使用 Application Services

vRealize Automation 6.2

vmware®

最新的技术文档可以从 VMware 网站下载：

https://docs.vmware.com/cn/

VMware 网站还提供最近的产品更新信息。

您如果对本文档有任何意见或建议，请把反馈信息提交至：

docfeedback@vmware.com

版权所有 © 2012 – 2015 VMware, Inc. 保留所有权利。 版权和商标信息。

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

北京办公室
北京市海淀区科学院南路 2 号
融科资讯中心 C 座南 8 层
www.vmware.com/cn

上海办公室
上海市浦东新区浦东南路 999 号
新梅联合广场 23 楼
www.vmware.com/cn

广州办公室
广州市天河北路 233 号
中信广场 7401 室
www.vmware.com/cn

目录

使用 Application Services 9

更新的信息 11

1 Application Services 简介 13
- Application Services 概览 13
- 核心架构原则 14
  - 部署任何应用程序和中间件服务 15
  - 多云支持 15
  - Application Services 可扩展性和开放式架构 15
  - Application Services 中的标准化 15
  - 用户安全 16
- 主要概念 16

2 安装并配置 Application Services 19

3 安装 Application Services 21
- 准备安装 Application Services 21
  - Application Services 系统要求 22
  - 设置 vCenter Server 群集配置 23
  - 为 vCloud Director 设置 Application Services 24
- 启动 Application Services 设备 26
  - 配置 Application Services 设备 27
  - 注册设备和 vRealize Automation 服务器 27
  - 导入预定义的示例内容 28
  - 打开 Application Services Web 界面 29
- 对连接到 Application Services Web 界面出现的问题进行故障排除 30
- 解锁 darwin_user 帐户 30
- 重新启动 Application Services 31
- 将 Application Services 配置为对外部 URL 使用代理 31
- 将 Application Services 注册到 vRealize Automation 32

4 升级 Application Services 35
- 升级 Application Services 35
- 回滚 Application Services 升级 36

5 设置用户和组 37

6 在 Application Services 中使用租户和业务组 39

7 将预定义的内容导入租户 41

8 使用 Application Services Web 界面 43
登录 Application Services 43
使用 Application Services Web 界面 44
使用 Application Services 按钮 44
了解 Application Services 图标 45

9 为 Application Services 环境设置应用程序置备 47
创建 vRealize Automation 自定义模板的虚拟机要求 48
在 vRealize Automation 中创建虚拟机模板 49
准备 vCenter Server Windows 虚拟机模板 49
准备 vCenter Server Linux 虚拟机模板 51
创建克隆部署的 vRealize Automation 蓝图 52
创建并配置链接克隆部署的 vRealize Automation 蓝图 53
更新 vRealize Automation 中现有的虚拟机模板 54
注册 vRealize Automation 云提供程序和模板 55
创建 vRealize Automation 部署环境 56
映射外部服务实例 57
创建策略实例 58
注册 Puppet 解决方案实例 59

10 为 vCloud Director 环境设置应用程序置备 67
创建 vCloud Director 自定义模板的虚拟机要求 68
在 vCloud Director 中创建 Windows 虚拟机模板 69
创建 vCloud Director vApp 69
在 vCloud Director 中配置 Windows 虚拟机模板 70
为 Windows 虚拟机模板启用 SID 更改和域加入 71
将 Windows 虚拟机模板添加到 vCloud Director 目录 72
在 vCloud Director 中创建 Linux 虚拟机模板 73
验证 vCloud Director 目录中的云模板配置 74
更新 vCloud Director 中的现有虚拟机模板 75
以 OVF 格式导出虚拟机模板 75
注册 vCloud Director 云提供程序和模板 75
创建 vCloud Director 部署环境 76

11 为 Amazon EC2 环境设置应用程序置备 79
为 Application Services 配置 Amazon EC2 环境 80
创建端点虚拟机 81
创建云通道以连接到 Amazon EC2 82
创建 Amazon EC2 自定义模板的虚拟机要求 83
创建 Amazon EC2 虚拟机模板或 AMI 83
注册 Amazon EC2 云提供程序和模板 85
创建 Amazon EC2 部署环境 86

12 开发 Application Services 组件 89
定义组件操作 89
支持的操作和自定义任务脚本 90
配置组件属性 91
属性的类型 91
定义属性值 95
绑定到其他属性 95
自动绑定到其他属性 96
在生命周期阶段和组件之间传递属性值 97
Application Services 预定义属性 97
预定义 IP 地址属性 98
预定义的节点数组索引属性 98
预定义的磁盘布局信息属性 98
安全组件属性 98
所需的组件属性 99
开发组件的最佳做法 99

