選択します。
- 2 [新規] アイコン (  ) をクリックします。
- 3 [名前] テキスト ボックスに、移行前の環境のテナント名と一致するテナント名を入力します。  
たとえば、移行前の環境のテナント名が DEVTenant の場合は、**DEVTenant** と入力します。
- 4 (オプション) [説明] テキスト ボックスに説明を入力します。
- 5 [URL 名] テキスト ボックスに、移行前の環境のテナント URL 名と一致するテナント URL 名を入力します。  
この URL 名は、vRealize Automation コンソール URL の末尾にテナント固有の識別子を追加するときに使用します。  
たとえば、移行前の環境の DEVTenant の URL 名が dev の場合は、**dev** と入力して `https://<vra-vb-lb-hostname.domain.name>/vcac/org/dev` という URL を作成します。
- 6 (オプション) 電子メール アドレスを [連絡先電子メール] テキスト ボックスに入力します。
- 7 [送信して次へ] をクリックします。

#### 次のステップ

[「追加された各テナントの管理者の作成」](#)。

## 追加された各テナントの管理者の作成

ターゲット環境に追加したテナントごとに、管理者を作成する必要があります。管理者を作成するには、ローカル ユーザー アカウントを作成し、そのアカウントにテナント管理者権限を割り当てます。

ターゲット環境の各テナントに対してこの手順を実行します。

### 前提条件

- [「vRealize Automation の移行前の環境からターゲット環境への各テナントの追加」](#)。
- ターゲット vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**管理者**としてターゲット vRealize Automation コンソールにログインします。

---

**注:** 高可用性環境の場合は、ターゲット仮想アプライアンス ロード バランサの完全修飾ドメイン名 `https://<vra-vb-lb-hostname.domain.name>/vcac` を使用してコンソールを開きます。

---

### 手順

- 1 [管理] - [テナント] を選択します。
- 2 追加したテナントをクリックします。  
たとえば、DEVTenant の場合は、[DEVTenant] をクリックします。
- 3 [ローカル ユーザー] をクリックします。
- 4 [新規] アイコン (+) をクリックします。
- 5 [ユーザー詳細] で、必要な情報を入力して、テナント管理者ロールを割り当てるローカル ユーザー アカウントを作成します。  
ローカル ユーザー名は、デフォルトのローカル ディレクトリ (vsphere.local) で一意である必要があります。
- 6 [OK] をクリックします。
- 7 [管理者] をクリックします。
- 8 [テナント管理者] 検索ボックスにローカル ユーザー名を入力し、Enter を押します。
- 9 検索の結果から適切な名前をクリックして、テナント管理者のリストにユーザーを追加します。
- 10 [完了] をクリックします。
- 11 コンソールからログアウトします。

### 次のステップ

- 最小インストールの場合：「[最小環境への移行前に Active Directory リンクのユーザーとグループを同期する](#)」
- 高可用性展開の場合：「[高可用性環境への移行前に Active Directory リンクのユーザーとグループを同期する](#)」

## 最小環境への移行前に Active Directory リンクのユーザーとグループを同期する

vRealize Automation の最小インストールにユーザーおよびグループをインポートするには、移行先の vRealize Automation を Active Directory リンクに接続する必要があります。

テナントごとに次の手順を実行します。テナントに複数の Active Directory がある場合は、テナントが使用する Active Directory ごとにこの手順を実行します。

### 前提条件

- 「追加された各テナントの管理者の作成」。
- Active Directory へのアクセス権限があることを確認します。
- テナント管理者として vRealize Automation にログインします。

### 手順

- 1 [管理] - [ディレクトリ管理] - [ディレクトリ] の順に選択します。
- 2 [ディレクトリを追加] アイコン (+) をクリックし、[Active Directory over LDAP/IWA の追加] を選択します。
- 3 Active Directory のアカウント設定を入力します。

#### ◆ ネイティブ以外の Active Directory 用

オプション	入力例
ディレクトリ名	一意のディレクトリ名を入力します。 ネイティブ以外の Active Directory を使用する場合は、[LDAP 経由の Active Directory] を選択します。
このディレクトリは DNS サービス ローケーションをサポートします	このオプションは選択解除します。
ベース DN	ディレクトリ サーバ検索の先頭に識別名 (DN) を入力します。 たとえば、[cn=users,dc=rainpole,dc=local] と入力します。
バインド DN	共通名 (CN) など、ユーザーを検索する権限がある Active Directory ユーザー アカウントの完全識別名 (DN) を入力します。 たとえば、[cn=config_admin infra,cn=users,dc=rainpole,dc=local] と入力します。
バインド DN パスワード	ユーザーを検索できるアカウントの Active Directory パスワードを入力し、[接続をテスト] をクリックして構成したディレクトリへの接続をテストします。

#### ◆ ネイティブの Active Directory 用

オプション	入力例
ディレクトリ名	一意のディレクトリ名を入力します。 ネイティブの Active Directory を使用する場合は、[Active Directory (統合 Windows 認証)] を選択します。
ドメイン名	参加するドメインの名前を入力します。
ドメイン管理者ユーザー名	ドメイン管理者のユーザー名を入力します。
ドメイン管理者パスワード	ドメイン管理者のパスワードを入力します。

オプション	入力例
バインド ユーザー UPN	メール アドレス形式を使用して、ドメインで認証できるユーザーの名前を入力します。
バインド DN パスワード	ユーザーを検索できるアカウントの Active Directory バインド アカウント パスワードを入力します。

4 [保存して次へ] をクリックします。

[ドメインの選択] にドメインのリストが表示されます。

5 デフォルトのドメイン設定を受け入れ、[次へ] をクリックします。

6 属性名が適切な Active Directory 属性にマップされていることを確認し、[次へ] をクリックします。

7 同期するグループおよびユーザーを選択します。

a [新規] アイコン (+) をクリックします。

b ユーザー ドメインを入力し、[グループの検索] をクリックします。

たとえば、**dc=vcac,dc=local** と入力します。

c 同期するグループを選択するには、[選択] をクリックし、[次へ] をクリックします。

d [ユーザーの選択] で、同期するユーザーを選択し、[次へ] をクリックします。

vRealize Automation の使用を必要とするユーザーおよびグループのみを追加します。[ネストされたグループの同期] は、ネスト内のすべてのグループが vRealize Automation の使用を必要とする場合以外には、選択しないでください。

8 ディレクトリと同期しているユーザーおよびグループを確認し、[ディレクトリの同期] をクリックします。

ディレクトリの同期には少し時間がかかりますが、バックグラウンドで実行されます。

#### 次のステップ

[\[ソース vRealize Automation 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行\]](#)

## 高可用性環境への移行前に Active Directory リンクのユーザーとグループを同期する

高可用性の vRealize Automation 環境にユーザーおよびグループをインポートするには、Active Directory リンクに接続する必要があります。

- 各テナントに対して手順 1 ～ 8 を実行します。テナントに複数の Active Directory がある場合は、テナントが使用する Active Directory ごとにこの手順を実行します。
- テナントに関連付けられている ID プロバイダごとに、手順 9 と 10 を繰り返します。

#### 前提条件

- [\[追加された各テナントの管理者の作成\]](#)。
- Active Directory へのアクセス権限があることを確認します。
- テナント管理者として vRealize Automation にログインします。

## 手順

- 1 [管理] - [ディレクトリ管理] - [ディレクトリ] の順に選択します。
- 2 [ディレクトリを追加] アイコン (+) をクリックし、[Active Directory over LDAP/IWA の追加] を選択します。
- 3 Active Directory のアカウント設定を入力します。

## ◆ ネイティブ以外の Active Directory 用

オプション	入力例
ディレクトリ名	一意のディレクトリ名を入力します。 ネイティブ以外の Active Directory を使用する場合は、[LDAP 経由の Active Directory] を選択します。
このディレクトリは DNS サービス ローケーションをサポートします	このオプションは選択解除します。
ベース DN	ディレクトリ サーバ検索の先頭に識別名 (DN) を入力します。 たとえば、[cn=users,dc=rainpole,dc=local] と入力します。
バインド DN	共通名 (CN) など、ユーザーを検索する権限がある Active Directory ユーザー アカウントの完全識別名 (DN) を入力します。 たとえば、[cn=config_admin infra,cn=users,dc=rainpole,dc=local] と入力します。
バインド DN パスワード	ユーザーを検索できるアカウントの Active Directory パスワードを入力し、[接続をテスト] をクリックして構成したディレクトリへの接続をテストします。

## ◆ ネイティブの Active Directory 用

オプション	入力例
ディレクトリ名	一意のディレクトリ名を入力します。 ネイティブの Active Directory を使用する場合は、[Active Directory (統合 Windows 認証)] を選択します。
ドメイン名	参加するドメインの名前を入力します。
ドメイン管理者ユーザー名	ドメイン管理者のユーザー名を入力します。
ドメイン管理者パスワード	ドメイン管理者アカウントのパスワードを入力します。
バインド ユーザー UPN	メール アドレス形式を使用して、ドメインで認証できるユーザーの名前を入力します。
バインド DN パスワード	ユーザーを検索できるアカウントの Active Directory バインド アカウント パスワードを入力します。

- 4 [保存して次へ] をクリックします。  
[ドメインの選択] ページにドメインのリストが表示されます。
- 5 デフォルトのドメイン設定を受け入れ、[次へ] をクリックします。
- 6 属性名が適切な Active Directory 属性にマップされていることを確認し、[次へ] をクリックします。
- 7 同期するグループおよびユーザーを選択します。
  - a [新規] アイコン (+) をクリックします。
  - b ユーザー ドメインを入力し、[グループの検索] をクリックします。  
たとえば、**dc=vcac,dc=local** と入力します。

- c 同期するグループを選択するには、[選択] をクリックし、[次へ] をクリックします。
- d **[ユーザーの選択]** ページで、同期するユーザーを選択し、[次へ] をクリックします。  
vRealize Automation の使用を必要とするユーザーおよびグループのみを追加します。[ネストされたグループの同期] は、ネスト内のすべてのグループが vRealize Automation の使用を必要とする場合以外には、選択しないでください。
- 8 ディレクトリと同期しているユーザーおよびグループを確認し、[ディレクトリの同期] をクリックします。  
ディレクトリの同期には少し時間がかかりますが、バックグラウンドで実行されます。
- 9 [管理] - [ディレクトリ管理] - [ID プロバイダ] の順に選択して、新しい ID プロバイダをクリックします。  
たとえば、[WorkspaceIDP\_\_1] をクリックします。
- 10 選択した ID プロバイダのページで各ノードのコネクタを追加します。
  - a [コネクタの追加] については説明に従ってください。
  - b vRealize Automation ロード バランサの完全修飾ドメイン名 (FQDN) をポイントするように [IdP ホスト名] プロパティの値を更新します。
  - c [保存] をクリックします。

#### 次のステップ

[\[ソース vRealize Automation 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行\]](#)。

## ソース vRealize Automation 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行

移行の前に、ソース vRealize Automation 環境で NSX ネットワークおよびセキュリティ インベントリ データ収集を実行する必要があります。

このデータ収集は、7.1、7.2、7.3 の展開からの移行時に、vRealize Automation 7.4 でロード バランサの再構成アクションを行うために必要です。

---

**注:** vRealize Automation 6.2.x から移行する場合は、ソース環境でこのデータ収集を実行する必要はありません。vRealize Automation 6.2.x では、ロード バランサの再構成アクションをサポートしていません。

---

#### 手順

- ◆ vRealize Automation 7.4 に移行する前に、ソースの vRealize Automation 環境で NSX ネットワークとセキュリティ インベントリのデータ収集を実行します。『vRealize Automation の管理』の手動によるエンドポイント データ収集の開始を参照してください。

#### 次のステップ

[\[ソース vRealize Automation IaaS Microsoft SQL データベースの手動によるクローン作成\]](#)。

## ソース vRealize Automation IaaS Microsoft SQL データベースの手動によるクローン作成

移行の前に、vRealize Automation 移行前の環境の IaaS Microsoft SQL データベースをバックアップし、それを vRealize Automation ターゲット環境に作成した新しい空のデータベースにリストアする必要があります。

### 前提条件

- [「ソース vRealize Automation 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行」](#)。
- SQL Server データベースのバックアップとリストアに関する情報を取得します。SQL Server データベースの完全バックアップの作成および SQL Server データベースの新しい場所へのリストアに関する [Microsoft Developer Network](#) の記事を参照してください。

### 手順

- ◆ ソースの vRealize Automation 6.2.x または 7.x IaaS Microsoft SQL データベースのフル バックアップを作成します。このバックアップを使用して、ターゲット環境に作成した新しい空のデータベースに SQL データベースをリストアします。

### 次のステップ

[「ターゲット vRealize Automation 環境のスナップショット作成」](#)。

## ターゲット vRealize Automation 環境のスナップショット作成

各 vRealize Automation ターゲット仮想マシンのスナップショットを作成します。移行に失敗した場合は、仮想マシンのスナップショットを使用して再び移行できます。

