

Application Services の使用

vRealize Automation 6.2

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

VMware の Web サイトでは最新の製品アップデートも提供されています。

このドキュメントに関するご意見およびご感想がある場合は、docfeedback@vmware.com までお送りください。

Copyright © 2012–2015 VMware, Inc. 無断転載を禁ず。著作権および商標情報。

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

ヴァイエルムウェア株式会社
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

Application Services の使用	9
更新情報	11
1 Application Services について	13
Application Services の概要	13
コア アーキテクチャの原理	15
任意のアプリケーションとミドルウェア サービスの展開	15
マルチクラウドのサポート	15
Application Services の拡張性とオープン アーキテクチャ	16
Application Services での標準化	16
ユーザー セキュリティ	16
主要な概念	16
2 Application Services のインストールおよび構成	21
3 Application Services のインストール	23
Application Services のインストール準備	23
Application Services のシステム要件	24
vCenter Server クラスタ構成のセットアップ	25
vCloud Director の Application Services のセットアップ	27
Application Services アプライアンスの起動	29
Application Services アプライアンスの構成	29
アプライアンスと vRealize Automation サーバの登録	30
事前定義済みサンプル コンテンツのインポート	31
Application Services Web インターフェイスを開く	31
Application Services Web インターフェイスに接続する際に問題が発生した場合のトラブルシューティング	33
darwin_user アカウントのロック解除	33
Application Services の再起動	34
外部 URL 用にプロキシを使用するように Application Services を構成する	34
vRealize Automation への Application Services の登録	35
4 Application Services のアップグレード	37
Application Services のアップグレード	37
Application Services のアップグレードのロールバック	38
5 ユーザーとグループの設定	39
6 Application Services でのテナントとビジネス グループの使用	41
7 テナントへの事前定義済みコンテンツのインポート	43

- 8 Application Services Web インターフェイスの使用 45
 - Application Services にログインする 45
 - Application Services Web インターフェイスの使用 46
 - Application Services のボタンの使用 47
 - Application Services のアイコンについて 48
- 9 Application Services 環境のためのアプリケーション プロビジョニングのセットアップ 49
 - vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 50
 - vRealize Automation での仮想マシン テンプレートの作成 51
 - vCenter Server Windows 仮想マシンのテンプレートを準備する 52
 - vCenter Server Linux 仮想マシンのテンプレートを準備する 54
 - クローン展開のための vRealize Automation ブループリントの作成 55
 - リンク クローン展開のための vRealize Automation ブループリントの作成と構成 56
 - vRealize Automation での既存の仮想マシン テンプレートの更新 58
 - vRealize Automation クラウド プロバイダとテンプレートの登録 58
 - vRealize Automation 展開環境の作成 60
 - 外部サービス インスタンスのマッピング 61
 - ポリシー インスタンスの作成 62
 - Puppet ソリューション インスタンスの登録 63
- 10 vCloud Director 環境のためのアプリケーション プロビジョニングのセットアップ 71
 - vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 72
 - vCloud Director での Windows 仮想マシン テンプレートの作成 73
 - vCloud Director vApp の作成 74
 - vCloud Director での Windows 仮想マシン テンプレートの構成 74
 - Windows 仮想マシン テンプレートに対する SID の変更とドメインへの参加の有効化 76
 - vCloud Director カタログへの Windows 仮想マシン テンプレートの追加 77
 - vCloud Director での Linux 仮想マシン テンプレートの作成 77
 - vCloud Director カタログからのクラウド テンプレートの構成の確認 79
 - vCloud Director での既存の仮想マシン テンプレートのアップデート 79
 - OVF フォーマットによる仮想マシン テンプレートのエクスポート 80
 - vCloud Director クラウド プロバイダとテンプレートの登録 80
 - vCloud Director 展開環境の作成 81
- 11 Amazon EC2 環境でのアプリケーション プロビジョニングの設定 83
 - Application Services の Amazon EC2 環境を構成する 84
 - エンドポイント VM を作成する 85
 - Amazon EC2 に接続するためのクラウド トンネルの作成 86
 - Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 88
 - Amazon EC2 仮想マシン テンプレートまたは AMI の作成 89
 - Amazon EC2 のクラウド プロバイダとテンプレートの登録 90
 - Amazon EC2 の展開環境の作成 92
- 12 Application Services コンポーネントの開発 93
 - コンポーネント アクションの定義 93
 - サポートされているアクション スクリプトとカスタム タスク スクリプト 94

コンポーネント プロパティの構成	95
プロパティのタイプ	95
プロパティ値の定義	100
ほかのプロパティへのバインド	100
ほかのプロパティへの自動バインド	101
ライフ サイクル ステージとコンポーネント間のプロパティ値の受け渡し	103
Application Services の事前定義済みプロパティ	103
事前定義済みの IP アドレス プロパティ	103
事前定義されたノード アレイ インデックス プロパティ	104
事前定義済みのディスク レイアウト情報プロパティ	104
セキュアなコンポーネント プロパティ	104
必要なコンポーネントのプロパティ	105
コンポーネントの開発のベスト プラクティス	105

13 Application Services ライブラリの管理 107

ライブラリへのオペレーティングシステムの追加	108
ライブラリへのタグの追加	109
ライブラリへのサービスの追加	109
ライブラリでサービス バージョンを作成する	110
サービス バージョン プロパティを定義する	111
サービス バージョンへのアクション スクリプトの追加	114
サービス バージョンを保守する	115
ライブラリからサービスを削除する	116
ライブラリへの Puppet サービスのインポート	116
ライブラリへの外部サービスの追加	117
ライブラリでの外部サービスの作成	118
外部サービス プロパティを定義する	119
高度な外部サービス バージョンのプロバイダ仕様を作成する	122
外部サービス バージョンを保守する	123
外部サービスを削除する	124
ライブラリへの論理テンプレートの追加	125
論理テンプレート バージョンの管理	127
ライブラリからの論理テンプレートの削除	128
ライブラリへのポリシーの追加	128
ポリシー定義スクリプトの作成	130
ポリシー バージョンの維持	131
ポリシーの削除	132
カスタム タスクをライブラリに追加する	132
カスタム タスク バージョンを保守する	134

14 アプリケーションの作成 137

アプリケーションの作成	138
アプリケーション バージョンの作成	138
アプリケーション ブループリントのモデル化	139
アプリケーション ブループリントを構成する	141
詳細ブループリントの作成	142
コンポーネント間の依存関係の作成	143
クラスタとしてノードを指定する	143

	ノード用に複数の NIC を定義する	144
	フレキシブル ディスク レイアウトの追加と管理	145
	詳細ブループリントへの外部サービスの追加	146
	アプリケーション バージョンをコピーする	146
	アプリケーションのコピー	147
	アプリケーション バージョンの削除	147
15	成果物の操作	149
	成果物リポジトリ仕様の作成	150
	成果物リポジトリ インスタンスの作成	152
	成果物仕様の作成	153
	成果物仕様の成果物リポジトリ インスタンスへのマップ	153
	アプリケーション ブループリントへの成果物のバインド	154
	成果物を使用したアプリケーションの展開	155
16	アプリケーションの展開	157
	展開プロファイルの設定と構成	157
	展開プロファイルの作成	157
	展開環境の構成	158
	アプリケーション プロパティの構成	162
	実行プランを確認しカスタム タスクを追加する	162
	既存の展開プロファイルを使用する	164
	複合展開プロファイルの作成	165
	展開プロファイルを使用した展開	168
	単一の展開プロファイルで展開する	168
	複合展開プロファイルを使用した展開	168
	アプリケーションをクイック展開する	169
	vCloud Automation Center サービス カタログへの展開プロファイルの公開	170
	単一展開プロファイルの公開	170
	複合展開プロファイルの公開	172
	vRealize Automation のサービス カタログ アイテムの申請	173
	[展開サマリ] ページの使用	174
	[複合展開サマリ] ページの使用	174
	展開および更新プロセスについて	175
	展開エラーについて	176
	展開の失敗の解決	177
17	アプリケーション展開の更新	179
	更新プロセスの開始と展開のスケール アウト	180
	更新プロセスを開始して展開をスケール インする	182
	更新プロセスの開始と構成の変更	185
	既存の更新プロファイルの使用	187
	更新プロファイルの昇格	188
	昇格された更新プロファイルを承諾または破棄する	189
	更新プロセスのロールバック	189
	カスタム タスクの実行の更新について	190
	展開のスケールアップに失敗した更新プロセスのトラブルシューティング	191
	構成を変更する更新プロセスが失敗した場合のトラブルシューティング	191

18	事前定義済みライブラリ コンポーネントの展開	193
	サンプル テンプレートの使用	193
	libpython パッケージを Ubuntu テンプレートに追加する	194
	Ubuntu 仮想マシンの SSH 接続を有効にする	194
	展開に使用できるサンプル アプリケーション	194
	Nanotrader アプリケーションを展開する	195
	Clustered DotShoppingCart アプリケーションを展開する	196
	Clustered Dukes Bank アプリケーションを展開する	197
	jPetStore アプリケーションを展開する	198
	Radiant CMS アプリケーションを展開する	199
	展開されたサンプル アプリケーションを更新する	200
	サンプル アプリケーションをスケーリングする更新プロセスの開始	200
	更新プロセスを開始してサンプル アプリケーション内の構成を変更する	201
	サンプルの Application Services ライブラリ タスク	202
	ドメインへの参加用事前定義済みタスクの追加	203
	APT Repository Config 事前定義済みタスクを追加する	204
	YUM Repository Config 事前定義済みタスクを追加する	205
	RHN 登録用事前定義済みタスクの追加	207
	サンプルのライブラリ サービス	208
19	展開の管理	209
	アプリケーションの展開 タスクとブループリントの詳細の表示	210
	展開された仮想マシンの詳細情報とアプリケーションの実行プランの表示	211
	ポリシー スキャンの開始	213
	クラウドからのアプリケーションの分解	214
	vCloud Automation Center からの展開のスケール イン	215
	vCloud Automation Center からの展開のスケールアウト	215
	vCloud Automation Center からのアプリケーションの分解	216
	Application Services からアプリケーション展開を削除する	217
	展開または更新プロセスのキャンセル	218
	ポリシー コンプライアンスの概要の表示	219
20	Application Services CLI の使用	221
	リモートでの CLI の開始	221
	一般的な CLI オプション	222
	クラウド トンネルの管理	223
	CLI によるアプリケーションの展開と更新	224
	CLI を使用した展開の分解	227
21	CLI インポート/エクスポート機能の使用	229
	CLI エクスポート コマンドの使用	229
	CLI インポート コマンドの使用	231
	インデックス	235

Application Services の使用

vRealize Automation Application Services (旧 vCloud Application Director) は、インフラストラクチャ クラウドでのアプリケーションのコンポーネントおよび依存するミドルウェア プラットフォーム サービスの展開、構成、および更新を含む、クラウド内のアプリケーション プロビジョニングを自動化します。Application Services 機能は、vCloud Director、vSphere、および Amazon Elastic Compute Cloud (Amazon EC2) をベースにしたインフラストラクチャ クラウドでのカスタム アプリケーションおよびパッケージ化されたアプリケーションの複雑な展開を簡単にします。

このドキュメントでは、Application Services を使用して、仮想およびクラウド ベースの複数のインフラストラクチャでアプリケーションを作成、展開、管理、および更新する方法について説明します。

対象者

この情報は、アプリケーションの展開に Application Services を使用するユーザーを対象としています。対象者には、アプリケーション インフラストラクチャ管理者およびクラウド管理者と連携して作業するアプリケーション アーキテクトとアプリケーション展開担当者が含まれます。

更新情報

この『Application Services の使用』ガイドは、製品のリリースごと、または必要に応じて更新されます。

『Application Services の使用』ガイドの更新履歴を表に示します。

リビジョン	説明
001654-02	<ul style="list-style-type: none">■ バージョン値を削除し、https://www.vmware.com/support/pubs/vcac-pubs.html にある vRealize Automation ドキュメントのサポート マトリックスへの参照を「vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 (P. 50)」、vCenter Server Linux 仮想マシンのテンプレートを準備する (P. 54)、「Application Services のシステム要件 (P. 24)」、および「vCenter Server クラスタ構成のセットアップ (P. 25)」に追加しました。 「Application Services のアップグレード (P. 37)」を更新しました。
001654-01	<ul style="list-style-type: none">■ 「vRealize Automation ブループリントからのスナップショットの作成 (P. 57)」と「リンク クローン展開のための vRealize Automation ブループリントの作成と構成 (P. 56)」を更新し、手順の例を分かりやすくしました。■ 「vRealize Automation での仮想マシン テンプレートの作成 (P. 51)」を更新しました。■ 「Application Services のアップグレード (P. 37)」を更新し、アップグレードの終了後に vRealize Automation の再起動が必要なことを明記しました。■ 「Amazon EC2 のクラウド プロバイダとテンプレートの登録 (P. 90)」のトピックが更新され、Amazon のリージョンコードの値をサポートしたリストに改善されました。■ 「展開された仮想マシンの詳細情報とアプリケーションの実行プランの表示 (P. 211)」のトピックを更新し、VMware Remote Console の情報を含めました。
001654-00	初版 6.2 リリース。

Application Services について

VMware vRealize Automation Application Services (旧 VMware vCloud Application Director) は、モデルベースのアプリケーション プロビジョニングソリューションであり、複数のインフラストラクチャ クラウドでアプリケーション展開のトポロジを簡単に作成および標準化できるようにします。アプリケーション アーキテクトは、ドラッグアンドドロップ インターフェイスを備えたグラフィックベースのキャンパスを使用して、アプリケーション ブループリントと呼ばれるアプリケーション展開のトポロジをモデル化できます。

アプリケーション ブループリントによってアプリケーションの構造が定義され、標準化されたアプリケーション インフラストラクチャ コンポーネントが使用できるようになります。またアプリケーション ブループリントには、カスタムおよびパッケージ化されたエンタープライズ アプリケーションのインストール依存関係とデフォルト設定が含まれます。アプリケーション ブループリントは、VMware、vCloud Director などの vRealize Automation ベースの IaaS クラウドや、VMware vCloud Air および Amazon EC2 などのパブリック クラウド間で移行可能な論理展開トポロジです。

この章では次のトピックについて説明します。

- [Application Services の概要 \(P. 13\)](#)
- [コア アーキテクチャの原理 \(P. 15\)](#)
- [主要な概念 \(P. 16\)](#)

Application Services の概要

VMware vRealize Automation Application Services は、モデル駆動型のオープンかつ拡張可能なアーキテクチャを備えています。Application Services は、標準コンポーネントまたはサービスのライブラリを使用して、ハイブリッドクラウド環境で、多層エンタープライズ アプリケーションの展開の更新ライフ サイクルを自動化および管理します。

エンタープライズ ユーザーは、動的なクラウド環境内で複雑なアプリケーションを標準化、展開、構成、更新、およびスケール調整できます。このようなアプリケーションは、単純な Web アプリケーションから複雑なカスタム アプリケーションやパッケージ化されたアプリケーションまで、多岐にわたります。

Application Services は vRealize Automation を使用して、ユーザーとグループのサポート、アクセス制御、カタログ管理を行います。vRealize Automation インスタンスには複数のテナントを、テナントには複数のビジネス グループを、ビジネス グループには複数のユーザーとオブジェクト（アプリケーションやサービスなど）を含めることができます。ユーザーがビジネス グループでオブジェクトの追加や編集を行うには、そのビジネス グループのメンバーである必要があります。ビジネス グループのプライベート オブジェクトを表示するには、そのグループに所属している必要があります。共有オブジェクトを表示する場合は、同じテナントの任意のビジネス グループに所属できます。Application Services に展開されたアプリケーションは、vRealize Automation のカタログ アイテムになります。ユーザーはここで、これらのカタログ アイテムのプロビジョニングを申請できます。

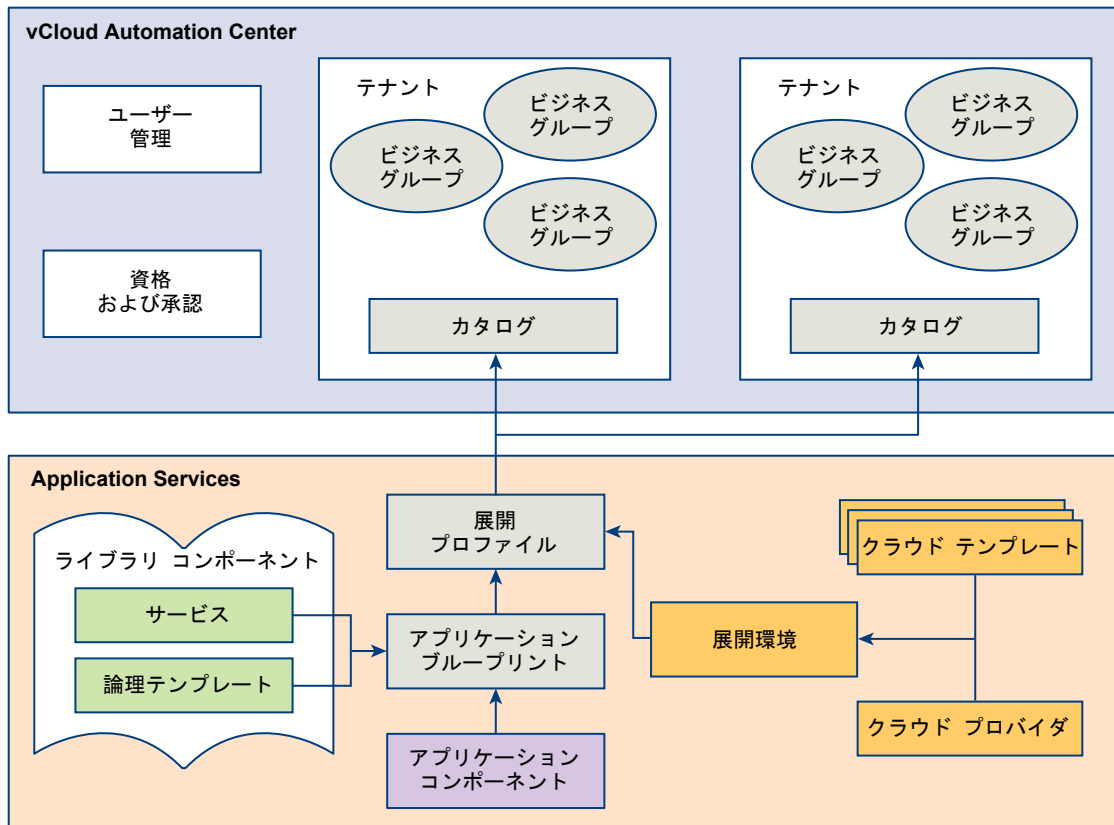
サポートされているクラウド環境へのアプリケーションの展開を自動化するには、ユーザーがクラウド抽象化レイヤ (CAL) のコンポーネントを構成する必要があります。クラウドテンプレートには、仮想マシンの作成時に仮想マシンに適用されるオペレーティング システムおよびデータが組み込まれた、事前定義済みの再利用可能なマシン イメージが含まれています。クラウドテンプレートは、Application Services ライブラリ内の論理テンプレートにマップされます。クラウドプロバイダは、展開用のクラウド インスタンスを提供します。展開環境は、クラウド プロバイダ インスタンスでの展開用の特定の環境を提供します。クラウド プロバイダと展開環境の両方で、クラウド環境内のコンポーネントを Application Services にマップします。

アプリケーション アーキテクトは、ドラッグ アンド ドロップ インターフェイスを使用して、仮想アプリケーションのブループリントを作成できます。アプリケーション アーキテクトは、標準の論理テンプレート、アプリケーション インフラストラクチャ サービス、コンポーネント、およびスクリプトのあらかじめ用意された拡張可能なライブラリを使用して、アプリケーション ブループリントをモデル化できます。このようなブループリントにより、ソフトウェア コンポーネント、依存関係、構成を含むアプリケーション構造が標準化され、繰り返し実行される展開に利用されます。

アプリケーション ブループリントは、展開環境内を移動させることができます。たとえば、ブループリントを利用できるようになると、アプリケーション開発、QA、およびリリース チームは、IT によって規定された基準で作業できます。チームは、標準ブループリントの展開、許可された構成のカスタマイズ、IT に認可された展開環境内での展開を繰り返し行うことができます。

アプリケーション ブループリントに基づき、展開プロファイルを使用してさまざまな展開を作成してプロトタイプをテストすることや、ミッション クリティカルな多層アプリケーションを本番環境に展開することができます。アプリケーション展開者は、このような保存されたブループリントから、アプリケーションをプライベート クラウドまたはパブリック クラウドに展開するための実行プランを生成できます。また、新しいバージョンが使用可能になったときに、更新プロセスを開始して展開済みアプリケーションのクラスタ化されたノードをスケール調整したり、展開済みアプリケーションの構成またはコードを変更したりすることもできます。

図 1-1. Application Services および vRealize Automation のワークフロー



コア アーキテクチャの原理

Application Services は、どの IaaS クラウドにも複雑なアプリケーションを自動展開するように設計されています。Application Services は、vCloud Director、vRealize Automation に展開するための vSphere、および Amazon EC2 で IaaS クラウド環境内の展開を自動化します。

- [任意のアプリケーションとミドルウェア サービスの展開 \(P. 15\)](#)

Application Services を使用することで、アプリケーション アーキテクトは仮想マシンのテンプレートとスクリプトを使用して、ミドルウェア サービスとアプリケーションが含まれるアプリケーション展開をモデル化できます。

- [マルチクラウドのサポート \(P. 15\)](#)

Application Services は、同じアプリケーションを複数のタイプのクラウドに展開するように設計されています。

- [Application Services の拡張性とオープン アーキテクチャ \(P. 16\)](#)

Application Services は vCloud コンポーネント用に最適化されており、他のコンポーネントに拡張可能です。

- [Application Services での標準化 \(P. 16\)](#)

Application Services では、標準化された構成プロパティを使用して再利用可能なサービスを作成して IT コンプライアンスの厳格な要件を満たすことができます。

- [ユーザー セキュリティ \(P. 16\)](#)

ユーザーは、登録済みテナントの SSO 認証情報と、割り当てられた少なくとも 1 つのアプリケーション ユーザーロールを使用して Application Services にログインします。

任意のアプリケーションとミドルウェア サービスの展開

Application Services を使用することで、アプリケーション アーキテクトは仮想マシンのテンプレートとスクリプトを使用して、ミドルウェア サービスとアプリケーションが含まれるアプリケーション展開をモデル化できます。

Application Services を使用すると、vCloud Director および vRealize Automation クラウド環境で Windows および Linux オペレーティングシステムにアプリケーションを展開できます。Amazon EC2 クラウド環境の場合、アプリケーションは Linux オペレーティングシステムにのみ展開できます。

- アプリケーション アーキテクトは、企業向けに定義された標準のクラウド プロバイダ ライブラリにある仮想マシンテンプレートを使用できます。
- アプリケーション アーキテクトは、アプリケーション ブループリント内で定義されたミドルウェア サービスにアプリケーション コンポーネントを追加することもできます。

マルチクラウドのサポート

Application Services は、同じアプリケーションを複数のタイプのクラウドに展開するように設計されています。

- Application Services は展開の設定を、アプリケーション ブループリントとは別に、展開プロファイルとしてカプセル化します。展開プロファイルにより、vCloud Director に基づくプライベート クラウド、Amazon EC2 に基づくパブリック クラウドの境界を越えてアプリケーション ブループリントの移行が可能になります。また、展開プロファイルでは、vRealize Automation プロビジョニング インフラストラクチャを使用して、プライベートおよびパブリックの vSphere クラウドにアクセスします。
- Application Services は CAL を使用して、クラウド プロバイダにプラグインします。vCloud Director と vSphere への展開を vRealize Automation と Amazon EC2 クラウド環境を介して自動化します。vCloud Director は、vCloud API および Open Virtualization Format (OVF) のようなオープンスタンダードを使用します。Application Services は、vCloud をベースにした、一般公開されている IaaS API を介して vCloud Director に統合されています。この統合化によって、vCloud Director クラウドのプライベートおよびパブリック展開環境でアプリケーションを展開できます。

vCloud Automation Center は REST API を使用して vSphere プライベート/パブリック クラウドにアプリケーションを展開します。

Application Services は、Amazon AWS API および Amazon Machine Images (AMI) を使用して、Amazon EC2 クラウドにアプリケーションを展開します。

Application Services の拡張性とオープン アーキテクチャ

Application Services は vCloud コンポーネント用に最適化されており、他のコンポーネントに拡張可能です。

- Application Services ライブラリには、ミドルウェア サービス用の（すぐに使える）事前定義済みサービスまたはアプリケーションが登録されています。また、クラウド ライブラリ内に存在する仮想マシン テンプレートに対するポイントを追加することもできます。Application Services ライブラリでは、インストール、構成、開始、更新、ロールバック、分解用の各スクリプトと適切な構成を使用して、仮想マシン テンプレート上に動的にインストール可能なカスタム サービスの定義を追加することもできます。
- 展開実行プランは、ブループリントを基にシステムによって生成されます。これらのプランにより、ユーザーは、展開中にタスクのステータスと進行状況を追跡できます。

これらのプランにカスタム タスク スクリプトを追加して、展開済み仮想マシンで実行できます。これらのスクリプトを使用して、セキュリティ パッチ、統合監査、サードパーティ製内部 IT システムを使用した品質とコンプライアンスの評価、スモーク テストなど、追加のタスクを実行できます。
- ビルドから展開までのプロセスを簡素化するため、組織では、展開の自動化を推進できます。具体的には、コマンドライン インターフェイスから常時稼働のビルドシステムまたはクラウド プロビジョニング ポータルを使用して、アプリケーションを生成および展開します。

Application Services での標準化

Application Services では、標準化された構成プロパティを使用して再利用可能なサービスを作成して IT コンプライアンスの厳格な要件を満たすことができます。

Application Services には、次の標準化された構成プロパティが含まれます。

- IT 認定済みの仮想マシン テンプレートおよびアプリケーション ブループリント内のミドルウェア サービスを追加できるモデル駆動型のアーキテクチャ。
- アプリケーション カタログ管理者、アプリケーション アーキテクト、および展開者間の構成名の値のペアをオーバーライドして、アプリケーションおよびミドルウェア サービスの構成値を標準化するための委任モデル。

ユーザー セキュリティ

ユーザーは、登録済みテナントの SSO 認証情報と、割り当てられた少なくとも 1 つのアプリケーション ユーザー ロールを使用して Application Services にログインします。

Application Services は、vRealize Automation を介した認証メカニズムもサポートします。Application Services で登録されているテナント用に vRealize Automation で構成されている ID ストア (IDP) のすべてのユーザーは、少なくとも 1 つのアプリケーション ユーザー ロールが割り当てられていれば SSO 認証情報を使用してログインできます。ユーザー ロールの割り当て方法については、「vRealize Automation のテナント管理」を参照してください。

主要な概念

アプリケーションを展開するには、CAL を構成する必要があります。ブループリントをモデル化する場合、クラウド プロバイダ ライブラリから直接取得する仮想マシン テンプレートに加えて、すぐに使えるアプリケーション インフラストラクチャ コンポーネントおよびスクリプトの拡張可能なライブラリを使用できます。アプリケーション展開トポロジをモデル化した後で、依存関係を作成して構成を編集し、実行プランを完成させることができます。アプリケーションを展開した後で、新しいバージョンが使用可能になったときに更新プロセスを開始してクラスタ化ノードを拡大/縮小したり、展開されたアプリケーションの構成やコードを変更したりできます。

次の定義は、プロビジョニングおよび更新のワークフローを理解するのに役立ちます。

CAL の主なコンポーネント。

クラウド テンプレート

オペレーティング システムとデータを含む事前定義済み再利用可能マシン イメージを含むクラウド テンプレート。クラウド環境のための仮想マシンを作成すると、クラウド テンプレートで定義される仕様とソフトウェアがその仮想マシンに適用されます。

表 1-1. サポートされているクラウド環境へのクラウド テンプレートのマッピング

Application Services 環境	クラウド環境
クラウド テンプレート	Application Services テンプレート
	vRealize Automation ブループリント
	Amazon マシン イメージ (AMI)

クラウド プロバイダ

展開のためのクラウド インスタンス。クラウド プロバイダの 1 つのタイプに複数のクラウド プロバイダを定義できます。

表 1-2. サポートされているクラウド環境へのクラウド プロバイダのマッピング

Application Services 環境	クラウド環境
クラウド プロバイダ	vCloud Director 組織
	vRealize Automation 6.1 のビジネス グループ
	Amazon リージョン

クラウド プロバイダのタイプ

展開を作成できるクラウド インフラストラクチャのタイプ。Application Services は、vCloud Director、vRealize Automation、および Amazon EC2 のみをサポートします。

展開環境

開発、テスト、ステージング、本番などのクラウド プロバイダ内の環境。クラウド プロバイダには、複数の展開環境を含めることができます。

表 1-3. サポートされているクラウド環境への展開環境のマッピング

Application Services 環境	クラウド環境
展開環境	vCloud Director 組織の vDC
	vRealize Automation の予約ポリシー
	Amazon VPC

vCloud Director の場合、展開環境は定義されているクラウド プロバイダのために組織の仮想データセンター (vDC) にマッピングされ、その vDC のリソースを使用します。組織の vDC は組織にリソースを提供し、プロバイダの vDC からパーティション分割されます。組織の vDC は、仮想システムを格納、展開、および運用できる環境を提供します。また、フロッピー ディスク、CD ROM などの仮想メディアにストレージも提供します。単一の組織には、組織の vDC を複数含めることができます。

vRealize Automation の場合、展開環境は予約ポリシーにマッピングされます。展開環境が選択されていない場合、vRealize Automation は vRealize Automation ブループリントの仮想マシンのリソース要件によって予約ポリシーを割り当てます。

Amazon EC2 の場合、展開環境は Amazon Virtual Private Cloud (VPC) とリージョン内の可用性ゾーンの 1 つの組み合わせにマッピングされます。

主なライブラリのコンポーネント。

ライブラリ	クラウド テンプレートへのポインタである論理テンプレートを含むライブラリ。複数のアプリケーションで使用でき、仮想マシンにインストールできる再利用可能サービス。アプリケーション展開でカスタマイズされた追加のタスクを実行できるタスク。
サービス	仮想マシンにインストールし、複数のアプリケーションで再利用できるスクリプトによるソフトウェア。
外部サービス	アプリケーション展開の外部にインストールされるハードウェア ロード バランサー、プレインストールされたデータベースなどのアプリケーション サービス。アプリケーションと外部サービスは、連携して動作するように構成する必要があります。
論理テンプレート	Application Services の事前定義済み仮想マシンの定義。論理テンプレートは、クラウド ライブラリの実際のクラウド テンプレートとサポートされているサービスにマッピングできます。論理テンプレートにより、アプリケーション ブループリントはクラウドに依存する必要がなくなります。
ポリシー	アプリケーション ライフ サイクルの操作を制御する、ユーザーが定義した定義のセット。たとえば、ポリシーでは、企業のガイドラインに基づいて特定のソフトウェアの使用をブラックリストに記載できます。ポリシーは、ポリシー インスタンスを使用して特定の展開環境で有効になると適用されます。
カスタム タスク	実行計画からカスタム タスクを追加して、アプリケーション展開にセキュリティ パッチを実行するなどのカスタマイズされた追加タスクを実行できます。カスタム タスクは、ライブラリで作成してアプリケーション展開に追加できます。Application Services は、ライブラリで事前定義済みタスクも提供します。これらのタスクは、APT リポジトリと YUM リポジトリの構成、Red Hat Network へのマシンの登録、またはドメインへの参加に使用できます。
オペレーティング システム	論理テンプレートとサービスの IT 組織がサポートするオペレーティングシステムを指定します。オペレーティングシステムのリストは [オペレーティングシステム] メニューに表示され、リストに追加できます。
タグ	ブループリント エディタで、読みやすくするために論理テンプレートとサービスのリストを編成します。タグのリストは、[タグ] メニューに表示され、リストに新しいタグを追加できます。

アプリケーション プロビジョニングの主なコンポーネント。

アプリケーション	オペレーティング システムのテンプレート、アプリケーションのコンポーネント、およびそれに依存するサービス（複数の仮想マシンにまたがって配布可能）の間の関係を定義する論理展開ユニット。
アプリケーション コンポーネント	EAR ファイル、WAR ファイルなどのコンポーネントのためのテンプレートとして使用されるカスタム コード。これらは、ノードまたはサービスに対するインストール、構成、開始、更新、ロールバック、および分解の各アクションのためのカスタム スクリプト パッケージです。
アプリケーション ブループリント	アプリケーションの展開のための論理トポロジ。ブループリントは、論理ノード、対応するサービスとオペレーティングシステム、依存関係、デフォルト構成、ネットワークとストレージのトポロジ要件を含むアプリケーションの構造をキャプチャします。
ノード	ブループリントで定義されている仮想マシン。
クラスタ化ノード	ブループリントで定義されている仮想マシンのクラスタ。
ディスク	対応する仮想マシンまたはノードに追加するディスク。

プロパティ	サービスとアプリケーション コンポーネントのための構成の名前と値のペア。これらは、スクリプトにパラメータを設定し、さまざまな構成を実行するためにスクリプトが使用する変数です。たとえば、 <code>installation_path</code> プロパティ値を設定し、アプリケーション展開プロセス中にこのプロパティを使用してサービスをインストールするパスを指定するようにインストール スクリプトを構成できます。
アクション	インストール、構成、開始、更新、ロールバック、および分解の各スクリプトのライフ サイクル ステージで、インストールするサービスとアプリケーション コンポーネントに使用します。
ストレージ	クラウド プロバイダがパフォーマンス要件を満たすためにディスクを各種のストレージに配置するために提供する領域。
展開プロファイル	クラスタ サイズ、CPU、メモリ、クラウド テンプレート、ネットワークを含むブルー プリントの展開設定の集合。
論理ネットワーク	ネットワークの論理的な抽象化として作成されたエンティティ。論理ネットワークは、アプリケーションのノード間のネットワーク位置を示すために使用するクラウドに依存しない抽象化です。論理ネットワークは、展開プロファイルの構成中にクラウド ネットワークに動的にマッピングされます。
クラウド ネットワーク	クラウド展開環境内で定義されるネットワーク。
実行プラン	仮想マシンを作成する順序を表示するタスク プランおよびライブラリとアプリケーション コンポーネントのためのアクション スクリプトがインストール、構成、開始、および更新されます。エージェントがタスクを実行する順序は、展開実行計画で定義されます。
分解	クラウドからの展開済み仮想マシンの削除。たとえば、vCloud Director vApp と関連付けられている仮想マシン、vRealize Automation 仮想マシン、および vCenter Server の関連付けられている仮想マシン、または展開済みアプリケーションの Amazon EC2 インスタンスをクラウド環境から分解できます。分解プロセスが完了すると、これらの仮想マシンが使用していたストレージ、IP アドレスなどの他のすべてのリソースは、それぞれのプールに戻されます。

Application Services のインストールおよび構成

2

Application Services には、本製品の基本概念を理解した上で使用できるように、事前定義済みのサンプル アプリケーション、サービス、仮想マシン テンプレートが含まれています。以下のタスクを実行して、Application Services をインストールし、vRealize Automation または Amazon EC2 環境を介して、事前定義済みサンプル アプリケーションを vCloud Director および vSphere に展開します。

開始する前に

[「Application Services の概要 \(P. 13\)」](#)に記載されている Application Services のプロビジョニング ワークフローを理解しておく必要があります。

手順

- 1 Application Services 仮想アプライアンスをインストールおよび設定します。

[「Application Services のインストール準備 \(P. 23\)」](#) および [「Application Services アプライアンスの起動 \(P. 29\)」](#) を参照してください。

- 2 プロキシを使用するように Application Services を構成します。

このタスクは、アプリケーションが自社ファイアウォールの外からファイルをダウンロードする必要がある場合に実行します。

[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。

- 3 Application Services Web インターフェイスにログインして、製品の機能を理解します。

[「Application Services にログインする \(P. 45\)」](#) および [「Application Services Web インターフェイスの使用 \(P. 46\)」](#) を参照してください。

- 4 クラウド プロバイダを登録します。

クラウド プロバイダを登録する際、クラウド環境で Application Services を次のいずれかのコンポーネントにマップします。

クラウド プロバイダのタイプ	参照先トピック
vCloud Director インスタンスと組織	「vCloud Director クラウド プロバイダとテンプレートの登録 (P. 80)」
vRealize Automation のビジネス グループ	「vRealize Automation クラウド プロバイダとテンプレートの登録 (P. 58)」
Amazon EC2 リージョン	「Amazon EC2 のクラウド プロバイダとテンプレートの登録 (P. 90)」

- 5 登録したクラウド プロバイダに展開環境を作成します。

Application Services 展開環境を、クラウド環境の次のいずれかのコンポーネントにマップして、アプリケーションを展開できるようにします。

展開環境	参照先トピック
vCloud Director 内の組織 vDC	「vCloud Director 展開環境の作成 (P. 81)」
vCloud Automation Center 内の予約ポリシー	「vRealize Automation 展開環境の作成 (P. 60)」
Amazon VPC および関連付けられた可用性ゾーン	「Amazon EC2 の展開環境の作成 (P. 92)」

- 6 クラウド テンプレートを論理テンプレートにマッピングします。

クラウド テンプレートを論理テンプレートにマップすると、Application Services 論理テンプレートをクラウドの実際のテンプレートにリンクしていることになります。

[「ライブラリへの論理テンプレートの追加 \(P. 125\)」](#) を参照してください。

- 7 [展開プロファイル] ウィザードで、事前定義されたサンプル アプリケーションを展開します。

[第 18 章「事前定義済みライブラリ コンポーネントの展開 \(P. 193\)」](#) および [「展開プロファイルの作成 \(P. 157\)」](#) を参照してください。

- 8 展開のステータスを確認します。

展開を実行するとコンポーネントがインストールされ、アプリケーションの依存関係に基づいて構成されます。[「展開および更新プロセスについて \(P. 175\)」](#) を参照してください。

ユーザー インターフェイスを使用して、アプリケーションの展開のステータスをリアルタイムで確認できます。[「\[展開サマリ\] ページの使用 \(P. 174\)」](#) を参照してください。

- 9 展開エラーをトラブルシューティングします。

展開エラーが発生したら、仮想マシン固有のログと展開のログを調べて問題点をトラブルシューティングします。

仮想マシン固有のログにアクセスする方法については、[「展開された仮想マシンの詳細情報とアプリケーションの実行プランの表示 \(P. 211\)」](#) を参照してください。問題を解決する方法については、『Application Services トラブルシューティング』のドキュメントを参照してください。

Application Services のインストール

Application Services 6.1 をインストールするには、vCloud Director または vSphere vCenter Server のいずれかで仮想アプライアンスを作成および展開します。

この章では次のトピックについて説明します。

- [Application Services のインストール準備 \(P. 23\)](#)
- [Application Services アプライアンスの起動 \(P. 29\)](#)
- [Application Services Web インターフェイスに接続する際に問題が発生した場合のトラブルシューティング \(P. 33\)](#)
- [darwin_user アカウントのロック解除 \(P. 33\)](#)
- [Application Services の再起動 \(P. 34\)](#)
- [外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)](#)
- [vRealize Automation への Application Services の登録 \(P. 35\)](#)

Application Services のインストール準備

Application Services アプライアンスのインストールを開始する前に、コンピューティング環境がハードウェアとソフトウェアのシステム要件を満たしていることを確認してください。

Application Services の場合、vCloud Director で特定の構成設定を使用することが必要です。以前インストールした vCenter Server および vCloud Director サーバの構成については、これらのサーバで Application Services と連携する設定が使用されていることを確認してください。

- [Application Services のシステム要件 \(P. 24\)](#)

Application Services の実行元にする仮想アプライアンスは、特定のハードウェアとソフトウェアの要件を満たす必要があります。また、特定のポートを Application Services 用に開く必要があります。
- [vCenter Server クラスタ構成のセットアップ \(P. 25\)](#)

vCloud Director または vSphere に Application Services をインストールするには、特定の条件を満たすように vCenter Server クラスタ構成のストレージとネットワークをセットアップする必要があります。
- [vCloud Director の Application Services のセットアップ \(P. 27\)](#)

vCloud Director を正常にインストールするには、Application Services で仮想アプライアンスを構成する必要があります。以前にインストールした vCloud Director のバージョンがある場合は、Application Services を最適化するために構成設定を確認する必要があります。

Application Services のシステム要件

Application Services の実行元にする仮想アプライアンスは、特定のハードウェアとソフトウェアの要件を満たす必要があります。また、特定のポートを Application Services 用に開く必要があります。

vCloud Director のディスク領域とメモリーの要件

Application Services には、少なくとも 20 GB のディスク領域、2 GB メモリー、および 1 vCPU (2 GHz の速度) が必要です。



注意 一部の組織 vDC の場合、vCloud Director は自身のセットアップを基に、仮想 CPU を 0.24 GHz にデフォルト設定していることがあります。vCloud Director 環境内でこの設定がデフォルトになっている場合、Application Services vApp が展開されている組織 vDC の vCPU 速度を 2 GHz に設定する必要があります。そうしないと、Application Services 仮想アプライアンスのパフォーマンスに影響が及びます。

ポート要件

特定のポートを Application Services 仮想マシン用に開く必要があります。

ポート	接続タイプ
TCP ポート 8443	HTTPS を介した、Application Services ユーザー インターフェイス接続用の外部ポート。
TCP ポート 8080	HTTP を介した、Application Services ユーザー インターフェイス接続用のオプションのポート。
TCP ポート 80	vami-lighttpd 用の外部ポート。
TCP ポート 5671	vFabric RabbitMQ 用の外部ポート。
TCP ポート 443	クラウド環境に接続するための、Application Services ユーザー インターフェイス用の外部ポート
TCP ポート 22	外部 SSH 接続用のオプションのポート。

お使いのリリースの最新のポート要件についての詳細は、<https://www.vmware.com/support/pubs/vcac-pubs.html> にある vRealize Automation のドキュメントのサポート マトリックスを参照してください。

Web インターフェイスのサポート

Application Services では、次の Web ブラウザがサポートされます。お使いのリリースの現在のバージョンの要件については、<https://www.vmware.com/support/pubs/vcac-pubs.html> にある vRealize Automation のドキュメントのサポート マトリックスを参照してください。

- Internet Explorer
- Mozilla Firefox
- Chrome

仮想化ソフトウェアの要件

Application Services を使用するには、次の VMware 製品をインストールしてセットアップする必要があります：

- vSphere ESXi。 <https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html> にある vSphere のドキュメントを参照してください。
- vCenter Server。 <https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html> にある vSphere のドキュメントを参照してください。
- vCloud Director 5.5。 詳細については、 https://www.vmware.com/support/pubs/vcd_pubs.html にある vCloud Director のドキュメントを参照してください。

- vRealize Automation 6.2。詳細については、<https://www.vmware.com/support/pubs/vcac-pubs.html> にある vRealize Automation のドキュメントを参照してください。

お使いのリリースの最新のシステム要件についての詳細は、<https://www.vmware.com/support/pubs/vcac-pubs.html> にある vRealize Automation のドキュメントのサポート マトリックスを参照してください。

Application Services ライブラリの仮想マシン テンプレートのサポート対象オペレーティング システム

カスタム仮想マシン テンプレートを作成して Application Services ライブラリで使用する場合は、サポート対象オペレーティング システムが次の製品で使用できることを確認します。

- vCloud Director。[[vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)](#)] を参照してください。
- vRealize Automation。[[vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 \(P. 50\)](#)] を参照してください。
- Amazon EC2。[[Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 \(P. 88\)](#)] を参照してください。

注意 連邦情報処理規格 (FIPS) が有効な Linux と Windows の仮想マシン テンプレートはサポートされていません。

vCenter Server クラスタ構成のセットアップ

vCloud Director または vSphere に Application Services をインストールするには、特定の条件を満たすように vCenter Server クラスタ構成のストレージとネットワークをセットアップする必要があります。

開始する前に

- お使いのリリースの現在のバージョンの要件については、<https://www.vmware.com/support/pubs/vcac-pubs.html> にある vRealize Automation のドキュメントのサポート マトリックスを参照してください。
- vSphere のリソース プールや DRS が有効な vCenter Server クラスタを作成する手順について理解しておく必要があります。vSphere ドキュメント センターを参照してください。
- vSphere ESXi を実行しているホストが 1 台以上あることを確認します。
- DRS が有効な vCenter Server クラスタがあることを確認します。
- vCenter Server のインストールの各 vSphere ESXi ホストで NTP クライアントを実行していることを確認します。

Application Services が展開されているホストの時間と、仮想マシンを展開するホストの時間を同期します。時間を同期しないと、プロビジョニング済み仮想マシンと通信するときに Application Services サーバで問題が発生することがあります。

ホストの時間を同期するには、**Configuring Time** システムの各 ESXi ホストで NTP サーバの vCloud Director オプションを設定します。

手順

- 1 DRS が有効なクラスタを作成します。

2 データストア要件とネットワーク要件の構成設定を確認します。

オプション	アクション
クラスタに複数のホスト	<ul style="list-style-type: none"> ■ すべてのホストに少なくとも 1 つの共有データストア（共有 LUN、NFS など）があることを確認します。 ■ クラスタのすべてのホストが少なくとも 1 つの一般的なネットワークに接続されていることを確認します。 ■ 仮想マシンのライブ移行をサポートするには、クラスタのすべてのホストが同一であることを確認します。
ネットワークまたは SAN ストレージ	<ul style="list-style-type: none"> ■ ホストに共有ストレージがあることを確認します。 <p>vCloud Director と Application Services ではローカル ストレージを使用できますが、共有ストレージがあれば今後のスケーラビリティを確保できます。</p> <ul style="list-style-type: none"> ■ ホストに少なくとも 1 つのネットワークがあることを確認します。

次に進む前に

vSphere の場合は、Application Services アプライアンスを展開します。[\[vSphere への Application Services アプライアンスの展開 \(P. 26\)\]](#) を参照してください。

vCloud Director の場合は、vApp 設定を構成して Application Services をインストールします。[\[vCloud Director の Application Services のセットアップ \(P. 27\)\]](#) を参照してください。

vSphere への Application Services アプライアンスの展開

Application Services を vSphere にインストールするには、OVF テンプレートを展開する必要があります。

OVF テンプレートの場合は、vSphere Client マシンからアクセスできるすべてのローカル ファイルシステム、またはリモート Web サーバから展開できます。ローカル ファイルシステムには、ローカル ディスク、リムーバブル メディア、共有ネットワーク ドライブを含めることができます。

開始する前に

- vSphere の [\[Application Services のインストール準備 \(P. 23\)\]](#) に記載された前提条件をすべて満たしていることを確認します。
- Application Services OVF テンプレートを展開に使用できる必要があります。vSphere Client への OVF テンプレートの展開方法を理解しておく必要があります。vSphere のドキュメントを参照してください。

手順

- 1 vSphere Client にログインします。
- 2 [ファイル] - [OVF テンプレートの展開] を選択します。
[OVF テンプレートの展開] ウィザードが開きます。
- 3 クラスタ、リソース プール、ストレージの設定を構成します。
- 4 IP アドレスのプール内のネットワークにテンプレートをマップします。
- 5 [展開後にパワーオン] チェック ボックスをオンにします。
- 6 デプロイ設定を確認して、[終了] をクリックします。

Application Services 仮想マシンが展開され、パワーオンされます。

次に進む前に

Application Services アプライアンスを初期化します。[\[Application Services アプライアンスの起動 \(P. 29\)\]](#) を参照してください。

vCloud Director の Application Services のセットアップ

vCloud Director を正常にインストールするには、Application Services で仮想アプライアンスを構成する必要があります。以前にインストールした vCloud Director のバージョンがある場合は、Application Services を最適化するために構成設定を確認する必要があります。

開始する前に

- Application Services の vCenter Server 設定を構成します。[「vCenter Server クラスタ構成のセットアップ \(P. 25\)」](#) を参照してください。
- 組織とカタログを作成する手順について理解しておく必要があります。最新の vCloud Director ドキュメントを参照してください。
- vCloud Director 管理者が、カタログにテンプレートをアップロードしていることを確認します。

注意 Application Services がアプリケーションの展開を実行する組織から、異なる vCloud 組織に含まれるカタログにテンプレートがアップロードされている場合は、そのカタログとテンプレートを公開する必要があります。カタログに正しい属性が設定されており、ユーザーがアクセスできることを確認します。

- Application Services が使用できる IP アドレスの外部プールと直接接続するネットワークが構成されていることを確認します。プールの IP アドレス数を判断するには、vCloud Director 管理者にお問い合わせください。

手順

- ◆ vCloud Director では、Application Services に個別の組織を割り当てます。

次に進む前に

vCloud Director のセットアップを確認します。[「vCloud Director 環境を確認する \(P. 27\)」](#) を参照してください。

カスタム仮想マシンのテンプレートを作成または更新する場合は、[「vCloud Director での Windows 仮想マシン テンプレートの作成 \(P. 73\)」](#) および [「vCloud Director での Linux 仮想マシン テンプレートの作成 \(P. 77\)」](#) を参照してください。

vCloud Director 環境を確認する

以前にインストールしたバージョンの vCloud Director がある場合は、vApps が仮想マシンおよび外部ネットワーク IP アドレスと通信できることを確認します。

Application Services は、NAT 経由の直接接続ネットワークへの仮想マシンの展開をサポートします。展開済み仮想マシンと Application Services サーバをネットワーク経由で通信させるには、IPPOOL アドレッシングを使用します。

注意 直接接続ネットワークへの DHCP アドレッシングはサポートされません。

開始する前に

- Application Services の vCenter Server 設定を構成します。[「vCenter Server クラスタ構成のセットアップ \(P. 25\)」](#) を参照してください。
- vCloud Director のカスタマイズ手順について理解しておく必要があります。最新の vCloud Director のドキュメントと当社のナレッジベースの記事 kb.vmware.com/kb/2005829 および kb.vmware.com/kb/2034092 を参照してください。
- ブラウザと連携する vCloud Director の VMRC プラグインがインストールされていることを確認します。互換性のある Web ブラウザの詳細については、[「Application Services のシステム要件 \(P. 24\)」](#) を参照してください。
- 必要な TCP ポートが Application Services 仮想マシン用に開いていることを確認します。[「Application Services のシステム要件 \(P. 24\)」](#) を参照してください。

手順

- 1 vCloud Director ユーザー インターフェイスで、1 つの仮想マシンを持つ vApp を作成します。
- 2 Application Services クラウド プロバイダにマッピングした vCloud 組織内の仮想マシンの vCloud ネットワーク構成で、これらの仮想マシンが Application Services アプライアンスに接続できるようになっていることを確認します。
- 3 vApp を展開します。
- 4 VMRC プラグインを使用して仮想マシンに接続し、外部 IP を ping できることを確認します。
- 5 展開済み仮想マシンが Application Services アプライアンスにアクセスできるように、TCP ポートを開きます。

次に進む前に

Application Services アプライアンスをダウンロードして展開します。[\[vCloud Director への Application Services アプライアンスの展開 \(P. 28\)\]](#) を参照してください。

vCloud Director への Application Services アプライアンスの展開


Application Services を vCloud Director にインストールするには、アプライアンスをダウンロードする必要があります。

vApps の追加については、vCloud Director のドキュメントを参照してください。

開始する前に

vCloud Director 用の [「Application Services のインストール準備 \(P. 23\)」](#) に記載されている前提条件を完了していることを確認します。

手順

- 1 Application Services アプライアンスをダウンロードして、両方のファイルを名前を変えずに同じフォルダ内に保存します。
アプライアンスは次のファイルで構成されています。
ApplicationDirector-<VersionNumber>-<*>_OVF10.ovf
ApplicationDirector-<VersionNumber>-<*>-system.vmdk
- 2 <VersionNumber> をアプライアンスのビルド番号に置き換えます。
- 3 vCloud Director にログインして、Application Services の展開先にする組織 vDC を選択します。
- 4 [カタログ] ビューを選択して、[vApp テンプレート] タブをクリックします。
- 5 [アップロード] () をクリックします。
- 6 [OVF をテンプレートとしてアップロード] ウィンドウで、必要な情報を入力します。
- 7 証明書の警告ポップアップ ウィンドウで [OK] をクリックして、アプライアンスのアップロードを続行します。
ファイル サイズが大きいことと、ネットワーク接続速度に依存することから、アップロード プロセスには長時間を要することを見込んでください。
- 8 アップロードしたテンプレートを右クリックして、[マイ クラウドへの追加]を選択します。
- 9 プロンプトに従って vApp を追加します。
- 10 [マイ クラウド]に移動します。
- 11 新たに追加した vApp を右クリックして、[開始]をクリックします。

次に進む前に

Application Services アプライアンスを初期化します。[「Application Services アプライアンスの起動 \(P. 29\)」](#) を参照してください。

Application Services アプライアンスの起動

新しく追加した vCloud Director の vApp または vSphere 仮想マシンを起動して、Application Services が正常にインストールされていることを確認し、残りのセットアップ プロシージャを完了する必要があります。

手順

- 1 [Application Services アプライアンスの構成 \(P. 29\)](#)
Application Services アプライアンスを初めて起動するときには、Application Services のシリアル番号を入力し、root ユーザーと darwin_user のパスワードを作成する必要があります。
- 2 [アプライアンスと vRealize Automation サーバの登録 \(P. 30\)](#)
Application Services アプライアンスを vRealize Automation サーバに登録します。登録すると、vRealize Automation ユーザー、ビジネス グループ、および Application Services のテナントを使用したり、Application Services ユーザー ロールを割り当てたりすることができます。
- 3 [事前定義済みサンプル コンテンツのインポート \(P. 31\)](#)
Application Services アプライアンスおよび vRealize Automation サーバに登録したら、vRealize Automation テナントに事前定義済みサンプル コンテンツを任意でインストールできます。
- 4 [Application Services Web インターフェイスを開く \(P. 31\)](#)
アプライアンスを構成して vRealize Automation サーバに登録し、事前定義済みサンプル オブジェクトを任意でインポートしたら、Web インターフェイスを開いて、検索や作業を行うことができます。

Application Services アプライアンスの構成

Application Services アプライアンスを初めて起動するときには、Application Services のシリアル番号を入力し、root ユーザーと darwin_user のパスワードを作成する必要があります。

vCloud Director Web コンソールを使用して、パワーオン済みの仮想マシンにアクセスすることもできます。vmware-vmrc プラグインのコピーが Web ブラウザにインストールされていることを確認します。【vCloud API プログラミング】を参照してください。

Application Services アプライアンスを起動する手順の一部として、アプライアンスを vRealize Automation サーバとテナントに登録して、事前定義済みのサンプル コンテンツをテナントにインポートするかどうかを選択します。

新しい Application Services リリースにアップグレードする場合は、[第4章「Application Services のアップグレード \(P. 37\)」](#)を参照してください。

開始する前に

- ブラウザと連携する vCloud Director の VMRC プラグインがインストールされていることを確認します。互換性のある Web ブラウザの詳細については、[「Application Services のシステム要件 \(P. 24\)」](#)を参照してください。
- vSphere クライアントが Web ブラウザの要件を満たしていることを確認します。VMware vSphere のドキュメントを参照してください。
- [「Application Services アプライアンスの起動 \(P. 29\)」](#)。

手順

- 1 Application Services アプライアンスを開きます。
 - a vCloud Director の [マイ クラウド] ビューで、Application Services アプライアンスに相当するコンソール列の下にあるイメージをクリックして、VMRC コンソール内をクリックします。
アプライアンスへの接続が確立され、vApp コンソール ビューが開くまで数秒かかることがあります。
 - b vSphere クライアントで、パワーオン済み仮想マシンを見つけて、[コンソール] タブをクリックします。
- 2 プロンプトで Application Services シリアル番号を入力して、Enter キーを押します。

- 3 root ユーザー アカウントのパスワードを入力して、Enter キーを押します。
- 4 プロンプトでパスワードを再入力して確定し、Enter キーを押します。
- 5 darwin_user アカウントのパスワードを入力して、Enter キーを押します。
darwin_user アカウントのパスワードは記憶しやすいものにしてください。再びこのパスワードを入力するようにプロンプトで求められたら、3 回だけ正しいパスワードの入力を試行できます。
- 6 プロンプトでパスワードを再入力して確定し、Enter キーを押します。
- 7 プロンプトが表示された場合、要求に応じて SSH ログインを有効にします。

次に進む前に

[「アプライアンスと vRealize Automation サーバの登録 \(P. 30\)」](#)。

アプライアンスと vRealize Automation サーバの登録

Application Services アプライアンスを vRealize Automation サーバに登録します。登録すると、vRealize Automation ユーザー、ビジネス グループ、および Application Services のテナントを使用したり、Application Services ユーザーロールを割り当てたりすることができます。

開始する前に

[「Application Services アプライアンスの構成 \(P. 29\)」](#)。

手順

- 1 vRealize Automation サーバの URL をプロンプトで入力して、Enter キーを押します。
たとえば、**https://vcac.it.mycompany.com** または **https://192.0.2.115:443** の形式でセキュアな URL を入力します。
- 2 vRealize Automation システム管理者アカウントのユーザー名を入力して、Enter キーを押します。
- 3 vRealize Automation システム管理者アカウントのパスワードを入力して、Enter キーを押します。
- 4 vRealize Automation サーバへの登録が失敗すると、登録のスキップまたは再試行のいずれかを求められます。
- 5 登録に成功すると、事前定義済みのサンプル コンテンツを Application Services にインポートするようにプロンプトで指示されます。「Yes」を表す **Y** を入力するか、No を表す **N** を入力して、Enter キーを押します。

Enter	説明
はい	「Yes」の場合、事前定義済みのサンプル コンテンツを指定のテナントとビジネス グループにインポートします。インポート後、テナント内のすべてのビジネス グループでコンテンツを共有できます。
いいえ	「No」の場合、事前定義済みのサンプル コンテンツをインポートしないでください。

注意 起動スクリプトを使用して、バックグラウンドでその他の必要なソフトウェアをインストールし、ソフトウェアの処理を開始するまでにかかる時間は 1 分未満です。その後、管理者ユーザー アカウントのパスワードの設定を求められます。この間に、システムが応答していないように見えることがあります。

次に進む前に

(任意) [「事前定義済みサンプル コンテンツのインポート \(P. 31\)」](#)。

事前定義済みサンプル コンテンツのインポート

Application Services アプライアンスおよび vRealize Automation サーバを登録したら、vRealize Automation テナントに事前定義済みサンプル コンテンツを任意でインストールできます。

事前定義済みサンプル コンテンツをインポートする前に、CLI へのログインに必要な情報をすべて入力するように求められます。

開始する前に

- [「アプライアンスと vRealize Automation サーバの登録 \(P. 30\)」](#) .
- テナント ユーザー アカウントにすべての Application Services ロールが割り当てられていることを確認します。ユーザーへのロールの割り当てについては、vRealize Automation の「テナント管理」を参照してください。
- テナント ユーザーがビジネス グループに属していることを確認します。

手順

- 1 テナントの URL 名を入力し、Enter キーを押します。
vRealize Automation のシステム管理者は、テナントを追加するときに URL 名を指定します。
- 2 テナントにアクセスする際に使用するアカウント名を入力し、Enter キーを押します。
ユーザー アカウントには、すべての Application Services のロールも割り当てられている必要があります。
- 3 テナント ユーザーのパスワードを入力し、Enter キーを押します。
- 4 テナント ユーザーが含まれるビジネス グループの名前を入力し、Enter キーを押します。
ビジネス グループ名にスペースが含まれている場合は、名前全体を二重引用符で囲みます。
- 5 CLI を使用したログインに失敗すると、再試行を求められます。**Y** または **N** を入力し、Enter を押します。

Enter	説明
はい	ログインを再試行します。手順 1 から 手順 4 の実行を求められます。
いいえ	ログインを再試行しません。事前定義済みサンプル コンテンツはインポートされません。

- 6 CLI を使用したログインに成功すると、事前定義済みサンプル コンテンツがインポートされ、キーを押して続行するように促されます。任意のキーを押します。

次に進む前に

[「Application Services Web インターフェイスを開く \(P. 31\)」](#) .

Application Services Web インターフェイスを開く

アプライアンスを構成して vRealize Automation サーバに登録し、事前定義済みサンプル オブジェクトを任意でインポートしたら、Web インターフェイスを開いて、検索や作業を行うことができます。

開始する前に

- [「アプライアンスと vRealize Automation サーバの登録 \(P. 30\)」](#)。

手順

- 1 Application Services 管理者ユーザー アカウントのパスワードを入力して、Enter キーを押します。
- 2 プロンプトでパスワードを再入力して確定し、Enter キーを押します。
bootup スクリプトにより、必要なサービスが開始され、Application Services サーバにアクセスするための URL が表示されます。

- 3 サポートされているブラウザで Application Services サーバに移動します。
URL 形式は **https://<Application_Services_IP_or_hostname>:8443/darwin/org/<tenantURL>** で、
<tenantURL> はテナント URL 名を表します。
- 4 vRealize Automation テナント管理者、または Application Services ロールが 1 つ以上割り当てられたテナント
ユーザーとしてログインします。ユーザーは少なくとも、テナント内の 1 つのビジネス グループに属する必要があります。

ブラウザで Web インターフェイスが開きます。「[Application Services Web インターフェイスの使用 \(P. 46\)](#)」を参照してください。

注意 vCloud Director に展開した Application Services アプライアンスをシャットダウンする場合は、vCloud Director ユーザー インターフェイスを使ってシャットダウンしてください。vCenter Server を使用して Application Services アプライアンスをシャットダウンしないでください。

次に進む前に

ログインを何回か試行して失敗したために darwin_user がロックされた場合は、アカウントをロック解除して手順を進めます。「[darwin_user アカウントのロック解除 \(P. 33\)](#)」を参照してください。

Application Services にログインして、製品の機能を理解しておく必要があります。「[Application Services にログインする \(P. 45\)](#)」および「[Application Services Web インターフェイスの使用 \(P. 46\)](#)」を参照してください。

クラウド環境のクラウド プロバイダを登録します。「[vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)](#)」、「[vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)](#)」、または「[Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)](#)」を参照してください。

Application Services Web インターフェイスに接続する際に問題が発生した場合のトラブルシューティング

複数の状況で、Application Services Web インターフェイスにアクセスしようとする接続の問題が発生する可能性があります。

表 3-1. 一般的な接続エラー

エラー	可能性のある原因	考えられる解決策
Application Services 仮想アプライアンスに、有効な IP アドレスまたはネットワークが存在しない。	<p>ネットワーク エラーの場合は次の問題が考えられます。</p> <ul style="list-style-type: none"> ■ 組織 vDC ネットワークが適切に構成されていない。 ■ [仮想マシンの構成] ウィザードで、ネットワーク設定が指定されていない。 ■ [仮想マシンの構成] ウィザードの IP 割り当ての設定で、IP アドレスが指定されていない。 ■ 固定 IP アドレスが使用できない。 ■ DHCP サーバが使用できない。 	ネットワークの問題を解決する必要があります。
Application Services 仮想アプライアンスが、起動時に有効な IP アドレスを一時的に取得できない。または、IP アドレスが起動後に変更される。	<p>ネットワーク エラーの場合は次の問題が考えられます。</p> <ul style="list-style-type: none"> ■ ネットワークに問題が発生した。 ■ 固定 IP アドレスが使用できない。 ■ DHCP サーバが使用できない。 ■ IP アドレスが明示的に変更された。 	<p>アプライアンス仮想マシンで、ルート権限を使用して次のコマンドを実行します。</p> <pre>/home/darwin/tools/darwin_util.sh -a AUTO</pre>
Application Services 仮想アプライアンスで、vFabric tc Server サービスにエラーが発生する。	<p>仮想アプライアンスを適切にシャットダウンせずに再起動した。</p> <p>tc Server サービスが正しく起動または再起動されなかった。</p>	<pre>/home/darwin/tcserver/darwin/logs/catalina.out</pre> <p>にあるログファイルでエラーの詳細を確認し、必要に応じて VMware テクニカル サポートにお問い合わせください。</p>

darwin_user アカウントのロック解除

SSH クライアントで **darwin_user** アカウントを使用してログインするときに 3 回正しいパスワードを入力しないと、**darwin_user** アカウントがロック アウトされます。

開始する前に

- Application Services アプライアンスの VMRC コンソールにログインしていることを確認します。
ルート アカウントの SSH は無効です。
- ルート権限があることを確認します。

手順

- 1 VMRC コンソールで、ルート ユーザーとしてログインします。
- 2 シェル プロンプトで失敗したログイン回数を確認します。

```
faillog -u darwin_user
```

- 3 失敗したアカウントをロック解除します。

```
faillog -u darwin_user -r
```

Application Services の再起動

場合によっては、Application Services の再起動が必要になることがあります。

手順

- 1 SSH クライアントまたは vCloud Director コンソールを使用して、Application Services 仮想マシンにログインします。
- 2 Application Services を再起動します。
 - **darwin_user** アカウントでログインした場合は、**sudo service vmware-darwin-tcserver restart** と入力します。
 - **root** アカウントでログインした場合は、**service vmware-darwin-tcserver restart** と入力します。

外部 URL 用にプロキシを使用するように Application Services を構成する

プライベート クラウドでアプリケーションを展開するためだけに、Application Services を使用する場合でも、展開によっては、企業ファイアウォール外部から URL にアクセスする必要があります。たとえば、アクション スクリプトによって、オープンソースの Web サイトからアプリケーションの部分をダウンロードする場合があります。そのような場合は、プロキシを使用するように Application Services を構成できます。

サービスやアプリケーションを作成する前、または既存の事前定義済みサンプル サービスやアプリケーションを展開する前に、このタスクを完了する必要があります。

Application Services には、プロキシを定義しない **darwin_global_noproxy.conf** というプロキシ ファイルも含まれています。展開環境でプロキシが不要な場合は、このファイルを **global_conf** プロパティ値に指定できます。Amazon EC2 展開環境では、アプリケーションを展開するためのプロキシは必要ありません。

現在、プロキシのサポートはグローバル レベルから展開環境レベルに拡大されています。詳細については、[\[vRealize Automation 展開環境の作成 \(P. 60\)\]](#) を参照してください。

開始する前に

- **アプリケーション アーキテクト** および **アプリケーション カタログ管理者** として Application Services にログインします。
- Application Services がインストールされている仮想マシンへのアクセス権があること、および **darwin_user** アカウントでオペレーティング システムにログインするためのパスワードがあることを確認します。このパスワードはインストール時に設定したものです。[\[Application Services アプライアンスの起動 \(P. 29\)\]](#) を参照してください。
- 新しいサービスの作成手順を理解しておく必要があります。[\[ライブラリへのサービスの追加 \(P. 109\)\]](#) を参照してください。

手順

- 1 vCloud Director または vRealize Automation の場合は、プロキシを使用するように Application Services 仮想アプライアンスを構成します。
 - a **darwin_user** として Application Services 仮想アプライアンスにログインします。
 - b **su** と入力して、root ユーザーに切り替えます。

- c テキスト エディタでファイル `/home/darwin/tcserver/darwin/webapps/darwin/conf/darwin_global.conf` を開きます。
 - d プロキシ IP とポート情報を更新して、ファイルを保存します。
プロキシ URL の形式は `http://<proxy>:<PortNumber>` です。
`yum update` または `install` コマンドを使用してリポジトリにアクセスする必要がある既存の事前定義済みアプリケーションまたはライブラリ サービスは、このファイルで指定されたプロキシを使用します。事前定義済みサービスとアプリケーションには、このプロキシへのアクセスに必要なスクリプトが含まれています。
- 2 アプリケーション カタログ管理者として Application Services にログインして、新しいサービスまたは既存のサービスのプロキシ固有のプロパティとスクリプトを定義します。
- a 新しいサービスの場合は、サービスをライブラリに追加します。
 - b 新しいサービスまたは既存のサービスで、`global_conf to the service` というプロパティをタイプ Content のサービスに追加して、値を **`https://<DarwinServerIP>:8443/darwin/conf/darwin_global.conf`** と定義します。



注意 `darwin_global.conf` をファイル ソースとして使用するサービス スクリプトによって、`http_proxy`、`https_proxy`、または `ftp_proxy` プロパティに値を追加すると、そのサービス スクリプトが実行されたときに、アプリケーション内の既存のプロキシ情報がこれらの構成済みプロパティによってオーバーライドされます。

- c プロキシが必要なサービスの各アクション スクリプトの冒頭に次の行を追加します。

```
# Import global_conf
. $global_conf
```
 - d サービスを保存します。
- 3 アプリケーション アーキテクトとして Application Services にログインし、アプリケーション プループリントでプロキシ固有のプロパティとスクリプトを定義して、アプリケーション プループリントの作成時にプロキシを使用するようにアプリケーションを構成します。
- a アプリケーション コンポーネントの場合、`global_conf` プロパティをタイプ Content のアプリケーション コンポーネントに追加して、値を **`https://<DarwinServerIP>:8443/darwin/conf/darwin_global.conf`** と定義します。
 - b プロキシが必要なアプリケーション コンポーネントの各アクション スクリプトの冒頭に次の行を追加します。

```
# Import global_conf
. $global_conf
```

次に進む前に

Application Services にログインして、製品の機能を理解しておく必要があります。[「Application Services にログインする \(P. 45\)」](#) および [「Application Services Web インターフェイスの使用 \(P. 46\)」](#) を参照してください。

クラウド環境のクラウド プロバイダを登録します。[「vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)」](#)、[「vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)」](#)、または [「Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)」](#) を参照してください。

vRealize Automation への Application Services の登録

Application Services を登録するときに、CLI を使用して Application Services 環境と vRealize Automation サーバ間の接続を確立することで、vRealize Automation サービス カタログにアクセスできるようにします。

通常、登録は Application Services アプライアンスの起動の一部として実行されます。登録を再確立する必要がある場合、または登録情報が変更された場合は、この手順を使用します。

Application Services と vRealize Automation の接続が確立されたら、他のユーザーが展開プロファイルを申請できるように、vRealize Automation Application Services カタログに展開プロファイルを公開できます。また、登録後に Application Services ロールをユーザーおよびユーザー グループに割り当てることもできます。

注意 有効にできる登録は 1 つだけです。複数登録しようとすると、エラー メッセージが表示されます。

開始する前に

- システム管理者として Application Services にログインします。
- darwin_user のパスワードを憶えていることを確認します。このパスワードはインストール時に設定したものです。[「Application Services アプライアンスの起動 \(P. 29\)」](#) を参照してください。
- Application Services CLI を起動します。[「リモートでの CLI の開始 \(P. 221\)」](#) を参照してください。
- vRealize Automation IaaS 環境とエンドポイントが適切に設定され、構成されている必要があります。『インストールおよび構成』および『仮想プラットフォームの IaaS 構成』を参照してください。
- SSO 管理者認証情報を使用可能にしておく必要があります。

手順

- 1 roo シェル プロンプトで、Application Services を vRealize Automation サーバに登録します。

```
register-vcac-server --componentRegistryUrl <vCACServerURL> --
ssoAdministratorUsername
<UserName> --ssoAdministratorPassword <Password>
```

componentRegistryUrl は、vRealize Automation セルフサービス カタログの URL です。

componentRegistryUrl のサンプルの URL は <https://vcac-oci-071.eng.test.com> になります。

登録には 30 秒ほどかかることがあり、完了後に「vCAC サーバが正常に登録されました」というステータス メッセージが表示されます。

- 2 (オプション) vRealize Automation サーバ情報を表示します。

```
list-vcac-info
```

- 3 (オプション) vRealize Automation サーバ情報が変更されている場合は、Application Services を vRealize Automation サーバに登録します。

```
register-vcac-server --componentRegistryUrl <vCACServerURL> --
ssoAdministratorUsername
<UserName> --ssoAdministratorPassword <Password> --update
```

- 4 (オプション) vRealize Automation サーバを Application Services から登録解除して、アプライアンスを別の vRealize Automation サーバに登録します。

```
unregister-vcac-server
```

vRealize Automation サーバが Application Services で使用できなくなった場合にのみ、このサーバを登録解除します。その後、Application Services を別の vRealize Automation サーバに登録すると、基のテンプレートが無効になったため、既存のクラウド プロバイダと展開環境は使用できなくなります。また、新旧の vRealize Automation サーバのビジネス グループが一致しない場合、既存の Application Services の内容へのアクセスはすべて失われます。

次に進む前に

Application Services を vRealize Automation に登録したら、クラウド プロバイダとテンプレートを登録します。[「vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)」](#) を参照してください。

Application Services のアップグレード

標準的なアップグレード プロセスを使用して、Application Services インストールを新しいリリースにアップグレードできます。

以前の Application Services リリースから現在のリリースにアップグレードできます。以前のバージョン (Application Director 6.0 または 6.0.1) から Application Services 6.2 にアップグレードするには、6.0 から 6.0.1、6.1、6.2 にこの順序でアップグレードします。詳細については、以前のリリースのドキュメントを参照してください。

注意 vCloud Application Director 6.0.1 から Application Services 6.1 に移行できます。その後、Application Services インストールを以降のリリースにアップグレードできます。Application Director 6.0.1 から Application Services 6.1 への移行の詳細については、6.1 バージョンの『Application Services の使用』のドキュメントを参照してください。

この章では次のトピックについて説明します。

- [Application Services のアップグレード \(P. 37\)](#)
- [Application Services のアップグレードのロールバック \(P. 38\)](#)

Application Services のアップグレード

Application Services アップグレード インストール プログラムをダウンロードして実行すると、Application Services のインストールをアップグレードできます。

Application Services アップグレード インストール プログラムをダウンロードして実行するには、ルート権限が必要です。

注意 Application Services をアップグレードする前に、vRealize Automation 6.2.x にアップグレードする必要があります。

アップグレード中は、インストールおよびアンインストール操作により Application Services サーバが停止されます。

開始する前に

- Application Services の現在のインストール済みバージョンが 6.1.xであることを確認します。
- Application Services サーバで、進行中または実行中の操作がないことを確認します。
- Application Services サーバにログインしていることを確認します。
- vRealize Automation 6.2.x にアップグレードしていることを確認します。