13 管理 Application Services 库 101
将操作系统添加到库 101
将标记添加到库 102
将服务添加到库 103
在库中创建服务版本 103
定义服务版本属性 104
将操作脚本添加到服务版本中 106
维护服务版本 107
从库中删除服务 108
将 Puppet 服务导入库 108
将外部服务添加到库 109
在库中创建外部服务 110
定义外部服务属性 111
为高级外部服务版本创建提供程序规范 113
维护外部服务版本 114
删除外部服务 115
将逻辑模板添加到库 115
维护逻辑模板版本 117
从库中删除逻辑模板 118
将策略添加到库 118
创建策略定义脚本 120
维护策略版本 121
删除策略 122
将自定义任务添加到库 122
维护自定义任务版本 123

14 创建应用程序 125
创建应用程序 126

创建应用程序版本 126
应用程序蓝图建模 127
配置应用程序蓝图 128
创建高级蓝图 129
创建组件之间的依赖关系 130
将节点指定为群集 130
为节点定义多个网卡 131
添加和管理可变磁盘布局 131
将外部服务添加到高级蓝图 132
复制应用程序版本 133
复制应用程序 133
删除应用程序版本 134

15 使用项目 135
创建项目存储库规范 135
创建项目存储库实例 137
创建项目规范 138
将项目规范映射到项目存储库实例 138
将项目绑定到应用程序蓝图 139
使用项目部署应用程序 140

16 部署应用程序 141
设置和配置部署配置文件 141
创建部署配置文件 141
配置部署环境 142
配置应用程序属性 145
检查执行计划并添加自定义任务 146
使用现有部署配置文件 147
创建复合部署配置文件 148
使用部署配置文件部署 151
使用单个部署配置文件进行部署 151
使用复合部署配置文件进行部署 151
快速部署应用程序 152
将部署配置文件发布到 vCloud Automation Center 服务目录 153
发布单个部署配置文件 153
发布复合部署配置文件 154
请求 vRealize Automation 服务目录项 155
使用部署摘要页面 156
使用复合部署摘要页面 156
了解部署和更新过程 157
了解部署故障 158
解决部署失败问题 158

17 更新应用程序部署 161
启动更新过程以扩大部署 162
启动更新过程以缩小部署 164

启动更新过程以修改配置 166
使用现有的更新配置文件 168
升级更新配置文件 168
接受或放弃升级的更新配置文件 169
回滚更新过程 170
了解运行自定义任务更新 171
对失败的扩展部署更新过程进行故障排除 171
对失败的修改配置更新过程进行故障排除 172

18 部署预定义库组件 173
使用示例模板 173
将 libpython 软件包添加到 Ubuntu 模板 174
为 Ubuntu 虚拟机启用 SSH 连接 174
可用于部署的示例应用程序 174
部署 Nanotrader 应用程序 174
部署群集化的 DotShoppingCart 应用程序 175
部署群集化的 Dukes Bank 应用程序 176
部署 jPetStore 应用程序 177
部署 Radiant CMS 应用程序 178
更新已部署的示例应用程序 179
启动更新过程以扩展示例应用程序 179
启动更新过程以修改示例应用程序中的配置 180
示例 Application Services 库任务 181
添加加入域预定义任务 182
添加 APT 存储库配置预定义任务 183
添加 YUM 存储库配置预定义任务 184
添加 RHN 注册预定义任务 185
示例库服务 186

19 管理部署 187
查看应用程序的部署任务和蓝图详细信息 187
查看应用程序的已部署虚拟机的详细信息和执行计划 189
启动策略扫描 190
从云中卸载应用程序 191
从 vCloud Automation Center 缩小部署 192
从 vCloud Automation Center 扩大部署 192
从 vCloud Automation Center 卸载应用程序 193
从 Application Services 中删除应用程序部署 194
取消部署或更新过程 195
查看策略合规性摘要 196

20 使用 Application Services CLI 197
远程启动 CLI 197
常规 CLI 选项 198
管理云通道 199
使用 CLI 部署和更新应用程序 200

使用 CLI 可卸载部署 202

21 使用 CLI 导入和导出功能 205

使用 CLI 导出命令 205

使用 CLI 导入命令 207

索引 209

使用 Application Services

vRealize Automation Application Services（以前称为 vCloud Application Director）可在云中自动置备应用程序，包括在基础架构云上部署、配置和更新应用程序的组件以及从属中间件平台服务。Application Services 功能简化了基于 vCloud Director、vSphere 和 Amazon Elastic Compute Cloud (Amazon EC2) 的基础架构云上的自定义应用程序和打包应用程序的复杂部署。