詳細については、vSphere のドキュメントを参照してください。

### 前提条件

[「ソース vRealize Automation IaaS Microsoft SQL データベースの手動によるクローン作成」](#)。

### 次のステップ

次のいずれかの手順を実行します。

- [「vRealize Automation 7.4 最小環境への vRealize Automation ソース データの移行」](#)。
- [「vRealize Automation 7.4 高可用性環境への vRealize Automation ソース データの移行」](#)。

## 移行手順

ソース vRealize Automation 環境のデータを移行するために実行する手順は、最小環境と高可用性環境のどちらに移行するかによって異なります。

この章には、次のトピックが含まれています。

- [vRealize Automation 7.4 最小環境への vRealize Automation ソース データの移行](#)
- [vRealize Automation 7.4 高可用性環境への vRealize Automation ソース データの移行](#)

### vRealize Automation 7.4 最小環境への vRealize Automation ソース データの移行

vRealize Automation 環境のデータを vRealize Automation 7.4 の新しいインストールに移行することができます。

ソース システムのすべてのテナントは、ターゲットで再作成し、「ID ストアの移行」手順を実行する必要があります。

#### 前提条件

- 「移行に必要な情報の収集」。
- 「ソース vRealize Automation 環境からの暗号化キーの取得」。
- 「vRealize Automation の移行前の環境からターゲット環境への各テナントの追加」。
- 「追加された各テナントの管理者の作成」。
- 「最小環境への移行前に Active Directory リンクのユーザーとグループを同期する」。
- 「ソース vRealize Automation IaaS Microsoft SQL データベースの手動によるクローン作成」。
- 「ターゲット vRealize Automation 環境のスナップショット作成」。
- ターゲット vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**root** としてターゲット vRealize Automation アプライアンス管理にログインします。

#### 手順

- 1 [vRA 設定] - [移行] の順に選択します。



## 2 ソース vRealize Automation アプライアンスの情報を入力します。

オプション	説明
ホスト名	ソース vRealize Automation アプライアンスのホスト名。
root ユーザー名	<b>root</b>
root パスワード	vRealize Automation アプライアンスを展開したときに入力した root パスワード。
移行パッケージの場所	移行パッケージが作成された、移行元の vRealize Automation 6.2.x または 7.x アプライアンス上の既存のディレクトリへのパス。

## 3 ターゲット vRealize Automation アプライアンスの情報を入力します。

オプション	説明
root ユーザー名	<b>root</b>
root パスワード	ターゲット vRealize Automation アプライアンスを展開したときに入力した root パスワード。
デフォルトのテナント	vsphere.local このフィールドを変更することはできません。
管理者ユーザー名	管理者 このフィールドを変更することはできません。
管理者パスワード	ターゲット vRealize Automation 環境を展開したときに入力した administrator@vsphere.local ユーザーのパスワード。

## 4 ターゲット IaaS データベース サーバの情報を入力します。

オプション	説明
データベース サーバ	リストアされた vRealize Automation IaaS Microsoft SQL データベースがある Microsoft SQL Server の場所。名前付きインスタンスとデフォルト以外のポートが使用されている場合は、「<SERVER,PORT\INSTANCE-NAME>」という形式で入力します。AlwaysOn 可用性グループ (AAG) 機能を使用するように、移行先の Microsoft SQL Server を構成する場合は、ポートやインスタンス名を使用せずに、AAG リスナーの名前として移行先の SQL Server を入力します。
クローン作成されたデータベース名	ソースでバックアップを作成し、ターゲット環境でリストアしたソース vRealize Automation 6.2.x または 7.x IaaS Microsoft SQL データベースの名前。
認証モード	<ul style="list-style-type: none"> <li>■ [Windows] Windows 認証モードを使用する場合、IaaS サービスのユーザーには、SQL Server の db_owner ロールを指定する必要があります。SQL Server 認証モードを使用する場合も同じ権限が適用されます。</li> <li>■ [SQL Server] [SQL Server] の場合は、[ログイン名] テキスト ボックスおよび [パスワード] テキスト ボックスが開きます。</li> </ul>
ログイン名	クローン作成された IaaS Microsoft SQL データベースで db_owner ロールを持つ SQL Server ユーザーのログイン名。
パスワード	クローン作成された IaaS Microsoft SQL データベースで db_owner ロールを持つ SQL Server ユーザーのパスワード。

オプション	説明
元の暗号化キー	移行前の環境から取得する元の暗号化キー。 <a href="#">「ソース vRealize Automation 環境からの暗号化キーの取得」</a> を参照してください。
新しいパスフレーズ	新しい暗号化キーを生成するために使用する一連の単語。このパスフレーズは、ターゲット vRealize Automation 環境に新しい IaaS コンポーネントをインストールするときに使用します。

## 5 [検証] をクリックします。

ページに検証の進行状況が表示されます。

- すべてのアイテムが正常に検証された場合は、手順 8 に進みます。
- アイテムが検証に失敗した場合は、エラー メッセージと IaaS ノードの検証ログ ファイルを検査します。ログ ファイルの場所については、[「移行ログの場所」](#)を参照してください。[設定の編集] をクリックし、問題のアイテムを編集します。手順 7 に進みます。

## 6 [移行] をクリックします。

ページに移行の進行状況が表示されます。

- 移行が正常に完了した場合は、このページにすべての移行タスクが完了と表示されます。
- 移行が失敗した場合は、仮想アプライアンスと IaaS ノードで移行ログ ファイルを検査します。ログ ファイルの場所については、[「移行ログの場所」](#)を参照してください。

移行を再開する前に、次の手順を完了します。

- ターゲット vRealize Automation 環境を、移行前にスナップショットを作成したときにキャプチャした状態に戻します。
- ソース IaaS データベースのバックアップを使用して、ターゲット IaaS Microsoft SQL データベースを復旧します。

### 次のステップ

[章 6 「移行後のタスク」](#)。

# vRealize Automation 7.4 高可用性環境への vRealize Automation ソース データの移行

高可用性環境として構成された vRealize Automation 7.4 の新しいインストールに、現在の vRealize Automation 環境データを移行できます。

ソース システムのすべてのテナントは、ターゲットで再作成し、「ID ストアの移行」手順を実行する必要があります。

### 前提条件

- [「移行に必要な情報の収集」](#)。
- [「ソース vRealize Automation 環境からの暗号化キーの取得」](#)。
- [「vRealize Automation の移行前の環境からターゲット環境への各テナントの追加」](#)。
- [「追加された各テナントの管理者の作成」](#)。

- 「高可用性環境への移行前に Active Directory リンクのユーザーとグループを同期する」。
- 「ソース vRealize Automation IaaS Microsoft SQL データベースの手動によるクローン作成」。
- 「ターゲット vRealize Automation 環境のスナップショット作成」。
- ターゲット vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**root** としてターゲット vRealize Automation アプライアンス管理にログインします。

#### 手順

- 1 [vRA 設定] - [移行] の順に選択します。
- 2 ソース vRealize Automation アプライアンス の情報を入力します。

オプション	説明
ホスト名	ソース vRealize Automation アプライアンスのホスト名。
root ユーザー名	<b>root</b>
root パスワード	ソース vRealize Automation アプライアンスを展開したときに入力した root パスワード。

- 3 移行元 vRealize Automation アプライアンス上の移行パッケージの場所を入力します。

オプション	説明
移行パッケージの場所	移行パッケージが作成された、移行元の vRealize Automation 6.2.x または 7.x アプライアンス上の既存のディレクトリへのパス。

- 4 ターゲット vRealize Automation アプライアンスの情報を入力します。

オプション	説明
root ユーザー名	<b>root</b>
root パスワード	ターゲット vRealize Automation アプライアンスを展開したときに入力した root パスワード。
デフォルトのテナント	vsphere.local
管理者ユーザー名	管理者
管理者パスワード	ターゲット vRealize Automation 環境を展開したときに入力した administrator@vsphere.local ユーザーのパスワード。

- 5 ターゲット IaaS データベース サーバの情報を入力します。

オプション	説明
データベース サーバ	リストアされた vRealize Automation IaaS Microsoft SQL データベースが配置された Microsoft SQL Server インスタンスの場所。名前付きインスタンスとデフォルト以外のポートが使用されている場合は、「<SERVER,PORT\INSTANCE-NAME>」という形式で入力します。AlwaysOn 可用性グループ (AAG) 機能を使用するように、移行先の Microsoft SQL Server を構成する場合は、ポートやインスタンス名を使用せずに、AAG リスナーの名前として移行先の SQL Server を入力します。
クローン作成されたデータベース名	ソースでバックアップを作成し、ターゲット環境でリストアしたソース vRealize Automation 6.2.x または 7.x IaaS Microsoft SQL データベースの名前。

オプション	説明
認証モード	<ul style="list-style-type: none"> <li>■ [Windows] Windows 認証モードを使用する場合、IaaS サービスのユーザーには、SQL Server の db_owner ロールを指定する必要があります。SQL Server 認証モードを使用する場合も同じ権限が適用されます。</li> <li>■ [SQL Server] [SQL Server] の場合は、[ログイン名] テキスト ボックスおよび [パスワード] テキスト ボックスが開きます。</li> </ul>
ログイン名	クローン作成された IaaS Microsoft SQL データベースで db_owner ロールを持つ SQL Server ユーザーのログイン名。
パスワード	クローン作成された IaaS Microsoft SQL データベースで db_owner ロールを持つ SQL Server ユーザーのパスワード。
元の暗号化キー	移行前の環境から取得する元の暗号化キー。 <a href="#">「ソース vRealize Automation 環境からの暗号化キーの取得」</a> を参照してください。
新しいパスフレーズ	新しい暗号化キーを生成するために使用する一連の単語。このパスフレーズは、ターゲット vRealize Automation 環境に新しい IaaS コンポーネントをインストールするたびに使用します。

## 6 [検証] をクリックします。

ページに検証の進行状況が表示されます。

- すべてのアイテムが正常に検証された場合は、手順 8 に進みます。
- アイテムが検証に失敗した場合は、エラー メッセージと IaaS ノードの検証ログ ファイルを検査します。ログ ファイルの場所については、[「移行ログの場所」](#)を参照してください。[設定の編集] をクリックし、問題のアイテムを編集します。手順 7 に進みます。

## 7 [移行] をクリックします。

ページに移行の進行状況が表示されます。

- 移行が正常に完了した場合は、このページにすべての移行タスクが完了と表示されます。
- 移行が失敗した場合は、仮想アプライアンスと IaaS ノードで移行ログ ファイルを検査します。ログ ファイルの場所については、[「移行ログの場所」](#)を参照してください。

移行を再開する前に、次の手順を完了します。

- ターゲット vRealize Automation 環境を、移行前にスナップショットを作成したときにキャプチャした状態に戻します。
- ソース IaaS データベースのバックアップを使用して、ターゲット IaaS Microsoft SQL データベースを復旧します。

### 次のステップ

[章 6 「移行後のタスク」](#)。

## 移行後のタスク

vRealize Automation を移行した後、状況に応じて移行後のタスクを実行します。

注: ID ストアを移行後、vRealize Code Stream のユーザーは手動で vRealize Code Stream ロールを再割り当てる必要があります。

この章には、次のトピックが含まれています。

- ソース vRealize Automation 6.2.x 環境からのテナントと IaaS 管理者の追加
- 接続テストの実行と移行後のエンドポイントの確認
- ターゲット vRealize Automation 7.4 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行
- 移行後のロード バランサを高可用性環境に再構成する
- 外部 Orchestrator サーバから vRealize Automation 7.4 への移行
- ターゲットの vRealize Orchestrator の vRealize Automation エンドポイントの再構成
- ターゲットの vRealize Orchestrator の vRealize Automation インフラストラクチャ エンドポイントの再構成
- vRealize Orchestrator カスタマイズのインストール
- ターゲット vRealize Automation に組み込まれた vRealize Orchestrator インフラストラクチャ エンドポイントの再構成
- ターゲット vRealize Automation 環境の Azure エンドポイントの再構成
- vRealize Automation 6.2.x Automation Application Services から 7.4 への移行
- 元のターゲット vRealize Automation IaaS Microsoft SQL データベースの削除
- 移行後にデータセンターの [場所] メニューの内容を更新する
- ソフトウェア エージェントの TLS 1.2 へのアップグレード
- 移行後にプロパティ ディクショナリ設定を変更する
- ターゲット vRealize Automation 7.4 環境の検証

## ソース vRealize Automation 6.2.x 環境からのテナントと IaaS 管理者の追加

移行後、各テナントの vRealize Automation 6.2.x テナント管理者を削除し、復旧する必要があります。

ターゲット vRealize Automation コンソールの各テナントで、次の手順を実行します。

---

**注:** vRealize Automation 7.x 環境から移行する場合は、この手順を実行する必要はありません。

---

### 前提条件

- vRealize Automation の最新バージョンへの正常な移行。
- ターゲット vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**管理者**としてターゲット vRealize Automation コンソールにログインします。