手順

- 1 VMware ダウンロード サイトに移動します。
- 2 **ApplicationServices-`<version-build>`_Upgrade_Installer.tgz** という名前のアップグレード ファイルをダウンロードします。

- 3 **ApplicationServices-<version-build>Upgrade_Installer.tgz** を vCloud Automation Center Application Services 6.1.x がインストールされているマシンにコピーします。
- 4 **ApplicationServices-<version-build>Upgrade_Installer.tgz** ファイルを展開する新しいディレクトリ (たとえば、**my_appservices_upgrade**) を作成します。
- 5 **ApplicationServices-<version-build>Upgrade_Installer.tgz** ファイルを新しいディレクトリ (たとえば、**my_appservices_upgrade**) に展開します。
tar xvfz ApplicationServices-<version-build>Upgrade_Installer.tgz
- 6 アップグレード インストーラを実行する修正権限を得るために、ユーザー認証情報を **root** に切り替えます。
su root
- 7 展開したアップグレード インストーラが存在するディレクトリから、インストールのアップグレード プログラムを実行します。
./install.sh
- 8 プロンプトへ応答します。

Application Services を新しいリリースにアップグレードします。

次に進む前に

アップグレードが完了したら、以下のタスクを実行する必要があります。

- vRealize Automation を再起動してコンポーネント レジストリの更新プロセスを完了します。
- Application Services Web ユーザー インターフェイスを使用する前に、ブラウザ キャッシュをクリアします。

Application Services のアップグレードのロールバック

Application Services のインストールをアップグレード中またはアップグレード後に問題が発生した場合は、アップグレード手順を開始したバージョンにインストールをロールバックできます。

開始する前に

[「Application Services のアップグレード \(P. 37\)」](#) で説明されているアップグレード手順を実行します。

手順

- 1 アップグレードのアンインストール プログラムを実行する修正権限を得るために、ユーザー認証情報を **root** に切り替えます。
su root
- 2 展開したアップグレード インストーラが存在するディレクトリから、アップグレードのアンインストール プログラムを実行します。
uninstall.sh
- 3 プロンプトへ応答します。
- 4 ロールバックが終了したら、Application Services Web ユーザー インターフェイスを使用する前にブラウザのキャッシュを消去します。

アップグレード プロセスを開始したリリースに Application Services がロールバックされます。

ユーザーとグループの設定

Application Services のインストール後、Application Services のどのタイプのオブジェクトを誰が使用できるのか、どのタスクの実行をこれらのユーザーまたはユーザー グループに許可するのかを決定する必要があります。vRealize Automation を使用して、ユーザー グループとビジネス グループを作成し、Application Services と vRealize Automation の両方でユーザー ロール、資格、および承認を割り当てます。

ロールを特定のユーザーとユーザー グループに割り当てることにより、管理者権限を選択的に割り当てることができます。オブジェクトおよびそのユーザーやグループを同じビジネス グループに割り当てると、特定の展開環境とクラウド テンプレートへのアクセスを制限できます。ユーザー グループでは、一度に複数のユーザーのロールと資格を割り当てることができます。ビジネス グループでは、Application Services の一連のサービスとリソースを一連のユーザーに関連付けることができます。

ユーザー ロールでは、ユーザーが Application Services で実行できる機能を指定します。このような機能には、ライブラリの管理、クラウド プロバイダと展開環境の管理、アプリケーションの作成、およびアプリケーションの展開などがあります。

Application Services には次のロールがあります。

ロール	関数
アプリケーション アーキテクト	アプリケーションを作成、変更、および削除します。
アプリケーション カタログ管理者	サービス、テンプレート、オペレーティングシステム、タスク、タグを定義します。
アプリケーション クラウド管理者	リソースと展開環境を定義します。
アプリケーション公開者と展開者	<ul style="list-style-type: none"> ■ アプリケーションを vRealize Automation カタログに展開します。 ■ サービス、ライブラリ アイテム、操作を作成、更新、公開します。

Application Services のロールを割り当てるには、テナント管理者として vRealize Automation にログインする必要があります。ユーザー、ユーザー グループ、ビジネス グループの管理の詳細については、vRealize Automation 6.1 のドキュメントを参照してください。

Application Services でのテナントとビジネス グループの使用

6

Application Services と vRealize Automation は、テナントとビジネス グループを共有します。テナントは、vRealize Automation 展開環境の組織単位です。ビジネス グループはテナントに属し、一連のサービスとリソースを一連のユーザーに関連付けます。

Application Services アブライアンスは、インスタンスとも呼ばれる vRealize Automation 展開環境、およびインスタンス内のテナントに登録されます。Application Services システム管理者はアブライアンスを様々なテナントに登録できますが、一度に 1 つのテナントにだけ登録できます。

テナントには複数のビジネス グループがあります。テナント内の各ビジネス グループにはメンバー、およびアプリケーション、サービス、外部サービスなどのオブジェクトがあります。ユーザーは複数のビジネス グループとテナントのメンバーになることができます。各オブジェクトはビジネス グループによって所有されています。ビジネス グループに属するユーザーだけが、そのグループのオブジェクトを編集できます。

Application Services のアプリケーション、サービス、および他のオブジェクトはプライベートまたは共有になります。特定のビジネス グループのメンバーであるユーザーだけが、そのグループのプライベート オブジェクトを表示または編集できます。テナント内のすべてのビジネス グループの全ユーザーが共有オブジェクトを表示できます。

ユーザー ロールはその役割の一部として、ユーザーが作成できるオブジェクトを決定します。たとえば、アプリケーション アーキテクト ロールが割り当てられたユーザーは、オブジェクトの中でも特にアプリケーションを作成できます。アプリケーションを作成する際、ユーザーはビジネス グループを割り当てて、アプリケーションがプライベートか共有かを指定します。これらを指定した後、所有ビジネス グループのメンバーだけがそのアプリケーションを編集できます。アプリケーションがプライベートの場合、所有ビジネス グループのメンバーだけがそのアプリケーションを表示できます。アプリケーションが共有の場合、テナント内のすべてのビジネス グループの全メンバーがそのアプリケーションを表示できます。アプリケーションがプライベートでも共有でも、所有ビジネス グループのメンバーのみがアプリケーションを編集できます。

様々なレベルの共有によって、ブループリント内のコンポーネント、およびアプリケーション自体で行える作業が決まります。アプリケーション ブループリントを構築したログインユーザーがブループリントを編集するには、アプリケーションの所有ビジネス グループに属する必要があります。論理テンプレート、サービス、外部サービス、タスク、成果物、成果物リポジトリはすべてビジネス グループに属し、プライベートまたは共有にすることができます。異なるビジネス グループに属する共有ブループリント コンポーネントを選択する場合、これらのコンポーネントは表示できても変更することはできません。コンポーネントの所有ビジネス グループに属している場合にだけ、コンポーネントを変更できます。アプリケーションを共有する場合は、論理テンプレート、サービス、外部サービスなどのアプリケーション ブループリント内のすべてのコンポーネントを明示的に共有する必要があります。

テナントへの事前定義済みコンテンツのインポート

7

登録されている vRealize Automation インスタンスのテナントに Application Services アプライアンスの事前定義済みサンプル コンテンツをインポートできます。

Application Services アプライアンスから vRealize Automation インスタンスに事前定義済みサンプル コンテンツをインポートするには、`/home/darwin/tools/import_oob_content.sh` スクリプトを使用します。このスクリプトは Application Services アプライアンスにあります。サンプル コンテンツを複数のテナントにインポートする場合は、テナントごとにスクリプトを実行する必要があります。

ログ ファイルは `/tmp/standalone_import.log` と `/tmp/standalone_import_all.log` です。

スクリプトに英語以外の文字を入力できるようにするには、環境変数 `LC_CTYPE` を `en_US.UTF-8` に設定する必要があります。

開始する前に

- Application Services アプライアンスが vRealize Automation インスタンスに登録されていることを確認します。
- vRealize Automation インスタンスのテナント名とテナント ユーザー 認証情報を確認します。
- テナント ユーザーが事前定義済みインポートのターゲット ビジネス グループに属していて、すべての Application Services ロールを備えていることを確認します。
- SSH で Application Services アプライアンスにログインします。

手順

- 1 現在のディレクトリを `/home/darwin/tools` に設定します。
書き込み権限が必要です。
- 2 `/import_oob_content.sh` スクリプトを実行します。
スクリプトで入力を求めるプロンプトが表示されます。
- 3 テナントの URL 名を入力し、Enter キーを押します。
vRealize Automation のシステム管理者は、テナントを追加するときに URL 名を指定します。
- 4 テナントにアクセスする際に使用するアカウント名を入力し、Enter キーを押します。
ユーザー アカウントには、すべての Application Services のロールも割り当てられている必要があります。
- 5 テナント ユーザーのパスワードを入力し、Enter キーを押します。
- 6 テナント ユーザーが含まれるビジネス グループの名前を入力し、Enter キーを押します。
ビジネス グループ名にスペースが含まれている場合は、名前全体を二重引用符で囲みます。

- 7 CLI を使用したログインに失敗すると、再試行を求められます。**Y** または **N** を入力し、Enter を押します。

Enter	説明
はい	ログインを再試行します。手順 3 から 手順 6 の実行を求められます。
いいえ	ログインを再試行しません。事前定義済みサンプル コンテンツはインポートされません。

- 8 CLI を使用したログインに成功すると、事前定義済みサンプル コンテンツがインポートされます。

次に進む前に

Darwin CLI **export-package** コマンドでエクスポート パッケージを作成し、**import-package** 補助コマンドでそのパッケージから別の Application Services アプライアンスのテナントに特定のアプリケーションまたはサービスをインポートできます。**export-package** コマンドと **import-package** コマンドの詳細については、『Application Services の使用』を参照してください。

Application Services Web インターフェイスの使用

8

Application Services Web インターフェイスを使用すると、クラウド プロバイダを登録し、仮想マシンのテンプレートとサービスのライブラリを保守し、アプリケーションを作成、展開し、vRealize Automation サービス カタログに展開プロファイルを公開し、展開済みアプリケーションを更新し、展開済みアプリケーションをクラウドから分解することができます。

注意 vRealize Automation サーバの登録と、異なるインスタンス間のパッケージのインポートとエクスポートには、CLI を使用する必要があります。

CLI を使用すると、アプリケーションの展開、展開されているアプリケーションの更新、展開されているアプリケーションの分解もできます。[「一般的な CLI オプション \(P. 222\)」](#) を参照してください。

この章では次のトピックについて説明します。

- [Application Services にログインする \(P. 45\)](#)
- [Application Services Web インターフェイスの使用 \(P. 46\)](#)

Application Services にログインする

登録したテナントの SSO 認証情報、および割り当て済みの Application Services ユーザー ロールで、Application Services Web インターフェイスにログインします。

ログイン後に特定のタスクを実行できるかどうかは、ユーザー アカウントに割り当てられるロールによって異なります。表示できるオブジェクトは、ユーザーに割り当てられているビジネス グループと、そのビジネス グループにオブジェクトも関連付けられているかどうかによって異なります。どの Application Services ユーザー ロールにもログインできません。[第 5 章「ユーザーとグループの設定 \(P. 39\)」](#) を参照してください。

開始する前に

- Application Services アプライアンスが vRealize Automation インスタンスとテナントに登録されていることを確認します。
- SSO 認証情報に登録済みテナントへのアクセス権が含まれていて、少なくとも 1 つの Application Services ユーザー ロールが割り当てられていることを確認します。
- Application Services Web インターフェイスの URL があることを確認します。この Web URL は、インストールが終了したときに、Application Services をホストする仮想マシンのコンソールに表示されます。Application Services を起動するための URL と手順については、[「Application Services アプライアンスの起動 \(P. 29\)」](#) を参照してください。
- サポートされている Web ブラウザのバージョンがコンピュータにインストールされていることを確認します。[「Application Services のシステム要件 \(P. 24\)」](#) を参照してください。

手順

- 1 Web ブラウザを開いて、登録済みテナントの Application Services URL を入力します。

登録済みテナントにアクセスするには、URL の最後のテナント ID を変更します。テナント ID を指定しない場合は、Application Services のデフォルトのテナント `vsphere.local` が開きます。

たとえば、登録済みテナント `dev` には、URL `https://192.0.2.255:8443/darwin/org/dev` でアクセスできます。

- 2 必要なロールが割り当てられているユーザー アカウントの認証情報を入力します。

ロールに割り当てられている製品領域にアクセスできます。たとえば、ユーザー アカウントにアプリケーション アーキテクト ロールが割り当てられている場合に Application Services Web インターフェイスにログインすると、[アプリケーション] ページが開きます。ロールに製品領域が割り当てられていない場合、ユーザー インターフェイスは表示されますが無効になります。

次に進む前に

Application Services の機能を理解しておく必要があります。[「Application Services Web インターフェイスの使用 \(P. 46\)」](#) を参照してください。

Application Services Web インターフェイスの使用

Web インターフェイスを使用して、使用可能なほとんどすべての機能にアクセスできます。

Web ユーザー インターフェイスには、Web アプリケーションの標準的な機能が含まれています。たとえば検索ボックスでは、検索しているオブジェクトに関連したフィルタ条件を選択できます。アンダースコアが含まれる文字列を検索する場合、Application Services はアンダースコアをワイルドカードとして無視します。たとえば、`test_qe` という名前のアプリケーションと `testapp` という名前のアプリケーションがある場合、検索文字列 `test_` を検索ボックスに入力すると、両方のアプリケーションが検索結果に表示されます。

Application Services アプライアンスにログインすると、ユーザー アカウントに関連付けられたグループに応じて、アプリケーション、展開、およびクラウド環境にアクセスできます。すべてのユーザーが Application Services ライブラリに含まれるサンプル アプリケーション、サービス、タスク、論理テンプレート、外部サービス、オペレーティングシステム、およびタグを表示できます。

Application Services タイトル バーには、メイン カテゴリである [アプリケーション]、[展開]、[クラウド]、[ライブラリ]、および [クラウド] と [ライブラリ] のサブカテゴリを表示するドロップダウンメニューが用意されています。Application Services タイトル バーで別のカテゴリを選択すると、ほかのページに移動できます。

[アプリケーション] ページには、すべてのアプリケーションが体系化されて表示されます。このページで、サンプル アプリケーションを編集したり、カスタム アプリケーションを作成してクラウド環境に展開できます。[展開] ページでは、成功した展開と失敗した展開の両方が一覧表示されます。このページから [展開サマリ] ページを開いて、クラスタ化ノードの拡大/縮小、構成の変更、展開されたアプリケーションの分解または削除を実行する更新プロセスを開始できます。[クラウド] カテゴリでは、クラウド プロバイダおよび展開環境を作成できます。[ライブラリ] カテゴリでは、アプリケーションのサービス、外部サービス、ポリシー、論理テンプレート、タスク、オペレーティングシステム、タグ、成果物リポジトリ、および成果物を作成できます。ライブラリ内のサンプル サービスやその他のコンポーネントを使用して、アプリケーション ブループリントをモデル化することもできます。

Application Services のボタンの使用

Application Services インターフェイスには、特定のタスクまたは複数のタスクを実行するためのボタンが用意されています。

表 8-1. Application Services のボタン

アイコン	名前	説明
	[更新プロファイルの昇格]	更新された展開で使用できます。更新プロファイルが別の展開環境に昇格されることを示します。
	[関係の追加]	ブループリントで使用できます。ブループリント内のコンポーネント間の関係を作成します。
	[ノード アレイに変換]	ブループリントで使用できます。クラスタ化されたノードを作成します。このボタンを使用してクラスタ化されたノードを単一ノードに変換することもできます。
	[追加]	ブループリントで使用できます。[NIC] タブに表示されているノードに NIC を追加します。または、[プロパティ] タブに表示されているプロパティにアプリケーション コンポーネント プロパティを追加します。
	[リセット]	[展開プロファイル] ウィザードで使用できます。元の値に戻します。[展開プロファイルの更新] ウィザード、サービス バージョン、論理テンプレート バージョン、タスク バージョンの各ページでも使用できます。
	[タスク情報の表示]	実行プランで使用できます。アクション スクリプトのテキスト、プロパティ、ログを表示します。
	[展開]	実行プランで使用できます。クラスタ ノード内のすべての仮想マシンを表示します。
	[折りたたむ]	実行プランで使用できます。クラスタ ノード内のすべての仮想マシンを最小化します。
	[スクリプト タスクの追加]	実行プランで使用できます。アプリケーション展開にカスタムのタスクを追加します。
	[通知]	タスクが完了していない場合、またはユーザーが許可されていないタスクを実行した場合に表示されます。
	[展開]	[展開サマリ] ページのステータス ウィンドウを最大化します。
	[最小化]	[展開サマリ] ページのステータス ウィンドウを折りたたみます。
	[展開]	[展開サマリ] ページの [実行プラン ステータス] ウィンドウでプロビジョニング タスクを最大化します。
	[ログの表示]	[展開の詳細] ページまたは展開後の実行プランで使用できます。
	[アプリケーション バージョンのコピー]	既存のアプリケーションの新規バージョンを作成します。
	[クイック展開]	[アプリケーション] ページおよびアプリケーション バージョンカードで使用できます。[展開プロファイル] ウィザードの各要素を構成せずにアプリケーションを展開します。
	[展開の更新]	クラスタ化されたノードのスケーリングまたは展開済みアプリケーションの構成の変更を行う更新プロセスを開始します。
	[展開の分解]	展開済みのアプリケーションをクラウド環境から削除します。

Application Services のアイコンについて

Application Services Web インターフェイスには、機能の可用性を示すアイコンが用意されています。

表 8-2. Application Services アイコン

アイコン	名前	説明
	グループ名	コンポーネントを所有するグループ名の横にある [アプリケーション]、[サービス]、[外部サービス]、[論理テンプレート]、および [展開] の各ページ上に表示されます。
	パブリックに共有	特定のテナント内でパブリックに共有されるコンポーネントの [アプリケーション]、[サービス]、[外部サービス]、[論理テンプレート]、および [展開] の各ページ上に表示されます。
	プライベートに共有	テナント内のビジネス グループ専用コンポーネントの [アプリケーション]、[サービス]、[外部サービス]、[論理テンプレート]、および [展開] の各ページ上に表示されます。
	ポリシー スキャン成功	ポリシー スキャンが成功した展開または展開済みアプリケーションの [展開] ページ上に表示されます。
	ポリシー スキャン失敗	ポリシー スキャンが失敗した展開または展開済みアプリケーションの [展開] ページ上に表示されます。
	自動バインドの利用	ブループリント キャンパス内で、[公開] に設定された別のプロパティに自動バインドできるプロパティの横に表示されます。
	自動バインドの公開	ブループリント キャンパス内で、[利用] に設定された別のプロパティのカスタマイズに使用できるプロパティの横に表示されます。
	追加の構成	[展開プロファイル] ウィザードと、vRealize Automation 展開の [展開サマリ] ページにある [VM 詳細ステータス] ウィンドウに表示されます。 [展開プロファイル] ウィザードでは、追加の構成を使用して、vRealize Automation ブループリントのカスタム プロパティをオーバーライドするか、または既存のプロパティに追加することができます。

Application Services 環境のためのアプリケーション プロビジョニングのセットアップ

9

Application Services は、vRealize Automation の仮想クラウド インフラストラクチャ機能を使用して、アプリケーションを vSphere 環境に展開します。

セットアップ プロセスの一部として、カスタムの vRealize Automation ブループリントを作成する必要があります。vRealize Automation ブループリントには、仮想、クラウド、または物理マシンのプロビジョニング仕様が含まれます。プロビジョニング仕様により、マシンの属性、プロビジョニング方法、およびポリシーと管理設定が決まります。マシンを申請する場合は、マシンの作成元であるブループリントを選択する必要があります。vRealize Automation は、ビルド プロファイルまたはブループリントで定義されたカスタム プロパティのセットを、申請されたマシンに適用します。vRealize Automation インターフェイスからクローンまたはリンク クローンのブループリントにアクセスし、プロパティを指定して vCenter Server テンプレートの複数のプロパティをオーバーライドします。

重要 Application Services は、vRealize Automation 仮想ブループリントから vSphere へのプロビジョニングのみをサポートしています。

Application Services では、クラウド プロバイダとテンプレートを登録する必要があります。クラウド プロバイダを登録したら、ビジネス グループに属する特定の vRealize Automation ブループリントを Application Services 内のクラウド テンプレートにマップします。この登録プロセスにより、マップされた vRealize Automation ブループリントが Application Services ライブラリで使用可能になります。このクラウド テンプレートを使用してアプリケーションを展開するには、クラウド テンプレートを論理テンプレートにマッピングする必要があります。その後、展開環境を作成し、その展開環境を適切な vRealize Automation 予約ポリシーにマップする必要があります。

アプリケーション プロビジョニングのためのクラウド環境の設定と構成に関する主要概念について理解しておく必要があります。[「主要な概念 \(P. 16\)」](#) を参照してください。

この章では次のトピックについて説明します。

- [vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 \(P. 50\)](#)
- [vRealize Automation での仮想マシン テンプレートの作成 \(P. 51\)](#)
- [vRealize Automation での既存の仮想マシン テンプレートの更新 \(P. 58\)](#)
- [vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)](#)
- [vRealize Automation 展開環境の作成 \(P. 60\)](#)

vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件

Application Services で使用するカスタム仮想マシンのテンプレートを作成するには、特定のソフトウェアと仮想化の要件が満たされ、特定の Linux コマンドと Windows コマンドが使用できることを確認します。

表 9-1. カスタム テンプレートのための仮想マシンの要件

要件	説明
オペレーティングシステム	<p>お使いのリリースの現在のバージョンの要件については、https://www.vmware.com/support/pubs/vcac-pubs.html にある vRealize Automation のドキュメントのサポート マトリックスを参照してください。</p> <ul style="list-style-type: none"> ■ Red Hat エンタープライズ Linux ■ Windows Server ■ CentOS ■ SUSE Linux Enterprise ■ Ubuntu
JRE	<p>JRE 1.7.0 をインストールします。</p> <ul style="list-style-type: none"> ■ Windows の場合。Windows 用 JRE は、Application Services 仮想アプライアンスにあるパッケージ <a href="http://<Application_Director_IP>/agent/jre-1.7.0_51-win64.zip">http://<Application_Director_IP>/agent/jre-1.7.0_51-win64.zip からインストールできます。[vCenter Server Windows 仮想マシンのテンプレートを準備する (P. 52)] を参照してください。 ■ Linux の場合。テンプレートの準備中に Linux 用 JRE をインストールするオプションがあります。[vCenter Server Linux 仮想マシンのテンプレートを準備する (P. 54)] を参照してください。
Windows エージェント ブートストラップ サービス	<p>Application Services 仮想アプライアンスから <a href="http://<Application_Director_IP>/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip">http://<Application_Director_IP>/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip ファイルをダウンロードします。</p> <p>Windows ベースの仮想マシン テンプレートにエージェント ブートストラップ サービスをインストールするには、[vCenter Server Windows 仮想マシンのテンプレートを準備する (P. 52)] を参照してください。</p>
vRealize Automation ゲスト エージェント	<p>Windows テンプレートの場合、vRealize Automation Appliance 管理コンソールのインストール ページから次のファイルをダウンロードします。</p> <p><a href="https://<vcac-va-hostname_or_IPaddress>:5480/installer/LinuxGuestAgentPkgs.zip">https://<vcac-va-hostname_or_IPaddress>:5480/installer/LinuxGuestAgentPkgs.zip</p> <p>このファイルには Windows および Linux 用の vRealize Automation ゲスト エージェントが含まれていますが、Windows テンプレートの場合のみ、ファイルをダウンロードする必要があります。Linux テンプレートの場合、ファイルは自動的にダウンロードされます。</p> <p>Windows と Linux のゲスト エージェントをテンプレートにインストールするには、[vCenter Server Windows 仮想マシンのテンプレートを準備する (P. 52)] と [vCenter Server Linux 仮想マシンのテンプレートを準備する (P. 54)] を参照してください。</p>
サポートされている Windows スクリプティング	<p>Application Services は、Windows CMD、PowerShell 2.0、および BeanShell 1.3.0 によるスクリプティングをサポートします。</p>
サポートされている Linux スクリプティング	<p>Application Services は、Bash と BeanShell 1.3.0 によるスクリプティングをサポートします。</p>

表 9-1. カスタム テンプレートのための仮想マシンの要件 (続き)

要件	説明
Linux コマンド	<p>次の Linux コマンドが仮想マシンで使用可能になっている必要があります。</p> <ul style="list-style-type: none"> ■ <code>wget</code> ■ <code>unzip</code> ■ <code>sha256sum</code> ■ <code>grep</code> ■ <code>sed</code> ■ <code>setsid</code> ■ <code>awk</code> ■ <code>ifconfig</code> ■ <code>apt-get</code> ■ <code>yum</code> ■ <code>chkconfig</code> ■ <code>dmidecode</code> ■ <code>perl</code>
オプションのサービス	<p>トラブルシューティングまたはその他の理由により、Linux <code>ssh</code> ログまたは Windows リモート デスクトップを使用して仮想マシンにリモート アクセスする場合、Linux 用の OpenSSH サーバとクライアントまたは Windows 用の Remote Desktop Service (RDS) がインストールされ、適切に動作している必要があります。</p>

重要 プート処理は中断してはならないため、最終的なオペレーティング システムのログイン プロンプトに到達するまで、仮想マシンのブート処理が一時停止することがないように仮想マシンを構成します。たとえば、仮想マシンが起動するときに、ユーザーの操作を求めるプロセスやスクリプトが表示されないようにします。この要件は、Application Services ライブラリのために作成された仮想マシン テンプレートにのみ適用されます。

vRealize Automation での仮想マシン テンプレートの作成

vCenter Server ブループリントを使用して vRealize Automation でクローン展開を開始するには、vCenter Server 仮想マシン テンプレートを作成する必要があります。

クローン展開は、vCenter Server 仮想マシン テンプレートに基づいて、完全な独立型の仮想マシンを作成します。リンク クローン展開は、仮想マシンのスナップショットを参照します。展開は、vCenter Server 仮想マシンのスナップショットに基づいて、完全な独立型の仮想マシンを作成します。クローン展開から、リンク クローン展開を設定できます。「仮想プラットフォームの laas 構成」を参照してください。

■ [vCenter Server Windows 仮想マシンのテンプレートを準備する](#) (P. 52)

vCenter Server ブループリントのための vRealize Automation Windows 仮想マシン テンプレートを準備し、クローン展開を作成する必要があります。Application Services は、vRealize Automation ブループリントを使用して Windows ベースのアプリケーションとサービスを vSphere 環境に展開します。

■ [vCenter Server Linux 仮想マシンのテンプレートを準備する](#) (P. 54)

vRealize Automation ブループリントのための vCenter Server Linux 仮想マシン テンプレートを準備し、クローン展開を作成する必要があります。Application Services は、vRealize Automation ブループリントを使用して、Linux ベースのアプリケーションとサービスを vSphere 環境に展開します。

■ [クローン展開のための vRealize Automation ブループリントの作成](#) (P. 55)

vRealize Automation のカスタム ブループリントを作成する場合は、クローン展開で、Application Services の Windows ベースまたは Linux ベースのアプリケーションとサービスを VMware vSphere 環境にプロビジョニングできます。

- [リンク クローン展開のための vRealize Automation ブループリントの作成と構成 \(P. 56\)](#)

vRealize Automation のカスタム ブループリントを作成する場合は、リンク クローン展開で Application Services の Windows ベースまたは Linux ベースのアプリケーションとサービスを vSphere 環境にプロビジョニングできます。リンク クローンとは、スナップショットに基づいた仮想マシンのコピーです。少ないディスク容量を使用し、すばやくプロビジョニングできるため、パフォーマンスが最優先でない場合に理想的です。

vCenter Server Windows 仮想マシンのテンプレートを準備する

vCenter Server ブループリントのための vRealize Automation Windows 仮想マシン テンプレートを準備し、クローン展開を作成する必要があります。Application Services は、vRealize Automation ブループリントを使用して Windows ベースのアプリケーションとサービスを vSphere 環境に展開します。

開始する前に

- お使いのリリースの正しいバージョンの要件を遵守してください。
<https://www.vmware.com/support/pubs/vcac-pubs.html> の vRealize Automation ドキュメントにあるサポート マトリックスを参照してください。
- vCenter Server 5、5.1、または 5.5 がインストールおよび構成されていることを確認します。
- vSphere Windows 仮想マシンが使用できることを確認します。[「vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 \(P. 50\)」](#) を参照してください。
- vSphere Windows 仮想マシンに vRealize Automation 5.2 または 6.0 の古いバージョンのゲスト エージェントまたは Application Services のエージェント ブートストラップがインストールされていないことを確認します。vSphere のドキュメントを参照してください。
- テンプレートに変換し、オペレーティング システムのカスタマイズ仕様を作成する方法を理解しておく必要があります。vSphere のドキュメントを参照してください。
- 既存の vRealize Automation エージェント サービスと Application Services エージェント ブートストラップ サービスを削除します。[「vRealize Automation での既存の仮想マシン テンプレートの更新 \(P. 58\)」](#) を参照してください。
- ネットワーク構成のすべての生成物がネットワーク構成ファイルから削除されていることを確認します。

手順

- 1 vSphere クライアントにログインし、Windows 仮想マシンを開きます。
- 2 Windows 管理者の認証情報を入力し、コマンド プロンプトを開きます。
- 3 サポートされている Java SE 7 Runtime Environment を
http://<Application_Director_IP>/agent/jre-1.7.0_72-win64.zip からダウンロードしてインストールします。
 - a `\opt\vmware-jre` フォルダを作成し、そのフォルダに JRE ファイルを解凍します。
 - b PowerShell コマンド ウィンドウを開き、「`type \opt\vmware-jre\bin\java -version`」と入力して、インストールを確認します。

インストールされている Java のバージョンが表示されます。
- 4 vRealize Automation ゲスト エージェントをインストールするには、次の手順を実行します。
 - a ディレクトリを `C:\` に変更します。
 - b https://<vcac-va-hostname_or_IPaddress>:5480/installer/ の vRealize Automation Appliance 管理コンソールのインストール ページから、32 ビット プラットフォーム用ファイル `GugentZip.zip` または 64 ビット プラットフォーム用ファイル `GugentZip_64.zip` をダウンロードします。

- c ダウンロードした ZIP ファイルを右クリックし、ポップアップ メニューから [ブロック解除] を選択します。
選択すると、ZIP ファイルの内容がブロック解除されます。
 - d インストール ファイルを **C:** に解凍します。
たとえば、**C:\VRMGuestAgent\WinService.exe**。
- 5 Application Services エージェントのブートストラップ ファイルを
http://<Application_Director_IP>/agent/vmware-appdirector-agent-bootstrap-windows_<VersionNumber>.zip からダウンロードします。
- a ダウンロードした ZIP ファイルを右クリックし、ポップアップ メニューから [ブロック解除] を選択します。
-
- 重要 この Windows セキュリティ機能を無効にしないと、Application Services エージェント ブートストラップ ファイルを使用できません。
-
- b **vmware-appdirector-agent-bootstrap-windows_<version>.zip** ファイルを **\temp** フォルダに解凍します。
 - c **rktools.exe** ユーティリティを実行し、**NTRights.exe** ユーティリティを **\temp** フォルダに追加します。
NTRights.exe ユーティリティは、エージェント ブートストラップ サービスを構成して **darwin** ユーザー アカウントで実行するために、**install.bat** スクリプトで使用されます。このユーティリティは Windows Server 2003 Resource Kit Tools (**rktools.exe**) に含まれ、Microsoft ダウンロード Web サイトからダウンロードできます。
 - d (オプション) **NTRights.exe** ユーティリティが使用できない場合は、エージェント ブートストラップ サービスを手動で構成し、**install.bat** スクリプトを実行した後に特定のユーザー アカウントで実行します。
- 6 vRealize Automation のエージェント ブートストラップ サービスをインストールするには、次の手順を実行します。
- a Windows CMD コンソールを開き、**\temp** フォルダに移動します。
 - b 次のコマンドを入力してエージェント ブートストラップをインストールします。
**install.bat password=<Password> cloudProvider=vcac
vcacServer=<IaaS_Server_FQDN> httpsMode=<true>**
install.bat スクリプトは、エージェント ブートストラップ サービスのために **darwin** というユーザー アカウントを作成し、設定されているパスワードを使用します。**<Password>** は、Windows のパスワード要件を満たす必要があります。
- 7 ユーザー **darwin_user** が存在することを確認します。
- a コマンド プロンプトで **lusrmgr.msc** と入力します。
 - b ユーザー **darwin_user** が存在し、管理者グループに属していることを確認します。
 - c パスワードを無期限に設定します。
この設定により、30 日後もテンプレートが使用できます。
このユーザーが使用できない場合は、Windows サーバのパスワードが正しいかどうかを確認します。
- 8 Application Services と VRM Agent サービスがインストールされていることを確認します。
- 9 Windows 仮想マシンをシャットダウンします。

10 Windows 仮想マシンのカスタマイズ仕様を作成します。

カスタマイズ仕様により、ホスト名、ネットワーク設定、ライセンス設定などの Windows オペレーティングシステムのプロパティを変更できます。ゲスト オペレーティングシステムをカスタマイズすると、同じ設定の仮想マシンが重複するホスト名を使用して展開される場合に発生する問題の防止に役立ちます。

- a [ホーム] - [カスタマイズ仕様マネージャ] を選択します。
- b [新規] をクリックし、カスタマイズ仕様を作成します。
- c [vSphere Client Windows ゲストのカスタマイズ] ウィザードで、カスタマイズ仕様の名前を指定してください。
- d [NetBIOS] セクションで [仮想マシン名を使用] をクリックし、[管理者パスワード] セクションで Windows オペレーティングシステムの管理者パスワードを入力します。
- e デフォルトの設定を受け入れて、[終了] をクリックします。

Windows ベースの vRealize Automation ブループリントは、このカスタマイズ仕様情報を使用して、クローンまたはリンクされた展開を作成します。

11 インベントリで Windows 仮想マシンを右クリックし、[テンプレート] - [テンプレートに変換] を選択します。

vCenter Server は、仮想マシンをテンプレートとしてマークし、[最近のタスク] ペインにタスクを表示します。

次に進む前に

クローンまたはリンクされたクローン展開の vRealize Automation ブループリントを作成します。[「クローン展開のための vRealize Automation ブループリントの作成 \(P. 55\)」](#) または [「リンク クローン展開のための vRealize Automation ブループリントの作成と構成 \(P. 56\)」](#) を参照してください。

vCenter Server Linux 仮想マシンのテンプレートを準備する

vRealize Automation ブループリントのための vCenter Server Linux 仮想マシン テンプレートを準備し、クローン展開を作成する必要があります。Application Services は、vRealize Automation ブループリントを使用して、Linux ベースのアプリケーションとサービスを vSphere 環境に展開します。

開始する前に

- vCenter Server がインストールされて構成されていることを確認します。お使いのリリースの現在のバージョンの要件については、<https://www.vmware.com/support/pubs/vcac-pubs.html> にある vRealize Automation のドキュメントのサポート マトリックスを参照してください。
- サポート対象の vSphere Linux 仮想マシンが使用可能であることを確認します。[「vRealize Automation カスタムテンプレートを作成するための仮想マシンの要件 \(P. 50\)」](#) を参照してください。
- テンプレートに変換し、オペレーティングシステムのカスタマイズ仕様を作成する方法を理解しておく必要があります。vSphere のドキュメントを参照してください。
- vCenter Server 仮想マシンが Application Services 仮想マシンの要件を満たしていることを確認します。[「vRealize Automation カスタムテンプレートを作成するための仮想マシンの要件 \(P. 50\)」](#) を参照してください。
- ネットワーク構成のすべての生成物がネットワーク構成ファイルから削除されていることを確認します。

手順

- 1 vSphere クライアントにログインして、Linux 仮想マシンを開きます。
- 2 root 権限を使用して仮想マシンにログインしてターミナルを開きます。
- 3 インストーラ スクリプト `http://<ApplicationServicesServerIP>/tools/preparevCACTemplate.sh` を仮想マシン テンプレートにダウンロードします。

ファイルのダウンロードには、`wget` または `scp` コマンドを使用できます。

- 4 インストーラ スクリプトに実行権限があることを確認します。

```
chmod +x ./preparevCACTemplate.sh
```

- 5 インストーラ スクリプトを実行します。

```
./preparevCACTemplate.sh
```

- 6 すべてのプロンプトに答えます。

インストーラ スクリプトに成功すると、成功のメッセージが表示されます。失敗した場合、エラー メッセージが表示されて、コンソールにログオンされます。

- 7 エラーが発生した場合は、解決して [手順 4](#) から [手順 6](#) を繰り返します。

- 8 Linux 仮想マシンをシャットダウンします。

- 9 インベントリで Linux 仮想マシンを右クリックし、[テンプレート] - [テンプレートに変換] を選択します。

vCenter Server によって、選択した仮想マシンがテンプレートとしてマークされ、[最近のタスク] ペインにタスクが表示されます。

次に進む前に

[「クローン展開のための vRealize Automation ブループリントの作成 \(P. 55\)」](#) または [「リンク クローン展開のための vRealize Automation ブループリントの作成と構成 \(P. 56\)」](#)。

クローン展開のための vRealize Automation ブループリントの作成

vRealize Automation のカスタム ブループリントを作成する場合は、クローン展開で、Application Services の Windows ベースまたは Linux ベースのアプリケーションとサービスを VMware vSphere 環境にプロビジョニングできます。

開始する前に

- vRealize Automation 6.1 以降がインストールおよび構成されていることを確認します。
- Application Services で使用する vRealize Automation のビジネス グループが作成されていることを確認します。
- クローン展開を作成する vRealize Automation ブループリントで、vCenter Server 仮想マシンが使用できることを確認します。

Windows ベースの vRealize Automation ブループリントで使用できるカスタマイズ仕様の名前が必要です。
[「vCenter Server Windows 仮想マシンのテンプレートを準備する \(P. 52\)」](#) を参照してください。

- vRealize Automation でブループリントを作成する方法を理解しておく必要があります。「仮想プラットフォームの laas 構成」を参照してください。

手順

- 1 **テナント管理者**として vRealize Automation コンソールにログインします。
- 2 [新規ブループリント] - [仮想] - [vSphere (vCenter)] を選択します。
- 3 [ブループリント情報] タブに情報を入力します。
- 4 [ビルド情報] タブで、[アクション] ドロップダウン メニューから [クローン作成] オプションを選択します。
- 5 [クローン作成元] をクリックして参照し、vRealize Automation で使用するために作成した vCenter Server テンプレートを選択します。
- 6 Windows ベースの vRealize Automation ブループリントの場合は、[カスタマイズ仕様] テキスト ボックスに、カスタマイズ仕様の名前を入力します。

カスタマイズ仕様は、vCenter Server 仮想マシンのテンプレートの準備の一環として作成済みです。

- 7 必要に応じて [プロパティ] タブと [セキュリティ] タブに情報を入力し、[OK] をクリックして変更を保存します。

vRealize Automation ブループリントが [ブループリント] メニューに表示されます。

次に進む前に

Application Services で使用するクローン展開用に構成した vRealize Automation ブループリントを登録します。
[\[vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)\]](#) を参照してください。

リンク クローン展開のための vRealize Automation ブループリントの作成と構成

vRealize Automation のカスタム ブループリントを作成する場合は、リンク クローン展開で Application Services の Windows ベースまたは Linux ベースのアプリケーションとサービスを vSphere 環境にプロビジョニングできます。リンク クローンとは、スナップショットに基づいた仮想マシンのコピーです。少ないディスク容量を使用して、すばやくプロビジョニングできるため、パフォーマンスが最優先でない場合に理想的です。

開始する前に

- vRealize Automation 6.1 以降がインストールおよび構成されていることを確認します。
- Application Services で使用する vRealize Automation のビジネス グループが作成されていることを確認します。
- リンク クローン展開を作成する vCenter Server ブループリントで、vRealize Automation 仮想マシンが使用できることを確認します。

Windows ベースの vRealize Automation ブループリントで使用するカスタマイズ仕様の名前が必要です。
[\[vCenter Server Windows 仮想マシンのテンプレートを準備する \(P. 52\)\]](#) を参照してください。

- vRealize Automation でブループリントを作成する方法を理解しておく必要があります。「仮想プラットフォームの laas 構成」を参照してください。

手順

- 1 テナント管理者として vRealize Automation コンソールにログインします。
- 2 [新規ブループリント] - [仮想] - [vSphere (vCenter)] を選択して、ブループリントに名前を付けます。たとえば、ブループリントに **vCAC_Clone** という名前を付けます。
- 3 [ブループリント情報] タブに情報を入力します。
- 4 [ビルド情報] タブで、[アクション] ドロップダウン メニューから [クローン作成] オプションを選択します。
- 5 [クローン作成元] ドロップダウン メニューから、vCenter Server で使用するために作成した vRealize Automation テンプレートを選択します。

- 6 Windows ベースの vRealize Automation ブループリントの場合は、[カスタマイズ仕様] テキスト ボックスに、カスタマイズ仕様の名前を入力します。

カスタマイズ仕様は、前提条件のセクションで説明したように vCenter Server 仮想マシンのテンプレート準備の一環として作成済みです。

- 7 [OK] をクリックして、変更内容を保存します。
- 8 ブループリントを選択し、ドロップダウン メニューからマシンを要求して、デフォルト設定を受け入れます。たとえば、**vCAC_Clone** という名前のブループリントを選択します。

申請情報は、vCenter Server テンプレートで定義されたリソースから取得されます。

マシン申請が処理されると、[インフラストラクチャ] - [マシン] - [管理対象マシン] メニューに表示されます。マシン名 (vCAC_Clone など) には、一連の固有の文字と数字が付加されます (vCAC_Clone-VFF17Y0 など)。

- 9 マシン (vCAC_Clone-VFF17Y0 など) にログインし、オペレーティングシステムが Windows かまたは Linux かに応じて、Application Services および vRealize Automation の各サービスを停止します。

サービス	説明
vmware-appdirector-agent-service-vcac	Linux の Application Services ブートストラップ サービス
vrn-agent	Linux の vRealize Automation エージェント サービス

サービス	説明
AppDAgentBootstrap	Windows の Application Services ブートストラップ サービス
VRMAgent または winservice	Windows の vRealize Automation 6.1 以降のエージェント サービス

- 10 Windows スクリプト **agent_reset.bat** または Linux スクリプト **agent_reset.sh** を実行して、ランタイム ログ ファイルをマシン (vCAC_Clone-VFF17Y0 など) から削除します。
- 11 マシンをパワーオフします。たとえば、vCAC_Clone-VFF17Y0 をパワーオフします。

次に進む前に

リンク クローン展開用にブループリントからスナップショットを作成します。[「vRealize Automation ブループリントからのスナップショットの作成 \(P. 57\)」](#) を参照してください。

vRealize Automation ブループリントからのスナップショットの作成

リンク クローンを作成する場合は、vRealize Automation 仮想マシンのスナップショットを作成する必要があります。このスナップショットは最小限のディスク領域しか使用しないため、リンク クローン展開はクローンの展開より迅速に行われます。

マシン スナップショットの作成の詳細については、[「vRealize Automation ブループリントからのスナップショットの作成 \(P. 57\)」](#) を参照してください。vSphere Client を使用した仮想マシンテンプレートのスナップショット取得の関連情報については、VMware vSphere ドキュメントを参照してください。

開始する前に

- vRealize Automation 6.1 以降がインストールおよび構成されていることを確認します。
- Application Services で使用する vRealize Automation のビジネス グループが作成されていることを確認します。
- スナップショットの作成元となる vRealize Automation ブループリントが使用可能になっていることを確認します。[「リンク クローン展開のための vRealize Automation ブループリントの作成と構成 \(P. 56\)」](#) を参照してください。

手順

- 1 [インフラストラクチャ] - [ブループリント] - [ブループリント] を選択します。
- 2 [新規ブループリント] - [仮想] - [vSphere (vCenter)] をクリックして、ブループリントに名前を付けます。たとえば、ブループリントに **vCAC_LinkedClone** という名前を付けます。
- 3 [ブループリント情報] タブに情報を入力します。
- 4 [ビルド情報] タブで、[アクション] ドロップダウン メニューから [リンク クローン] を選択します。
- 5 vCAC_Clone-VFF17Y0 マシンのスナップショットを選択します。
 - a [クローン作成元] テキスト ボックスの横にある [参照] アイコンをクリックします。
[スナップショットの選択] ダイアログ ボックスが表示されます。
 - b マシン [vCAC_Clone-VFF17Y0] を選択して、[次へ] をクリックします。
 - c スナップショット **vCAC_Clone-VFF17Y0_snapshot** を選択して、[次へ] をクリックします。
 - d [OK] をクリックして、変更内容を保存します。
[クローン作成元] テキスト ボックスに vCAC_Clone-VFF17Y0_snapshot が表示されます。
- 6 Windows ベースの vRealize Automation ブループリントの場合は、[カスタマイズ仕様] テキスト ボックスに、カスタマイズ仕様の名前を入力します。

カスタマイズ仕様は、vCenter Server 仮想マシンのテンプレートの準備の一環として作成済みです。

- 7 必要に応じて [プロパティ] および [アクション] タブを構成し、[OK] をクリックして変更内容を保存します。
新しいブループリントは、たとえば **vCAC_LinkedClone** のように [ブループリント] ページに表示されます。

次に進む前に

Application Services で使用するリンク クローン展開用に構成された vRealize Automation ブループリントを登録します。[\[vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)\]](#) を参照してください。

vRealize Automation での既存の仮想マシン テンプレートの更新

既存の Linux または Windows のテンプレートのコンテンツを更新するか、またはそのコンテンツを使用してテンプレートを新規作成する前に、適切なコマンドを実行してエージェント ブートストラップ サービスを削除する必要があります。

Linux テンプレートの場合、**agent_reset.sh** コマンドによって Application Services エージェント ブートストラップのステータスがリセットされ、既存のランタイム ログ ファイルが削除されます。仮想マシンに root としてログインし、このコマンドを実行できます。

```
/opt/vmware-appdirector/agent-bootstrap/agent_reset.sh
```

Windows テンプレートの場合、**agent_reset.bat** コマンドによって既存のランタイム ログ ファイルが削除されます。PowerShell コマンド ウィンドウで、次のコマンドを入力します。

```
\opt\vmware-appdirector\agent-bootstrap\agent_reset.bat
```

Windows テンプレートの場合は、既存の Application Services エージェント ブートストラップと vRealize Automation 6.0 以上のゲスト エージェントも削除できます。PowerShell コマンド ウィンドウで、次のコマンドを入力します。

```
\opt\vmware-appdirector\agent-bootstrap\agent_bootstrap_removal.bat
```

vRealize Automation クラウド プロバイダとテンプレートの登録

Application Services は、vRealize Automation ビジネス グループに接続して、クラウド プロバイダを登録します。


vRealize Automation のビジネス グループは、ビジネス、部署、または他の組織単位に相当することが多いサービスやリソースのセットにユーザーを関連付けます。各ビジネス グループは、マシンの申請に使用する vRealize Automation の 1 つ以上のブループリントにアクセスできます。これらのブループリントは、1 つ以上のビジネス グループに所属できます。マシンを申請するには、ユーザーは少なくとも 1 つのビジネス グループに所属している必要があります。vRealize Automation ブループリントのマシン リソース情報は手動で追加するか、またはクローンまたはリンク クローン vSphere テンプレートから取得します。

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- vRealize Automation 6.1 以降がインストールおよび構成されていることを確認します。
- Application Services で使用する vRealize Automation のビジネス グループが作成および構成されていることを確認します。
- CLI を使用して、Application Services を vRealize Automation に登録していることを確認します。 [\[vRealize Automation への Application Services の登録 \(P. 35\)\]](#) を参照してください。
- vRealize Automation の少なくとも 1 つのブループリントが、接続先の Application Services クラウド プロバイダのビジネス グループで使用できることを確認します。このブループリントは、Application Services の要件を満たす仮想マシンのテンプレートをポイントする必要があります。 [\[vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 \(P. 50\)\]](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [クラウド プロバイダ] を選択します。

- 2 [追加] アイコン () をクリックします。
- 3 クラウド プロバイダ情報を作成します。

vRealize Automation 6.0 以降では、Application Services を vRealize Automation に登録する際に、ログイン ユーザーのホスト名とビジネス グループ名があらかじめ入力されています。

オプション	説明
名前と説明	いずれかのフィールドに、vRealize Automation ビジネス グループの名前を含めます。これらのフィールドのテキストは、[Deployment Profile] ウィザードの [クラウド プロバイダ] 列に表示されます。
クラウド プロバイダのタイプ	vCAC、vCloud、および EC2 が、サポートされるタイプです。
vCAC インフラストラクチャ IP/ホスト名/URL	登録済み vRealize Automation サーバの IP アドレスまたはホスト URL が表示されます。このフィールドは読み取り専用です。 安全な vCAC IP の例には、https://192.0.2.115 や https://192.10.1.113:443 などがあります。 安全なホスト名 URL の例は、https://vcac.it.vmware.com です。
ユーザー名とパスワード	vRealize Automation で、選択済みビジネス グループのビジネス グループ マネージャの認証情報を追加します。 ユーザー名を入力する際はドメイン名が必要です。ユーザー名の例は、myname@mycompany.com です。
ビジネス グループ	[ビジネス グループ] ドロップダウン メニューからビジネス グループを選択します。メニューに、ログイン ユーザーのビジネス グループが表示されます。
共有	クラウド プロバイダのステータスとして、[プライベート] と表示されます。この値は常に [プライベート] で、読み取り専用です。

- 4 入力した値が正しいかどうかをテストするには、[接続の検証] をクリックします。
- 5 [保存] をクリックして、クラウド プロバイダを登録します。
クラウド プロバイダの必須エントリが無効または空白になっている場合は、修正を求めるプロンプトが表示されます。
- 6 vRealize Automation ブループリントを登録するには、[編集] をクリックします。
- 7 [テンプレート] セクションで、  をクリックします。
- 8 リストから 1 つ以上のクラウド テンプレートまたは vRealize Automation ブループリントの横にあるチェック ボックスをオンにし、[OK] をクリックします。
グローバル ブループリントおよびクラウド プロバイダの選択済みビジネス グループに所属するブループリントのみが表示されます。
登録したテンプレートは、論理テンプレートをクラウド テンプレートにマッピングする際に選択できるクラウド テンプレートのリストに追加されます。
- 9 vRealize Automation クラウド プロバイダの既存の情報を変更するには、[編集] をクリックします。変更を加えてから、ツールバーの [保存] をクリックします。
- 10 [保存] をクリックします。
ベスト プラクティスとして、vRealize Automation クラウド プロバイダを編集する際に、クラウド テンプレート情報、vRealize Automation インフラストラクチャ IP、およびホスト名を更新します。

注意 vRealize Automation 環境で vRealize Automation ブループリント名を変更しても、Application Services ユーザー インターフェイスに新しいブループリント名は表示されません。新しいブループリント名は、クラウド テンプレートのメタ データでのみ更新されます。

クラウド プロバイダが登録され、そのテンプレートが Application Services で使用できるようになります。

次に進む前に

Application Services 論理テンプレートに、vRealize Automation クラウド テンプレートをマップします。[「ライブラリへの論理テンプレートの追加 \(P. 125\)」](#) を参照してください。

展開環境を作成し、アプリケーションを vRealize Automation 環境に展開します。[「vRealize Automation 展開環境の作成 \(P. 60\)」](#) を参照してください。

vRealize Automation 展開環境の作成

アプリケーションをクラウド環境に展開する前に、展開環境を vRealize Automation 予約ポリシーにマップできます。予約ポリシーを選択していない場合には、vRealize Automation は、展開時に適切な予約ポリシーを採用します。

vRealize Automation ビジネス グループは、予約ポリシーにマップされた予約をいくつか保有できます。予約には、マシンを構築するためのメモリ、ストレージ、ネットワークなどのリソース プールがあります。予約の設定に基づいて、指定されたストレージ領域に仮想マシンを保存し、接続できるネットワークを決定できます。予約ポリシーを使用すると、定義されたサービス レベルを作成するか、または特定の種類のリソースを特定の目的で使用できるようにするために、類似するリソースをグループ化できます。

vRealize Automation ブループリントへの予約を含む予約ポリシーを追加できます。このブループリントからマシンを申請すると、仮想マシンは指定されたストレージ領域に保存され、予約ですでに定義されているネットワークに割り当てられます。予約ポリシーをブループリントにマップしない場合、vRealize Automation は vRealize Automation ブループリントの仮想マシンのリソース要件に基づいて予約を割り当てます。

開始する前に

- **アプリケーション クラウド管理者**として Application Services にログインします。
- vRealize Automation 6.1 以降がインストールおよび構成されていることを確認します。
- Application Services で使用する vRealize Automation のビジネス グループが作成および構成されていることを確認します。
- ビジネス グループにアクセスするためのテナント管理者権限を保有していることを確認します。
テナント管理者権限を保有するユーザーが、ビジネス グループの作成と保守を行います。ビジネス グループの詳細については、『仮想プラットフォームの IaaS 構成』を参照してください。
- ビジネス グループのビジネス グループ マネージャであることを確認します。
- vRealize Automation で予約ポリシーが作成され、構成されていることを確認します。Application Services 展開環境は、vRealize Automation の予約ポリシーと同じです。
- 1 台の vRealize Automation クラウド プロバイダが Application Services に登録されていることを確認します。[「vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [展開環境] を選択します。
- 2 [新規] をクリックします。

- 3 展開環境の情報を入力します。

オプション	説明
展開環境の名前と説明	これらのいずれかのテキスト ボックスに予約ポリシーの名前を含めます。これらのテキスト ボックスのテキストは、[展開プロファイル] ウィザードの [展開環境] 列に表示されます。
クラウド プロバイダ	クラウド プロバイダがリストに表示されない場合、ダイアログ ボックスをキャンセルし、[クラウド] - [クラウド プロバイダ] を選択して、vRealize Automation クラウド プロバイダを追加します。
予約ポリシー	<p>[選択] をクリックし、選択したクラウド プロバイダが提供する予約ポリシーのリストから選択します。</p> <p>名前に auto を含む vRealize Automation の予約ポリシーは、予約ポリシー リストに表示されません。予約ポリシー リストに表示されるように予約ポリシーの名前を変更してください。</p> <p>[予約ポリシーなし] を選択すると、vRealize Automation はリソースの可用性に基づいて予約を割り当てます。</p>

- 4 予約ポリシーを選択し、[OK] をクリックします。

vRealize Automation の予約ポリシーは、Application Services 展開環境の名前にマップされます。

- 5 [保存] をクリックします。

- 6 (オプション) vRealize Automation 展開環境の既存の情報を変更するには、ツールバーの [編集] をクリックして変更を行います。

一部の展開では、プロキシ設定のプロパティを編集することもできます。これらのプロキシ設定を変更すると、外部データにアクセスできます。[表示] をクリックして、次のプロパティを編集します。

- http.proxyHost
- http.proxyPort
- http.proxyUser
- http.proxyPassword

プロキシ設定のプロパティを編集しなければ、グローバルのプロキシ設定が検討されます。

展開プロファイルを作成する際に、この展開環境を選択して vSphere にアプリケーションを展開できます。

次に進む前に

外部サービス インスタンスのマッピング、ポリシー インスタンスの作成、または展開環境へのソリューション インスタンスの登録を実行できます。[「外部サービス インスタンスのマッピング \(P. 61\)」](#)、[「ポリシー インスタンスの作成 \(P. 62\)」](#) または [「Puppet ソリューション インスタンスの作成 \(P. 65\)」](#) を参照してください。

外部サービス インスタンスのマッピング

アプリケーションが展開環境に展開されるときに外部サービス インスタンスが必ず使用されるように、外部サービス インスタンスを展開環境にマッピングできます。

開始する前に

- **アプリケーション クラウド管理者**として Application Services にログインします。
- クラウド プロバイダによっては、少なくとも 1 つの vCloud Director、vRealize Automation、または Amazon EC2 展開環境が Application Services でマッピングされている必要があります。[「vCloud Director 展開環境の作成 \(P. 81\)」](#)、[「vCloud Director 展開環境の作成 \(P. 81\)」](#)、[「vRealize Automation 展開環境の作成 \(P. 60\)」](#)、または [「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。
- カスタム外部サービスを使用する場合、事前構成済みまたはカスタム外部サービスがライブラリで使用可能であることを確認します。[「ライブラリへの外部サービスの追加 \(P. 117\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [展開環境] を選択します。
- 2 既存の展開環境を選択します。
- 3 [外部サービス インスタンス] タブを選択し、[登録] をクリックして、外部サービス インスタンスを作成します。
- 4 展開環境に関する外部サービス インスタンス情報を入力します。

オプション	説明
名前と説明	フィールドのいずれかに外部サービス名を入力します。[展開プロファイル] ウィザードの [展開環境] 列に、[名前] フィールドのテキストが表示されます。
外部サービス バージョン	外部サービスがドロップダウン リストに表示されない場合は、ダイアログ ボックスをキャンセルして、[ライブラリ] - [外部サービス] を選択し、カスタム外部サービスをライブラリに追加します。
サービス プロバイダ	下矢印をクリックして、外部サービスに関連付けられたサービス プロバイダのリストから選択します。 注意 このオプションは、カスタムの外部サービスを展開環境にマッピングすると表示されます。

外部サービス バージョンまたはプロバイダ仕様バージョンで定義されたプロパティが、[プロパティ] セクションに自動的に入力されます。

- 5 [保存] をクリックします。
- 6 外部サービス インスタンス名の横にある矢印をクリックし、前のページに戻ります。

ポリシー インスタンスの作成

ポリシー インスタンスを使用すると、特定の展開環境におけるすべてのアプリケーション プロビジョニング アクティビティに対して、既存のポリシー定義を有効にすることができます。

注意 複合展開機能では、ポリシーのコンプライアンス スキャンはサポートされません。

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- クラウド プロバイダによっては、少なくとも 1 つの vCloud Director、vRealize Automation、または Amazon EC2 展開環境が Application Services でマッピングされている必要があります。[「vCloud Director 展開環境の作成 \(P. 81\)」](#)、[「vCloud Director 展開環境の作成 \(P. 81\)」](#)、[「vRealize Automation 展開環境の作成 \(P. 60\)」](#)、または [「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。
- カスタム ポリシーを使用している場合は、カスタム ポリシーがライブラリ内で使用可能であることを確認します。[「ライブラリへのポリシーの追加 \(P. 128\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [展開環境] を選択します。
- 2 既存の展開環境を選択します。
- 3 [ポリシー インスタンス] タブを選択し、[ポリシー インスタンスの作成] をクリックしてポリシー インスタンスを作成します。

- 4 展開環境に関するポリシー インスタンス情報を入力します。

オプション	説明
名前と説明	フィールドのいずれかにポリシー名を入力します。これらのフィールドのテキストは、展開環境のポリシー インスタンス セクションに表示されます。
ポリシー	ポリシーがドロップダウン リストに表示されない場合は、ダイアログ ボックスをキャンセルして [ライブラリ] - [ポリシー] を選択し、カスタム ポリシーをライブラリに追加します。
バージョン	下矢印をクリックして、ポリシーに関連付けられたバージョンのリストから選択します。
重大	展開の現在のコンプライアンス状態に関係なく、展開が非準拠状態になる可能性がある操作を停止するには、このチェック ボックスを選択します。 このオプションを使用する前に、展開環境でポリシー コンプライアンスを完全に確立しておくことをお勧めします。重要度の低いポリシー インスタンスでは、展開のコンプライアンス ステータスは通知されますが、操作は停止されません。

注意 ポリシー インスタンスで新しいプロパティ値を指定すると、ポリシーで定義された既存のライブラリ値が新しいプロパティ値によってオーバーライドされます。

ポリシー バージョンで定義されたプロパティとスクリプトが、[プロパティ] セクションと [アクション] セクションに自動的に入力されます。

ポリシー アクション スクリプトを作成すると、ポリシー インスタンスのコピーが作成されます。ライブラリ内のポリシー スクリプトを変更しても、そのポリシーに関連付けられた既存のポリシー インスタンスには影響しません。スクリプトの変更を有効にするには、ポリシー インスタンスを作成する必要があります。

- 5 [保存] をクリックします。
- 6 ポリシー インスタンス名の横にある矢印をクリックし、前のページに戻ります。

Puppet ソリューション インスタンスの登録

Puppet マスターなどのソリューション インスタンスを登録するには、Application Services サーバと連携するように Puppet マスターを構成し、Application Services の展開環境内に Puppet マスターのソリューション インスタンスを作成する必要があります。

登録プロセスについては、手動オプションまたは自動オプションを実装できます。自動オプションの説明については、[「Puppet マスター ソリューション インスタンスの登録 \(P. 68\)」](#) を参照してください。

手順

- 1 [登録のために Puppet マスター環境を準備する \(P. 63\)](#)
展開環境内に Puppet マスターを登録するには、Application Services サーバで動作するように Puppet マスターを準備する必要があります。
- 2 [Puppet ソリューション インスタンスの作成 \(P. 65\)](#)
Puppet マスターなどのソリューション インスタンスを Application Services の展開環境で作成して、Puppet ベース サービスを使用するアプリケーションがその展開環境で展開されたときに、Puppet マスターが必ず使用されるようにする必要があります。

登録のために Puppet マスター環境を準備する

展開環境内に Puppet マスターを登録するには、Application Services サーバで動作するように Puppet マスターを準備する必要があります。

登録プロセスを自動化することもできます。[「Puppet マスター ソリューション インスタンスの登録 \(P. 68\)」](#) を参照してください。

開始する前に

- Puppet Enterprise バージョン 3.0.1 から 3.2.3、または Puppet Open Source バージョン 3.2.4 から 3.6.2 にインストールされていることを確認します。
- Puppet エージェントの最新のサポート対象バージョンがインストールされていることを確認します。エージェントをインストールする手順については、Puppet Labs のドキュメントが掲載された Web サイトを参照してください。
- Puppet マスター以外に Puppet がインストールされているクライアント マシンがあることを確認します。
- Application Services と Puppet マスター サーバのシステム時間が同期されていることを確認します。時間の同期には、NTP サーバを使用できます。2 台のサーバ間の時間差は 60 秒を超過することはできません。

手順

- 1 Puppet マスター上のディレクトリに移動して、`appd_nodes` というディレクトリを作成します。
 - Puppet Enterprise で、`/etc/puppetlabs/puppet/manifests/` に移動します。
 - Puppet Open Source で、`/etc/puppet/manifests/` に移動します。
- 2 `/etc/puppetlabs/puppet/manifests/site.pp` または `/etc/puppet/manifests/site.pp` ファイルを開いて、コマンド `import 'appd_nodes/*.pp'` をファイルの最初の行に追加します。
- 3 コマンドラインで、`empty_manifest_file.pp` ファイルを作成します。
 - Puppet Enterprise で
「`touch /etc/puppetlabs/puppet/manifests/appd_nodes/empty_manifest_file.pp`」と入力します。
 - Puppet Open Source で
「`touch /etc/puppet/manifests/appd_nodes/empty_manifest_file.pp`」と入力します。
- 4 `mcollective` サーバ構成ファイルを開いて、`plugin.ssl_serializer = yaml` という行がない場合には追加します。
 - Puppet Enterprise で、ファイル `/etc/puppetlabs/mcollective/server.cfg` を開きます。
 - Puppet Open Source で、ファイル `/etc/mcollective/server.cfg` を開きます。
- 5 `appdinteg.rb` エージェント ファイルと `appdinteg.ddl` エージェント ファイルをダウンロードします。
`http://<Application_Services_IP>/artifacts/solutions/puppet/mcollective/agent/appdinteg.ddl`
`http://<Application_Services_IP>/artifacts/solutions/puppet/mcollective/agent/appdinteg.rb`
- 6 Puppet マスターの `<LIBDIR>/mcollective/agent` ディレクトリに `appdinteg.rb` エージェント ファイルと `appdinteg.ddl` エージェント ファイルをコピーします。
`<LIBDIR>` は MCollective `libdir` 設定の値で、通常は `/opt/puppet/libexec/mcollective/` に設定されます。
- 7 MCollective を再起動し、新たにコピーされたエージェント ファイルを表示します。
 - Puppet Enterprise で「`/etc/init.d/pe-mcollective restart`」と入力します。
 - Puppet Open Source で「`/etc/init.d/mcollective restart`」と入力します。
- 8 Puppet がインストールされているクライアント マシンで証明書を生成します。
`puppet certificate generate <Name> --ssldir <TempCredsDir> --ca-location remote --ca_server <CAPuppetMaster>`
`<Name>` は、Puppet マスターに対応する Application Services 証明書の文字列の名前です。`<TempCredsDir>` は、`/tmp` フォルダにローカルに作成されたディレクトリで (`/tmp/appd_mc/credentials` など)、`<CAPuppetMaster>` は、Puppet マスターのホスト名または IP アドレスです。

- 9 (オプション) Puppet マスターで、証明書への自動署名が有効になっていない場合は、**sudo puppet cert sign <Name>** コマンドを入力します。
- 10 署名されたパブリック証明書ファイルを MCollective の承認済みクライアント リストにコピーします。
cp <CertDirectory>/<Name>.pem <AuthorizedClientsDirectory>
 <CertDirectory>は Puppet マスターの certdir 構成設定で、<AuthorizedClientsDirectory> は MCollective サーバの plugin.ssl_client_cert_dir 構成設定です。
 Puppet Enterprise のサンプル コピー コマンド: **cp /etc/puppetlabs/puppet/ssl/ca/signed/vmware-appd.pem /etc/puppetlabs/mcollective/ssl/clients/**
- 11 Puppet がインストールされているクライアント マシンで、Puppet マスターから署名証明書を取得します。
puppet certificate find <Name> --ssldir <TempCredsDir> --ca-location remote --ca_server <CAPuppetMaster>
puppet certificate find ca --ssldir <TempCredsDir> --ca-location remote --ca_server <CAPuppetMaster>
- 12 Puppet がインストールされているクライアント マシンで、Puppet マスターからパブリック証明書を取得します。
 - Puppet Enterprise で
「puppet certificate find pe-internal-mcollective-servers --ssldir <TempCredsDir> --ca-location remote --ca_server <CAPuppetMaster>」と入力します。
 - Puppet Open Source で
「puppet certificate find mcollective-servers --ssldir <TempCredsDir> --ca-location remote --ca_server <CAPuppetMaster>」と入力します。

Application Services により、`/etc/puppetlabs/puppet/manifests/appd_nodes` または `/etc/puppet/manifests/appd_nodes` ディレクトリにプロビジョニングする各仮想マシンのノード定義ファイル `*.pp` が作成されます。

次に進む前に

Application Services の展開環境で Puppet マスターを登録します。[「Puppet ソリューション インスタンスの作成 \(P. 65\)」](#) を参照してください。

Puppet ソリューション インスタンスの作成

Puppet マスターなどのソリューション インスタンスを Application Services の展開環境で作成して、Puppet ベース サービスを使用するアプリケーションがその展開環境で展開されたときに、Puppet マスターが必ず使用されるようにする必要があります。

展開環境ごとに1つの Puppet マスター ソリューション インスタンスのみ作成可能です。複数の展開環境のソリューション インスタンスが、同一の Puppet マスターを参照できます。Puppet マスター サーバでローカルに実行する Ruby スクリプトの使用については、[「Puppet マスター ソリューション インスタンスの登録 \(P. 68\)」](#) を参照してください。

開始する前に

- **アプリケーション クラウド管理者**として Application Services にログインします。
- クラウド プロバイダによっては、少なくとも1つの vCloud Director、vRealize Automation、または Amazon EC2 展開環境が Application Services でマッピングされている必要があります。[「vCloud Director 展開環境の作成 \(P. 81\)」](#)、[「vCloud Director 展開環境の作成 \(P. 81\)」](#)、[「vRealize Automation 展開環境の作成 \(P. 60\)」](#)、または [「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。
- Puppet マスターが Application Services サーバと連携するように構成されていることを確認します。[「登録のために Puppet マスター環境を準備する \(P. 63\)」](#) または [「Puppet マスター ソリューション インスタンスの登録 \(P. 68\)」](#) を参照してください。

- カスタム ポリシーを使用している場合は、カスタム ポリシーがライブラリ内で使用可能であることを確認します。
[ライブラリへのポリシーの追加 (P. 128)] を参照してください。

手順

- 1 Application Services の Web インターフェイスにログインします。
- 2 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [展開環境] を選択します。
- 3 既存の展開環境を選択します。
- 4 [ソリューション インスタンス] タブを選択して [ソリューション インスタンスの作成] をクリックします。
- 5 展開環境のソリューション インスタンス情報を入力します。

オプション	説明
名前と説明	ソリューション名を含めます。 ソリューションのプロパティまたはエージェント構成情報を説明セクションで追加できます。
ソリューション	ドロップダウン リストから使用可能な Puppet マスター v1.0.0 ソリューションを選択します。

ソリューション プロパティおよび対応するエージェントが入力されます。

- 6 [プロパティ] セクションに、必要なプロパティ値を入力します。

プロパティ	説明
puppet_server	Puppet マスター サーバの修飾ドメイン名。 エージェントで Puppet マスター証明書を信頼するために、Puppet マスターをインストールしたときに設定した有効な DNS 名のいずれかを使用します。Application Services サーバに puppet_server_ip プロパティも指定されていない場合は、この DNS 名の Puppet マスターに接続できるようにする必要があります。
puppet_server_ip	Puppet マスター サーバの IP アドレス。 Puppet マスター サーバのホスト名を Application Services サーバまたはエージェント ノードで解決できない場合は、IP アドレスを指定する必要があります。
puppet_environment	Puppet エージェントに割り当てる Puppet マスター環境の名前。この名前は、Puppet エージェントの環境構成プロパティに対応します。
agent_run_interval	Puppet エージェントがライブラリを適用する頻度。デフォルトの実行間隔は 30 分 (30m) です。次の単位で実行間隔を指定できます。 <ul style="list-style-type: none"> ■ 秒。たとえば、30 または 30s と指定します。 ■ 分。たとえば、30m と指定します。 ■ 時間。たとえば、6h と指定します。 ■ 日。たとえば、5d と指定します。 ■ 年。たとえば、2y と指定します。 実行間隔が 0 の場合、Puppet エージェントは連続的に実行されます。
identity_private_key	証明書のプライベート キーが Puppet マスターによって Application Services サーバに発行されます。 Application Services は、Puppet マスターおよび MCollective の認証にこの証明書を使用します。証明書の形式は、ヘッダとトレイラを含む PEM エンコードです。ヘッダは、たとえば「-----BEGIN RSA PRIVATE KEY--」のような形式になります。 このプロパティの値として、Puppet マスターを準備するために作成したファイル <TempCredsDir>/private_keys/<Name>.pem の内容をコピーします。
identity_public_cert	identity_private_key プロパティに対応する X509 証明書。 証明書の形式は、ヘッダとトレイラを含む PEM エンコードです。ヘッダは、たとえば「-----BEGIN CERTIFICATE--」のような形式になります。 このプロパティの値として、Puppet マスターを準備するために作成したファイル <TempCredsDir>/certs/<Name>.pem の内容をコピーします。

プロパティ	説明
<code>ca_cert</code>	<p>Puppet マスターおよび MCollective での認証に使用されるデジタル証明書を発行する認証局の X509 証明書。</p> <p>証明書の形式は、ヘッダとトレイラを含む PEM エンコードです。ヘッダは、たとえば「-----BEGIN CERTIFICATE--」のような形式になります。</p> <p>このプロパティの値として、Puppet マスターを準備するために作成したファイル <code><TempCredsDir>/certs/ca.pem</code> の内容をコピーします。</p>
<code>mc_servers_shared_cert</code>	<p>MCollective クラスタの共有サーバ証明書またはパブリック キー。</p> <p>証明書の形式は、ヘッダとトレイラを含む PEM エンコードです。ヘッダは、たとえば「-----BEGIN CERTIFICATE-----」または「-----BEGIN PUBLIC KEY--」のような形式になります。</p> <p>このプロパティの値として、Puppet マスターを準備するために作成した Puppet エンタープライズのファイル <code><TempCredsDir>/certs/pe-internal-mcollective-servers.pem</code> または Puppet オープンソースのファイル <code><TempCredsDir>/certs/mcollective-servers.pem</code> の内容をコピーします。</p> <p>トレイラは、たとえば「-----END CERTIFICATE-----」または「-----END RSA PUBLIC KEY--」のような形式になります。</p>
<code>mc_messaging_server_password</code>	<p>MCollective サーバ構成ファイルの構成設定 <code>plugin.activemq.pool.1.password</code> に対応するパスワード。</p> <p>たとえば、Puppet エンタープライズの MCollective サーバ構成ファイルは <code>/etc/puppetlabs/mcollective/server.cfg</code> にあります。</p>
<code>identity_cert_name</code>	<p>ID 証明書の名前。</p> <p>この名前は、Puppet マスターの準備中に Puppet マスターに対応する Application Services 証明書を生成したときに設定した文字列です。</p>
<code>puppet_version</code>	<p>Puppet エンタープライズの Puppet エンタープライズスイートバージョンおよび Puppet オープンソースの Puppet パッケージバージョン。</p> <p>たとえば、Puppet エンタープライズバージョンは 3.2.3 で、Puppet オープンソースバージョンは 3.6.2 です。</p>
<code>mc_messaging_server_username</code>	<p>MCollective サーバ構成ファイルの構成設定 <code>plugin.activemq.pool.1.user</code> に対応するユーザー名。</p> <p>たとえば、Puppet エンタープライズの MCollective サーバ構成ファイルは <code>/etc/puppetlabs/mcollective/server.cfg</code> にあります。</p>
<code>mc_messaging_server_port</code>	<p>MCollective サーバ構成ファイルの構成設定 <code>plugin.activemq.pool.1.port</code> に対応するポート。</p> <p>たとえば、Puppet エンタープライズの MCollective サーバ構成ファイルは <code>/etc/puppetlabs/mcollective/server.cfg</code> にあります。</p>
<code>node_manifest_dir</code>	<p>ノード定義が表示されるディレクトリは、Application Services を使用して展開された仮想マシンで配置されます。</p> <p>Puppet マスターの <code>site.pp</code> で、<code>*.pp</code> ファイルをこのディレクトリからインポートできるようにする必要があります。</p>
<code>mc_appd_agent_ddl</code>	<p>MCollective Application Services エージェント DDL ファイルの場所。</p> <p>この値をオーバーライドする必要はありません。</p>
<code>mc_client_setup_script</code>	<p>MCollective Client セットアップ Beanshell スクリプトの場所。</p> <p>この値をオーバーライドする必要はありません。</p>
<code>global_conf</code>	<p>各ノードの Darwin グローバル構成をダウンロードするための URL。</p>
<code>agent_post_install_wait</code>	<p>エージェントのインストール後に待機する秒数。</p>

- 7 [エージェント] セクションに必要な値を入力します。

オプション	説明
pe_installer_payload_base_path	オペレーティングシステムおよびアーキテクチャ固有の Puppet エンタープライズ tar ファイルがホストされる基本 URI。Windows は対象外です。 スクリプトは <code>pe_installer_payload_base_path</code> およびオペレーティングシステムの現在のバージョンとアーキテクチャに基づいた URI をマシンで検出します。この自動検出は、 <code>pe_installer_payload</code> プロパティの値を指定することでスキップできません。自動検出では、特定のディレクトリ構造が https://s3.amazonaws.com/pe-builds/ にある、Puppet Labs でホストされる tar ファイルの構造をミラーしていることを前提としています。
installer_payload_base_path	Windows バージョンの <code>pe_installer_payload_base_path</code> 。
is_enterprise	登録中の Puppet マスターが、値が <code>true</code> に設定された Puppet エンタープライズであるか、または値が <code>false</code> に設定された Puppet オープンソースであるかを示します。

- 8 [エージェント] セクションに任意の値を入力します。

オプション	説明
pe_installer_payload	Puppet Enterprise の tar ファイルをダウンロードするためのリンク。Windows は対象外です。 すべての OS 間で統一されている Puppet Enterprise の tar ファイルの完全な URL を明示的に指定しない限り、この値は空のままにします。値を指定しない場合、スクリプトは <code>pe_installer_payload_base_path</code> およびオペレーティングシステムの現在のバージョンとアーキテクチャに基づいた URI をマシンで検出します。URI の例としては、 https://s3.amazonaws.com/pe-builds/released/3.0.1/puppet-enterprise-3.0.1-all.tar.gz のようになります。
installer_payload	Windows バージョンの <code>pe_installer_payload</code> 。

- 9 [保存] をクリックします。

Application Services で、入力したプロパティおよびエージェントの値が確認され、必要な値が欠けている場合はエラーメッセージが表示されます。

- 10 サーバへの接続を確認するには [接続の検証] をクリックします。

- 11 ソリューション インスタンス名の横にある矢印をクリックし、前のページに戻ります。

- 12 (オプション) Puppet がインストールされたクライアントマシンで、Puppet マスターからパブリック証明書をダウンロードするときに作成した一時的な認証情報ディレクトリを削除するコマンドを入力します。

```
rm -rf <TempCredsDir>
```

Puppet マスター ソリューション インスタンスの登録

Puppet マスター サーバでローカルに実行する Ruby スクリプトを使用している Application Services 展開環境で、Puppet マスター ソリューション インスタンスを登録できます。

Ruby スクリプトの使用に相当する代替方法については、「[Puppet ソリューション インスタンスの作成 \(P. 65\)](#)」を参照してください。

開始する前に

- **アプリケーション クラウド管理者**として Application Services にログインします。
- Ruby バージョン 1.8.7 を使用できる Puppet Master サーバがあり、Puppet 3.2 以上の場合には Ruby バージョン 2.0.x、Puppet 3.5 以上の場合には Ruby バージョン 2.1.x がインストールされている必要があります。

手順

- 1 Puppet マスター サーバにログインします。

- 2 Application Services サーバ URL から Ruby スクリプトをダウンロードします。

```
http://<ApplicationDirectorIP>/artifacts/solutions/puppet/RegisterWithAppD.rb
```

- 3 (オプション) 登録オプションを表示するには、**-h** オプションを使用して Ruby スクリプトを実行します。

```
ruby RegisterWithAppD.rb -h
```

Options:

<code>-i, --appd_ip APPSERVICESIP</code>	Specify the IP address or host name of the
Application	Services instance.
<code>-u, --appd_user USERNAME</code>	Specify the user name of the administrator
account for	the Application Services instance. The default is
	admin.
<code>-p, --appd_password PASSWORD</code>	Specify the password for the administrator
account.	
<code>-t, --appd_tenant TENANT</code>	Specify the name of the tenant assigned to the
user.	
<code>-g, --appd_group GROUP</code>	Specify the business group assigned to the user.
<code>-d, --appd_de_name DENAME</code>	Specify the name of the deployment environment to
	register this puppet master under.
<code>-n, --node_manifest_dir DIR</code>	Specify the directory where Application Services
places	node definition files for servers under its
management.	The default is the 'appd_nodes' sub-directory
	under the
<code>-c, --cert_name CN</code>	site manifest directory.
	Specify the 'subject' for the certificate that
	Application Services uses to authenticate to Puppet
	Master. The default is 'vmware-appd'.
<code>-s, --[no-]skip_mc_agent</code>	Install the Application Services MCollective
agent and	restart the MCollective daemon. Installs the MC agent
	by default. Use <code>-s</code> to skip.