本文档介绍了如何使用 Application Services 在跨虚拟和基于云的基础架构中创建、部署、管理和更新应用程序。

目标受众

这些信息主要面向任何需要使用 Application Services 进行应用程序部署的人员。适用读者包括与应用程序基础架构管理员和云管理员进行协作的应用程序架构师和应用程序部署人员。

更新的信息

本《*使用 Application Services*》指南随每一版本的产品更新或在必要时进行更新。

下表提供了《*使用 Application Services*》指南的更新历史记录。

修订版本	描述
001654-02	■ 移除了版本值，并在第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”、第 51 页，“准备 vCenter Server Linux 虚拟机模板”、第 22 页，“Application Services 系统要求”和第 23 页，“设置 vCenter Server 群集配置”中添加了对 vRealize Automation 文档中的支持列表（网址为 https://www.vmware.com/support/pubs/vcac-pubs.html）的引用。 更新了第 35 页，“升级 Application Services”。
001654-01	■ 更新了第 54 页，“从 vRealize Automation 蓝图创建快照”和第 53 页，“创建并配置链接克隆部署的 vRealize Automation 蓝图”以更好地介绍示例步骤。 ■ 更新了第 49 页，“在 vRealize Automation 中创建虚拟机模板”。 ■ 更新了第 35 页，“升级 Application Services”以说明升级后必须重新启动 vRealize Automation。 ■ 更新了 第 85 页，“注册 Amazon EC2 云提供程序和模板” 主题，更好地列出支持的 Amazon Region 代码值。 ■ 更新了第 189 页，“查看应用程序的已部署虚拟机的详细信息和执行计划”主题以包含 VMware Remote Console 的相关信息。
001654-00	初始 6.2 版本。

Application Services 简介 1

VMware vRealize Automation Application Services 以前称为 VMware vCloud Application Director，是一个基于模型的应用程序置备解决方案，可简化在多个基础架构云上创建和标准化应用程序部署拓扑的过程。应用程序架构师可以使用具有拖放界面的基于图形的画布来对名为应用程序蓝图的应用程序部署拓扑进行建模。

应用程序蓝图可定义应用程序的结构、支持标准化应用程序基础架构组件的使用，以及包括自定义和打包的企业应用程序的安装依赖关系和默认配置。应用程序蓝图是可在基于 VMware 的 IaaS 云（如 vCloud Director、vRealize Automation）之间移植和可在公共云（如 VMware vCloud Air 与 Amazon EC2）之间移植的逻辑部署拓扑。

本章讨论了以下主题：

- 第 13 页，“Application Services 概览”
- 第 14 页，“核心架构原则”
- 第 16 页，“主要概念”

Application Services 概览

VMware vRealize Automation Application Services 具有模型驱动的、可扩展的开放架构。通过其标准组件的库或服务，Application Services 可实现自动化并为混合云环境中的多层企业应用程序管理部署更新生命周期。

企业用户可以在动态云环境中标准化、部署、配置、更新和扩展复杂的应用程序。这些应用程序的范围可以从简单的 Web 应用程序到复杂的自定义应用程序和打包应用程序。

Application Services 使用 vRealize Automation 进行用户和组支持、访问控制和目录管理。vRealize Automation 实例可以具有多个租户，一个租户可以包含多个业务组，并且一个业务组可以包含多个用户和对象（如应用程序和服务）。用户必须是业务组的成员，才能添加或编辑该业务组中的对象。要查看业务组中的专用对象，用户必须属于该组。要查看共享对象，用户可以属于同一租户中的任一业务组。在 Application Services 中部署的应用程序成为 vRealize Automation 中的目录项，用户可以为置备请求这些目录项。

要在支持的云环境中自动部署应用程序，用户必须在云抽象层 (CAL) 中配置组件。云模板包含预定义的可重复使用计算机映像，该映像包括操作系统和在创建映像时应用于虚拟机的数据。云模板将映射至 Application Services 库中的逻辑模板。云提供程序为部署提供云实例。部署环境为云提供程序实例中的部署提供特定的环境。云提供程序和部署环境都会将组件从云环境映射至 Application Services。