### 手順

- 1 [管理] - [テナント] を選択します。
- 2 テナント名をクリックします。
- 3 [管理者] をクリックします。
- 4 各テナント管理者名とユーザー名のリストを作成します。
- 5 すべての管理者を削除するまで、各管理者をポイントして削除アイコン ([削除]) をクリックします。
- 6 [完了] をクリックします。
- 7 [テナント] ページで、テナント名をもう一度クリックします。
- 8 [管理者] をクリックします。
- 9 適切な検索ボックスに削除した各ユーザーの名前を入力し、Enter キーを押します。
- 10 検索結果から該当するユーザー名をクリックして、そのユーザーを管理者として追加して戻します。  
完了すると、テナント管理者のリストは、削除した管理者のリストと同じになります。
- 11 [完了] をクリックします。

## 接続テストの実行と移行後のエンドポイントの確認

vRealize Automation 7.4 に移行すると、ターゲット環境のエンドポイントに変更が加えられます。

vRealize Automation 7.4 への移行後には、該当するすべてのエンドポイントに対して [接続をテスト] アクションを実行する必要があります。また、移行後の一部のエンドポイントで調整が必要になる場合があります。詳細については、『vRealize Automation の構成』の「アップグレードまたは移行後のエンドポイントを使用する場合の考慮事項」を参照してください。

アップグレードまたは移行されたエンドポイントのデフォルトのセキュリティ設定では、信頼されていない証明書を受け入れません。

信頼されていない証明書を使用していた場合、以前の vRealize Automation インストール環境からアップグレードまたは移行した後、vSphere と NSX のすべてのエンドポイントに対して、次の手順を実行して証明書の検証を有効にする必要があります。そうしないと、エンドポイントの操作が証明書のエラーで失敗します。詳細については、VMware ナレッジベースの記事「Endpoint communication is broken after upgrade to vRA 7.3 (KB2150230)」(<http://kb.vmware.com/kb/2150230>) および「How to download and install vCenter Server root certificates to avoid Web Browser certificate warnings (KB2108294)」(<http://kb.vmware.com/kb/2108294>) を参照してください。

- 1 アップグレード後または移行後に、vRealize Automation vSphere エージェント マシンにログインし、[サービス] タブを使用して vSphere エージェントを再起動します。

移行ではすべてのエージェントが再起動されない場合があるため、必要に応じて手動で再起動します。

- 2 少なくとも 1 つの ping レポートが終了するのを待ちます。ping レポートの完了には 1、2 分かかります。
- 3 vSphere エージェントがデータ収集を開始したら、IaaS 管理者として vRealize Automation にログインします。
- 4 [インフラストラクチャ] - [エンドポイント] - [エンドポイント] の順にクリックします。
- 5 vSphere エンドポイントを編集し、[接続をテスト] をクリックします。
- 6 証明書のプロンプトが表示されたら、[OK] をクリックして証明書を受け入れます。

証明書のプロンプトが表示されない場合、証明書が現在、プロキシ エージェント マシンや DEM マシンなどのエンドポイントのサービスをホストする Windows マシンの信頼されたルート認証局に正しく保存されていない可能性があります。

- 7 [OK] をクリックして、証明書の承認を適用し、エンドポイントを保存します。
- 8 vSphere エンドポイントごとにこの手順を繰り返します。
- 9 NSX エンドポイントごとにこの手順を繰り返します。

[接続をテスト] 操作は成功したもの、一部のデータ収集やプロビジョニング操作が失敗した場合、エンドポイントとして機能するすべてのエージェント マシンとすべての DEM マシンに同じ証明書をインストールできます。または、既存のマシンから証明書をアンインストールして、問題のあるエンドポイントに対して上記の手順を繰り返します。

## ターゲット vRealize Automation 7.4 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行

移行の後、ターゲット vRealize Automation 7.4 環境で NSX ネットワークおよびセキュリティ インベントリ データ収集を実行する必要があります。

このデータ収集は、vRealize Automation 7.4 で 7.1、7.2、および 7.3 の展開用にロード バランサの再構成アクションを行うために必要です。

---

**注:** このデータ収集は、vRealize Automation 6.2.x から 7.4 に移行した場合は、実行する必要はありません。

---

### 前提条件

- 「ソース vRealize Automation 環境での NSX ネットワークおよびセキュリティ インベントリ データ収集の実行」。
- vRealize Automation 7.4 に正常に移行します。

## 手順

- ◆ vRealize Automation 7.4 に移行する前に、ターゲットの vRealize Automation 環境で NSX ネットワークとセキュリティ インベントリのデータ収集を実行します。『vRealize Automation の管理』の「手動によるエンドポイント データ収集の開始」を参照してください。

## 移行後のロード バランサを高可用性環境に再構成する

高可用性環境に移行する場合、移行の完了後に各ロード バランサに対して次のタスクを実行する必要があります。

### 前提条件

[\[vRealize Automation 7.4 高可用性環境への vRealize Automation ソース データの移行\]](#) .

## 手順

- 1 次の項目についてロード バランサを構成することで、健全性チェック設定を元に戻し、レプリカ ノードが受信トラフィックを受け入れられるようにします。
  - vRealize Automation アプライアンス
  - Model Manager をホストする IaaS Web サーバ
  - Manager Service
- 2 ロード バランサのタイムアウト設定を変更してデフォルトに戻します。

## 外部 Orchestrator サーバから vRealize Automation 7.4 への移行

既存の外部 Orchestrator サーバは、vRealize Automation に組み込まれている vRealize Orchestrator インスタンスに移行することができます。

vRealize Orchestrator を外部サーバ インスタンスとして導入し、その外部インスタンスと連携するように vRealize Automation を構成することができます。または、vRealize Automation アプライアンスに含まれている vRealize Orchestrator サーバを構成して使用することもできます。

VMware では、外部 vRealize Orchestrator を、vRealize Automation に組み込まれた Orchestrator サーバに移行することをお勧めします。外部 Orchestrator から組み込み Orchestrator への移行には、次の利点があります。

- 総所有コストが削減されます。
- デプロイ モデルが簡素化されます。
- 運用効率が向上します。

---

**注:** 外部 vRealize Orchestrator の使用は、次の場合に考慮します。

- vRealize Automation 環境内の複数のテナント
  - 物理的に分散した環境
  - ワークロードの処理
  - 特定のプラグイン（6.5 以前のバージョンの Site Recovery Manager プラグインなど）の使用
-



## Migration Scenarios

The procedure of migrating an external vRealize Orchestrator instance to a vRealize Orchestrator instance embedded in vRealize Automation varies depending on the setup that you have. Several migration scenarios exist based on whether the external Orchestrator server is Windows-based or a virtual appliance, using the embedded database or an external one, and other conditions. You can combine the migration process with an upgrade of vRealize Orchestrator, vRealize Automation, or both. In this case, the migration procedure depends on the source versions of the products.

### Migration Scenario Matrix

You can choose a migration scenario based on the source deployment.

vRealize Orchestrator Deployment	vRealize Automation Deployment	Migration Scenario
vRealize Orchestrator 6.0.3 Virtual Appliance	vRealize Automation 6.2.3	<a href="#">「外部 vRealize Orchestrator 6.x 仮想アプライアンスから vRealize Automation 7.4 への移行」</a>
vRealize Orchestrator 6.0.4 on Windows	vRealize Automation 6.2.4	<a href="#">「Windows 上の外部 vRealize Orchestrator 6.x から vRealize Automation 7.4 への移行」</a>
vRealize Orchestrator 6.0.4 Virtual Appliance	vRealize Automation 6.2.4	<a href="#">「外部 vRealize Orchestrator 6.x 仮想アプライアンスから vRealize Automation 7.4 への移行」</a>
vRealize Orchestrator 6.0.5 Virtual Appliance	vRealize Automation 6.2.5	<a href="#">「外部 vRealize Orchestrator 6.x 仮想アプライアンスから vRealize Automation 7.4 への移行」</a>
vRealize Orchestrator 7.0 Virtual Appliance with an external Oracle Database 12 c	vRealize Automation 7.0 or IaaS	<a href="#">Migrate an External vRealize Orchestrator 7.x to vRealize Automation 7.2</a>
vRealize Orchestrator 7.0.1 Virtual Appliance with an external PostgreSQL 9.3.9 database	vRealize Automation 7.0.1 or IaaS	<a href="#">Migrate an External vRealize Orchestrator 7.x to vRealize Automation 7.2</a>
vRealize Orchestrator 7.1 Virtual Appliance	vRealize Automation 7.1	<a href="#">Migrate an External vRealize Orchestrator 7.x to vRealize Automation 7.2</a>
vRealize Orchestrator 7.2 Virtual Appliance	vRealize Automation 7.2	<a href="#">Migrate an External vRealize Orchestrator 7.x to vRealize Automation 7.2</a>
vRealize Orchestrator 7.3 Virtual Appliance	vRealize Automation 7.3	<a href="#">「外部 vRealize Orchestrator 7.x から vRealize Automation 7.4 への移行」</a>
vRealize Orchestrator 6.0.3 on Windows	vRealize Automation 6.2.3	<a href="#">「Orchestrator 構成を Windows から仮想アプライアンスに移行」</a>

## Orchestrator 構成を Windows から仮想アプライアンスに移行

Orchestrator 5.5.x および 6.x の Windows スタンドアロン構成を Orchestrator Appliance に移行します。

### 前提条件

- 移行先のバージョンの Orchestrator ノードを展開および構成します。『VMware vRealize Orchestrator のインストールおよび構成』の「スタンドアロン Orchestrator サーバの構成」を参照してください。

- 送信元の Orchestrator が SHA1 パッケージ署名証明書を使用している場合は、より強力な署名アルゴリズムを使用する証明書を再生成してください。推奨される署名アルゴリズムは SHA2 です。
- 移行元と移行先の Orchestrator インスタンスの両方で、Orchestrator サーバ サービスを停止します。
- 移行元 Orchestrator サーバのデータベースを、データベース スキーマとともにバックアップします。

**注:** 新しい環境が完全に構成されるまで移行元の Orchestrator 環境を使用する場合は、移行元のデータベースのコピーを作成します。作成しない場合は移行先 Orchestrator を構成して同じデータベースを使用することができますが、その場合は移行元の Orchestrator 環境が機能なくなります。これはデータベース スキーマが移行先 Orchestrator のバージョンにアップグレードされるためです。

## 手順

- 1 移行先の Orchestrator サーバから移行ツールをダウンロードします。
  - a コントロール センターに **root** としてログインします。
  - b [設定をエクスポート/インポート] ページを開いて、[設定をインポート] タブをクリックします。
  - c 移行ツールをページ説明に従ってダウンロードするか、  
[https://<orchestrator\\_server\\_IP\\_or\\_DNS\\_name>:8283/vco-controlcenter/api/server/migration-tool](https://<orchestrator_server_IP_or_DNS_name>:8283/vco-controlcenter/api/server/migration-tool) から直接ダウンロードします。
- 2 移行元 Orchestrator サーバから Orchestrator 構成をエクスポートします。
  - a Orchestrator のインストール フォルダにダウンロードされたアーカイブを抽出します。  
 Windows ベースのインストールの場合、Orchestrator インストール フォルダのデフォルトのパスは **C:\Program Files\VMware\Orchestrator** です。
  - b **PATH** 環境変数は、Orchestrator とともにインストールされる Java JRE の **bin** フォルダをポイントするように設定します。
  - c Windows コマンド プロンプトを使用して、Orchestrator インストール フォルダ下の **bin** フォルダに移動します。  
 デフォルトでは、**bin** フォルダのパスは **C:\Program Files\VMware\Orchestrator\migration-cli\bin** です。
  - d コマンド ラインから **export** コマンドを実行します。

```
C:\Program Files\VMware\Orchestrator\migration-cli\bin\vro-migrate.bat export
```

このコマンドは VMware vRealize Orchestrator 構成ファイルとプラグインを 1 つのエクスポート アーカイブに結合します。

ファイル名が **orchestrator-config-export-orchestrator\_ip\_address-date\_hour.zip** のアーカイブは、**migration-cli** と同じフォルダに作成されます。

### 3 構成を移行先の Orchestrator インスタンスにインポートします。

- a コントロール センターに **root** としてログインします。
- b コントロール センターの [設定をエクスポート/インポート] を開き、[設定をインポート] タブをクリックします。
- c 移行元 Orchestrator インスタンスからエクスポートした **.ZIP** ファイルを参照して、選択します。
- d 構成をエクスポートするときに使用したパスワードを入力します。  
パスワードを使用して構成をエクスポートしていない場合は空白のままにします。
- e インポート タイプを選択します。
- f 構成を外部 Orchestrator サーバにインポートする場合は、データベース設定をインポートするかどうかを選択します。

---

**注:** 移行元および移行先の Orchestrator サーバが同じ外部データベースを使用するように構成されていない場合は、[データベース設定を移行] チェック ボックスをオフにして、データベース スキーマが新しいバージョンにアップグレードされないようにします。オフにしない場合、移行元 Orchestrator 環境の機能が停止します。