- 4 Ruby スクリプトを実行して、Puppet Master ソリューション インスタンスを Application Services インスタンスと共に登録します。

```
ruby RegisterWithAppD.rb -i <APPSERVICESIP> -u <USERNAME> -p <PASSWORD>
-d <DENAME> -t <TENANT> -g <GROUP>
```

- 5 スクリプトを実行してソリューション インスタンスが正常に登録されたら、通知を受信していることを確認します。
- 6 Application Services の Web インターフェイスにログインします。
- 7 Application Services のタイトルバーでドロップダウン メニューをクリックし、[クラウド]-[展開環境] を選択します。
- 8 ソリューション インスタンスが登録されている既存の展開環境を選択します。
- 9 [ソリューション インスタンス] タブを選択して、ソリューション インスタンスをクリックします。

10 Puppet マスター オープン ソースで Ruby スクリプトを実行している場合は、次の手順を実行します。

- a ツールバーで [編集] をクリックします。
- b [エージェント] セクションで、is_enterprise プロパティの値を false に設定します。
- c [保存] をクリックして終了します。

11 サーバへの接続を確認するには [接続の検証] をクリックします。

正常に接続されたことを知らせる通知が送信され、[接続の検証] ボタンの横に緑色のチェック マークが表示されます。

次に進む前に

Puppet コンテンツを Application Services ライブラリにサービスとしてインポートします。[「ライブラリへの Puppet サービスのインポート \(P. 116\)」](#) を参照してください。

vCloud Director 環境のためのアプリケーション プロビジョニングのセットアップ

10

Application Services を使用して vCloud Director 環境にアプリケーションを展開するには、vCloud Director 5.1.2 または 5.5 をインストールする必要があります。

アプリケーションのプロビジョニングをセットアップするには、1 つ以上のユーザー アカウントが定義された vCloud Director 組織を作成する必要があります。また、組織 vDC を作成し、Application Services でクラウド プロバイダを作成してその組織 vDC に登録できるようにする必要があります。[\[vCloud Director の Application Services のセットアップ \(P. 27\)\]](#) を参照してください。vCloud Director 内にカスタムの仮想マシン テンプレートも作成する必要があります。

Application Services では、クラウド プロバイダとテンプレートを登録する必要があります。クラウド プロバイダを登録するには、vCloud Director 組織を指定する必要があります。クラウド テンプレートを登録したら、vCloud Director vApp テンプレートを Application Services ライブラリ内の論理テンプレートにマップします。次に、展開環境を作成し、その展開環境を内部の vCloud Director 組織 vDC にマップする必要があります。

アプリケーション プロビジョニングのためのクラウド環境の設定と構成に関する主要概念について理解しておく必要があります。[\[主要な概念 \(P. 16\)\]](#) を参照してください。

この章では次のトピックについて説明します。

- [vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)](#)
- [vCloud Director での Windows 仮想マシン テンプレートの作成 \(P. 73\)](#)
- [vCloud Director での Linux 仮想マシン テンプレートの作成 \(P. 77\)](#)
- [vCloud Director カタログからのクラウド テンプレートの構成の確認 \(P. 79\)](#)
- [vCloud Director での既存の仮想マシン テンプレートのアップデート \(P. 79\)](#)
- [OVF フォーマットによる仮想マシン テンプレートのエクスポート \(P. 80\)](#)
- [vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)](#)
- [vCloud Director 展開環境の作成 \(P. 81\)](#)

vCloud Director カスタム テンプレートを作成するための仮想マシンの要件

Application Services では、vCloud Director で作成されたクラウド テンプレートに論理テンプレートをマップします。このようなクラウド テンプレートが Application Services で適切に機能するには、特定の要件を満たす必要があります。

表 10-1. カスタム テンプレートのための仮想マシンの要件

要件	説明
オペレーティングシステム	<p>サポートされているオペレーティングシステムは次のとおりです。</p> <ul style="list-style-type: none"> ■ Red Hat Enterprise Linux 6.1、6.3、6.5 (32 ビットおよび 64 ビット) ■ Windows Server 2008 R2 Enterprise with Service Pack 1 (64 ビット)、Windows Server 2012 R2 (32 ビットおよび 64 ビット) ■ CentOS 5.10、6.1、6.3、6.5 (32 ビットおよび 64 ビット) ■ SUSE Linux Enterprise 11 SP2 (32 ビットおよび 64 ビット) ■ Ubuntu 12.04.2 (32 ビットおよび 64 ビット)
VMware Tools	<p>VMware Tools がインストールされている必要があります。vCloud Director 5.1.2、または 5.5、あるいは vCenter Server 5.0、5.1、または 5.5 のバージョンである必要があります。</p>
CD/DVD ドライブ	<p>vCloud Director 仮想マシンで 1 台以上の CD/DVD ドライブを使用できるようにする必要があります。vCloud Director のドキュメントを参照してください。</p>
JRE	<p>Application Services 仮想アプライアンスで使用可能な以下のパッケージのいずれかからサポートされている JRE をインストールします。</p> <ul style="list-style-type: none"> ■ JRE for Windows <p><code>http://<Application_Services_IP>/agent/jre-1.7.0_72-win64.zip</code></p> <p>[vCloud Director での Windows 仮想マシン テンプレートの構成 (P. 74)] を参照してください。</p> ■ JRE for Linux <ul style="list-style-type: none"> ■ <code>http://<Application_Services_IP>/agent/jre-1.7.0_72-lin32.zip</code> ■ <code>http://<Application_Services_IP>/agent/jre-1.7.0_72-lin64.zip</code> <p>仮想マシン テンプレートに JRE をインストールするには、[vCloud Director での Linux 仮想マシン テンプレートの作成 (P. 77)] を参照してください。</p>
Linux エージェント ブートストラップ サービス	<p>Linux エージェント ブートストラップのパッケージ <code>vmware-appdirector-agent-service</code> を VMware 製品のダウンロード サイト <code>http://vmware.com/web/vmware/downloads</code> からダウンロードします。次のいずれかのパッケージから Linux エージェント ブートストラップ スクリプトをインストールします。</p> <ul style="list-style-type: none"> ■ <code>http://<Application_Services_IP>/agent/vmware-appdirector-agent-service_6.0.0.0-0_x86_64.rpm</code> ■ <code>http://<Application_Services_IP>/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.rpm</code> ■ <code>http://<Application_Services_IP>/agent/vmware-appdirector-agent-service_6.0.0.0-0_amd64.deb</code> ■ <code>http://<Application_Services_IP>/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.deb</code> <p>仮想マシン テンプレートにエージェント ブートストラップ サービスをインストールするには、[vCloud Director での Linux 仮想マシン テンプレートの作成 (P. 77)] を参照してください。</p> <p><code>preparevCACTemplate.sh</code> スクリプトを使用すると、これらのファイルがダウンロードされます。</p>
vCloud Director 上の Windows エージェント ブートストラップ サービス	<p>Application Services 仮想アプライアンスから <code>http://<Application_Services_IP>/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip</code> ファイルをダウンロードします。</p> <p>Windows ベースの仮想マシン テンプレートにエージェント ブートストラップ サービスをインストールするには、[vCloud Director での Windows 仮想マシン テンプレートの構成 (P. 74)] を参照してください。</p>
サポートされている Windows スクリプティング	<p>Application Services は、Windows CMD または PowerShell 2.0 によるスクリプトをサポートしています。</p>

表 10-1. カスタム テンプレートのための仮想マシンの要件 (続き)

要件	説明
サポートされている Linux スクリプティング	Application Services では、Bash によるスクリプトをサポートしています。
Linux コマンド	次の Linux コマンドが仮想マシンで使用可能になっている必要があります。 <ul style="list-style-type: none"> ■ <code>wget</code> ■ <code>md5sum</code> ■ <code>grep</code> ■ <code>sed</code> ■ <code>setsid</code> ■ <code>awk</code> ■ <code>ifconfig</code> ■ <code>apt-get</code> ■ <code>yum</code>
オプションのサービス	トラブルシューティングまたはその他の理由により、Linux <code>ssh</code> ログまたは Windows リモート デスクトップを使用して仮想マシンにリモート アクセスする場合、Linux 用の OpenSSH サーバとクライアントまたは Windows 用の Remote Desktop Service (RDS) がインストールされ、適切に動作している必要があります。

重要 ブート処理は中断してはならないため、最終的なオペレーティング システムのログイン プロンプトに到達するまで、仮想マシンのブート処理が一時的に停止することがないように仮想マシンを構成します。たとえば、仮想マシンが起動するときに、ユーザーの操作を求めるプロセスやスクリプトが表示されないようにします。この要件は、Application Services ライブラリのために作成された仮想マシン テンプレートにのみ適用されます。

vCloud Director での Windows 仮想マシン テンプレートの作成

Windows 仮想マシン テンプレートを使用すると、Application Services 5.1.2 または 5.5 で vCloud Director を使用してアプリケーション ブループリントを作成し、Windows ベースのサービスおよび .NET アプリケーションなどのアプリケーションを展開できます。

Windows 仮想マシン テンプレートを作成するには、事前に vCloud Director 環境を設定しておく必要があります。

- [vCloud Director vApp の作成](#) (P. 74)

Windows ベースの vCloud Director vApp を作成するときは、仮想マシンに VMware Tools をインストールして構成する必要があります。

- [vCloud Director での Windows 仮想マシン テンプレートの構成](#) (P. 74)

Java SE 7 Runtime Environment (JRE) パッケージおよびエージェント ブートストラップ ファイルを Windows 仮想マシン用に構成し、Application Services で正常に機能するようにする必要があります。

- [Windows 仮想マシン テンプレートに対する SID の変更とドメインへの参加の有効化](#) (P. 76)

Windows 仮想マシン テンプレートに対して、[SID の変更] を使用し、[ドメインへの参加] を有効にできます。[SID の変更] オプションを使用すると、展開されているすべての仮想マシンが一意的なセキュリティ ID (SID) を取得できます。このオプションは、Windows 仮想マシン テンプレートを準備する際に有効にします。[ドメインへの参加] オプションを使用すると、展開されている仮想マシンは指定されている Windows Active Directory ドメインのメンバーになることができます。

- [vCloud Director カタログへの Windows 仮想マシン テンプレートの追加](#) (P. 77)

Windows 仮想マシン テンプレートを vCloud Director カタログに追加して、Application Services と連動するように登録します。

vCloud Director vApp の作成

Windows ベースの vCloud Director vApp を作成するときは、仮想マシンに VMware Tools をインストールして構成する必要があります。

開始する前に

- vCloud Director 5.1.2 または 5.5 がインストールおよび構成されていることを確認します。
- Application Services での使用を計画している vCloud 組織が、vCloud Director ライブラリ テンプレートの作成、共有権限を持つユーザー アカウントによって作成および構成されていることを確認します。
- vCloud Director で vApp を作成する方法を理解しておく必要があります。vCloud Director のドキュメントを参照してください。
- 推奨されるストレージとメモリの値、および設定手順用のリモート デスクトップ サービスについては、Windows Server 2008 R2 Enterprise with Service Pack 1 のシステム要件を参照してください。
- VMware Tools をインストールして構成する方法を理解しておく必要があります。

手順

- 1 vCloud Director にログインします。
- 2 1 台の仮想マシンで vApp を作成します。

Windows Server 2008 R2 Enterprise with Service Pack 1 または Windows Server 2012 R2 オペレーティングシステムとアプリケーションのサポートに適した仮想ストレージと仮想メモリを割り当てます。

- 3 [完全インストール] オプションを使用して、Windows Server 2008 R2 Enterprise with Service Pack 1 または Windows Server 2012 R2 を仮想マシンにインストールします。
- 4 仮想マシンに VMware Tools をインストールします。

次に進む前に

サポートされている Java SE 7 Runtime Environment パッケージと Application Services エージェント ブートストラップ ファイルを、Windows 仮想マシンにインストールします。[「vCloud Director での Windows 仮想マシン テンプレートの構成 \(P. 74\)」](#) を参照してください。

vCloud Director での Windows 仮想マシン テンプレートの構成

Java SE 7 Runtime Environment (JRE) パッケージおよびエージェント ブートストラップ ファイルを Windows 仮想マシン用に構成し、Application Services で正常に機能するようにする必要があります。

エージェント ブートストラップは、仮想マシンを最初に起動した後に Windows で NT サービスとして実行されるデーモンです。Application Services エージェント ブートストラップは、PowerShell スクリプトを使用してブートストラップコードを実行します。

開始する前に

- vCloud Director 5.1.2 または 5.5 がインストールおよび構成されていることを確認します。
- 少なくとも 1 つの vApp が vCloud Director で作成されていることを確認します。[「vCloud Director vApp の作成 \(P. 74\)」](#) を参照してください。

手順

- 1 ローカルの管理者グループに属しているユーザーとして Windows 仮想マシンにログインします。
- 2 サポートされている Java SE 7 Runtime Environment を http://<Application_Director_IP>/agent/jre-1.7.0_51-win64.zip からダウンロードしてインストールします。
- 3 JRE ファイルを `C:\opt\vmware-jre\` ディレクトリに解凍します。

- 4 PowerShell コマンドウィンドウで **C:\opt\vmware-jre\bin** ディレクトリに移動し、**java -version** を入力してインストールを確認します。
インストールされている Java のバージョンが表示されます。
- 5 Application Services エージェント ブートストラップ ファイルを http://<Application_Director_IP>/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip からダウンロードしてインストールします。
- 6 **vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip** ファイルを新しいフォルダに解凍します。
- 7 **agent_bootstrap.ps1** ファイルのプロパティを右クリックして表示し、[ブロック解除] をクリックしてファイルのセキュリティを無効にします。

重要 この Windows セキュリティ機能を無効にしないと、Application Services エージェント ブートストラップ ファイルを使用できません。

- 8 **NTRights.exe** ユーティリティを新たに作成したフォルダに追加します。
NTRights.exe ユーティリティを **install.bat** コマンドで使用して、エージェント ブートストラップ サービスを darwin ユーザー アカウントで実行されるよう構成します。このユーティリティは Windows Server 2003 Resource Kit Tools (**rktools.exe**) に含まれ、Microsoft ダウンロード Web サイトからダウンロードできます。
- 9 (オプション) **NTRights.exe** ユーティリティが使用できない場合は、**install.bat** コマンドを実行した後に、エージェント ブートストラップ サービスが特定のユーザー アカウントで実行されるように手動で構成します。
install.bat コマンドでは、エージェント ブートストラップ サービス用の darwin と呼ばれるユーザー アカウントが作成され、設定したパスワードが使用されます。<Password> は、Windows のパスワード要件を満たす必要があります。
- 10 エージェント ブートストラップ サービスをインストールするには、Windows CMD コンソールを開き、新しいフォルダに移動して、**install.bat password=<Password> cloudProvider=vcd** と入力します。
- 11 コマンドラインから **services.msc** と入力して Application Services エージェント ブートストラップ サービスを開きます。
- 12 [ログオン] タブで [としてログオン] オプションを [このアカウント] に設定し、ログイン認証情報を入力します。
このユーザー アカウントのユーザー名は **.\darwin** です。
- 13 PowerShell コマンドウィンドウで、次のサービス関連タスクを実行します。
 - a **net start AppDAgentBootstrap** と入力して、サービスが正常に開始されることを確認します。
 - b **net stop AppDAgentBootstrap** と入力してサービスを停止します。
- 14 テンプレート作成中に生成されたランタイム ログ ファイルを削除します。
agent_reset.bat
- 15 (オプション) この仮想マシンの将来のインスタンスにリモートでアクセスできるようにリモート デスクトップを有効にします。
リモート アクセスをトラブルシューティングに使用することもできます。
- 16 Windows 仮想マシンをシャットダウンし、vCloud Director で vApp を停止します。
- 17 **C:\opt\vmware-appdirector\agent-bootstrap\agent_bootstrap.ps1** に移動し、\$gcStatus の値を \$True に変更して、**Write-Host "Skipping guest customization check for vCD 5.1.2"** コマンドをファイルに追加します。

次に進む前に

vApp を vCloud Director のカタログに保存する前に、SID の変更およびドメインへの参加機能を選択して、Windows テンプレートを Windows Active Directory ドメインのメンバーにできます。[\[Windows 仮想マシン テンプレートに対する SID の変更とドメインへの参加の有効化 \(P. 76\)\]](#) を参照してください。

vApp テンプレートを vCloud Director カタログに追加し、Application Services で使用できるように登録できます。[\[vCloud Director カタログへの Windows 仮想マシン テンプレートの追加 \(P. 77\)\]](#) を参照してください。

Windows 仮想マシン テンプレートに対する SID の変更とドメインへの参加の有効化

Windows 仮想マシン テンプレートに対して、[SID の変更] を使用し、[ドメインへの参加] を有効にできます。[SID の変更] オプションを使用すると、展開されているすべての仮想マシンが一意的なセキュリティ ID (SID) を取得できます。このオプションは、Windows 仮想マシン テンプレートを準備する際に有効にします。[ドメインへの参加] オプションを使用すると、展開されている仮想マシンは指定されている Windows Active Directory ドメインのメンバーになることができます。

企業管理のドメインに参加する必要がある Windows 仮想マシンを多数展開する場合は、構成済みのドメイン設定を含む Windows テンプレートを使用します。より小規模な展開または異なるドメインに参加する柔軟性を必要とする展開に対して効果的な方法は、構成済みのドメイン設定なしの Windows テンプレートを使用し、展開の実行計画にドメインへの参加の事前定義タスクを追加します。[\[ドメインへの参加用事前定義済みタスクの追加 \(P. 203\)\]](#) を参照してください。

開始する前に

- Windows vApp を正しく作成し、仮想マシンのテンプレートを正しく構成していることを確認します。[\[vCloud Director vApp の作成 \(P. 74\)\]](#) および [\[vCloud Director での Windows 仮想マシン テンプレートの構成 \(P. 74\)\]](#) を参照してください。
 - Windows のドメイン名、ドメイン アカウント、およびアカウントパスワードがすぐに使用できることを確認します。
 - 指定されているドメイン コントローラに仮想マシンが接続できることを確認します。
- DHCP サービスがネットワークに存在する必要があります。そうすれば、DHCP によって指定されている DNS サーバがゲストのカスタマイズ設定で指定されているドメイン名を解決できます。

手順

- 1 ソース テンプレートから vApp をインスタンス化し、関連付けられている Windows 仮想マシンを探します。
- 2 ローカルの管理者グループに属しているユーザーとして Windows 仮想マシンにログインします。
- 3 仮想マシンのすべてのネットワーク インターフェイスで DHCP を使用するよう構成します。
 - a ネットワーク コントロール パネルを開きます。
 - b [ネットワークと共有センター] - [アダプタ設定の変更] を選択します。
 - c それぞれのネットワーク接続を開き、[プロパティ] を選択します。
 - d [IP アドレスを自動的に取得する] と [DNS サーバのアドレスを自動的に取得する] にインターネット プロトコルバージョン 4/6 を設定します。
- 4 Windows 仮想マシンをシャットダウンし、vCloud Director で vApp を停止します。
- 5 vCloud Director で、仮想マシンのプロパティを開き、[ゲスト OS のカスタマイズ] タブを選択して、[ゲストのカスタマイズ] を有効にします。
- 6 [ゲスト OS のカスタマイズ] タブで、[SID の変更] を選択します。
- 7 [ドメインへの参加] を選択して、Windows のドメイン名、ドメイン アカウント、およびアカウントパスワードを入力します。
- 8 [ライブラリに追加] を選択して、vApp から新しい vApp テンプレートを作成します。

展開中に、Active Directory はログイン認証情報を認証し、Windows 仮想マシンがドメインのメンバーになることを許可します。

次に進む前に

vCloud Director ライブラリに vApp テンプレートを追加します。[\[vCloud Director カタログへの Windows 仮想マシン テンプレートの追加 \(P. 77\)\]](#) を参照してください。

vCloud Director カタログへの Windows 仮想マシン テンプレートの追加

Windows 仮想マシン テンプレートを vCloud Director カタログに追加して、Application Services と連動するように登録します。

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- vCloud Director 5.1.2 または 5.5 がインストールおよび構成されていることを確認します。
- Windows テンプレートが Application Services 仮想マシンの要件を満たしていることを確認します。[\[vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)\]](#) を参照してください。
- Windows vApp の作成と、仮想マシン テンプレートの構成に必要な準備タスクを完了します。[\[vCloud Director vApp の作成 \(P. 74\)\]](#) および [\[vCloud Director での Windows 仮想マシン テンプレートの構成 \(P. 74\)\]](#) を参照してください。

手順

- 1 vCloud Director Web ユーザー インターフェイスにログインします。
- 2 Windows vApp が存在する vCloud Org の場所に移動します。
- 3 vApp を選択し、[ライブラリに追加] をクリックして、ライブラリに vApp テンプレートを作成します。

Windows テンプレートは、論理テンプレートをクラウドテンプレートにマッピングする際に選択できるクラウドテンプレートのリストに追加されます。

次に進む前に

Application Services で使用するために Windows テンプレートを登録します。[\[vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)\]](#) を参照してください。

vCloud Director での Linux 仮想マシン テンプレートの作成

Linux 仮想マシン テンプレートを使用すると、vCloud Director で Application Services を使用してアプリケーションブループリントを作成し、Linux ベースのサービスとアプリケーションを展開できます。

開始する前に

vCloud Director

- vCloud Director 5.1.2 または 5.5 がインストールおよび構成されていることを確認します。
- Application Services での使用を計画している vCloud 組織が、vCloud Director カタログ テンプレートの作成、共有権限を持つユーザー アカウントによって作成および構成されていることを確認します。
- vCloud Director で vApp を作成する方法を理解しておく必要があります。vCloud Director のドキュメントを参照してください。

複数の仮想マシンの vApp テンプレートの作成はサポートされていません。

- vApp が Application Services 仮想マシンの要件を満たしていることを確認します。[\[vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)\]](#) を参照してください。

- VMware Tools をインストールして構成する方法を理解しておく必要があります。

手順

- 1 vCloud Director にログインし、1 つの仮想マシンで vApp を作成します。
適切な仮想ストレージと仮想メモリを割り当て、Linux オペレーティングシステムとアプリケーションをサポートします。
- 2 サポートされている Linux オペレーティングシステムを仮想マシンにインストールします。
- 3 仮想マシンに VMware Tools をインストールします。
- 4 root 権限を使用して仮想マシンにログインしてターミナルを開きます。
- 5 YUM パッケージの管理ツールまたは APT-GET パッケージツールを使用して、サポートされている JRE パッケージをダウンロードしてインストールします。

サポートされている JRE パッケージがインストールされている場合は、この手順をスキップします。

- a `http://<Application_Director_IP>/agent/jre-1.7.0_51-lin-<ArchitectureName>.zip` から JRE パッケージをダウンロードします。ここで、<ArchitectureName> は 32 または 64 です。
 - b `/opt/vmware-jre` ディレクトリを作成します。
 - c JRE パッケージを解凍します。
`unzip -d /opt/vmware-jre`
 - d 「`/opt/vmware-jre/bin/java -version`」と入力して、インストールを確認します。
インストールされている Java のバージョンが表示されます。
 - e `/usr/java/default`、`/usr/java/latest`、および `/usr/java/1.7.0_21` の各ディレクトリから既存の JRE パッケージを削除します。
- 6 Application Services エージェント ブートストラップ サービスをダウンロードしてインストールします。
 - a ディレクトリを `/tmp/` フォルダに変更します。
 - b RPM ベースの仮想マシンの場合は、`http://<Application_Director_IP>/agent/vmware-appdirector-agent-service_6.0.0.0-0_<ArchitectureName>.rpm` からこのフォルダにエージェント ブートストラップをダウンロードし、ターミナルで
`rpm -i vmware-appdirector-agent-service_6.0.0.0-0_<ArchitectureName>.rpm` と入力してサービスをインストールします。ここで、<ArchitectureName> は、i386 (32 ビット) または x86_64 (64 ビット) です。
 - c DEB ベースの仮想マシンの場合は、`http://<Application_Director_IP>/agent/vmware-appdirector-agent-service_6.0.0.0-0_<ArchitectureName>.deb` からこのフォルダにエージェント ブートストラップをダウンロードし、ターミナルで
`dpkg -i vmware-appdirector-agent-service_6.0.0.0-0_<ArchitectureName>.deb` と入力してサービスをインストールします。ここで、<ArchitectureName> は、i386 (32 ビット) または amd64 (64 ビット) です。
 - 7 Linux 仮想マシンをシャットダウンし、vCloud Director で vApp を停止します。
 - 8 Linux vApp が存在する vCloud Org の場所に移動します。
 - 9 vApp を選択し、[ライブラリに追加] をクリックして、ライブラリに vApp テンプレートを作成します。

次に進む前に

Application Services で使用するために Linux テンプレートを登録します。[「vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)」](#) を参照してください。

vCloud Director カタログからのクラウド テンプレートの構成の確認

Application Services でクラウド テンプレートを使用する前に、展開エラーを回避できるようにテンプレート構成を確認します。

クラウド テンプレートにカスタマイズ スクリプトが存在する場合、Application Services はそのスクリプトを自身のゲスト カスタマイズ スクリプトで上書きします。ゲスト カスタマイズ スクリプトを使用して仮想マシンをセットアップすると、その仮想マシンは Application Services サーバと通信して展開プロセスを完了できます。

開始する前に

- vCloud Director 5.1.2 または 5.5 がインストールおよび構成されていることを確認します。
- 仮想マシン テンプレートが vCloud Director にアップロードされており、vCloud Director カタログに vApp テンプレートとして登録されていることを確認します。

手順については、vCloud Director のドキュメントを参照してください。

手順

- 1 vCloud Director の Web インターフェイスから、クラウド環境に対する vApp のコピー作成申請を手動で開始します。
- 2 ゲスト カスタマイズ プロセスが成功したことを確認するには、`/var/log/vmware-vmc/customization.log` にあるログ ファイルを開いて、カスタマイズが **成功** ステータスで完了したという記録が残っているかどうかを確認します。
- 3 vCloud Director から IP アドレスを割り当てる場合は、アクティブで正しい IP アドレスが存在することを確認します。
- 4 `/opt/vmware-appdirector/agent/logs/agent_bootstrap.log` にあるエージェント ブートストラップ ログ ファイルを開いて、内容を確認します。
- 5 Application Services エージェント ブート サービスが実行され、Application Services エージェントの JAR ファイルを Application Services サーバからダウンロードしようとしたことを確認します。

この時点では、JAR ファイルのダウンロードは失敗します。ダウンロード プロセスは、Application Services が Application Services の実行プランから vApp を展開したときに成功します。

次に進む前に

[[vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)](#)] に示した項目が vApp に存在しており、それらが正しく機能していることを確認します。

vCloud Director での既存の仮想マシン テンプレートのアップデート

既存の Linux または Windows のテンプレートのコンテンツをアップデートするか、またはそのコンテンツを使用してテンプレートを新規作成するには、適切なコマンドを実行してエージェント ブートストラップ サービスを削除する必要があります。

Linux テンプレートの場合、`agent_reset.sh` コマンドによって Application Services エージェント ブートストラップのステータスがリセットされ、既存のランタイム ログ ファイルが削除されます。仮想マシンに root としてログインし、このコマンドを実行できます。

```
/opt/vmware-appdirector/agent-bootstrap/agent_reset.sh
```

Windows テンプレートの場合、**agent_reset.bat** コマンドによって既存のランタイム ログ ファイルが削除されます。PowerShell コマンド ウィンドウで、次のコマンドを入力します。

```
\opt\vmware-appdirector\agent-bootstrap\agent_reset.bat
```

該当するコマンドを実行したら、vApp インスタンスを新しい vApp テンプレートとしてカタログに再度配置することができます。

OVF フォーマットによる仮想マシン テンプレートのエクスポート

Application Services を使用すると、仮想マシン テンプレートを OVF フォーマットで作成して、vCloud Director カタログにエクスポートできます。

仮想マシン テンプレートを OVF フォーマットで vCloud Director カタログにエクスポートすることを計画している場合は、エクスポート パッケージにカスタム OVF プロパティを追加しないでください。カスタム OVF プロパティによって、vCloud Director のゲスト OS のカスタマイズ機能が無効になります。この機能が無効になると、Application Services は仮想マシン テンプレートにアクセスできません。

vCloud Director クラウド プロバイダとテンプレートの登録

Application Services は、vCloud Director インスタンスと組織に接続して、クラウド プロバイダを登録します。vCloud Director ユーザーがアクセスできるカタログから、仮想マシン テンプレートにアクセスできます。vCloud 組織の vDC は展開環境にマッピングされます。

vCloud Director を Application Services と併用するための設定の詳細については、[「Application Services のインストール準備 \(P. 23\)」](#) を参照してください。

展開環境にリンクされたクラウド プロバイダは削除できません。展開環境にリンクされていないクラウド プロバイダを削除すると、論理テンプレートからのすべてのクラウド テンプレート マッピングが削除されます。

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- vCloud Director 5.1.2 または 5.5 がインストールおよび構成されていることを確認します。
- Application Services の要件を満たす仮想マシン テンプレートを作成します。[「vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)」](#) を参照してください。
- Application Services との使用を計画している vCloud 組織が、組織の管理者権限で作成および構成されていることを確認します。
- Application Services でクラウド プロバイダのクラウド テンプレートをマップする場合、マッピング先の vCloud Director テンプレートが単一の仮想マシンであることを確認します。複数の仮想マシン vApp テンプレートはサポートされません。

複数の仮想マシンがあるテンプレートは Application Services クラウド プロバイダに登録できません。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [クラウド プロバイダ] を選択します。
- 2 [新規] をクリックします。
- 3 クラウド プロバイダ情報を作成します。

オプション	説明
名前と説明	vCloud Director 組織名をこれらのフィールドのいずれかに含めます。これらのフィールドのテキストは、[Deployment Profile] ウィザードの [クラウド プロバイダ] 列に表示されます。
クラウド プロバイダのタイプ	vCloud Director 5.1.2 または 5.5 がサポートされるタイプです。

オプション	説明
クラウド IP/ホスト名	クラウド IP またはホスト名は、vCloud Director インスタンスの IP アドレスまたはホスト名と一致する必要があります。
組織名	組織名は、vCloud Director の組織の名前と一致する必要があります。
ユーザー名とパスワード	その組織の組織管理者権限を持つ組織ユーザーの認証情報。このユーザーを vCloud Director システム管理者にすることはできません。 注意 ユーザー名に @ 記号が含まれている場合、ログイン認証情報を入力するときに %40 に置き換えてください。

- 4 入力した値が正しいかどうかをテストするには、[接続の検証] をクリックします。
- 5 [保存] をクリックして、クラウド プロバイダを登録します。
クラウド プロバイダの必須エントリが無効または空白になっている場合は、[保存] をクリックしたときに、それらを修正するように求めるプロンプトが表示されます。
- 6 vCloud Director 仮想マシン テンプレートを登録するには、ツールバーで [編集] をクリックします。
- 7 [テンプレート] セクションで、[新規] をクリックしてテンプレートを登録します。
- 8 ドロップダウン メニューからカタログを選択して [テンプレートの取得] をクリックします。
複数の仮想マシンがあるテンプレートは Application Services クラウド プロバイダに登録できません。
単一の仮想マシンがあるクラウド テンプレートがドロップダウン メニューに表示されます。
- 9 リスト内のテンプレートの横にあるチェック ボックスをオンにし、[OK] をクリックします。
登録したテンプレートは、論理テンプレートをクラウド テンプレートにマッピングする際に選択できるクラウド テンプレートのリストに追加されます。
- 10 vCloud Director クラウド プロバイダの既存の情報を変更するには、ツールバーで [編集] をクリックして変更を行ってから、[保存] をクリックします。

クラウド プロバイダが登録され、その仮想マシン テンプレートおよび仮想データセンターが Application Services で使用できるようになります。

次に進む前に

これらの vCloud Director クラウド テンプレートを Application Services 論理テンプレートにマップします。[「ライブラリへの論理テンプレートの追加 \(P. 125\)」](#) を参照してください。

展開環境を作成し、アプリケーションを vCloud Director 環境に展開します。[「vCloud Director 展開環境の作成 \(P. 81\)」](#) を参照してください。

vCloud Director 展開環境の作成

アプリケーションを展開する前に、展開環境を vCloud Director 組織 vDC にマップする必要があります。

vCloud Director の vDC は、仮想システムを格納、展開、操作できる環境を提供します。たとえば、展開、テスト、ステージング、本番で個別の展開環境にすることもできます。vCloud Director の場合、展開環境は組織 vDC にマップされます。

展開環境が現在展開プロファイルまたは現在の展開で使用されている場合、削除できません。

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- 少なくとも 1 つの vCloud Director クラウド プロバイダが Application Services に登録されていることを確認します。[「vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)」](#) を参照してください。
- 組織 vDC が vCloud Director 組織で作成され、構成されていることを確認します。Application Services 展開環境は、vCloud Director の組織 vDC に相当します。

vCloud Director を Application Services と併用するための設定の詳細については、[「Application Services のインストール準備 \(P. 23\)」](#) を参照してください。

- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。第 12 章「[Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [展開環境] を選択します。
- 2 [新規] をクリックします。
- 3 展開環境の情報を入力します。

オプション	説明
展開環境の名前と説明	vDC 名をこれらのフィールドのいずれかに含めます。これらのフィールドのテキストは [展開プロファイル] ウィザードの [展開環境] 列下に表示されます。
クラウド プロバイダ	クラウド プロバイダがリストに表示されない場合、ダイアログ ボックスをキャンセルし、[クラウド] - [クラウド プロバイダ] を選択して、vCloud Director クラウド プロバイダを追加します。
組織 vDC URI	[選択] をクリックし、選択したクラウド プロバイダが提供する vDC のリストから選択します。

- 4 組織 vDC を選択して [OK] をクリックします。
vCloud Director の組織 vDC は Application Services 展開環境名にマップされます。
- 5 [保存] をクリックします。
- 6 (オプション) vCloud Director 展開環境の既存の情報を変更するには、ツールバーで [編集] をクリックして変更を行ってから、[保存] をクリックします。

作成した展開環境が展開環境のリストに追加され、展開プロファイルを作成するときにそこから選択できます。

次に進む前に

外部サービス インスタンスのマッピング、ポリシー インスタンスの作成、または展開環境へのソリューション インスタンスの登録を実行できます。[「外部サービス インスタンスのマッピング \(P. 61\)」](#)、[「ポリシー インスタンスの作成 \(P. 62\)」](#) または [「Puppet ソリューション インスタンスの作成 \(P. 65\)」](#) を参照してください。

Amazon EC2 環境でのアプリケーションプロビジョニングの設定

11

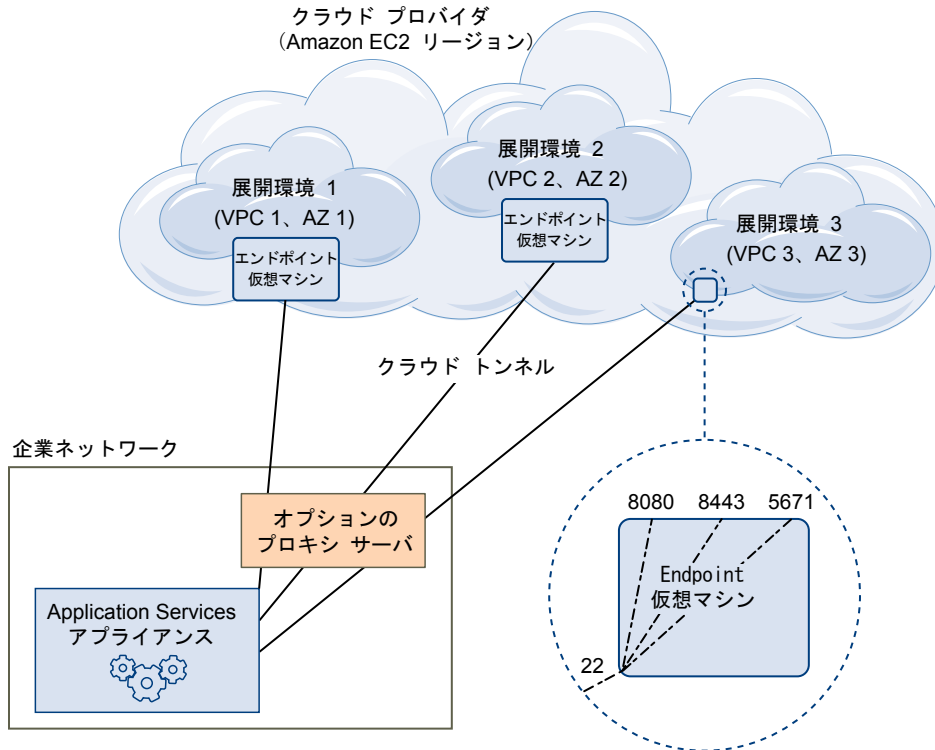
Application Services を使用して Amazon EC2 環境にアプリケーションを展開するには、1 つ以上のクラウド環境をセットアップする必要があります。

Amazon EC2 クラウド環境でアプリケーション プロビジョニングをセットアップするには、Amazon Web Services (AWS) ユーザー アカウントのサインアップと Amazon リージョンの VPC の作成および構成が必要です。Application Services を使用すると、クラウド プロバイダを作成し、それらを Amazon リージョンの VPC に接続できます。仮想マシン テンプレートを Amazon EC2 環境に作成する必要があります。アプリケーションの展開時にこれらのテンプレートを使用して仮想マシンを作成できます。

Application Services では、クラウド プロバイダとテンプレートを登録する必要があります。クラウド プロバイダを登録するには、Amazon EC2 リージョンを指定する必要があります。クラウド テンプレートを登録したら、クラウド環境の AMI を Application Services ライブラリの論理テンプレートにマップします。その後、展開環境を作成し、その展開環境を Amazon 仮想プライベート クラウド (VPC) と可用性ゾーンの組み合わせにマッピングする必要があります。

Application Services アプリケーションを Amazon EC2 に展開するには、VPC 内に Endpoint 仮想マシンと呼ばれる特別な仮想マシンを作成する必要があります。Endpoint 仮想マシンは、企業ネットワーク内の Application Services アプライアンスから Amazon VPC 環境までセキュアなチャネル（クラウド トンネルと呼ばれる）を経由して通信する場合の通信先の役割を果たします。VPC にアプリケーションを展開する際には、Endpoint 仮想マシンを介して、Application Services アプライアンスと通信します。クラウド トンネルは CLI を使用して作成および管理できます。[「クラウド トンネルの管理 \(P. 223\)」](#) を参照してください。

図 11-1. Application Services と Amazon EC2 のトポロジ



アプリケーション プロビジョニングのためのクラウド環境の設定と構成に関する主要概念について理解しておく必要があります。「[主要な概念 \(P. 16\)](#)」を参照してください。

この章では次のトピックについて説明します。

- [Application Services の Amazon EC2 環境を構成する \(P. 84\)](#)
- [Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 \(P. 88\)](#)
- [Amazon EC2 仮想マシン テンプレートまたは AMI の作成 \(P. 89\)](#)
- [Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)](#)
- [Amazon EC2 の展開環境の作成 \(P. 92\)](#)

Application Services の Amazon EC2 環境を構成する

Amazon EC2 環境を設定するには、Amazon リージョン内にアプリケーション展開のターゲット場所として VPC を作成する必要があります。この VPC を Application Services アプリケーションが展開されるように構成する必要があります。

Amazon EC2 に展開する場合、外部ネットワークの NIC は弾性 IP アドレスを受信します。また、展開環境でインスタンス同士が通信できるように、展開ごとに新しいセキュリティ グループが作成されます。このセキュリティ グループでは、80 番、8080 番、8081 番、8443 番、および 22 番の各ポートに対する外部からのアクセスを許可しています。上記以外のポートを開く必要がある場合は、Amazon EC2 の管理コンソールを使用して新しいセキュリティ グループを特定し、適切なルールを追加します。

開始する前に

- Amazon AWS ユーザー アカウントを設定します。

- Amazon では、Amazon EC2 インスタンスの数、アカウントの弾性 IP アドレスの数、および API 呼び出しの数にデフォルトの上限値を設けています。インスタンス、弾性 IP アドレス、または API 呼び出しの上限値を増やすよう依頼する場合は、Amazon サポート部門までお問い合わせください。Amazon EC2 の詳細については、『Amazon AWS ドキュメント』を参照してください。

手順

- 1 AWS ユーザー アカウントを使用して AWS 管理コンソールにログインします。
- 2 [AWS Identity and Access Management (IAM)] ダッシュボードで、管理者アクセス権限またはパワー ユーザー アクセス権限を持つ新規のユーザーを設定し、そのユーザーのアクセス キーを生成します。
新規の IAM ユーザーを設定せずに、マスター AWS ユーザー アカウントにアクセス キーを設定してもかまいません。
- 3 後で Application Services で使用するために、このアクセス キーをダウンロードおよび保存します。
- 4 Amazon EC2 の展開済み仮想マシンにログインするために、titan_keypair という SSH キー ペアを生成し、対応するプライベート キーを保存します。
- 5 VPC を作成します。AWS 管理コンソールの VPC ダッシュボードで、パブリック サブネットおよびプライベート サブネットを持つ VPC を作成します。
 - a ウィザード オプションを使用して、パブリック サブネットおよびプライベート サブネットを持つ VPC を作成します。
2 つのサブネットが作成されます。どちらのサブネットのインスタンスもインターネットに対するアクセス権を持ちます。サブネットは、アプリケーションの展開中にコンテンツをダウンロードするときに便利です。
 - b サブネットの可用性ゾーンを指定します。
両方のサブネットに同じ可用性ゾーンを使用できます。
 - c titan_keypair SSH キー ペアを使用します。
自分自身のキー ペアを使用することもできます。
- 6 VPC で、EndpointAccess という名前のセキュリティ グループを作成し、送受信ポート アクセス ルールを設定します。

ポート アクセス	送信元と宛先
Inbound 22	0.0.0.0/0
Inbound ALL	<SubnetofVPC>
Outbound ALL	0.0.0.0/0
(オプション) Inbound 2222	0.0.0.0/0

次に進む前に

VPC 内に Linux 仮想マシンを作成します。[「エンドポイント VM を作成する \(P. 85\)」](#) を参照してください。

エンドポイント VM を作成する

Application Services を使用してアプリケーションを展開するには、VPC で Endpoint 仮想マシンと呼ばれる Linux 仮想マシンを Amazon EC2 に作成します。

Application Services アプライアンスは複数の Endpoint 仮想マシンと連携できます。ただし、Endpoint 仮想マシンは、単一 Application Services アプライアンスの Amazon リージョン、VPC、および可用性ゾーンのエンドポイントとしてのみ指定できます。

開始する前に

- Amazon AWS ユーザー アカウントが設定されていることを確認します。

- VPC が、アプリケーションを展開する場合に Amazon リージョン内でターゲットの場所として使用できることを確認します。[「Application Services の Amazon EC2 環境を構成する \(P. 84\)」](#) を参照してください。

手順

- 1 Application Services を使用して管理するには、1 つの Endpoint 仮想マシンを各 VPC にインストールします。
- 2 Amazon ベースの CentOS 仮想マシンを外部サブネットで作成します。
 - a amazon/ami-vpc-nat-1.0.0-beta.i386-ebc AMI または amazon/ami-vpc-nat-1.0.0-beta.x86_64-ebc AMI のいずれかを使用します。
AMI は、IP マスカレード デバイスとして機能するように事前構成されています。
 - b この仮想マシン用に EndpointAccess セキュリティ グループを使用します。
- 3 VPC で使用するために弾性 IP アドレスを設定し、その IP アドレスを Endpoint 仮想マシンに割り当てます。
- 4 OpenSSH クライアントを使用して、その弾性 IP アドレスとプライベート キーで Endpoint 仮想マシンにログインします。

```
ssh -i <PathToPrivateKeyFile> ec2-user@<ApplicationDirectorEndpointVM>
```
- 5 CLI で `/etc/ssh/sshd_config` 構成ファイルを開き、**GatewayPorts yes**、**ClientAliveInterval 30**、および **ClientAliveCountMax 3** 行を追加します。
- 6 (オプション) 送信 SSH が企業ファイアウォールによってブロックされる場合、ファイアウォール管理者に SSH トラフィック用の代替ポートを 2222 に設定するように依頼してください。
Port 2222 行を `/etc/ssh/sshd_config` 構成ファイルに追加します。
- 7 SSH デーモンを再開します。

```
sudo service sshd restart
```
- 8 内部ポート 80 アクセスを 8080 に経路再設定する iptable ルールを作成します。

```
sudo iptables -t nat -I PREROUTING --source 0/0 --destination internal-ip-address-of-endpoint-vm -p tcp --dport 80 -j REDIRECT --to-ports 8080
```
- 9 Endpoint 仮想マシンからログアウトします。

次に進む前に

Amazon EC2 展開にクラウドトンネルをセットアップして Application Services サーバと通信します。[「Amazon EC2 に接続するためのクラウド トンネルの作成 \(P. 86\)」](#) を参照してください。

Amazon EC2 に接続するためのクラウド トンネルの作成

Amazon EC2 VPC に展開されたアプリケーションが Application Services サーバと通信できるように、クラウドトンネル インスタンスを作成し有効にする必要があります。

Application Services から Endpoint 仮想マシンまでのネットワークは、展開される Amazon EC2 インスタンスごとに最低 1Mbps のアップロード帯域幅を確保する必要があります。ダウンロードを高速化するには、アプリケーションを Application Services アプライアンスからクラウドトンネル経由でダウンロードする代わりに、Amazon Simple Storage Service に格納しておきます。

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- Endpoint 仮想マシンが正しく設定および構成されていることを確認します。[「Application Services の Amazon EC2 環境を構成する \(P. 84\)」](#) を参照してください。
- Endpoint 仮想マシンの弾性 IP アドレスとプライベート IP アドレスが使用可能であることを確認します。

- 自社ネットワークから Endpoint 仮想マシンまでのクラウド トンネルを確立するために Endpoint 仮想マシンのプライベート キーが使用可能であることを確認します。
- Application Services が実行されているネットワークから Amazon EC2 にアクセスするためにプロキシ サーバが必要かどうかを確認します。
 プロキシ サーバまたはネットワークは、ネットワーク外部に対して標準 SSH 22 番ポートへのアクセスを許可する必要があります。
- Application Services CLI を起動します。[「リモートでの CLI の開始 \(P. 221\)」](#) を参照してください。

手順

- 1 SSH クライアントを使用して、Endpoint 仮想マシンのダウンロード済みプライベート キー ファイルを Application Services アプライアンスにコピーし、同ファイルを `/tmp` ディレクトリにコピーします。

- 2 `roo` シェルで、セキュアなクラウド トンネル インスタンスを作成します。

```
create-cloud-tunnel --name <TunnelName> --description <"TunnelDescription"> --
enabled false --externalAddress <EndpointVMElasticIP>
--sshPort 22 --internalAddress <EndpointVMPrivateIP> --proxyUrl <ProxyURL> --
username ec2-user --privateKeyPath <PrivateKeyFilePath>
```

`--sshPort` パラメータを使用して、22 番以外のポートを指定できます。`--proxyUrl` は、Endpoint 仮想マシンとの接続に使用するプロキシ サーバを指定するためのオプションのパラメータです。

- 3 セキュアなクラウド トンネル接続を有効にします。

```
enable-cloud-tunnel --name <TunnelName>
```

- 4 セキュアなクラウド トンネル接続が確立されていることを確認します。

```
test-cloud-tunnel --name <TunnelName>
```

セキュアなクラウド トンネル接続の確認には数分かかることがあります。

Application Services によって、Amazon EC2 の Endpoint 仮想マシンへの接続が確立されます。

次に進む前に

仮想マシン要件を満たすように考慮して、Amazon EC2 AMI を作成します。[「Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 \(P. 88\)」](#) および [「Amazon EC2 仮想マシン テンプレートまたは AMI の作成 \(P. 89\)」](#) を参照してください。

既存のクラウド トンネル接続を CLI で管理することもできます。[「クラウド トンネルの管理 \(P. 223\)」](#) を参照してください。

Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件

サポートされている Linux オペレーティングシステムを使用して、アプリケーション用のカスタム仮想マシンテンプレートまたは Amazon マシン イメージ (AMI) を作成できます。既存の AMI または仮想マシン テンプレートをカスタマイズして、Application Services で使用することもできます。

表 11-1. カスタム AMI の仮想マシンの要件

要件	説明
オペレーティングシステム	CentOS 6.3
JRE	<p>JRE 1.7.0 をインストールする必要があります。サポートされている推奨 JRE は、Application Services 仮想アプリケーションで使用できるいずれかのパッケージからインストールできます。</p> <p>JRE for Linux</p> <ul style="list-style-type: none"> ■ <a href="http://<Application_Director_IP>/agent/jre-7u45-linux-amd64.rpm">http://<Application_Director_IP>/agent/jre-7u45-linux-amd64.rpm ■ <a href="http://<Application_Director_IP>/agent/jre-7u45-linux-i586.rpm">http://<Application_Director_IP>/agent/jre-7u45-linux-i586.rpm <p>JRE を RPM ベースの AMI にインストールするには、「Amazon EC2 仮想マシン テンプレートまたは AMI の作成 (P. 89)」 を参照してください。</p>
Linux エージェント ブートストラップ サービス	<p>Linux エージェント ブートストラップのパッケージ vmware-appdirector-agent-service-ec2 を、VMware 製品のダウンロード サイト http://vmware.com/web/vmware/downloads からダウンロードします。次のいずれかのパッケージから Linux エージェント ブートストラップスクリプトをインストールします。</p> <ul style="list-style-type: none"> ■ <a href="http://<Application_Director_IP>/agent/vmware-appdirector-agent-service-ec2_6.0.0.0-0_x86_64.rpm">http://<Application_Director_IP>/agent/vmware-appdirector-agent-service-ec2_6.0.0.0-0_x86_64.rpm ■ <a href="http://<Application_Director_IP>/agent/vmware-appdirector-agent-service-ec2_6.0.0.0-0_i386.rpm">http://<Application_Director_IP>/agent/vmware-appdirector-agent-service-ec2_6.0.0.0-0_i386.rpm <p>エージェント ブートストラップサービスを RPM ベースの AMI にインストールするには、「Amazon EC2 仮想マシン テンプレートまたは AMI の作成 (P. 89)」 を参照してください。</p> <p>preparevCACTemplate.sh スクリプトを使用すると、これらのファイルがダウンロードされます。</p>
サポートされている Linux スクリプティング	Application Services では、Bash によるスクリプトをサポートしています。
Linux コマンド	<p>次の Linux コマンドが仮想マシンで使用可能になっている必要があります。</p> <ul style="list-style-type: none"> ■ wget ■ md5sum ■ grep ■ sed ■ setsid ■ awk ■ ifconfig
オプションのサービス	<p>トラブルシューティングやその他の理由で、Linux ssh ログを使用してリモートから仮想マシンにアクセスすることを計画している場合は、Linux の OpenSSH サーバおよびクライアントがインストールされ正常に動作している必要があります。</p>

重要 ブート処理は中断してはならないため、最終的なオペレーティングシステムのログイン プロンプトに到達するまで、仮想マシンのブート処理が一時停止することがないように仮想マシンを構成します。たとえば、仮想マシンが起動するときに、ユーザーの操作を求めるプロセスやスクリプトが表示されないようにします。この要件は、Application Services ライブラリのために作成された仮想マシン テンプレートにのみ適用されます。

Amazon EC2 仮想マシン テンプレートまたは AMI の作成

Application Services で Amazon AMI を使用するには、既存の AMI をカスタマイズして新しい AMI を作成する必要があります。

開始する前に

- AWS 管理コンソール、および Amazon EC2 インスタンスまたは仮想マシンを起動、構成、終了する手順を理解しておく必要があります。
- AWS 管理コンソールにアクセスするための適切な認証情報が用意されていることを確認します。
- Application Services アプリケーションの実行元になる AWS リージョンを特定し、AMI がそのリージョンにアクセスできることを確認します。
- Amazon EC2 AMI が Application Services 仮想マシンの要件を満たしていることを確認します。[「Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 \(P. 88\)」](#) を参照してください。

AMI の使用方法の詳細については、Amazon AWS のドキュメントを参照してください。

手順

- 1 AWS コンソールにログインし、アプリケーションの展開先となる適切な Amazon リージョンで EBS バックアップされた AMI を特定します。
- 2 AMI を使用して Amazon EC2 インスタンスを起動し、root ユーザーとしてログインしてターミナルを開きます。
- 3 RPM ベース AMI 用のサポートされている JRE パッケージをダウンロードおよびインストールします。
 - a Application Services と同じネットワークに存在するマシン上で、
`http://<Application_Director_IP>/agent/jre-7u45-linux-<ArchitectureName>.rpm` から JRE パッケージをダウンロードします。ここで、<ArchitectureName> は、i586 (32 ビットの場合) または amd64 (64 ビットの場合) です。
 - b JRE パッケージを Amazon EC2 インスタンスにアップロードします。
 - c Amazon EC2 インスタンスで `rpm -i jre-7u45-linux-<ArchitectureName>.rpm` と入力して、パッケージをインストールします。
- 4 RPM ベース AMI 用の Application Services エージェント ブートストラップサービスをダウンロードおよびインストールします。
 - a Application Services と同じネットワークに存在するマシン上で、
`http://<Application_Director_IP>/agent/vmware-appdirector-agent-service-ec2_5.2.0.0-<ArchitectureName>.rpm` からエージェント ブートストラップサービスをダウンロードします。ここで、<ArchitectureName> は、i386 (32 ビットの場合) または x86_64 (64 ビットの場合) です。
 - b エージェント ブートストラップサービスを Amazon EC2 インスタンスにアップロードします。
 - c Amazon EC2 インスタンスで、
`rpm -i vmware-appdirector-agent-service-ec2_5.2.0.0-<ArchitectureName>.rpm` と入力して、サービスをインストールします。
- 5 (オプション) CentOS オペレーティングシステムに 2 基目のイーサネット デバイス eth1 を追加します。
 - a `ifcfg-eth0` ファイルの場所を特定します。
 - b 同ファイルの内容を `ifcfg-eth1` ファイルにコピーします。
 - c `ifcfg-eth1` ファイルで、すべての `DEVICE=eth0` 文字列を `DEVICE=eth1` で置換し、変更内容を保存します。

2 基目のイーサネット デバイスを追加したことにより、この AMI に基づき、2 つ以上の NIC を備えた仮想マシンを展開できます。

- 6 (オプション) Linux のコマンド履歴を消去して、これから作成する AMI のセキュリティを確保します。
 - a `.bash_history` ファイルに保存された Linux コマンド履歴を削除します。


```
cat /dev/null > ~/.bash_history
```
 - b 現在の Bash ログイン セッションの Linux コマンド履歴を削除します。


```
history -c
```
- 7 AWS コンソールで、Amazon EC2 インスタンスを停止します。



注意 インスタンスを終了すると、すべての変更内容が失われます。

- 8 Amazon EBS でバックアップされた AMI を作成するには、停止した Amazon EC2 インスタンスを選択し、[イメージ名] を入力して、EBS バックアップされたイメージに変換するプロセスを起動します。

AMI の作成には数分かかる場合があります。

AMI が使用可能になったら、Amazon EC2 インスタンスを終了できます。

次に進む前に

Amazon EC2 リージョンを Application Services クラウド プロバイダにマッピングして、AMI または仮想マシン テンプレートに対するアクセス権を付与します。[\[Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)\]](#) を参照してください。

Amazon EC2 のクラウド プロバイダとテンプレートの登録

Application Services では、クラウド プロバイダを登録すると、有効な AWS アカウントに設定されたアクセス キーを使用して Amazon EC2 リージョンに接続できます。

クラウド プロバイダ情報を指定するときには、以下の Amazon リージョン コードの値がサポートされています。サポートされている Amazon リージョン コードの値の最新リストについては、Amazon AWS の リージョンとエンドポイント ドキュメントを参照してください。たとえば、http://docs.aws.amazon.com/general/latest/gr/rande.html#ec2_region を参照してください。

表 11-2. 共通してサポートされている Amazon リージョン コードの値

Amazon リージョン コードの値	地域名
ap-northeast-1	アジア パシフィック (東京) リージョン
ap-southeast-1	アジア パシフィック (シンガポール) リージョン
ap-southeast-2	アジア パシフィック (シドニー) リージョン
eu-west-1	欧州 (アイルランド) リージョン
sa-east-1	南米 (サンパウロ) リージョン
us-east-1	米国東部 (バージニア北部) リージョン
us-west-1	米国西部 (北カリフォルニア) リージョン
us-west-2	米国西部 (オレゴン) リージョン

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- AWS ユーザー アカウントにアクセス キーが設定されていることを確認します。
- 仮想プライベート クラウド (VPC) の要件を満たす仮想マシン テンプレートを作成します。[\[Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 \(P. 88\)\]](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [クラウド プロバイダ] を選択します。
- 2 [新規] をクリックします。
- 3 クラウド プロバイダ情報を作成します。

オプション	説明
名前と説明	詳細な情報を [説明] テキスト ボックスに入力します。[展開プロファイル] ウィザードの [クラウド プロバイダ] 列に、テキスト ボックスのテキストが表示されます。
クラウド プロバイダのタイプ	Amazon EC2 はサポート対象のタイプです。
Amazon リージョン	Amazon リージョン コードの値を指定します。たとえば、リージョンが米国東部 (バージニア北部) の場合、Amazon リージョンは us-east-1 になります。 サポートされていない値を入力すると、新しい展開環境を作成しようとしたときにアベイラビリティ トンネルにコンテンツが表示されなくなります。
プロキシ URL	企業ネットワークで外部接続のためのプロキシ サーバが必要になった場合は、Web プロキシ URL を追加して、仮想プライベート クラウド (VPC) サーバが Amazon EC2 に接続できるようにします。
ユーザー名とパスワード	AWS アカウントで設定されたアクセス キー。

- 4 入力した値が正しいかどうかをテストするには、[接続の検証] をクリックします。
- 5 [保存] をクリックします。
クラウド プロバイダに必要なエントリが無効または空白の場合は、[保存] をクリックしたときに、修正するように求められます。
- 6 Amazon AMI を登録するには、[編集] をクリックします。
- 7 [テンプレート] セクションで、[新規] をクリックして、AMI または仮想マシンテンプレートの登録を続けます。
- 8 ドロップダウン メニューからカテゴリを選択し、[テンプレートの取得] をクリックします。
AWS アカウントが所有している AMI、または AWS アカウント専用の AMI を選択できます。
ユーザーが作成したか AWS アカウント専用の、フィルタリングされた AMI またはクラウド テンプレートが、ドロップダウン メニューに表示されます。
- 9 リスト内のテンプレートの横にあるチェック ボックスをオンにし、[OK] をクリックします。
登録したテンプレートは、論理テンプレートをクラウド テンプレートにマッピングする際に選択できるクラウド テンプレートのリストに追加されます。
- 10 Amazon EC2 クラウド プロバイダの既存の情報を変更するには、[編集] をクリックして変更を行います。
- 11 [保存] をクリックします。

Amazon EC2 リージョンが登録され、その AMI を Application Services で使用できるようになります。

次に進む前に

展開環境を作成し、アプリケーションをクラウド環境に展開します。[「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。

これらの AMI を Application Services 論理テンプレートにマップします。[「ライブラリへの論理テンプレートの追加 \(P. 125\)」](#) を参照してください。

Amazon EC2 の展開環境の作成

展開環境を Amazon EC2 VPC および関連付けられた可用性ゾーンにマッピングして、アプリケーションをクラウドに展開できるようにします。

VPC を使用すると、アプリケーションを Application Services から展開できる、論理的に隔離された AWS クラウド領域をプロビジョニングできます。

展開環境が現在展開プロファイルまたは展開で使用されている場合は、削除できません。

開始する前に

- アプリケーション クラウド管理者として Application Services にログインします。
- Amazon EC2 環境が、Application Services アプライアンスとセキュアな接続を確立するように構成されていることを確認します。[「Application Services の Amazon EC2 環境を構成する \(P. 84\)」](#) を参照してください。
- 少なくとも 1 社の Amazon クラウド プロバイダが Application Services に登録されていることを確認します。[「Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)」](#) を参照してください。
- セキュアなクラウド トンネル接続が使用できることを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [展開環境] を選択します。
- 2 [新規] をクリックします。
- 3 展開環境の情報を入力します。

オプション	説明
展開環境と説明	VPC の名前と可用性ゾーン名をこれらのフィールドのいずれかに含めます。これらのフィールドのテキストは [展開プロファイル] ウィザードの [展開環境] 列下に表示されます。
クラウド プロバイダ	クラウド プロバイダがリストに表示されない場合、ダイアログ ボックスをキャンセルして [クラウド] - [クラウド プロバイダ] を選択し、Amazon EC2 クラウド プロバイダを追加します。
仮想プライベート クラウド	[選択] をクリックし、リージョンで使用可能な Amazon VPC のリストを表示します。
可用性ゾーン	[選択] をクリックし、Amazon リージョンでの可用性ゾーンのリストを表示します。
クラウド トンネル	Application Services サーバを Endpoint 仮想マシンに接続するクラウド トンネル インスタンスを選択します。この Endpoint 仮想マシンは、[仮想プライベート クラウド] セクションでこの展開環境用にすでに設定されている Amazon EC2 VPC にある必要があります。[検証] をクリックして、クラウド トンネルのセキュアな SSH 接続ステータスを確認します。

- 4 [保存] をクリックします。
- 5 Amazon EC2 展開環境の既存の情報を変更するには、[編集] をクリックして変更を行います。
- 6 [保存] をクリックします。

展開環境がリストに追加され、展開プロファイルを作成するときに選択できるようになります。

次に進む前に

外部サービス インスタンスのマッピング、ポリシー インスタンスの作成、または展開環境へのソリューション インスタンスの登録を実行できます。[「外部サービス インスタンスのマッピング \(P. 61\)」](#)、[「ポリシー インスタンスの作成 \(P. 62\)」](#) または [「Puppet ソリューション インスタンスの作成 \(P. 65\)」](#) を参照してください。

Application Services コンポーネントの開発

12

Application Services は、仮想マシンにインストール可能なコンポーネントを作成および開発するためのオープンなフレームワークを提供します。

ライブラリに登録されている Application Services コンポーネントとして、JBoss や MySQL などの事前定義済みサンプル サービス、APT リポジトリを構成するスクリプトなどの事前定義済みタスク、WAR や SQL_SCRIPT などのブループリント アプリケーション コンポーネントなどがあります。

ここでのコンポーネントには、サービス、アプリケーション コンポーネント、カスタム タスクが含まれます。これらのコンポーネントはすべて、その共通基盤フレームワークとしてアクションとプロパティを使用します。

注意 Application Services ライブラリのサンプル コンポーネントは、テスト環境のみで使用してください。

展開可能なコンポーネントの開発に関するトピックに頻出する主要概念について理解しておく必要があります。[「主要な概念 \(P. 16\)」](#) を参照してください。

この章では次のトピックについて説明します。

- [コンポーネント アクションの定義 \(P. 93\)](#)
- [コンポーネント プロパティの構成 \(P. 95\)](#)
- [Application Services の事前定義済みプロパティ \(P. 103\)](#)
- [セキュアなコンポーネント プロパティ \(P. 104\)](#)
- [必要なコンポーネントのプロパティ \(P. 105\)](#)
- [コンポーネントの開発のベスト プラクティス \(P. 105\)](#)

コンポーネント アクションの定義

各コンポーネントには、サービスまたはアプリケーション コンポーネントのインストール、構成、開始、更新、ロールバック、および分解を行う各スクリプトのための事前定義されたライフ サイクル ステージまたはアクションが含まれます。

アプリケーション カタログ管理者は、INSTALL、CONFIGURE、START、UPDATE、ROLLBACK、または TEARDOWN のライフ サイクル ステージの少なくとも 1 つに対して、Bash、Windows CMD、PowerShell、または BeanShell のスクリプトを提供する必要があります。これらのスクリプトは、コンポーネントのプロパティを使用するためにカスタマイズされます。

たとえば、仮想マシンに Apache Tomcat サーバを展開する場合は、次のスクリプトを追加できます。

INSTALL	Tomcat サーバのインストール ビットをダウンロードし、Tomcat サービスをインストールします。
CONFIGURE	JAVA_OPTS、CATALINA_OPTS、およびその他の必要なすべての構成を設定します。
START	Tomcat サーバで開始コマンドを使用して、Tomcat サービスを開始します。

更新

更新スクリプトを使用して Tomcat サービスの構成を変更するか、またはクラスタ サイズを変更して展開されているアプリケーションをスケーリングし、ロード バランサーを使用してクラスタ化されたノードを管理します。

ROLLBACK

Tomcat サービスの更新処理がパフォーマンスまたはセキュリティの問題によって失敗した場合、または更新処理は成功しても、展開しているアプリケーションにエラーがある場合は、ロールバック スクリプトを変更します。

TEARDOWN

Tomcat サービスの分解スクリプトを変更します。たとえば、展開が分解される前にアプリケーションで特定のアクションを実行します。

アプリケーション アーキテクトは、たとえば、インストーラの場所、インストール パス、および Tomcat の開始コマンドをスクリプトのプロパティとして宣言することによって、スクリプトをパラメータ化できます。パラメータは、スクリプトを汎用にレンダリングします。これらの汎用スクリプトは、変更することなく異なる環境にサービスを展開できます。

パラメータ値は、アクション スクリプトから変更することもできます。これらの変更されたプロパティは、他のコンポーネントに対するプロパティ値として参照できます。『Application Services ライブラリ サービスの使用』のガイドを参照してください。

アクションに対して定義されるスクリプトは、`/tmp/` ディレクトリで実行されます。Linux スクリプトは、`/tmp/<runId>/<ComponentName>-<LifecycleStageName>` にあります。`<runId>` は、各展開の一意的ジョブ識別子で、[展開の概要] ページの [タスク詳細] ステータス ウィンドウにあります。Windows スクリプトは、`\Users\darwin\AppData\Local\Temp` にあります。

注意 アクション スクリプトを実行する際に、ユーザーの操作を求めるメッセージを表示するプロセスがないことを確認します。中断によってスクリプトが一時停止し、無期限にアイドル状態になる原因になり、最終的に失敗します。また、終了ステータスがゼロ以外の状態で Windows CMD スクリプトが終了すると、Application Services エージェントは展開を停止して、[失敗した展開] のマークを付けます。**exit /b 0** を使用して成功状態を示し、**exit /b non-zero** を使用してエラー状態を示します。

[「サポートされているアクション スクリプトとカスタム タスク スクリプト \(P. 94\)」](#) を参照してください。

アプリケーション展開にサービス タスクまたはカスタム タスクを追加し、コンポーネント アクションを定義できます。[「ライブラリへのサービスの追加 \(P. 109\)」](#) および [「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。

サポートされているアクション スクリプトとカスタム タスク スクリプト

Application Services は、Linux ベースのアプリケーションで Bash または BeanShell のスクリプト記述をサポートし、Windows ベースのアプリケーションでは Windows CMD、PowerShell、または BeanShell のスクリプト記述をサポートします。

ライフ サイクル ステージに適しているアクション スクリプトのタイプを選択するには、ブループリント キャンパスで [スクリプト タイプ] 列をダブルクリックし、ドロップダウン メニューからスクリプトを選択します。カスタム タスク スクリプトに 1 つ以上のオペレーティング システムを選択します。ドロップダウン メニューからサポートされているスクリプトを設定できます。

選択するスクリプトのタイプによっては、コードを記述し、スクリプトの変数を介して関連するプロパティにアクセスできます。ライフ サイクル ステージのアクション スクリプトでは、同じオペレーティング システム ファミリーで、同じコンポーネントの各ライフ サイクルに異なるスクリプト タイプを使用できます。たとえば、INSTALL ステージに Windows CMD スクリプトを使用し、CONFIGURE ステージに PowerShell スクリプトを使用できます。[「プロパティのタイプ \(P. 95\)」](#) を参照してください。

アクション スクリプトを記述する際は、スクリプトのタイプによって終了コードと戻りコードが異なります。アプリケーション アーキテクトは、アプリケーションの展開に適用できる適切なスクリプトの終了コードを設定する必要があります。スクリプトに終了コードや戻りコードがない場合、スクリプトで実行される最後のコマンドが終了ステータスになります。[「展開および更新プロセスについて \(P. 175\)」](#) を参照してください。

表 12-1. アクション スクリプトの終了コードと戻りコード

スクリプト タイプ	説明
Bash	アクション スクリプトでは、 return 0 コードまたは exit 0 コードを使用して成功状態を示すことができます。エラー状態を示すには、 return non-zero または exit non-zero を使用できます。
Windows CMD	アクション スクリプトで、 exit 0 コードと exit non-zero コードは使用しないでください。スクリプトでこれらのコードを使用すると、計算されたプロパティのタスク処理が中断します。 exit /b 0 を使用して成功状態を示し、 exit /b non-zero を使用してエラー状態を示します。
Windows PowerShell	exit 0 を使用して成功状態を示し、 exit non-zero を使用してエラー状態を示します。
BeanShell	System.exit(0); を使用して成功状態を示し、 System.exit(1);/non-zero を使用してエラー状態を示します。

Windows PowerShell を使用してスクリプトを記述する場合、アクション スクリプトで **warning**、**verbose**、**debug**、および **host** の各コールは使用できません。

コンポーネント プロパティの構成

コンポーネント プロパティを使用して、スクリプトをパラメータ化できます。これにより、Application Services は、定義されたプロパティを環境変数として、仮想マシン上で実行中のスクリプトに渡すことができます。

仮想マシンの Application Services エージェントは、ライフサイクル ステージのスクリプトを実行する前に、Application Services サーバとやり取りしてプロパティを解決します。その後、それらのプロパティからスクリプト固有の変数を作成し、スクリプトに渡します。

- [プロパティのタイプ \(P. 95\)](#)

Application Services では、文字列、アレイ、コンテンツ、計算値、ブール値、単一選択、整数値、二重浮動小数点のプロパティがサポートされます。

- [プロパティ値の定義 \(P. 100\)](#)

アプリケーション カタログ管理者は、サービスおよびカスタム タスクのライブラリにプロパティを定義できます。

- [ほかのプロパティへのバインド \(P. 100\)](#)

いくつかの展開シナリオでは、コンポーネントは、自身をカスタマイズするために、別のコンポーネントのプロパティ値を必要とします。Application Services では、このプロセスのことを、別のプロパティへのバインドと呼んでいます。

- [ほかのプロパティへの自動バインド \(P. 101\)](#)

サービス内に設定された自動バインド メタ データと、ノード、ノードアレイ、サービス、および WAR アプリケーション コンポーネント内のデフォルトのメタ データを使用することで、他のプロパティに自動バインドすることができます。

- [ライフ サイクル ステージとコンポーネント間のプロパティ値の受け渡し \(P. 103\)](#)

各コンポーネントに定義されたプロパティは、各ライフ サイクル ステージのアクション スクリプトで使用できます。

プロパティのタイプ

Application Services では、文字列、アレイ、コンテンツ、計算値、ブール値、単一選択、整数値、二重浮動小数点のプロパティがサポートされます。

注意 プロパティの名前には、英数字、ハイフン (-)、またはアンダースコア (_) のみを使用できます。大文字と小文字は区別されます。

文字列プロパティ

文字列プロパティの値は、文字列、または別の文字列プロパティにバインドされた値です。文字列値には、どの ASCII 文字でも含めることができます。バインド プロパティの場合は、ブループリント キャンバスの [プロパティ] タブを使用して、バインドする適切なプロパティを選択します。これにより、プロパティ値が Raw 文字列データとしてアクション スクリプトに渡されます。

文字列プロパティの例	スクリプト構文	使用例
admin_email = "admin@email987.com"	Bash - \$admin_email	echo \$admin_email
	Windows CMD - %admin_email%	echo %admin_email%
	Windows PowerShell - \$admin_email	write-output \$admin_email
	BeanShell - admin_email	print(admin_email);

アレイ プロパティ

アレイ プロパティの値は、<["value1", "value2", "value3"...]> という形式で定義された文字列のアレイ、または別のアレイ プロパティにバインドされた値です。アレイ プロパティの値を定義する場合は、文字列のアレイを角括弧で囲む必要があります。文字列のアレイの場合、アレイ要素の値には、任意の ASCII 文字を含めることができます。アレイ プロパティ値に含まれるバックスラッシュ文字を正しくエンコードするには、1 つ余分にバックスラッシュを追加します（例：["c:\<test1>\<test2>"]）。バインド プロパティの場合は、ブループリント キャンバスの [プロパティ] タブを使用して、バインドする適切なプロパティを選択します。

たとえば、アプリケーション サーバ仮想マシンのクラスタの負荷を分散しているロード バランサ仮想マシンについて考えます。このような場合、ロード バランサ サービスのアレイ プロパティを定義して、そのアレイ プロパティに各アプリケーション サーバ仮想マシンの IP アドレスのアレイを設定します。

次のロード バランサ サービス構成スクリプトでは、アレイ プロパティを使用して、Red Hat、Windows、および Ubuntu の各オペレーティング システム間で適切なロード バランシング機能を構成しています。

アレイ プロパティの例	スクリプト構文	使用例
operating_systems = ["Red Hat", "Windows", "Ubuntu"]	Bash - \${operating_systems[@]} (文字列のアレイ全体の場合) \${operating_systems[N]} (個々のアレイ要素の場合)	for ((i = 0 ; i < \$ #operating_systems[@]; i++)); do echo \${operating_systems[i]} done
	Windows CMD - %operating_systems_<N>% ここで、<N> はアレイ内の要素の位置	for /F "delims== tokens=2" %A in ('set operating_systems_') do (echo %A)
	Windows PowerShell - \$operating_systems (文字列のアレイ全体の場合) \$operating_systems[N] (個々のアレイ要素の場合)	foreach (\$os in \$operating_systems) { write-output \$os }
	BeanShell - operating_systems[N] ここで、<N> はアレイ内の要素の位置	for(index=0;index < operating_systems.length; index++) { print(operating_systems[index]); }

コンテンツ プロパティ

コンテンツ プロパティの値は、コンテンツをダウンロードするためのファイルへの URL です。Application Services エージェントは、URL から仮想マシンにコンテンツをダウンロードして、仮想マシン内のローカル ファイルの場所をスクリプトに渡します。

コンテンツ プロパティは、HTTP または HTTPS プロトコルを使用した有効な URL として定義する必要があります。たとえば、Hyperic 製品が Application Services アプライアンスでホストされていて、URL がアプライアンスのその場所を参照しているとします。Application Services エージェントは、この製品を指定された場所から展開先の仮想マシンにダウンロードします。

文字列プロパティの例	スクリプト構文	使用例
HQ_PACKAGE = "http://<DarwinServerIP>/artifacts/services/hyperic/hyperic-hq-agent-linux.tar.gz"	Bash - \$HQ_PACKAGE	tar -zxvf \$HQ_PACKAGE
	Windows CMD - %HQ_PACKAGE %	start /wait c:\unzip.exe %HQ_PACKAGE%

文字列プロパティの例	スクリプト構文	使用例
	Windows PowerShell - \$HQ_PACKAGE	& c:\unzip.exe \$HQ_PACKAGE
	BeanShell - HQ_PACKAGE	<pre> import java.io.BufferedOutputStream; import java.io.File; import java.io.FileInputStream; import java.io.FileOutputStream; import java.util.zip.ZipEntry; import java.util.zip.ZipInputStream; destDir = new File(bsh.cwd); if (! destDir.exists()) { destDir.mkdir(); } zipIn = new ZipInputStream(new FileInputStream(HQ_PACKAGE)); entry = zipIn.getNextEntry(); // iterates over entries in the zip file while (entry != null) { String filePath = bsh.cwd + File.separator + entry.getName(); if (! entry.isDirectory()) { // if the entry is a file, extracts it bos = new BufferedOutputStream(new FileOutputStream(filePath)); bytesIn = new byte[4096]; read = 0; while ((read = zipIn.read(bytesIn)) != -1) { bos.write(bytesIn, 0, read); } bos.close(); } else { // if the entry is a directory, make the directory dir = new File(filePath); dir.mkdir(); } } zipIn.closeEntry(); </pre>

文字列プロパティの例	スクリプト構文	使用例
		<pre>entry = zipIn.getNextEntry(); } zipIn.close();</pre>

計算値プロパティ

計算値プロパティ タイプでは、定義時に初期値を設定できません。その代わりに、計算値プロパティ タイプが、INSTALL、CONFIGURE、または START ライフサイクル スクリプトから値を取得します。割り当てられた値は、以降の使用可能なライフサイクルと依存コンポーネントに伝播されます。

注意 アクションスクリプトで定義された計算値プロパティを、UPDATE、ROLLBACK、TEARDOWN ライフサイクル スクリプトで使用することはできません。計算値プロパティの値は、これらの各ライフサイクル スクリプトで設定する必要があります。

文字列プロパティの例	スクリプト構文	使用例
my_unique_id = ""	Bash - \$my_unique_id	export my_unique_id="0123456789"
	Windows CMD - %my_unique_id%	set my_unique_id=0123456789
	Windows PowerShell - \$my_unique_id	\$my_unique_id = "0123456789"
	BeanShell - my_unique_id	my_unique_id = "0123456789";

ブール値のプロパティ

ブール値のプロパティ タイプは、[値] ドロップダウン メニューで [はい] と [いいえ] の選択肢を提供しますが、ページの別の部分をクリックすると、選択する値は True または False に変わります。Application Services は、混乱を避け、vRealize Automation コンソール インターフェイスとの整合性を維持するために [はい] と [いいえ] の値を表示しますが、実行スクリプトでの使用では [はい] を True に、[いいえ] を False に変更します。

単一選択のプロパティ

単一選択のプロパティでは、ユーザーが値を 1 つだけ選択できる複数の文字列の値を指定できます。

整数値のプロパティ

整数値のプロパティ タイプでは、ゼロ、正の整数、または負の整数が値として許可されます。

二重浮動小数点のプロパティ

二重浮動小数点のプロパティは、4 桁までの精度で浮動小数点の値を保存します。

プロパティ タイプ リファレンス

プロパティ タイプ リファレンスには、構成時に参照できるプロパティ タイプが表示されます。

プロパティ タイプ	バインドするプロパティ タイプ	自動バインドの公開	自動バインドの利用	初期プロパティ値の承認
文字列	文字列、アレイ、コンテンツ、計算値	はい	はい	はい
コンテンツ	該当なし	該当なし	該当なし	はい
アレイ	文字列、コンテンツ、計算値	該当なし	はい	はい
計算値	該当なし	はい	該当なし	該当なし

アレイ プロパティは、クラスタ内の事前定義済みノード アレイ プロパティ all にバインドすることもできます。all プロパティは、正規表現で、クラスタ化されたノードの特定のプロパティが取るすべての値を収集する方法のことで、

プロパティのバインドの詳細については、「[ほかのプロパティへのバインド \(P. 100\)](#)」を参照してください。自動バインドの公開と自動バインドの利用の概念を理解するには、「[ほかのプロパティへの自動バインド \(P. 101\)](#)」を参照してください。

文字列プロパティの値がさまざまなタイプのプロパティにバインドする例を以下に示します。

プロパティ タイプの例	バインドするプロパティ タイプ	バインド結果 (A が B にバインド)
文字列 (プロパティ A)	文字列 (プロパティ B="Hi")	A="Hi"
文字列 (プロパティ A)	コンテンツ (プロパティ B="http://my.com/content")	A="http://my.com/content"
文字列 (プロパティ A)	アレイ (プロパティ B=["1","2"])	A=["1","2"]
文字列 (プロパティ A)	計算値 (プロパティ B="Hello")	A="Hello"

アレイプロパティの値がさまざまなタイプのプロパティにバインドする例を以下に示します。

プロパティ タイプの例	バインドするプロパティ タイプ	バインド結果 (A が B にバインド)
アレイ (プロパティ A)	文字列 (プロパティ B="Hi")	A="Hi"
アレイ (プロパティ A)	コンテンツ (プロパティ B="http://my.com/content")	A="http://my.com/content"
アレイ (プロパティ A)	計算値 (プロパティ B="Hello")	A="Hello"

プロパティ値の定義

アプリケーション カタログ管理者は、サービスおよびカスタム タスクのライブラリにプロパティを定義できます。

ライブラリに定義されたプロパティのデフォルト値を、サービスが使用される特定のアプリケーションのニーズに合わせてブループリント内でオーバーライドすることで、コンポーネント スクリプトをカスタマイズできます。このプロパティ値を展開プロファイル内でさらにオーバーライドして、サービスが展開される環境に合わせることもできます。

注意 プロパティ値では大文字と小文字が区別されます。新しいプロパティ値を指定しても、入力した値が間違っているとそのプロパティは無効になります。

たとえば、Apache Tomcat サーバを展開する前準備として、アプリケーション カタログ管理者がサービスの JVM HEAP_SIZE を 512MB に設定しているとします。このとき、アプリケーション アーキテクトは、大規模アプリケーションに対応できるように、この値を 1024MB に再定義できます。さらに、展開担当者は、アプリケーションを本番環境に展開する際に、大きな負荷にも対処できるように、この値を 2048MB でオーバーライドできます。

プロパティ値は、デフォルトではオーバーライドできませんが、アプリケーション カタログ管理者はオーバーライド可能な機能を有効にできます。アプリケーション アーキテクトは、オーバーライド可能なプロパティに対して、一部の規制された環境でそのプロパティの値を強制し、展開者に対してオーバーライド可能な機能を無効にできます。

プロパティ値は、アプリケーション展開にサービスまたはカスタム タスクを追加するときに定義できます。[「ライブラリへのサービスの追加 \(P. 109\)」](#) および [「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。

ほかのプロパティへのバインド

いくつかの展開シナリオでは、コンポーネントは、自身をカスタマイズするために、別のコンポーネントのプロパティ値を必要とします。Application Services では、このプロセスのことを、別のプロパティへのバインドと呼んでいます。

アプリケーション カタログ管理者は、スクリプト内のプロパティ定義を変更できます。たとえば WAR コンポーネントの場合、Apache Tomcat サーバのインストール場所を必要とすることがあります。WAR コンポーネントでは、server_home プロパティ値を Apache Tomcat サーバの install_path プロパティ値に設定できます。

コンポーネントの Bash スクリプトは、独自のプロパティのみを使用できます。Application Services では、プロパティをハード コードされた値に設定する以外に、プロパティをブループリント内の別のプロパティにバインドできます。別のプロパティにバインドする場合、別のコンポーネントのプロパティ値と仮想マシンのプロパティ値 (IP アドレスなど) に基づいて、スクリプトをカスタマイズできます。プロパティを別のプロパティにバインドするには、[プロパティの編集] ダイアログ ボックスの [ブループリント値] ドロップダウン メニューからプロパティ値を選択します。

単一の仮想マシン ノードの場合は、[ブループリント値] ドロップダウン メニュー内の参照されるプロパティは、`<NodeName>:<ComponentName>:<PropertyName>` です。

クラスタ化された仮想マシン ノードの場合は、[ブループリント値] ドロップダウン メニュー内の参照されるプロパティは、`all(<NodeName>:<ComponentName>:<PropertyName>)` です。別のプロパティがこのクラスタ プロパティを参照するとき、クラスタ内のすべての仮想マシンから `<PropertyName>` プロパティ値が取得されます。クラスタ化された仮想マシンの事前定義済みの `all(<NodeName>:node_array_index)` プロパティでは、クラスタ内のノードアレイ インデックスのコレクションが取得されます。[「事前定義されたノードアレイ インデックス プロパティ \(P. 104\)」](#) を参照してください。

クラスタ ノードおよび単一ノードの場合、`self:<ComponentName>:<PropertyName>` 値を使用して、ターゲット コンポーネントが実行されている仮想マシンのコンポーネント プロパティを示します。たとえば、WAR コンポーネントが Apache Tomcat サーバに展開される場合、現在の仮想マシンで実行されている Apache Tomcat サーバを参照するために、`WAR server_home` プロパティを `self:tomcat:install_path` に設定できます。

IP アドレスのようなノード レベルのプロパティは、`<NodeName>:ip` または `self:ip` と表示されます。このようなプロパティは仮想マシンに排他的に属し、仮想マシン内の特定のコンポーネントには所属しません。



複数の NIC を持つ仮想マシンの IP アドレスの場合、Application Services は `<NodeName>:NIC<x>_ip` または `self:NIC<x>_ip` プロパティのいずれかを指定します。ここで、`<x>` には NIC 番号が反映されます。[「事前定義済みの IP アドレス プロパティ \(P. 103\)」](#) を参照してください。

図 12-1. クラスタ化されたノード内の別のプロパティへのバインド

ほかのプロパティへの自動バインド

サービス内に設定された 自動バインド メタ データと、ノード、ノード アレイ、サービス、および WAR アプリケーション コンポーネント内のデフォルトのメタ データを使用することで、他のプロパティに自動バインドすることができます。

自動バインド タイプは、アプリケーション カタログ管理者によってサービス内で定義された 自動バインド メタ データに基づき、他のプロパティがブループリント内で自身とバインドできるように、[消費] または [公開] として表示されます。

ブループリント キャンパス内では、[公開] アイコン  は、[自動バインドの公開] プロパティの横に表示され、[消費] アイコン  は、[自動バインドの消費] プロパティの横に表示されます。アイコンのヒントには、自動バインド タイプと自動バインド タグ情報が表示されます。たとえば、`webapps_dir` プロパティの横の [消費] アイコンのヒントには、「自動バインドの消費：サーブレット コンテナ、DIR 情報」が表示されます。アプリケーション カタログ管理者は、サービスの作成または既存サービスの編集時に、サービスに自動バインド のタイプとタグのパラメータを設定します。[「ライブラリへのサービスの追加 \(P. 109\)」](#) を参照してください。

自動バインドのための事前構成済みコンポーネント

WAR は、自動バインドのために事前構成された唯一のアプリケーション コンポーネントです。事前定義済みの自動バインド タイプとタグのパラメータは編集できません。

アプリケーション コンポーネントをサービスに追加すると、プロパティはどちらの側でも設定できます。この場合、自動バインドは 2 回行われます。アプリケーション コンポーネントのソース プロパティが、サービスおよび親ノードまたはノード アレイのターゲット プロパティから設定されます。次に、サービスのソース プロパティが、アプリケーション コンポーネントのターゲット プロパティから設定されます。

表 12-2. 事前構成済みの WAR アプリケーション コンポーネント

アプリケーション コンポーネント	プロパティ	自動バインド タイプ	自動バインド タグ
Linux WAR ファイル	Context	公開	Context
Linux WAR ファイル	service_start	利用	Servlet Container、Start
Linux WAR ファイル	service_stop	利用	Servlet Container、Stop
Linux WAR ファイル	webapps_dir	利用	Servlet Container、DIR

一部のノードおよびノード アレイには、プロパティに対して自動バインド タイプとタグのパラメータが事前定義されています。これらのパラメータを編集することはできません。サービスまたはアプリケーション コンポーネントがノードまたはノード アレイに追加されると、ソース プロパティがサービスまたはアプリケーション コンポーネントから取得され、ターゲット プロパティがノードまたはノード アレイから取得されます。

表 12-3. 事前定義済みのノードおよびノード アレイのプロパティ

ノード タイプ	プロパティ	自動バインド タイプ	自動バインド タグ
ノード、ノード アレイ	ホスト名	公開	ホスト名
ノード、ノード アレイ	ip	公開	IP アドレス
ノード アレイ	node-array-index	公開	Node Array Index

依存関係が存在する場合、ソース プロパティはリンク先のコンポーネントから取得されます。ターゲット プロパティはリンク元のコンポーネント、および親サービスまたはノードのコンポーネントから取得されます。

プロパティの互換性

自動バインドは、ソースおよびターゲットのプロパティ タイプとノード タイプの互換性によって決まります。

表 12-4. プロパティ タイプとノード タイプの互換性

ソース プロパティ タイプ	ターゲット プロパティ タイプ	ノード タイプ	互換性あり
アレイ	文字列	ノード	はい
アレイ	計算値	ノード	はい
アレイ	文字列	ノード アレイ	はい
アレイ	計算値	ノード アレイ	はい
文字列	文字列	ノード	はい
文字列	計算値	ノード	はい
文字列	文字列	ノード アレイ	不可
文字列	計算値	ノード アレイ	不可

ライフ サイクル ステージとコンポーネント間のプロパティ値の受け渡し

各コンポーネントに定義されたプロパティは、各ライフ サイクル ステージのアクション スクリプトで使用できます。

算出されたプロパティの場合、プロパティの値を変更し、その値をアクション スクリプトの次のライフ サイクル ステージに渡すことができます。たとえば、コンポーネント A にステージング済みと定義された `progress_status` 値がある場合、INSTALL および CONFIGURE ライフ サイクル ステージでは、それぞれのアクション スクリプトでその値を `progress_status=installed` に変更します。コンポーネント B がコンポーネント A にバインドされている場合、アクション スクリプトのライフ サイクル ステージにおける `progress_status` のプロパティ値は、コンポーネント A と同じになります。

ブループリントで、コンポーネント B が A に依存するように定義します。この依存関係によって、コンポーネントが同じノード内にあっても異なるノード間にあっても、コンポーネント間の正しいプロパティ値の受け渡しが定義されます。

たとえば、サポートされているスクリプトを使用して、アクション スクリプト内のプロパティ値を更新できます。

- `Bash progress_status="completed"`
- `Windows CMD set progress_status=completed`
- `Windows PowerShell $progress_status="completed"`
- `BeanShell progress_status="completed"`

注意 アレイおよびコンテンツのプロパティでは、ライフ サイクル ステージのアクション スクリプトとコンポーネント間での、変更されたプロパティ値の受け渡しはサポートされていません。

Application Services の事前定義済みプロパティ

Application Services では、一般に使用されているプロパティが事前定義済みプロパティとして提供されています。これらのプロパティは、クラスタ化されたノードで使用できます。

通常の場合、ほとんどのサービスは、そのサービスを実行している仮想マシンの IP アドレスを必要とします。クラスタ化されたノードの場合、サービスはクラスタ内のすべての仮想マシンの IP アドレスを必要とすることがあります。したがって、仮想マシンの IP アドレスは事前定義済みプロパティとして指定されます。

事前定義済みの IP アドレス プロパティ

ノードは複数の NIC を保有でき、各 NIC には展開されている仮想マシンの 1 つの IP アドレスが割り当てられています。

次の IP アドレスは `<NodeName>:NIC<x>_ip` プロパティで使用可能になっています。<x> は NIC 番号です。

Clustered Dukes Bank サンプル アプリケーションでは、ロード バランサのノードに 2 つの NIC が定義されているので、ロード バランサのノードのプロパティは `load_balancer:NIC0_ip` と `load_balancer:NIC1_ip` と表示されています。NIC0 と NIC1 が仮想マシンでそれぞれ `eth0` と `eth1` に割り当てられる保証はありません。NIC は、ネットワーク インターフェイスのブループリントの論理名です。これらの NIC は論理ネットワークにマッピングされ、それらは特定のクラウド ネットワークにマッピングされます。NIC0_ip プロパティは、仮想マシンの `eth0` の IP アドレスではなく、ブループリントの定義に基づいて仮想マシンに割り当てられている IP アドレスを返します。

`load_balancer:ip` プロパティも提供されます。このプロパティは、最初の NIC と NIC0 の IP アドレスを参照します。コンポーネントがクラスタ内のすべての仮想マシンの IP アドレスを必要とする場合は、`all(<NodeName>:NIC<x>_ip)` プロパティを使用します。現在の仮想マシンの IP アドレスを参照する場合は、`self:ip` プロパティを使用します。コンポーネントは、クラスタ内のすべての仮想マシンの IP アドレスではなく、自身が属している仮想マシンを知る必要がある場合があるので、このプロパティはクラスタ化されたノードに役立ちます。クラスタ内の特定の仮想マシンのプロパティを取得する場合は、`all(<NodeName>:node_array_index)` プロパティを使用できます。[「事前定義されたノード アレイ インデックス プロパティ \(P. 104\)」](#) を参照してください。

事前定義されたノード アレイ インデックス プロパティ

クラスタ化ノードでは、`node_array_index` という特殊なプロパティによって、クラスタ内での現在の仮想マシンの位置が識別されます。

クラスタ内の 1 番目の仮想マシンの `node_array_index` プロパティ値は 0、2 番目の仮想マシンの `node_array_index` プロパティ値は 1、という具合に続きます。

たとえば、クラスタ化ノードへの展開で、1 番目の仮想マシンがマスター仮想マシンであれば、そのマスター仮想マシンがクラスタ内のほかの仮想マシンを従属マシンとして管理します。このマスター仮想マシンは異なる方法で構成する必要があります。たとえば、コンポーネントのプロパティの 1 つに `myPosition` がある場合、それを事前定義済みプロパティ `self.node_array_index` にバインドする必要があります。コンポーネント スクリプトは、`myPosition` のプロパティ値が 0 かどうかを確認し、0 であれば、以降の構成を実行できます。

クラスタ化ノードの場合、事前定義されたノード アレイ プロパティは `all(<NodeName>.node_array_index)` です。このプロパティは、クラスタ内のすべての仮想マシンのインデックスを表し、クラスタのサイズを特定するのに使用できます。

事前定義済みのディスク レイアウト情報プロパティ

フレキシブル ディスクによりストレージの柔軟性が高まり、追加ディスクをノードに追加できるようになります。

追加のディスク情報を使用するには、サービス定義でバインド可能なプロパティを作成し、そのプロパティを `disk_layout_info` プロパティにバインドする必要があります。`disk_layout_info` プロパティには、アプリケーション ブループリントに追加されるすべてのディスク情報が含まれています。

たとえばアプリケーション ブループリントでは、ディスク情報がブループリントにより `Disk0`、`Disk1`、`Disk2` などとしてキャプチャされます。

次の情報を取得できます。

- 名前
- `sizeInGb`
- `metaTags`
- `seq`
- `mountPoint`
- `fileSystem`

セキュアなコンポーネント プロパティ

プロパティは展開したコンポーネントを構成するために使用されます。場合によって、パスワードなどの機密性の高いデータを格納するために使用されることがあります。

たとえば、WAR コンポーネントはデータベースにアクセスするためにパスワードなどの機密性の高いデータを格納する必要がある場合があります。これらのプロパティはセキュアとしてマークできます。セキュアなプロパティの値はマスクされ、Application Services でアスタリスクとして表示されます。

プロパティがセキュアから非セキュアに変更されると、Application Services はセキュリティ保護のため、プロパティ値をリセットします。プロパティの新しい値を設定する必要があります。

重要 セキュアなプロパティが `echo` コマンドまたはその他の同様のコマンドを使用してスクリプトに出力されると、これらの値はログ ファイルにプレーン テキストで表示されます。ログ ファイルの値はマスクされません。

必要なコンポーネントのプロパティ

プロパティは、必要に応じて設定できます。たとえば、展開中にプロパティに依存するスクリプト用にプロパティが定義されていない場合、展開が失敗することがあります。

たとえば、Apache Tomcat サーバを実行するには Java が必要で、JAVA_HOME プロパティ値を設定する必要があります。

プロパティに必須のマークが付いている場合、ライブラリ、ブループリント、展開プロファイルなど、プロパティの少なくとも 1 つのライフ サイクル ステージで値を指定する必要があります。たとえば、アプリケーション カatalog 管理者は必須プロパティを定義し、オーバーライド可能のマークを付けて、ライブラリのプロパティに値を設定しないこともできます。アプリケーション アーキテクトは、ブループリントでこのプロパティの値を指定するか、または展開プロファイルでオーバーライド可能のマークを付ける必要があります。アプリケーション アーキテクトがこのプロパティに値を設定していない場合、展開担当者はアプリケーションを展開する前に展開プロファイルでこのプロパティに値を設定する必要があります。

コンポーネントの開発のベスト プラクティス

ライブラリのサンプル コンポーネントには、プロパティとアクション スクリプトを定義する方法に関する多くの例が含まれているため、サンプル コンポーネントについて理解しておく必要があります。

注意 ライブラリのサンプル コンポーネントはテスト環境のみで使用してください。

Application Services でコンポーネントを開発するときは、これらのベスト プラクティスに従います。

- 一部のインストーラでは、tty コンソールへのアクセスが必要になります。`/dev/console` からの入力をリダイレクトします。

たとえば、事前定義済みの RabbitMQ サービスは、インストール スクリプトで `./rabbitmq_rhel.py --setup-rabbitmq < /dev/console` コマンドを使用します。

- コンポーネントが複数のライフ サイクル ステージを使用する場合は、INSTALL ライフ サイクル ステージでプロパティ値を変更できます。新しい値は、次のライフ サイクル ステージに送られます。アクション スクリプトは展開中にプロパティの値を計算し、依存しているその他のスクリプトにその値を提供できます。

注意 複数のライフ サイクル ステージを使用するコンポーネントのコンテンツ プロパティの値は変更できません。

たとえば、Clustered Dukes Bank サンプル アプリケーションでは、インストール ライフ サイクル ステージ中に JBossAppServer サービスが JVM_ROUTE プロパティを計算します。このプロパティは、JBossAppServer サービスがライフ サイクルを構成するために使用します。次に、Apache ロード バランサ サービスは、その JVM_ROUTE プロパティを `all(appserver:JBossAppServer:JVM_ROUTE)` プロパティにバインドし、`node0` と `node1` の最終計算値を取得します。

アプリケーションの展開を正常に完了するために、コンポーネントが別のコンポーネントからのプロパティ値を必要とする場合、アプリケーションブループリントに依存関係を明示的に記述する必要があります。

- スクリプトを中断することなく実行するには、戻り値をゼロ (0) に設定する必要があります。

この設定により、エージェントはすべてのプロパティをキャプチャして、Application Services サーバに送信できます。

Application Services ライブラリの管理

Application Services には、標準の論理テンプレート、サンプル サービス、タスク スクリプト、オペレーティングシステム、外部サービス、ポリシー、タグから成る事前入力された拡張可能なライブラリが用意されています。アプリケーション アーキテクトは、このライブラリを使用して、多層エンタープライズ アプリケーションのブループリントを迅速に作成できます。クラウド管理者は、展開担当者が展開時に多層アプリケーションに適用する事前定義済みポリシーを、展開環境で有効にすることができます。

重要 Application Services には、多数の事前定義済み論理テンプレート、サービス、外部サービスが含まれます。アプリケーション カタログ管理者は、これらの事前定義済みテンプレートやサービスをそのまま、または修正して使用できるので、自分で一から作成する必要はありません。ベスト プラクティスは、サービスのコピーを作成し、オリジナルを後で参照できるように保存しておくことです。

ライブラリ内の事前定義済みサービス、タスク、オペレーティングシステム、およびタグは、Application Services のどのユーザーグループでも利用できます。Default グループを除く各グループには、論理テンプレートを追加する必要があります。

Application Services ライブラリを使用して、カスタムのタスク（展開プロファイルの実行プランに追加できるカスタマイズされたスクリプト）を作成できます。

ライブラリの管理に関する主要概念を理解しておく必要があります。[「主要な概念 \(P. 16\)」](#) を参照してください。

この章では次のトピックについて説明します。

- [ライブラリへのオペレーティング システムの追加 \(P. 108\)](#)
- [ライブラリへのタグの追加 \(P. 109\)](#)
- [ライブラリへのサービスの追加 \(P. 109\)](#)
- [ライブラリへの Puppet サービスのインポート \(P. 116\)](#)
- [ライブラリへの外部サービスの追加 \(P. 117\)](#)
- [ライブラリへの論理テンプレートの追加 \(P. 125\)](#)
- [ライブラリへのポリシーの追加 \(P. 128\)](#)
- [カスタム タスクをライブラリに追加する \(P. 132\)](#)

ライブラリへのオペレーティング システムの追加

Application Services に含まれる事前インストール済みオペレーティング システムのリストにオペレーティング システムを追加する必要がある場合があります。

ライブラリでオペレーティング システムの定義を作成して、論理テンプレートで使用するオペレーティング システムや特定のサービスでサポートするオペレーティング システムを指定できます。Application Services は Linux および Windows オペレーティング システムをサポートします。

注意 ライブラリからオペレーティング システムを削除することはできません。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- [オペレーティング システム] ページの事前インストール済みオペレーティング システムについて理解しておく必要があります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [オペレーティング システム] を選択します。
- 2 [新規] をクリックします。
- 3 オペレーティング システムの名前と説明を設定します。
ベスト プラクティスとして、OS 名とバージョン番号を使用し、説明にビットの情報を追加します。
- 4 ドロップダウン メニューからアーキテクチャを選択します。

オプション	説明
[x86_32]	32 ビットのオペレーティング システムに使用します。
[i386]	32 ビットのオペレーティング システムに使用します。
[x86_64]	64 ビットのオペレーティング システムに使用します。

- 5 ドロップダウン メニューからオペレーティング システム ファミリを選択し、[保存] をクリックします。
サービスのアクション スクリプトを作成する場合、スクリプトの種類は指定されているオペレーティング システム ファミリに依存します。Linux ベースのアプリケーションに対しては Bash または BeanShell のスクリプトを使用し、Windows ベースのアプリケーションに対しては Windows CMD、PowerShell、または BeanShell のスクリプトを使用します。
- 6 [OS バージョンの作成] をクリックして、オペレーティング システムのバージョンを作成します。
OS の複数のバージョンを作成できます。
- 7 OS バージョンの情報の入力を完了し、[保存] をクリックします。
OS は同じ名前のままになります。OS のメジャーおよびマイナー バージョンを指定し、サービス パックの詳細を追加できます。

新しいオペレーティング システムが [オペレーティング システム] ページに追加されます。その他のコンポーネントを作成して、ライブラリまたはアプリケーション プループリントに追加するときに、オペレーティング システムを選択できます。

ライブラリへのタグの追加

論理テンプレート、サービス、または外部サービスを作成する際には、1 つ以上のタグをコンポーネントに関連付けることができます。

注意 ライブラリからタグを削除することはできません。

タグの各タイプは交換できません。たとえば、タグのタイプを [サーバタイプ] から [プロパティ記述子] に変更すると、ブループリントキャンパスでの表示に問題が生じたり、サービスを使用するときにエラーメッセージが生成されたりすることがあります。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- [タグ] ページに表示されている事前インストール済みのタグについて理解しておく必要があります。

手順

- 1 Application Services タイトルバーで、ドロップダウンメニューをクリックして [ライブラリ]-[タグ] を選択します。
- 2 [新規] をクリックします。
- 3 タグの名前と説明をテキストボックスに入力します。
- 4 ドロップダウンメニューからタグのタイプを選択します。

[プロパティ記述子] タグタイプは、サービス プロパティの [タグの自動バインド] ドロップダウンメニューに選択肢として表示されます。[サーバタイプ] タグは、サービス テンプレートまたは論理テンプレートを作成するときに、ドロップダウンメニューの選択肢として表示されます。
- 5 [保存] をクリックします。

新たに作成されたタグは、[タグ] ページに表示されます。サービス、タスク、または論理テンプレートをライブラリまたはアプリケーションブループリントに追加するときに、タグを選択できます。

ライブラリへのサービスの追加

サービスは、アプリケーションが必要とするソフトウェアサービスをインストール、構成、起動、更新するスクリプトで構成されています。Application Services ライブラリにカスタムサービスを追加できます。

手順

- 1 [ライブラリでサービスバージョンを作成する](#) (P. 110)
Application Services ライブラリを使用して、ソフトウェアサービスバージョンを作成します。Application Services ライブラリには事前定義済みサービスも含まれます。
- 2 [サービスバージョンプロパティを定義する](#) (P. 111)
Application Services は、仮想マシンで実行中のスクリプトに、環境変数として定義されたプロパティを渡します。サービスバージョンのプロパティでは、サービスのスクリプトで使用される変数を定義します。
- 3 [サービスバージョンへのアクションスクリプトの追加](#) (P. 114)
アプリケーションカタログ管理者は、少なくとも 1 つのライフサイクルステージにアクションスクリプトを提供する必要があります。アプリケーションの展開では、インストール、構成、開始、ロールバック、および分解の各スクリプトを作成でき、既存の展開を更新するための更新スクリプトを作成することもできます。これらのスクリプトは、コンポーネントのプロパティを使用するためにカスタマイズされます。
- 4 [サービスバージョンを保守する](#) (P. 115)
既存のサービスバージョンからサービスをコピーまたは編集できます。

5 ライブラリからサービスを削除する (P. 116)

サービスは [サービス] ページから削除できます。

ライブラリでサービス バージョンを作成する

Application Services ライブラリを使用して、ソフトウェア サービス バージョンを作成します。Application Services ライブラリには事前定義済みサービスも含まれます。

場合によって、新しいサービスを作成する代わりに、事前定義済みサービスのスクリプトおよび変数を編集することもできます。ベスト プラクティスとして、変更を行う前に事前定義済みサービスのコピーを作成します。[「サービス バージョンを保守する \(P. 115\)」](#) を参照してください。

サービスはパブリックに共有することもグループ内でプライベートに共有することもできます。共有はコンポーネント レベルでサポートされ、個別のバージョン レベルではサポートされません。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- オペレーティング システムおよび説明タグ一覧にアイテムを追加する必要があるかどうかを確認します。[「ライブラリへのオペレーティング システムの追加 \(P. 108\)」](#) および [「ライブラリへのタグの追加 \(P. 109\)」](#) を参照してください。
- SELinux が有効なオペレーティング システムを追加する予定の場合は、インストールの失敗を防ぐために、許可モードが有効であるか、インストールするアプリケーションで特定の例外が適切であることを確認してください。
- サービスがブループリントで使用されているか、論理テンプレートに事前インストール済みサービスとして含まれている場合、サービスは削除できません。サービスを削除する前に、参照されるブループリントまたはプレインストールされたサービスを含む論理テンプレートを削除してください。
- サービスの命名または記述方法に関する例を表示するには、[ライブラリ] - [サービス] を選択して事前定義済みサービスを表示します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [サービス] を選択します。
- 2 [新規] をクリックします。
- 3 サービスの名前を設定し、説明を追加して [保存] をクリックします。
アクション スクリプトが関連するソフトウェア コンポーネントに対応した名前を使用し、追跡するために修飾子を追加します。たとえば、特定のオペレーティング システムでサービスを使用する場合、サービス名を <ServiceName>_Windows または <ServiceName>_Linux と設定できます。
- 4 [サービス バージョンの作成] をクリックしてサービス バージョンを作成します。
1 つのサービスについて複数のバージョンを作成できます。
サービス バージョンを作成するためのページが開きます。
- 5 サービス バージョン情報を入力します。
サービス バージョン名は同じままにします。
 - a メジャー リリース、マイナー リリース、またはマイクロ リリースを修飾子付き/なしで指定します。
たとえば、1.0、1.5、または 1.0.1-Linux などのバージョン番号を使用できます。
 - b (オプション) 特定のオペレーティング システムのサービス バージョンを作成している場合は、[説明] セクションに使用するオペレーティング システム バージョンと適用可能な必須構成を記述します。

- 6 タグを設定し、アプリケーションの展開ブループリントを作成するときに表示されるサービスのリストを編成します。

注意 サーバタイプとして指定されたタグのみがドロップダウンメニューに表示されます。

複数のタグを追加できます。

- 7 リストにないタグを作成するには、[キャンセル] をクリックして [ライブラリ] - [タグ] を選択します。
- 8 (オプション) [サポートされる OS] セクションで、このサービスで使用するスクリプトが特定のオペレーティングシステムでしか実行できない場合、ここでそれらのオペレーティングシステムを選択します。

これらのオペレーティングシステムのいずれかがテンプレートに含まれない限り、ブループリント エディタでは、Application Services でサービスをテンプレートに追加することはできません。サービスがどのオペレーティングシステムでも使用できる場合、このフィールドを空白のままにします。

- 9 リストにないオペレーティングシステム名を使用するには、[キャンセル] をクリックして [ライブラリ] - [オペレーティングシステム] を選択し、オペレーティングシステム名を作成します。
- 10 (オプション) [サポートされるコンポーネント] セクションで、アプリケーション コンポーネントの特定のタイプしかこのサービスで実行できない場合、ここでそのコンポーネントを指定します。

たとえば、vFabric tc Server インスタンスでは WAR および JAR コンポーネントしか実行できません。データベース サーバで実行できるのは、SQL スクリプトのみです。選択したコンポーネントによって、アプリケーション ブループリントでこのサービスに追加できるアプリケーション コンポーネントが制限されます。コンポーネントをサービスに追加できる場合は、このフィールドを空白のままにします。

- 11 論理テンプレートの作成中または編集中に、[含まれるサービス] セクションでサービスを一覧表示するには、[テンプレートにプレインストールする] チェック ボックスを選択します。

このチェックボックスを選択すると、サービスがすでにテンプレートにインストールされていることを示します。

このチェック ボックスを選択しなかった場合、サービスがサービスのリストに表示され、アプリケーションの展開ブループリントを作成するときに含めることができます。

プレインストールされたサービスを含む論理テンプレートもアプリケーション ブループリントの [OS テンプレート] セクションに表示され、テンプレートの一部として使用できます。

次に進む前に

サービス バージョン プロパティを構成します。[「サービス バージョン プロパティを定義する \(P. 111\)」](#) を参照してください。

サービス バージョン プロパティを定義する

Application Services は、仮想マシンで実行中のスクリプトに、環境変数として定義されたプロパティを渡します。サービス バージョンのプロパティでは、サービスのスクリプトで使用される変数を定義します。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- サービスを定義する方法の例を表示するには、[ライブラリ] - [サービス] を選択して、事前定義済みサービスを表示します。
- サービス バージョンが Application Services で使用できることを確認します。[「ライブラリでサービス バージョンを作成する \(P. 110\)」](#) を参照してください。

手順

- 1 既存のサービス バージョンを開き、[編集] をクリックします。

- 2 [プロパティ名] と [説明] では、変数を定義してオプションの説明を追加します。

プロパティ名の先頭に数字を指定することはできません。

変数が追加および定義されたら、サービス バージョンのインストール、構成、開始、更新、ロールバック、および分解スクリプトを作成できます。

- 3 タイプを変更するには、ドロップダウン メニューからプロパティ タイプを選択します。

メニューにタイプを追加することはできません。

- 4 スクリプトの実行時に、このプロパティを置き換える値を入力します。

たとえば `http_port` というプロパティの場合、このフィールドに **80** を入力できます。

注意 vCloud Director または vRealize Automation で、ファイル ソースとして `darwin_global.conf` ファイルを使用するサービス スクリプトによって `http_proxy`、`https_proxy`、または `ftp_proxy` プロパティに値を追加すると、スクリプトが実行されたときに、展開済みアプリケーション内の既存のプロキシ情報がこれらのプロパティによってオーバーライドされます。Amazon EC2 では、アプリケーションを展開するためのプロキシは必要ありません。

プロパティ タイプが [計算済み] に設定されている場合は、値を入力できません。

[必須] チェックボックスが選択され、[ブループリントでオーバーライド可] チェック ボックスが選択されていない場合は、プロパティに値を入力する必要があります。

- 5 自動バインド ドロップダウン メニューから、自動バインド タイプを選択します。

自動バインド タイプは、ブループリント キャンパス内で自動バインドするために、プロパティが割り当てられるロールを示します。

オプション	説明
[なし]	プロパティは自動バインドしません。
[消費]	プロパティ値は、自動バインドにより設定されます。
[公開]	プロパティ値を使用して、自動バインドにより別のプロパティ値をカスタマイズできます。

注意 1 つのプロパティに対して、利用と公開のロールは設定できません。

[消費] または [公開] の自動バインド タイプでは、プロパティに対して少なくとも 1 つの自動バインド タグを定義する必要があります。消費プロパティの自動バインド タグは、公開プロパティのタグのサブセットである必要があります。タグ セットは同一である必要はありません。

- 6 プロパティ タイプを定義します。

定義したプロパティ タイプは、設定できる自動バインド タイプに影響します。プロパティ タイプと自動バインド タイプの次の組み合わせは使用できません。

プロパティ タイプ	設定できない自動バインド タイプ
アレイ	公開
計算値	利用
コンテンツ	公開または利用
単一 Select	公開または利用

- 7 自動バインド ドロップダウン メニューで [新規] をクリックし、自動バインド タイプを選択して、プロパティにプロパティ記述子タグを追加します。

注意 ドロップダウン メニューには、プロパティ記述子としてのみ指定されたタグが表示されます。

複数の自動バインド タグを追加できます。

たとえば、Apache 2.2.0 サービスの `appsrv_routes` プロパティの自動バインド タイプが [消費] に割り当てられ、vFabric tc Server 2.1.0 サービスの `JVM_ROUTE` プロパティの自動バインド タイプが [公開] に割り当てられている場合、`appsrv_routes` プロパティは `JVM_ROUTE` プロパティ値を使用して、自身をカスタマイズします。自動バインド機能がこれらのサービス プロパティに接続できるようにするために、サブレット コンテナ タグとルート タグを、それぞれ `appsrv_routes` プロパティと `JVM_ROUTE` プロパティに追加できます。vFabric tc Server および Apache サービスを含むアプリケーションを作成する場合に、2 つのノード間に依存関係が存在すると、プロパティはブループリント エディタで互いに自動バインドします。

- 8 アプリケーションの展開に必要なプロパティで、[必須] チェック ボックスを選択します。

必須プロパティ値が空白の場合は、アプリケーションを展開する前にプロパティ値を入力するように求められます。

プロパティ タイプが [計算済み] の場合、[必須] チェック ボックスは適用されません。プロパティ タイプが [コンテンツ] の場合、[必須] チェックボックスはデフォルトで選択されており、[セキュア] チェック ボックスは適用されません。

たとえば、Apache JServ Protocol (AJP) を使用する `http_proxy_port` プロパティの場合は、値を **8009** に設定する必要があります。

- 9 定義するパスワードまたはその他のプロパティの値を見えなくする場合は、[セキュアド] チェック ボックスをオンにします。

プロパティが [セキュアド] から [アンセキュアド] に変更されると、Application Services は、セキュリティ上の理由によってプロパティ値をリセットします。プロパティに新しいパスワード値を設定する必要があります。

たとえば、サンプルの MySQL サービス内の `db_password` プロパティは、セキュアです。

- 10 [ブループリントでオーバーライド可] チェック ボックスを選択し、アプリケーション アーキテクトなどのユーザーが、アプリケーション ブループリント内のプロパティの値をオーバーライドできるようにします。

プロパティ タイプが [計算済み] の場合、[ブループリントでオーバーライド可] チェック ボックスは適用されません。

プロパティがブループリントでオーバーライドされない場合は、自動バインド タイプを [消費] に設定できません。

たとえば、アプリケーション カタログ管理者は、512 MB の JVM ヒープ サイズが割り当てられるように vFabric tc Server サービスを構成することがあります。ただし、大規模な展開に対応するために、アプリケーション アーキテクトでは設定を 1024 MB に変更できます。

- 11 [プロパティ] セクションから選択された行を削除するには、[削除] をクリックします。

- 12 (オプション) 元のプロパティ値に戻すには、[リセット] をクリックします。

- 13 選択した CSV ファイルにプロパティ値をインポートするには、[アップロード] をクリックします。

- 14 サービス プロパティ定義を保存します。

次に進む前に

サービス バージョンのアクション スクリプトを作成します。[「サービス バージョンへのアクション スクリプトの追加 \(P. 114\)」](#) を参照してください。

サービス バージョンへのアクション スクリプトの追加

アプリケーション カタログ管理者は、少なくとも 1 つのライフ サイクル ステージにアクション スクリプトを提供する必要があります。アプリケーションの展開では、インストール、構成、開始、ロールバック、および分解の各スクリプトを作成でき、既存の展開を更新するための更新スクリプトを作成することもできます。これらのスクリプトは、コンポーネントのプロパティを使用するためにカスタマイズされます。

すべてのライフ サイクル ステージにスクリプトを追加する必要はありません。特定のステージを必要としない場合は、それを無視できます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者および展開者**として Application Services にログインします。
- 外部 Web サイトからソフトウェアをダウンロードするスクリプトを使用する場合は、アプリケーションの展開に使用する仮想マシンが外部ネットワークにアクセスできることを確認します。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- スクリプトをサービスに追加する方法の例を表示するには、[ライブラリ] - [サービス] を選択して、事前定義されたサービスを表示します。
- サービス バージョンに対してプロパティが定義されていることを確認します。[「サービス バージョン プロパティを定義する \(P. 111\)」](#) を参照してください。

手順

- 1 既存のサービス バージョンを開き、[編集] をクリックします。
- 2 ドロップダウン メニューから、アクション スクリプトのスクリプト タイプを選択します。
Linux ベースのアプリケーションに対しては Bash または BeanShell のスクリプトを作成し、Windows ベースのアプリケーションに対しては Windows CMD、PowerShell、または BeanShell のスクリプトを作成します。
- 3 [スクリプト] 列で、ハイパーリンクをクリックして [スクリプトの編集] ダイアログ ボックスを開きます。
スクリプトを記述するか、またはダイアログ ボックスにスクリプトをコピーできます。
アプリケーション カタログ管理者は、サービスのインストールと構成をパラメータ化できます。サービス用に定義されたプロパティはスクリプト内で使用できます。
- 4 スクリプト タイプ フィールドで下矢印をクリックし、ライフ サイクル ステージのアクション スクリプト タイプを選択します。
- 5 定義したプロパティを挿入するには、[挿入するプロパティを選択] リストの下矢印をクリックします。
- 6 (オプション)
アプリケーションの展開中にスクリプトが正常に実行されたら仮想マシンを再起動するために、[再起動] チェック ボックスをオンにします。
- 7 [OK] をクリックします。
- 8 [再起動] 列で、アクション スクリプトが正常に実行された後にエージェント ブートストラップが仮想マシンを再開できるように、このチェック ボックスをオンにします。
仮想マシンが再開された後に、エージェントはサービス バージョンで定義されている次のライフ サイクル ステージ スクリプトに進みます。
- 9 [リセット] をクリックして、スクリプトをクリアします。
この操作によって、ライフ サイクル ステージの行は削除されません。
- 10 サービス バージョンの作成が終了したら、[保存] をクリックします。

11 [OK] をクリックします。

作成したサービスがページに追加されます。

次に進む前に

既存のサービス バージョンのコピーを作成するか、またはサービスを編集します。[「サービス バージョンを保守する \(P. 115\)」](#) を参照してください。

サービス バージョンを保守する

既存のサービス バージョンからサービスをコピーまたは編集できます。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- アプリケーション ブループリントに追加するサービスに含まれるプロパティとスクリプトを編集する場合は、[「ライブラリへのサービスの追加 \(P. 109\)」](#) に記載されたタスクについて理解しておく必要があります。
- 外部 Web サイトからソフトウェアをダウンロードするスクリプトを使用する場合は、アプリケーションの展開に使用する仮想マシンが外部ネットワークにアクセスできることを確認します。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- サービスのコピーを作成して、オリジナルは後で参照できるように保存し、コピーしたバージョンを編集します。
- そのサービス バージョンを所有するビジネス グループのメンバーである必要があります。

サービス バージョンを所有するビジネス グループのメンバーでない場合は、[コピー] ボタンおよび [編集] ボタンは無効になっています。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [サービス] を選択します。
- 2 既存のサービス バージョンをコピーします。
 - a サービス バージョンを開き、ツールバーで [コピー] をクリックします。
 サービスをコピーすると、すべてのプロパティ、アクション スクリプト、オペレーティングシステム、説明タグのリスト、サポートされるアプリケーション コンポーネントが保持されます。既存のサービス バージョンの名前は変更できません。
 - b バージョン番号を変更し、説明を更新してから [保存] をクリックします。
 バージョン番号と説明によって、新しいサービスが親バージョンと区別されます。
- 3 変数を変更するかアクション スクリプトを更新するには、ツールバーで [編集] をクリックし、変更を加えてから [保存] をクリックします。

次に進む前に

アプリケーション ブループリントをモデル化するために使用しないサービスは削除できます。[「ライブラリからサービスを削除する \(P. 116\)」](#) を参照してください。

ライブラリからサービスを削除する

サービスは [サービス] ページから削除できます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- サービスがブループリントで使用されていないこと、または論理テンプレートに事前インストール済みサービスとして含まれていないことを確認します。

参照先ブループリントまたは事前インストール済みサービスを含む論理テンプレートを削除します。

- そのサービス バージョンを所有するビジネス グループのメンバーである必要があります。
- サービス バージョンを所有するビジネス グループのメンバーになっていない場合、[削除] ボタンは無効になっています。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [サービス] を選択します。
- 2 サービスを開きます。
- 3 [削除] を選択して既存のサービス バージョンを削除し、選択内容を確認します。
- 4 サービス名の横にある矢印をクリックし、[サービス] ページに戻ります。
- 5 サービスを削除して選択内容を確認します。

ライブラリへの Puppet サービスのインポート

Puppet サービスは、アプリケーション ブループリントをビルドするために使用できます。Puppet モジュールおよび定義済みリソースを、サービスとして Application Services ライブラリにインポートする必要があります。

複数の Puppet モジュールを同時にインポートできます。

Puppet モジュールをインポートすると、これらのサービスに Puppet Services と呼ばれる特殊なタグが付きます。新たにインポートされたサービスにアクション スクリプトはありません。Application Services はそのライフ サイクル アクティビティを、対応する展開環境の登録済み Puppet マスター インスタンスに委任します。

インポートした Puppet コンテンツは更新できません。サービスを削除したり、既存の Puppet サービス バージョンを変更したり、更新済み Puppet コンテンツをライブラリにインポートしたりすることができます。デフォルト バージョンは、存在しないバージョンの情報がある 1.0.0-unknown for Puppet モジュールです。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- Application Services CLI を起動します。[「リモートでの CLI の開始 \(P. 221\)」](#) を参照してください。
- CLI を実行するために使用するリモート マシンは、Puppet モジュールがインストールされていて Puppet CLI が使用できる必要があります。

Puppet マスター マシンを使用して、Puppet モジュールを 2 台の異なるマシンで保守することもできます。

- インポートしている Puppet サービスが Puppet マスターで使用できることを確認します。

手順

- ◆ CLI プログラムにログインしてから、`roo` シェルで Apache などの Puppet モジュールをインポートするコマンドを入力します。

```
import-puppet-manifest --typeFilter "^apache$" --targetGroup pubs
```

コマンドのパラメータは大文字と小文字が区別されます。大文字と小文字を間違えると、コマンドは無視されます。

パラメータ	説明
typeFilter	インポートを指定した検索表現に一致するタイプに制限します。 表現の構文の詳細については、Puppet のドキュメントを参照するか、 puppet man resource_type と入力して man ページを表示してください。 たとえば、mysql モジュールをインポートするには import-puppet-manifests --typeFilter "^mysql\$" と入力します。
osfamily	インポートされるタイプのオペレーティングシステムを指定します。オプションは WINDOWS と LINUX です。 Linux がデフォルトのオペレーティングシステムです。
puppetPath	Puppet バイナリ ファイルの場所を指定します。
shared	成果物が共有されているか、プライベートかを示します。 このコマンドで true または false を定義できます。デフォルト設定である false は、 成果物がインポート操作を実行するユーザーのグループに対してプライベートであることを示します。
targetGroup	インポートしたマニフェストのビジネス グループを指定します。これは、サービスを 所有するビジネス グループです。

このコマンドで、apache Puppet モジュールをライブラリにインポートします。

次に進む前に

Puppet コンテンツをブループリント キャンパスの [Puppet サービス] セクションから新規または既存のアプリケーション ブループリントに追加し、アプリケーションを展開します。「[アプリケーションの作成 \(P. 138\)](#)」 および「[展開プロファイルの設定と構成 \(P. 157\)](#)」を参照してください。

ライブラリへの外部サービスの追加

Application Services ライブラリにおける外部サービスは、アプリケーション展開の外部にインストールされるサービスのことで、外部サービスとアプリケーションは、互いに連携して動作するように構成する必要があります。

手順

- 1 [ライブラリでの外部サービスの作成 \(P. 118\)](#)

Application Services を使用すると、基本または高度な外部サービスを作成できます。

- 2 [外部サービス プロパティを定義する \(P. 119\)](#)

Application Services は、仮想マシンで実行中のスクリプトに、環境変数として定義されたプロパティを渡します。外部サービス バージョンのプロパティによって、外部サービスのスクリプトで使用される変数を定義します。

- 3 [高度な外部サービス バージョンのプロバイダ仕様を作成する \(P. 122\)](#)

高度な外部サービスを使用するには、プロバイダ仕様を作成する必要があります。プロバイダ仕様とは、VMware Data Director や Amazon RDS などのさまざまなサービス プロバイダ環境で外部サービスを構成、更新、ロールバック、分解する方法を定義した 1 つ以上のスクリプトのセットのことです。プロバイダ仕様には、プロバイダに接続するために定義可能なプロパティおよびプロバイダ固有のパラメータもあります。

- 4 [外部サービス バージョンを保守する \(P. 123\)](#)

基本または高度な外部サービスを既存のバージョンからコピーしたり、サービスを変更したりできます。

5 外部サービスを削除する (P. 124)

外部サービスを管理するために、重複するサービスや古いサービスを削除できます。

ライブラリでの外部サービスの作成

Application Services を使用すると、基本または高度な外部サービスを作成できます。

基本の外部サービスは、複数のサービス プロバイダをサポートしません。論理テンプレートを定義し、CONFIGURE、UPDATE、ROLLBACK、および TEARDOWN ライフ サイクル ステージのスクリプトを追加するオプションが必要です。基本の外部サービスの一部の例には、アプリケーション スキーマがインストールされた既存のデータベース、Workday などの SAAS アプリケーション、または Single Sign-On (SSO) サーバが含まれます。

高度な外部サービスには、外部サービスの複数のプロバイダのサポートを追加するプロバイダ仕様のオプションが含まれます。高度な外部サービス プロパティ値をプロバイダ仕様で定義した後でインスタンスを展開環境にマッピングすると、これらの値が外部サービス インスタンスに入力されます。たとえば、外部 Oracle データベースは VMWare Data Director または Amazon RDS を使用してプロビジョニングし、構成できます。アプリケーションの外部ロード バランサは、F5 ロード バランサ インスタンスまたは Apache サーバなどのソフトウェア ロード バランサと一緒に実装できます。

外部サービスはパブリックに共有することもグループ内でプライベートに共有することもできます。共有はコンポーネント レベルでサポートされ、個別のバージョン レベルではサポートされません。外部サービスを共有する場合、外部サービス インスタンス プロファイルは共有されません。各グループはその展開環境で外部サービス インスタンス プロファイルを定義する必要があります。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- オペレーティング システムおよび説明タグ一覧にアイテムを追加する必要があるかどうかを確認します。[「ライブラリへのオペレーティング システムの追加 \(P. 108\)」](#) および [「ライブラリへのタグの追加 \(P. 109\)」](#) を参照してください。
- 外部サービスの命名または記述方法に関する例を表示するには、[ライブラリ]-[外部サービス] を選択して事前定義済み外部サービスを表示してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ]-[外部サービス] を選択します。
- 2 [新規] をクリックします。
- 3 外部サービスの名前を設定し、説明を追加して [保存] をクリックします。
自分のサーバまたはデータベースに対応する名前を使用してください。
- 4 [外部サービス バージョンの作成] をクリックして外部サービス バージョンを作成します。
1 つの外部サービスについて複数のバージョンを作成できます。
- 5 このダイアログ ボックスで [基本] または [詳細] を選択して [OK] をクリックします。
- 6 外部サービス バージョン情報を入力します。
外部サービス バージョン名は同じままにします。
 - a メジャー リリース、マイナー リリース、またはマイクロ リリースを修飾子付き/なしで指定します。
たとえば、1.0、1.5、または 1.0.1-Dev などのバージョン番号を使用できます。
 - b (オプション) [説明] セクションには、特定のデータベースの外部サービス バージョンを作成している場合、使用するデータベース バージョンと適用可能な必須構成を記述します。

- 7 [新規] をクリックしてドロップダウン メニューからタグを設定します。

タグは外部サービスをグループ化し、アプリケーションの展開ブループリントを作成するときに対応するグループで表示できます。

複数のタグを追加できます。

- 8 リストにないタグを作成するには、[キャンセル] をクリックして [ライブラリ] - [タグ] を選択します。
- 9 (オプション) [サポートされるコンポーネント] セクションで、アプリケーション コンポーネントの特定のタイプしか高度な外部サービスで実行できない場合、[新規] をクリックしてこうしたコンポーネントをここで指定します。

たとえば、データベース サーバでは SQL スクリプトしか実行できません。選択したコンポーネントによって、アプリケーション ブループリントでこの高度な外部サービスに追加できるアプリケーション コンポーネントが制限されます。コンポーネントを高度な外部サービスに追加できる場合は、このフィールドを空白のままにします。

- 10 外部サービスのオペレーティング システムを指定します。

- a 基本の外部サービスの場合、[論理テンプレート] セクションでドロップダウン メニューから Windows または Linux ベースのテンプレートを選択します。

論理テンプレートに関連付けられたサポートされる OS が入力されます。

- b 高度な外部サービスの場合、高度な外部サービスで使用するスクリプトが特定のオペレーティング システムでのみ実行できる場合、[新規] をクリックしてこれらのオペレーティング システムを [サポートされる OS] セクションに追加します。

これらのオペレーティング システムのいずれかがテンプレートに含まれない限り、ブループリント エディタでは、Application Services で外部サービスをテンプレートに追加できません。高度な外部サービスがどのオペレーティング システムでも使用できる場合、このフィールドを空白のままにします。

- 11 リストにないオペレーティング システム名を使用するには、[キャンセル] をクリックして [ライブラリ] - [オペレーティング システム] を選択し、オペレーティング システム名を作成します。

次に進む前に

外部サービス バージョン プロパティを構成します。[「外部サービス プロパティを定義する \(P. 119\)」](#) を参照してください。

外部サービス プロパティを定義する

Application Services は、仮想マシンで実行中のスクリプトに、環境変数として定義されたプロパティを渡します。外部サービス バージョンのプロパティによって、外部サービスのスクリプトで使用される変数を定義します。

事前構成済み外部サービスまたはカスタム外部サービスと共にアプリケーションを展開する場合、展開が処理中の間は、その外部サービスのプロパティ定義またはプロバイダ仕様プロパティを編集できません。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者および展開者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- 外部サービスを定義する方法の例を表示するには、[ライブラリ] - [外部サービス] を選択して、事前定義済みサービスを表示します。
- 外部サービス バージョンが Application Services で利用できることを確認します。[「ライブラリでの外部サービスの作成 \(P. 118\)」](#) を参照してください。

手順

- 1 既存の外部サービス バージョンを開き、[編集] をクリックします。

- 2 [プロパティ名] と [説明] では、変数を定義してオプションの説明を追加します。
プロパティ名の先頭に数字を指定することはできません。
- 3 タイプを変更するには、ドロップダウン メニューからプロパティ タイプを選択します。
メニューにタイプを追加することはできません。
- 4 スクリプトの実行時に、このプロパティを置き換える値を入力します。
たとえば `http_port` というプロパティの場合、このフィールドに **80** を入力できます。

注意 vCloud Director または vRealize Automation で、ファイルソースとして `darwin_global.conf` ファイルを使用するサービス スクリプトによって `http_proxy`、`https_proxy`、または `ftp_proxy` プロパティに値を追加すると、スクリプトが実行されたときに、展開済みアプリケーション内の既存のプロキシ情報がこれらのプロパティによってオーバーライドされます。Amazon EC2 では、アプリケーションを展開するためのプロキシは必要ありません。

プロパティ タイプが [計算済み] に設定されている場合は、値を入力できません。

[必須] チェックボックスが選択され、[ブループリントでオーバーライド可] チェック ボックスが選択されていない場合は、プロパティに値を入力する必要があります。

- 5 自動バインド ドロップダウン メニューで [新規] をクリックし、自動バインド タイプを選択して、プロパティにプロパティ記述子タグを追加します。

注意 ドロップダウン メニューには、プロパティ記述子としてのみ指定されたタグが表示されます。

複数の自動バインド タグを追加できます。

たとえば、Apache 2.2.0 サービスの `appsrv_routes` プロパティの自動バインド タイプが [消費] に割り当てられ、vFabric tc Server 2.1.0 サービスの `JVM_ROUTE` プロパティの自動バインド タイプが [公開] に割り当てられている場合、`appsrv_routes` プロパティは `JVM_ROUTE` プロパティ値を使用して、自身をカスタマイズします。自動バインド機能がこれらのサービス プロパティに接続できるようにするために、サブレット コンテナ タグとルート タグを、それぞれ `appsrv_routes` プロパティと `JVM_ROUTE` プロパティに追加できます。vFabric tc Server および Apache サービスを含むアプリケーションを作成する場合に、2 つのノード間に依存関係が存在すると、プロパティはブループリント エディタで互いに自動バインドします。

- 6 自動バインド ドロップダウン メニューから、自動バインド タイプを選択します。

自動バインド タイプは、ブループリント キャンパス内で自動バインドするために、プロパティが割り当てられるロールを示します。

オプション	説明
[なし]	プロパティは自動バインドしません。
[消費]	プロパティ値は、自動バインドにより設定されます。
[公開]	プロパティ値を使用して、自動バインドにより別のプロパティ値をカスタマイズできます。

注意 1 つのプロパティに対して、利用と公開のロールは設定できません。

[消費] または [公開] の自動バインド タイプでは、プロパティに対して少なくとも 1 つの自動バインド タグを定義する必要があります。消費プロパティの自動バインド タグは、公開プロパティのタグのサブセットである必要があります。タグ セットは同一である必要はありません。

- 7 アプリケーションの展開に必要なプロパティで、[必須] チェック ボックスを選択します。

必須プロパティ値が空白の場合は、アプリケーションを展開する前にプロパティ値を入力するように求められます。

プロパティ タイプが [計算済み] の場合、[必須] チェック ボックスは適用されません。プロパティ タイプが [コンテンツ] の場合、[必須] チェック ボックスはデフォルトで選択されており、[セキュア] チェック ボックスは適用されません。

たとえば、Apache JServ Protocol (AJP) を使用する http_proxy_port プロパティの場合は、値を **8009** に設定する必要があります。

- 8 定義するパスワードまたはその他のプロパティの値を見えなくする場合は、[セキュアド] チェック ボックスをオンにします。

プロパティが [セキュアド] から [アンセキュアド] に変更されると、Application Services は、セキュリティ上の理由によってプロパティ値をリセットします。プロパティの新しい値を設定する必要があります。

- 9 [オーバーライド] ドロップダウン メニューから、プロパティのオーバーライド オプションを選択します。

オプション	説明
ブループリント	アプリケーション ブループリントでプロパティをオーバーライドできます。 クラウド管理者は、外部サービス インスタンスを展開環境に登録するときに、このプロパティを表示できません。
展開環境	クラウド管理者は、展開環境でプロパティをオーバーライドできます。 アプリケーション ブループリント キャンパスにプロパティが表示され、ユーザーはプロパティに自動バインドを実行できます。アプリケーション ブループリントでは、プロパティ値を編集できません。
オーバーライド不可	プロパティはオーバーライドできません。

- 10 [ブループリントでオーバーライド可] チェック ボックスを選択し、アプリケーション アーキテクトなどのユーザーが、アプリケーション ブループリント内のプロパティの値をオーバーライドできるようにします。

プロパティ タイプが [計算済み] の場合、[ブループリントでオーバーライド可] チェック ボックスは適用されません。

プロパティがブループリントでオーバーライドされない場合は、自動バインド タイプを [消費] に設定できません。

たとえば、アプリケーション カタログ管理者は、512 MB の JVM ヒープ サイズが割り当てられるように vFabric tc Server サービスを構成することがあります。ただし、大規模な展開に対応するために、アプリケーション アーキテクトでは設定を 1024 MB に変更できます。

- 11 [プロパティ] セクションから選択された行を削除するには、[削除] をクリックします。

- 12 (オプション) 元のプロパティ値に戻すには、[リセット] をクリックします。

- 13 選択した CSV ファイルにプロパティ値をインポートするには、[アップロード] をクリックします。

- 14 外部サービス プロパティ定義を保存します。

次に進む前に

展開環境で外部サービス インスタンスを作成して、展開環境下のすべての展開で外部サービス定義を有効にしない限り、外部サービス定義を指定しても展開に影響しません。[「外部サービス インスタンスのマッピング \(P. 61\)」](#) を参照してください。

カスタム外部サービス バージョンのプロバイダ仕様を作成します。[「高度な外部サービス バージョンのプロバイダ仕様を作成する \(P. 122\)」](#) を参照してください。

高度な外部サービス バージョンのプロバイダ仕様を作成する

高度な外部サービスを使用するには、プロバイダ仕様を作成する必要があります。プロバイダ仕様とは、VMware Data Director や Amazon RDS などのさまざまなサービス プロバイダ環境で外部サービスを構成、更新、ロールバック、分解する方法を定義した 1 つ以上のスクリプトのセットのことです。プロバイダ仕様には、プロバイダに接続するために定義可能なプロパティおよびプロバイダ固有のパラメータもあります。

外部サービス用プロバイダ仕様の作成手順の例は、[ライブラリ] - [外部サービス] ページから表示できます。

ユーザーが変更できるのは、外部サービスのグループ メンバーシップ設定だけです。変更されたメンバーシップ設定は、外部サービスのサービス プロバイダ仕様に反映されます。

サービス プロバイダ仕様のグループ メンバーシップを個別に更新することはできません。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者および展開者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。第 12 章「[Application Services コンポーネントの開発 \(P. 93\)](#)」を参照してください。
- 外部 Web サイトからソフトウェアをダウンロードするスクリプトを使用する場合は、アプリケーションの展開に使用する仮想マシンが外部ネットワークにアクセスできることを確認します。
- 外部サービス バージョンのプロパティが定義されていることを確認します。Application Services を参照してください。[「ライブラリでの外部サービスの作成 \(P. 118\)」](#)を参照してください。

手順

- 1 既存のサービス バージョンを開き、[プロバイダ仕様バージョンの作成] をクリックします。
- 2 プロバイダ仕様名を入力し、[バージョンの作成] を選択します。
- 3 メジャー リリース、マイナー リリース、またはマイクロ リリースを修飾子付き/なしで指定します。
プロバイダ仕様名はそのままにします。
- 4 (オプション) 特定の構成のプロバイダ仕様を作成する場合は、[説明] セクションに、使用する構成と適用する要件を記述します。
- 5 サポートされる OS を Windows または Linux ベースのオペレーティング システムに設定している場合は、[論理テンプレート] セクションのドロップダウン メニューに、対応する論理テンプレートが表示されます。
- 6 適切なプロパティを追加します。
- 7 ドロップダウン メニューから、アクション スクリプトのスクリプト タイプを選択します。
Linux ベースのアプリケーションに対しては Bash または BeanShell のスクリプトを作成し、Windows ベースのアプリケーションに対しては Windows CMD、PowerShell、または BeanShell のスクリプトを作成します。
- 8 [スクリプト] 列で、ハイパーリンクをクリックして [スクリプトの編集] ダイアログ ボックスを開きます。
スクリプトを記述するか、またはダイアログ ボックスにスクリプトをコピーできます。
アプリケーション カタログ管理者は、サービスのインストールと構成をパラメータ化できます。サービス用に定義されたプロパティはスクリプト内で使用できます。
- 9 スクリプト タイプ フィールドで下矢印をクリックし、ライフ サイクル ステージのアクション スクリプト タイプを選択します。
- 10 定義したプロパティを挿入するには、[挿入するプロパティを選択] リストの下矢印をクリックします。

11 (オプション)

アプリケーションの展開中にスクリプトが正常に実行されたら仮想マシンを再起動するために、[再起動] チェックボックスをオンにします。

12 [OK] をクリックします。

13 [再起動] 列で、アクション スクリプトが正常に実行された後にエージェント ブートストラップが仮想マシンを再開できるように、このチェック ボックスをオンにします。

仮想マシンが再開された後に、エージェントはサービス バージョンで定義されている次のライフ サイクル ステージ スクリプトに進みます。

14 [リセット] をクリックして、スクリプトをクリアします。

この操作によって、ライフ サイクル ステージの行は削除されません。

15 プロバイダ仕様バージョンの作成が終了したら、[保存] をクリックします。

作成したプロバイダ仕様バージョンが高度な外部サービスに追加されます。

次に進む前に

展開環境で外部サービス インスタンスを作成して、展開環境下のすべての展開で外部サービス定義を有効にしない限り、外部サービス定義を指定しても展開に影響しません。[「外部サービス インスタンスのマッピング \(P. 61\)」](#) を参照してください。

外部サービスをアプリケーション プループリントに追加します。[「詳細プループリントへの外部サービスの追加 \(P. 146\)」](#) を参照してください。

外部サービスを更新するか、既存の外部サービスをコピーします。[「外部サービス バージョンを保守する \(P. 123\)」](#) を参照してください。

外部サービス バージョンを保守する

基本または高度な外部サービスを既存のバージョンからコピーしたり、サービスを変更したりできます。

外部サービスの定義手順の例は、[ライブラリ] - [外部サービス] ページから表示できます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- 外部サービス バージョンが Application Services で利用できることを確認します。[「ライブラリでの外部サービスの作成 \(P. 118\)」](#) を参照してください。
- 外部サービスのコピーを作成して、オリジナルは後で参照できるように保存し、コピーしたバージョンを編集します。
- 自分が、外部サービス バージョンを所有するビジネス グループのメンバーであることを確認します。
外部サービス バージョンを所有するグループのメンバーでない場合、[コピー] ボタンおよび[編集] ボタンは無効になっています。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [外部サービス] を選択します。

- 2 既存の外部サービス バージョンをコピーします。
 - a 外部サービス バージョンを開き、ツールバーで [コピー] をクリックします。
外部サービスをコピーすると、すべてのプロパティ、オペレーティング システム、説明タグのリスト、サポートされるアプリケーション コンポーネント、プロバイダ仕様スクリプトおよびプロパティが保持されます。既存の外部サービス バージョンの名前は変更できません。
 - b バージョン番号を変更し、説明を更新してから [保存] をクリックします。
バージョン番号と説明によって、新しい外部サービスが親バージョンと区別されます。
- 3 既存のプロパティを変更します。
 - a ツールバーの [編集] をクリックします。
 - b 変更してから [保存] をクリックします。
- 4 プロバイダ仕様を更新します。
 - a プロバイダ仕様バージョンを開きます。
 - b ツールバーの [編集] をクリックします。
 - c 変更してから [保存] をクリックします。
- 5 既存のプロバイダ仕様をコピーします。
 - a プロバイダ仕様バージョンを開きます。
 - b ツールバーの [コピー] をクリックします。
 - c バージョン番号を変更し、説明を更新してから [保存] をクリックします。
バージョン番号と説明によって、新しいプロバイダ仕様が親バージョンと区別されます。