移行の前に移行先 Orchestrator を使用するデータベースを構成しておく必要があります。

---

- g [インポート] をクリックして移行を終了します。

構成が正常にインポートされたことがメッセージに示されます。移行先の Orchestrator インスタンスの Orchestrator サーバ サービスが自動的に再起動されます。

### 4 移行先の vRealize Orchestrator で、移行元 Orchestrator で使用されているサーバとは異なる認証プロバイダのサーバを使用している場合は、認証プロバイダで使用するよう構成された SSL 証明書を移行先 Orchestrator のトラスト ストアにインポートします。

- a コントロール センターの [証明書] ページで、[URL からインポート] をクリックします。
- b vRealize Automation または vSphere インスタンスの URL を指定します。

移行が正常に終了したことを示すメッセージが表示されます。Orchestrator サーバ サービスが自動的に再起動されます。

#### 次のステップ

コントロール センターの [設定を検証] ページで、Orchestrator が正しく設定されていることを確認します。

## Windows 上の外部 vRealize Orchestrator 6.x から vRealize Automation 7.4 への移行

vRealize Automation をバージョン 6.x からバージョン 7.4 にアップグレードしたら、Windows 上にインストールされている既存の外部 Orchestrator 6.x を、vRealize Automation 7.4 に組み込まれている Orchestrator サーバに移行できます。

---

**注:** 複数の vRealize Automation アプライアンス ノードがある分散 vRealize Automation 環境では、プライマリの vRealize Automation ノードに対してのみ移行手順を実行します。

---

### 前提条件

- vRealize Automation のバージョンを 7.4 にアップグレードまたは移行します。詳細については、『vRealize Automation のインストールまたはアップグレード』の「vRealize Automation のアップグレード」を参照してください。
- 送信元の Orchestrator が SHA1 パッケージ署名証明書を使用している場合は、より強力な署名アルゴリズムを使用する証明書を再生成してください。推奨される署名アルゴリズムは SHA2 です。
- 外部 Orchestrator の Orchestrator サーバサービスを停止します。
- 外部 Orchestrator サーバのデータベースを、データベース スキーマを含めバックアップします。

### 手順

- 1 移行先の Orchestrator サーバから移行ツールをダウンロードします。
  - a SSH を使用して vRealize Automation アプライアンス に **root** としてログインします。
  - b `/var/lib/vco/downloads` ディレクトリにある **migration-tool.zip** アーカイブをダウンロードします。
- 2 移行元 Orchestrator サーバから Orchestrator 構成をエクスポートします。
  - a **PATH** 環境変数は、Orchestrator とともにインストールされる Java JRE の **bin** フォルダをポイントするように設定します。
  - b 移行ツールを、外部 Orchestrator がインストールされている Windows サーバにアップロードします。
  - c Orchestrator のインストール フォルダにダウンロードされたアーカイブを抽出します。  
Windows ベースのインストールの場合、Orchestrator インストール フォルダのデフォルトのパスは **C:\Program Files\VMware\Orchestrator** です。

- d 管理者として Windows コマンド プロンプトを実行して、Orchestrator インストール フォルダ内の **bin** フォルダに移動します。

デフォルトでは、**bin** フォルダのパスは **C:\Program Files\VMware\Orchestrator\migration-cli\bin** です。

- e コマンド ラインから **export** コマンドを実行します。

```
C:\Program Files\VMware\Orchestrator\migration-cli\bin\vro-migrate.bat export
```

このコマンドは VMware vRealize Orchestrator 構成ファイルとプラグインを 1 つのエクスポート アーカイブに結合します。

アーカイブは **migration-cli** フォルダと同じフォルダ内に作成されます。

- 3 エクスポートした構成を、vRealize Automation 7.4 に組み込まれた Orchestrator サーバに移行します。

- a vRealize Automation アプライアンス で、組み込み vRealize Orchestrator サーバの Orchestrator サーバ サービスとコントロール センター サービスを停止します。

```
service vco-server stop && service vco-configurator stop
```

- b エクスポートした構成ファイルを vRealize Automation アプライアンス の **/usr/lib/vco/tools/configuration-cli/bin** ディレクトリにアップロードします。
- c エクスポートした Orchestrator 構成ファイルの所有権を変更します。

```
chown vco:vco orchestrator-config-export-<orchestrator_ip_address>--<date>_<hour>.zip
```

- d **import** コマンドを使用して **vro-configure** スクリプトを実行し、Orchestrator 構成ファイルを組み込み vRealize Orchestrator サーバにインポートします。

```
./vro-configure.sh import --type embedded --path orchestrator-config-export-<orchestrator_appliance_ip>--<date>_<hour>.zip
```

- e データベースのキーストアからすべての証明書を削除します。

```
./vro-configuration.sh untrust --reset-db
```

- 4 **db-migrate** コマンドを使用して **vro-configure** スクリプトを実行し、データベースを内部 PostgreSQL データベースに移行します。

```
./vro-configure.sh db-migrate --sourceJdbcUrl <JDBC_connection_URL> --sourceDbUsername <database_user> --sourceDbPassword <database_user_password>
```

**注:** 特殊文字を含むパスワードは一重引用符で囲んでください。

<JDBC\_connection\_URL> は、使用するデータベースのタイプによって異なります。

PostgreSQL: jdbc:postgresql://<host>:<port>/<database\_name>

MSSQL: jdbc:jtds:sqlserver://<host>:<port>/<database\_name>; if using SQL authentication and MSSQL:  
jdbc:jtds:sqlserver://<host>:<port>/<database\_name>;domain=<domain>;useNTLMv2=TRUE if using Windows authentication.

Oracle: jdbc:oracle:thin:@<host>:<port>:<database\_name>

デフォルトのデータベースのログイン情報は以下のとおりです。

<database_name>	vmware
<database_user>	vmware
<database_user_password>	vmware

これで、Windows にインストールされている外部 vRealize Orchestrator 6.x が vRealize Automation 7.4 に組み込まれた vRealize Orchestrator インスタンスに正常に移行されました。

#### 次のステップ

組み込み vRealize Orchestrator サーバを設定します。[「組み込み vRealize Orchestrator サーバの構成」](#) を参照してください。

## 外部 vRealize Orchestrator 6.x 仮想アプライアンスから vRealize Automation 7.4 への移行

vRealize Automation をバージョン 6.x からバージョン 7.4 にアップグレードしたら、既存の外部 Orchestrator 6.x 仮想アプライアンスを vRealize Automation 7.4 に組み込まれている Orchestrator サーバに移行できます。

**注:** 複数の vRealize Automation アプライアンス ノードがある分散 vRealize Automation 環境では、プライマリ の vRealize Automation ノードに対してのみ移行手順を実行します。

## 前提条件

- vRealize Automation のバージョンを 7.4 にアップグレードまたは移行します。詳細については、『vRealize Automation のインストールまたはアップグレード』の「vRealize Automation のアップグレード」を参照してください。
- 送信元の Orchestrator が SHA1 パッケージ署名証明書を使用している場合は、より強力な署名アルゴリズムを使用する証明書を再生成してください。推奨される署名アルゴリズムは SHA2 です。
- 外部 Orchestrator の Orchestrator サーバサービスを停止します。
- 外部 Orchestrator サーバのデータベースを、データベース スキーマを含めバックアップします。

## 手順

- 1 移行先の Orchestrator サーバから移行元 Orchestrator に移行ツールをダウンロードします。
  - a SSH を使用して vRealize Orchestrator 6.x 仮想アプライアンスに **root** としてログインします。
  - b `/var/lib/vco` ディレクトリの下で、**scp** コマンドを実行して **migration-tool.zip** アーカイブをダウンロードします。

```
scp root@<vra-va-hostname.domain.name>:/var/lib/vco/downloads/migration-tool.zip ./
```

- c **unzip** コマンドを実行して、移行ツールのアーカイブを抽出します。

```
unzip migration-tool.zip
```

- 2 移行元 Orchestrator サーバから Orchestrator 構成をエクスポートします。
  - a `/var/lib/vco/migration-cli/bin` ディレクトリで、**export** コマンドを実行します。

```
./vro-migrate.sh export
```

このコマンドは VMware vRealize Orchestrator 構成ファイルとプラグインを 1 つのエクスポート アーカイブに結合します。

ファイル名 **orchestrator-config-export-<orchestrator\_ip\_address>-<date>\_<hour>.zip** を使用したアーカイブが `/var/lib/vco` フォルダに作成されます。

- 3 エクスポートした構成を、vRealize Automation 7.4 に組み込まれた Orchestrator サーバに移行します。
  - a SSH を使用して vRealize Automation アプライアンス に **root** としてログインします。
  - b 組み込み vRealize Orchestrator サーバの Orchestrator サーバサービスとコントロール センター サービスを停止します。

```
service vco-server stop && service vco-configurator stop
```

- c `/usr/lib/vco/tools/configuration-cli/bin` ディレクトリの下で、`scp` コマンドを実行して、エクスポートした構成アーカイブをダウンロードします。

```
scp root@<orchestrator_ip_or_dns_name>:/var/lib/vco/orchestrator-config-export-
<orchestrator_ip_address>-<date>_<hour>.zip ./
```

- d エクスポートした Orchestrator 構成ファイルの所有権を変更します。

```
chown vco:vco orchestrator-config-export-<orchestrator_ip_address>-
<date>_<hour>.zip
```

- e `import` コマンドを使用して `vro-configure` スクリプトを実行し、Orchestrator 構成ファイルを組み込み vRealize Orchestrator サーバにインポートします。

```
./vro-configure.sh import --type embedded --path orchestrator-config-export-
<orchestrator_appliance_ip>-<date>_<hour>.zip
```

- 4 移行元とする外部 Orchestrator サーバが組み込み PostgreSQL データベースを使用している場合は、データベース構成ファイルを編集します。

- a `/var/vmware/vpostgres/current/pgdata/postgresql.conf` ファイルで、`listen_addresses` 行をコメント解除します。

- b `listen_addresses` の値をワイルドカード (\*) に設定します。

```
listen_addresses = '*'
```

- c `/var/vmware/vpostgres/current/pgdata/pg_hba.conf` ファイルに行を追加します。

```
host all all <vra-va-ip-address>/32 md5
```

---

**注:** `pg_hba.conf` ファイルでは、IP アドレスとサブネットマスクの代わりに CIDR プリフィックス形式を使用する必要があります。

---

- d PostgreSQL サーバサービスを再起動します。

```
service vpostgres restart
```



- 5 **db-migrate** コマンドを使用して **vro-configure** スクリプトを実行し、データベースを内部 PostgreSQL データベースに移行します。

```
./vro-configure.sh db-migrate --sourceJdbcUrl <JDBC_connection_URL> --sourceDbUsername <database_user> --sourceDbPassword <database_user_password>
```

注: 特殊文字を含むパスワードは一重引用符で囲んでください。

<JDBC\_connection\_URL> は、使用するデータベースのタイプによって異なります。

PostgreSQL: jdbc:postgresql://<host>:<port>/<database\_name>

MSSQL: jdbc:jtds:sqlserver://<host>:<port>/<database\_name>; if using SQL authentication and MSSQL:  
jdbc:jtds:sqlserver://<host>:<port>/<database\_name>;domain=<domain>;useNTLMv2=TRUE if using Windows authentication.