次に進む前に

アプリケーション ブループリントで使用しない古いバージョンの外部サービスを削除します。[「ライブラリからサービスを削除する \(P. 116\)」](#) を参照してください。

外部サービスを削除する

外部サービスを管理するために、重複するサービスや古いサービスを削除できます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- 外部サービスがブループリントで使用されていないことを確認します。[「詳細ブループリントへの外部サービスの追加 \(P. 146\)」](#) を参照してください。
- 自分が、外部サービス バージョンを所有するビジネス グループの一員であることを確認します。
外部サービス バージョンを所有するグループの一員でない場合、[削除] ボタンは無効になります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [外部サービス] を選択します。
- 2 既存の外部サービスを開きます。
- 3 [削除] をクリックして、既存のサービス バージョンを削除して、選択内容を確認します。
- 4 外部サービス名の横にある矢印をクリックして、[外部サービス] ページに戻ります。
- 5 外部サービスを削除して選択内容を確認します。

ライブラリへの論理テンプレートの追加

アプリケーション ブループリントをモデル化するには、互換性のあるサービスとスクリプトをカスタムまたはサンプルの論理テンプレートに追加できます。論理テンプレートは、vCloud Director、vRealize Automation、または Amazon EC2 の実際のクラウドテンプレートにマップできます。論理テンプレートにより、アプリケーション ブループリントはクラウドに依存する必要がなくなります。

論理テンプレートを定義するには、テンプレートにオペレーティング システムと共にインストールされているサービスを記述できます。通常、IT 組織では、いくつかのパフォーマンス監視エージェントとウイルス対策ソフトウェアをテンプレートにインストールします。また、たとえば、展開を高速化するために、vFabric tc Server を論理テンプレートに事前インストールすることもあります。論理テンプレートの展開時に常に特定のサービスを使用する場合は、そのサービスを事前にインストールしておくことで、展開のたびに追加せずに済みます。

論理テンプレートは、パブリックに、またはグループ内でプライベートに共有できます。共有はコンポーネント レベルでサポートされ、個別のバージョン レベルではサポートされません。

変更内容は頻繁に保存してください。Application Services ユーザー インターフェイス セッションは、ユーザー操作のない状態が 30 分間続くと期限切れになります。セッションが期限切れになると、保存していない変更内容はすべて失われます。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- Application Services で適切に機能するための要件を満たすクラウドテンプレートを、vCloud Director、vRealize Automation、または Amazon EC2 に作成します。[「vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)」](#)、[「vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 \(P. 50\)」](#)、または [「Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 \(P. 88\)」](#) を参照してください。
- 少なくとも 1 つのクラウド プロバイダが Application Services に登録されていることを確認します。[「vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)」](#)、[「vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)」](#)、または [「Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)」](#) を参照してください。
- アプリケーションが企業ファイアウォール外から URL にアクセスする必要がある場合は、該当するサービスおよびアプリケーション コンポーネントがプロキシを使用するように構成します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。
- オペレーティング システムおよび説明タグ一覧にアイテムを追加する必要があるかどうかを確認します。[「ライブラリへのオペレーティング システムの追加 \(P. 108\)」](#) および [「ライブラリへのタグの追加 \(P. 109\)」](#) を参照してください。
- アプリケーションまたはサービスをテンプレートに事前インストールする場合は、そのサービスを作成および設定してからテンプレートに追加します。[「ライブラリへのサービスの追加 \(P. 109\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [論理テンプレート] を選択します。
- 2 [新規] をクリックします。
- 3 論理テンプレートの名前を設定し、説明を追加して、[保存] をクリックします。

使用しているクラウドテンプレートやオペレーティング システムを把握するため、説明にはクラウドテンプレートまたはオペレーティング システムの名前を含めます。

- 4 [LT バージョンの作成] をクリックして、論理テンプレート バージョンを作成します。

1 つの論理テンプレートに対して複数のバージョンを作成できます。

論理テンプレートのバージョンを作成するためのページが開きます。

- 5 テンプレートのバージョン情報を入力します。

オプション	説明
名前	論理テンプレートのバージョン名はそのままにします。
バージョン	メジャー バージョンおよびマイナー バージョンを示す修飾子 (1.2.1 や 1.0.1-CentOS56-32bit など) を追加します。
説明	論理テンプレートに関する詳細な情報を追加します。たとえば、使用するスクリプトや必要なディスク領域などを記述します。
タグ	<p>搭載されている機能に基づいて論理テンプレートを分類します。Application Services は、アプリケーションの展開ブループリントの作成時に表示されるタグに基づいてテンプレートを編成します。</p> <p>注意 サーバタイプとして指定されたタグのみがドロップダウン メニューに表示されます。</p> <p>複数のタグを追加できます。</p> <p>一覧に表示されないタグを使用するには [キャンセル] をクリックし、[ライブラリ]-[タグ] を選択してタグを作成します。</p>
サポートされる OS	<p>論理テンプレートにインストールするオペレーティングシステムを指定します。この OS 情報は、アプリケーションブループリントで、このテンプレートに追加できるサービスを制限するのに使用されます。オペレーティングシステムによっては、一部のサービスが実行されません。たとえば、Ubuntu オペレーティングシステムを指定した場合に、ブループリントでこのテンプレートを使用して Ubuntu と互換性のないサービスを追加しようとする、Application Services によって拒否されます。</p> <p>リストにないオペレーティングシステム名を使用するには、[キャンセル] をクリックして [ライブラリ]-[オペレーティングシステム] を選択し、オペレーティングシステム名を作成します。</p>

- 6 クラウド テンプレートを論理テンプレートにマッピングします。

1 つの論理テンプレート バージョンに複数のクラウド テンプレートを追加できます。また、クラウドごとに異なるクラウド テンプレートを選択できます。同じクラウド プロバイダを使用している場合でも、展開時にさまざまなクラウド テンプレートの中から選択して、テンプレート構成を変えることができるようにする必要がある場合もあります。

クラウド テンプレートと論理テンプレートの重複するマッピング (重複行) と空行は保存されません。

たとえば、複数のクラウド テンプレートに対して、同じ論理テンプレートを使用できます。本番環境に展開する場合は、ディスク領域の大きいクラウド テンプレートを選択し、テストまたはステージング環境では、ディスク領域の小さいクラウド テンプレートを選択する、といったことが可能です。

- a [クラウド プロバイダ名] 列で下矢印をクリックして、クラウド プロバイダを選択します。

作成済みのクラウド プロバイダの一覧が表示されます。

- b [クラウド テンプレート] 列で下矢印をクリックして、クラウド テンプレートを指定します。

ユーザーと同じグループに属すクラウド テンプレートがドロップダウン メニューに表示されます。クラウド テンプレートのリストが空の場合、既存のクラウド テンプレートがユーザーのグループに属していないか、クラウド テンプレートが登録されていません。

- 7 (オプション) 複数のクラウド テンプレートを 1 つの論理テンプレートにマッピングするには、[手順 6](#) を繰り返します。

- 8 (オプション) 事前インストール サービスを定義します。
 - a [サービス名] 列で下矢印をクリックして、事前インストール サービスを選択します。
 - b 一覧に表示されないサービスを使用するには、[キャンセル] をクリックし、[ライブラリ] - [サービス] を選択して、サービスおよびそのサービスに含めるアクション スクリプトを作成します。
 ブループリントを作成した後、事前インストール サービスを論理テンプレートに追加しても、追加された事前インストール サービスがノードに追加されることはありません。
- 9 (オプション) 新しい事前インストール サービスをノードに追加します。
 - a 事前インストール サービスを含む論理テンプレートをアプリケーション ブループリントにドラッグします。
 - b サービスとコンポーネントを新しい論理テンプレートに転送します。
 - c 依存関係があれば再構築し、古いテンプレートを削除します。
- 10 テンプレートの作成を終了したら、[保存] をクリックします。

作成した論理テンプレートは、[論理テンプレート] ページに追加されます。また、このテンプレートは、アプリケーションの展開ブループリントを作成するときに取り込むことができる論理テンプレートの一覧にも表示されます。

次に進む前に

既存の論理テンプレートの特定のバージョンをコピーして、そのテンプレートを編集または削除します。[「論理テンプレート バージョンの管理 \(P. 127\)」](#) を参照してください。

論理テンプレート バージョンの管理

既存の論理テンプレートを別のクラウド テンプレートにマッピングするか、あらかじめ組み込まれているサービスを追加するために、論理テンプレート バージョンを編集できます。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- Application Services で適切に機能するための要件を満たすクラウド テンプレートを、vCloud Director、vRealize Automation、または Amazon EC2 に作成します。[「vCloud Director カスタム テンプレートを作成するための仮想マシンの要件 \(P. 72\)」](#)、[「vRealize Automation カスタム テンプレートを作成するための仮想マシンの要件 \(P. 50\)」](#)、[「Amazon EC2 カスタム テンプレートを作成するための仮想マシンの要件 \(P. 88\)」](#) を参照してください。
- アプリケーションが企業ファイアウォール外から URL にアクセスする必要がある場合は、該当するサービスおよびアプリケーション コンポーネントがプロキシを使用するように構成します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。
- オペレーティング システムおよび説明タグ一覧にアイテムを追加する必要があるかどうかを確認します。[「ライブラリへのオペレーティング システムの追加 \(P. 108\)」](#) および [「ライブラリへのタグの追加 \(P. 109\)」](#) を参照してください。
- アプリケーションまたはサービスをテンプレートにあらかじめ組み込んでおくことを計画している場合は、サービスを作成して構成します。[「ライブラリへのサービスの追加 \(P. 109\)」](#) を参照してください。
- 自分が、論理テンプレート バージョンを所有するビジネス グループの一員であることを確認します。
 論理テンプレート バージョンを所有するビジネス グループに属さないユーザーの場合は、[コピー] ボタンと [編集] ボタンが無効になっています。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [論理テンプレート] を選択します。

- 2 既存の論理テンプレート バージョンから論理テンプレートをコピーします。
 - a 論理テンプレート バージョンを開き、ツールバーの [コピー] をクリックします。
クラウド テンプレート、サービス、オペレーティング システムのすべてと、アプリケーション コンポーネントを特定する説明タグのリストが保持されます。既存の論理テンプレート バージョンの名前は変更できません。
 - b バージョン番号を変更し、説明を更新します。必要に応じて、クラウド プロバイダ、クラウド テンプレート、またはサービスを更新します。

バージョン番号と説明によって、新しい論理テンプレートと親バージョンが区別されます。

ベスト プラクティスとして、テンプレートのコピーを作成し、オリジナルを後で参照できるように保存しておきます。
 - c [保存] をクリックします。
- 3 論理テンプレートの既存のクラウド プロバイダ、クラウド テンプレート、またはサービスを変更するには、ツールバーの [編集] をクリックし、変更を加えてから [保存] をクリックします。

次に進む前に

旧バージョンの論理テンプレートがある場合は、ライブラリから削除できます。[「ライブラリからの論理テンプレートの削除 \(P. 128\)」](#) を参照してください。

ライブラリからの論理テンプレートの削除

論理テンプレートは [論理テンプレート] ページから削除できます。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- アプリケーション ブループリントの論理テンプレートおよびそのテンプレートを参照するその他のすべてのオブジェクトが削除されていることを確認します。
- 自分が、論理テンプレート バージョンを所有するビジネス グループの一員であることを確認します。
論理テンプレート バージョンを所有するビジネス グループの一員でない場合、[削除] ボタンは無効になります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [論理テンプレート] を選択します。
- 2 論理テンプレートを開きます。
- 3 [削除] をクリックして既存の論理テンプレート バージョンを削除し、選択内容を確認します。
- 4 論理テンプレート名の横にある矢印をクリックして、論理テンプレートのページに戻ります。
- 5 論理テンプレートを削除して選択内容を確認します。

ライブラリへのポリシーの追加

ポリシー定義を作成し、アプリケーション展開のコンプライアンス状態を評価するルールのカスタム セットをキャプチャできます。

Application Services には、次の事前定義済みポリシー定義が含まれます。

- ブラックリスト サービス - ブループリント ノードまたは論理テンプレートで不要なサービスが使用されているかどうかを確認します。
- メモリ ポリシー - ノード メモリ値を定義されている最小メモリ値および最大メモリ値と比較します。

- CPU ポリシー - ノード CPU カウント値を指定されている最小メモリ値および最大メモリ値と比較します。
- 最大 VM カウント ポリシー - 展開のすべてのノードにおける仮想マシン インスタンスの総数を確認します (外部サービスのための一時的な仮想マシンを除く)。
- 必須サービス - ブループリント ノードに指定された必須サービスが欠けているかどうかを確認します。

ポリシーは、自身のテナントの全ビジネス グループに表示されますが、他のテナントでは表示されません。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。第 12 章「[Application Services コンポーネントの開発 \(P. 93\)](#)」を参照してください。
- ポリシーの命名や定義の方法の例を表示するには、[ライブラリ] - [ポリシー] を選択して事前定義されたポリシーを表示します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [ポリシー] を選択します。
- 2 [新規] をクリックします。
- 3 ポリシーの名前を設定し、説明を追加してから [保存] をクリックします。
追跡目的には、ポリシー定義に関連付けられている名前を使用します。たとえば、展開中に特定のサービスがアプリケーションに常に適用されるかどうかをポリシーが確認する場合、説明にそのサービス名を記載できます。
- 4 [ポリシー バージョンの作成] をクリックしてポリシー バージョンを作成します。
ポリシーには、複数のバージョンを作成できます。
ポリシー バージョンを作成するためのページが開きます。
- 5 ポリシー バージョンの情報の入力を完了します。
ポリシー バージョンの名前は変わりません。
 - a メジャー リリース、マイナー リリース、またはマイクロ リリースを修飾子付き/なしで指定します。
たとえば、1.0、1.5、1.0.1-CPU-Policy などのバージョン番号を使用します。
 - b (オプション) [説明] セクションで、ポリシーの定義を説明します。ポリシーが前のバージョンに基づいている場合、2 つのポリシー バージョンの違いの詳細を説明します。
- 6 [プロパティ名] と [説明] では、変数を定義してオプションの説明を追加します。
プロパティ名の先頭に数字を指定することはできません。
- 7 タイプを変更するには、ドロップダウン メニューから文字列またはアレイのプロパティタイプを選択します。
メニューにタイプを追加することはできません。
- 8 スクリプトの実行時に、このプロパティを置き換える値を入力します。
たとえば、max_cpu_count というプロパティでは、このフィールドに「**10**」と入力し、アプリケーションで許可される CPU の最大数を指定します。
- 9 定義するパスワードまたはその他のプロパティの値を見えなくする場合は、[セキュアド] チェック ボックスをオンにします。
プロパティが [セキュアド] から [アンセキュアド] に変更されると、Application Services は、セキュリティ上の理由によってプロパティ値をリセットします。プロパティに新しいパスワード値を設定する必要があります。
- 10 [プロパティ] セクションから選択された行を削除するには、[削除] をクリックします。

11 (オプション) 元のプロパティ値に戻すには、[リセット] をクリックします。

12 作業が終了したら、[OK] をクリックして変更内容を保存します。

作成したポリシーがページに追加されます。

次に進む前に

ポリシー定義のスクリプトを追加し、展開のコンプライアンス状態を評価します。[「ポリシー定義スクリプトの作成 \(P. 130\)」](#) を参照してください。

ポリシー定義スクリプトの作成

各ポリシー定義には、展開のコンプライアンス状態を評価するための、ライフ サイクル ステージの SCAN スクリプトが含まれています。Application Services は、操作を展開で実行する前に（分解プロセスは除く）、またはユーザーが明示的に展開へのポリシー スキャンを開始したときに、ポリシー スキャン アクション スクリプトを呼び出します。

スキャン アクション スクリプトには、Application Services の REST API 仕様で定義されている展開のモデルが含まれます。また、スキャン アクション スクリプトは展開のブループリントで使用される追加コンポーネントを受け取ります。

特定の展開環境でポリシー インスタンスを作成して、ポリシーを有効にする必要があります。展開中にポリシー違反が発生すると、ポリシーにはフラグが付けられ、ユーザーはコンプライアンス ビューのサマリ ページで違反の詳細を表示できます。

注意 Java Script は、ポリシー定義スクリプトを記述するために唯一サポートされている言語です。

ポリシー定義で定義されるポリシー プロパティは、個々の変数としてスクリプトに提供されます。プロパティ名と同じ名前の変数を宣言すると、スクリプトはその変数にアクセスできます。

スクリプト入力変数	説明
var min_cpu_count	min_cpu_count プロパティに対応し、スクリプトの値は consume に設定されます。
eventPayload	<p>ポリシー コンプライアンスについて評価された展開の詳細情報が含まれます。</p> <p>通常のプロパティの場合、eventPayload オブジェクトにアクセスするには、var eventPayload を追加する必要があります。</p> <p>ポリシー スクリプトは、eventPayload 変数を、次のプロパティを持つ Java オブジェクトと見なすことがあります。</p> <div> <div> V2 API で定義されている、DeploymentProfile タイプの deploymentProfile </div> <div> 展開の最新情報を取得する展開プロファイルを示します。更新の場合、更新プロファイルに属する変更のすべてが含まれます。 </div> </div> <div> <div> V2 API で定義されている、ブループリントタイプのブループリント </div> <div> deploymentProfile から参照される実際のブループリント オブジェクトを示します。 </div> </div> <div> <div> LogicalTemplate が V2 API で定義されている、List<LogicalTemplate> タイプの logicalTemplates </div> <div> アプリケーション ブループリント内のさまざまなノードから参照される論理テンプレートのリストを示します。 </div> </div> <div> <div> ServiceVersion が V2 API で定義されている、List<ServiceVersion> タイプの serviceVersions </div> <div> アプリケーション ブループリント内のさまざまなノードから参照されるサービスバージョンのリストを示します。 </div> </div>

ポリシー スクリプトは、コンプライアンス評価の結果を通知するために、次のプロパティを出力することになります。スクリプトでは、そのプロパティを変数として宣言する必要があります。

スクリプト出力変数	説明
complianceResult	<p>必須の文字列タイプ。スクリプトがこの設定に失敗すると、コンプライアンス結果はエラーと見なされます。変数の有効な値は次のとおりです。</p> <p>準拠 評価されるポリシーに展開が準拠していることを示します。</p> <p>Non_Compliant 評価されるポリシーに展開が違反していることを示します。</p> <p>エラー 評価結果の生成時にエラーが発生したことを示します。</p>
complianceMessage	<p>文字列のタイプ。</p> <p>この任意の値は、ポリシー違反の理由の大きな概要を示します。値は、2048 文字未満の文字列で設定できます。</p>

スクリプトは、Java Script で使用できる標準の `println` 関数でログ メッセージを作成できます。ログは Application Services によってキャプチャされ、ポリシー スクリプトのエラーの診断や、ポリシー評価結果の詳細情報の確認に役立ちます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- ライブラリ内に少なくとも 1 つのポリシーが作成されていることを確認します。[「ライブラリへのポリシーの追加 \(P. 128\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ]-[ポリシー] を選択します。
- 2 ポリシーを開き、ポリシー定義スクリプトを追加します。
- 3 [スクリプト] 列で、ハイパーリンクをクリックして [スクリプトの編集] ダイアログ ボックスを開きます。
既存の事前定義済みポリシー定義を参照して、ダイアログ ボックスでスクリプトを作成できます。
- 4 [OK] をクリックします。

次に進む前に

展開環境でポリシー インスタンスを作成して、展開環境下のすべての展開でそのポリシー定義を有効にしない限り、ポリシー定義を指定しても展開に影響しません。[「ポリシー インスタンスの作成 \(P. 62\)」](#) を参照してください。

ポリシー バージョンの維持

ポリシーを編集すると、ポリシー定義を追加または削除し、アクション スクリプトを編集し、またはスキャン パラメータを変更できます。ポリシー定義を編集しても、それから作成されている既存のポリシー インスタンスには影響しません。変更は、新しいポリシー インスタンスにのみ実装されます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- ポリシー定義およびそのアクションの例を表示するには、[ライブラリ]-[ポリシー] を選択して事前定義されたポリシーを表示します。
- Application Services ライブラリで、ポリシー バージョンが使用できることを確認します。[「ライブラリへのポリシーの追加 \(P. 128\)」](#) を参照してください。

- 今後の参照のために元のポリシーを保存するために、ポリシーのコピーを作成し、コピーされたバージョンを編集します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [ポリシー] を選択します。
- 2 既存のポリシー バージョンをコピーします。
 - a ポリシー バージョンを開き、ツールバーの [コピー] をクリックします。
 ポリシーをコピーする際には、アクション スクリプトとすべてのプロパティが維持されます。既存のポリシー バージョンの名前を変更することはできません。
 - b バージョン番号を変更し、説明を更新してから [保存] をクリックします。
 バージョン番号と説明により、新しいポリシーを親バージョンから区別します。
- 3 既存のポリシーのプロパティとアクション スクリプトを変更します。
 - a ツールバーの [編集] をクリックします。
 - b 変更してから [保存] をクリックします。

次に進む前に

重複しているポリシーまたは古いポリシーを削除します。[「ポリシーの削除 \(P. 132\)」](#) を参照してください。

ポリシーの削除

アプリケーション 展開に適用できないポリシー定義がある場合は、そのポリシーを Application Services ライブラリから削除できます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- 展開環境に削除対象のポリシー インスタンスが存在していないことを確認します。[「vCloud Director 展開環境の作成 \(P. 81\)」](#)、[「vRealize Automation 展開環境の作成 \(P. 60\)」](#)、または [「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [ポリシー] を選択します。
- 2 既存のポリシーを開きます。
- 3 既存のポリシー バージョンの削除を実行し、確認メッセージが表示されたら確定します。
- 4 ポリシー名の横の矢印をクリックして、[ポリシー] ページに戻ります。
- 5 ポリシーの削除を実行し、確認メッセージが表示されたら確定します。

カスタム タスクをライブラリに追加する

Application Services を使用すると、カスタム タスクを作成して、セキュリティ パッチの実行などのカスタマイズ済みタスクをアプリケーション展開で実行することができます。

Application Services カタログには、APT または YUM リポジトリを構成する事前定義済みタスク、マシンを Red Hat ネットワークに登録するスクリプト、または仮想マシンが指定の Windows Active Directory ドメインに接続できるようにするスクリプトが含まれています。これらのカスタマイズされたタスクを、展開プロファイルの実行プランに追加できます。場合によっては、カスタム タスクを作成するのではなく、事前定義済みタスクを編集することもできます。

変更内容は頻繁に保存してください。Application Services ユーザー インターフェイス セッションは、ユーザー操作のない状態が 30 分間続くと期限切れになります。セッションが期限切れになると、保存していない変更内容はすべて失われます。

開始する前に

- **アプリケーション クラウド管理者**や**アプリケーションの公開者**および**展開者**として Application Services にログインします。
- オペレーティング システムおよび説明タグ一覧にアイテムを追加する必要があるかどうかを確認します。[「ライブラリへのオペレーティング システムの追加 \(P. 108\)」](#) および [「ライブラリへのタグの追加 \(P. 109\)」](#) を参照してください。
- タスク内のスクリプトが企業ファイアウォール外から URL にアクセスする必要がある場合は、該当するサービスおよびアプリケーション コンポーネントがプロキシを使用するように構成します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ]-[タスク] を選択します。
- 2 [新規] をクリックします。
- 3 カスタム タスクの名前を設定し、説明を追加して、[保存] をクリックします。
ベスト プラクティスとして、スクリプトが実行するタスクに応じた名前を使用します。
- 4 [タスク バージョンの作成] をクリックして、カスタム タスク バージョンを作成します。
1 つのカスタム タスクに複数のバージョンを作成できます。
カスタム タスク バージョンを作成するためのが表示されます。
- 5 カスタム タスク バージョンの情報を入力します。

オプション	説明
名前	タスク バージョン名は同じままにします。
説明	カスタム タスクに関する詳細情報を追加します。たとえば、ノードに追加されたときにスクリプトが何を実行するかを実行プランに記入します。
バージョン	メジャー リリース、マイナー リリース、またはマイクロ リリースを修飾子付き/なしで指定できます。たとえば、1.0、1.5 などの一意のバージョン番号を使用できます。
サポートされる OS	このタスクで使用されるスクリプトが特定のオペレーティング システムでしか実行できない場合、ここでそれらのオペレーティング システムを選択します。実行プランでは、リストにこれらのオペレーティング システムのいずれかが含まれていないと、Application Services で [カタログ タスク名] リストにカスタム タスクを表示できません。少なくとも 1 つのオペレーティング システムをテキスト ボックスに追加する必要があります。 リストに含まれていないオペレーティング システム名を作成するには、[キャンセル] をクリックし、[ライブラリ]-[オペレーティング システム] を選択します。

- 6 [プロパティ] セクションで、カスタム タスクで使用される変数を定義します。

オプション	説明
プロパティ名	[プロパティ名] 列の最初の行をクリックして、変数を定義します。たとえば、リポジトリを構成するカスタム タスクや、サービスまたはアプリケーション コンポーネントの展開タスクが正常に終了したときに通知電子メールを送信するカスタム タスクを作成できます。
説明	カスタム タスクに関する詳細情報を入力します。

オプション	説明
ファイルタイプ	タイプを変更するには、ドロップダウン メニューからタイプを選択します。利用可能なプロパティのタイプは、文字列、コンテンツ、およびアレイです。メニューにタイプを追加することはできません。
値	カスタム タスクの実行時に、このプロパティを置き換える値を入力します。たとえば、remove_all というプロパティの場合は、このテキスト ボックスに true または false を入力できます。
セキュア	定義したパスワードのチェック ボックスをオンにするか、値を明示しないその他のプロパティのチェック ボックスをオンにします。たとえば、JBoss サービスの JBOSS_JMX_PWD プロパティはセキュアになっています。 プロパティが [セキュア] から [アンセキュア] に変更されると、Application Services は、セキュリティ上の理由によってプロパティ値をリセットします。プロパティの新しい値を設定する必要があります。
[削除] ボタン	[プロパティ] セクションから、選択された行を削除します。

- 7 [スクリプト エディタ] セクションで、カスタム タスクのスクリプトを追加します。

カタログ管理者は、サービスのインストールと構成をパラメータ化できます。サービス用に定義されたプロパティはスクリプト内で使用できます。

- [スクリプトの編集] ダイアログ ボックスを展開し、ドロップダウン メニューからスクリプト タイプを設定して、スクリプトを記述するかコピーします。
- アプリケーションの展開中にスクリプトが正常に実行されたら仮想マシンを再起動するために、[再起動] チェック ボックスをオンにします。
- [挿入するプロパティを選択] リストの下矢印をクリックし、カスタム タスク プロパティを追加します。
- 終了したら [OK] をクリックします。

- 8 カスタム タスクの作成が終了したら、[保存] をクリックします。

作成したカスタムが [タスク] ページに追加されます。

次に進む前に

カスタム タスクは実行プランに追加し、展開環境に展開することができます。[「実行プランを確認しカスタム タスクを追加する \(P. 162\)」](#) を参照してください。現在のニーズを満たすようにカスタム タスクを変更します。[「カスタム タスクバージョンを保守する \(P. 134\)」](#) を参照してください。

カスタム タスク バージョンを保守する

スクリプトを構成したり、カスタム タスク バージョンの既存のプロパティを再定義したりできます。オリジナルのカスタム タスク バージョンを保持するために、コピーを作成してコピーしたバージョンを変更できます。

注意 アプリケーション展開をカスタマイズするために展開プロファイルでカスタム タスクが使用されている場合、カスタム タスクを削除できません。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- オペレーティング システムおよび説明タグ一覧にアイテムを追加する必要があるかどうかを確認します。[「ライブラリへのオペレーティング システムの追加 \(P. 108\)」](#) および [「ライブラリへのタグの追加 \(P. 109\)」](#) を参照してください。
- タスク内のスクリプトが企業ファイアウォール外から URL にアクセスする必要がある場合は、該当するサービスおよびアプリケーション コンポーネントがプロキシを使用するように構成します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。

- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。第 12 章「[Application Services コンポーネントの開発 \(P. 93\)](#)」を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [タスク] を選択します。
- 2 既存のカスタム タスク バージョンからカスタム タスクをコピーします。
 - a カスタム タスク バージョンを開いて、ツールバーで [コピー] をクリックします。

すべてのプロパティ、スクリプト、オペレーティングシステムが保持されます。既存のカスタム タスク バージョンの名前は変更できません。
 - b バージョン番号を変更し、説明を更新して、必要に応じてオペレーティングシステムを更新します。

バージョン番号と説明によって、新しいカスタム タスクが親バージョンと区別されます。
 - c [保存] をクリックします。
- 3 既存のプロパティまたは既存のスクリプトを変更するには、ツールバーで [編集] をクリックし、変更を加えてから [保存] をクリックします。

アプリケーションの作成

Application Services には、アプリケーション アーキテクトがアプリケーション ブループリントをモデル化するために使用できるドラッグ アンド ドロップ キャンバスがあります。このブループリントを使用して、クラウドで展開するためのアプリケーションを作成できます。

次の項目をブループリントにドラッグできます。

- ライブラリからの論理テンプレート

Application Services サンプル テンプレートを使用するか、カスタム テンプレートを作成できます。オペレーティング システム イメージを含み、サービスがインストールおよび構成済みになっている可能性があるクラウド テンプレートが Application Services 論理テンプレートにマッピングされます。

- ライブラリからのアプリケーション インフラストラクチャ コンポーネントおよびスクリプト

付属のサンプル サービスは再使用可能なソフトウェア コンポーネントです。Application Services には、サービスのベスト プラクティスに従ったインストール スクリプトと構成スクリプトが含まれています。

- アプリケーション コンポーネント

これらのアプリケーションはサービスの上で動作します。アプリケーション コンポーネントを構成した後で、互換性のあるサービスまたはオペレーティング システム イメージに展開できます。

共有

Application Services は次の理由により、vRealize Automation に依存しています。

ログイン済み Application Services ユーザーは 1 つ以上のビジネス グループに属します。1 つ以上のビジネス グループは単一のテナントに属します。1 つ以上のテナント

注意 Application Services からのアプリケーション展開の削除については、[「Application Services からアプリケーション展開を削除する \(P. 217\)」](#)を参照してください。

アプリケーションの作成に関するトピックで頻出する主要概念について理解しておく必要があります。[「主要な概念 \(P. 16\)」](#)を参照してください。

この章では次のトピックについて説明します。

- [アプリケーションの作成 \(P. 138\)](#)
- [アプリケーション バージョンの作成 \(P. 138\)](#)
- [アプリケーション ブループリントのモデル化 \(P. 139\)](#)
- [アプリケーション ブループリントを構成する \(P. 141\)](#)
- [詳細ブループリントの作成 \(P. 142\)](#)
- [アプリケーション バージョンをコピーする \(P. 146\)](#)

- [アプリケーションのコピー \(P. 147\)](#)
- [アプリケーション バージョンの削除 \(P. 147\)](#)

アプリケーションの作成


Application Services を使用すると、アプリケーション展開のモデル化、依存関係の作成、およびアプリケーション構成の編集を実行できます。

アプリケーションをテナント全体で共有するように指定することも、アプリケーションを所有するビジネス グループ専用指定することもできます。アプリケーションを共有する場合は、論理テンプレート、サービス、外部サービスなどのアプリケーション ブループリント内のすべてのコンポーネントを明示的に共有する必要があります。アプリケーションに関連付けられている展開プロファイルは共有されません。それらは常に、アプリケーションを所有するビジネス グループ専用です。

開始する前に

- **アプリケーション アーキテクト**として Application Services にログインします。
- アプリケーションの命名、および説明のテキスト ボックスに入力するテキストの内容については、[アプリケーション] ページに既に含まれている事前定義されたアプリケーションを参照してください。これらのアプリケーションを表示するには、Default ビジネス グループのメンバーであるユーザーとしてログインする必要があります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 [ アプリケーションの新規作成] をクリックします。
- 3 [名前] テキスト ボックスにアプリケーションの名前を入力します。
- 4 (オプション) [説明] テキスト ボックスに、アプリケーションの説明を入力します。
- 5 [ビジネス グループ] ドロップダウン メニューで、アプリケーションを所有するビジネス グループを選択します。
- 6 [共有] ドロップダウン メニューで、アプリケーションの共有状態を選択します。
プライベート アプリケーションを所有するビジネス グループのメンバーのみが、プライベート アプリケーションにアクセスできます。テナントのビジネス グループすべてのメンバーは、共有アプリケーションにアクセスできます。
- 7 [保存] をクリックします。


アプリケーション バージョンの作成

アプリケーションには、複数のアプリケーション バージョンを作成できます。アプリケーション バージョンを作成する場合は、アプリケーション名は同じですが、ブループリントのコンテンツは既存のアプリケーションからコピーされません。新しいアプリケーション バージョンのブループリントを作成する必要があります。

開始する前に

- **アプリケーション アーキテクト**として Application Services にログインします。
- アプリケーションの命名、および説明のテキスト ボックスに入力するテキストの内容については、[アプリケーション] ページに既に含まれている事前定義されたアプリケーションを参照してください。これらのアプリケーションを表示するには、Default グループのメンバーとしてログインする必要があります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションを開きます。
- 3 [ アプリケーション バージョンの作成] をクリックします。

- 4 [バージョン] テキスト ボックスにアプリケーションのメジャー バージョンおよびマイナー バージョンを入力します。
たとえば、**1.0.0-SNAPSHOT** のように、メジャーおよびマイナー バージョンに修飾子を追加する場合があります。
- 5 (オプション) [説明] テキスト ボックスに、アプリケーション バージョンの説明を入力します。
- 6 [保存] をクリックします。
元のアプリケーションの [アプリケーション バージョン] セクションに、アプリケーション バージョンが表示されます。

次に進む前に

アプリケーション バージョンのアプリケーション ブループリントをモデル化します。[「アプリケーション ブループリントのモデル化 \(P. 139\)」](#) を参照してください。

アプリケーション ブループリントのモデル化

アプリケーション ブループリントでは、インストールの依存関係、構成の変更、編集可能な Application Services スクリプトを詳細に制御できます。Application Services は、ブループリントから実行プランを生成します。これは改訂可能で、サポートされているクラウド環境へのアプリケーションの展開に使用します。

アプリケーションのモデル化と作成は、ブループリントキャンバスで実行します。ブループリント キャンバスの左側には、Application Services ライブラリに登録されている論理テンプレートが表示されます。右側には、ライブラリとアプリケーション コンポーネントの利用可能なサービスが一覧表示されます。論理テンプレートを選択し、ブループリント キャンバスの中央にドラッグすると、アプリケーションのモデル化を開始できます。

変更内容は頻繁に保存してください。Application Services ユーザー インターフェイス セッションは、ユーザー操作のない状態が 30 分間続くと期限切れになります。セッションが期限切れになると、保存していない変更内容はすべて失われます。

開始する前に

- アプリケーション アーキテクトとして Application Services にログインします。
- Application Services に少なくとも 1 つのアプリケーション バージョンが作成されていることを確認します。[「アプリケーション バージョンの作成 \(P. 138\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 既存のアプリケーションを選択し、特定のバージョンのアプリケーションを開きます。
- 3 [ブループリントの作成] をクリックして、アプリケーション ブループリントを作成します。
アプリケーション ブループリントをモデル化するためのキャンバスが表示されます。
- 4 1 つ以上の論理テンプレートを選択し、キャンバスにドラッグしてノードを作成します。
たとえば、3 層アプリケーションを作成するとします。この場合、[OS テンプレート] リストから 3 つの項目をキャンバスにドラッグするか、[OS テンプレート] リスト、[データベース サーバ] リスト、および [アプリケーション サーバ] リストからそれぞれ 1 つずつ項目を選択してキャンバスにドラッグします。各リストの名前は、テンプレートに関連付けられているタグに対応しています。
- 5 (オプション) キャンバス上でノードを 1 つ選択し、ブループリントの下の [詳細] タブでノードの名前を変更します。
ノード名に、文字、数字、アンダースコア、またはハイフン以外の文字が含まれている場合、その文字がハイフンで置換されている。
たとえば、3 層アプリケーションの場合は、3 つのノードの名前をそれぞれ、Application_Server、Database_Server、Load_Balancer などとする必要があります。

- 6 (オプション) アプリケーションを vCloud Director または vCloud Automation Center に展開する場合は、ブループリントの下 [詳細] タブでホスト名を設定します。

このホスト名は、ネットワークで仮想マシン コンピュータの識別子として使用されます。[ホスト名] テキスト ボックスを空のままにすると、Application Services によってランダムな文字を使用したホスト名が生成されます。

ホスト名は次の命名規則に従っている必要があります。

- 15 文字以下である。
- 文字で始まる。
- 文字、数字、またはハイフンを使用できるが、末尾にはハイフンを使用できない。
- アプリケーション ブループリント内の別のノードと同じホスト名は使用できない。
- 一意の文字列を生成するために、ホスト名の末尾に `${random}` という式を付加できる。

たとえば、`Apache${random}` というホスト名を指定すると、アプリケーションが展開された後、`Apache9INOIK3YT` のような文字列が生成されます。一意の文字列を含むホスト名は、[展開サマリ] ページで確認できます。

クラスタ化されたノードのホスト名は、末尾のノード アレイ インデックスも含めて 15 文字以内にすることがあります。たとえば、`AppServer` というクラスタ化されたノードでは、クラスタ内の各仮想マシンのホスト名が、`AppServer-1`、`AppServer-2` などとなります。

- 7 (オプション) 展開済み仮想マシンのデフォルトの CPU 数またはメモリ容量を変更するには、キャンパス上で変更対象のノードを選択し、ブループリントの下 [詳細] タブで値を編集します。

ブループリントの下に表示されるタブは、選択されたノードに対応しています。

ブループリントの CPU とメモリの値は、Amazon EC2 で作成される仮想マシンでの対応する値とは一致しないことがあります。Amazon EC2 では、CPU とメモリの特定の組み合わせを持つ仮想マシンのみを許可しています。このため、ユーザーがブループリントで割り当てた値を使用して、使用可能な最も近い組み合わせが決定されます。

- 8 1 つ以上のサービスまたはアプリケーション コンポーネントを選択してノードにドラッグします。

たとえば、MySQL サービスをデータベース サーバ ノード上にドラッグする、JBoss サービスをアプリケーション サーバ ノード上にドラッグする、Apache サービスを Load Balancer ノード上にドラッグする、といった具合です。

特定のノードに、そのノードと互換性のないサービスまたはアプリケーション コンポーネントをドロップすることはできません。たとえば、SQL SCRIPT というアプリケーション コンポーネントは、MySQL サービスにドラッグすることはできませんが、JBoss サービスにドラッグすることはできません。

こうした互換性制限は、アプリケーション カタログ管理者が、ライブラリ サービスでサポートされるオペレーティング システムとコンポーネントを設定することによって生じます。アプリケーション カタログ管理者は、ライブラリで使用可能なオペレーティング システムとタグのリストにエンTRIES を追加することもできます。たとえば、ライブラリの MySQL サービスには、サポートされているコンポーネントとして SQL SCRIPT が登録されています。MySQL サービスに追加できるのは、SQL SCRIPT アプリケーション コンポーネント タイプのみです。

また、MySQL サービスには、サポートされているオペレーティングシステムとして、CentOS32 6.3.0、CentOS64 6.3.0、CentOS32 6.4.0、CentOS64 6.4.0、および Ubuntu32 12.4.2 が設定されています。MySQL サービスは、これらのうちのいずれかのオペレーティングシステムを含む論理テンプレートに追加できます。

注意 ブループリントを作成した後に事前インストール済みサービスを論理テンプレートに追加しても、その新しい事前インストール済みサービスはノードに追加されません。その場合は、ノードを再作成して、事前インストール済みサービスを追加する必要があります。

SCRIPT と Other の 2 つのアプリケーション コンポーネントは、ノードまたは任意のサービスに追加できます。

- 9 [保存] をクリックします。

次に進む前に

アプリケーション ブループリントに追加したサービスおよびアプリケーション コンポーネントを構成します。[「アプリケーション ブループリントを構成する \(P. 141\)」](#) を参照してください。

アプリケーション ブループリントを構成する

サービスおよびアプリケーション コンポーネントのプロパティとアクション スクリプトを構成して、アプリケーションの展開をカスタマイズできます。

開始する前に

- **アプリケーション アーキテクト**として Application Services にログインします。
- アプリケーションが企業ファイアウォール外から URL にアクセスする必要がある場合は、該当するサービスおよびアプリケーション コンポーネントがプロキシを使用するように構成します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。
- アプリケーションのカスタマイズを計画している場合は、別のプロパティへのバインドの基本概念を理解しておく必要があります。[「ほかのプロパティへのバインド \(P. 100\)」](#) を参照してください。
- アプリケーション ブループリントに、サービスおよびアプリケーション コンポーネントが存在することを確認します。[「アプリケーション ブループリントのモデル化 \(P. 139\)」](#) を参照してください。

手順

- 1 サービスまたはアプリケーション コンポーネントを選択し、ブループリントの下にある [詳細] タブおよび[アクション] タブで情報を編集します。

アプリケーション ブループリントで変更できるのは、アプリケーション カタログ管理者がオーバーライド可能として指定したプロパティのみです。

[アクション] タブでは、インストール、構成、開始、更新、ロールバック、分解を含む、コンポーネントのライフサイクルの全ステージでスクリプトにアクセスできます。アプリケーション カタログ管理者は、[ライブラリ]-[サービス]でサービス スクリプトを編集できます。

- 2 プロパティを編集するには、テーブル内のプロパティをクリックします。

シングル ノードまたはクラスタ化ノードを使用している場合は、プロパティの少なくとも 1 つのライフサイクル ステージで必要とされるプロパティ値を指定する必要があります。たとえば、Apache Tomcat サーバを実行するには Java が必要で、JAVA_HOME プロパティ値を設定する必要があります。

[プロパティの編集] ダイアログ ボックスが表示されます。

- 3 (オプション) プロパティまたは成果物を別のプロパティにバインドするには、[プロパティの編集] ダイアログ ボックスの [ブループリント値] ドロップダウン メニューからプロパティ値または成果物の名前を選択します。

別のプロパティにバインドすると、ほかのノードのランタイム プロパティの値 (IP アドレスなど) に基づいてスクリプトをカスタマイズできます。成果物にバインドすると、物理的な場所を意識せずにコンポーネントを使用できます。

- 4 アプリケーションのブループリントの作成が終了したら、[保存] をクリックします。

Application Services により、作成したアプリケーション トポロジが確認され、エラーの一覧を記載したメッセージ ボックスが表示されます。たとえば、選択したプロパティ タイプがスクリプト タイプと互換性がない場合や、サービスまたはコンポーネントに必要なスクリプトが含まれていない場合に、メッセージが表示されます。一部のエラーは、アプリケーションを保存する前に修正する必要があります。

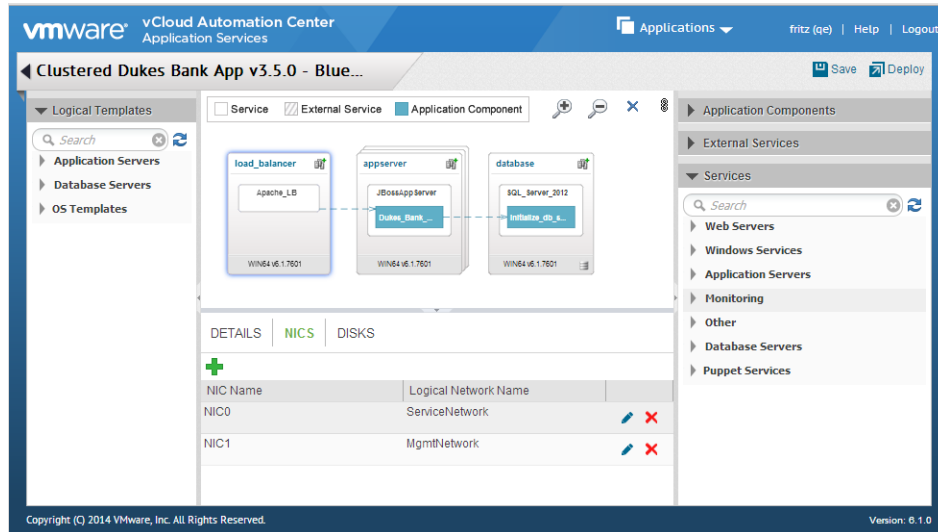
次に進む前に

展開プロファイルを作成します。[「展開プロファイルの作成 \(P. 157\)」](#) を参照してください。

詳細ブループリントの作成

Application Services では、コンポーネント、クラスタ化ノード、複数のネットワーク間の依存関係を含む高度なアプリケーション ブループリントを作成できます。

図 14-1. クラスタ化された Windows アプリケーションの例



- [コンポーネント間の依存関係の作成](#) (P. 143)

展開タスクを実行する必要がある順序を定義するために、依存関係がブループリントに追加されます。サービスまたはアプリケーション コンポーネントなどの 1 つのアイテムから別のサービスまたはアプリケーション コンポーネントへの依存関係リンクを作成すると、最初のアイテムの作成タスクが正常に終了してから 2 番目のタスクが開始されるようになります。

- [クラスタとしてノードを指定する](#) (P. 143)

展開のスケール調整を行う場合は、特定のノードに対して複数の仮想マシンまたは 1 つのクラスタを展開し、ロード バランサを使用してそれらを管理する必要があります。

- [ノード用に複数の NIC を定義する](#) (P. 144)

ほとんどの展開では、サーバは DMZ ゾーンに展開されるか、ファイアウォールによって保護されたネットワークに展開されます。

- [フレキシブル ディスク レイアウトの追加と管理](#) (P. 145)

フレキシブル ディスク レイアウトにより、ストレージの柔軟性を高め、ノードにディスクを追加できます。アプリケーション ブループリントを作成する際に、ディスクを追加することもできます。ディスクは、プロビジョニング中に動的に作成されてノードに追加されます。


- [詳細ブループリントへの外部サービスの追加](#) (P. 146)

アプリケーションでの処理に外部サービスが必要な場合は、外部の基本サービスまたはアドバンスドサービスをブループリントに追加できます。

コンポーネント間の依存関係の作成

展開タスクを実行する必要がある順序を定義するために、依存関係がブループリントに追加されます。サービスまたはアプリケーション コンポーネントなどの 1 つのアイテムから別のサービスまたはアプリケーション コンポーネントへの依存関係リンクを作成すると、最初のアイテムの作成タスクが正常に終了してから 2 番目のタスクが開始されるようになります。

手順

- 1 サービスまたはアプリケーション コンポーネント間の依存関係を作成するには、キャンバス上部のツールバーで [依存関係の追加] () をクリックします。
- 2 最初のコンポーネントを選択し、次にそのコンポーネントが依存するコンポーネントを選択します。
たとえば、ロード バランサは通常アプリケーションを起動して実行するまで構成できないため、Apache サービスから WAR コンポーネントへの依存関係を追加することができます。
青い点線が表示され、依存するコンポーネントをポイントします。
- 3 終了したら、キャンバスの上にあるツールバーの [保存] をクリックします。

例: JBoss から MySQL への依存関係を作成する

たとえば、JBoss サービスから MySQL サービスへの依存関係ラインを作成することもできます。

この表示される青いラインは、アプリケーション サーバ JBoss が MySQL データベースを持っていることに依存し、作成および構成されていることを示します。

次に進む前に

アプリケーションを展開します。第 16 章「[アプリケーションの展開 \(P. 157\)](#)」を参照してください。


クラスタとしてノードを指定する

展開のスケール調整を行う場合は、特定のノードに対して複数の仮想マシンまたは 1 つのクラスタを展開し、ロード バランサを使用してそれらを管理する必要があります。

開始する前に

別のプロパティ、ノード アレイ インデックス プロパティへのバインド、およびコンポーネント アクションの定義に関する基本概念を理解しておく必要があります。第 12 章「[Application Services コンポーネントの開発 \(P. 93\)](#)」を参照してください。

手順

- 1 仮想マシンのクラスタを指定するには、ノードの [ノード アレイへの変換] アイコン () をクリックします。
- 2 ブループリントの下部で、クラスタ サイズを設定します。
- 3 `node_array_index` などのプロパティにバインドし、現在のスクリプトが実行されている仮想マシンを特定します。
プロパティを `all(node:ip)` にバインドすると、クラスタ内のすべての仮想マシンの IP アドレスを確認できます。
- 4 (オプション) 別のプロパティがクラスタ プロパティを参照する場合は、クラスタ化されたノード内のプロパティ値のアレイにアクセスするように、コンポーネント プロパティを定義します。
- 5 キャンバス上部のツールバーで、[保存] をクリックします。

例: ロード バランサの指定

クラスタの使用方法を理解するには、Clustered Dukes Bank サンプル アプリケーションを確認してください。ロード バランサ ノードで [Apache_LB] をクリックし、http_node_ips プロパティが all(appserver:ip) を参照していることに留意します。https_node_ips プロパティでは、指定されたクラスタ サイズ内のノードごとに、すべての IP アドレスを参照します。

次に進む前に

アプリケーションを展開します。第 16 章「アプリケーションの展開 (P. 157)」を参照してください。

ノード用に複数の NIC を定義する

ほとんどの展開では、サーバは DMZ ゾーンに展開されるか、ファイアウォールによって保護されたネットワークに展開されます。

Clustered Duke's Bank サンプル アプリケーションでは、Load Balancer ノードがパブリック ネットワークからアクセスする必要のある唯一のノードです。Database ノードと Appserver ノードは、ファイアウォールで保護されたプライベート ネットワークに展開する必要があります。Load Balancer ノードも Database ノードと AppServer ノードにアクセスする必要があります。


Application Services では、この状況を解決するために、ロード バランサーで 2 つの NIC を定義できます。各 NIC で論理ネットワーク名を指定する必要があります。展開時に、論理ネットワークは実際のクラウド ネットワークにマッピングされます。仮想マシンを作成するときに、仮想マシンの NIC の数がこのノードから派生されます。

Dukes Bank サンプル アプリケーションでは Load Balancer ノードに 2 つの NIC があります。NIC0 は ServiceNetwork をポイントし、NIC1 は MgmtNetwork をポイントします。Database ノードと Appserver ノードには ServiceNetwork をポイントする 1 つの NIC があります。展開時に ServiceNetwork をファイアウォールによって保護されるクラウド ネットワークにマッピングし、MgmtNetwork をパブリック クラウド ネットワークにマッピングできます。

開始する前に

仮想マシンに複数の NIC がある場合、事前定義済み IP アドレス プロパティの概念について理解しておく必要があります。「事前定義済みの IP アドレス プロパティ (P. 103)」を参照してください。

手順

- 1 複数の NIC を追加するには、ノードを選択してノードの下にある [NICs] タブをクリックします。
ノードには最大 10 個の NIC を追加できます。
- 2 NIC を追加するには、[追加] アイコン () をクリックし、論理ネットワーク名を指定します。
- 3 (オプション) 別のノードとして同じネットワークの一部にするには、ドロップダウン メニューからネットワーク名を選択します。
- 4 キャンバスの別の場所をクリックするか、Enter を押して、NIC の追加を終了します。
- 5 キャンバス上部のツールバーで、[保存] をクリックして変更内容を保存します。

次に進む前に

アプリケーションを展開します。第 16 章「アプリケーションの展開 (P. 157)」を参照してください。

フレキシブル ディスク レイアウトの追加と管理

フレキシブル ディスク レイアウトにより、ストレージの柔軟性を高め、ノードにディスクを追加できます。アプリケーション ブループリントを作成する際に、ディスクを追加することもできます。ディスクは、プロビジョニング中に動的に作成されてノードに追加されます。

ディスクを異なるデータストアに配置して管理することもできます。ユーザーがブループリントのノードにフレキシブル ディスク レイアウトを定義すると、アプリケーション展開担当者はさらに展開をカスタマイズできます。

- それぞれのディスクを個々のデータストアにマッピングします。
- それぞれの追加ディスクにディスク サイズを設定します。

追加ディスクは、仮想マシン ノードの一部になります。ディスクを使用するサービスおよびサービスを使用するユーザーの両方がディスクを定義します。そのために、サービスとユーザーの間に前提条件が設定されます。

たとえば、Oracle サービスの作業に少なくとも 1 つのデータ ディスク、1 つの REDO ログ ディスクが必要な場合、ユーザーはノードに 2 つのディスクを追加する必要があります。そうしなければ、サービス スクリプトが正しく動作しません。

注意 同じノード アレイの個々のノードに異なるディスク設定を指定することはできません。

データベースは、フレキシブル ディスク レイアウトを活用できます。新しい Oracle 11g データベース サービスは、Application Services に追加設定なしで使用できるサービスとして追加されます。このサービスは、フレキシブル ディスク レイアウトを活用するように変更されます。

開始する前に

仮想マシンに複数のディスクを追加することを計画する場合は、定義済みディスク レイアウトの Info プロパティについて理解しておく必要があります。を参照してください。 [「事前定義済みのディスク レイアウト情報プロパティ \(P. 104\)」](#)

手順

- 1 ノードを選択し、ブループリントの下の [ディスク] タブの [新規] をクリックします。

アプリケーション ブループリントにディスクを追加する場合は、次のオプションを検討します。

表 14-1. ディスク オプション

オプション	説明
名前	ノード内の一意のディスク名。
マウント パス	マウントするディスクのパス。マウント パスが指定されていないか、ディスクがフォーマットされていない場合、ディスクは Application Services によってマウントされません。
ファイル システム	ext3、NTFS などのファイル システムを指定できます。
サイズ (GB)	プロビジョニングされるディスクのサイズ。
タグ	ディスクとストレージ間のマッピングを決定するために指定された文字列。
説明	ディスクに関する追加情報を指定します。

- 2 (オプション) それぞれの追加ディスクに複数のディスク タグを割り当てます。

それぞれのディスク タグは、ディスク レイアウト情報の複数のディスクに使用できます。ディスク タグは、ディスクの使用目的を示すために使用します。データベース サービス向けに最適化されている 4 つの定義済みディスク タグは次のとおりです。

表 14-2. ディスク タグ

タグ	説明
アーカイブ	データベース アーカイブ ファイルを格納します。
バイナリ	Oracle バイナリなどのバイナリ ファイルを格納します。

表 14-2. ディスク タグ (続き)

タグ	説明
データ	Oracle テーブル スペース ファイルなどのデータ ファイルを格納します。
ログ	データベース オンライン ログを格納します。

- 3 終了したら、キャンパスの上にあるツールバーの [保存] をクリックします。

次に進む前に

アプリケーションを展開します。を参照してください。 [第 16 章「アプリケーションの展開 \(P. 157\)」](#)

詳細ブループリントへの外部サービスの追加

アプリケーションでの処理に外部サービスが必要な場合は、外部の基本サービスまたはアドバンスド サービスをブループリントに追加できます。

開始する前に

- 基本または詳細の外部サービスの基本概念を理解しておく必要があります。[「ライブラリへの外部サービスの追加 \(P. 117\)」](#) を参照してください。
- サービス定義の基本的な概念を理解してください。[「サービスバージョン プロパティを定義する \(P. 111\)」](#) を参照してください。
- ノード間の依存関係を作成する方法を理解しておく必要があります。[「コンポーネント間の依存関係の作成 \(P. 143\)」](#) を参照してください。

手順

- 1 既存のアプリケーション ブループリントを開きます。
- 2 外部サービスを選択してブループリント キャンバスにドラッグします。
たとえば、ブループリントに vPostgres 外部サービスを追加できます。
- 3 (オプション) ブループリントの適切なコンポーネント間に依存関係を作成します。
青い点線が表示され、依存するコンポーネントをポイントします。
- 4 編集するプロパティをクリックします。
外部サービスのプロパティは、ライブラリの他のすべてのサービスと同様に編集または定義できます。
- 5 [保存] をクリックします。

次に進む前に

[第 16 章「アプリケーションの展開 \(P. 157\)」](#)。

アプリケーション バージョンをコピーする

既存のアプリケーション バージョンをコピーすると、展開プロファイルを除くすべてのブループリント コンテンツがコピーされ、アプリケーション名は同じままになります。アプリケーション バージョンとバージョンの説明のみを編集できます。


開始する前に

- **アプリケーション アーキテクト**として Application Services にログインします。
- コンポーネントをアプリケーション ブループリントに追加し、コンポーネント間の依存関係を作成し、サービスとスクリプトを変更するプロセスに慣れていない場合は、[「アプリケーションの作成 \(P. 138\)」](#) を参照してください。

- 自分が、アプリケーション バージョンを所有するビジネス グループの一員であることを確認します。

アプリケーション バージョンを所有するビジネス グループのメンバーになっていない場合、[コピー] ボタンは無効になっています。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションを開いてアプリケーション バージョンを選択します。
- 3 ツールバーの [コピー] ボタン () をクリックしてアプリケーション バージョンをコピーします。
- 4 アプリケーション バージョンを変更し、説明を追加して [保存] をクリックします。
コピーしたアプリケーション バージョンが開きます。

次に進む前に

展開プロファイルを作成します。[「展開プロファイルの作成 \(P. 157\)」](#) を参照してください。


アプリケーションのコピー

アプリケーションをコピーすると、アプリケーション バージョンのすべてのコンポーネントを保持し、新しいアプリケーション名を設定し、アプリケーション バージョンをスタンドアロン アプリケーションに変更できます。

開始する前に

- **アプリケーション アーキテクト**として Application Services にログインします。
- コンポーネントをアプリケーション ブループリントに追加し、コンポーネント間の依存関係を作成し、サービスとスクリプトを変更するプロセスに慣れていない場合は、[「アプリケーションの作成 \(P. 138\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 [アプリケーション] ページから [アプリケーション カード] ビューをポイントします。
- 3 [コピー] () をクリックして、アプリケーションをコピーします。
- 4 ドロップダウン メニューから既存のアプリケーション バージョンを選択します。
- 5 アプリケーションに新しい名前を設定して説明を追加します。

実行計画には、構成されたプロパティとサービス、依存関係、ノード構成、またはカスタム タスクの詳細を含めることができます。

- 6 [OK] をクリックします。

新しいアプリケーションが [アプリケーション] ページに表示されます。

次に進む前に

展開プロファイルを作成します。[「展開プロファイルの作成 \(P. 157\)」](#) を参照してください。

アプリケーション バージョンの削除

古いアプリケーション バージョンは、アプライアンスから削除できます。

アプリケーション バージョンを削除すると、そのブループリントが削除されます。展開プロファイルを含むアプリケーション バージョンは削除できません。

開始する前に

- **アプリケーション アーキテクト**として Application Services にログインします。

- コンポーネントをアプリケーション ブループリントに追加し、コンポーネント間の依存関係を作成し、サービスとスクリプトを変更するプロセスに慣れていない場合は、「[アプリケーションの作成 \(P. 138\)](#)」を参照してください。
- 自分が、アプリケーション バージョンを所有するビジネス グループの一員であることを確認します。
アプリケーション バージョンを所有するビジネス グループのメンバーになっていない場合、[削除] ボタンは無効になっています。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションを開いてアプリケーション バージョンを選択します。
- 3 [削除] と [OK] をクリックして確認します。

アプリケーション バージョンが削除され、[アプリケーション バージョン] リストが表示されます。

成果物の操作

成果物管理により、組織はビルド ファイルを使用するアプリケーションや、その他の種類のソフトウェア成果物を、物理的な場所にかかわらず展開することができます。

成果物とはビルド プロセスのスクリプトまたは出力のことです。成果物管理では、アプリケーション ブループリントはタイプおよび名前を使用して成果物を指定します。場所や一意識別子は使用しません。成果物管理は成果物の物理的な場所や ID を監視し、展開中に必要な成果物を提供します。

成果物管理では、アプリケーション カタログ管理者が成果物リポジトリ仕様を作成し、それらを成果物リポジトリ インスタンスにマップします。成果物リポジトリ仕様は、成果物を無制限に含むことのできるストアを識別します。これには成果物リポジトリ インスタンスとそれに含まれる成果物を識別するプロパティがあります。Application Services は Jenkins および YUM に、設定不要の成果物リポジトリ仕様および追加の成果物リポジトリ仕様を定義する容量を提供します。

成果物リポジトリ仕様には成果物リポジトリ インスタンスから成果物を取得できるエージェント側およびサーバ側スクリプトがあります。スクリプトはメタデータを作成し、それを成果物とともに保存します。成果物を追跡および検索することもできます。スクリプトのライフサイクルは、成果物の解決、成果物のダウンロード、および追跡 ID の検索です。独自の Bash、Windows Cmd、Windows Powershell、BeanShell および JavaScript スクリプトを作成することができます。

アプリケーション アーキテクトおよび他のアプリケーション管理者は、成果物仕様を作成して、実際の Jenkins ビルドプロジェクトなどの成果物リポジトリ インスタンスにマッピングしてから、これらの成果物仕様をサービスや外部サービスのプロパティ、またはアプリケーション ブループリントの WAR ファイルまたは JAR ファイルなどのアプリケーション コンポーネントにバインドします。成果物仕様は成果物を名前、説明、およびビジネス グループでのみ識別します。選択した展開環境により、実際の成果物リポジトリ インスタンスが決定され、使用する展開プロファイルにより、実際の成果物が指定されます。展開環境ごとに 1 つのリポジトリ インスタンスに成果物をマッピングすることができます。

この章では次のトピックについて説明します。

- [成果物リポジトリ仕様の作成 \(P. 150\)](#)
- [成果物リポジトリ インスタンスの作成 \(P. 152\)](#)
- [成果物仕様の作成 \(P. 153\)](#)
- [成果物仕様の成果物リポジトリ インスタンスへのマップ \(P. 153\)](#)
- [アプリケーション ブループリントへの成果物のバインド \(P. 154\)](#)
- [成果物を使用したアプリケーションの展開 \(P. 155\)](#)

成果物リポジトリ仕様の作成



成果物リポジトリ仕様は成果物リポジトリ インスタンスのテンプレートです。

成果物リポジトリ仕様は、成果物リポジトリ インスタンスにアクセスし、それに含まれる成果物を特定するためのプロパティを定義します。この仕様には、プロパティ値を使用して展開中に成果物リポジトリ インスタンスおよび成果物を解決するアクション スクリプトが含まれています。

開始する前に

- アプリケーション アーキテクトまたはアプリケーション カatalog管理者として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ]-[成果物リポジトリ] を選択します。
- 2 [追加] アイコン () をクリックします。
- 3 成果物リポジトリ仕様の名前と説明を入力します。
- 4 リポジトリ内の物理的な成果物を識別する追跡識別子を入力します。
ビルド ID、ジョブ ID、コミット ID、バージョン番号、またはその他の識別子として機能する一意の文字列を使用します。システムでは、この識別子を使用して、同じリポジトリ内の同じ成果物の異なるバージョンを特定します。この識別子は、アクション スクリプト内で使用できます。
- 5 [ビジネス グループ] ドロップダウン メニューからビジネス グループを選択します。
ビジネス グループ ユーザーのみがこの仕様から作成された成果物リポジトリ インスタンスにアクセスできます。
- 6 リポジトリ アクセス プロパティを追加します。
リポジトリ アクセスプロパティは、成果物リポジトリ インスタンスの特定に必要なサーバ URL およびその他の情報を指定します。
 - a リポジトリアクセス プロパティの [追加] アイコン () をクリックします。
可能な場合、代わりにリポジトリ アクセス プロパティ リスト内の空の行を使用することができます。
 - b [プロパティ名] テキスト ボックスにプロパティ名を入力します。
 - c (オプション) [説明] テキスト ボックスに説明を入力します。

- d [タイプ] ドロップ ダウン メニューからプロパティのタイプを選択します。

ファイルタイプ	説明
文字列	この仕様から作成されたりポジトリの一意の識別子を指定します。最大 1024 文字の英数字を使用できます。
Integer	1 つの正または負の整数値を指定します。
コンテンツ	スクリプトにより成果物としてダウンロードされるオブジェクトの URL を指定します。
アレイ	配列を ["value1","value2",...] の形式で指定します。
計算値	スクリプトにより提供される計算値を指定します。
Boolean	[はい] または [いいえ] を指定します。
Double	倍精度整数を指定します。
単一 Select	単一の select 文を定義します。

- e (オプション) [値] テキスト ボックスにプロパティ値を入力します。

仕様から作成された成果物リポジトリ インスタンスすべてに同じ値を使用する場合にのみ、ここに値を入力します。

- f (オプション) [保護] チェック ボックスをオンにします。

オンにすると、文字がマスクされてプロパティの値が非表示になります。

7 成果物の識別子プロパティを追加します。

- a 成果物識別子プロパティの [追加] アイコン (+) をクリックします。

可能な場合、代わりに成果物識別子プロパティ リスト内の空の行を使用することができます。

- b [プロパティ名] テキスト ボックスにプロパティ名を入力します。

- c (オプション) [説明] テキスト ボックスに説明を入力します。

- d [タイプ] ドロップ ダウン メニューからプロパティのタイプを選択します。

ファイルタイプ	説明
文字列	この仕様から作成されたりポジトリの一意の識別子を指定します。最大 1024 文字の英数字を使用できます。
Integer	1 つの正または負の整数値を指定します。
コンテンツ	スクリプトにより成果物としてダウンロードされるオブジェクトの URL を指定します。
アレイ	配列を ["value1","value2",...] の形式で指定します。
計算値	スクリプトにより提供される計算値を指定します。
Boolean	[はい] または [いいえ] を指定します。
Double	倍精度整数を指定します。
単一 Select	単一の select 文を定義します。

- e (オプション) [値] テキスト ボックスにプロパティ値を入力します。

成果物リポジトリ インスタンスのすべての成果物に同じ値を使用する場合にのみ、ここに値を入力します。

- f (オプション) [保護] チェック ボックスをオンにします。

オンにすると、文字がマスクされてプロパティの値が非表示になります。

- 8 成果物の場所を解決するスクリプトを作成します。
 - a スクリプトのタイプを RESOLVE_ARTIFACT スクリプトの [スクリプト タイプ] ドロップダウン メニューから選択します。
 - b [スクリプト] テキスト ボックスをクリックしてスクリプトを編集します。
 - c 編集ウィンドウにスクリプト テキストを入力します。
 編集ウィンドウの上部にあるドロップダウン メニューを使用して、スクリプトのタイプを変更したり、挿入するプロパティを選択したりすることができます。
 - d [OK] をクリックし、編集ウィンドウを閉じてスクリプトを保存します。
 [スクリプト] テキスト ボックスにスクリプト テキストが表示されます。
- 9 終了したら [保存] をクリックして、変更内容を保存します。

次に進む前に

[「成果物リポジトリ インスタンスの作成 \(P. 152\)」](#)。


成果物リポジトリ インスタンスの作成

物理的なリポジトリと成果物リポジトリ仕様を関連付けることにより、成果物リポジトリ インスタンスを作成します。

開始する前に

- **アプリケーション クラウド管理者**として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- 適切な成果物リポジトリ仕様が存在することを確認します。
- 適切な展開環境が存在することを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [クラウド] - [展開環境] を選択します。
- 2 使用する展開環境をクリックします。
 展開の詳細ウィンドウが表示されます。
- 3 [成果物リポジトリ インスタンス] タブをクリックします。
- 4 [追加] アイコン () をクリックします。
 [新規成果物リポジトリ インスタンス] ページが表示されます。
- 5 [成果物] ドロップダウン メニューから成果物リポジトリ仕様を選択します。
- 6 [名前] テキスト ボックスに名前を入力します。
- 7 (オプション) [説明] テキスト ボックスに説明を入力します。
- 8 [プロパティ] セクションに必要なプロパティ値をすべて入力します。
- 9 [保存] をクリックします。

次に進む前に

[「成果物仕様の作成 \(P. 153\)」](#)。


成果物仕様の作成

成果物仕様は 1 つまたは複数の成果物リポジトリ インスタンスにある論理成果物を説明します。成果物仕様では物理的な詳細は提供されません。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。第 12 章「[Application Services コンポーネントの開発 \(P. 93\)](#)」を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [成果物] を選択します。
- 2 成果物の [追加] アイコン () をクリックします。
- 3 [名前] テキスト ボックスに成果物の名前を入力します。
- 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 [ビジネス グループ] ドロップダウン メニューからビジネス グループを選択します。
ビジネス グループのユーザーのみが成果物にアクセスできます。
- 6 終了したら [保存] をクリックします。

次に進む前に

[「成果物仕様の成果物リポジトリ インスタンスへのマップ \(P. 153\)」](#)。

成果物仕様の成果物リポジトリ インスタンスへのマップ


成果物仕様を成果物リポジトリインスタンスにマップする場合、展開中に成果物を検索する場所を指定します。


成果物リポジトリ インスタンスの成果物仕様への現在のマッピングを変更することもできます。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- 適切な成果物仕様が存在することを確認します。[「成果物仕様の作成 \(P. 153\)」](#)を参照してください。
- 適切な成果物リポジトリ インスタンスが存在することを確認します。[「成果物リポジトリ インスタンスの作成 \(P. 152\)」](#)を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [ライブラリ] - [成果物] を選択します。
- 2 成果物リポジトリの [追加] アイコン () をクリックします。
成果物のリポジトリのウィンドウが表示されます。

- 3 [成果物名] ドロップダウン メニューから成果物名を選択します。
成果物仕様にすでにマップされている成果物リポジトリ インスタンスが成果物リポジトリ インスタンス リストに表示されます。
- 4 成果物リポジトリ インスタンスの [追加] アイコン () をクリックします。
ドロップダウン メニューのある新しい行がリストに表示されます。
- 5 ドロップダウン メニューから成果物リポジトリ インスタンスを選択します。
選択したインスタンスのプロパティ値が成果物識別子テーブルに表示されます。
- 6 (オプション) 必要に応じて、現在のプロパティ値を置き換える新しい値を入力します。
- 7 (オプション) 成果物リポジトリ インスタンスの成果物仕様への現在のマッピングを変更します。
 - a [成果物名] ドロップダウン メニューから成果物仕様を選択します。
 - b 変更する成果物リポジトリ インスタンスを選択します。
 - c 別の成果物リポジトリ インスタンスを選択します。
- 8 [OK] をクリックします。

次に進む前に

[「アプリケーション ブループリントへの成果物のバインド \(P. 154\)」](#)

アプリケーション ブループリントへの成果物のバインド

成果物をアプリケーション ブループリントにバインドする場合、実際には成果物仕様をブループリントのアプリケーション コンポーネント、サービス、または外部サービスのプロパティにバインドします。

開始する前に

- **アプリケーション アーキテクト**として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- 成果物仕様と成果物リポジトリ インスタンスの適切なマッピングが存在することを確認します。[「成果物仕様の成果物リポジトリ インスタンスへのマップ \(P. 153\)」](#) を参照してください。
- アプリケーションおよびアプリケーション ブループリントが存在することを確認します。[「アプリケーションの作成 \(P. 138\)」](#) および [「アプリケーション ブループリントを構成する \(P. 141\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーション、アプリケーション バージョン、およびブループリントを開きます。
- 3 既存のアプリケーション コンポーネント、サービス、または外部サービスを選択するか、あるいは新しいものを追加します。
- 4 [プロパティ] タブをクリックします。
- 5 プロパティを探して成果物にバインドします。
たとえば、JAR ファイルには、ダウンロード可能ファイルの URL を指定できる `jar_file` プロパティがあります。あるいは、yum リポジトリからパッケージをインストールする場合のパッケージ名として一意のテキスト文字列を生成およびバインドできます。
- 6 プロパティの [編集] アイコンをクリックします。
[プロパティの編集] ダイアログ ボックスが表示されます。

- 7 成果物へのバインドがラベル付けされた [ブループリント値] ドロップダウン メニューから成果物名を選択します。
- 8 終了したら [保存] をクリックします。
- 9 プロンプトが表示されたら、[OK] をクリックして保存するか、ブループリントの変更内容を破棄します。

次に進む前に

[「成果物を使用したアプリケーションの展開 \(P. 155\)」](#)。

成果物を使用したアプリケーションの展開

1 つまたは複数の成果物を使用してアプリケーションをアプリケーション コンポーネントとして展開できます。

展開プロファイル用に選択した展開環境により、使用する成果物リポジトリ インスタンスが決定されます。

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- 成果物リポジトリ インスタンスを含む展開環境が存在することを確認します。[「成果物リポジトリ インスタンスの作成 \(P. 152\)」](#) を参照してください。
- アプリケーションおよびアプリケーション バージョンが存在することを確認します。
- 成果物バインドを含むアプリケーション ブループリントが存在することを確認します。[「アプリケーション ブループリントへの成果物のバインド \(P. 154\)」](#) を参照してください。
- 展開プロファイルが存在することを確認します。
- 展開プロファイルのアプリケーション プロパティへのアクセスを確認します。[「アプリケーション プロパティの構成 \(P. 162\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーション、アプリケーション バージョン、および展開プロファイルを順番に開きます。
- 3 展開プロファイル ウィザードの [アプリケーション プロパティ] ページを開きます。
- 4 マップされたオブジェクト タイプのタブをクリックします。たとえば、[アプリケーション コンポーネント] または [サービス] をクリックして該当するプロパティを表示します。
- 5 成果物リポジトリと成果物名を [新しい値] テキスト ボックスで探します。
値には **リポジトリ** および **成果物** プリフィックスが付きます。
- 6 [新しい値] テキスト ボックスの名前をクリックします。
[プロパティの値] ダイアログ ボックスが表示されます。
- 7 (オプション) 成果物リポジトリ仕様で定義された追跡識別子の値を入力できます。
設定不要の成果物リポジトリ仕様の追跡識別子は、JOB_ID (Jenkins ビルド用) および VERSION (Yum ビルド用) です。その他の仕様には、別の追跡識別子を付けることができます。テキスト ボックスを空白のままにして後で展開時に追跡識別子を入力したり、追跡識別子を入力するためにここに戻ってきたり、またはアプリケーションのクイック展開を使用したりすることができます。
- 8 (オプション) 展開の成果物管理をオーバーライドすることもできます。固有の値を入力するテキスト ボックスにアプリケーション コンポーネントの URL を指定します。以降はこの URL が優先されます。
- 9 [OK] をクリックします。

アプリケーションの展開

Application Services は、ハイブリッドクラウド環境での多層エンタープライズ アプリケーションの展開を簡素化して自動化します。

Application Services を使用すると、展開、テスト、本番などのライフ サイクルのさまざまな展開プロファイルを作成できます。展開プロファイルを複合展開プロファイルに追加して、そのプロパティをバインドし、まとめて順番に展開できます。展開プロファイル設定は Application Services に保存されます。保存済み展開プロファイルを再利用して、アプリケーション バージョンをサポートされるクラウド環境に展開できます。

展開されたアプリケーションを更新すると、Application Services は、変更された値に基づいて更新プロファイルを作成します。この更新プロファイルには、定義済みの更新ライフ サイクル ステージから生成された更新スクリプトが含まれます。

アプリケーションの展開に関する主要概念について理解するには、[「主要な概念 \(P. 16\)」](#) を参照してください。

この章では次のトピックについて説明します。

- [展開プロファイルの設定と構成 \(P. 157\)](#)
- [展開プロファイルを使用した展開 \(P. 168\)](#)
- [vCloud Automation Center サービス カatalog への展開プロファイルの公開 \(P. 170\)](#)
- [vRealize Automation のサービス カatalog アイテムの申請 \(P. 173\)](#)
- [\[展開サマリ\] ページの使用 \(P. 174\)](#)
- [\[複合展開サマリ\] ページの使用 \(P. 174\)](#)
- [展開および更新プロセスについて \(P. 175\)](#)
- [展開エラーについて \(P. 176\)](#)

展開プロファイルの設定と構成

展開プロファイルを使用すると、特定の展開環境で使用できる、クラウドテンプレート、ネットワーク、およびアプリケーションのプロパティ値などを構成できます。

アプリケーションを展開する前に、展開プロファイル内で、実行プランの確認、実行プランへのカスタム タスクの追加、展開プロファイル設定の確認および変更を行うことができます。

展開プロファイルの作成


それぞれの展開環境に対して異なる展開プロファイルを作成します。展開プロファイルでは、特定の展開のアプリケーションのプロパティに対して [展開時にオーバーライド] オプションが有効になっている場合、そのプロパティを入力またはオーバーライドできます。

たとえば、特定の展開環境のデータベース ポートを 3307 に変更することがあります。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- Application Services に少なくとも 1 つのアプリケーションが作成されていることを確認します。第 14 章「アプリケーションの作成 (P. 137)」を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択して [追加] アイコン () をクリックします。
選択メニューが表示されます。
- 4 メニューから [展開プロファイルの作成] を選択します。
メニューには既存の展開プロファイルのクラスタを作成することができる [複合展開プロファイルの作成] も含まれています。
- 5 展開プロファイルの名前を入力します。
プロファイルには、使用されている環境の種類、特定のオーバーライド プロパティ、またはクラスタ化されたノードの構成を示す名前を付けることができます。
たとえば、MyTimecard というアプリケーションでは、プロファイルに myTimecard-QA という名前を付け、そのプロファイルが本番環境ではなく QA 環境用であることを示します。
- 6 共有アプリケーション用の展開プロファイルを作成する場合は、ドロップダウン メニューからビジネス グループを選択します。
プライベート アプリケーション用のビジネス グループを選択することはできません。
- 7 [展開] をクリックします。

[展開プロファイル] ウィザードが表示され、[展開環境] ページが強調表示されます。

次に進む前に

アプリケーション展開の論理テンプレートとネットワーク テンプレートをマッピングします。「展開環境の構成 (P. 158)」を参照してください。

展開環境の構成

展開環境で使用可能なクラウド テンプレートとネットワークのリストを取得して、論理テンプレートと論理ネットワークにマッピングできます。また、Application Services でカスタム プロパティを構成して、vRealize Automation ブループリント カスタム プロパティをオーバーライドして事前定義済みディスクをストレージにマッピングするか、既存のプロパティに追加することもできます。

[仮想マシンのテンプレート] セクションに一覧表示されているノードは、アプリケーション ブループリントに示されているようにアプリケーションのコンポーネントに対応します。アプリケーションにスクリプトの実行を必要とするカスタム外部サービスが含まれている場合、仮想マシンのテンプレートは一時的な仮想マシンにマッピングされます。

[ネットワーク] セクションには、ブループリントに定義されている論理ネットワークが一覧表示されます。選択してマッピングするネットワークは、展開環境によって異なります。展開プロファイルには、vCloud Director および Amazon EC2 ネットワークのネットワーク リストも含まれています。vCloud Director の場合、Application Services は DCHP の有無にかかわらず、外部ネットワークおよび vCloud 経由ネットワークをサポートします。ネットワーク リストが空の場合は、vCloud Director 管理者にお問い合わせください。Amazon EC2 の場合、Application Services は NAT 経由、パブリック、およびプライベートのネットワークをサポートしています。NAT 経由およびパブリックのネットワークはイ

インターネットにアクセスできます。NAT 経由でないプライベート ネットワークは Application Services アプライアンスにのみアクセスできます。Amazon EC2 に展開するには、外部ネットワークに配置する仮想マシンを慎重に決定する必要があります。外部ネットワークの各 NIC は弾性 IP アドレスを取得し、これによってそのインターフェイスをインターネット上に配置します。NIC を外部ネットワークに配置するのは、絶対必要な場合のみにしてください。

クラウド テンプレートをマッピングして、外部クラウド ネットワークに接続することができます。外部クラウド ネットワークに接続するには、次の手順を実行します。

- ネットワーク プロファイルが外部タイプであることを確認します。
- ネットワーク プロファイルをネットワーク パスに割り当てるか、予約を作成します。

ドロップダウン リストからクラウド ネットワーク名を選択して、ノードまたはクラスターで定義されている NIC へのネットワーク マップを選択します。クラウド ネットワーク名の横にあるアイコンをクリックして、ネットワークの詳細を表示します。

vRealize Automation ブループリントを任意で定義して個々のカスタム プロパティ、またはビルド プロファイルとしてグループ化したカスタム プロパティのコレクションを定義できます。仮想マシンが作成されると、これらのカスタム プロパティはその仮想マシンに適用されます。Application Services により、vRealize Automation ブループリント カスタム プロパティをオーバーライドしたり、既存のプロパティに追加したりすることができます。たとえば、既存の vRealize Automation ネットワーク情報をオーバーライドするには、アプリケーション ブループリントの特定のノードのネットワーク アドレスまたは固定 IP アドレスをカスタム プロパティで指定できます。この定義済みカスタム プロパティは、仮想マシンの作成時に常に適用されます。

カスタム プロパティはキーと値のペアです。これらのプロパティは `key=value` として定義できます。

ブループリントでのカスタム プロパティの定義と使用についての関連情報については、『カスタム プロパティのリファレンス』を参照してください。

注意 Application Services の予約済みプロパティと内部プロパティは、カスタム プロパティとして使用しないでください。

[「Application Services の予約済みプロパティと内部プロパティ \(P. 161\)」](#) を参照してください。

注意 [追加の構成] オプションが使用可能な状態では、[追加の構成] オプションを使用してカスタム プロパティを追加するよりも vRealize Automation のマシン ブループリントにカスタム プロパティを追加することをお勧めします。vRealize Automation のマシン ブループリントに追加するカスタム プロパティは、vRealize Automation カタログからアプリケーションのプロビジョニングを申請するときにノード プロパティとして表示されます。ただし、[追加の構成] オプションを使用してマシン ブループリントの一部として表示されないカスタム プロパティ（ブループリントの関連付けされたビルド プロファイルまたはビジネス グループの一部ではないカスタム プロパティを含む）を追加できます。

開始する前に

- アプリケーション クラウド管理者やアプリケーションの公開者および展開者として Application Services にログインします。
- ブループリントで使用されている各論理テンプレートに少なくとも 1 つのクラウド テンプレートがマッピングされていることを確認します。[「ライブラリへの論理テンプレートの追加 \(P. 125\)」](#) を参照してください。
- Application Services に少なくとも 1 つのアプリケーションが作成されていることを確認します。[第 14 章「アプリケーションの作成 \(P. 137\)」](#) を参照してください。
- クラウド環境によっては、展開環境で少なくとも 1 つの vCloud Director または Amazon EC2 ネットワークが使用可能である必要があります。[「vCloud Director 展開環境の作成 \(P. 81\)」](#) または [「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。
- クラウド テンプレートを Application Services クラウド プロバイダに登録します。[「vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)」](#)、[「vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)」](#) および [「Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)」](#) を参照してください。
- Application Services 展開環境のカスタム プロパティおよび予約済みカスタム プロパティについて理解しておく必要があります。vRealize Automation [「カスタム プロパティのリファレンス」](#) を参照してください。

- 展開プロファイルが作成されていることを確認します。「[展開プロファイルの作成 \(P. 157\)](#)」を参照してください。

手順

- 1 リストから展開環境を選択し、[マップ詳細] をクリックします。

[展開環境] セクションに対応するリストを確認するには、タイトル バーをクリックしてドロップダウン メニューから [クラウド] - [展開環境] を選択します。

- 2 [外部サービス] セクションで、外部サービスを展開環境の外部サービス インスタンスにマッピングします。

外部サービス インスタンスのリストが空の場合、既存のインスタンスが展開環境に登録されていません。アクセスできる場合は、少なくとも 1 つの外部サービス インスタンスに登録するか、クラウド管理者にインスタンスに登録するように依頼してください。


関連付けられた外部サービス インスタンスがドロップダウン メニューに表示されます。

- 3 [仮想マシンのテンプレート] セクションで、論理テンプレートをクラウド環境のクラウド テンプレートにマッピングします。

クラウド テンプレートのリストが空の場合、既存のクラウド テンプレートがユーザーのグループに属していないか、クラウド テンプレートが登録されていません。アクセスできる場合は、既存のクラウド テンプレートがあるグループにログインするか、少なくとも 1 つのクラウド テンプレートを Application Services に登録します。クラウド管理者にクラウド テンプレートを登録するように依頼することもできます。

vRealize Automation の場合は、展開環境と同じ予約ポリシーを持つクラウド テンプレートがドロップダウン メニューに表示されます。


ユーザーと同じグループに属すクラウド テンプレートがドロップダウン メニューに表示されます。


- 4 (オプション) vRealize Automation 展開の場合は、[追加の構成] アイコン () をクリックして、カスタム プロパティをアプリケーション ブループリントの各ノードに追加します。

Application Services の予約済みプロパティと内部プロパティは使用しないでください。

- a サンプルの Clustered Dukes Bank アプリケーションでは、カスタム プロパティを定義して管理ネットワークを vCenter Server 直接ネットワークにマッピングし、サービス ネットワークを vCenter Server 経由ネットワークにマッピングします。

vCenter Server 直接ネットワークは network1 で、vCenter Server 経由ネットワークは network2 です。

- b appserver ノード行で、[追加の構成] アイコン () をクリックし、**virtualmachine.network0.name=network2** を appserver ダイアログ ボックスの [追加の構成情報] に入力して [保存] をクリックします。

- c database ノード行で、[追加の構成] アイコン () をクリックし、**virtualmachine.network0.name=network2** を database ダイアログ ボックスの [追加の構成情報] に入力して [保存] をクリックします。

- d load_balancer ノード行で、[追加の構成] アイコン () をクリックし、**virtualmachine.network0.name=network1 virtualmachine.network1.name=network2** を load_balancer ダイアログ ボックスの [追加の構成情報] に入力して [保存] をクリックします。

- 5 vCloud Director 展開環境と Amazon EC2 展開環境の場合は、[ネットワーク] セクションで、カタログの各論理ネットワークでサポートされているクラウド ネットワークを選択します。

たとえば、ロード バランサでアプリケーションをテスト環境に展開している場合、両方のロードバランサのネットワーク NIC に内部ネットワークを選択することがあります。本番環境の展開プロファイルを作成する場合、1 つのロードバランサの NIC には内部ネットワークを選択し、もう 1 つのロードバランサの NIC には外部ネットワークを選択する場合があります。

6 [ディスク] セクションで、個々のディスクを特定のストレージにマッピングします。

次のシナリオでは、[ディスク] セクションが空になる場合があります。

- 対応するアプリケーション ブループリントが追加のディスクを定義できない場合。この場合は、このセクションを空のままにして展開を続行します。
- vRealize Automation 以外の展開環境を選択した場合。この場合、警告メッセージ「**選択された展開環境では仮想マシンのディスクのカスタマイズができません。ブループリントで定義されたディスクはプロビジョニングされません。アクションスクリプトがブループリントのカスタマイズ済みディスク設定に依存する場合、展開に失敗することがあります。**」を無視しても、問題なく展開を続行できます。

フレキシブル ディスクレイアウトにより、ディスクを特定のデータストアに配置して、最小限のコストで最大限のパフォーマンスを実現できます。

たとえば、オペレーティングシステム ディスクを高速データストアで作成してパフォーマンスを高め、アーカイブディスクを低速データストアで作成してコストを削減します。

7 選択が終了したら [次へ] をクリックします。

[アプリケーション プロパティ] タブが表示されます。

次に進む前に

ノード コンポーネント、サービス コンポーネント、およびアプリケーション コンポーネントに該当するプロパティ値を定義します。[「アプリケーション プロパティの構成 \(P. 162\)」](#) を参照してください。

Application Services の予約済みプロパティと内部プロパティ

vFabric Application Director には予約済みのプロパティと内部プロパティがありますが、カスタムのプロパティを作成するときにこれらのプロパティを使用することはできません。これらのプロパティを使用すると、エラー メッセージが表示されます。

予約済みプロパティと内部プロパティ

予約済みプロパティ	内部プロパティ
agent.download.url	ホスト名
agent.jar.md5sum	VirtualMachine.Cpu.Count
agent.max.logsize	VirtualMachine.Memory.Size
agent.root.folder	VirtualMachine.Disk<N>.Size
amqp.heartbeat	VirtualMachine.Admin.CustomizeGuestOSDelay
amqp.host	VirtualMachine.Admin.UseGuestAgent
amqp.port	VirtualMachine.Software<N>.Name
queue.name	VirtualMachine.Software<N>.ScriptPath
server.url	
temp.key	
APPD_REQUEST_ID	
DEPLOYMENT_NAME	
DEPLOYMENT_URI	
DEPLOYMENT_LOCATION_URI	
VM_URI	
VM_NAME	
os.M_NAMEfamily	

予約済みプロパティ	内部プロパティ
APPD_REQUEST_ID	

アプリケーション プロパティの構成

ホスト名や vCPU などのノード プロパティに、新しい値を定義できます。アプリケーション ブループリントで [展開時にオーバーライド可能] チェック ボックスが選択されているアプリケーション コンポーネントとサービス プロパティにも新しい値を定義できます。

展開を正常に行うには、ライブラリ、ブループリント、または展開プロファイルから必要なノード プロパティに値を割り当てます。メモリ割り当てや CPU 数などのノード プロパティはシステムによって自動的に定義されますが、ユーザーはそれらをオーバーライドできます。たとえば、必要なサービスの JVM ヒープ サイズが 512 MB の場合があります。しかし大規模な展開のために、その設定をオーバーライドして、サイズを 1024 MB に変更できます。

vCloud Director または vRealize Automation 展開で仮想マシンを簡単に識別できるように、ホスト名を設定できます。一意の文字列を生成するには、ホスト名の最後に `${random}` 式を付加できます。

クラスタ化されたノードのクラスタ サイズも、新しい値で定義できます。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- Application Services に少なくとも 1 つのアプリケーションが作成されていることを確認します。[第 14 章「アプリケーションの作成 \(P. 137\)」](#) を参照してください。
- [展開環境] タブが構成されていることを確認します。[\[展開環境の構成 \(P. 158\)\]](#) を参照してください。

手順

- 1 ブループリントで設定したプロパティが、コンポーネントのタブに正確に表示されていることを確認します。
- 2 (オプション) [サービス]、[アプリケーション コンポーネント]、または [ノード] タブをクリックします。
- 3 特定のサービス、アプリケーション コンポーネント、またはノード プロパティを選択します。
- 4 テーブル内で特定の行の [新しい値] 列をクリックし、展開プロファイルで使用する値を入力します。

Amazon EC2 への展開の場合、`global_conf` プロパティ値を

`https://<DarwinServerIP>:8443/darwin/conf/darwin_global_noproxy.conf` に変更する必要があります。

- 5 元の値に戻すには、[値のリセット] ボタン (🔄) をクリックします。
- 6 (オプション) [ディスク マッピング] セクションで新しい整数値を入力し、ディスク サイズを増加または減少させます。
- 7 変更が終了したら、[次へ] をクリックします。

[実行プラン] タブが表示されます。

次に進む前に

実行プラン内のプロビジョニング タスクと依存関係を確認します。[\[実行プランを確認しカスタム タスクを追加する \(P. 162\)\]](#) を参照してください。

実行プランを確認しカスタム タスクを追加する

アプリケーション ブループリントに基づいて、展開実行プランが生成されます。アプリケーションを展開する前にこの実行プランを確認し、カスタマイズした追加のタスクを実行するカスタム タスクをアプリケーション展開に追加できます。

実行プランの青色の点線は、展開タスクを実行する特定の順序を示しています。

ホストおよびエージェント ブートストラップのプロビジョニング タスクが、各ノードのコンポーネントの横に表示されます。vRealize Automation 環境に展開されるアプリケーションの場合は、ホストおよびエージェント ブートストラップのタスクのほかに、ネットワーク ブートストラップのプロビジョニング タスクも表示されます。これらのプロビジョニング タスクでは、エージェントが各コンポーネントのインストールとセットアップ タスクを実行する前に処理されるプロセスを表示します。展開に失敗すると、プロビジョニング タスクのログを確認して、問題をトラブルシューティングできます。展開実行プランには、ホスト間のカスタム タスク、エージェント ブートストラップまたはネットワーク ブートストラップのプロビジョニング タスクを追加できません。

アプリケーションにスクリプトの実行が必要な外部サービスが含まれている場合は、一時仮想マシンが実行プランに表示されます。これらの仮想マシンは、ホストおよびエージェント ブートストラップのプロビジョニング タスクでスクリプトが正常に実行された後に、Application Services によって削除されます。一時仮想マシンを削除する前にアプリケーションでエラーが発生する場合は、外部サービスの仮想マシンが含まれるノードを特定し、それらをお使いのクラウド環境から削除する必要があります。

ブループリントは、すべての展開環境のアプリケーションに共通する実行プランを作成する際に便利です。場合によっては、展開環境ごとに実行プランをカスタマイズする必要があります。たとえば、アプリケーションが本番の展開環境に展開される場合、展開後に電子メールの送信が必要になることがあります。テストの展開環境では、このようなチェックは不要です。電子メールのカスタム タスクを作成して、サービスまたはアプリケーション コンポーネントの展開タスクが正常に終了したときに通知電子メールを送信できます。このタスクは、本番の展開環境への展開を行う、展開プロファイルの実行プランに追加できます。




注意 カスタム タスクの実行中にユーザー操作を求めるプロセスが存在しないことを確認します。中断によってタスクが一時停止し、無期限にアイドル状態が続きます。ユーザーは、1 時間後にアプリケーションの展開を中止できます。または 3 日後にアイドル状態の Application Services の展開が失敗します。


開始する前に


- アプリケーションの公開元および配備者として Application Services にログインします。
- Application Services に少なくとも 1 つのアプリケーションが作成されていることを確認します。[第 14 章「アプリケーションの作成 \(P. 137\)」](#) を参照してください。
- [アプリケーション プロパティ] タブが構成されていることを確認します。[「アプリケーション プロパティの構成 \(P. 162\)」](#) を参照してください。
- Application Services カタログに少なくとも 1 つのカスタム タスクが作成されていることを確認します。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。
- アプリケーション内の必要なノード プロパティのすべてに、展開を正常に実行するための値が割り当てられていることを確認します。
- クラウド環境によっては、展開環境で少なくとも 1 つの vCloud Director または Amazon EC2 ネットワークが使用可能である必要があります。[「vCloud Director 展開環境の作成 \(P. 81\)」](#) または [「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。

手順

- 1 実行プランのプロビジョニング タスク、コンポーネント、および依存関係を確認します。
- 2 ノードがクラスタ化されている場合は、[クラスタの展開] ボタン  をクリックして、ノードを展開します。

クラスタ化されたノードが展開されない場合、カスタム タスクは、クラスタの 1 番目の仮想マシンにのみ追加されます。アプリケーション アーキテクトによってノードがクラスタ化ノードに変更されると、既存のカスタム タスクはクラスタの 1 番目の仮想マシンにのみ適用されます。展開者は、展開時に、カスタム タスクがクラスタ内の 1 番目の仮想マシンに適用されるか、すべての仮想マシンに適用されるかを確認し、適切な手順を実行する必要があります。

- 3 スクリプトが関連付けられているサービスおよびコンポーネントの場合は、コンポーネント名またはサービス名の横の下矢印をクリックし、スクリプトまたはスクリプトで使用する変数定義を表示します。
- 4 (オプション) [スクリプト タスクの追加] ボタン  を選択して、カスタム タスクをノードにドラッグします。

[スクリプト タスクの追加] ボタンをドラッグすると、カスタム タスクを挿入できる場所を示すアンカー  が表示されます。クラスタ化されたノードの場合は、各ノードにカスタム タスクを追加します。

たとえば、1 つ以上のカスタム タスクを Application Server ノード、Database Server ノード、または Load Balancer ノードにドラッグできます。

カスタム タスクをノードにドロップすると、[カスタム タスクの追加] ダイアログ ボックスが表示されます。

- 5 (オプション) [ライブラリ タスク名] ドロップダウン メニューからタスクを選択します。

そのノードのオペレーティング システムでサポートされるカスタム タスクが表示されます。たとえば、カスタム タスクが CentOS 6.3 オペレーティング システムでサポートされており、ノードのオペレーティング システムが Ubuntu 12.4.2 の場合、タスクはメニューに表示されません。

カスタム タスクを選択すると、タスク、スクリプト、およびプロパティの詳細情報がダイアログ ボックスに表示されます。

- 6 (オプション) [プロパティ] タブでプロパティ値をオーバーライドするには、プロパティをクリックします。

たとえば、電子メールの送信カスタム タスクでは、プロパティの 1 つは受信者の電子メール アドレスです。プロパティの電子メール アドレス値を受信者の電子メール アドレスに設定できます。

- a [プロパティの編集] ダイアログ ボックスで、プロパティの新しい値を入力するか、ドロップダウン メニューから既存のプロパティを選択し、そのプロパティをアプリケーション ブループリントのプロパティの 1 つとバインドします。
- b [保存] をクリックします。

- 7 (オプション) [カスタム タスクの追加] ダイアログ ボックスで、カスタム タスクのスクリプトとプロパティの詳細情報を確認し、[OK] をクリックします。
- 8 [次へ] をクリックして、展開プロファイルの設定を確認します。
- 9 [保存] をクリックします。
- 10 [OK] をクリックします。

アプリケーション バージョンの展開プロファイルの一覧が表示されます。

次に進む前に

展開プロファイルを使用してアプリケーションを展開します。[「単一の展開プロファイルで展開する \(P. 168\)」](#) を参照してください。

既存の展開プロファイルを使用する

アプリケーション バージョンの既存の展開プロファイルを再利用できます。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- アップロードした CentOS 6.3 (32 ビット) テンプレートをクラウド プロバイダに登録します。[「vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)」](#)、[「vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)」](#)、または [「Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)」](#) を参照してください。
- ブループリントで使用されている各論理テンプレートに少なくとも 1 つのクラウド テンプレートがマッピングされていることを確認します。[「ライブラリへの論理テンプレートの追加 \(P. 125\)」](#) を参照してください。

- Application Services で、展開プロファイルが使用できることを確認します。[「展開プロファイルの設定と構成 \(P. 157\)」](#) を参照してください。
- クラウド環境によっては、展開環境で少なくとも 1 つの vCloud Director または Amazon EC2 ネットワークが使用可能である必要があります。[「vCloud Director 展開環境の作成 \(P. 81\)」](#) または [「Amazon EC2 の展開環境の作成 \(P. 92\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択し、既存の展開プロファイルをクリックします。
[展開プロファイル] ウィザードが開きます。
- 4 (オプション) ブループリントに新しいノードを作成した場合は、新しいノードをクラウドテンプレートにマッピングします。

既存の展開プロファイルに保存されたプロパティのオーバーライドは、ブループリント値が更新されても、新しい値の列でオーバーライドされたままになります。削除されたプロパティは、展開プロファイルに表示されません。

次に進む前に

展開タスクを完了し、アプリケーションを展開します。[「展開プロファイルの作成 \(P. 157\)」](#) および [「単一の展開プロファイルで展開する \(P. 168\)」](#) を参照してください。

複合展開プロファイルの作成

複合展開プロファイルでは、共有プロパティ バインドを使用して、複数の展開プロファイルを 1 つの展開可能ユニットに結合します。

開始する前に

アプリケーションの公開元および配備者として Application Services にログインします。

手順

- 1 [複合展開プロファイルの設定 \(P. 165\)](#)
複合展開プロファイルでは、複数の展開プロファイルを結合できます。結合すると、アプリケーションの複合展開になります。
- 2 [展開プロファイルの選択 \(P. 166\)](#)
展開プロファイルは 1 つ以上選択したり、展開の順序を決定したりできます。
- 3 [プロパティのバインド \(P. 167\)](#)
関連付けられているアプリケーションで使用されるノード、サービス、およびアプリケーション コンポーネントのプロパティをバインドできます。

複合展開プロファイルの設定


複合展開プロファイルでは、複数の展開プロファイルを結合できます。結合すると、アプリケーションの複合展開になります。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 適切なビジネス グループのメンバーであることを確認します。
- アプリケーションが存在することを確認します。[「アプリケーションのコピー \(P. 147\)」](#) を参照してください。

- アプリケーションにアプリケーション バージョンがあることを確認します。[「アプリケーション バージョンの作成 \(P. 138\)」](#) を参照してください。
- そのアプリケーション バージョンの展開プロファイルが 1 つ以上あることを確認します。[「展開プロファイルの作成 \(P. 157\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択して [追加] アイコン () をクリックします。
メニューが表示されます。
- 4 メニューから [複合展開プロファイルの作成] を選択します。
- 5 複合展開プロファイルの名前を入力します。
- 6 ビジネス グループを選択し、[展開] をクリックします。
ビジネス グループには、複合展開プロファイルに含める展開プロファイルが含まれている必要があります。

[展開プロファイルの追加] ページが表示されます。

次に進む前に

新しい複合展開プロファイルが設定されましたが空です。展開プロファイルを選択する必要があります。

展開プロファイルの選択

展開プロファイルは 1 つ以上選択したり、展開の順序を決定したりできます。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 複数の展開プロファイルがあることを確認します。[「展開プロファイルの作成 \(P. 157\)」](#) を参照してください。
- [「複合展開プロファイルの設定 \(P. 165\)」](#)。

手順

- 1 [アプリケーション] ドロップダウン メニューでアプリケーションを選択します。
アプリケーションには、複合展開プロファイルに含める展開プロファイルがある必要があります。
- 2 [アプリケーション バージョン] ドロップダウン メニューからアプリケーション バージョンを選択します。
- 3 [展開プロファイル] ドロップダウン メニューから展開プロファイルを選択します。
メニューには、複合展開プロファイルと同じビジネス グループ内の展開プロファイルだけが表示されます。
- 4 [[Add]] をクリックします。
展開プロファイルが選択テーブルに表示されます。
- 5 [手順 1](#) から [手順 4](#) を繰り返して、別の展開プロファイルを追加します。
同じ展開プロファイルを何回か追加すると、各複製のエイリアスに下線と増分シーケンス番号が付加されます。

- 6 (オプション) 展開プロファイルの展開順序を変更します。

展開プロファイルは、デフォルトでトップダウンの順に展開されます。順番を変更すると、順序を変更できます。展開プロファイルの順序を変更しても、物理的な位置は変わりません。

- a 変更する展開プロファイルの順番をクリックします。
- b 番号を上書きするか、上下の矢印をクリックします。

- 7 (オプション) 展開プロファイルの [詳細表示] リンクをクリックして、そのアプリケーションや展開環境などの情報を表示します。

続行するには、[詳細表示] ペインを閉じる必要があります。

- 8 展開プロファイルをグラフィカル形式で表示する場合は、[視覚化]をクリックします。

[詳細表示] リンクは展開プロファイルのグラフィカル表示にも表示されます。

- 9 [次へ] をクリックします。

[プロパティのバインド] ページが表示されます。

次に進む前に

展開プロファイルの選択と順序付けを完了しました。これで、これらの展開プロファイルに接続されたプロパティをバインドできます。


プロパティのバインド

関連付けられているアプリケーションで使用されるノード、サービス、およびアプリケーション コンポーネントのプロパティをバインドできます。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- [「展開プロファイルの選択 \(P. 166\)」](#)。

手順

- 1 [追加] アイコン () をクリックします。
- 2 [プロパティ] 列の [選択] をクリックします。
[プロパティの選択] ダイアログ ボックスが表示されます。
- 3 [展開プロファイル] ドロップダウン メニューから、複合展開プロファイルに追加した展開プロファイルのいずれかを選択します。
- 4 [サービス] または [アプリケーション コンポーネント] のラジオ ボタンを選択します。
選択肢は展開プロファイルによって異なる可能性があります。
- 5 対応するドロップダウン メニューからアイテムを選択します。
選択内容により、使用可能なプロパティが決まります。
- 6 [プロパティ] ドロップダウン メニューからプロパティを選択します。
- 7 [OK] をクリックします。
- 8 [プロパティへのバインド] 列の [選択] をクリックします。
- 9 [展開プロファイル] ドロップダウン メニューから、複合展開プロファイルに追加した展開プロファイルのいずれかを選択します。

メニューに、現在の展開プロファイルの前に展開される展開プロファイルが表示されます。表示される展開プロファイルには、現在の展開プロファイルより小さい (古い) 番号が付いています。

- 10 [ノード]、[サービス] または [アプリケーション コンポーネント] のラジオ ボタンを選択します。
選択肢は展開プロファイルによって異なる可能性があります。
- 11 [プロパティへのバインド] ドロップダウン メニューからプロパティを選択します。
- 12 [OK] をクリックします。
- 13 必要に応じて、追加プロパティをバインドする手順を繰り返します。
- 14 [保存] をクリックします。

展開プロファイルを使用した展開

展開プロファイルまたは複合展開プロファイルを保存すると、関連アプリケーションを展開できます。

単一の展開プロファイルで展開する

アプリケーションを Application Services ユーザー インターフェイスから展開できます。

コマンドライン インターフェイスからアプリケーションを展開することもできます。[\[CLI によるアプリケーションの展開と更新 \(P. 224\)\]](#) を参照してください。

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- 少なくとも 1 つの展開プロファイルが Application Services で使用可能であることを確認します。[\[展開プロファイルの作成 \(P. 157\)\]](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択して展開プロファイルを作成するか、既存のプロファイルを使用します。
- 4 [展開プロファイル] ウィザードのプロンプトに従って変更を行います。
- 5 展開プロファイルの設定を確認します。
アプリケーションを展開する前に、展開プロファイルを必要に応じて変更できます。
- 6 [展開] をクリックします。
[展開サマリ] ページが表示された後、約 30 秒間隔で更新され、展開のステータスが表示されます。
- 7 (オプション) ツールバーの [更新] をクリックすると、展開のリアルタイム ステータスが更新されます。

次に進む前に

アプリケーションをクラウドに展開するときにバックグラウンドで実行されるプロセスについて理解しておく必要があります。[\[展開および更新プロセスについて \(P. 175\)\]](#) を参照してください。

進行中の展開のステータスを確認します。[\[「展開サマリ」ページの使用 \(P. 174\)\]](#) を参照してください。

複合展開プロファイルを使用した展開

Application Services ユーザー インターフェイスで、複合展開プロファイルを使用して、プロパティがバインド済みの複数のアプリケーションを、選択した順序で展開できます。

コマンドライン インターフェイスからアプリケーションを展開することもできます。[\[CLI によるアプリケーションの展開と更新 \(P. 224\)\]](#) を参照してください。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 複合展開プロファイルが Application Services で使用できることを確認します。[「複合展開プロファイルの作成 \(P. 165\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択して複合展開プロファイルを作成するか、または既存のプロファイルを使用します。
複合展開プロファイルを作成するには、1 つ以上の展開プロファイルが作成されている必要があります。
- 4 選択した展開プロファイルとプロパティ バインドを確認します。
複合展開プロファイルに変更を加えてから、複合展開プロファイルに示されたアプリケーションを展開できます。
- 5 [展開] をクリックします。
[複合展開サマリ] ページが表示されます。このページは約 30 秒で更新され、各展開プロファイルの展開ステータスが表示されます。
- 6 (オプション) ツールバーの [更新] をクリックすると、展開のリアルタイム ステータスが更新されます。
- 7 [展開名] 列の名前をクリックすると、展開プロファイルの [展開サマリ] ページが表示されます。
[戻る] 矢印をクリックすると、[複合展開サマリ] ページに戻ります。

次に進む前に

アプリケーションをクラウドに展開するときにバックグラウンドで実行されるプロセスについて理解しておく必要があります。[「展開および更新プロセスについて \(P. 175\)」](#) を参照してください。

進行中の展開のステータスを確認します。[「\[展開サマリ\] ページの使用 \(P. 174\)」](#) を参照してください。

アプリケーションをクイック展開する

クラウド環境から展開済みアプリケーションを分解する場合、[展開プロファイル] ウィザードで要素を構成しなくてもアプリケーションをすばやく再展開できます。

クイック展開を使用して必要なプロパティを更新し、[展開プロファイル] ウィザードに関係のないプロパティをオーバーライドしてアプリケーション ブループリントを展開することもできます。


アプリケーションをクイック展開する場合、最新のアプリケーション バージョンおよび関連付けられた展開プロファイルが展開に使用できます。古いアプリケーション バージョンをクイック展開するには、アプリケーションを開いて特定のアプリケーション バージョンをクイック展開します。

注意 vRealize Automation 展開の場合、別の NIC をアプリケーションに追加し、既存の展開プロファイルを使用してアプリケーションをクイック展開すると、エラーが表示されます。[展開プロファイル] ウィザードを開き、[マップ詳細] を選択してクラウド ネットワークを新たに追加した NIC にマッピングする必要があります。


開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 少なくとも 1 つの展開プロファイルが Application Services で使用可能であることを確認します。[「展開プロファイルの作成 \(P. 157\)」](#) を参照してください。

- 関連する展開プロファイルを所有するビジネス グループのメンバーである必要があります。

関連する展開プロファイルを所有するビジネス グループのメンバーでない場合、[クイック展開] ボタン  が無効になります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーション ページで、展開するアプリケーションを選択します。
- 3 クイック展開するアプリケーション バージョンを選択します。
- 4 [クイック展開]  をクリックして展開プロセスを開始します。
アプリケーション バージョンを選択しないと、デフォルトで最新のアプリケーション バージョンが展開されます。
- 5 [ターゲット] ドロップダウン メニューで、関連付けられた展開プロファイルを選択します。
展開に必要なプロパティと展開でオーバーライド可能なプロパティが表示されます。
- 6 (オプション) 必要なプロパティまたはオーバーライド可能なプロパティに新しい値を設定し、[展開] をクリックします。
必要なプロパティまたはオーバーライド可能なプロパティの変更した値は展開プロファイルに保存されないため、将来の展開で再利用できません。

[展開サマリ] ページが表示され、進行中の展開のステータスが表示されます。

次に進む前に

[展開サマリ] ページのステータス ウィンドウを使用して、展開ステータスを追跡します。[「\[展開サマリ\] ページの使用 \(P. 174\)」](#) を参照してください。

vCloud Automation Center サービス カタログへの展開プロファイルの公開

Application Services を使用すると、vRealize Automation サービス カタログに展開プロファイルまたは複合展開プロファイルを公開できます。vRealize Automation ユーザーは、その環境でアプリケーションの使用を開始するためのカタログ アイテムを申請できます。

単一展開プロファイルの公開

Application Services 展開プロファイルを vRealize Automation サービス カタログに公開できます。vRealize Automation ユーザーは、その環境でアプリケーションの使用を開始するためのカタログ アイテムを申請できます。

公開されている展開プロファイルを vRealize Automation サービス カタログから削除することはできません。

ブループリントに成果物をマッピングした展開を公開する場合は、このようなブループリントで頻繁に使用される追跡 ID とコンテンツ URL を vRealize Automation サービス カタログからオーバーライドすることができます。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- Application Services で、展開プロファイルが使用できることを確認します。[「展開プロファイルの設定と構成 \(P. 157\)」](#) を参照してください。
- vRealize Automation の IaaS 環境と資格が適切にセットアップおよび構成されている必要があります。
vRealize Automation 『インストールおよび構成』 および 『仮想プラットフォームの IaaS 構成』 のドキュメントを参照してください。

- 展開プロファイルにカスタム タスクがある場合は、SSO ユーザーが vRealize Automation サービス カタログからアイテムを申請でき、アプリケーションの最新バージョンおよびそれに関連する展開プロファイルを迅速に展開できるように、必要なすべてのプロパティ値が定義されていることを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーションのバージョンを選択し、既存の展開プロファイルをクリックするか、または展開プロファイルを作成します。
[展開プロファイル] ウィザードが開きます。
- 4 (オプション) ブループリントに新しいノードを作成した場合は、新しいノードをクラウド テンプレートにマッピングします。
既存の展開プロファイルに保存されたプロパティのオーバーライドは、ブループリント値が更新されても、新しい値の列でオーバーライドされたままになります。削除されたプロパティは、展開プロファイルに表示されません。
- 5 展開プロファイルの設定を確認します。
公開する前に、展開プロファイルに変更を行うことができます。
- 6 [公開] をクリックします。
- 7 [発行] ダイアログ ボックスで、展開プロファイルの名前にさらに情報を追加し、[OK] をクリックします。
たとえば、ClusteredDukesBankDP 展開プロファイルに Publish_vCAC を追加できます。[説明] セクションに、展開プロファイルを使用する Dev、QE、または IT 環境などのオプションの説明を追加します。
公開された名前と説明は、vRealize Automation サービス カタログに表示されます。この名前と説明は、展開プロファイルの名前と説明とは異なります。
- 8 vRealize Automation のテナントにログインします。
vRealize Automation コンソールが開きます。
- 9 [管理] - [カタログ管理] - [カタログ アイテム] を選択して、カタログで公開されているアイテムを開きます。
- 10 使用可能なサービスを [サービス] ドロップダウン メニューのカタログ項目に割り当てて、[更新] をクリックします。
カタログ項目リストが表示されます。
- 11 (オプション) 右側のナビゲーションで、カタログ項目に対してサービスを使用可能にするには、[資格] を選択して資格を開きます。
 - a [詳細] タブで資格の名前を入力し、[次へ] をクリックします。
 - b [アイテムおよび承認] タブで、公開されたアイテムをドロップダウン メニューにある使用可能なカタログ アイテムに割り当てます。
 - c [使用可能なリソース アクション] セクションで、**アプリケーション展開の詳細** リソース アクションを入力し、ドロップダウン メニューからそれを選択します。
- 12 [更新] をクリックします。

次に進む前に

開発のために、vRealize Automation カatalog アイテムを申請します。[\[vRealize Automation のサービス カatalog アイテムの申請 \(P. 173\)\]](#) を参照してください。

複合展開プロファイルの公開

Application Services の複合展開プロファイルを vRealize Automation サービス カタログに公開できます。
vRealize Automation ユーザーは、カタログ アイテムを申請してその環境でアプリケーションの使用を開始できます。

公開されている展開プロファイルを vRealize Automation サービス カタログから削除することはできません。

ブループリントに成果物をマッピングした展開を公開する場合は、このようなブループリントで頻繁に使用される追跡 ID とコンテンツ URL を vRealize Automation サービス カタログからオーバーライドすることができます。

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- 複合展開プロファイルが Application Services で使用できることを確認します。[「展開プロファイルの設定と構成 \(P. 157\)」](#) を参照してください。
- vRealize Automation の IaaS 環境と資格が適切にセットアップおよび構成されている必要があります。
vRealize Automation 『インストールおよび構成』 および 『仮想プラットフォームの IaaS 構成』 のドキュメントを参照してください。
- 展開プロファイルのいずれかにカスタム タスクがある場合は、SSO ユーザーが vRealize Automation サービス カタログからアイテムを申請でき、アプリケーションの最新バージョンおよびそれに関連する展開プロファイルを迅速に展開できるように、必要なすべてのプロパティ値が展開プロファイルに定義されていることを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択し、既存の複合展開プロファイルをクリックするか、または作成します。
複合展開プロファイルを作成するには、1 つ以上の展開プロファイルが使用可能である必要があります。
- 4 展開プロファイルとプロパティ バインドを確認します。
複合展開プロファイルは、公開する前に変更できます。
- 5 [公開] をクリックします。
- 6 [公開] ダイアログ ボックスで、複合展開プロファイルの名前と詳しい情報を加え、説明を追加して [OK] をクリックします。

たとえば、展開されるアプリケーションに関してオプションの説明を [説明] セクションに追加します。

公開された名前と説明は、vRealize Automation サービス カタログに表示されます。この名前と説明は、展開プロファイルの名前と説明とは異なります。
- 7 vRealize Automation のテナントにログインします。
vRealize Automation コンソールが開きます。
- 8 [管理] - [カタログ管理] - [カタログ アイテム] を選択して、カタログで公開されているアイテムを開きます。
- 9 使用可能なサービスを [サービス] ドロップダウン メニューのカタログ項目に割り当てて、[更新] をクリックします。
カタログ項目リストが表示されます。

- 10 カタログ アイテムに対してサービスを使用可能にするには、右側のナビゲーションで [資格] を選択して資格を開きます。
 - a [詳細] タブで資格の名前を入力し、[次へ] をクリックします。
 - b [アイテムおよび承認] タブで、公開されたアイテムをドロップダウン メニューにある使用可能なカタログ アイテムに割り当てます。
 - c [使用可能なリソース アクション] セクションで、**アプリケーション展開の詳細** リソース アクションを入力し、ドロップダウン メニューからそれを選択します。
- 11 [更新] をクリックします。

次に進む前に

開発のために、vRealize Automation カタログ アイテムを申請します。[\[vRealize Automation のサービス カタログ アイテムの申請 \(P. 173\)\]](#) を参照してください。

vRealize Automation のサービス カタログ アイテムの申請

サービス カタログ項目を要求する場合は、指定されたクラウド環境に対して項目をプロビジョニングします。

vRealize Automation の標準的なユーザーは、Application Services 環境に直接アクセスできません。したがって、ユーザーはセルフサービス ポータルである vRealize Automation サービス カタログにアクセスし、Application Services アプリケーションの展開を申請できます。

開始する前に

- アプリケーションの公開者および展開者として、vRealize Automation の IaaS 環境にログインできることを確認します。vRealize Automation のインストールおよび構成方法と、カタログ アイテムに資格を設定する方法については、vRealize Automation のドキュメントを参照してください。
- 少なくとも 1 つの vRealize Automation のカタログ アイテムに資格とサービスが割り当てられていることを確認します。[\[単一展開プロファイルの公開 \(P. 170\)\]](#) を参照してください。
- **アプリケーションの公開者および展開者**として vRealize Automation コンソールにログインします。

手順

- 1 [管理] - [カタログ管理] - [カタログ アイテム] を選択します。
- 2 使用可能なサービスが割り当てられているカタログ アイテムを探します。
- 3 [カタログ] タブをクリックし、サービスにアクセスします。
- 4 サービス アイコンの [申請] をクリックして、説明を入力します。
たとえば、開発とテストのための項目を要求できます。
- 5 [次へ] をクリックして、項目のプロパティを確認します。
展開で、ブループリントの成果物のマッピングを使用し、ブループリントにプロパティ値として追跡識別子またはコンテンツ URL が含まれる場合は、サービス カタログのこの値をオーバーライドできます。
CPU、メモリ、クラスタ サイズ、ホスト名などのノード プロパティは、ノードのマシン ブループリントで定義したカスタム プロパティも含め編集が可能です。
- 6 [送信] をクリックして、カタログ項目を要求します。

展開が完了すると、Application Services は vRealize Automation にリソースを作成し、作成したリソースが [アイテム] タブに表示されます。

次に進む前に

vRealize Automation のリソースを選択して、進行状況を追跡します。Application Services の [展開サマリ] ページを使用して、リアルタイムでステータスを確認することもできます。[\[展開サマリ\] ページの使用 \(P. 174\)](#) を参照してください。

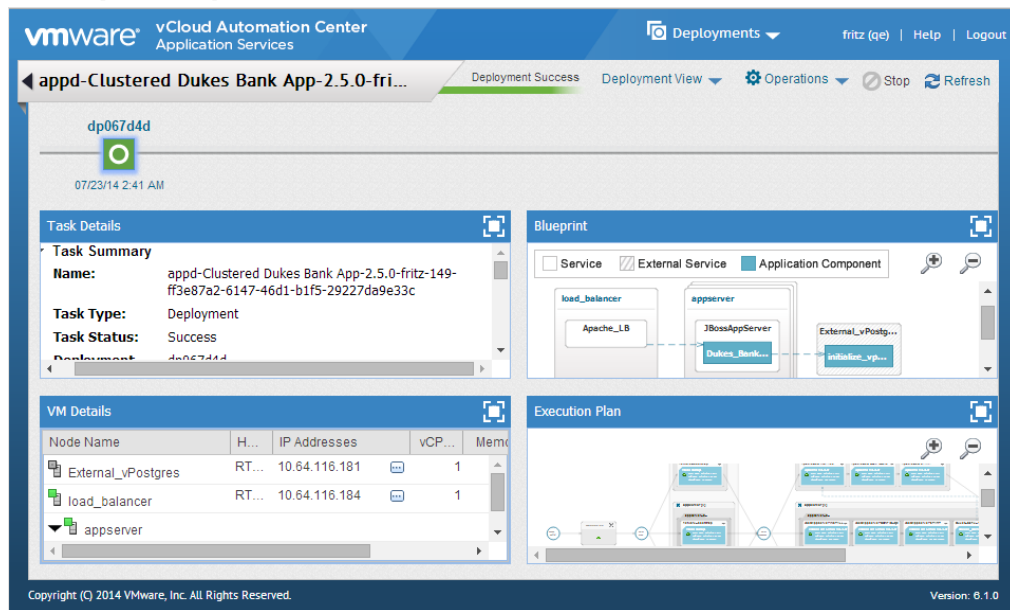
[展開サマリ] ページの使用

Application Services には、[展開サマリ] ページでアプリケーションの展開ステータスをリアルタイムで確認するためのグラフィカル ユーザー インターフェイスが用意されています。

vCloud Director および vRealize Automation のユーザー インターフェイスを使用して、展開済み仮想マシンのステータスを確認することもできます。Amazon VPC およびこれに関連付けられた可用性ゾーンで Application Services 展開のステータスを表示する方法については、Amazon AWS のドキュメントを参照してください。

[展開サマリ] ページでは、展開が実行されている間、展開、更新、分解の各プロセスについて展開全般のステータスがツールバーに表示されます。展開が終了すると、展開の各タスクの成否に応じて、ステータス バーが赤色または緑色に変わります。[タスクのステータス] ウィンドウの上部のタスク タイムラインには、アプリケーションの展開が開始された時刻、後続の更新展開（展開の拡大/縮小または構成の変更など）、分解プロセスが実行されたかどうかを示すタイム スタンプが表示されます。

図 16-1. [展開サマリ] ページ



[展開サマリ] ページ内のウィンドウを展開すると、アプリケーション展開の詳細とステータスが表示されます。[\[アプリケーションの展開 タスクとブループリントの詳細の表示 \(P. 210\)\]](#) および [\[展開された仮想マシンの詳細情報とアプリケーションの実行プランの表示 \(P. 211\)\]](#) を参照してください。

[複合展開サマリ] ページの使用

Application Services には、[複合展開サマリ] ページでアプリケーションの展開ステータスをリアルタイムで確認するためのグラフィカル ユーザー インターフェイスが用意されています。

vCloud Director および vRealize Automation のユーザー インターフェイスを使用して、展開済み仮想マシンのステータスを確認することもできます。Amazon VPC およびこれに関連付けられた可用性ゾーンで Application Services 展開のステータスを表示する方法については、Amazon AWS のドキュメントを参照してください。

[複合展開サマリ] ページでは、展開が実行されている間、展開全体のステータスがツールバーに表示されます。このページには、展開の順序、ステータス、展開名、展開プロファイル名、展開環境、アプリケーション、アプリケーションバージョン、処理を開始したログイン ユーザー、開始時間、および終了時間も表示されます。

[展開名] 列の名前は、各展開プロファイルの [展開サマリ] ページへのリンクになっています。名前をクリックすると、展開プロファイルの [展開サマリ] ページが開きます。[展開サマリ] ページの詳細については、[「\[展開サマリ\] ページの使用 \(P. 174\)」](#) を参照してください。

展開が終了すると、展開の各タスクの成否に応じて、ステータス バーが赤色または緑色に変わります。

展開および更新プロセスについて

アプリケーションをクラウドに展開するときや、クラウドに展開済みのアプリケーションを更新するとき、いくつかのプロセスがバックグラウンドで実行されます。仮想マシンが作成され、ソフトウェアが仮想マシンにプロビジョニングされます。展開エラーを簡単に特定しトラブルシューティングできるように、展開と更新プロセスについて理解しておくことが重要です。

Application Services では、実行プラン内の展開の実行中にプロセスを表示できます。ホストおよびエージェント ブートストラップのプロビジョニング タスクが、各ノードのコンポーネントの横に表示されます。vRealize Automation 環境に展開されるアプリケーションの場合は、ホストおよびエージェント ブートストラップのタスクのほかに、ネットワーク ブートストラップのプロビジョニング タスクも表示されます。

アプリケーションをクラウドに展開するプロセスや、展開済みアプリケーションを更新するプロセスには、次の手順が含まれます。複合展開プロファイルを使用する場合は、展開済みアプリケーションそれぞれで次の手順を繰り返します。

- 1 Application Services はポリシー評価を実行し、展開プロセスのコンプライアンスを判断します。重大としてマークされているポリシー インスタンス違反が存在する場合、展開プロセスまたは更新プロセスは停止します。
- 2 Application Services は、展開プロファイルでマップされたクラウド テンプレートをインスタンス化することで、仮想マシンをプロビジョニングします。

展開済みアプリケーションを拡大/縮小する更新プロセスの場合、Application Services は、展開プロファイルでマップされた vCloud Director カタログまたは vRealize Automation のクラウド テンプレートをインスタンス化することで、拡大/縮小されたクラスタ化仮想マシンをプロビジョニングします。拡大/縮小されたクラスタ化仮想マシンは、アプリケーションをクラウドに展開するときに最初に適用されたテンプレートと同じテンプレートを使用します。展開済みアプリケーションを更新するときは、テンプレートを変更できません。

更新プロファイルの残りのプロセスは、展開プロセスと同じです。

- 3 Application Services は、クラウド サービスに対してネットワーク接続の確立を申請し、展開内のすべての仮想マシンの IP アドレスを受信します。IP アドレスが割り当てられたら、セットアップ プロセスが正常に完了するように、仮想マシンが再起動します。ホスト名がアプリケーション ブループリントから生成され、ノード名として割り当てられます。

ホスト名が割り当てられない場合は論理名から取得され、ランダムに生成された一連の文字がホスト名に付加されます。
- 4 各仮想マシンに含まれるブートストラップ スクリプトによって、Application Services サーバのエージェントが仮想マシンにダウンロードされます。エージェントは、Java 仮想マシンで実行される JAR ファイルです。ブートストラップ スクリプトは、物理テンプレートにインストールされる必要があります。
- 5 ブートストラップ スクリプトにより、エージェント プロセスが開始されます。
- 6 エージェントは、Application Services サーバで認証されます。
- 7 各仮想マシンのエージェントは、Application Services の実行プランを仮想マシンにダウンロードします。
- 8 エージェントは、各コンポーネントのインストールおよびセットアップ タスクを、展開実行プランで指定された順番で実行します。
- 9 エージェントはスクリプトごとに依存タスクが正常に終了するのを待ち、全てのコンテンツを仮想マシン ディレクトリ `/tmp/<runid>/content/<ComponentName>/<PropertyName>` にダウンロードし、タスク スクリプトをディレクトリ `/tmp/<runid>/<TaskName>` にダウンロードします。エージェントは、サーバによって送信されたパラメータ値に応じてタスクを実行します。タスクが完了すると、エージェントはタスクのステータスをサーバに通知します。

スクリプトがゼロ以外の終了ステータスで終了すると、エージェントはそのタスクをエラーとしてマークします。それ以外の場合は、エージェントはスクリプトを完了としてマークし、次のタスクに進みます。1 つのタスクが失敗すると展開全体が停止し、失敗した展開というマークが付きます。その展開の以降のタスクは実行されません。エラーの原因は、[詳細] タブで確認できます。すべてのタスクが正常に終了したら、展開は [正常に展開済み] とマークされます。

注意 スクリプトを中断することなく実行する場合、戻り値をゼロ (0) に設定する必要があります。この値により、エージェントは提供されたすべてのプロパティを取得し、そのプロパティを Application Services サーバに送信できます。

トラブルシューティングの目的で、展開の詳細情報にアクセスしたり、実行プランに含まれるプロビジョニング タスク ログを表示したりできます。タスク情報は、展開の各コンポーネントで使用するインストール、構成、開始、更新、ロールバック、および分解スクリプトのログ ファイルに保存されます。これらのログには、**stdout** および **stderr** ログ ファイルに送信されるすべての情報が格納されます。

展開エラーについて

アプリケーションをクラウドに展開したり、クラウドに展開済みのアプリケーションを更新したりすると、仮想マシンが作成され、ソフトウェアが仮想マシンにプロビジョニングされます。これらのプロセス中に展開に失敗して、展開先の環境に影響が及ぶことがあります。

複数の展開段階で展開に失敗することがあります。Application Services で実行プランを表示すると、どこで失敗したのかを特定することができます。[展開の再開] オプションを使用すると、失敗した展開を再開できます。このオプションは、[展開ビュー] の [実行プラン] パネルにあります。

表 16-1. 展開段階

展開段階	失敗の種類
プロビジョニング前	<ul style="list-style-type: none"> ■ ポリシーの検証 ■ テンプレートの検証 ■ DE 到達可能性の検証
プロビジョニング	<p>ブート ストラップの失敗は次の理由で発生します。</p> <ul style="list-style-type: none"> ■ 仮想マシンの最大数が消費された ■ OS インストール ■ IP 設定 ■ ストレージの問題 <p>エージェントのブート ストラップの失敗は次の理由で発生します。</p> <ul style="list-style-type: none"> ■ テンプレートのファイアウォールの問題 ■ テンプレートのゲートウェイの問題 ■ エージェントから Application Services への通信中に、ネットワークやファイアウォールの問題によってタイムアウトが発生する
実行前	vRealize Automation ネットワーク ブートストラップの問題。
実行	<ul style="list-style-type: none"> ■ インストールまたは構成段階中に生じた、コンテンツのダウンロードの失敗 ■ インストールまたは構成開始段階中のスクリプトの失敗
実行後	外部サービスが展開の一部になっている場合、プロビジョニング解除に失敗することがあります。

再開操作でカスタム タスクを追加または削除することはできません。カスタム タスクの失敗によって生じる展開の失敗は再開できません。

展開の失敗の解決

アプリケーションのプロビジョニング フローの段階で展開に失敗したら、必ずアプリケーション展開を分解する必要があります。展開の失敗が頻繁に起きると重大な問題が生じるため、新しい展開を行わずに修正する必要があります。展開の再開操作を行うと、エラーの状態を修正して、失敗した展開を再開することができます。

プロビジョニング前の失敗

プロビジョニング前の失敗は、展開の初期段階に起こります。この段階で展開に失敗すると、エラー メッセージが表示されます。同じ展開内で展開の再開操作を起動する前に、エラーを修正してください。修正後、プロビジョニング前のプロセスを再開できます。

プロビジョニング前の失敗では、ポリシーとテンプレートの検証が原因でエラーが生じることがほとんどです。

プロビジョニングの失敗

ホストのプロビジョニングまたはエージェントのブートストラップに失敗すると、プロビジョニングに失敗することがあります。ホストまたはエージェントのブートストラップ ノードに赤いアイコンが表示されます。Application Services エージェントのインストールの問題またはエージェントに関する問題によって、仮想マシンのいずれかでブートストラップに失敗すると、新たにプロビジョニング済み仮想マシンが分解されて、再プロビジョニングされます。展開が再開すると、適切に動作していない仮想マシンは分解されて、再びプロビジョニングされます。

注意 ネットワーク ブートストラップの失敗はサポートされていません。

実行

実行の失敗は、コンテンツのダウンロードの失敗またはスクリプトの失敗によって生じることがあります。再開オプションが有効になっている場合は、失敗したコンポーネントのプロパティとスクリプトを編集できます。展開を再開すると、インストール、構成、または起動スクリプトなど、コンポーネントの最初のスクリプトの実行が開始されます。仮想マシンまたはスクリプトを再開すると、すでにインストールされた状態になっているために、再インストールまたはインストールの開始が行われなかったことがあります。インストールを開始するには、仮想マシンがクリーンな状態になるようにスクリプトを編集する必要があります。

アプリケーション展開の更新

既存のアプリケーション展開を更新する場合、その更新に必要な変更の新しい値を取得する更新プロファイルを作成します。更新をロールバックして前の有効な更新にリストアし、続けて追加の更新プロセスを開始することもできます。

重要 複合展開プロファイルを使用して展開したアプリケーションは、再開、あらゆるタイプの更新、およびスクリプトを使用した分解をサポートしません。ただし、このようなアプリケーションはクイック分解をサポートします。

保存した更新プロファイルを複数回展開して、既存の展開を更新できます。vCloud Application Director 6.0、6.0.1 および Application Services 6.1 を使用して、ノードのクラスタをスケール イン/スケール アウトするか、または既存のサービスおよびアプリケーション コンポーネントの構成を変更すると、vCloud Director および vRealize Automation で既存の展開の更新プロセスを複数回開始できます。

既存のアプリケーションのノード クラスタを拡大/縮小する場合、アプリケーションの追加のノードをサポートするのに十分なリソースがクラウドにあることを確認してください。

展開に外部サービスが含まれ、構成が外部サービス インスタンスで変更されている場合、通知アイコンが展開カードに表示されます。更新またはロールバック プロセスを開始する前に、アイコンをクリックして外部サービス プロパティを更新してください。

重要 vCloud Application Director 6.0、6.0.1、および Application Services 6.1 は、Amazon EC2 での既存の展開の更新をサポートしません。

依存関係があるアプリケーションで変更されたプロパティと影響を受けるプロパティを識別しやすいようにノードのクラスタ サイズを更新すると、Application Services はスケール イン/スケール アウトしたノードを変更済みとしてハイライト表示し、依存プロパティは、[更新] ウィザードの [ブループリント] ウィンドウおよびレビュー ページで、影響を受けるものとしてハイライト表示されます。依存プロパティを含むコンポーネントは、影響を受けるコンポーネントとしてハイライト表示されます。

プロパティを更新して構成を変更する場合、Application Services はプロパティを変更済みとしてハイライト表示し、依存プロパティは [更新] ウィザードの [ブループリント] ウィンドウとレビュー ページで影響を受けるものとしてハイライト表示されます。新しいプロパティ値を含むコンポーネントは変更済みコンポーネントとしてハイライト表示されます。依存プロパティを含むコンポーネントは、影響を受けるコンポーネントとしてハイライト表示されます。

更新をロールバックすると、Application Services は [ロールバック] ウィザードの [ブループリント] ウィンドウとレビュー ページで変更済みのプロパティとロールバックするプロパティをハイライト表示します。新しいプロパティ値を含むコンポーネントは、ロールバックするコンポーネントとしてハイライト表示されます。依存プロパティを含むコンポーネントは、変更済みコンポーネントとしてハイライト表示されます。

この章では次のトピックについて説明します。

- [更新プロセスの開始と展開のスケール アウト \(P. 180\)](#)
- [更新プロセスを開始して展開をスケール インする \(P. 182\)](#)
- [更新プロセスの開始と構成の変更 \(P. 185\)](#)

- [既存の更新プロファイルの使用 \(P. 187\)](#)
- [更新プロファイルの昇格 \(P. 188\)](#)
- [更新プロセスのロールバック \(P. 189\)](#)
- [カスタム タスクの実行の更新について \(P. 190\)](#)
- [展開のスケーリングに失敗した更新プロセスのトラブルシューティング \(P. 191\)](#)
- [構成を変更する更新プロセスが失敗した場合のトラブルシューティング \(P. 191\)](#)

更新プロセスの開始と展開のスケール アウト

Application Services を使用すると、たとえば、パフォーマンスを高め拡張性のあるアプリケーション展開のニーズを維持するために、既存の展開の更新プロファイルを作成してノードのクラスタをスケール アウトできます。展開済みアプリケーションのクラスタ化ノードをスケール アウトする場合、アプリケーション ブループリントでクラスタとしてモデル化されたノードのクラスタ サイズのみを構成できます。

展開済みアプリケーションをスケール アウトする更新プロセス中に、更新設定に基づいて仮想マシンが作成され、必要なアクション スクリプトが新しい仮想マシンで実行されます。多層アプリケーションでは、外部サービス以外のスケール アウトされたクラスタ化ノードに依存する場合、更新スクリプトを依存ノードで実行する必要があります。

たとえば、展開済み Clustered Dukes Bank アプリケーションでは、AppServer ノードをスケーリングして追加の負荷を処理できます。更新プロセス中に、AppServer のインストール、構成、開始スクリプトが新しくスケール アウトされた仮想マシンで実行されます。Apache_LB サービスの `http_node_ips` および `appsrv_routes` プロパティは AppServer ノードに依存しているため、AppServer クラスタ サイズの変更は Apache_LB サービスに影響を及ぼし、それにより更新スクリプトの実行が開始されます。

最初のアプリケーションの展開中に、依存するサービスまたはアプリケーション コンポーネントの UPDATE ライフ サイクル ステージ スクリプトを定義します。更新プロセス中に更新スクリプトを追加または変更することもできます。更新プロセス中に更新スクリプトを構成すると、スクリプトは将来の展開に備えて更新プロファイルに保存されます。

注意 更新プロセス中に、インストール、構成、開始スクリプトを変更できません。構成できるのは更新スクリプトのみです。

更新プロセスを開始して、コマンドライン インターフェイスから展開済みアプリケーションをスケール アウトすることもできます。[「CLI によるアプリケーションの展開と更新 \(P. 224\)」](#)を参照してください。Application Services で REST API を使用すると、展開済みアプリケーションのスケール アウトを自動化できます。[「Application Services REST API の使用」](#)のドキュメントを参照してください。

展開をスケーリングする更新プロセスは失敗することがあります。スケール アウトの更新が失敗したら、クリーンアップして新しい仮想マシンを削除する必要があります。新たに作成した仮想マシンを手動でプロビジョニング解除するのは単調で、外部サービス ユーザーにとっては難しい作業です。手動での作業を避けるには、仮想マシンの自動削除を検討します。次のフラグを `true` に設定し、スケール アウト失敗後の仮想マシンの自動クリーンアップを有効にします。

■ VM_CLEANUP_AFTER_UPDATE_FAILURE

展開後にフラグを設定し、後でスケール アウトまたはロールバックすることができます。仮想マシンを自動的にプロビジョニング解除しない場合は、このフラグを `False` に設定します。


開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#)を参照してください。
- 展開済みのアプリケーションには、少なくとも 1 つのクラスタ化ノードが含まれている必要があります。[「クラスタとしてノードを指定する \(P. 143\)」](#)を参照してください。
- カスタム タスクを追加する場合は、Application Services ライブラリに少なくとも 1 つのカスタム タスクが作成されていることを確認します。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#)を参照してください。

- クラウド環境への最初の展開が成功したことを確認します。
展開に失敗した場合や拡大/縮小操作に失敗した後に、クラスタ化ノードを拡大/縮小することはできません。
- 展開環境のストレージ領域の制限に関する詳細については、クラウド管理者にお問い合わせください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 正常に展開されているアプリケーション展開を選択します。
- 3 ツールバーの [操作] ドロップダウン メニューから、[アップデート] を選択します。
[プロファイルの更新] ページが開きます。
- 4 [アップデートプロファイルの作成] を選択します。
[更新プロファイル] ダイアログ ボックスが開きます。
- 5 [更新タイプ] ドロップダウン メニューで、[スケール アウト] を選択します。
- 6 スケール調整された更新プロセスに名前を付け、説明 (省略可能) を入力して、[OK] をクリックします。
説明には、この更新に含まれる変更内容に関する情報を追加します。
- 7 (オプション) ブループリント イメージをクリックして、サービス コンポーネントまたはアプリケーション コンポーネント間の依存関係 (強調表示された部分) を確認します。
新規の更新スクリプトを作成したり、既存の更新スクリプトを変更したりできるように、依存するコンポーネントをすべて書き留めておきます。
- 8 [ノード] タブで、1 つ以上のクラスタ化ノードのクラスタ サイズ値を増やします。
- 9 (オプション) [サービス] または [アプリケーション コンポーネント] タブから、クラスタ化ノードにバインドされたプロパティで使用可能なすべての依存コンポーネントの更新スクリプトを作成します。
UPDATE ライフ サイクル ステージスクリプトが定義されている場合は、スクリプトに関連付けられているサービス コンポーネントまたはアプリケーション コンポーネントとともにスクリプトが表示されます。スクリプトが定義されていない場合、更新プロセスに適用可能なスクリプトを作成します。
たとえば、クラスタ化されたノードにバインドされている `all(node_array:ip)` プロパティがノードにある場合は、更新スクリプトを実行する必要があります。
- 10 (オプション) 更新スクリプトが正常に実行された後にエージェントが仮想マシンを再起動する必要がある場合は [再起動] チェック ボックスをオンにして、[次へ] をクリックします。
スケール アウトされたクラスタ化ノードと依存コンポーネントの更新タスクが実行プランに表示されます。更新プロセス中は、元の実行プランが表示されません。
- 11 (オプション) 更新スクリプトに `APPD_UPDATE_PROPS` プロパティを追加して、変更されたすべてのプロパティのリストを表示します。
更新スクリプトは、更新プロファイルに保存され、複数の更新に使用できます。`APPD_UPDATE_PROPS` をプロパティ名またはプロパティ名の修飾子として使用することはできません。
たとえば、サンプルの MySQL サービスの更新スクリプトには、データベース ポートおよびパスワードを更新するための `APPD_UPDATE_PROPS` プロパティが含まれています。
- 12 (オプション) 更新スクリプトに `APPD_PREV` プロパティを追加して、プロパティの前の値を表示します。
更新スクリプトは、更新プロファイルに保存され、複数の更新に使用できます。プロパティ名またはプロパティ名の修飾子として `APPD_PREV` は使用できません。
たとえば、サンプルの MySQL サービスの更新スクリプトには、データベース パスワードの値を表示するための `APPD_PREV` プロパティが含まれています。

- 13 (オプション) [クラスタの展開] ボタン  をクリックしてクラスタ化ノードを展開し、カスタム タスクを選択して、タスクを各ノードにドラッグします。

[カスタム タスクの追加] ダイアログ ボックスでタスク プロパティを構成して、変更内容を保存できます。

- 14 実行プランでスケール アウトしたクラスタ化ノードを確認し、必要があれば更新スクリプトも確認して [次へ] をクリックします。

実行プランの青色の点線は、展開タスクを実行する特定の順序を示しています。

サービス更新スクリプトの横にある下矢印をクリックすると、スクリプトの詳細またはスクリプトで使用される変数の定義が表示されます。

- 15 変更されたプロパティおよび更新のアクションを確認します。

更新スクリプトが定義された依存プロパティがハイライト表示されます。

- 16 [アップデート] をクリックして、更新されたアプリケーションを展開します。

更新プロセスは、スケール アウトした更新をクラウドに展開します。

次に進む前に

展開のステータスは、[展開サマリ] ページで確認できます。[\[展開サマリ\] ページの使用 \(P. 174\)](#) を参照してください。

展開済みアプリケーションがクラウドで更新されるときにバックグラウンドで実行されるさまざまなプロセスについて理解しておく必要があります。[\[展開および更新プロセスについて \(P. 175\)\]](#) を参照してください。

更新プロセスを開始して展開をスケール インする

Application Services を使用すると、既存の展開の更新プロファイルを作成することで、たとえば、1 つ以上のノードで構成されるクラスタをスケール インして、未使用のリソースを解放して拡張性のあるアプリケーションでの変化する負荷に対応することができます。展開済みアプリケーションのクラスタ化ノードをスケール インする場合、アプリケーションブループリントでクラスタとしてモデル化されたノードのクラスタ サイズのみを構成できます。

展開済みアプリケーションをスケール インする更新プロセスでは、更新設定に基づき、更新スクリプトが各依存コンポーネントに対して実行され、その後、分解アクション スクリプトによってノードが削除されます。多層アプリケーションで、スケール インされたクラスタ化ノードに依存するノードが存在する場合は、更新スクリプトをその依存ノードに対して実行する必要があります。

たとえば、展開済みの Clustered Dukes Bank App で、appServer ノードをスケール インして負荷を軽減できます。更新プロセスでは、更新スクリプトが、依存する load_balancer ノードに対して実行されます。また、分解アクション スクリプトが、展開から appServer_2 ノードと appServer_3 ノードを削除します。このアクション スクリプトはクラウド プロバイダ API も呼び出して、クラウド環境から上記のノードを削除します。

スケールイン時に、カスタム タスクをスケールイン ノードに加えて更新を実行できます。ただし、このスケールインの更新プロファイルが再利用されている場合は、関連付けられたカスタム タスクはこのノードに含まれません。最初のスケールイン時に、カスタム タスク情報がスケールイン ノードとともに削除されているためです。したがって、スケールインを再利用する場合は、カスタム タスクを使用できません。

最初にアプリケーションを展開する際に、依存するサービスおよびアプリケーション コンポーネントに対して UPDATE および TEARDOWN ライフ サイクル ステージ スクリプトを定義します。更新スクリプトは依存ノードに対して、分解スクリプトはスケール イン ノードに対して実行されます。更新プロセス中に更新および分解スクリプトを追加または変更することもできます。更新プロセス中に更新または分解スクリプトを構成すると、そのスクリプトは将来の展開に備えて更新プロファイルに保存されます。

注意 更新プロセス中に、インストール、構成、開始スクリプトを変更できません。構成できるのは更新スクリプトと分解スクリプトのみです。

Application Services で REST API を使用すると、展開済みアプリケーションのスケール インを自動化できます。[\[Application Services REST API の使用\]](#) のドキュメントを参照してください。

展開をスケール インする更新プロセスは失敗することがあります。スケール イン更新プロセスが失敗したら、クリーンアップして新しい仮想マシンのプロビジョニング解除を再試行する必要があります。新たに作成した仮想マシンを手動でプロビジョニング解除するのは単調で、外部サービス ユーザーにとっては難しい作業です。手動での作業を避けるには、仮想マシンの自動削除を検討します。次のフラグを True に設定すると、最初の試行で削除に失敗した場合に仮想マシンのクリーンアップが再試行されます。

■ UPDATE_RETRY_VM_DEPROVISIONING_AFTER_FAILURE_FLAG

次のフラグを True に設定すると、数ミリ秒後に仮想マシンの削除が再試行されます。

■ UPDATE_RETRY_VM_DEPROVISIONING_AFTER_FAILURE_DELAY_INTERVAL

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- 展開済みのアプリケーションには、少なくとも 1 つのクラスタ化ノードが含まれている必要があります。[「クラスタとしてノードを指定する \(P. 143\)」](#) を参照してください。
- カスタム タスクを追加する場合は、Application Services ライブラリに少なくとも 1 つのカスタム タスクが作成されていることを確認します。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。
- クラウド環境への最初の展開が成功したことを確認します。
展開に失敗した場合や拡大/縮小操作に失敗した後に、クラスタ化ノードを拡大/縮小することはできません。
- 展開環境のストレージ領域の制限に関する詳細については、クラウド管理者にお問い合わせください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして **[展開]** を選択します。
- 2 正常に展開されているアプリケーション展開を選択します。
- 3 ツールバーの **[操作]** ドロップダウン メニューから、**[アップデート]** を選択します。
[プロファイルの更新] ページが開きます。
- 4 **[アップデートプロファイルの作成]** を選択します。
[更新プロファイル] ダイアログ ボックスが開きます。
- 5 **[更新タイプ]** ドロップダウン メニューで、**[スケール イン]** を選択します。
- 6 スケール調整された更新プロセスに名前を付け、説明（省略可能）を入力して、**[OK]** をクリックします。
説明には、この更新に含まれる変更内容に関する情報を追加します。
- 7 (オプション) ブループリント イメージをクリックして、サービス コンポーネントまたはアプリケーション コンポーネント間の依存関係（強調表示された部分）を確認します。
新規の更新スクリプトを作成したり、既存の更新スクリプトを変更したりできるように、依存するコンポーネントをすべて書き留めておきます。
- 8 **[ノード]** タブで、クラスタ サイズの値を減らします。
Application Services は、最近作成されたノードを選択します。
たとえば、appserver_0、appserver_1、appserver_2、appserver_3 の各ノードで構成されるクラスタで、クラスタ サイズを 2 ノードに減らした場合、appserver_2 と appserver_3 が削除対象ノードとして選択されます。
- 9 (オプション) **[削除するノードの選択]** ボタンをクリックしてクラスタ内のすべてのノードを一覧表示し、削除するノードを手動で選択します。

- 10 (オプション) [サービス] または [アプリケーション コンポーネント] タブから、クラスタ化ノードにバインドされたプロパティで使用可能なすべての依存コンポーネントの更新スクリプトを作成します。

UPDATE および TEARDOWN ライフ サイクル ステージが定義されている場合は、スクリプトが、関連付けられたサービス コンポーネントまたはアプリケーション コンポーネントと共に表示されます。