Oracle: jdbc:oracle:thin:@<host>:<port>:<database\_name>

デフォルトのデータベースのログイン情報は以下のとおりです。

<database_name>	vmware
<database_user>	vmware
<database_user_password>	vmware

- 6 データベースのキーストアからすべての証明書を削除します。

```
./vro-configuration.sh untrust --reset-db
```

- 7 Orchestrator プラグインを再インストールします。
- コントロール センターに **root** としてログインします。
  - [トラブルシューティング] をクリックします。
  - [プラグインを強制的に再インストール] をクリックします。

- 8 Orchestrator サーバ サービスを開始します。

- 9 **postgresql.conf** および **pg\_hba.conf** ファイルをデフォルトの構成に戻します。

- PostgreSQL サーバ サービスを再起動します。

これで、外部 vRealize Orchestrator 6.x 仮想アプライアンスが vRealize Automation 7.4 に組み込まれた vRealize Orchestrator インスタンスに正常に移行されました。

## 次のステップ

組み込み vRealize Orchestrator サーバを設定します。[「組み込み vRealize Orchestrator サーバの構成」](#)を参照してください。

## 外部 vRealize Orchestrator 7.x から vRealize Automation 7.4 への移行

既存の外部 Orchestrator インスタンスから構成をエクスポートし、これを vRealize Automation に組み込まれている Orchestrator サーバにインポートすることができます。

---

**注:** 複数の vRealize Automation アプライアンス ノードがある場合は、プライマリの vRealize Automation ノードに対してのみ移行手順を実行します。

---

### 前提条件

- vRealize Automation のバージョンを 7.4 にアップグレードまたは移行します。詳細については、『vRealize Automation のインストールまたはアップグレード』の「vRealize Automation のアップグレード」を参照してください。
- 外部 Orchestrator の Orchestrator サーバサービスを停止します。
- 外部 Orchestrator サーバのデータベースを、データベース スキーマを含めバックアップします。

### 手順

- 1 外部 Orchestrator サーバから構成をエクスポートします。
  - a 移行元のバージョンに応じて、**root** または **管理者** として外部 Orchestrator サーバのコントロール センターにログインします。
  - b [起動オプション] ページから Orchestrator サーバサービスを停止して、データベースに不要な変更が加えられないようにします。
  - c [設定のエクスポート/インポート] ページに移動します。
  - d [設定をエクスポート] ページで、[サーバ設定をエクスポート]、[バンドル プラグイン]、[プラグイン設定をエクスポート] を選択します。
- 2 エクスポートした設定を組み込み Orchestrator インスタンスに移行します。
  - a エクスポートした Orchestrator 構成ファイルを vRealize Automation アプライアンスの `/usr/lib/vco/tools/configuration-cli/bin` ディレクトリにアップロードします。
  - b SSH を使用して vRealize Automation アプライアンス に **root** としてログインします。

- c 組み込み vRealize Orchestrator サーバの Orchestrator サーバ サービスとコントロール センター サービスを停止します。

```
service vco-server stop && service vco-configurator stop
```

- d **import** コマンドを使用して **vro-configure** スクリプトを実行し、Orchestrator 構成ファイルを組み込み vRealize Orchestrator サーバにインポートします。

```
./vro-configure.sh import --type embedded --path orchestrator-config-export-  
<orchestrator_appliance_ip>--<date>_<hour>.zip
```

- 3 移行元とする外部 Orchestrator サーバが組み込み PostgreSQL データベースを使用している場合は、データベース構成ファイルを編集します。

- a `/var/vmware/vpostgres/current/pgdata/postgresql.conf` ファイルで、`listen_addresses` 行をコメント解除します。

- b `listen_addresses` の値をワイルドカード (\*) に設定します。

```
listen_addresses = '*'
```

- c `/var/vmware/vpostgres/current/pgdata/pg_hba.conf` ファイルに行を追加します。

```
host all all <vra-va-ip-address>/32 md5
```

---

**注:** `pg_hba.conf` ファイルでは、IP アドレスとサブネット マスクの代わりに CIDR プリフィックス形式を使用する必要があります。

---

- d PostgreSQL サーバ サービスを再起動します。

```
service vpostgres restart
```

- 4 **db-migrate** コマンドを使用して **vro-configure** スクリプトを実行し、データベースを内部 PostgreSQL データベースに移行します。

```
./vro-configure.sh db-migrate --sourceJdbcUrl <JDBC_connection_URL> --sourceDbUsername <database_user> --sourceDbPassword <database_user_password>
```

**注:** 特殊文字を含むパスワードは一重引用符で囲んでください。

<JDBC\_connection\_URL> は、使用するデータベースのタイプによって異なります。

PostgreSQL: jdbc:postgresql://<host>:<port>/<database\_name>

MSSQL: jdbc:jtds:sqlserver://<host>:<port>/<database\_name>; if using SQL authentication and MSSQL:  
jdbc:jtds:sqlserver://<host>:<port>/<database\_name>;domain=<domain>;useNTLMv2=TRUE if using Windows authentication.

Oracle: jdbc:oracle:thin:@<host>:<port>:<database\_name>

デフォルトのデータベースのログイン情報は以下のとおりです。

<database_name>	vmware
<database_user>	vmware
<database_user_password>	vmware

- 5 データベースのキーストアからすべての証明書を削除します。

```
./vro-configuration.sh untrust --reset-db
```

- 6 Orchestrator プラグインを再インストールします。
- コントロール センターに **root** としてログインします。
  - [トラブルシューティング] をクリックします。
  - [プラグインを強制的に再インストール] をクリックします。

- 7 Orchestrator サーバ サービスを開始します。

- 8 **postgresql.conf** および **pg\_hba.conf** ファイルをデフォルトの構成に戻します。

- PostgreSQL サーバ サービスを再起動します。

外部 Orchestrator サーバ インスタンスが vRealize Automation に組み込まれている vRealize Orchestrator インスタンスに正常に移行されました。

## 次のステップ

組み込み vRealize Orchestrator サーバを設定します。「[組み込み vRealize Orchestrator サーバの構成](#)」を参照してください。

## 組み込み vRealize Orchestrator サーバの構成

外部の vRealize Orchestrator 構成をエクスポートし、vRealize Automation にインポートしてから、vRealize Automation に組み込まれた vRealize Orchestrator サーバを構成します。

### 前提条件

設定を外部から内部 vRealize Orchestrator に移行します。

### 手順

- 1 vRealize Automation アプライアンスのコマンド プロンプト セッションに root としてログインします。
- 2 vRealize Orchestrator コントロール センターのサービスおよびサーバを起動します。

```
service vco-configurator start && service vco-server start
```

- 3 組み込みの vRealize Orchestrator コントロール センターに root としてログインします。

<https://<vrealize-automation-appliance-FQDN>:8283/vco-controlcenter/config>

**注:** 外部および内部の vRealize Orchestrator バージョンが同じである場合、次の手順を省略することができます。

- 4 コントロール センターで、[設定を検証] をクリックし、vRealize Orchestrator が適切に設定されていることを確認します。
- 5 コントロール センターで、[証明書]、[パッケージ署名証明書] の順にクリックして、新しいパッケージ署名証明書を生成します。
- 6 コントロール センターで、[認証プロバイダを設定] をクリックします。  
[デフォルト テナント] と [管理グループ] は、デフォルト値 **vsphere.local** および **vsphere.local\vcoadmins** に設定されています。デフォルトの値を環境に応じて変更します。
- 7 vRealize Automation アプライアンスの管理インターフェイスの、[サービス] で、**vco-server** が登録済みであることを確認します。
- 8 外部 vRealize Orchestrator サーバの **vco** サービスを選択し、[登録解除] をクリックします。

## 次のステップ

- 外部 vRealize Orchestrator サーバにある信頼された証明書を、組み込み vRealize Orchestrator のトラストストアにインポートします。詳細については、『VMware vRealize Orchestrator のインストールおよび構成』の「Orchestrator の証明書の管理」を参照してください。
- vRealize Automation レプリカ ノードを vRealize Automation クラスタに参加させて、vRealize Orchestrator 構成を同期します。

詳細については、『vRealize Automation のインストールまたはアップグレード』の「Reconfigure the Target Embedded vRealize Orchestrator to Support High Availability」を参照してください。

---

**注:** vRealize Orchestrator のインスタンスは自動的にクラスタ化されており、使用可能です。

---

- クラスタ内のすべてのノードで **vco-configurator** サービスを再起動します。
- 移行した組み込み vRealize Orchestrator サーバをポイントするように、vRealize Orchestrator エンドポイントを更新します。
- vRealize Automation ホストと IaaS ホストを vRealize Automation プラグインのインベントリに追加します。これは vRA ホスト ワークフローの vRA ホストの追加と IaaS ホストの追加を実行して行います。

## vRealize Automation 証明書を信頼するように組み込みの vRealize Orchestrator を更新する

vRealize Automation アプライアンス 証明書または IaaS 証明書を更新または変更する場合は、vRealize Orchestrator を更新して、新しい、または更新された証明書を信頼する必要があります。

この手順は、組み込みの vRealize Orchestrator インスタンスを使用しているすべての vRealize Automation 展開に適用されます。外部 vRealize Orchestrator インスタンスを使用する場合は、「[Update External vRealize Orchestrator to Trust vRealize Automation Certificates](#)」を参照してください。

---

**注:** この手順では、テナントとグループの認証がデフォルトの設定にリセットされます。認証の設定がカスタマイズされている場合は、手順の完了後に認証を再設定できるよう、変更内容をメモしてください。

---

vRealize Orchestrator 証明書の更新および置き換えについては、vRealize Orchestrator のドキュメントを参照してください。

この手順を完了せずに vRealize Automation 証明書の置き換えまたは更新を行うと、vRealize Orchestrator コントロール センターにアクセスできなくなり、vco-server と vco-configurator のログ ファイルにエラーが表示される場合があります。

vRealize Automation とは異なるテナントとグループに対して認証するように vRealize Orchestrator が設定されている場合にも、証明書を更新する際に問題が発生することがあります。<https://kb.vmware.com/kb/2147612> を参照してください。

### 手順

- 1 vRealize Orchestrator サーバとコントロール センター サービスを停止します。

```
service vco-server stop
service vco-configurator stop
```

- 2 vRealize Orchestrator 認証プロバイダをリセットします。
  - a `/var/lib/vco/tools/configuration-cli/bin/vro-configure.sh reset-authentication` コマンドを実行します。
  - b `/etc/vco/app-server/vco-registration-id` を削除します。
  - c `vcac-vami vco-service-reconfigure` を実行します。
- 3 vRealize Orchestrator サーバとコントロール センター サービスを起動します。

```
service vco-server start
service vco-configurator start
```

## 外部 Orchestrator および組み込み Orchestrator のコントロール センターの違い

外部 vRealize Orchestrator のコントロール センターで使用可能な一部のメニュー項目は、組み込み Orchestrator インスタンスのデフォルトのコントロール センター ビューに含まれていません。

組み込み Orchestrator サーバのコントロール センターでは、いくつかのオプションがデフォルトで非表示になっています。

メニュー項目	詳細
[ライセンス]	組み込み Orchestrator では vRealize Automation をライセンス プロバイダとして使用するよう事前構成されています。
[設定をエクスポート/インポート]	エクスポートされた vRealize Automation コンポーネントに、組み込み Orchestrator の構成が含まれています。
[データベースを構成]	組み込み Orchestrator では、vRealize Automation で使用されているデータベースを使用します。
[カスタム エクスペリエンス改善プログラム]	カスタマー エクスペリエンス向上プログラム (CEIP) には vRealize Automation アプライアンス管理インターフェイスから参加できます。 『vRealize Automation の管理』の「カスタム エクスペリエンス改善プログラム」を参照してください。

デフォルトのコントロール センター ビューでは非表示になっている別のオプションには、[認証プロバイダを設定] ページの [ホスト アドレス] テキスト ボックスや [登録解除] ボタンがあります。

**注:** vRealize Automation に組み込まれている vRealize Orchestrator でコントロール センターのオプションをすべて表示するには、Orchestrator 管理の詳細ページ ([https://<vra-vahostname.domain.name\\_or\\_load\\_balancer\\_address>:8283/vco-controlcenter/#/?advanced](https://<vra-vahostname.domain.name_or_load_balancer_address>:8283/vco-controlcenter/#/?advanced)) にアクセスし、キーボードの [F5] ボタンを押してページを更新する必要があります。

## ターゲットの vRealize Orchestrator の vRealize Automation エンドポイントの再構成

次の手順を使用して、組み込みのターゲット vRealize Orchestrator の vRealize Automation エンドポイントを再構成します。

## 前提条件

- vRealize Automation の最新バージョンへの正常な移行。
- vRealize Orchestrator クライアントを使用して、ターゲットの vRealize Orchestrator に接続します。詳細については、vRealize Orchestrator のドキュメントで、「VMware vRealize Orchestrator クライアントの使用」を参照してください。

## 手順

- 1 上部のドロップダウン メニューから [設計] を選択します。
- 2 [インベントリ] をクリックします。
- 3 [vRealize Automation] を展開します。
- 4 最小環境から移行した場合は、移行元 vRealize Automation アプライアンス ホストの完全修飾ドメイン名 (FQDN) を含むエンドポイントを識別します。高可用性環境から移行した場合は、移行元アプライアンスのロード バランサの FQDN を含むエンドポイントを識別します。

FQDN を含むエンドポイントを検索する場合は、次の手順を実行します。	FQDN を含むエンドポイントが見つからない場合は、次の手順を実行します。
<ol style="list-style-type: none"> <li>1 [ワークフロー] をクリックします。</li> <li>2 展開ボタンをクリックし、[ライブラリ] - [vRealize Automation] - [構成] の順に選択します。</li> <li>3 次のいずれかを実行します。 <ul style="list-style-type: none"> <li>■ 最小環境から移行した場合は、移行元 vRealize Automation アプライアンス ホストの FQDN を含むすべてのエンドポイントで [vRA ホストの削除] ワークフローを実行します。</li> <li>■ 高可用性環境から移行した場合は、移行元アプライアンスのロード バランサの FQDN を含むすべてのエンドポイントで [vRA ホストの削除] ワークフローを実行します。</li> </ul> </li> </ol>	<ol style="list-style-type: none"> <li>1 [リソース] をクリックします。</li> <li>2 上部のツールバーの更新アイコンをクリックします。</li> <li>3 展開ボタンをクリックし、[ライブラリ] - [vCACCAFE] - [構成] の順に選択します。</li> <li>4 次のいずれかを実行します。 <ul style="list-style-type: none"> <li>■ 最小環境から移行した場合は、移行元 vRealize Automation アプライアンス ホストの FQDN を含む URL プロパティを持つ各リソースを削除します。</li> <li>■ 高可用性環境から移行した場合は、移行元 vRealize Automation アプライアンスのロード バランサの FQDN を含む URL プロパティを持つ各リソースを削除します。</li> </ul> </li> </ol>

- 5 [ワークフロー] をクリックします。
- 6 展開ボタンをクリックし、[ライブラリ] - [vRealize Automation] - [構成] の順に選択します。
- 7 ターゲットの vRealize Automation のアプライアンス ホスト（高可用性展開から移行した場合は、ロード バランシングされたホスト）を追加するには、[コンポーネント レジストリを使用して vRA ホストを追加] ワークフローを実行します。

## ターゲットの vRealize Orchestrator の vRealize Automation インフラストラクチャ エンドポイントの再構成

次の手順を使用して、組み込みのターゲット vRealize Orchestrator の vRealize Automation インフラストラクチャ エンドポイントを再構成します。

## 前提条件

- vRealize Automation の最新バージョンへの正常な移行。



- vRealize Orchestrator クライアントを使用して、ターゲットの vRealize Orchestrator に接続します。詳細については、vRealize Orchestrator のドキュメントで、「VMware vRealize Orchestrator クライアントの使用」を参照してください。

#### 手順

- 1 上部のドロップダウン メニューから [設計] を選択します。
- 2 [インベントリ] をクリックします。
- 3 [vRealize Automation Infrastructure] を展開します。
- 4 最小環境から移行した場合は、移行元 vRealize Automation インフラストラクチャ ホストの完全修飾ドメイン名 (FQDN) を含むエンドポイントを識別します。高可用性環境から移行した場合は、移行元アプライアンスのロード バランサの FQDN を含むエンドポイントを識別します。

FQDN を含むエンドポイントを検索する場合は、次の手順を実行します。	FQDN を含むエンドポイントが見つからない場合は、次の手順を実行します。
<ol style="list-style-type: none"> <li>1 [ワークフロー] をクリックします。</li> <li>2 展開ボタンをクリックし、[ライブラリ] - [vRealize Automation] - [インフラストラクチャ管理] - [構成] の順に選択します。</li> <li>3 次のいずれかを実行します。 <ul style="list-style-type: none"> <li>■ 最小環境から移行した場合は、移行元 vRealize Automation インフラストラクチャ ホストの FQDN を含むすべてのエンドポイントで [IaaS ホストの削除] ワークフローを実行します。</li> <li>■ 高可用性環境から移行した場合は、移行元 vRealize Automation インフラストラクチャ ホストのロード バランサの FQDN を含むすべてのエンドポイントで [IaaS ホストの削除] ワークフローを実行します。</li> </ul> </li> </ol>	<ol style="list-style-type: none"> <li>1 [リソース] をクリックします。</li> <li>2 上部のツールバーの更新アイコンをクリックします。</li> <li>3 展開ボタンをクリックし、[ライブラリ] - [vCAC] - [構成] の順に選択します。</li> <li>4 次のいずれかを実行します。 <ul style="list-style-type: none"> <li>■ 最小環境から移行した場合は、移行元 vRealize Automation インフラストラクチャ ホストの FQDN を含む <b>host</b> プロパティを持つ各リソースを削除します。</li> <li>■ 高可用性環境から移行した場合は、移行元 vRealize Automation インフラストラクチャ ホストのロード バランサの FQDN を含む <b>host</b> プロパティを持つ各リソースを削除します。</li> </ul> </li> </ol>

- 5 [ワークフロー] をクリックします。
- 6 展開ボタンをクリックし、[ライブラリ] - [vRealize Automation] - [構成] の順に選択します。
- 7 ターゲットの vRealize Automation のインフラストラクチャ ホスト（高可用性展開から移行した場合は、ロード バランシングされたホスト）を追加するには、[vRA ホストの IaaS ホストを追加] ワークフローを実行します。

## vRealize Orchestrator カスタマイズのインストール

ワークフローを実行して、カスタマイズされた状態変更ワークフロー スタブと vRealize Orchestrator メニュー操作ワークフローをインストールできます。

詳細については、ライフ サイクルの拡張性の「vRealize Orchestrator カスタマイズのインストール」を参照してください。

#### 前提条件

vRealize Automation の最新バージョンへの正常な移行。

## ターゲット vRealize Automation に組み込まれた vRealize Orchestrator インフラストラクチャ エンドポイントの再構成

vRealize Automation 6.2.x 環境から移行する場合は、ターゲットの組み込み vRealize Orchestrator サーバを指しているインフラストラクチャ エンドポイントの URL を更新する必要があります。

### 前提条件

- vRealize Automation 7.4 に正常に移行します。
- ターゲットの vRealize Automation コンソールにログインします。
  - a ターゲット仮想アプライアンスの完全修飾ドメイン名 `https://<vra-vd-hostname.domain.name>/vcac` を使用して vRealize Automation コンソールを開きます。  
  
高可用性環境の場合は、ターゲット仮想アプライアンス ロード バランサの完全修飾ドメイン名 `https://<vra-vd-lb-hostname.domain.name>/vcac` を使用してコンソールを開きます。
  - b Infrastructure as a Service (IaaS) 管理者ユーザーとしてログインします。

### 手順

- 1 [インフラストラクチャ] - [エンドポイント] - [エンドポイント] の順に選択します。
- 2 [エンドポイント] ページで、vRealize Orchestrator エンドポイントを選択し、[編集] をクリックします。
- 3 [アドレス] テキスト ボックスで、vRealize Orchestrator エンドポイントの URL を編集します。
  - 最小環境に移行した場合は、vRealize Orchestrator エンドポイント URL を `https://<vra-vd-hostname.domain.name>:443/vco` に置き換えます。
  - 高可用性環境に移行した場合は、vRealize Orchestrator エンドポイント URL を `https://<vra-vd-lb-hostname.domain.name>:443/vco` に置き換えます。
- 4 [OK] をクリックします。
- 5 vRealize Orchestrator エンドポイントでデータ収集を手動で実行します。
  - a [エンドポイント] ページで、vRealize Orchestrator エンドポイントを選択します。
  - b [アクション] - [データ収集] の順に選択します。  
  
データ収集が成功したことを確認します。

## ターゲット vRealize Automation 環境の Azure エンドポイントの再構成

移行後に、Microsoft Azure エンドポイントを再構成する必要があります。

各 Azure エンドポイントで次の手順を実行します。

### 前提条件

- 最新バージョンの vRealize Automation 7.4 に正常に移行します。

- ターゲットの vRealize Automation コンソールにログインします。
  - a ターゲット仮想アプライアンスの完全修飾ドメイン名 `https://<vra-vb-hostname.domain.name>/vcac` を使用して vRealize Automation コンソールを開きます。

高可用性環境の場合は、ターゲット仮想アプライアンス ロード バランサの完全修飾ドメイン名 `https://<vra-vb-hostname.domain.name>/vcac` を使用してコンソールを開きます。
  - b Infrastructure as a Service (IaaS) 管理者ユーザーとしてログインします。

#### 手順

- 1 [管理] - [vRO 構成] - [エンドポイント] を選択します。
- 2 Azure エンドポイントを選択します。
- 3 [編集] をクリックします。
- 4 [詳細] をクリックします。
- 5 [クライアントシークレット] テキスト ボックスで、元のクライアントシークレットを入力します。
- 6 [完了] をクリックします。
- 7 Azure エンドポイントごとにこの手順を繰り返します。

## vRealize Automation 6.2.x Automation Application Services から 7.4 への移行

VMware vRealize Application Services の移行ツールを使用して、既存のアプリケーション サービスのブループリントと展開プロファイルを VMware vRealize Application Services 6.2.x から vRealize Automation 7.4 に移行できます。

#### 前提条件

vRealize Automation の最新バージョンへの正常な移行。

#### 手順

- ◆ VMware vRealize Application Services の移行ツールをダウンロードするには、次の手順を完了します。
  - a [ダウンロード VMware vRealize Automation](#) をクリックします。
  - b [ドライバとツール] - [VMware vRealize Application Services Migration Tool] を選択します。

## 元のターゲット vRealize Automation IaaS Microsoft SQL データベースの削除

移行の完了後、元の IaaS データベースを削除できます。

## 前提条件

vRealize Automation の最新バージョンへの正常な移行。

移行後の環境では、ターゲット vRealize Automation 環境をインストールしたときに作成した元の vRealize Automation IaaS Microsoft SQL データベースは使用しません。移行が完了したら、元の IaaS データベースを Microsoft SQL Server から削除しても問題ありません。

## 移行後にデータセンターの [場所] メニューの内容を更新する

移行後、欠落しているカスタム データセンターの場所があれば、[場所] ドロップダウン メニューに追加する必要があります。

vRealize Automation の最新バージョンへの移行後、[コンピュー ト リソース] ページの [場所] ドロップダウン メニュー内のデータセンターの場所は、デフォルトのリストに戻ります。カスタム データセンターの場所が欠落していますが、すべてのコンピュー ト リソースの構成が正常に移行され、**Vrm.DataCenter.Location** プロパティは影響を受けていません。[場所] メニューにカスタム データセンターの場所を追加することができます。

## 前提条件

vRealize Automation の最新バージョンに移行します。

## 手順

- ◆ [場所] ドロップダウン メニューに欠落しているデータセンターの場所を追加します。『vRealize Automation の構成』の「シナリオ：複数の拠点にまたがる導入環境向けにデータセンターの場所を追加する」を参照してください。

## ソフトウェア エージェントの TLS 1.2 へのアップグレード

vRealize Automation 7.1、7.2、7.3、または 7.3.1 を 7.4 に移行した後、ソフトウェア エージェントを移行前の環境からトランスポート レイヤー セキュリティ (TLS) 1.2 にアップグレードするには、いくつかのタスクを実行する必要があります。

vRealize Automation 7.4 以降、TLS 1.2 は、vRealize Automation とブラウザの間のデータ通信で唯一サポートされる TLS プロトコルです。移行後、既存のすべての仮想マシンと、移行前の vRealize Automation 7.1 または 7.3 環境からの既存の仮想マシン テンプレートをアップグレードする必要があります。

## ソース環境の仮想マシン テンプレートの更新

ソフトウェア エージェントが TLS 1.2 プロトコルを使用できるように 7.4 への移行が完了した後、既存の vRealize Automation 7.1、7.2、7.3 または 7.3.1 のテンプレートを更新する必要があります。

ゲスト エージェントおよびエージェント ブートストラップ コードを、ソース環境のテンプレートで更新する必要があります。リンク クローン オプションを使用している場合、新規作成した仮想マシンおよびそれらのスナップショットを使用したテンプレートの再マッピングが必要になることがあります。

テンプレートをアップグレードするには、次のタスクを実行します。

- 1 vSphere にログインします。

- 2 vRealize Automation 7.1、7.2、7.3 または 7.3.1 の各テンプレートを仮想マシンに変換し、そのマシンをオンにします。
- 3 適切なソフトウェアのインストーラをインポートし、各仮想マシンでソフトウェアのインストーラを実行します。
- 4 各仮想マシンを再度テンプレートに変換します。

この手順を使用すると、Linux または Windows 用のソフトウェア インストーラを特定できます。

#### 前提条件

- 「ソフトウェア エージェント パッチの適用」 (vRealize Automation 7.1 または 7.3 から 7.4 に移行した場合)。
- vRealize Automation 7.1、7.2、7.3 または 7.3.1 から 7.4 への正常な移行。

#### 手順

- 1 ブラウザを起動し、仮想アプライアンスの完全修飾ドメイン名 (<https://<vra-virtual-machine-hostname.domain.name>>) を使用して、vRealize Automation 7.4 アプライアンスのスプラッシュ ページを開きます。
- 2 [ゲストおよびソフトウェア エージェント] ページをクリックします。
- 3 Linux または Windows ソフトウェアのインストーラの手順を実行してください。

#### 次のステップ

[「ソフトウェア エージェントのアップグレードが必要な仮想マシンの特定」](#)。

## ソフトウェア エージェントのアップグレードが必要な仮想マシンの特定

vRealize Automation コンソールの健全性サービスを使用して、TLS 1.2 へのソフトウェア エージェントの更新が必要な仮想マシンを特定できます。

vRealize Automation ソース環境にパッチを適用したときに、すべての仮想マシンがアップグレードされないことがあります。健全性サービスを使用して、TLS 1.2 へのソフトウェア エージェントの更新が必要な仮想マシンを特定できます。プロビジョニング後の手順のために、ターゲット環境のすべてのソフトウェア エージェントを更新する必要があります。

#### 前提条件

- 「ソフトウェア エージェント パッチの適用」 (vRealize Automation 7.1 または 7.3 から 7.4 に移行した場合)。
- vRealize Automation 7.1、7.2、7.3 または 7.3.1 から 7.4 に正常に移行している。
- プライマリ仮想アプライアンスで vRealize Automation 7.4 にログインしている。