たとえば、ある Load Balancer ノードで all(node_array:ip) プロパティが特定のクラスタ化ノードにバインドされている場合、更新スクリプトを実行する必要があります。

- 11 (オプション) 更新スクリプトが正常に実行された後にエージェントが仮想マシンを再起動する必要がある場合は [再起動] チェック ボックスをオンにして、[次へ] をクリックします。

スケール インされたクラスタ化ノードと依存コンポーネントの更新タスクが実行プランに表示されます。更新プロセス中は、元の実行プランが表示されません。

- 12 (オプション) 更新スクリプトに APPD_UPDATE_PROPS プロパティを追加して、変更されたすべてのプロパティのリストを表示します。


更新スクリプトは、更新プロファイルに保存され、複数の更新に使用できます。APPD_UPDATE_PROPS をプロパティ名またはプロパティ名の修飾子として使用することはできません。

たとえば、サンプルの MySQL サービスの更新スクリプトには、データベース ポートおよびパスワードを更新するための APPD_UPDATE_PROPS プロパティが含まれています。

- 13 (オプション) 更新スクリプトに APPD_PREV プロパティを追加して、プロパティの前の値を表示します。

更新スクリプトは、更新プロファイルに保存され、複数の更新に使用できます。プロパティ名またはプロパティ名の修飾子として APPD_PREV は使用できません。

たとえば、サンプルの MySQL サービスの更新スクリプトには、データベース パスワードの値を表示するための APPD_PREV プロパティが含まれています。

- 14 (オプション) [クラスタの展開] ボタン  をクリックしてクラスタ化ノードを展開し、カスタム タスクを選択して、タスクを各ノードにドラッグします。

[カスタム タスクの追加] ダイアログ ボックスでタスク プロパティを構成して、変更内容を保存できます。

- 15 スケール インされたクラスタ化ノードと更新スクリプトを実行プランで確認して、[次へ] をクリックします。

実行プランの青色の点線は、展開タスクを実行する特定の順序を示しています。

サービス更新スクリプトの横にある下矢印をクリックすると、スクリプトの詳細またはスクリプトで使用する変数の定義が表示されます。

- 16 変更されたプロパティおよび更新のアクションを確認します。

更新スクリプトが定義された依存プロパティがハイライト表示されます。

- 17 [アップデート] をクリックして、更新されたアプリケーションを展開します。

更新プロセスによって、スケール イン更新がクラウドに展開されます。

次に進む前に

展開のステータスは、[展開サマリ] ページで確認できます。[\[展開サマリ\] ページの使用 \(P. 174\)](#) を参照してください。

展開済みアプリケーションがクラウドで更新されるときにバックグラウンドで実行されるさまざまなプロセスについて理解しておく必要があります。[\[展開および更新プロセスについて \(P. 175\)\]](#) を参照してください。

更新プロセスの開始と構成の変更

更新プロセスを開始して、Tomcat や MySQL などの展開済みアプリケーションの既存のサービスまたは WAR や SQL などのアプリケーション コンポーネントの構成やコードを変更できます。展開済みアプリケーションの構成を変更する場合、アプリケーション プロパティ値のみを構成できます。

初期展開の後で、既存のアプリケーション コンポーネントやサービスを変更したり、依存プロパティがあるコンポーネントを変更したりする場合、すべての影響を受ける依存コンポーネントで更新スクリプトを実行する必要があります。初期アプリケーション展開中、または更新プロセスで、サービスまたはアプリケーション コンポーネントの UPDATE および ROLLBACK ライフ サイクル ステージ スクリプトを定義します。

たとえば、Clustered Dukes Bank App で MySQL サービスの db_port プロパティを変更する場合、展開更新プロセス中に更新スクリプトが MySQL サービスで実行されます。Appserver ノードの Dukes_Bank_App アプリケーション コンポーネントも更新スクリプトを実行します。これは、db_port プロパティが MySQL サービスの db_port プロパティにバインドされているためです。

注意 更新プロセス中は、インストール、構成、または開始スクリプトを変更できません。構成できるのは更新またはロールバック スクリプトのみです。

更新プロセスを開始して、コマンドライン インターフェイスからサービスまたはアプリケーション コンポーネントの構成およびコードを変更することもできます。[「CLI によるアプリケーションの展開と更新 \(P. 224\)」](#) を参照してください。Application Services の REST API を使用して、展開済みアプリケーションの構成の変更を自動化できます。『Application Services REST API の使用』 のドキュメントを参照してください。

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- カスタム タスクを追加する場合は、Application Services ライブラリに少なくとも 1 つのカスタム タスクが作成されていることを確認します。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。
- クラウド環境への最初の展開が成功したことを確認します。

更新プロセスを起動して既存のサービスの構成とコードを変更し、クラスタ化ノードをスケール調整するには、展開が成功している必要があります。

- 展開対象のアプリケーションには、展開時にオーバーライド可能なサービス プロパティまたはアプリケーション コンポーネント プロパティが少なくとも 1 つ定義されている必要があります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして **[展開]** を選択します。
- 2 正常に展開されているアプリケーション展開を選択します。
- 3 ツールバーの **[操作]** ドロップダウン メニューから、**[アップデート]** を選択します。
[プロファイルの更新] ページが開きます。
- 4 **[アップデートプロファイルの作成]** を選択します。
[更新プロファイル] ダイアログ ボックスが開きます。
- 5 **[更新タイプ]** ドロップダウン メニューで、**[構成]** を選択します。
- 6 構成更新プロセスに名前を付け、説明を入力して、**[OK]** をクリックします。
説明には、この更新に含まれる変更内容に関する情報を追加します。

- 7 ブループリント イメージをクリックして、サービス コンポーネントまたはアプリケーション コンポーネント間の依存関係（強調表示された部分）を確認します。

新規の更新スクリプトを作成したり、既存の更新スクリプトを変更したりできるように、依存するコンポーネントをすべて書き留めておきます。

- 8 [サービス] または [アプリケーション コンポーネント] タブから、適切なプロパティおよび更新スクリプトを変更します。

UPDATE および ROLLBACK ライフ サイクル ステージが定義されている場合は、スクリプトが、関連付けられたサービスまたはアプリケーション コンポーネントと共に表示されます。変更済みの依存コンポーネントの更新スクリプトを作成する必要があります。

クラスタ化されたノードでサービス コンポーネントまたはアプリケーション コンポーネントのプロパティを変更したときは、変更内容がクラスタのすべてのノードに適用されます。

たとえば、Clustered Dukes Bank App で initialize_db_script アプリケーション コンポーネントの db_password プロパティを変更する場合、Dukes_Bank_App の db_password プロパティは initialize_db_script アプリケーション コンポーネントに依存します。initialize_db_script は変更されたコンポーネントであり、Dukes_Bank_App は影響を受けるコンポーネントなので、更新スクリプトが両方で実行されます。Dukes_Bank_App の更新スクリプトは、Appserver クラスタのすべてのノードで実行されます。

- 9 (オプション) 更新スクリプトが正常に実行された後にエージェントが仮想マシンを再起動する必要がある場合は [再起動] チェック ボックスをオンにして、[次へ] をクリックします。

変更され影響を受けるコンポーネントの更新タスクは、実行プランに表示されます。更新プロセス中は、元の実行プランが表示されません。

- 10 (オプション) 更新スクリプトに APPD_UPDATE_PROPS プロパティを追加して、変更されたすべてのプロパティのリストを表示します。

更新スクリプトは保存されず、特定の更新プロセスにのみ適用されます。APPD_UPDATE_PROPS をプロパティ名またはプロパティ名の修飾子として使用することはできません。

たとえば、サンプルの MySQL サービスの更新スクリプトには、データベース ポートおよびパスワードを更新するための APPD_UPDATE_PROPS プロパティが含まれています。

- 11 (オプション) 更新スクリプトに APPD_PREV プロパティを追加して、プロパティの前の値を表示します。

更新スクリプトは保存されず、特定の更新プロセスにのみ適用されます。プロパティ名またはプロパティ名の修飾子として APPD_PREV は使用できません。

たとえば、サンプルの MySQL サービスの更新スクリプトには、データベース パスワードの値を表示するための APPD_PREV プロパティが含まれています。

- 12 (オプション) [クラスタの展開] ボタン  をクリックしてクラスタ化ノードを展開し、カスタム タスクを選択してそのタスクを各ノードにドラッグします。

クラスタ化ノードでサービス プロパティまたはアプリケーション コンポーネント プロパティを変更した場合、変更はすべてのノードに適用されます。[カスタム タスクの追加] ダイアログ ボックスでタスク プロパティを構成して、変更内容を保存できます。

- 13 変更したクラスタ化ノードと更新スクリプトを更新実行プランで確認します。

実行プランの青色の点線によって、展開タスクの実行順が示されます。

構成更新スクリプトの横にある下矢印をクリックし、スクリプトの詳細またはスクリプトで使用される変数定義を表示します。

- 14 変更したクラスタ化ノードと更新スクリプトをロールバック実行プランで確認し、[次へ] をクリックします。

実行プランの青色の点線によって、展開タスクの実行順が示されます。

構成更新スクリプトの横にある下矢印をクリックし、スクリプトの詳細またはスクリプトで使用される変数定義を表示します。

15 更新によって変更されるコンポーネントおよび影響を受けるコンポーネントを確認します。

変更されるプロパティおよびそれに依存するプロパティがハイライト表示されます。

16 [アップデート] をクリックして、アプリケーションの変更された構成を展開します。

更新プロセスでは、構成済み展開がクラウドに展開されます。

次に進む前に

[展開サマリ] ページで展開の状態を確認します。[「\[展開サマリ\] ページの使用 \(P. 174\)」](#) を参照してください。

展開済みアプリケーションがクラウドで更新されるときにバックグラウンドで実行されるプロセスについて理解しておく必要があります。[「展開および更新プロセスについて \(P. 175\)」](#) を参照してください。

既存の更新プロファイルの使用

展開者は、既存の更新プロファイルを使用して、前回展開したアプリケーションを同じクラウド環境で複数回更新できます。更新プロファイルには、最初の更新プロセスでよく使用されるプロパティとスクリプトが保存されています。

保存した更新プロファイルは、アプリケーション バージョンの展開に対してのみ適用できます。

CLI では、新規または既存の更新プロファイルを保存することはできません。CLI を使用して既存の更新プロファイルを適用する際に、更新プロセスによって構成の変更やクラスタ化ノードの拡大/縮小を行う場合は、スクリプトまたはプロパティをオーバーライドできません。

CLI を使用してアプリケーションを展開する場合は、更新プロファイルを選択しても更新プロファイル名のオート コンプリート オプションは表示されません。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 少なくとも 1 つの展開が Application Services で正常に更新されていることを確認します。

手順

1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。

2 正常に更新される展開を選択します。

[展開サマリ] ページが開きます。

3 展開サマリの上のツールバーで、[展開ビュー] - [アップデートビュー] を選択します。

使用できる更新プロファイル、アプリケーション名とバージョンなどの展開に関連するすべての詳細は、[更新の概要] ページに表示されます。

4 使用する更新プロファイルのタイプを検索し、クリックします。

更新プロファイルは、スケール アウト、スケール イン、または構成の各更新プロセス向けに作成できます。各更新プロファイルには、そのプロファイルによって実行された展開更新の成否の回数が表示されます。

[展開プロファイル] ウィザードが開きます。

次に進む前に

タスクを最後まで実行して展開を更新します。[「更新プロセスの開始と展開のスケール アウト \(P. 180\)」](#)、[「更新プロセスを開始して展開をスケール インする \(P. 182\)」](#)、および [「更新プロセスの開始と構成の変更 \(P. 185\)」](#) を参照してください。

更新プロファイルの昇格

展開者は、既存の更新プロファイルを昇格して、複数の展開環境にまたがる展開を更新できます。同じアプリケーションバージョンから作成された複数の展開に同じタイプの更新プロセスを適用できます。

たとえば、展開者は、同じアプリケーションバージョンから作成された異なる展開に対して、セキュリティに関連する構成の更新をアプリケーションの 1 つ以上のコンポーネントに適用できます。

REST API を使用して異なる展開環境にまたがって更新プロファイルを昇格することはサポートされていません。


開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 正常に更新される展開を選択します。
[展開サマリ] ページが開きます。
- 3 展開サマリの上のツールバーで、[展開ビュー] - [アップデートビュー] を選択します。
使用できる更新プロファイル、アプリケーション名とバージョンなどの展開に関連するすべての詳細は、[更新の概要] ページに表示されます。
- 4 昇格する既存の更新プロファイルの場所を特定します。
- 5 更新プロファイルを 1 つ以上の展開環境に昇格させるには、[アップデートプロファイルの昇格] アイコンをクリックします。
[更新昇格の管理] ダイアログ ボックスが開き、使用できる展開環境のリストが表示されます。
- 6 [選択可能] 列で展開環境を選択し、[選択に追加] 矢印をクリックします。
展開環境が [選択済み] 列に表示されます。
- 7 (オプション) [選択済み] 列から展開環境を取り下げる場合は、展開環境を選択し、[選択から削除] 矢印をクリックします。
展開環境が [選択可能] 列に表示されます。
- 8 [選択済み] 列の展開環境を確認し、[送信] をクリックします。

更新プロファイルが選択された展開環境に正常に昇格すると、更新プロファイルの概要カードに緑色の成功のアイコンが表示されます。

単一または複数の展開環境への展開に通知アイコン  が表示されます。たとえば、Clustered Dukes Bank の更新プロファイルが昇格されると、選択された展開環境内の既存の Clustered Dukes Bank アプリケーションのすべての展開に通知が表示されます。

次に進む前に

選択された展開環境の 1 つに属している展開に昇格された更新プロファイルを適用します。[「昇格された更新プロファイルを承諾または破棄する \(P. 189\)」](#) を参照してください。

昇格された更新プロファイルを承諾または破棄する


展開者は、通知をクリックして昇格された更新プロファイルのコンポーネントを確認し、変更を承諾するか破棄するかを判断できます。変更を承諾する場合、Application Services で変更を適用する前に、昇格された更新プロファイルと展開との互換性チェックを実行します。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 少なくとも 1 つの更新プロファイルが 1 つ以上の展開環境に昇格されることを確認します。[「更新プロファイルの昇格 \(P. 188\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。

- 2 通知アイコン  のある展開を見つけ、そのアイコンをクリックします。


更新のサマリ ページが開きます。

- 3 [適用] をクリックします。

Application Services で、昇格された更新プロファイルと現在の更新される展開との互換性を確認します。深刻な非互換性が見つかった場合、ユーザーは通知を中止または破棄できます。展開内に同じ名前の更新プロファイルが存在するなど、深刻でない非互換性が見つかった場合は、適用を続けるか、通知を破棄または中止できます。

- 4 ウィザードの [アプリケーション プロパティ] タブおよび [実行プラン] タブで新しい値または構成を確認し、[更新の適用] を選択します。

- 5 (オプション) 昇格された更新プロファイルの新しい値または構成を適用しない場合は、[更新の破棄] を選択します。

昇格された更新プロファイルが破棄され、通知アイコン  が展開から削除されます。

次に進む前に

通知が表示された別の展開を選択し、昇格された更新プロファイルの確認および適用、または破棄を続けることができます。

更新プロセスのロールバック

Application Services では、構成を変更する更新プロセスが失敗したり、更新された展開にパフォーマンスまたは可用性の面で問題が生じたりした場合、ロールバック機能を使用して、最後に成功した更新の直後の状態に展開を戻すことができます。

ロールバック スクリプトは、(ロールバックすると) プロパティが変更されるコンポーネント、およびその他の依存ノードに対して実行されます。たとえば、展開済みの Clustered Dukes Bank App で MySQL のポート番号の設定を 3306 から 3307 に変更した場合、mysql サービスと dukes_bank_app コンポーネントに対してロールバック スクリプトが実行されます。この更新プロセスに失敗した場合、更新した内容をロールバックして前のポート値 3306 に戻すことができます。[Rollback] ウィザードでは、[現在の値] が 3307、[ロールバック後の値] が 3306 と表示されます。ロールバック スクリプトが mysql サービスと dukes_bank_app コンポーネントに対して実行され、処理が完了します。

最初にアプリケーションを展開するとき、依存するサービスやアプリケーション コンポーネントに対して、ROLLBACK ライフ サイクル ステージ スクリプトを定義します。また、更新プロセスをロールバックして構成を変更する場合、ロールバック スクリプトを追加または変更することもできます。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- クラウド環境への最初の展開が成功したことを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 前回成功した状態にロールバックするアプリケーション展開を選択します。
- 3 ツールバーの [操作] ドロップダウン メニューから、[ロールバック] を選択します。
- 4 ブループリント イメージをクリックして、サービス コンポーネントまたはアプリケーション コンポーネント間の依存関係（強調表示された部分）を確認します。

新規のロールバック スクリプトを作成したり、既存のロールバックを変更したりできるように、依存するコンポーネントをすべて書き留めておきます。
- 5 [サービス] または [アプリケーション コンポーネント] タブで、該当するプロパティとロールバック スクリプトを変更します。

ROLLBACK ライフサイクル ステージが定義されている場合は、スクリプトと、そのスクリプトに関連付けられたサービス コンポーネントまたはアプリケーション コンポーネントがいっしょに表示されます。コンポーネントの変更とロールバックについて、ロールバック スクリプトを作成する必要があります。

クラスタ化されたノードでサービス コンポーネントまたはアプリケーション コンポーネントのプロパティを変更したときは、変更内容がクラスタのすべてのノードに適用されます。

外部サービス インスタンスの変更によってトリガされた展開の構成プロパティを更新した場合、更新後のプロパティをロールバックすることはできません。
- 6 (オプション) ロールバック スクリプトが正常に終了した後、エージェントが仮想マシンを再起動する必要がある場合は、[再起動] チェック ボックスをオンにして、[次へ] をクリックします。

ロールバック対象コンポーネントのロールバック タスクが実行プランに表示されます。ロールバック プロセス中は、元の実行プランは表示されません。
- 7 実行プランの変更対象クラスタ化ノードとロールバック スクリプトを確認して、[次へ] をクリックします。

実行プランの青色の点線によって、展開タスクの実行順が示されます。

構成ロールバック スクリプトの横にある下矢印をクリックすると、スクリプトの詳細およびスクリプトで使用されている変数定義が表示されます。
- 8 更新プロセスのロールバック対象コンポーネントを確認します。

ロールバック対象プロパティは強調表示されます。
- 9 [ロールバック] をクリックします。

ロールバック プロセスによって、クラウド内で最後に成功した状態に展開が戻されます。

次に進む前に

[展開サマリ] ページで展開の状態を確認します。[\[展開サマリ\] ページの使用 \(P. 174\)](#) を参照してください。

カスタム タスクの実行の更新について

カスタム タスクの実行の更新により、既存の展開環境の仮想マシン上で一連の任意のカスタム タスクを実行できます。

カスタム タスクの実行の更新によって、次のタスクを実行できます。

- [仮想マシン レベルで操作する]。コンポーネントおよびサービスで複数のコンポーネントを同時に操作できます。
- [プロパティを変更せずにスクリプトを実行する]。既存の展開環境でサービスやコンポーネントのプロパティを変更せずに、スクリプトを実行できます。

REST API を介してカスタム タスクの実行を起動するには、次の情報を指定する必要があります。

- 実行する必要のあるすべてのタスク

- 各タスクの入力パラメータ
- 各タスクが実行されるノード
- ノード内の各タスクの実行の依存関係。たとえば、1 つ以上のタスクが、他のタスクの実行と完了を待機する場合があります。

カスタム タスクの実行の更新が開始すると、タスクは展開の概要ページに表示されます。展開タイムラインには、「カスタム タスクの実行の更新」を現在実行しているノードが示されます。

カスタム タスクの実行では、ロールバック操作がサポートされます。カスタム タスクの実行に失敗した場合、または以前行った変更に戻す場合は、新たにカスタム タスクを実行してその変更に戻す必要があります。これは、Application Services で任意のカスタム タスクのロールバック スクリプトを生成できないためです。

カスタム タスクの実行を管理するための新しい REST API が 2 つ導入されました。

- カスタム タスクの実行の更新プロファイルを作成およびスケジュール設定
- カスタム タスクの実行の更新プロファイルを更新およびスケジュール設定

REST API の詳細については、『Application Services REST API の使用』を参照してください。

展開のスケーリングに失敗した更新プロセスのトラブルシューティング

展開をスケーリングする更新プロセスが失敗した場合は、既知の解決方法および推奨事項が参考になります。

問題

展開をスケーリングする更新プロセスに失敗した。

原因

以前、スケーリングに失敗した展開に対して、展開をスケーリングする更新プロセスを実行しようとした可能性があります。

解決方法

- 1 クラウド管理者は、クラスタ化されたすべての展開ノードについて、新規の仮想マシンを削除する必要があります。
- 2 クラウド管理者は、更新プロセスでのスケーリング処理中に削除されなかった仮想マシンを削除する必要があります。
- 3 前回失敗した展開に対して新たにスケール アウト更新プロセスを起動します。

Application Services では、以前に適用され、成功したプロパティ値が事前に入力されています。新しい値は、以前の失敗した展開から取得されています。

たとえば、AppServer1 のクラスタ化ノード数を 1 から 2 に拡大し、AppServer2 のクラスタ化ノード数を 1 から 3 に拡大したとき、AppServer2 ノードが原因で更新プロセスに失敗したとします。2 回目の更新プロセスでは、AppServer1 の値に 2 が設定され、AppServer2 ノードには前回よりも大きな値を指定します。

- 4 展開に失敗して仮想マシンを削除した場合は、新たにスケール イン更新プロセスを起動します。
- 5 すべての失敗したノードをスケール アウトする更新プロセスを新たに起動します。更新スクリプトがロード バランサなどのすべての依存コンポーネントで実行され、各コンポーネントが有効になります。

構成を変更する更新プロセスが失敗した場合のトラブルシューティング

構成を変更する更新プロセスが失敗した場合は、いくつかのソリューションを適用できます。

問題

構成を変更する更新プロセスが失敗します。

原因

ポート番号などの失敗したプロパティをリセットすることにより、以前に失敗した更新の構成を変更するための更新プロセスを開始して続行しようとしています。

解決方法

- 別の更新プロセスを開始し、以前に失敗した更新の構成を変更してください。

Application Services では、以前に適用され、成功したプロパティ値が事前に入力されています。新しい値は、以前の失敗した展開から取得されています。

別の更新プロセスを開始して構成を変更する場合は、失敗したすべてのプロパティを手動で更新する必要があります。

- 更新スクリプトを作成または変更し、失敗したプロパティを無視します。

失敗したすべてのプロパティを更新しない場合は、失敗したプロパティを無視する更新スクリプトを記述して更新プロセスを続行することもできます。

変更された更新スクリプトは保存されずに、その特定の更新プロセスにのみ適用されます。

事前定義済みライブラリ コンポーネントの展開

18

Application Services ライブラリには、事前定義済みのサンプル サービス、論理テンプレート、アプリケーション コンポーネント、およびタスクが含まれています。

サンプル アプリケーションは、Windows または Linux ベースの論理テンプレート、サービス、およびそのアプリケーション専用で構成されたアプリケーション コンポーネントで構成されます。事前定義済みアプリケーションに含まれるサービスのアクション スクリプトのみを構成して、サポートされているクラウド環境に展開する必要があります。展開プロファイルを作成してサンプル アプリケーションを展開する場合は、互換性のあるカスタマイズ済みタスクを実行プランに追加できます。

重要 事前定義済みのサンプルテンプレート、アプリケーション コンポーネント、サービス、およびタスクは、テスト環境でのみ使用してください。サンプルテンプレートを本番環境で使用することを計画している場合は、テンプレート内のオペレーティングシステムに最新のセキュリティ パッチを適用してから展開するようにしてください。

Application Services の Default グループに割り当てられたどのユーザー アカウントも、サンプル アプリケーションにアクセスして展開を実行できます。

ライブラリ内のサンプルのテンプレートとサービスを表すすべてのアイコンは、そのテンプレートまたはサービスの名前に基づいています。たとえば、ライブラリ内の MySQL サービスの横には、正式な [MySQL] アイコンが表示されます。サービスまたはテンプレートの名前が固有の場合は、汎用のアイコンが表示されます。

サンプルのライブラリ コンポーネントの展開に関するトピックに頻出する主要概念について理解しておく必要があります。「[主要な概念 \(P. 16\)](#)」を参照してください。

この章では次のトピックについて説明します。

- [サンプル テンプレートの使用 \(P. 193\)](#)
- [展開に使用できるサンプル アプリケーション \(P. 194\)](#)
- [展開されたサンプル アプリケーションを更新する \(P. 200\)](#)
- [サンプルの Application Services ライブラリ タスク \(P. 202\)](#)
- [サンプルのライブラリ サービス \(P. 208\)](#)

サンプル テンプレートの使用

Application Services は、オペレーティングシステムがインストールされた CentOS および Ubuntu のサンプル テンプレートと、アプリケーションの展開に必要なすべてのライブラリを提供します。

たとえば、CentOS サンプルテンプレートを使用するには、VMware 製品ダウンロード サイトから次のファイルをダウンロードします。

- CentOS 6.3 32 ビット OVF
- CentOS 6.3 32 ビット VMDK

使用可能なサンプル テンプレートのリストについては、Application Services の Readme ファイルを参照してください。

CentOS 6.3 32 ビット OVF パッケージを vCloud Director カタログまたは vRealize Automation にアップロードします。vCloud Director カタログまたは vRealize Automation の詳細については、それぞれのドキュメントを参照してください。

注意 CentOS 32 ビット サンプル テンプレートの場合は物理アドレス拡張 (PAE) が有効になっていないため、最大 3.25 GB の RAM を仮想マシンに割り当てることができます。Ubuntu 32 ビット サンプル テンプレートの場合は PAE が有効になっているため、仮想マシンに 4GB を超える RAM を割り当てることができます。

libpython パッケージを Ubuntu テンプレートに追加する

サンプルの Ubuntu テンプレートが正常に機能するには、libpython XML パッケージを追加する必要があります。

手順

- 1 python パッケージを Ubuntu ミラー サイトからダウンロードします。
`wget http://<UbuntuMirrorSite>/libpython2.7_2.7.3-0ubuntu3.4_amd64.deb`
- 2 python パッケージを Ubuntu テンプレートにインストールします。
`dpkg -i libpython2.7_2.7.3-0ubuntu3.4_amd64.deb`

Ubuntu 仮想マシンの SSH 接続を有効にする

Ubuntu 仮想マシンが正しく動作するためには、SSH ホスト キーが生成されている必要があります。

開始する前に

ブラウザを使用するために、vCloud Director VMRC プラグインがインストールされていることを確認します。互換性のある Web ブラウザの詳細については、[「Application Services のシステム要件 \(P. 24\)」](#) を参照してください。

手順

- 1 Ubuntu OVF パッケージを vCloud Director カタログにアップロードします。
- 2 vCloud Director VMRC コンソール内をクリックします。
- 3 Ubuntu 仮想マシンにログインします。
- 4 SSH 接続を有効にします。
`ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key`
`"ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key"`
- 5 SSH サーバを再起動します。
`service ssh restart`

Ubuntu 仮想マシンに対する SSH 接続が有効になります。

展開に使用できるサンプル アプリケーション

事前定義済みの単一階層または 3 階層のアプリケーションを使用して、単純な Web アプリケーションまたは複雑な電子メール スイートを展開できます。サンプル アプリケーションのアプリケーション ブループリントには、クラウドに展開されたときに完全に機能する、論理テンプレート、サービス、外部サービス、およびアプリケーション コンポーネントが含まれています。

アプリケーション アーキテクトは、Application Services のサンプル アプリケーションをベースとして使用し、カスタム アプリケーションをモデル化および作成できます。

Nanotrader アプリケーションを展開する

サンプルの Nanotrader アプリケーションは 3 階層の Web アプリケーションで、vFabric Web Server、vFabric tc Server、vFabric RabbitMQ、および vFabric SQLFire コンポーネントを使用して NanoTrader アプリケーションを展開します。

事前定義済みサンプル アプリケーションはテスト環境でのみ使用してください。

開始する前に

- アップロードした CentOS 6.3 (32 ビット) テンプレートをクラウド プロバイダに登録します。[\[vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)\]](#)、[\[vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)\]](#)、または [\[Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)\]](#) を参照してください。
- クラウド テンプレートを CentOS 6.3 (32 ビット) 論理テンプレートにマッピングします。[\[ライブラリへの論理テンプレートの追加 \(P. 125\)\]](#) を参照してください。
- アプリケーション アーキテクトおよびアプリケーションの公開元および配備者として Application Services にログインします。
- 展開プロファイルの作成とアプリケーションの展開に関する基本概念を理解しておきます。[第 16 章「アプリケーションの展開 \(P. 157\)」](#) を参照してください。
- `spring-nanotrader-asynch-services.war`、`spring-nanotrader-services.war`、`spring-nanotrader-web.tgz`、および `spring-nanotrader-web.war` ファイルが HTTPD サーバにダウンロードされていることを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 [Nanotrader] を選択します。
- 3 画面上部のブループリント スクリーンショットをクリックします。
- 4 ブループリント エディタでサービスと SCRIPT コンポーネントを選択し、URL 値を追加して適切なファイルをダウンロードします。

サービスとコンポーネント	説明
[vFabric_Web_Server]	deployment_archive プロパティ用の <code>spring-nanotrader-web.tgz</code> ファイル
[vFabric_tc_Server]	external_template プロパティ用の <code>spring-nanotrader-web.war</code> ファイル
[spring-nanotrader-services]	war_file プロパティ用の <code>spring-nanotrader-services.war</code> ファイル

- 5 [保存] をクリックします。
- 6 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 7 構成した Nanotrader アプリケーションを開きます。
- 8 展開プロファイルを作成し、アプリケーションを展開します。
- 9 展開されたアプリケーションにアクセスするには、サポートされる Web ブラウザを開いて **`http://<Load_Balancer_IP>/#login`** という URL を入力します。
ここで、<Load_Balancer_IP> は、展開済みロード バランサーの IP アドレスです。
- 10 NanoTrader アカウントを作成してユーザー名とパスワードを設定します。

次に進む前に

更新プロセスを開始して、Nanotrader アプリケーションの構成をスケール調整または変更します。[「サンプル アプリケーションをスケールリングする更新プロセスの開始 \(P. 200\)」](#) および [「更新プロセスを開始してサンプル アプリケーション内の構成を変更する \(P. 201\)」](#) を参照してください。

Clustered DotShoppingCart アプリケーションを展開する

Clustered DotShoppingCart は 3 階層の Windows ベース サンプル アプリケーションで、Microsoft SQL Server をデータベース サーバとして、Microsoft IIS Server をアプリケーション サーバとして、Apache HTTP Server をロード バランサとして使用します。

Create_DB スクリプトは、SQL Server データベースで DotShoppingCart データベースを作成して初期化します。DotShoppingCart_App スクリプトは DotShoppingCart アプリケーションを Microsoft IIS サーバにインストールし、DotShoppingCart データベースを使用します。Microsoft IIS AppServer ノードは、より大きな負荷に対処するために複数のノードにスケールアウトできるクラスタとして定義されています。Apache HTTP サーバはロード バランシングを担当します。

事前定義済みサンプル アプリケーションはテスト環境でのみ使用してください。

開始する前に

- Windows Server 2008 R2 SP1 vApp テンプレートをクラウド プロバイダに登録します。[「vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)」](#) を参照してください。
- クラウド テンプレートを Windows Server 2008 R2 SP1 論理テンプレートにマッピングします。[「ライブラリへの論理テンプレートの追加 \(P. 125\)」](#) を参照してください。
- アプリケーション アーキテクトおよびアプリケーションの公開元および配備者として Application Services にログインします。
- DotShoppingCart 3.0 (オープンソース エディション) パッケージ **OpenSourceEdition.V3.zip** ファイルがダウンロードされていることを確認します。Web サーバは、HTTP プロトコルを使用してファイルを転送するように構成する必要があります。Web ブラウザでファイルにアクセスするために使用される URL を識別します。
- 展開プロファイルの作成とアプリケーションの展開に関する基本概念を理解しておきます。[第 16 章「アプリケーションの展開 \(P. 157\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 [Clustered DotShoppingCart] を選択します。
- 3 画面上部のブループリント スクリーンショットをクリックします。
- 4 ブループリント エディタで DotShoppingCart_App アプリケーション コンポーネントを選択して、DSC_ZIP プロパティ値に有効な URL を入力します。

展開中は、AppServer Windows 仮想マシンの Application Services エージェントが **OpenSourceEdition.V3.zip** ファイルの有効な URL にアクセス可能である必要があります。

- 5 (オプション) 複数の AppServer ノードにする予定がある場合、[AppServer] を選択してクラスタ サイズを増加します。
- 6 展開プロファイルを作成し、アプリケーションを展開します。
- 7 展開されたアプリケーションにアクセスするには、サポートされる Web ブラウザを開いて **http://<Load_Balancer_IP>:8081** という URL を入力します。

ここで、<Load_Balancer_IP> は、展開済みロード バランサーの IP アドレスです。

次に進む前に

更新プロセスを開始して、Clustered DotShoppingCart アプリケーションの構成をスケール調整または変更します。「[サンプル アプリケーションをスケーリングする更新プロセスの開始 \(P. 200\)](#)」 および 「[更新プロセスを開始してサンプル アプリケーション内の構成を変更する \(P. 201\)](#)」 を参照してください。

Clustered Dukes Bank アプリケーションを展開する

Clustered Dukes Bank アプリケーションは、3 階層の Application Services サンプル アプリケーションで、MySQL または Microsoft SQL Server をデータベースとして、JBoss Server をアプリケーション サーバとして、Apache HTTP サーバをロード バランサとして使用しており、テスト環境に展開できます。

initialize_db_script スクリプトは、MySQL または SQL Server で dukes_db database を作成および初期化します。Dukes_Bank_App EAR アプリケーション コンポーネントは、JBoss サーバに展開されている dukes_db データベースを使用します。JBoss アプリケーション サーバ ノードはクラスタとして定義されるため、複数のノードにスケール アウトして大きな負荷にも対処できます。Apache HTTP サーバはロード バランシングを担当します。

事前定義済みサンプル アプリケーションはテスト環境でのみ使用してください。

開始する前に

- アップロード済みの CentOS 6.3 32 ビット Linux テンプレートまたは Windows Server 2008 R2 SP1 vApp テンプレートをクラウド プロバイダに登録します。「[サンプル テンプレートの使用 \(P. 193\)](#)」 および 「[vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)](#)」 を参照してください。
- クラウド テンプレートを CentOS63 (32 ビット) または Windows Server 2008 R2 SP1 論理テンプレートにマッピングします。「[ライブラリへの論理テンプレートの追加 \(P. 125\)](#)」 を参照してください。
- アプリケーション アーキテクトおよびアプリケーションの公開元および配備者として Application Services にログインします。
- Linux ベースの Clustered Dukes Bank でシステムがインターネットへのアクセスにプロキシを必要とする場合は、プロキシ接続が正しく設定されているかどうかを確認します。「[外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)](#)」 を参照してください。
- Windows ベースの Clustered Dukes Bank の場合は、テンプレートで使用する有効な JRE インストール パスが Windows 仮想マシンに設定されていることを確認します。
- 展開プロファイルの作成、アプリケーションの展開、展開済みアプリケーションの更新に関する基本概念を理解しておく必要があります。第 16 章「[アプリケーションの展開 \(P. 157\)](#)」 を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 [Clustered Dukes Bank App] をクリックします。
- 3 アプリケーションのバージョンを選択します。

オプション	アクション
Linux ベースの Clustered Dukes Bank	バージョン 2.1.0 を選択します。
Windows ベースの Clustered Dukes Bank	バージョン 3.0.0 を選択します。

- 4 画面上部のブループリント スクリーンショットをクリックします。
- 5 ブループリント エディタで JBossAppServer サービスを選択して適切なプロパティ値を入力します。
 - Windows ベースの Clustered Dukes Bank の場合は、JAVA_INSTALL_DIR プロパティを更新します。
 - Linux ベースの Clustered Dukes Bank の場合は、JAVA_HOME プロパティを更新します。

展開中は、JBossAppServer サービスの Application Services エージェントがインストール済み JRE にアクセス可能である必要があります。

- 6 Linux ベースの Clustered Dukes Bank でクラウド テンプレートがプロキシを使用しないプライベート ネットワーク内に存在する場合は、アプリケーションを展開する際に展開プロファイルの各仮想マシンに YUM 構成タスクを追加し、プライベート ネットワーク内でホストされている CentOS 6.3 (32 ビット) リポジトリを使用するように `repository_url` プロパティを設定します。
- 7 展開プロファイルを作成し、アプリケーションを展開します。
- 8 展開したアプリケーションにアクセスするには、サポートされている Web ブラウザを開き、アドレス バーに **`http://<Load_Balancer_IP>:8081/bank/main.faces`** と入力します。
ここで、`<Load_Balancer_IP>` は、展開済みロード バランサーの IP アドレスです。
- 9 ログイン名 **200**、パスワード **foobar** を使用して、アプリケーションを起動します。

次に進む前に

更新プロセスを起動して、Clustered Dukes Bank バージョン 3.0.0 または 2.1.0 の構成をスケール調整または変更します。「[サンプル アプリケーションをスケーリングする更新プロセスの開始 \(P. 200\)](#)」 および「[更新プロセスを開始してサンプル アプリケーション内の構成を変更する \(P. 201\)](#)」を参照してください。

jPetStore アプリケーションを展開する

jPetStore 1.0.0 は単一階層のサンプル Web ストア アプリケーションで、テスト環境に展開可能なデータベースとして SQLFire を使用して、vFabric tc Server 上に jPetStore App WAR ファイルを展開します。

jPetStore アプリケーションのブループリントを展開するには、VMware vFabric SQLFire 1.0.0 インストーラ (`vFabric_SQLFire_10_Installer.jar`) ファイルをダウンロードして SQLFire サービス プロパティ `SQLFire10_Installer` を使用してそのファイルを指し示す必要があります。

事前定義済みサンプル アプリケーションはテスト環境でのみ使用してください。

開始する前に

- アップロード済みの jPetStore 1.0.0 用 CentOS 6.3 (32 ビット) テンプレートをクラウド プロバイダに登録します。「[サンプル テンプレートの使用 \(P. 193\)](#)」 および「[vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)](#)」を参照してください。
- このクラウド テンプレートを CentOS63 (32 ビット) ディスク論理テンプレート jPetStore 1.0.0 にマッピングします。「[ライブラリへの論理テンプレートの追加 \(P. 125\)](#)」を参照してください。
- アプリケーションの公開元および配備者として Application Services にログインします。
- システムがインターネットへのアクセスにプロキシを必要とする場合は、プロキシ接続が正しく設定されているかどうかを確認します。「[外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)](#)」を参照してください。
- 展開プロファイルの作成とアプリケーションの展開に関する基本概念を理解しておきます。第 16 章「[アプリケーションの展開 \(P. 157\)](#)」を参照してください。

手順

- 1 VMware ダウンロード ページ (<https://my.vmware.com/web/vmware/downloads>) から vFabric SQLFire 1.0.0 インストーラ パッケージ (`vFabric_SQLFire_10_Installer.jar`) をユーザーの `httpd` サーバまたは Application Services サーバ上の `httpd` サーバにダウンロードします。

Application Services サーバ上の `httpd` サーバを使用している場合は、`vFabric_SQLFire_10_Installer.jar` を `/opt/vmware/darwin/htdocs/artifacts/services/sqlfire` フォルダにダウンロードします。

インストーラ パッケージをダウンロードするには、顧客アカウントの認証情報を入力する必要があります。
- 2 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 3 [jPetStore] アプリケーションをクリックして jPetStore バージョン 1.0.0 を開きます。

- 4 [ブループリントの表示] イメージをクリックして [jPetStore v1.0.0] - [ブループリント] ページを開きます。
 - a jPetStore ブループリント イメージの [SQLFire] をクリックします。
 - b [詳細]、[プロパティ]、[アクション] タブを含む行で [プロパティ] タブをクリックします。
 - c **SQLFire10_Installer** プロパティ名を含む行の[編集] (✎) アイコンをクリックします。
 - d 以下のオプションのいずれか 1 つを使用して、[SQLFire のプロパティの編集] ページでブループリントの値を指定します。
 - [ライブラリの値を使用] をクリックして `http://${darwin.content.server.ip}/artifacts/services/sqlfire/vFabric_SQLFire-1.0.jar` のデフォルトのライブラリの値を受け入れます。ライブラリの値は Application Services サーバにある httpd サーバを指します。

注意 このライブラリの値を使用するには、VMware ダウンロード ページからダウンロードした **vFabric_SQLFire_10_Installer.jar** が `/opt/vmware/darwin/htdocs/artifacts/services/sqlfire` に配置され、**vFabric_SQLFire-1.0.jar** という名前である必要があります。

 - [固有の値を入力する] テキスト ボックスをクリックし、別の httpd の場所にある **vFabric_SQLFire_10_Installer.jar** ファイルへのファイルパスを入力します。
 - e [保存] をクリックして [SQLFire のプロパティの編集] ページを閉じます。
- 5 [展開] をクリックしてブループリントを展開します。
- 6 展開プロファイルを作成します。
- 7 アプリケーションを展開します。
- 8 展開されたアプリケーションにアクセスするには、サポートされている Web ブラウザを開き、アドレス バーに `http://jPetStore_VM_IP:8080/jpetstore-1.0.0` と入力します。

Radiant CMS アプリケーションを展開する

Radiant CMS は単一ノードのアプリケーションです。Ruby On Rails サービスおよび MySQL データベースを展開し、MySQL データベースを使用するように構成された Radiant アプリケーションをインストールします。

事前定義済みサンプル タスクはテスト環境でのみ使用してください。

開始する前に

- アップロードした CentOS 6.3 (32 ビット) テンプレートをクラウド プロバイダに登録します。[\[vCloud Director クラウド プロバイダとテンプレートの登録 \(P. 80\)\]](#)、[\[vRealize Automation クラウド プロバイダとテンプレートの登録 \(P. 58\)\]](#)、または [\[Amazon EC2 のクラウド プロバイダとテンプレートの登録 \(P. 90\)\]](#) を参照してください。
- クラウド テンプレートを CentOS 6.3 (32 ビット) 論理テンプレートにマッピングします。[\[ライブラリへの論理テンプレートの追加 \(P. 125\)\]](#) を参照してください。
- アプリケーションの公開元および配備者として Application Services にログインします。
- Application Services がプロキシを使用するように構成されていることを確認します。[\[外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)\]](#) を参照してください。
- 展開プロファイルの作成とアプリケーションの展開に関する基本概念を理解しておきます。[第 16 章「アプリケーションの展開 \(P. 157\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。

- 2 [Radiant CMS] をクリックします。

このアプリケーションに関するすべてのサービスとアプリケーション コンポーネントは事前定義済みのため、追加の構成を行う必要はありません。

- 3 画面上部のブループリント スクリーンショットをクリックします。
- 4 展開プロファイルを作成し、アプリケーションを展開します。
- 5 アプリケーションにアクセスするには、Radiant CMS 仮想マシンの IP アドレス `http://<VM_IP>/` をサポート対象の Web ブラウザに追加し、Radiant のデフォルトの管理者ユーザー名と Radiant パスワードを使用してログインします。

展開されたサンプル アプリケーションを更新する

既存のサンプル アプリケーション展開の更新を実行すると、その更新に必要な変更を行うために新しい値をキャプチャするプロセスが開始されます。

ノード クラスタを拡大/縮小するか、既存のサービスおよびアプリケーション コンポーネントの構成とコードを変更することによって、それぞれ、単一階層または 3 階層のサンプル アプリケーションの更新プロセスが開始されます。

サンプル アプリケーションをスケーリングする更新プロセスの開始

ノードのクラスタをスケーリングすることにより、事前定義済みサンプル アプリケーションの展開の更新プロセスを複数回開始できます。展開されたアプリケーションのクラスタ化されたノードをスケール アウトまたはスケール インする場合は、アプリケーション ブループリントでクラスタとしてモデル化されているノードのクラスタ サイズのみを構成できます。

次の事前定義済みサンプル アプリケーションをスケーリングできます。

- Nanotrader アプリケーション
- Clustered DotShoppingCart application
- Clustered Dukes Bank 3.0.0 または 2.1.0 アプリケーション バージョン

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。第 12 章「[Application Services コンポーネントの開発 \(P. 93\)](#)」を参照してください。
- 展開済みのアプリケーションには、少なくとも 1 つのクラスタ化ノードが含まれている必要があります。「[クラスタとしてノードを指定する \(P. 143\)](#)」を参照してください。
- クラウド環境への最初の展開が成功したことを確認します。
展開に失敗した場合や拡大/縮小操作に失敗した後に、クラスタ化ノードを拡大/縮小することはできません。
- 展開環境のストレージ領域の制限に関する詳細については、クラウド管理者にお問い合わせください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 正常に展開された事前定義済みサンプル アプリケーションを選択します。
- 3 ツールバーの [操作] ドロップダウン メニューから、[アップデート] を選択します。
[プロファイルの更新] ページが開きます。
- 4 [アップデートプロファイルの作成] を選択します。
[更新プロファイル] ダイアログ ボックスが開きます。
- 5 [更新タイプ] ドロップダウン メニューで、[スケール アウト] または [スケール イン] を選択します。

- 6 スケール調整された更新プロセスに名前を付け、説明（省略可能）を入力して、[OK] をクリックします。
説明には、この更新に含まれる変更内容に関する情報を追加します。
- 7 [ノード] タブで 1 つ以上のクラスタ化ノードのクラスタ サイズを増やし、[次へ] をクリックします。
- 8 スケール調整されたクラスタ化ノードを実行プランで確認して、[次へ] をクリックします。
実行プランの青色の点線は、展開タスクを実行する特定の順序を示しています。
- 9 更新によって変更されるプロパティを確認します。
影響を受けるプロパティがハイライト表示されます。
- 10 [アップデート] をクリックして、更新されたアプリケーションを展開します。

次に進む前に

展開のステータスは、[展開サマリ] ページで確認できます。[「\[展開サマリ\] ページの使用 \(P. 174\)」](#) を参照してください。

更新プロセスを開始してサンプル アプリケーション内の構成を変更する

更新プロセスを開始して、展開済みのサンプル アプリケーションにある既存のサービスおよびアプリケーション コンポーネントの構成とコードを変更できます。展開済みアプリケーションの構成を変更する場合、アプリケーション プロパティ値のみを構成できます。

次の事前定義済みサンプル アプリケーションの構成とコードを変更できます。

- Nanotrader アプリケーション
- Clustered DotShoppingCart application
- Clustered Dukes Bank 3.0.0 または 2.1.0 アプリケーション バージョン

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- コンポーネントのプロパティとアクションを定義して構成するための基本概念を理解します。[第 12 章「Application Services コンポーネントの開発 \(P. 93\)」](#) を参照してください。
- クラウド環境への最初の展開が成功したことを確認します。
更新プロセスを起動して既存のサービスの構成とコードを変更し、クラスタ化ノードをスケール調整するには、展開が成功している必要があります。
- 展開対象のアプリケーションには、展開時にオーバーライド可能なサービス プロパティまたはアプリケーション コンポーネント プロパティが少なくとも 1 つ定義されている必要があります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 正常に展開された事前定義済みサンプル アプリケーションを選択します。
- 3 ツールバーの [操作] ドロップダウン メニューから、[アップデート] を選択します。
[プロファイルの更新] ページが開きます。
- 4 [アップデートプロファイルの作成] を選択します。
[更新プロファイル] ダイアログ ボックスが開きます。
- 5 [更新タイプ] ドロップダウン メニューで、[構成] を選択します。
- 6 構成更新プロセスに名前を付け、説明を入力して、[OK] をクリックします。
説明には、この更新に含まれる変更内容に関する情報を追加します。

- 7 [サービス] タブから、サンプル アプリケーションの該当するプロパティを変更します。

サンプル アプリケーション	サービス バージョン	プロパティ
Nanotrader	vFabric_Web_Server 5.1.1	http_port、deployment_archive、および webserver_conf_file
	vFabric_tc_Server 2.7.1	port、db_ip、db_port、および jdbc_url
	vFabric_SQLFire_Server 1.0.3	schema_file および dataload_file
Clustered DotShoppingCart	Apache_LB 2.2.22	http_port
	SQL_Server_2008 1.0.0	SA_PWD
Clustered Dukes Bank 3.0.0	Apache_LB 2.2.22	http_port および http_proxy_port
	JBossAppServer 5.1.0	JBOSS_JMX_USER および JBOSS_JMX_PWD
	SQL_Server_2008 1.0.0	SA_PWD
	initialize_db_script	db_username および db_password
	Dukes_Bank_App	JAR_FILE、EAR_FILE、db_password、および db_user
Clustered Dukes Bank 2.1.0	Apache_LB 2.2.22	http_port および http_proxy_port
	JBossAppServer 5.1.0	JBOSS_JMX_USER および JBOSS_JMX_PWD
	MySQL 5.0.0	db_port および db_root_password
	initialize_db_script	db_username、db_password、init_db_username、および init_db_password
	Dukes_Bank_App	JAR_FILE、EAR_FILE、db_port、db_password、および db_user

- 8 更新によって変更されるコンポーネントおよび影響を受けるコンポーネントを確認します。

変更されるプロパティおよびそれに依存するプロパティがハイライト表示されます。

- 9 [アップデート] をクリックして、アプリケーションの変更された構成を展開します。

次に進む前に

展開のステータスは、[展開サマリ] ページで確認できます。[\[展開サマリ\] ページの使用 \(P. 174\)](#) を参照してください。

サンプルの Application Services ライブラリ タスク

展開時には実行プランに事前定義済みタスクを追加できます。これらのタスクには、アプリケーションの展開プロセスに必須のタスクと省略可能なタスクがあります。

たとえば、Red Hat Linux ベースのアプリケーションが正しく動作するためには、実行プランに RHN Repository 事前定義済みタスクを追加および構成して、YUM リポジトリをインストールまたは更新する必要があります。

カスタムのタスクを作成し、事前定義済みアプリケーションに追加して、クラウドに展開することもできます。[タスク] ページでカスタム タスクを使用できるようにするには、そのタスクをライブラリに追加する必要があります。[\[カスタム タスクをライブラリに追加する \(P. 132\)\]](#) を参照してください。

■ [ドメインへの参加用事前定義済みタスクの追加 \(P. 203\)](#)

ドメインへの参加用事前定義済みタスクにより、Windows 仮想マシンは、展開プロセスの一環として Active Directory ドメインに参加できます。事前定義済みタスクでは Windows ドメイン マネージャの **Netdom.exe** ユーティリティを開始して、この操作を自動化します。

■ [APT Repository Config 事前定義済みタスクを追加する \(P. 204\)](#)

APT Repository Config 事前定義済みタスクは、Ubuntu またはその他の DEB ベースのオペレーティングシステムで APT リポジトリがソフトウェアをインストールまたは更新できるように更新するスクリプトです。

- [YUM Repository Config 事前定義済みタスクを追加する \(P. 205\)](#)
YUM Repository Config 事前定義済みタスクは、CentOS またはその他の RPM ベースのオペレーティングシステムで YUM リポジトリがソフトウェアをインストールまたは更新できるように更新するスクリプトです。
- [RHN 登録用事前定義済みタスクの追加 \(P. 207\)](#)
RHN 登録用事前定義済みタスクは、Red Hat Enterprise Linux または Red Hat Network に接続可能なその他の Red Hat オペレーティングシステムを、YUM を使用してインストールできるように YUM リポジトリを更新するスクリプトです。

ドメインへの参加用事前定義済みタスクの追加



ドメインへの参加用事前定義済みタスクにより、Windows 仮想マシンは、展開プロセスの一環として Active Directory ドメインに参加できます。事前定義済みタスクでは Windows ドメイン マネージャの **Netdom.exe** ユーティリティを開始して、この操作を自動化します。


事前定義済みタスクを使用すれば、静的なドメイン設定を使用して Windows 仮想マシン テンプレートを手動で構成する必要はありません。タスクはカスタマイズすることができます。事前定義済みタスクでは、Active Directory ドメインへの参加プロセスを完了するために、追加の再起動サイクルが必要になります。

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- Application Services がプロキシを使用するように構成されていることを確認します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。
- 複数の展開の場合は、アプリケーションを展開する前に、Windows 仮想マシン テンプレートに対して[SID の変更] オプションと [ドメインへの参加] オプションを手動で有効にします。[「Windows 仮想マシン テンプレートに対する SID の変更とドメインへの参加の有効化 \(P. 76\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 Windows ベースのアプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択します。
- 4 展開プロファイルを作成します。
- 5 [展開プロファイル] ウィザードで、[実行プラン] のプロンプトに従って操作します。
- 6 ノードがクラスタ化されている場合は、[クラスタの展開]  をクリックします。
クラスタ化されたノードが展開されない場合、事前定義済みタスクはクラスタ内の 1 番目の仮想マシンにのみ追加されます。
- 7 [スクリプト タスクの追加]  をクリックして、事前定義済みタスクをブループリントにドラッグします。

事前定義済みタスクをドラッグすると、事前定義済みタスクをドロップできる場所を示すアンカー  が表示されます。

事前定義済みタスクをノードにドラッグ アンド ドロップすると、[カスタム タスクの追加] ダイアログ ボックスが表示されます。

- 8 [ライブラリ タスク名] ドロップダウン メニューから事前定義済みタスクを選択します。
サポートされるオペレーティング システム、事前定義済みタスクの詳細、スクリプト、プロパティの詳細が表示されます。

- 9 [プロパティ] タブで、プロパティを構成します。
 - a domain_name プロパティを選択し、[プロパティの編集] ダイアログ ボックスで Windows ドメインの新しい名前を入力して、[保存] をクリックします。
 - b domain_user プロパティを選択し、[プロパティの編集] ダイアログ ボックスで Active Directory に参加できるドメイン ユーザーの名前を指定して、[保存] をクリックします。
 - c domain_password プロパティを選択し、[プロパティの編集] ダイアログ ボックスにドメイン ユーザーのパスワードを入力して、[保存] をクリックします。
 - d (オプション) apply_ou プロパティを選択し、Active Directory ドメイン内の特定の組織単位が Windows 仮想マシンに参加できるように [プロパティの編集] ダイアログ ボックスで新しい値を **はい** に変更して、[保存] をクリックします。
 - e (オプション) domain_ou プロパティを選択し、[プロパティの編集] ダイアログ ボックスで Active Directory ドメインの組織単位を指定して、[保存] をクリックします。
- 10 [OK] をクリックします。
ドメインへの参加用事前定義済みタスクが実行プランに追加されます。
- 11 展開プロファイルの設定を確認して、アプリケーションを展開します。

次に進む前に

カスタマイズ済みタスクを Application Services ライブラリに追加するかどうかを検討します。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。

APT Repository Config 事前定義済みタスクを追加する

APT Repository Config 事前定義済みタスクは、Ubuntu またはその他の DEB ベースのオペレーティング システムで APT リポジトリがソフトウェアをインストールまたは更新できるように更新するスクリプトです。

新規のリポジトリを追加したり、既存のすべてのリポジトリを削除したりするように、APT Repository Config の各プロパティを構成できます。2 つ以上のリポジトリが必要な場合は、複数のタスクを作成し、実行プランで 1 つのタスクを別のタスクの隣に追加することでタスクをリンクすることができます。



事前定義済みサンプル タスクはテスト環境でのみ使用してください。

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- ソフトウェア パッケージのインストールまたは更新に APT を必要とするサービス コンポーネントまたはアプリケーション コンポーネントを追加する前に、この事前定義済みタスクが展開実行プランに追加されていることを確認します。
- Application Services がプロキシを使用するように構成されていることを確認します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択します。
- 4 展開プロファイルを作成します。
- 5 [展開プロファイル] ウィザードで、[実行プラン] のプロンプトに従って操作します。

- 6 ノードがクラスタ化されている場合は、[クラスタの展開]  をクリックします。
クラスタ化されたノードが展開されない場合、事前定義済みタスクはクラスタ内の 1 番目の仮想マシンにのみ追加されます。
- 7 [スクリプト タスクの追加]  をクリックして、事前定義済みタスクをブループリントにドラッグします。

事前定義済みタスクをドラッグすると、事前定義済みタスクをドロップできる場所を示すアンカー () が表示されます。

事前定義済みタスクをノードにドラッグ アンド ドロップすると、[カスタム タスクの追加] ダイアログ ボックスが表示されます。
- 8 [ライブラリ タスク名] ドロップダウン メニューから事前定義済みタスクを選択します。
サポートされるオペレーティング システム、事前定義済みタスクの詳細、スクリプト、プロパティの詳細が表示されます。
- 9 [プロパティ] タブで、プロパティを構成します。
 - a repository_name プロパティを選択し、[プロパティの編集] ダイアログ ボックスでリポジトリを特定する新しい一意の値を入力して、[保存] をクリックします。
 - b source_str プロパティを選択し、[プロパティの編集] ダイアログ ボックスの [URL 値] テキスト ボックスに **http://site.example.com/debian distribution component1 component2 ...** と入力して、[保存] をクリックします。
たとえば Ubuntu URL は、http://us.archive.ubuntu.com/ubuntu/ lucid main のようになります。
 - c remove_all_repos プロパティを選択し、[プロパティの編集] ダイアログ ボックスで適切な値を定義します。
この値を **true** に設定すると、新規の構成を追加する前にほかのすべてのリポジトリが削除されます。デフォルトの **false** をそのまま受け入れて、新規のリポジトリを追加することもできます。
- 10 [保存] をクリックします。
- 11 [OK] をクリックします。
APT Repository Config 事前定義済みタスクが実行プランに追加されます。
- 12 展開プロファイルの設定を確認して、アプリケーションを展開します。

次に進む前に

カスタマイズ済みタスクを Application Services ライブラリに追加するかどうかを決定します。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。

YUM Repository Config 事前定義済みタスクを追加する

YUM Repository Config 事前定義済みタスクは、CentOS またはその他の RPM ベースのオペレーティング システムで YUM リポジトリがソフトウェアをインストールまたは更新できるように更新するスクリプトです。

新規のリポジトリを追加したり、既存のすべてのリポジトリを削除したりするように、YUM Repository Config の各プロパティを構成できます。2 つ以上のリポジトリが必要な場合は、複数のタスクを作成し、実行プランで 1 つのタスクを別のタスクの隣に追加することでタスクをリンクすることができます。



事前定義済みサンプル タスクはテスト環境でのみ使用してください。


開始する前に

- Application Services ロールが割り当てられているユーザー アカウントで Application Services にログインします。
- アプリケーションの公開元および配備者として Application Services にログインします。

- ソフトウェア パッケージのインストールまたは更新に YUM を必要とするサービス コンポーネントまたはアプリケーション コンポーネントを追加する前に、事前定義タスクが展開実行プランに追加されていることを確認します。
- Application Services がプロキシを使用するように構成されていることを確認します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択します。
- 4 展開プロファイルを作成します。
- 5 [展開プロファイル] ウィザードで、[実行プラン] のプロンプトに従って操作します。
- 6 ノードがクラスタ化されている場合は、[クラスタの展開] () をクリックします。
クラスタ化されたノードが展開されない場合、事前定義済みタスクはクラスタ内の 1 番目の仮想マシンにのみ追加されます。
- 7 [スクリプト タスクの追加] () をクリックして、事前定義済みタスクをブループリントにドラッグします。

事前定義済みタスクをドラッグすると、事前定義済みタスクをドロップできる場所を示すアンカー () が表示されます。

事前定義済みタスクをノードにドラッグ アンド ドロップすると、[カスタム タスクの追加] ダイアログ ボックスが表示されます。

- 8 [ライブラリ タスク名] ドロップダウン メニューから事前定義済みタスクを選択します。
サポートされるオペレーティング システム、事前定義済みタスクの詳細、スクリプト、プロパティの詳細が表示されます。
- 9 [プロパティ] タブで、プロパティを構成します。
 - a repository_name プロパティを選択し、[プロパティの編集] ダイアログ ボックスでリポジトリを特定する新しい一意の値を入力して、[保存] をクリックします。
 - b repository_url プロパティを選択し、[プロパティの編集] ダイアログ ボックスに URL 値を入力して、[保存] をクリックします。
たとえば、CentOS 6.3 (32 ビット) の URL は <http://vault.centos.org/6.3/os/i386/> のようになります。
 - c remove_all_repos プロパティを選択し、[プロパティの編集] ダイアログ ボックスで適切な値を定義します。
 - d この値を **true** に設定すると、新規の構成を追加する前にほかのすべてのリポジトリが削除されます。
 - e デフォルトの **false** をそのまま受け入れると、新規のリポジトリが追加されます。
- 10 [保存] をクリックします。
- 11 [OK] をクリックします。
YUM Repository Config 事前定義済みタスクが実行プランに追加されます。
- 12 展開プロファイルの設定を確認して、アプリケーションを展開します。

次に進む前に

Application Services ライブラリにカスタム タスクを追加することを検討します。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。

RHN 登録用事前定義済みタスクの追加

RHN 登録用事前定義済みタスクは、Red Hat Enterprise Linux または Red Hat Network に接続可能なその他の Red Hat オペレーティングシステムを、YUM を使用してインストールできるように YUM リポジトリを更新するスクリプトです。


事前定義タスクでは、VMware_AppDirector_<\$RANDOM> (<\$RANDOM> は仮想マシンの登録を一意にするための短い文字列) というマシン名によって与えられる資格情報を使用して、仮想マシンを Red Hat Network に登録します。

事前定義済みサンプル タスクはテスト環境でのみ使用してください。


開始する前に


- **アプリケーションの公開元および配備者**として Application Services にログインします。
- ソフトウェア パッケージのインストールまたは更新に YUM を必要とするサービス コンポーネントまたはアプリケーション コンポーネントを追加する前に、事前定義タスクが展開実行プランに追加されていることを確認します。
- Application Services がプロキシを使用するように構成されていることを確認します。[「外部 URL 用にプロキシを使用するように Application Services を構成する \(P. 34\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [アプリケーション] を選択します。
- 2 アプリケーションの名前をクリックします。
アプリケーション バージョンのリストが表示されます。
- 3 アプリケーション バージョンを選択します。
- 4 展開プロファイルを作成します。
- 5 [展開プロファイル] ウィザードで、[実行プラン] のプロンプトに従って操作します。
- 6 ノードがクラスタ化されている場合は、[クラスタの展開]  をクリックします。

クラスタ化されたノードが展開されない場合、事前定義済みタスクはクラスタ内の 1 番目の仮想マシンにのみ追加されます。

- 7 [スクリプト タスクの追加]  をクリックして、事前定義済みタスクをブループリントにドラッグします。

事前定義済みタスクをドラッグすると、事前定義済みタスクをドロップできる場所を示すアンカー  が表示されます。

事前定義済みタスクをノードにドラッグ アンド ドロップすると、[カスタム タスクの追加] ダイアログ ボックスが表示されます。

- 8 [ライブラリ タスク名] ドロップダウン メニューから事前定義済みタスクを選択します。
サポートされるオペレーティングシステム、事前定義済みタスクの詳細、スクリプト、プロパティの詳細が表示されます。
- 9 [プロパティ] タブで、プロパティを構成します。
 - a rhn_username プロパティを選択し、仮想マシンを Red Hat Network に登録するときに使用するユーザー名を入力して、[保存] をクリックします。
 - b rhn_password プロパティを選択し、仮想マシンを Red Hat Network に登録するときに使用するパスワードを入力して、[保存] をクリックします。
- 10 [OK] をクリックします。

RHN 登録用事前定義タスクが実行プランに追加されます。

11 展開プロファイルの設定を確認して、アプリケーションを展開します。

次に進む前に

カスタマイズしたタスクがある場合は、Application Services ライブラリに追加できます。[「カスタム タスクをライブラリに追加する \(P. 132\)」](#) を参照してください。

サンプルのライブラリ サービス

Application Services のライブラリには、サービスなどの事前定義済みコンポーネントが含まれており、複数のアプリケーションで再使用できます。これらのサービスは、Application Services のすべてのビジネス グループで利用できます。

ライブラリ サービス

Application Services タイトル バーで、ドロップダウン メニューをクリックして、[ライブラリ]-[サービス] を選択し、利用可能なサンプル サービスを表示できます。[ライブラリ] メニューには、標準の論理テンプレート、タスク、オペレーティング システム、タグ、ポリシー、および外部サービスも表示されます。

アプリケーション アーキテクトは、アプリケーション ブループリントを作成し、サンプル サービスを該当するノードに追加して構成できます。サンプル サービスは、事前定義済みアプリケーションを展開するときにも構成できます。

アプリケーション ブループリントでは、これらのサンプルサービスがアプリケーション サーバ、データベース サーバ、Web サーバ、Windows サービス、監視、Puppet サービスなどにグループ化されています。

注意 事前定義済みのサンプル ライブラリ サービスは、テスト環境でのみ使用してください。

展開の管理

アプリケーションを展開すると、Application Services の [展開] ページにアイテムが追加されます。

[展開] ページには次のアイテムが表示されます。

- すべての展開の一覧
- 個々の展開の詳細（クラウド プロバイダ、展開環境、展開プロファイル、アプリケーションのタイプとバージョン、更新プロセスなど）
- 展開のポリシーと使用可能な更新

また、ポリシー スキャンや更新プロセスの開始、クラウドから展開したアプリケーションの分解、Application Services からのアプリケーション展開レコードの削除も実行できます。

注意 複合展開では、ポリシー スキャンは機能しません。

[展開サマリ] ページから、更新とコンプライアンスのサマリ ページに移動できます。

この章では次のトピックについて説明します。

- [アプリケーションの展開 タスクとブループリントの詳細の表示 \(P. 210\)](#)
- [展開された仮想マシンの詳細情報とアプリケーションの実行プランの表示 \(P. 211\)](#)
- [ポリシー スキャンの開始 \(P. 213\)](#)
- [クラウドからのアプリケーションの分解 \(P. 214\)](#)
- [vCloud Automation Center からの展開のスケール イン \(P. 215\)](#)
- [vCloud Automation Center からの展開のスケール アウト \(P. 215\)](#)
- [vCloud Automation Center からのアプリケーションの分解 \(P. 216\)](#)
- [Application Services からアプリケーション展開を削除する \(P. 217\)](#)
- [展開または更新プロセスのキャンセル \(P. 218\)](#)
- [ポリシー コンプライアンスの概要の表示 \(P. 219\)](#)

アプリケーションの展開 タスクとブループリントの詳細の表示

特定のアプリケーションの展開または更新プロセスの進行状況、成功、失敗に関する詳細については、[展開サマリ] ページに表示できます。このページには、割り当てられた IP アドレス、選択されたクラウド ネットワーク、実行されたインストール、構成、起動、更新の各スクリプトのログが表示されます。

[展開 サマリ] ページには、展開の全体的なステータスが表示されます。このページに一覧表示される各展開に含まれているのは、展開が開始された時点のアプリケーション ブループリントおよび展開プロファイルのスナップショットです。実際のアプリケーション ブループリントまたは展開プロファイルを変更しても、その変更内容が個々の展開に表示されるブループリントまたは展開に反映されることはありません。[\[「展開サマリ」 ページの使用 \(P. 174\)\]](#) を参照してください。

[複合展開サマリ] ページには、アプリケーションの展開すべての詳細が表示され、[\[「複合展開サマリ」 ページの使用 \(P. 174\)\]](#) で説明されているように [展開サマリ] ページに簡単にアクセスできます。



すべての Application Services ロールは、ビジネス グループの展開を表示できます。

開始する前に

アプリケーションが展開されていること、または更新プロセスが開始されていることを確認します。[\[「単一の展開プロファイルで展開する \(P. 168\)」](#) または [第 17 章「アプリケーション展開の更新 \(P. 179\)」](#) を参照してください。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 展開名をクリックします。

[展開サマリ] ページにステータス ウィンドウが表示されます。[\[仮想マシンの詳細\]](#) ステータス ウィンドウから各仮想マシンに迅速にアクセスし、vCloud Director に展開されている仮想マシンにログインできます。
- 3 タスクのステータス、依存タスク、または IP アドレスを取得している仮想マシンを確認するには、[展開] アイコン  をクリックします。[展開サマリ] ページに、該当するウィンドウが開きます。
- 4 [最小化] アイコン  をクリックすると、展開ステータス サマリ全体が表示されます。
- 5 展開ステータス全体を把握するには、[タスク詳細] ステータス ウィンドウを確認します。
- 6 展開タスクの名前を検索するには、[タスク詳細] ステータス ウィンドウの [タスク サマリ] セクションを確認します。

[タスク サマリ] セクションには、展開または更新プロファイルの詳細、ユーザー ロール、ポリシーのコンプライアンス情報、展開の開始/終了時刻、アプリケーションの最終更新日時、ログを含むサブフォルダの実行 ID 番号、およびアプリケーション名に割り当てられた値が一覧表示されます。

vCloud Director、vCloud Automation Center、または Amazon EC2 で展開に関する情報を検索するには、vApp、マシン、またはインスタンスの名前を入力する必要があります。vCloud Director、vCloud Automation Center、および Amazon EC2 のユーザー インターフェイスから展開の進行状況を確認することもできます。
- 7 展開環境、クラウド プロバイダの名前、クラウド プロバイダの説明、ホスト IP アドレス、vCloud Director 組織名、ビジネス グループおよび予約ポリシー、または Amazon VPC とこれに関連付けられた可用性ゾーン、およびユーザー名の詳細については、[タスク詳細] ステータス ウィンドウの [展開の場所] セクションを確認します。

このセクションの展開情報は、展開された時点のアプリケーション ブループリントと展開プロファイルのスナップショットです。実際のアプリケーション ブループリントを変更しても、その変更内容が個々の展開に表示されるブループリントに反映されることはありません。クラウド プロバイダと展開情報には、クラウド プロバイダ マッピング、およびアプリケーション展開用に作成された展開環境の詳細が反映されます。
- 8 アプリケーション ブループリントの名前とバージョン情報を表示するには、[タスク詳細] ステータス ウィンドウの [アプリケーション詳細] セクションを開きます。

- 9 ブループリント プロファイルおよび展開プロファイルのプロパティ オーバーライドを表示するには、[ブループリント] ステータス ウィンドウを展開します。

このステータス ウィンドウには、展開された時点でブループリントに含まれていた設定と定義を反映したスナップショットが表示されます。実際のアプリケーション ブループリントを変更しても、その変更内容が個々の展開に表示されるブループリントに反映されることはありません。展開は、アプリケーションのコンポーネントやサービスに影響を与えることなく削除できます。

- a サービスまたはアプリケーション コンポーネントを選択して、[プロパティ] タブをクリックします。
- b 更新プロセスの場合は、[プロパティ] タブの [前の値] 列に、前回の更新プロセスのライブラリ、ブループリント、または展開プロファイルの値が表示されます。

同じタグの [新しい値] 列には、現在の更新プロセスで追加された値が表示されます。

- 10 展開されたブループリントのステータス ウィンドウで失敗したタスクを表示します。

サービス コンポーネントまたはアプリケーション コンポーネントでタスクが失敗すると、タスク失敗アイコン (❌) が、そのサービス コンポーネントまたはアプリケーション コンポーネントに表示されます。1 つのタスクが失敗すると展開全体が停止し、失敗した展開というマークが付きます。その展開の以降のタスクは実行されません。以降のすべてのタスクには、タスクが実行されなかったことを示すアイコンが表示されます。

更新展開が発生すると、更新対象ノードのアプリケーションのコンポーネントとプロパティが強調表示されます。更新展開が失敗すると、アプリケーション コンポーネントに失敗のマークが付き、影響を受けるプロパティが赤色で強調表示されます。

次に進む前に

実行プランで仮想マシンの詳細と使用可能なタスクを確認します。[「展開された仮想マシンの詳細情報とアプリケーションの実行プランの表示 \(P. 211\)」](#) を参照してください。

展開された仮想マシンの詳細情報とアプリケーションの実行プランの表示

展開サマリ ページから、仮想マシン固有の情報の [仮想マシンの詳細] ステータス ウィンドウを展開できます。[実行プラン] ステータス ウィンドウを展開して、アプリケーションで定義された依存関係に基づいて実行されるプロビジョニングタスクを表示することもできます。

vCloud Director、vRealize Automation、または Amazon EC2 のノード名、論理テンプレート、およびクラウドテンプレートなどの仮想マシン関連情報、メモリ割り当て、CPU 数、ネットワーク接続の詳細を確認できます。ホスト名を表示して、vCloud Director または vRealize Automation 展開の仮想マシンを容易に識別することもできます。

開始する前に

アプリケーションが展開されていること、または更新プロセスが開始されていることを確認します。[「単一の展開プロファイルで展開する \(P. 168\)」](#) または [第 17 章「アプリケーション展開の更新 \(P. 179\)」](#) を参照してください。

手順

- 1 仮想マシンの IP アドレスを見つけるには、[仮想マシンの詳細] ステータス ウィンドウを確認します。
各仮想マシンの IP アドレスが、割り当てたときの IP アドレス列に表示されます。


たとえば、ロード バランサ ノードを含む 3 階層アプリケーションでは、ロード バランサの IP アドレスを見つけてその IP アドレスをエンド ユーザーに与えることができます。
- 2 [仮想マシンの詳細] ステータス ウィンドウの仮想マシンの表で、仮想マシン固有のエージェント ブートストラップ ログを見つけます。

各仮想マシンのエージェント ブートストラップ ログは [ログ] 列で更新されます。ブートストラップ プロセスが失敗すると、展開は「展開失敗」とラベルが付けられ、その理由が [タスク詳細] ウィンドウに表示されます。

- 3 ウィンドウを展開して [クラウド テンプレート] 列の省略記号ボタン ([...]) をクリックし、vCloud Director、vRealize Automation、または Amazon EC2 のいずれかから仮想マシン固有の詳細なクラウド テンプレート情報を取得します。

この情報には、ディスク サイズ、CPU、メモリなどの情報が含まれます。

- 4 ウィンドウを展開して [ネットワークの情報] 列の省略記号 ([...]) をクリックし、vCloud Director または Amazon EC2 からネットワークの詳細を取得します。