#### 手順

- 1 [管理] - [健全性] の順にクリックします。
- 2 [新しい構成] をクリックします。

## 3 [構成の詳細] ページで、必要な情報を提供します。

オプション	コメント
名前	<b>SW Agent verification</b> と入力します。
説明	<b>Locate software agents for upgrade to TLS 1.2</b> などの、オプションの説明を追加します。
製品	vRealize Automation 7.4.0 を選択します。
スケジュール	[なし] を選択します。

## 4 [次へ] をクリックします。

## 5 [テストスイートの選択] ページで、[vRealize Automation のシステム テスト] と [vRealize Automation のテナント テスト] を選択します。

## 6 [次へ] をクリックします。

## 7 [パラメータの構成] ページで、必要な情報を提供します。

表 6-1. vRealize Automation 仮想アプライアンス

オプション	説明
公開 Web サーバのアドレス	<ul style="list-style-type: none"> <li>■ 最小インストールの場合、vRealize Automation アプライアンス ホストのベース URL。例：https://&lt;va-host.domain&gt;/</li> <li>■ 高可用性展開の場合、vRealize Automation ロード バランサのベース URL。例：https://&lt;load-balancer-host.domain&gt;/</li> </ul>
SSH コンソールのアドレス	vRealize Automation アプライアンスの完全修飾ドメイン名。例：<va-host.domain>
SSH コンソール ユーザー	<b>root</b>
SSH コンソール パスワード	root のパスワード。
サービスの応答の最大時間（ミリ秒）	デフォルトを受け入れ：2000

表 6-2. vRealize Automation システム テナント

オプション	説明
システム テナント管理者	管理者
システム テナント パスワード	管理者のパスワード。

表 6-3. vRealize Automation ディスク容量の監視

オプション	説明
警告レベルしきい値の割合	デフォルトを受け入れ：75
重大レベルしきい値の割合	デフォルトを受け入れ：90

表 6-4. vRealize Automation テナント

オプション	説明
テスト対象テナント	テスト対象として選択されたテナント。
ファブリック管理者のユーザー名	ファブリック管理者のユーザー名。たとえば、admin@va-host.local。  <small>注: すべてのテストを実行するために、このファブリック管理者にはテナント管理者と laaS 管理者のロールも必要です。</small>
ファブリック管理者パスワード	ファブリック管理者のパスワード。

- 8 [次へ] をクリックします。
- 9 [サマリ] ページで情報を確認し、[完了] をクリックします。  
ソフトウェア エージェントの検証設定が完了しました。
- 10 SW Agent verification カードで、[実行] をクリックします。
- 11 テストが完了したら、SW Agent verification カードの中央をクリックします。
- 12 SW Agent verification の結果ページで、テスト結果を参照して、[名前] 列の [ソフトウェア エージェントのバージョン チェック] テストを見つけます。テスト結果が失敗の場合は、[原因] 列の [原因] リンクをクリックして、ソフトウェア エージェントが古い仮想マシンを表示します。

#### 次のステップ

ソフトウェア エージェントが古い仮想マシンがある場合は、[vSphere 上のソフトウェア エージェントのアップグレード](#) を参照してください。

## vSphere 上のソフトウェア エージェントのアップグレード

vRealize Automation アプライアンス管理を使用して、移行後に vSphere の古いソフトウェア エージェントを TLS 1.2 にアップグレードできます。

この手順により、仮想マシンの古いソフトウェア エージェントをソース環境から TLS 1.2 にアップデートします。これは vRealize Automation 7.4 への移行に必要です。

#### 前提条件

- [「ソフトウェア エージェント パッチの適用」](#) (vRealize Automation 7.1 または 7.3 から 7.4 に移行した場合)。
- vRealize Automation 7.1、7.2、7.3 または 7.3.1 から 7.4 への正常な移行。
- 健全性サービスを使用して、古いソフトウェア エージェントを持つ仮想アプライアンスを識別している。

#### 手順

- 1 プライマリ vRealize Automation アプライアンスで、vRealize Automation アプライアンスを展開したときに入力したパスワードを使用して、**root** として vRealize Automation アプライアンス管理にログインします。  
高可用性環境の場合、マスター アプライアンスでアプライアンス管理を開きます。
- 2 [vRA 設定] - [ソフトウェア エージェント] の順にクリックします。

- 3 [Toggle TLS 1.0、1.1] をクリックします。

[TLS v1.0, v1.1 Status] が有効になります。

- 4 テナント認証情報には、ソース vRealize Automation アプライアンスに要求される情報を入力します。

オプション	説明
テナント名	ソース vRealize Automation アプライアンス上のテナントの名前。  <small>注: テナントユーザーにはソフトウェア アーキテクト ロールが割り当てられている必要があります。</small>
ユーザー名	ソース vRealize Automation アプライアンス上のテナント管理者のユーザー名。
パスワード	テナント管理者のパスワード。

- 5 [接続をテスト] をクリックします。

接続が確立されると、成功のメッセージが表示されます。

- 6 ソース アプライアンスには、ソース vRealize Automation アプライアンスの IP アドレスまたは完全修飾ドメイン名を入力します。

ソースとターゲット アプライアンス両方が同一のテナント認証情報を使用する必要があります。

- 7 [バッチの一覧表示] をクリックします。

バッチ選択肢リスト テーブルが表示されます。

- 8 [表示] をクリックします。

古いソフトウェア エージェントを持つ仮想マシンのリストがテーブルに表示されます。

- 9 アップグレード可能状態の仮想マシンのソフトウェア エージェントをアップグレードします。

- 個々の仮想マシンのソフトウェア エージェントをアップグレードするには、仮想マシンのグループの [表示] をクリックして、アップグレードする仮想マシンを識別します。それから [実行] をクリックし、アップグレード プロセスを開始します。
- 仮想マシンのバッチのソフトウェア エージェントをアップグレードするには、アップグレードするグループを識別し、[実行] をクリックしてアップグレード プロセスを開始します。

アップグレードする仮想マシンが 200 台以上ある場合は、次のパラメータの値を入力して、バッチ アップグレード プロセスの速度を制御できます。

オプション	説明
バッチ サイズ	バッチ アップグレードで選択した仮想マシンの数。この数を変えると、アップグレードの速度を調整することができます。
キュー深度	同時に実行されるアップグレードの平行実行の数。たとえば、20 と指定します。この数を変えると、アップグレードの速度を調整することができます。



オプション	説明
バッチ エラー	バッチ アップグレードの低速化の原因となった REST エラーの数。たとえば、アップグレードの安定性を向上するために 5 回エラーが発生したときに現在のバッチ アップグレードを停止する場合は、テキスト フィールドに 5 と入力します。
バッチの失敗	バッチ アップグレードの低速化の原因となった、失敗したソフトウェア エージェント アップグレードの数。たとえば、アップグレードの安定性を向上するために 5 回エラーが発生したときに現在のバッチ アップグレードを停止する場合は、テキスト フィールドに 5 と入力します。
バッチ ポーリング	アップグレード プロセスをチェックするためにアップグレード プロセスをポーリングする頻度。この数を変えると、アップグレードの速度を調整することができます。

アップグレード プロセスが低速すぎるか、失敗したアップグレードが多すぎる場合は、これらのパラメータを調整してアップグレードのパフォーマンスを向上できます。

**注:** [更新] をクリックすると、バッチのリストがクリアされます。アップグレード プロセスには影響しません。TLS 1.2 が設定されるかどうかに関する情報も更新されます。また、[更新] をクリックすると vRealize Automation サービスの健全性チェックも実行されます。サービスが実行されていない場合、エラー メッセージが表示され、他のすべてのアクション ボタンが無効になります。

10 [Toggle TLS 1.0、1.1] をクリックします。

[TLS v1.0, v1.1 Status] が無効になります。

## Amazon Web Service または Azure 上のソフトウェア エージェントのアップグレード

Amazon Web Service (AWS) または Azure 上の古いソフトウェア エージェントは手動でアップグレードすることができます。

- 移行した vRealize Automation サーバの予約で指定されたトンネル プロパティを更新する必要があります。

### 前提条件

- 「[ソフトウェア エージェント パッチの適用](#)」(vRealize Automation 7.1 または 7.3 から 7.4 に移行した場合)。
- vRealize Automation 7.1、7.2、7.3 または 7.3.1 から 7.4 への正常な移行。
- ソフトウェア トンネルが存在し、トンネル仮想マシンの IP アドレスが既知である。

### 手順

- アップグレードする必要がある各ノードのノード ファイルを作成します。

```
/usr/lib/vcac/server/webapps/ROOT/software/initializeUpdateSoftwareAgents.py -a <
$DestinationVRAServer> -t <$Tenant> -tu <$TenantUser> -S <$SourceVRAServer>
```

## 2 Linux または Windows 仮想マシン上のソフトウェア エージェントをアップグレードするプラン ファイルを作成します。

- /var/log/vcac/agentupdate/{tenant}/{subtenant-UUID} にある migrate params ファイルを変更し、AWS または Azure エンドポイントに対応するプライベート IP アドレスの値を含めます。

```
"key": "ipAddress",
  "value": {
    "type": "string",
    "value": "<$PrivateIp:$PrivatePort>"
  }
}
```

- Linux マシンをアップグレードする場合は、このコマンドを使用します。

```
/usr/lib/vcac/server/webapps/ROOT/software/updateSoftwareAgents.py -a <
$DestinationVRAServer> -t <$Tenant> -S <$SourceVRAServer> -tu <$TenantUser> -CL
Software.LinuxAgentUpdate74 --source_cloud_provider azure
```

- Windows マシンをアップグレードする場合は、このコマンドを使用します。

```
/usr/lib/vcac/server/webapps/ROOT/software/updateSoftwareAgents.py -a <
$DestinationVRAServer> -t <$Tenant> -S <$SourceVRAServer> -tu <$TenantUser> -CW
Software.WindowsAgentUpdate74 --source_cloud_provider azure
```

- このコマンドは、プラン ファイルを実行します。

```
/usr/lib/vcac/server/webapps/ROOT/software/updateSoftwareAgents.py -a <
$DestinationVRAServer> -t <$Tenant> -tu <$TenantUser> --
plan_file /usr/lib/vcac/server/webapps/ROOT/software/plan
```

- 3 手順 1 のノード ファイルと手順 2 のプラン ファイルを使用してソフトウェア エージェントをアップデートするには、このコマンドを使用します。

```
/usr/lib/vcac/server/webapps/ROOT/software/updateSoftwareAgents.py -a <
$DestinationVRAServer> -t <$tenant> -tu <$TenantUser> --component_windows
Software.WindowsAgentUpdate74 --component_linux Software.LinuxAgentUpdate74 --
plan_file /usr/lib/vcac/server/webapps/ROOT/software/plan --plan_index 0 --
node_file /usr/lib/vcac/server/webapps/ROOT/software/node --source_cloud_provider azure
--action plan_batch -S <$SourceVRAServer>
```

代わりに、ノードのインデックスを指定することでノード ファイルから一度に 1 台のノードを実行するには、このコマンドを使用します。

```
/usr/lib/vcac/server/webapps/ROOT/software/updateSoftwareAgents.py -a <
$DestinationVRAServer> -t <$tenant> -tu <$TenantUser> --component_windows
Software.WindowsAgentUpdate74 --component_linux Software.LinuxAgentUpdate74 --
plan_file /usr/lib/vcac/server/webapps/ROOT/software/plan --plan_index 0 --
node_file /usr/lib/vcac/server/webapps/ROOT/software/node --source_cloud_provider azure
--action execute_node -S <$SourceVRAServer> --node_index <0 through n-1>
```

この手順を実行するとき、vRealize Automation 仮想アプライアンスおよびホスト マシンからログの末尾を監視してサーバ エージェントのアップグレード プロセスを表示することができます。

アップグレード後、アップグレード プロセスにより、Windows または Linux 用のソフトウェア アップデート スクリプトは vRealize Automation 7.4 仮想アプライアンスにインポートされます。vRealize Automation 仮想アプライアンス ホストにログインすると、ソフトウェア コンポーネントが正常にインポートされていることを確認できます。コンポーネントをインポートした後、ソフトウェア アップデートは古いイベント ブローカ サービス (EBS) に送信され、指定された仮想マシンにソフトウェア アップデート スクリプトが中継されます。アップグレードが完了し、新しいソフトウェア エージェントが稼動すると、ping 要求を送信して新しい vRealize Automation 仮想アプライアンスにバインドします。