- 5 各ノードに定義されたカスタム プロパティを表示して vRealize Automation ブループリントのプロパティをオーバーライドするには、[追加の構成] アイコン () をクリックします。

- 6 進行中の展開については、ページの右上隅の [更新] をクリックしてステータスを更新します。








このページは約 30 秒後に更新されます。


- 7 開始時刻、終了時刻、最終更新時刻などの特定のタスク詳細については、[実行プラン] ステータス ウィンドウを展開します。

各タスクでは、IP アドレスを取得してエージェントがブートストラップされた後で、コンポーネントが展開、インストール、構成されます。これらのプロセスは、実行プランのタスク間の矢印によって示されるアプリケーションの依存関係に従って行われます。


- 8 各タスクのステータスを確認します。


タスクのステータスによって、アイコンがその横に表示されます。


アイコン	説明
	タスクがまだ開始されておらず、実行されませんでした。
	タスクが進行中です。このステータス アイコンは、タスクが正常に終了したときに表示されます。このアイコンは、ホスト、エージェント ブートストラップ、ネットワーク ブートストラップのプロビジョニング タスクの全体の終了ステータスを示すためにも表示されます。
	各プロビジョニング タスクの詳細を展開して表示します。
	タスクは再起動が必要です。
	タスクが失敗しました。このステータス アイコンは、タスクが失敗した場合に表示されます。このアイコンは、ホスト、エージェント ブートストラップ、ネットワーク ブートストラップのプロビジョニング タスクの全体の失敗ステータスを示すためにも表示されます。
	1 つ以上の失敗したプロビジョニング タスクのログを展開して表示します。
	タスクは依存関係の 1 つが実行を終了するのを待機しています。

- 9 コンポーネントまたはアクション スクリプトおよびそのプロパティの詳細については、[実行プラン] ステータス ウィンドウを展開し、タスクの横の矢印アイコン () をクリックして、[コンポーネント プロパティの表示] を選択します。

ここで一覧表示される詳細は、展開の時点でブループリントに含まれていた設定および定義です。

- 10 アクション スクリプトを表示するには、[実行プラン] ステータス ウィンドウを展開し、タスクの横の矢印アイコン () をクリックして、[アクション スクリプトの表示] を選択します。

- 11 仮想マシンのログ ファイルにアクセスするには、タスクの横の矢印アイコン () をクリックし、[仮想マシン ログの表示] を選択します。

- 12 スクリプトのプロパティ値を表示するには、タスクの横の矢印アイコン () をクリックし、[コンポーネント プロパティの表示] を選択します。

- 13 更新された展開の詳細については、[実行プラン] ステータス ウィンドウを展開し、更新されたノードを調べます。

拡張された展開の場合、実行プランに変更されたクラスタ化ノードと影響を受けるノードの更新スクリプトが表示されます。ホスト、エージェント ブートストラップ、ネットワーク ブートストラップのプロビジョニングタスクは、スケール アウトされたノードでのみ表示されます。依存ノードにはプロビジョニングタスクがありません。

展開されたアプリケーションの構成を変更するための更新プロセスの場合、実行プランに変更され影響を受けるノードの更新スクリプトが表示されます。

次に進む前に

クラウドから展開されたアプリケーションを分解するには、[「クラウドからのアプリケーションの分解 \(P. 214\)」](#) を参照してください。

Application Services から展開レコードを削除するには、[「Application Services からアプリケーション展開を削除する \(P. 217\)」](#) を参照してください。

ポリシー スキャンの開始

展開でポリシー スキャンを開始して、展開に対応する展開環境下に定義されているすべてのポリシー インスタンスを評価できます。展開または更新プロセスを開始する最初の手順として、Application Services はポリシー評価を実行します。

ポリシー スキャンはアクティブな展開でしか実行できません。たとえば、更新プロセスに失敗すると、ポリシー スキャンを再試行する前に現在の展開の状態をアクティブにリストアする必要があります。

ポリシー インスタンスの展開環境で使用されるポリシーを更新する場合、ユーザーは既存のポリシー インスタンスを削除して、更新されたポリシーに基づいてポリシー インスタンスを作成する必要があります。

開始する前に

- **アプリケーションの公開元および配備者**として Application Services にログインします。
- ポリシー またはポリシー インスタンスが変更されていることを確認します。
- 古いポリシー定義を含む展開が Application Services で使用できることを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 スキャンする既存の展開の名前を検索テキスト ボックスに入力します。
- 3 検索結果のリストからスキャンする展開を選択します。
[展開サマリ] ページが開きます。
- 4 展開サマリの上のツールバーで、[操作] - [スキャン] を選択します。
- 5 スキャンの展開プロセスを確認します。

[コンプライアンス サマリ] ページが開き、全体的なコンプライアンス ステータスとポリシーの詳細が表示されます。ポリシー スキャンがタイム ラインに表示され、スキャンが開始された時点にタイム スタンプが付きます。

次に進む前に

展開スキャンのポリシー コンプライアンスの詳細を追跡します。[「ポリシー コンプライアンスの概要の表示 \(P. 219\)」](#) を参照してください。

クラウドからのアプリケーションの分解

Application Services では、分解プロセスを開始して、vCloud Director vApp および関連する仮想マシン、vRealize Automation 仮想マシンおよび関連する vCenter Server 内の仮想マシン、または Amazon EC2 インスタンスをクラウド環境から削除できます。

Application Services ユーザー インターフェイスまたはコマンドライン インターフェイスから、展開済みアプリケーションを分解できます。[\[CLI を使用した展開の分解 \(P. 227\)\]](#) を参照してください。ユーザー インターフェイスでは、サービス バージョンおよびアプリケーション コンポーネントに対して TEARDOWN ライフ サイクル ステージのスクリプトを定義し、アプリケーションおよび関連する仮想マシンまたは更新されたアプリケーションを削除できます。

アプリケーションの一部である 1 つ以上の仮想マシンをインストールした後に展開が失敗するか、アプリケーションが使用されていない場合は、Application Services を使用してアプリケーションを分解できます。アプリケーション内のすべての仮想マシンは、クラウド内のホストから削除されます。

クラウドからの展開の分解では、Application Services の [展開] ページから展開レコードは削除されません。[展開] ページから展開レコードを削除するには、[\[Application Services からアプリケーション展開を削除する \(P. 217\)\]](#) を参照してください。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- アプリケーションの一部である仮想マシンがクラウド内に存在していることを確認します。たとえば、一定日数の経過後に仮想マシンを削除するポリシーが環境に設定されていると、仮想マシンはすでに削除されている可能性があります。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 検索テキスト ボックスに、分解する展開の名前を入力します。
- 3 検索結果リストから、分解する展開を選択します。
[展開サマリ] ページが開きます。
- 4 展開サマリの上のツールバーで、[操作] - [分解] を選択します。
 - [展開の分解] ダイアログ ボックスで、[クイック分解] を選択してこの展開の仮想マシンを削除し、[分解] をクリックします。
 - [展開の分解] ダイアログ ボックスで、[アシストによる分解] を選択して [分解] ウィザードを開き、[分解] をクリックします。

分解の簡易プロセスの場合、TEARDOWN ライフ サイクル ステージのスクリプトは実行されません。

複合アプリケーション展開では、簡易分解プロセスのみがサポートされ、そのアプリケーション展開のすべてを同時に分解する必要があります。

アシスト付きの分解プロセスの場合は、[分解] ウィザードに構成可能な TEARDOWN ライフ サイクル ステージのスクリプトが表示されます。スクリプト内のタスクが実行される順序を表示し、このスクリプト内の変更を確認してから、スクリプトを実行してこの展開の仮想マシンを削除することもできます。

- 5 (オプション) 分解プロセスに失敗したら、このプロセスを繰り返します。

[タスクのステータス] ウィンドウの上部にあるタスクのタイム ラインに、分解プロセスのステータスが表示されます。クラウド内のホストから仮想マシンが正常に削除されると、ステータスが [タスク サマリ] セクションと展開全体のステータスに表示されます。

分解プロセスの開始後は、分解プロセスが失敗し、仮想マシンがクラウド環境に存在する場合でも、展開済み仮想マシンは更新できません。クラウド管理者に、クラウドから展開を分解するように依頼してください。

次に進む前に

[展開] ページから展開のレコードを削除するには、[「Application Services からアプリケーション展開を削除する \(P. 217\)」](#)を参照してください。

vCloud Automation Center からの展開のスケール イン


vRealize Automation コンソールから、既存の展開内の 1 つまたは複数のクラスタ化ノードのスケール インを申請できます。アプリケーション ブループリントでクラスタとしてモデル化されたノードのクラスタ サイズのみを構成できます。

マッピングされた成果物を持つブループリントを使う展開環境では、拡大/縮小することはできません。成果物の管理の詳細については、[第 15 章「成果物の操作 \(P. 149\)」](#)を参照してください。

開始する前に

- テナント管理者として vRealize Automation コンソールにログインします。
- vRealize Automation から、クラスタ化ノードを含む少なくとも 1 つの展開済みアプリケーションにアクセスできることを確認します。

手順

- 1 [アイテム] - [アプリケーションの展開] を選択します。
- 2 検索テキスト ボックスに、展開の名前を入力し、虫眼鏡アイコン  をクリックするか、Enter を押します。
- 3 検索結果のリストから展開を選択します。
[アイテムの詳細] ページが表示されます。
- 4 [アクション] メニューから [スケール イン] を選択します。
[スケール イン] ダイアログ ボックスに [申請情報] タブが表示されます。
- 5 [説明] テキスト ボックスに説明を入力します。
この必須の説明により、送信後に申請の進捗を監視することができます。
- 6 (オプション) [理由] テキスト ボックスに、申請の理由を入力します。
- 7 [次へ] をクリックします。
- 8 [スケール イン基準] テキスト ボックスの上下矢印をクリックして、スケール インするクラスタの数を入力します。
- 9 [送信] をクリックします。
申請の確認のボックスが表示されます。
- 10 [OK] をクリックします。

次に進む前に

vRealize Automation コンソールの [申請] タブでスケール イン申請のステータスを監視できます。

vCloud Automation Center からの展開のスケール アウト

vRealize Automation コンソールから、既存の展開内の 1 つまたは複数のクラスタ化ノードのスケール アウトを申請できます。アプリケーション ブループリントでクラスタとしてモデル化されたノードのクラスタ サイズのみを構成できます。

マップ済みの成果物があるブループリントを使用する展開は、スケール アウトできません。成果物の管理の詳細については、[第 15 章「成果物の操作 \(P. 149\)」](#)を参照してください。

開始する前に

- テナント管理者として vRealize Automation コンソールにログインします。

- vRealize Automation からクラスタ化ノードを含む少なくとも 1 つの展開済みアプリケーションにアクセスできることを確認します。

手順

- 1 [アイテム] - [アプリケーションの展開] を選択します。
- 2 検索テキスト ボックスに、展開の名前を入力し、虫眼鏡アイコン (🔍) をクリックするか、Enter を押します。
- 3 検索結果のリストから展開を選択します。
[アイテムの詳細] ページが表示されます。
- 4 [アクション] メニューから [スケール アウト] を選択します。
[スケール アウト] ダイアログ ボックスに [申請情報] タブが表示されます。
- 5 [説明] テキスト ボックスに説明を入力します。
この必須の説明により、送信後に申請の進捗を監視することができます。
- 6 (オプション) [理由] テキスト ボックスに、申請の理由を入力します。
- 7 [次へ] をクリックします。
- 8 [スケール アウト基準] テキスト ボックスの上下矢印をクリックして、スケール アウトするクラスタの数を入力します。
- 9 [送信] をクリックします。
申請の確認のボックスが表示されます。
- 10 [OK] をクリックします。

次に進む前に

vRealize Automation コンソールの [申請] タブでスケール アウト申請のステータスを監視できます。

vCloud Automation Center からのアプリケーションの分解

vRealize Automation コンソールから、標準および複合アプリケーションと仮想マシンの分解を申請することができます。

Application Services ユーザー インターフェイスまたはコマンドライン インターフェイスから、展開済みアプリケーションを分解することもできます。[「クラウドからのアプリケーションの分解 \(P. 214\)」](#) および [「CLI を使用した展開の分解 \(P. 227\)」](#) を参照してください。Application Services ユーザー インターフェイスでは、サービス バージョンおよびアプリケーション コンポーネントに対して TEARDOWN ライフ サイクル ステージのスクリプトを定義し、アプリケーションおよび関連付けられている仮想マシンまたは更新されたアプリケーションを削除できます。

アプリケーションの一部である vCenter Server および vRealize Automation 内の 1 つ以上の仮想マシンをインストールした後に展開が失敗するか、アプリケーションが使用されていない場合は、vRealize Automation を使用してアプリケーションを分解できます。アプリケーション内のすべての仮想マシンはホストから削除されます。

vRealize Automation を使用した展開の分解では、Application Services の [展開] ページから展開レコードは削除されません。[展開] ページから展開レコードを削除するには、[「Application Services からアプリケーション展開を削除する \(P. 217\)」](#) を参照してください。

開始する前に

- テナント管理者として vRealize Automation コンソールにログインします。
- vRealize Automation の少なくとも 1 つの展開済みアプリケーションにアクセスできることを確認します。
- アプリケーションの一部である仮想マシンがクラウド内に存在していることを確認します。たとえば、一定日数の経過後に仮想マシンを削除するポリシーが環境に設定されていると、仮想マシンはすでに削除されている可能性があります。

手順

- 1 [アイテム] - [アプリケーションの展開] を選択します。
- 2 検索テキストボックスに分解する展開の名前を入力し、虫眼鏡アイコン (🔍) をクリックするか、Enter を押します。
- 3 検索結果リストから分解する展開を選択します。
[アイテムの詳細] ページが表示されます。
- 4 [アクション] メニューから [削除] を選択します。
[削除] ダイアログ ボックスに [申請情報] タブが表示されます。
- 5 [説明] テキスト ボックスに説明を入力します。
この必須の説明により、送信後に申請の進捗を監視することができます。
- 6 (オプション) [理由] テキスト ボックスに、申請の理由を入力します。
- 7 [次へ] をクリックします。
- 8 [展開の分解] タブの [分解オプションを選択] ラジオ ボタンを選択します。

オプション	説明
アシストによる分解	アプリケーションを使用して展開されたすべての仮想マシンを削除し、TEARDOWN ライフ サイクル ステージ スクリプトを実行します。
クイック分解	アプリケーションを使用して展開されたすべての仮想マシンを削除しますが、TEARDOWN ライフ サイクル ステージ スクリプトは実行しません。

- 9 [送信] をクリックします。
申請の確認のボックスが表示されます。
- 10 [OK] をクリックします。

次に進む前に

vRealize Automation コンソールの [申請] タブで分解申請のステータスを監視できます。

[展開] ページから展開のレコードを削除するには、[「Application Services からアプリケーション展開を削除する \(P. 217\)」](#) を参照してください。

Application Services からアプリケーション展開を削除する

クラウドからアプリケーションを削除した後、または特定の展開についての詳細情報が不要になった場合は、Application Services の [展開] ページで展開を削除できます。

Application Services の [展開] ページから展開を削除しても、展開済みのアプリケーション vApp とその vCloud Director の仮想マシン、vRealize Automation 仮想マシン、この仮想マシンと関連する vCenter Server の仮想マシン、および Amazon EC2 インスタンスがクラウドから削除されることはありません。クラウドからアプリケーションを削除する手順については、[「クラウドからのアプリケーションの分解 \(P. 214\)」](#) を参照してください。

開始する前に

アプリケーションの公開元および配備者として Application Services にログインします。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 削除する展開の名前を検索テキスト ボックスに入力します。
- 3 検索結果のリストから、削除する展開を選択します。
[展開サマリ] ページが開きます。

- 4 展開サマリの上のツールバーで、[操作] - [削除] を選択します。
- 5 [展開の削除] をクリックします。
- 6 最初にクラウドの展開済みアプリケーションを分解せずに [展開] ページから展開を削除する場合は、vCloud Director、vRealize Automation、または Amazon EC2 を使用して、クラウドに残っているコンポーネントを削除する必要があります。

オプション	説明
vCloud Director	vApp と関連する仮想マシンを削除します。
vRealize Automation	vRealize Automation 仮想マシン、およびこの仮想マシンと関連付けられた vCenter Server の仮想マシンを削除します。
Amazon EC2	展開のインスタンスを停止し、展開に対応する Security Group を削除して、展開のインスタンスに割り当てられた弾性 IP アドレスを解放します。

[展開] ページから展開レコードが削除されます。

展開または更新プロセスのキャンセル

展開または更新プロセスがいつまでも進行中で、展開ステータスの成功も失敗も示されない場合は、展開または更新プロセスおよびクラウド環境でのプロビジョニングを停止できます。

展開または更新プロセスを停止すると、ステータスが「停止中」に変わり、Application Services が展開または更新プロセスを正常に停止するまでこのステータスは変わりません。停止すると、ステータスは「停止」に変わります。Application Services は、展開または更新プロセスを削除しません。

複数展開は停止することも再開することもできません。

注意 長時間が経過しても展開ステータスが「停止」に変わらない場合は、プロセスがハングする可能性があります。[停止] ボタンを再度クリックすることで、展開を強制的に停止できます。長時間の経過後に展開ステータスが「停止」に変わらない場合には、プロセスを強制的に停止します。詳細については、『Application Services のトラブルシューティング』を参照してください。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- 展開または更新プロセスが進行中であることを確認します。

手順

- 1 キャンセルする展開または更新プロセスの名前を検索テキスト ボックスに入力します。
- 2 検索結果のリストから展開または更新プロセスを選択します。
[展開サマリ] ページが開きます。
- 3 展開サマリの上のツールバーで [停止] をクリックします。
- 4 選択内容を確認します。

注意 長時間が経過しても展開ステータスが「停止」に変わらない場合は、プロセスがハングする可能性があります。この場合、[停止] を再度クリックすることで強制的にハード停止を実行できます。2 回目 [停止] をクリックすると、停止のプロセスで使用される通常の一連のイベントを完了しなくても、ただちに展開を強制的に停止できます。プロンプトが表示されたら [OK] をクリックしてこのアクションを確認します。長時間の経過後に展開ステータスが「停止」に変わらない場合には、プロセスを強制的に停止します。

展開を通常の方法で停止すると、実行中のプロセスは完全に停止されます。強制的に停止すると展開は突然停止しますが、現在実行中のプロセスは引き続き実行されます。いずれの場合でも、関連付けられた仮想マシンはクリーンアップされません。

次に進む前に

展開を通常の方法で停止すると、展開を再開し、クラウドから分解し、vRealize Automation からアプリケーション展開レコードを削除することができます。第 19 章「[展開の管理 \(P. 209\)](#)」を参照してください。

更新プロセスについては、別の更新プロセスを開始することでその展開の操作を続行できます。第 17 章「[アプリケーション展開の更新 \(P. 179\)](#)」を参照してください。

ポリシー コンプライアンスの概要の表示

展開の全体的なコンプライアンス ステータスおよびそれに関連するポリシーの詳細を表示できます。

展開中に、ポリシー定義の SCAN ライフ サイクルスクリプトは、展開のコンプライアンス状態を評価します。全体的なコンプライアンス ステータスのアイコンは、展開のポリシー定義のコンプライアンスに違反するかどうかによって赤または緑に変化します。[コンプライアンスの概要] ページに表示されるステータスは、展開が開始されたときのアプリケーション ブループリントのスナップショットです。準拠しているポリシー定義と違反しているポリシー定義の数および重要度の高い定義と重要度の低い定義の数が、全体的なコンプライアンス ステータスに表示されます。関連する展開プロファイルは、全体的なステータスの横にあります。

全体的なコンプライアンス ステータスの上にあるポリシー スキャン タイムラインには、展開スキャンが開始された時間を示すタイム スタンプおよびその後の展開コンプライアンス スキャンが含まれます。たとえば、その後のスキャンは、カタログで更新されたポリシー定義が既存の展開に準拠しているかどうかをチェックします。

開始する前に

- アプリケーションの公開元および配備者として Application Services にログインします。
- ポリシー インスタンスが展開に適用されていることを確認します。

手順

- 1 Application Services タイトル バーで、ドロップダウン メニューをクリックして [展開] を選択します。
- 2 検索テキスト ボックスに、ポリシー定義を保持する展開の名前を入力します。
- 3 検索結果のリストから展開を選択します。
[展開サマリ] ページが開きます。
- 4 展開サマリの上のツールバーで、[展開ビュー] - [コンプライアンス ビュー] を選択します。
全体的なコンプライアンス ステータスが表示されます。テーブルに詳細なポリシー ステータスが一覧表示されます。
- 5 [ポリシー] 列で、コンプライアンスに違反するポリシー名をクリックし、コンプライアンス ログに詳細を表示します。
コンプライアンス ログの情報に基づいて、ポリシー定義における違反を修正してアプリケーションを展開できます。
たとえば、メモリ制限ポリシーに、アプリケーション展開の 10 の仮想マシンの中の 2 つの仮想マシンがポリシーに違反していることが示されているとします。ポリシーの [詳細] 列と [コンプライアンス ログ] セクションを表示して違反を特定し、アプリケーションに修正を適用します。

次に進む前に

展開が実行中の場合は、[展開サマリ] ページ から展開ステータスを追跡します。[[展開サマリ](#)] ページの使用 (P. 174)] を参照してください。

Application Services CLI の使用

Application Services CLI は、REST API を使用して HTTPS 経由で Application Services サーバと通信する Spring Roo ベースのクライアントです。

管理者ユーザーは、Application Services CLI を使用して Application Services アプライアンスを vRealize Automation サーバに登録します。アプリケーション公開者および展開者のロールが割り当てられたユーザー アカウントでは、CLI を使用してアプリケーションをクラウドから展開または分解できます。ユーザー ロールの詳細については、[第 5 章「ユーザーとグループの設定 \(P. 39\)」](#)を参照してください。

注意 **admin** ユーザーのパスワードは、アプライアンスを最初に起動したときに設定した管理者パスワードです。

Application Services のユーザー名とパスワードを ISO-8859-1 対応にする必要がある

Application Services ユーザー インターフェイスでは、ユーザー名とパスワードの値にユニコード文字セットを使用できますが、Application Services CLI は ISO-8859-1 文字セットのみをサポートします。Application Services CLI を使用する予定がある場合は、Application Services のユーザー名とパスワードに ISO-8859-1 文字セットに準拠する値を使用します。

この章では次のトピックについて説明します。

- [リモートでの CLI の開始 \(P. 221\)](#)
- [一般的な CLI オプション \(P. 222\)](#)
- [クラウド トンネルの管理 \(P. 223\)](#)
- [CLI によるアプリケーションの展開と更新 \(P. 224\)](#)
- [CLI を使用した展開の分解 \(P. 227\)](#)

リモートでの CLI の開始

Application Services CLI をリモート マシンから起動できます。

CLI をリモートから実行してサーバの負荷を軽減し、Application Services アプライアンスでの CLI の共有を避けるのがベスト プラクティスです。リモート マシンから CLI へ、セキュリティ保護された接続を使用します。

CLI コマンドの中には、英語以外の文字を含むことのできるビジネス グループまたは展開の名前を使用するものもあります。このような文字を表示するには、CLI を完全に国際化されたシェル クライアントまたは Putty や iTerm2 などのターミナルインターフェイスで実行する必要があります。また、環境変数 LC_CTYPE を **en_US.UTF-8** に設定して、英語以外の文字を入力できるようにします。

su - root で CLI にログオンすると、変数が自動的に設定されます。ハイフンを省略して **su root** でログオンした場合は、次のコマンドで変数を設定する必要があります。

```
export LC_CTYPE=en_US.UTF-8;
```

開始する前に

- Application Services アプライアンスのパスワードを把握していることを確認します。
- リモートマシンに Java JRE 1.7 がインストールされていることを確認します。
- リモートマシンが HTTPS を使用して Application Services アプライアンスに接続できることを確認します。

手順

- 1 `darwin-cli.jar` ファイルを Application Services サーバ `http://<DarwinServerIP>/tools/darwin-cli.jar` からリモートマシン上の書き込みが許可されているフォルダにダウンロードします。

`<DarwinServerIP>` は Application Services サーバ IP アドレスに置換してください。

セッション中は、CLI によってログファイルが作成されます。

- 2 コマンドプロンプトを開き、クライアントを起動します。

```
java -jar /<PathToJarFolder>/darwin-cli.jar
```

Darwin CLI バナーが表示され、`appd >` CLI プロンプトが表示されます。

- 3 Application Services サーバにログインします。

```
login --serverUrl https://<DarwinServerIP>:8443/darwin --username <UserName >--  
password <password >--tenantId <tenantid>
```

`login` コマンドまたはパスワードを追加できるコマンドに `--password` パラメータを指定して実行すると、指定したパスワードが現在のディレクトリにある `darwin-cli-history.log` ファイルにプレーンテキストとして保存されます。デフォルトでは、このログファイルは CLI によって削除されます。セキュリティを高めるため、このログファイルは削除してください。

`--password` パラメータを省略すると、パスワードの入力が求められます。パスワードは、プレーンテキストとしてシステムに保存されません。

`--tenantId` パラメータを使用すると、テナント名が `<tenantid>` のテナントにログインします。`--tenantId` パラメータを省略すると、テナントはデフォルトの `vsphere.local` になります。

コマンドラインプロンプトに、ユーザー名、ドメインおよびテナントが表示されます。

一般的な CLI オプション

Application Services のコマンドラインインターフェイスのオプションを使用すると、アプリケーションの展開、展開済みアプリケーションの更新、またはクラウドからのアプリケーションの分解ができます。

roo シェルプロンプトで CLI プログラムにログインした後に、`[Tab]` キーを押して、使用できるコマンドオプションのリストを表示します。コマンドオプションは、割り当てられたユーザーロールによって異なります。ユーザーロールとそのユーザーロールに許可される機能については、[第 5 章「ユーザーとグループの設定 \(P. 39\)」](#)を参照してください。

単語間にスペースを挿入して複数の単語を使用する場合は、それらの単語を引用符で囲みます。コマンドを入力した後は、多くの場合、コマンドが正常に実行されたどうかに加えて、さまざまな詳細情報が表示されます。

注意 CLI で使用できないコマンドオプションは使用しないでください。

表 20-1. 一般的な CLI コマンド

CLI コマンド	説明
<code>help</code>	使い方の情報を表示します。
<code>cliversion</code>	CLI のバージョン情報を表示します。
<code>login</code>	現在のユーザーを roo シェルにログインします。
<code>logout</code>	roo シェル プロンプトを閉じずに、現在のユーザーをログアウトします。 ログアウトして、別のユーザーとしてログインすることもできます。
<code>status</code>	ユーザーがログインしているかどうかを示します。ユーザーがログインしている場合はユーザー名が表示されます。
<code>exit</code>	CLI プログラムを終了します。

クラウド トンネルの管理

Application Services アプライアンスと Amazon EC2 VPC の Endpoint 仮想マシン間のセキュアなクラウド トンネル接続の作成、更新、有効化、削除などの操作を実行できます。

アプリケーション クラウド管理者として Application Services にログインします。

[「Amazon EC2 に接続するためのクラウド トンネルの作成 \(P. 86\)」](#) を参照してください。

表 20-2. CLI によるクラウド トンネルの管理

CLI コマンド	説明
<code>create-cloud-tunnel --name <TunnelName> --description <"TunnelDescription"> --enabled false --externalAddress <EndpointVMElasticIP> --sshPort 22 --internalAddress <EndpointVMPrivateIP> --proxyUrl <ProxyURL> --username <UserName> --privateKeyPath <PrivateKeyFilePath></code>	セキュアなクラウド トンネルを作成します。 たとえば、次のコマンドを使用して、EC2 Tunnel という名前のクラウド トンネルを作成できます。 <code>create-cloud-tunnel --name EC2 Tunnel --description <"Test EC2 tunnel connection"> --enabled false --externalAddress 50.18.100.100 --sshPort 22 --internalAddress 192.0.2.255 --proxyUrl http://proxy.vmware.com:3128 --username root --privateKeyPath /tmp/private_key_file</code>
<code>update-cloud-tunnel --name <PreviousTunnelName> --new-name <NewTunnelName> --description <"TunnelDescription"> --externalAddress <EndpointVMElasticIP> --sshPort 22 --internalAddress <EndpointVMPrivateIP> --proxyUrl <ProxyURL> --username <UserName> --privateKeyPath <PrivateKeyFilePath></code>	既存のクラウド トンネルのさまざまなパラメータの値を変更します。 大半のパラメータ値は、クラウド トンネルが無効になっている場合だけ更新できます。
<code>list-cloud-tunnels --name <TunnelName></code>	使用可能なすべてのクラウド トンネルの一覧または特定のクラウド (クラウド名を指定した場合) を取得します。 特定のクラウド トンネルの詳細を表示するには、 <code>list-cloud-tunnels --name EC2 Tunnel</code> というコマンドを使用します。
<code>enable-cloud-tunnel --name <TunnelName></code>	既存のクラウド トンネルを有効にします。 このコマンドは、セキュアなクラウド トンネル接続を確立するバックグラウンド動作を開始します。

表 20-2. CLI によるクラウド トンネルの管理 (続き)

CLI コマンド	説明
<code>disable-cloud-tunnel --name <TunnelName></code>	既存のクラウド トンネルを無効にします。 このコマンドは、セキュアなクラウド トンネル接続を切断するバックグラウンド動作を開始します。
<code>delete-cloud-tunnel --name <TunnelName></code>	既存のクラウド トンネルを削除します。 クラウド トンネルを削除するには、まず、無効にする必要があります。 Application Services アプライアンスと Amazon EC2 環境間のクラウド トンネル接続を削除します。
<code>test-cloud-tunnel --name <TunnelName></code>	クラウド トンネルのセキュアな SSH 接続が確立されているかどうかを確認し、現在のステータスをレポートします。 クラウド トンネルを有効または無効にすると、バックグラウンドで接続動作または切断動作が開始されます。このコマンドは、バックグラウンド動作のステータスをレポートします。また、展開時に、Application Services との接続が正しく機能するかどうかを確認します。 クラウド トンネルを有効または無効にしたら、このコマンドを実行して、有効化または無効化が成功していることを確認します。

CLI によるアプリケーションの展開と更新

アプリケーションを展開するには、アプリケーション公開者または展開者としてログインする必要があります。

アプリケーションを展開する前に、完成し保存された検証エラーのない展開プロファイルが用意されていることを確認します。アプリケーションには名前も必要です。更新プロセスを起動してクラスタ化されたノードの規模を拡大/縮小する前に、展開済みアプリケーションにクラスタ ノードが含まれていることを確認します。

アプリケーションのクイック展開、アプリケーション展開のステータスの確認、展開済みアプリケーションのスケールアウトの開始、展開済みアプリケーションのサービスおよびアプリケーション コンポーネントの構成変更は、CLI を使用して実行できます。Application Services ユーザー インターフェイスを使用して、これらすべてのタスクを実行できます。[「アプリケーションをクイック展開する \(P. 169\)」](#)、[「\[展開サマリ\] ページの使用 \(P. 174\)」](#)、[「更新プロセスの開始と展開のスケールアウト \(P. 180\)」](#)、または [「更新プロセスの開始と構成の変更 \(P. 185\)」](#) を参照してください。

また、vCloud Director、vRealize Automation、および Amazon EC2 のユーザー インターフェイスで展開済み仮想マシンのステータスを表示することもできます。

表 20-3. CLI からのアプリケーションの展開または更新

CLI コマンド	説明
<code>deploy-application --destination <ApplicationName>--<ApplicationVersion>--<ProfileName></code>	アプリケーション ブループリントを展開します。 たとえば、Clustered Dukes Bank アプリケーション バージョン 2.1.0 を prod-dep 展開プロファイルを使用して展開するには、 deploy-application --destination "Clustered Dukes Bank App-2.1.0-prod-dep" というコマンドを使用します。 deploy と入力して Tab キーを押すと、コマンドの後続部分のオプションが表示されます。表示された選択肢リストで、スペースで区切られた複数の単語に引用符が付いていない場合は、入力時に引用符で囲んでください。
<code>deployment-status --deploymentName <DeploymentName></code>	最新展開のステータスを表示します。 アプリケーションの展開に失敗した場合は、仮想マシン固有のログ ファイルを収集して問題のトラブルシューティングを行います。 『VMware vCloud Automation Center Application Services トラブルシューティング ガイド』を参照してください。

表 20-3. CLI からのアプリケーションの展開または更新 (続き)

CLI コマンド	説明
<pre> deploy-application --destination <ApplicationName><ApplicationVersion>-- <DeploymentProfileName> --propertiesFile <FileName>.xml </pre>	<p>新しいプロパティ値を指定して、最新のアプリケーション バージョンを展開します。</p> <p>展開ファイルを作成して、必須展開および展開時にオーバーライド可能な展開の新規プロパティ値を定義できます。次のコード例では、Apache_LB および AppServer ノードの global_conf、http_node_port、および http_server_port プロパティの新しい値を定義しています。</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <ns1:config-update-properties xmlns:ns1="http://www.test.com/darwin/schema/beans/api"> <node> <name>load_balancer</name> <node-component> <name>Apache_LB</name> <property> <key>http_proxy_port</key> <value>9001</value> </property> </node-component> </node> <node> <name>appserver</name> <node-component> <name>Dukes_Bank_App</name> <property> <key>EAR_FILE</key> <value><![CDATA[http://192.10.1.113/Share/UPRConfig/CDB/dukesbank-2b-Update1-IndexPage.ear]]></value> </property> <property> <key>JAR_FILE</key> <value><![CDATA[http://192.10.1.113/Share/UPRConfig/CDB/mysql-connector-java-5.1.8.jar]]></value> </property> </node-component> </node> </ns1:config-update-properties> </pre> <p>たとえば、Clustered Dukes Bank アプリケーション バージョン 2.1.0 を staging-dep 展開プロファイルと new.props プロパティファイルを使用して展開するには、deploy-application --destination " Clustered Dukes Bank App-2.1.0-DP_MAN_VCD" --propertiesFile ~/new.props.xml というコマンドを使用します。</p> <p>deploy と入力して Tab キーを押すと、コマンドの後続部分のオプションが表示されます。表示された選択肢リストで、スペースで区切られた複数の単語に引用符が付いていない場合は、入力時に引用符で囲んでください。</p>

表 20-3. CLI からのアプリケーションの展開または更新 (続き)

CLI コマンド	説明
<pre>update-scaleout --deploymentName <DeploymentName> --propertiesFile <FileName>.xml</pre>	<p>更新プロセスを起動して、展開済みアプリケーションを拡大/縮小します。</p> <p>ノードの新しいクラスタ サイズを定義するプロパティ ファイルを作成する必要があります。次のコード例では、AppServer1 ノードと DBServer1 ノードのクラスタ サイズを定義しています。</p> <pre><?xml version="1.0" encoding="UTF-8"?> <ns1:scaleout-properties xmlns:ns1="http://www.test.com/darwin/sche ma/beans/api"> <cluster-node> <name>appserver</name> <scale-out-by>1</scale-out-by> </cluster-node> </ns1:scaleout-properties></pre> <p>たとえば、Clustered Dukes Bank アプリケーション バージョン 2.1.0 を scaleout プロパティ ファイルを使用してスケールアウトするには、update-scaleout --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-6-cc0a3b20-43a0-4a22-be3e-49d4f31ab8e8" --propertiesFile ~/scaleout.xml というコマンドを使用します。</p> <p>update と入力して Tab キーを押すと、コマンドの後続部分のオプションが表示されます。表示された選択肢リストで、スペースで区切られた複数の単語に引用符が付いていない場合は、入力時に引用符で囲んでください。</p>
<pre>update-config --deploymentName <DeploymentName> -- configUpdatePropertiesFile <ConfigUpdatePropertiesFileName>.xml</pre>	<p>更新プロセスを起動して、展開済みアプリケーションの既存のサービスまたはアプリケーション コンポーネントの構成を変更します。</p> <p>既存のサービスを構成するには、構成更新プロパティ ファイルを作成する必要があります。次のコード例では、AppServer ノードの vFabric tc Server サービスで global_conf プロパティと db_port プロパティを変更しています。</p> <pre><?xml version="1.0" encoding="UTF-8"?> <ns1:config-update-properties xmlns:ns1="http://www.test.com/darwin/sche ma/beans/api"> <node> <name>load_balancer</name> <node-component> <name>Apache_LB</name> <property> <key>http_proxy_port</key> <value>9001</value> </property> </node-component> </node> </ns1:config-update-properties></pre> <p>たとえば、Clustered Dukes Bank アプリケーション バージョン 2.1.0 の構成を、configupdate プロパティ ファイルを使用して変更するには、update-config --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-2-19d63535-673e-4766-b380-de4e6ec3676a" --configUpdatePropertiesFile ~/configupdate.xml というコマンドを使用します。</p> <p>update と入力して Tab キーを押すと、コマンドの後続部分のオプションが表示されます。表示された選択肢リストで、スペースで区切られた複数の単語に引用符が付いていない場合は、入力時に引用符で囲んでください。</p>

CLI を使用した展開の分解

展開を分解するには、アプリケーション公開者または展開者としてログインする必要があります。

クラウドから展開済みのアプリケーションを分解する前に、展開済みアプリケーションの展開名を確認します。分解プロセスのステータスは、Application Services ユーザー インターフェイスで監視できます。[「クラウドからのアプリケーションの分解 \(P. 214\)」](#) を参照してください。

表 20-4. CLI で展開を削除する

CLI コマンド	説明
<code>teardown --deploymentName <DeploymentName></code>	<p>vCloud Director から展開済みのアプリケーションを分解します。</p> <p>たとえば、<code>teardown --name appd-TestApp-1.0.0-admin-3-a99309aa-aa5b-45c2-b6a0-bac4a421178e</code> コマンドでクラウドからアプリケーションを削除します。</p> <p>teardown と入力して Tab キーを押すと、コマンドの後続部分のオプションが表示されます。表示された選択肢リストで、スペースで区切られた複数の単語に引用符が付いていない場合は、入力時に引用符で囲んでください。</p>

CLI インポート/エクスポート機能の使用

事前構成済みのアプリケーション ブループリントとそれに関連する展開プロファイル、サービス、外部サービス、ポリシー、成果物リポジトリ、および使用可能なカスタム タスクを Application Services インスタンス間でインポートおよびエクスポートすると、アプリケーションをより詳細にカスタマイズできるようになります。

Application Services 6.1 の異なるインスタンス間でパッケージをインポートおよびエクスポートしたり、Application Services 5.0、5.2、および 6.0 のパッケージを 6.1 にインポートしたりできます。

CLI を Application Services リモート マシンから起動できます。完全に国際化されたシェル クライアントやターミナル インターフェイスを使用している場合は、英語以外の文字を含む CLI コマンドも実行できます。[「リモートでの CLI の開始 \(P. 221\)」](#) を参照してください。

Application Services パッケージをインポートおよびエクスポートするには、アプリケーション アーキテクトやアプリケーション カタログ管理者としてログインする必要があります。

インポートまたはエクスポート パッケージのサイズが、インスタンスで使用できるメモリより大きい場合は、メモリの割り当てに関するエラー メッセージが表示されます。たとえば、`java -Xmx6000m -jar darwin-cli.jar` コマンドを使用して、インスタンスに 6 GB のメモリを割り当てることができます。

■ CLI エクスポート コマンドの使用 (P. 229)

CLI エクスポート コマンドを実行すると 1 つのパッケージが作成されます。このパッケージは、アプリケーション、アプリケーションに関連するブループリントと展開プロファイル、外部サービス、ポリシー、成果物リポジトリ、サービス、論理テンプレート、および異なる Application Services インスタンス間でのエクスポートに利用できるカスタム タスクで構成されています。

■ CLI インポート コマンドの使用 (P. 231)

CLI インポート コマンドを実行すると、1 つのパッケージがインポートされます。このパッケージには、アプリケーション、アプリケーションに関連するブループリントと展開プロファイル、外部サービス、ポリシー、成果物リポジトリ、サービス、論理テンプレート、および異なる Application Services インスタンス間で使用可能なカスタム タスクが含まれています。

CLI エクスポート コマンドの使用

CLI エクスポート コマンドを実行すると 1 つのパッケージが作成されます。このパッケージは、アプリケーション、アプリケーションに関連するブループリントと展開プロファイル、外部サービス、ポリシー、成果物リポジトリ、サービス、論理テンプレート、および異なる Application Services インスタンス間でのエクスポートに利用できるカスタム タスクで構成されています。

エクスポート パッケージには、圧縮アーカイブ ファイル形式と非圧縮ファイル形式の 2 つの形式があります。圧縮形式には、デフォルトの **.zip** ファイル タイプがあります。非圧縮形式は XML ファイルで、どのようなファイル タイプでも対応できます。import-package コマンドでは、両方の形式が許可されます。

名前にタブ、アンダースコア、改行、またはキャリッジリターン文字が含まれる名前のオブジェクトはエクスポートできません。

アプリケーション、サービス、外部サービス、スクリプトタスク、またはポリシーバージョンをエクスポートする場合、セキュアなプロパティがすべてデフォルトで削除され、パスワードなどの機密情報が別の Application Services インスタンスにエクスポートされることを防ぎます。

インポート コマンドを実行すると、パッケージ全体が、指定されたサーバにインポートされます。エクスポート パッケージの要素を選択的にインポートすることはできません。

export-package コマンド

CLI export-package コマンドの形式は次のとおりです。

```
export-package --exportFilePath <pathname> --fromGroup <obgname> ---<CommandOption> [--<CommandOption>] ...] [--uncompressed]
```

次の表で、export-package コマンドの必須コンポーネントについて説明します。

CLI コマンド コンポーネント	説明
export-package	export-package コマンド名。
--exportFilePath <pathname>	作成するエクスポートパッケージのパス名を指定します。たとえば、 --exportFilePath /home/dev/joomla.zip では、 joomla.zip エクスポートパッケージの場所と名前を指定します。エクスポートパッケージに .xml ファイル拡張子がある場合は、 --uncompressed オプションを使用する必要があります。
--fromGroup <busgroupname>	エクスポート対象のオブジェクトを所有するビジネス グループを指定します。ポリシーは除外されます。テナントのどのユーザーもポリシーにアクセスできるため、所有するビジネス グループは無視されます。
<---CommandOption>	エクスポートするオブジェクトのタイプ、名前、バージョンを指定します。たとえば、コマンド オプション --applicationVersion distmaps:1.0.0 では、 distmaps という名前、 1.0.0 というバージョンのアプリケーションを指定します。export-package コマンドには、複数のコマンド オプションを指定できますが、少なくとも 1 つのコマンド オプションの指定が必要です。

次の表で、1 つ以上のオブジェクトをエクスポートする際に使用できるコマンド オプションについて説明します。

CLI コマンド オプション	説明
--applicationVersion <Name>:<VersionID>	<p>エクスポートするオブジェクトとバージョンを指定します。</p> <ul style="list-style-type: none"> ■ <Name> は、エクスポートするアプリケーション、サービス、スクリプトタスク、外部サービス、ポリシー、または成果物リポジトリの名前です。名前は大文字と小文字が区別されます。名前にスペースが含まれる場合は、二重引用符で囲みます。 ■ <VersionID> は、フォーム <major>.<minor>.<micro> {-<qualifier>} のオブジェクトのバージョン ID です (たとえば、1.0.0-beta)。 <p>バージョン名をコンマで区切ることで、単一のエクスポート コマンドで同じタイプの複数のオブジェクトをエクスポートできます。バージョン名のリストのバージョン名にスペースがある場合は、リスト全体を二重引用符で囲みます。</p> <pre>export-package --exportFilePath pkgname.zip --fromGroup Dev --applicationVersion "aname1:1.0.0,aname two:1.0.0,aname3:1.0.0"</pre> <p>同じエクスポート コマンドで複数のタイプのオブジェクトをエクスポートすることもできます。</p> <pre>export-package --exportFilePath pkgname.zip --fromGroup Dev --applicationVersion aname:1.0.0 --serviceVersion sname:1.0.0 --scriptTaskVersion stname:1.0.0</pre>
--serviceVersion <Name>:<VersionID>	
--scriptTaskVersion N<ame>:<VersionID>	
--externalServiceVersion <Name>:<VersionID>	
--policyVersion <Name>:<VersionID>	

CLI コマンド オプション	説明
<code>--repoVersion <Name>:1.0.0</code>	ALL keyword は、指定のビジネス グループの全バージョンのオブジェクト タイプをエクスポートします。 <code>export-package --exportFilePath pkgname.zip --fromGroup Dev</code> <code>--applicationVersion ALL --serviceVersion ALL --scriptTaskVersion ALL</code>
<code>--uncompressed</code>	パッケージを非圧縮形式でエクスポートします。 <code>.xml</code> ファイル拡張子のエクスポートパッケージに必要です。

依存関係のエクスポート

オブジェクトをエクスポートする場合は、その依存関係もエクスポートします。アプリケーションのような上位オブジェクトの場合、依存関係にはサービス、仮想マシン テンプレート、スクリプト タスクなどが含まれることがあります。エクスポート コマンドによって、上位オブジェクトの所有ビジネス グループは検証されますが、別のビジネス グループに属する可能性のある依存オブジェクトの所有ビジネス グループは検証されません。このコマンドによって、上位オブジェクトの一部として依存オブジェクトがそのままエクスポートされます。たとえば、アプリケーションが開発ビジネス グループに属しているけれども、アプリケーションのサービスが生産ビジネス グループに属しているとします。開発グループのメンバーは、そのサービスを含むアプリケーションをエクスポートできます。ただし、生産ビジネス グループのメンバーは、サービスだけエクスポートすることができ、アプリケーションはエクスポートできません。もちろん、両方のビジネス グループのメンバーが、両方のオブジェクトを一緒に、または別々にエクスポートすることができます。

エクスポートの要件

次の表では、オブジェクトをエクスポートまたはインポートするために、所有ビジネス グループに属する必要がある場合を示します。

エクスポートされたオブジェクト	ビジネス グループの要件
アプリケーション サービス 外部サービス VmTemplate	次のタスクでは、所有ビジネス グループのメンバーシップが必要です。 <ul style="list-style-type: none"> ■ オブジェクトをエクスポートする。 ■ プライベート オブジェクトを依存関係としてエクスポートする。 次のタスクでは、テナントの任意のビジネス グループのメンバーシップが許可されます。 <ul style="list-style-type: none"> ■ 共有アプリケーションまたはサービスを表示する。 ■ 共有オブジェクトを依存関係としてエクスポートする。
スクリプト タスク (常に共有)	次のタスクでは、所有ビジネス グループのメンバーシップが必要です。 <ul style="list-style-type: none"> ■ オブジェクトをエクスポートする。 次のタスクでは、テナントの任意のビジネス グループのメンバーシップが許可されます。 <ul style="list-style-type: none"> ■ オブジェクトを表示する。
ポリシー (常に共有)	次のタスクでは、テナントの任意のビジネス グループのメンバーシップが許可されます。 <ul style="list-style-type: none"> ■ オブジェクトをエクスポートする。
成果物リポジトリ (常に共有)	次のタスクでは、所有ビジネス グループのメンバーシップが必要です。 <ul style="list-style-type: none"> ■ オブジェクトをエクスポートする。

CLI インポート コマンドの使用

CLI インポート コマンドを実行すると、1 つのパッケージがインポートされます。このパッケージには、アプリケーション、アプリケーションに関連するブループリントと展開プロファイル、外部サービス、ポリシー、成果物リポジトリ、サービス、論理テンプレート、および異なる Application Services インスタンス間で使用可能なカスタム タスクが含まれています。

エクスポート コマンドで作成したパッケージのみインポートできます。エクスポート コマンド以外の他の方法で変更または作成したパッケージはインポートできません。

オブジェクトをインポートするには、インポート パッケージの各オブジェクトの所有ビジネス グループに属する必要があります。

import-package コマンドを実行すると、パッケージ全体がターゲット サーバにインポートされます。パッケージからオブジェクトを選択的にインポートすることはできません。

同じ名前の外部サービスを複数の異なるビジネス グループにインポートすることはできず、外部サービスが他のビジネス グループによって所有されている場合、その外部サービスを依存関係としてインポートすることはできません。

- ソース アプライアンスに展開プロファイルを含むアプリケーションが、共有アプリケーションか、ターゲット アプライアンスのターゲット グループでプライベートに所有されているアプリケーションであることを確認します。ターゲット グループ以外のビジネス グループで所有されているプライベート アプリケーションへの展開プロファイルのインポートはサポートされていません。
- 展開プロファイルの作成時に使用されたブループリントがインポート時に変更されないようにしてください。ブループリントがわずかでも変更されていると、ノードとコンポーネントはマッピングできず、この使用事例はサポートされません。

展開プロファイルを、共有アプリケーションの依存関係として、またはターゲット ビジネス グループで所有されるプライベート アプリケーションの依存関係としてインポートできます。また、展開プロファイルの作成時に使用されたブループリントがインポート中に変更されないようにしてください。わずかでも変更されると、ノードとコンポーネントがマッピングできなくなります。展開プロファイルをインポートする前に、実行展開プランに追加したカスタム タスクを保存します。[「実行プランを確認しカスタム タスクを追加する \(P. 162\)」](#) を参照してください。

インポート プロセスを正常に終了するには、パッケージのオブジェクトに ASCII 文字のみを含めるようにします。

import-package コマンド

CLI import-package コマンドの形式は次のとおりです。

```
import-package --importFilePath <pathname> --targetGroup <obgname> --
ConflictResolutionAction <action>
```

次の表で、import-package コマンドの必須コンポーネントについて説明します。

CLI コマンド コンポーネント	説明
import-package	import-package コマンド名。
--importFilePath <pathname>	インポート パッケージのパス名を指定します。たとえば、 --importFilePath /home/dev/joomla.zip では、 joomla.zip インポート パッケージの場所と名前を指定します。
--targetGroup <busgroupname>	インポートするオブジェクトの所有ビジネス グループを指定します。ポリシーは除外されます。テナントのどのユーザーもポリシーにアクセスできるため、所有するビジネス グループは無視されます。
--ConflictResolutionAction <action>	CHECK、SKIP、OVERWRITE、または IMPORTASNEW の競合解決の操作を指定します。import-package コマンドで、競合解決の操作を 1 つ指定する必要があります。

次の表で、オブジェクトをインポート パッケージでインポートするときに使用する競合解決の操作について説明します。

競合解決の操作	説明
CHECK	インポート パッケージのすべてのオブジェクトを表示して、オブジェクトがターゲット上にあるかどうかを示します。オブジェクト名とバージョン ID を比較して、一致を判別します。バージョン ID のフォームは <major>.<minor>.<micro>.<qualifier> です (たとえば、 2.0.1.beta)。
SKIP	オブジェクトがターゲットにない場合、そのオブジェクトをパッケージからコピーします。オブジェクトがターゲットにある場合、そのオブジェクトはコピーされません。コピーされたオブジェクトは、 --TargetGroup で指定された所有ビジネス グループに属します。

競合解決の操作	説明
OVERWRITE	オブジェクトがターゲットにある場合、そのオブジェクトのコンテンツは、パッケージのコンテンツで上書きされます。ターゲットにないオブジェクトを、 --TargetGroup で指定された所有ビジネス グループにコピーします。ログイン ユーザーがオブジェクトをインポートするには、パッケージのオブジェクトの所有ビジネス グループに属する必要があります。
IMPORTASNEW	オブジェクトがターゲットにない場合、パッケージのオブジェクトを新しい名前でターゲットにコピーします。新しいオブジェクトは、 --TargetGroup で指定された所有ビジネス グループに属します。名前変更したオブジェクトには、 --suffix オプションでサフィックスを指定する必要があります。

次の表で、使用可能なコマンド オプションについて説明します。

CLI コマンド オプション	説明
--shared	テナントの全ビジネス グループのメンバーで、パッケージ内の新しいアイテムを共有します。省略した場合は、パッケージの新しいアイテムは、所有ビジネス グループのメンバーのプライベートなアイテムになります。
--suffix <sfxtext>	下線、および <sfxtext> で表される指定のテキストを、パッケージから一致としてコピーされた各オブジェクトの末尾に追加します。たとえば、 --suffix NEW を指定して、 apache:1.0.0 をパッケージからコピーすると、オブジェクトの新しい名前はバージョン 1.0.0 の apache _NEW になります。新しいオブジェクトの所有ビジネス グループは、 --targetGroup で指定したターゲット グループです。

例: SKIP および OVERWRITE

```
import-package --importFilePath /home/dev/dukes.zip --conflictResolutionAction SKIP --
targetGroup Development
import-package --importFilePath /home/dev/dukes.zip --conflictResolutionAction
OVERWRITE --targetGroup Development
```


インデックス

記号

`${random}` 139

数字

3 階層アプリケーション、サンプル 194

A

Active Directory ドメイン

参加 76

事前定義済みタスクの追加 203

Amazon EC2

アプリケーションのプロビジョニング 83

カスタム AMI 88

クラウド プロバイダの登録 90

展開プロファイル 157

ネットワーク 158

Amazon EC2 との接続 86

Amazon EC2 AMI のテンプレート、作成 89

AMI、作成 89

Application Services

Web インターフェイス 45

Web インターフェイスを開く 31

アーキテクチャの原理 15

新しいリリースにアップグレードする 37

アプライアンスの起動 29

アプリケーションの作成 138

概要 13

カスタム タスクの追加 132

再起動しています 34

使用 9

シリアル番号の入力 29

設定 21

用語集 16

ライブラリの管理 107

Application Director OVF、展開 26

Application Services アプライアンス

darwin_user パスワードの作成 29

vCloud Application Center サーバの登録 29

vRealize Automation サーバへの登録 30

構成 29

展開 28

Application Services コンポーネント、開発 93

Application Services のオブジェクト、共有 41

APT Repository Config、事前定義済みタスクの追加 204

B

Bash、サポート対象のスクリプト 94

BeanShell、サポート対象のスクリプト 94

C

CentOS、サンプル テンプレート 193

CHECK オプション、インポート 231

CLI

Application Services の登録 35

アプリケーションをデプロイ 224

一般的なコマンド 222

インポート 229

エクスポート 229

クイック展開の使用 224

更新プロセス 224

展開 ステータス 224

展開の分解 227

ユーザーとグループの管理 221

ユーザーとグループの作成 221

ユーザー名とパスワードに ISO-8859-1 文字セットを使用する 221

CLI (コマンドライン インターフェイス) 223

CLI)、インポート 231

CLI プログラムの起動、リモート 221

Clustered DotShoppingCart

サンプル アプリケーションの構成の変更 201

展開 196

Clustered Dukes Bank、更新プロセス 200

Clustered Dukes、サンプル アプリケーションの構成の変更 201

Clustered Dukes Bank アプリケーション、展開 197

コンポーネント

開発 93

ベスト プラクティス 105

D

darwin_user アカウント、ロック解除 33

Default グループ、ログイン 45

DHCP、構成 76

DotShoppingCart、更新プロセス 200

E

Endpoint 仮想マシン、作成 92

G

global_conf 162

I

IMPORTASNEW オプション、インポート 231
IP アドレス、事前定義済みプロパティ 103

J

JRE

インストール 74
カスタム テンプレートの要件 88
ダウンロード 74

JRE の要件

カスタム テンプレート 50
カスタム テンプレートのための 72

L

libpython パッケージ、ダウンロード 194
Linux VM、テンプレートの更新 58
Linux 仮想マシン テンプレート、アップデート 79
Linux テンプレート、作成 77, 89

N

Nanotrader

更新プロセス 200
サンプル アプリケーションの構成の変更 201
展開 195

NIC、複数の定義 144

O

OOB (out-of-the-box)、サービス 16

P

PowerShell、サポート対象のスクリプト 94
Puppet サービス、ライブラリへのインポート 116
Puppet マスター
 インスタンスの作成 65
 準備 63
 ソリューション インスタンスの登録 68
 登録 63

R

Radiant CMS、展開 199
REST API
 ユーザーとグループの管理 221
 ユーザーとグループの作成 221
RHN の登録、事前定義済みタスクの追加 207

S

SID、有効化 76
SKIP オプション、インポート 231
SQLFire
 jPetStore アプリケーションの展開 198
 vFabric_SQLFire-1.0.jar のダウンロード 198
SSH 接続、Ubuntu 仮想マシンの有効化 194

U

Ubuntu、サンプル テンプレート 193
URL、ファイアウォール外部のアクセス 34

V

vCenter Server、仮想マシン テンプレート 51
vCenter Server の Windows 仮想マシン テンプレート
 エージェント ブートストラップのインストール 52
 エージェント ブートストラップのダウンロード 52
 カスタマイズ仕様 52
vCloud Application Director
 ライブラリへのポリシーの追加 128
 論理テンプレートの追加 125
vCloud Application Services、インストール 23
vCloud Automation Center
 Application Services アプライアンスへの登録 29
 Application Services の登録 35
 SSO 170
 アプリケーションのプロビジョニング 49
 カタログ アイテムの申請 173
 クラウド プロバイダ 58
 公開 170
 サービス カatalog への公開 170
 資格 170
 展開プロファイル 157
 テンプレート 58
 登録 35
 ビジネス グループ 58
 複合展開プロファイルの公開 172
 ブループリント 49
 予約 60
 予約ポリシー 60
vCloud Director
 アプリケーションのプロビジョニング 71
 カスタム テンプレートの作成 72
 クラウド プロバイダ 80
 クラウド プロバイダとテンプレートの登録 80
 構成 27
 設定の確認 27
 展開環境の作成 81
 テンプレート 80
 ネットワーク 158
vCloud Director カatalog、Windows テンプレート
 の追加 77
仮想マシン
 エンドポイント 85, 86
 Linux 84
 NAT 84
 カスタム テンプレートの要件 72
 再起動 122
 削除 216
 詳細の表示 211
 テンプレート 79
 分解 216
VMware Tools 74
VMware Tools の要件、カスタム テンプレートのため
 の 72

VMware vCenter クラスタ

構成 25

単一ホスト 25

複数のホスト 25

vRealize Automation

gugent 52, 54

カスタム テンプレート 50

ゲスト エージェント 52, 54

ネットワーク 158

ブループリント 51

W

Web インターフェイス

Application Services 45

Application Services を開く 31

検索基準 46

Web ブラウザ、サポートされているバージョン 24

Windows

仮想マシン テンプレート 73, 76

仮想マシン テンプレートの追加 77

Windows CMD、サポート対象のスクリプト 94

Windows テンプレート、構成 74

Windows VM、既存のテンプレートの更新 58

Windows 仮想マシンテンプレート、アップデート 79

Y

YUM Repository Config、事前定義済みタスクの追加 205

あ

アーキテクチャの原理、Application Services 15

アクション

コンポーネントのライフ サイクル ステージ 93

ライフ サイクル ステージ 93

アクセス キー 84

アップグレード

アップグレード インストール プログラムの実行 37

インストールのアップグレード プログラムの使用後にロールバック 38

アプライアンス、Application Services の起動 29

アプリケーション

エクスポート 229

共有 138

クイック展開 169

クラウドからの仮想マシンの削除 214

構成 141

コピー 147

サービス カタログへの公開 170

作成 137

成果物を使用した展開 155

展開 157, 168

展開の詳細の表示 210

展開プロファイル 157, 164, 169

展開プロファイルの設定 157

バージョン 138

複合展開プロファイルを使用した展開 168

分解 214

アプリケーション コンポーネント、成果物の管理 154

アプリケーション展開

更新 179

更新プロセスのロールバック 189

更新プロファイルの昇格 188

構成の変更 185

スケールアウト 180

スケールイン 182

変更する更新プロセスの開始 185

アプリケーションの展開、削除 217

アプリケーションのブループリント、作成 142

アプリケーションのプロビジョニング

Amazon EC2 83

vCloud Automation Center 49

vCloud Director 71

アプリケーション バージョン

コピー 146

削除 147

アプリケーション ブループリント 139

アレイ、プロパティ 95

い

依存関係、作成 143

一時仮想マシン、外部サービス 158

違反、ポリシー定義の表示 219

インストール、準備 23

インストールの前提条件 23

え

エージェント ブートストラップ

インストール 74

再起動 122

ダウンロード 74

エージェント ブートストラップ サービス

カスタム テンプレート 50

カスタム テンプレートの要件 88

エージェント ブートストラップのインストール、

vCenter Server の Windows 仮想マシン テンプレート 52

エージェント ブートストラップのダウンロード、

vCenter Server の Windows 仮想マシン テンプレート 52

エラー

構成の変更のトラブルシューティング 191

展開のスケールリングのトラブルシューティング 191

お

オペレーティング システム

Application Services ライブラリ 108

カスタム テンプレート 50

カスタム テンプレートのための 72, 88

か

- 外部 URL、ファイアウォール外部のアクセス 34
- 外部 URL のプロキシ、構成 34
- 外部サービス
 - コピー 123
 - 削除 124
 - 作成 118
 - 追加 146
 - ブループリント 146
 - 編集 123
 - マッピング 61
 - ライブラリへの追加 117
- 外部サービス インスタンス 60
- 外部サービス バージョン、エクスポート 229
- 外部サービス プロパティ、定義 119
- 確認、ポリシー定義の表示 219
- カスタマイズ仕様、vCenter Server の Windows 仮想マシンテンプレート 52
- カスタム タスク
 - エクスポート 229
 - コピー 134
 - 削除 134
 - 追加 132, 162
 - 編集 134
- カスタム テンプレート
 - JRE の要件 50, 72, 88
 - VMware Tools の要件 72
 - エージェント ブートストラップ サービス 50
 - エージェント ブートストラップの要件 88
 - オペレーティング システム 50
 - オペレーティング システムのための 72
 - 仮想マシンの要件 72
- カスタム テンプレートのためのエージェント ブートストラップ サービスの要件 72, 88
- 仮想データセンター (vDC) 81
- 仮想データセンター (VCD)、作成 81
- 仮想プライベート クラウド (VPC)
 - 展開環境の定義 92
 - 登録 90
- 仮想マシン テンプレート
 - Amazon EC2 の作成 89
 - SID の有効化 76
 - vCenter Server 51
 - Windows の場合 73
 - アップデート 79
 - 更新 58
 - ドメインへの参加の有効化 76
- 仮想マシンのエンドポイント、Application Services での使用 83
- 可用性ゾーン
 - Amazon EC2 のテンプレート 90
 - 展開環境の定義 92
- 環境変数、スクリプトでの使用 95

き

- 既存の展開プロファイル 164
- 基本アプリケーション ブループリント 139
- 基本の外部サービス、作成 118
- 共有オプション、インポート 231

く

- クラウドからアプリケーション、削除 214
- クラウド抽象化レイヤ (CAL)、Application Services 13
- クラウド トンネル
 - Application Services での使用 83
 - 管理 223
- クラウド プロバイダ
 - vCloud Automation Center 58
 - 登録 90
- クラウド論理テンプレート、ライブラリへの追加 125
- クラスタ化ノード、指定 143
- グループ
 - 管理 221
 - 作成 221
 - 設定 39
- クローン、展開 51
- クローン展開 55

け

- 計算値、プロパティ 95
- ゲスト、カスタマイズ 79
- 検索基準、Web インターフェイス 46

こ

- 更新、サンプル アプリケーションのスケールイン 200
- 更新情報 11
- 更新展開 174
- 更新プロセス
 - Clustered DotShoppingCart 200
 - Clustered Dukes Bank 200
 - Nanotrader 200
 - キャンセル 218
 - サンプル アプリケーションのスケールイン 200
 - サンプル アプリケーションのスケールアウト 200
 - ロールバック 189
- 更新プロファイル
 - 昇格 188
 - 展開に使用 187
- 更新プロファイルの取り下げ 188
- 構成
 - Amazon EC2 84
 - Linux テンプレート 54
 - vCloud Application Director 86
 - Windows テンプレート 52
- 構成の変更、更新に失敗した場合のトラブルシューティング 191
- 高度なアプリケーション ブループリント 142
- 高度な外部サービス、作成 118

コンテンツ、プロパティ 95
 コンプライアンス サマリ、表示 219
 コンポーネントのライフ サイクル ステージ、アクション 93
 コンポーネント プロパティ
 構成 95
 必須 105

さ

サービス
 コピー 115
 削除 116
 設定不要 109
 定義 111
 編集 115
 ライブラリへの追加 109
 サービス カタログ、アイテムの申請 173
 サービス バージョン
 エクスポート 229
 作成 110
 サービス プロパティ、定義 111
 再起動
 エージェント ブートストラップ 114
 仮想マシン 114
 削除、サービス 116
 作成
 Linux テンプレート 54
 Windows vApp 74
 Windows テンプレート 52
 アクション スクリプト 114
 クラウド トンネル 86
 サンプル、タスク 193
 サンプル アプリケーション
 3 階層アプリケーション 194
 APT Repository Config タスクの追加 204
 Clustered Dukes Bank アプリケーションの展開 197
 DOTShoppingCart の展開 196
 jPetStore 198
 Radiant CMS の展開 199
 RHN 登録タスクの追加 207
 YUM Repository Config タスクの追加 205
 アプリケーションをスケールする更新プロセス 200
 仮想マシン テンプレート 193
 コードの変更 201
 サービス 193
 単一階層アプリケーション 194
 展開 195
 展開済みアプリケーションの更新 200
 ドメインへの参加タスクの追加 203
 ライブラリ コンポーネントの展開 193
 ライブラリ タスク 202
 サンプル アプリケーション、構成の変更 201
 サンプル アプリケーションの展開 196
 サンプル サービス、ライブラリ サービス 208

サンプル テンプレート
 CentOS 193
 使用 193

し

システム要件 24
 事前定義済みサービス、ライブラリへの追加 109
 事前定義済みサンプル コンテンツ、インポート 31, 43
 事前定義済みプロパティ
 IP アドレス 103
 ディスク レイアウト 104
 ノード アレイ インデックス プロパティ 104
 実行プラン 168
 自動バインド
 公開 111, 119
 公開プロパティ 101
 事前構成済みコンポーネント 101
 消費 111, 119
 消費プロパティ 101
 タイプ 101, 111, 119
 タグ 111, 119
 タグ情報 101
 昇格された更新プロファイル
 承諾 189
 破棄 189
 詳細ブループリント、作成 142

す

スクリプト
 Bash 94
 BeanShell 94
 PowerShell 94
 Windows CMD 94
 サポート対象のアクション 94
 サポート対象のカスタム タスク 94
 ストレージ オブション、構成 25
 スナップショット、作成 57

せ

成果物
 アプリケーション ブループリントへのバインド 154
 アプリケーションを使用した展開 155
 作成 153
 説明 149
 成果物管理、説明 149
 成果物リポジトリ インスタンス
 作成 152
 成果物仕様へのマッピング 153
 説明 149
 成果物仕様
 作成 153
 成果物リポジトリ仕様へのマッピング 153
 説明 149

- 成果物リポジトリ、インスタンスの作成 152
- 成果物リポジトリ仕様、説明 149
- セキュアなコンポーネント、プロパティ 104
- セキュリティ設定、ユーザー グループ ベース 16
- 接続の問題、トラブルシューティング 33
- 設定
 - アプリケーション コンポーネント プロパティ 162
 - サービス プロパティ 162
 - ノード プロパティ 162
 - ホスト名 162
- 設定不要、サービス 109

そ

- ソリューション インスタンス
 - Puppet マスターの登録 68
 - 作成 65
- ソリューションの登録
 - 自動 63
 - 手動 63

た

- タグ、ライブラリへの追加 109
- タスクのステータス、表示 210, 211
- [タスクのステータス] ウィンドウ 174
- タスクのタイムライン 174
- 単一階層アプリケーション、サンプル 194
- 単一ノード、バインド 100
- 弾性 IP アドレス 84

ち

- チェック リスト、Application Services の設定 21

つ

- 追加の構成 158

て

- ディスク レイアウト、事前定義済みプロパティ 104
- テナント
 - Application Services での使用 41
 - 事前定義済みサンプル コンテンツのインポート 31
- 展開
 - アプリケーションの更新 179
 - 管理 209
 - 既存の更新プロファイル 187
 - キャンセル 218
 - 更新 211
 - 更新プロセスのロールバック 189
 - 更新プロファイルの昇格 188
 - 構成の変更 185
 - 削除 217
 - サマリの表示 210
 - 実行 ID 210
 - スケール アウト 180
 - スケール イン 182
 - タスクのステータスの表示 210

- ノード クラスタのスケール アウト 215
- ノード クラスタのスケール イン 215
- 変更する更新プロセスの開始 185

展開環境

- Amazon EC2 向けに作成 92
- vCloud Director の作成 81

展開サマリ、表示 211

- [展開サマリ] ページ 174

展開 ステータス、表示 211

展開の失敗

- 実行プランの表示 176, 177
- 失敗した展開の再開 176
- レジューム 177

展開のスケールリング、更新に失敗した場合のトラブルシューティング 191

展開のスケール アウト、更新プロセスの開始 180

展開のスケール イン、更新プロセスの開始 182

展開のロールバック 179

展開プロセス、展開を強制的に停止 218

展開プロファイル

- アプリケーションの展開 168
- インポート 231
- 公開 170
- 設定 157
- 複合展開プロファイルの使用 165

展開プロファイル ウィザード 168

テンプレート

- Amazon EC2 の登録 90
- JRE の要件 88
- OVF フォーマット 26, 80
- vCloud Automation Center 58
- vCloud Director 80
- Windows 用の構成 74
- エクスポート 80
- 仮想マシン 15, 79
- 仮想マシンの登録 90
- サンプル 193

と

登録

- Application Services 用に作成 30
- Puppet マスター 63
- Puppet マスター ソリューション インスタンス 68
- ソリューション インスタンス 63
- ドメインの結合、有効化 76
- ドメインへの参加、事前定義済みタスクの追加 203
- トラブルシューティング
 - 構成の変更の更新 191
 - 接続の問題 33
 - 展開のスケールリングの更新 191

ね

- ネットワーク オプション、構成 25
- ネットワーク帯域幅 86

の

- ノード アレイ インデックス、事前定義済みプロパティ 104
- ノード クラスタ
 - スケールアウト 215
 - スケールイン 215

は

- バインド、プロパティ 100

ひ

- ビジネス グループ
 - Application Services での使用 41
 - vCloud Automation Center 58
- 標準化された構成、プロパティ 16

ふ

- ファイアウォール、ファイアウォール外部の URL のアクセス 34
- ブートストラップ エージェント Linux
 - インストール 77
 - ダウンロード 77
- ブートストラップの要件、カスタム テンプレートのための 88
- 複合展開プロパティ、プロパティのバインド 167
- 複合展開プロファイル
 - アプリケーションの展開 168
 - 公開 172
 - 作成 165
 - 設定 165
 - 展開 168
 - 展開プロファイルの順序付け 166
 - 展開プロファイルの選択 166
- 複数の NIC、定義 144
- ブループリント、vRealize Automation 51
- ブループリントの詳細、表示 210
- フレキシブル ディスク 145
- フレキシブル ディスクレイアウト 158
- プロバイダ仕様
 - 作成 122
 - 編集 123
- プロパティ
 - アレイ 95
 - カスタム 158
 - 計算値 95
 - コンテンツ 95
 - 事前定義 103
 - セキュアなコンポーネント 104
 - 特殊な更新プロパティ 180, 182, 185
 - 内部 161
 - ノード 158
 - 必須 105
 - 文字列 95
 - 予約済み 161

プロパティ値

- 解析 103
- 定義 100

プロパティのバインド 141**プロビジョニングタスク**

- エージェント ブートストラップ 162, 175, 211
- ネットワーク ブートストラップ 162, 175
- ホスト 162, 175, 211

プロファイル

- アプリケーション展開 157, 164
- アプリケーション展開の設定 157

分解

- アプリケーション 214
- 申請 216

ほ

- ポート、要件 24
- ホスト名 139, 162, 175
- ホスト名の設定 139
- ポリシー
 - 削除 132
 - ライブラリへの追加 128
- ポリシー インスタンス
 - 更新 213
 - 作成 62
- ポリシー スキャン、起動 213
- ポリシー スキャン タイムライン、表示 219
- ポリシー定義スクリプト
 - 作成 130
 - 変数 130
- ポリシー バージョン
 - エクスポート 229
 - コピー 131
 - 編集 131
- ポリシー プロファイル、定義 128

ま

- マップ、論理テンプレート 158
- マルチクラウド、サポート 15

め

- メディアの格納場所、プロビジョニング環境 49
- メモリ、ディスク領域とメモリ サイズ 24

も

- 文字列、プロパティ 95

ゆ**ユーザー**

- 管理 221
- 作成 221
- 設定 39
- ユーザー アカウント
 - 設定 39
 - ロック解除 33

- ユーザー インターフェイス
 - アイコン 48
 - ボタン 47
- ユーザー ロール、テナントとビジネス グループの使用 41

よ

- 用語集 16
- 予約、vCloud Automation Center 60
- 予約ポリシー、vCloud Automation Center 60

ら

- ライフ サイクル ステージ、アクション 93
- ライブラリ
 - Application Services 16
 - Puppet サービスのインポート 116
 - オペレーティング システム 108
 - 外部サービスの削除 124
 - 外部サービスの作成 118
 - 外部サービスの追加 117
 - 外部サービス バージョンの保守 123
 - 外部サービス プロパティの定義 119
 - 管理 107
 - 基本の外部サービスの作成 118
 - 高度な外部サービスの作成 118
 - タグの追加 109
 - ポリシーの追加 128
- ライブラリ サービス
 - サンプル コンポーネント 208
 - プロパティの定義 111

り

- 理解
 - 更新プロセス 175
 - 展開プロセス 175
- リポジトリ仕様、作成 150
- リンク クローン
 - スナップショットの作成 57
 - 展開 56, 57
- リンククローン、展開 51

ろ

- ロール、割り当て 39
- ログイン、Default グループ 45
- ロック解除、darwin_user アカウント 33
- ロックされた darwin ユーザー アカウント、ロック解除 33
- 論理テンプレート
 - 仮想マシン 125
 - コピー 127
 - 削除 128
 - 編集 127