---

**注:** 有用なログ ファイル

---

- ソース vRealize Automation の Catalina 出力 : /var/log/vcac/catalina.out。このファイルでは、エージェントの移行が行われるときに行われるアップグレード要求を確認できます。このアクティビティは、ソフトウェア プロビジョニング要求を実行する場合と同様です。
- ターゲット vRealize Automation の Catalina 出力 : /var/log/vcac/catalina.out。このファイルでは、ping 要求をレポートする移行された仮想マシンが確認できます。バージョン番号 7.4.0-SNAPSHOT が含まれます。これらは、sw-agent-UUID などの EBS トピック名を比較することで照合できます。
- ターゲット vRealize Automation マシン マスター アップグレード ログ ファイルのエージェント アップデート フォルダ : var/log/vmware/vcac/agentupdate/updateSoftwareAgents.log。このファイルの末尾を監視すると、どのアップグレード処理が進行中なのかを確認できます。
- テナント フォルダ下で利用可能な個々のログ : /var/log/vcac/agentupdate/{tenant}/{subtenant-UUID}。個々のノードがエラーや進行中の拡張機能がある多数のファイルとしてここに表示されます。

- 移行された仮想マシン : /opt/vmware-appdirector/agent/logs/darwin\*.log。この場所をスポットチェックすると、受信されているソフトウェア アップデート要求、および agent\_bootstrap + ソフトウェア エージェントの最後に発生する再起動のリストを確認できます。

## 移行後にプロパティ ディクショナリ設定を変更する

vRealize Automation 6.2.x から移行した後、プロパティ ディクショナリの **Label** コントロール タイプ プロパティを、ブループリントでオーバーライド不可と設定します。

vRealize Automation 6.2.x プロパティ ディクショナリにあった Label コントロールは、vRealize Automation 7.x にはありません。移行の際、**Label** コントロールは変換され、移行後のプロパティ ディクショナリでは **TextBox** タイプのコントロールになります。

移行後に、vRealize Automation プロパティ ディクショナリで手動により、またはエクスポートおよびインポート機能を使用して、影響を受けるプロパティをオーバーライド不可と設定します。

## ターゲット vRealize Automation 7.4 環境の検証

ターゲット vRealize Automation 環境にすべてのデータが正常に移行されたかどうかを確認できます。

### 前提条件

- vRealize Automation の最新バージョンに移行します。
- ターゲットの vRealize Automation コンソールにログインします。
  - ターゲット仮想アプライアンスの完全修飾ドメイン名 `https://<vra-virtual-hostname.domain.name>/vcac` を使用して vRealize Automation コンソールを開きます。  
  
高可用性環境の場合は、ターゲット仮想アプライアンス ロード バランサの完全修飾ドメイン名 `https://<vra-lb-hostname.domain.name>/vcac` を使用してコンソールを開きます。
  - テナント管理者のユーザー名とパスワードを使用してログインします。

### 手順

- 1 [インフラストラクチャ] - [管理対象マシン] の順に選択し、すべての管理対象仮想マシンが存在することを確認します。
- 2 [コンピューティングリソース] をクリックし、[データ収集]、[今すぐ申請]、[更新] の順にクリックし、エンドポイントが機能していることを確認します。
- 3 [設計] をクリックし、[ブループリント] ページで各ブループリントの要素を検証します。
- 4 [XaaS] をクリックし、[カスタム リソース]、[リソース マッピング]、[XaaS ブループリント]、および [リソース アクション] の内容を確認します。
- 5 [管理] - [カタログ管理] を選択し、[サービス]、[カタログ アイテム]、[アクション]、[資格] の内容を確認します。
- 6 [アイテム] - [デプロイ] を選択し、プロビジョニングされた仮想マシンの詳細を確認します。
- 7 [展開] ページで、プロビジョニングされたパワーオフの仮想マシンを選択し、[アクション] - [パワーオン] を選択します。次に、[送信] > [OK] の順にクリックします。仮想マシンが正常にパワーオンになっていることを確認します。

- 8 [カタログ] をクリックし、新しいカタログ アイテムを申請します。
- 9 [全般] タブで、申請情報を入力します。
- 10 マシンのアイコンをクリックし、すべてのデフォルト設定を受け入れて [送信] > [OK] の順にクリックします。
- 11 申請が正常に終了したことを確認します。

## 移行に関するトラブルシューティング

移行に関するトラブルシューティングのトピックでは、vRealize Automation の移行時に問題が発生した場合の解決策について説明します。

この章には、次のトピックが含まれています。

- PostgreSQL バージョンが原因のエラー
- 移行中に作成された展開が一部の仮想マシンに存在しない
- 移行ログの場所
- 移行後にサービス カタログに表示されるカタログ アイテムを申請できない
- vRealize Automation の無効になった [データ収集] のラジオ ボタン
- ソフトウェア エージェントのアップグレードのトラブルシューティング

### PostgreSQL バージョンが原因のエラー

更新された PostgreSQL データベースが含まれているソースの vRealize Automation 6.2.x 環境では、管理者アクセスがブロックされます。

#### 問題

アップグレードされた PostgreSQL データベースが vRealize Automation 6.2.x で使用されている場合、管理者は vRealize Automation からこのデータベースへのアクセスを提供するエントリを **pg\_hba.conf** ファイルに追加する必要があります。

#### ソリューション

- 1 **pg\_hba.conf** ファイルを開きます。
- 2 このデータベースへのアクセス権を付与するには、次のエントリを追加します。

```
host all <vcac-database-user> <vra-va-ip> <trust-method>
```

### 移行中に作成された展開が一部の仮想マシンに存在しない

移行時に仮想マシンの状態が指定なしの場合、ターゲット環境で対応する展開が作成されません。

## 問題

移行時に、移行前の環境で仮想マシンの状態が「情報なし」であった場合、移行先で対応する展開環境が作成されません。

## ソリューション

- ◆ 移行後に仮想マシンが「情報なし」の状態でなくなると、パルク インポートを使用して移行先に仮想マシンをインポートすることができます。

## 移行ログの場所

検証または移行の問題をトラブルシューティングするには、移行プロセスを記録したログを確認します。

表 7-1. ソース vRealize Automation アプライアンス

ログ	場所
パッケージ作成ログ	/var/log/vmware/vcac/migration-package.log

表 7-2. ターゲット vRealize Automation アプライアンス

ログ	場所
移行ログ	/var/log/vmware/vcac/migrate.log
移行実行ログ	/var/log/vmware/vcac/mseq.migration.log
移行実行出力ログ	/var/log/vmware/vcac/mseq.migration.out.log
検証実行ログ	/var/log/vmware/vcac/mseq.validation.log
検証実行出力ログ	/var/log/vmware/vcac/mseq.validation.out.log

表 7-3. ターゲット vRealize Automation インフラストラクチャ ノード

ログ	場所
移行ログ	C:\Program Files (x86)\VMware\VCAC\InstallLogs- YYYYMMDDHHMMXX\Migrate.log
検証ログ	C:\Program Files (x86)\VMware\VCAC\InstallLogs- YYYYMMDDHHMMXX\Validate.log

## 移行後にサービス カタログに表示されるカタログ アイテムを申請できない

vRealize Automation の最新バージョンに移行すると、前のバージョンの特定のプロパティ定義を使用するカタログ アイテムがサービス カタログに表示されますが、申請することはできません。

## 問題

6.2.x 以前のバージョンから移行していて、次のコントロール タイプまたは属性を持つプロパティ定義が設定されている場合は、これらの要素がアップグレード後のプロパティ定義に含まれていないため注意が必要です。これらの定義を使用するカタログ アイテムは、移行前のように動作しなくなります。

- コントロール タイプ：チェック ボックスまたはリンク
- 属性：関係、正規表現、またはプロパティのレイアウト

## 原因

vRealize Automation 7.0 以降では、プロパティ定義でこれらの要素が使用されなくなりました。プロパティ定義を再作成するか、組み込みのコントロール タイプまたは属性ではなく vRealize Orchestrator スクリプト アクションを使用するようにプロパティ定義を設定する必要があります。

スクリプト アクションを使用して、これらのコントロール タイプまたは属性を vRealize Automation 7.x に移行します。

## ソリューション

- 1 vRealize Orchestrator でこれらのプロパティ値を返すスクリプト アクションを作成します。このアクションは単純な値を返す必要があります。たとえば、文字列、整数、またはその他のサポートされているタイプです。このアクションは、依存する他のプロパティを入力パラメータとして取ることがあります。
- 2 vRealize Automation コンソールで、製品の定義を設定します。
  - a [管理] - [プロパティ ディクショナリ] - [プロパティ定義] の順に選択します。
  - b プロパティ定義を選択して、[[編集]] をクリックします。
  - c [アドバイスの表示] ドロップダウン メニューで、[ドロップダウン] を選択します。
  - d [値] ドロップダウン メニューで、[外部値] を選択します。
  - e スクリプト アクションを選択します。
  - f [OK] をクリックします。
  - g スクリプト アクションに含まれる入力パラメータを設定します。既存の関係を維持するには、パラメータを他のプロパティにバインドします。
  - h [OK] をクリックします。

## vRealize Automation の無効になった [データ収集] のラジオ ボタン

vRealize Automation 6.2.x から 7.x への移行後、移行先の vRealize Automation の [コンピュータ リソース] ページでは [データ収集] のラジオボタンが無効になります。

## 原因

エンドポイントを参照する移行元環境にエージェントをインストールし、同じエンドポイントを参照する移行先環境に別の名前のエージェントをインストールすると、移行先環境で管理者としてエンドポイントに対するテスト接続を実行できます。ただし、移行先環境の vRealize Automation にファブリック管理者としてログインした場合、[データ収集] の [コンピュータ リソース] ページにあるラジオ ボタンが無効になります。



## ソリューション

この状況を回避するには、移行先環境にインストールされているエージェントに、移行元環境にインストールされているエージェントと同じ名前を付けます。

# ソフトウェア エージェントのアップグレードのトラブルシューティング

vRealize Automation アプライアンス管理を使用してソフトウェア エージェントをアップグレードする際に、ログ ファイルを確認して発生した問題の原因を特定することができます。

## 問題

ソフトウェア エージェントのアップグレード時、問題が発生することがあります。ソフトウェア エージェントのアップグレード処理時に、ログ ファイルを確認すると、問題の発生箇所を特定することができます。

---

### 注: サーバ ログ

---

- プロセスを観察するには、サーバの updateSoftwareAgents.log ファイル (/storage/log/vmware/vcac/agentupdate/updateSoftwareAgents.log) に tail コマンドを実行します。
- どのソフトウェア エージェントが成功しているか確認するには、ターゲット アプライアンスの catalina.out ファイル (/var/log/vcac/catalina.out) に tail コマンドを実行します。

7.4.0-SNAPSHOT に対してレポートされる「ping」などの文字列を検索します。

詳細情報は、次の場所で確認できます。

- /var/cache/vcac/agentupdate/{Tenant}/{UUID}/UUID.plan
- /var/cache/vcac/agentupdate/{Tenant}/{UUID}/UUID.log
- /var/cache/vcac/agentupdate/sqa/UUID/UUID.log (OS 単位)

メジャー バッチのアップグレードを開始する前に、テスト仮想アプライアンス ソフトウェア エージェントのアップグレードを必ず実行する必要があります。プロセスの概要は以下のとおりです。

- ターゲット仮想アプライアンスに行われた最初の要求を確認して、エージェントのバージョンを識別します。
- アップグレードでソース仮想アプライアンスに行われた要求を確認します。
- ターゲット仮想アプライアンスの新しい 7.4 バージョンをレポートするエージェントを確認します。
- これらのイベントの間、/storage/log/vmware/vcac/agentupdate/updateSoftwareAgents.log の updateSoftwareAgents.log ファイルを確認します。

---

### 注: クライアント ログ

---

Linux エージェントのログは appdirector のエージェント ログ フォルダにあります: /opt/vmware-appdirector/agent/logs/\*.log

次のようなログ エラーが表示される可能性があります。これは、アップグレード処理中に EBS キューが断続的に機能することによる一時的なものです。

Feb 15 2018 16:54:10.105 ERROR [EventPoller-sw-agent-0ad2418d-5b42-4231-a839-a05dd618e43e] []  
com.vmware.vcac.platform.event.broker.client.rest.RestEventSubscribeHandler - Error while polling events  
for subscription '{}'.  
org.springframework.web.client.HttpClientErrorException: 404 Not Found

at

org.springframework.web.client.DefaultResponseErrorHandler.handleError(DefaultResponseErrorHandler.java:91) ~[nobel-agent.jar:na]

at org.springframework.web.client.RestTemplate.handleResponse(RestTemplate.java:641) ~[nobel-agent.jar:na]

at org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:597) ~[nobel-agent.jar:na]

at org.springframework.web.client.RestTemplate.execute(RestTemplate.java:557) ~[nobel-agent.jar:na]

at org.springframework.web.client.RestTemplate.exchange(RestTemplate.java:503) ~[nobel-agent.jar:na]

at

com.vmware.vcac.platform.event.broker.client.rest.RestEventSubscribeHandler.pollEvents(RestEventSubscribeHandler.java:297) ~[nobel-agent.jar:na]

at com.vmware.vcac.platform.event.broker.client.rest.RestEventSubscribeHandler  
\$EventPoller.run(RestEventSubscribeHandler.java:329) ~[nobel-agent.jar:na]