应用程序架构师可使用拖放界面创建可视化应用程序蓝图。应用程序架构师可使用标准逻辑模板、应用程序基础架构服务、组件和脚本的预填充和可扩展库来为应用程序蓝图建模。这些蓝图可让重复部署的应用程序的结构标准化，包括软件组件、依赖关系和配置。

应用程序蓝图在部署环境中具有可移植性。例如，蓝图可用之后，应用程序开发、QA 和发行团队便可在 IT 所设置的标准下工作。这些团队可重复部署标准蓝图、自定义允许的配置并在 IT 许可的部署环境中进行部署。

通过应用程序蓝图，您可以使用部署配置文件创建不同的部署来测试原型，也可以在生产环境中部署任务关键型多层应用程序。从这些保存的蓝图中，应用程序部署者可生成执行计划以便将应用程序部署到私有云或公共云。您也可以启动更新过程来扩展已部署应用程序的群集节点，并在已部署应用程序的新版本可用时更改其配置或代码。

图 1-1 Application Services 和 vRealize Automation 工作流

核心架构原则

Application Services 旨在跨任何 IaaS 云自动部署复杂应用程序。Application Services 可以使用 vCloud Director 和 vRealize Automation，在 IaaS 云环境中自动进行部署，以便部署到 vSphere 和 Amazon EC2。

- 部署任何应用程序和中间件服务第 15 页，

  通过 Application Services，应用程序架构师可以使用虚拟机模板和脚本对中间件服务和应用程序的应用程序部署进行建模。

- 多云支持第 15 页，

  Application Services 旨在将相同的应用程序部署到多种类型的云。

- Application Services 可扩展性和开放式架构第 15 页，

  Application Services 已针对 vCloud 组件进行优化并可扩展到其他组件。

- Application Services 中的标准化第 15 页，

  借助 Application Services，您可以使用标准化的配置属性创建可重用的服务以满足 IT 合规性的严格要求。

- 用户安全第 16 页，

  用户必须至少已分配有一个应用程序用户角色，才能使用已注册租户的 SSO 凭据登录到 Application Services。

部署任何应用程序和中间件服务

通过 Application Services，应用程序架构师可以使用虚拟机模板和脚本对中间件服务和应用程序的应用程序部署进行建模。

使用 Application Services，可以在 vCloud Director 和 vRealize Automation 云环境的 Windows 和 Linux 操作系统中部署应用程序。对于 Amazon EC2 云环境，您只能在 Linux 操作系统中部署应用程序。

- 应用程序架构师可以使用为其企业定义的标准化云提供程序库中的虚拟机模板。
- 应用程序架构师还可以向应用程序蓝图中定义的中间件服务添加应用程序组件。

多云支持

Application Services 旨在将相同的应用程序部署到多种类型的云。

- Application Services 将部署设置封装在部署配置文件中，以便与应用程序蓝图分隔开来。部署配置文件使应用程序蓝图可在基于 vCloud Director 的私有云以及基于 Amazon EC2 的公共云之间移植，并使用 vRealize Automation 置备基础架构访问 vSphere 私有云和公共云。
- Application Services 使用 CAL 连接云提供程序。通过 vRealize Automation 和 Amazon EC2 云环境可实现到 vCloud Director 和 vSphere 的自动部署。vCloud Director 使用 vCloud API 和开放式虚拟化格式 (OVF) 等开放式标准。Application Services 通过可公开获得的基于 vCloud 的 IaaS API 与 vCloud Director 集成。此集成允许您在 vCloud Director 云的专用部署和公共部署中部署应用程序。

  vCloud Automation Center 使用 REST API 将应用程序部署到 vSphere 私有云和公共云。

  Application Services 使用 Amazon AWS API 和 Amazon 计算机映像 (AMI) 将应用程序部署到 Amazon EC2 云。

Application Services 可扩展性和开放式架构

Application Services 已针对 vCloud 组件进行优化并可扩展到其他组件。

- Application Services 库具有预定义（现成）的服务或适用于中间件服务的应用程序。此外，可向云库中的虚拟机模板添加指针。通过 Application Services 库还可使用安装、配置、启动、更新、回滚和卸载脚本在虚拟机模板上添加动态可安装自定义服务的定义，以及相应的配置。
- 部署执行计划由系统基于蓝图生成。这些计划有助于用户在部署过程中跟踪任务的状态和进程。

  可将自定义任务脚本添加到这些计划，以在部署的虚拟机中运行。可以使用这些脚本执行其他任务，如使用第三方内部 IT 系统运行安全修补程序、审核集成、质量与合规性检查，以及运行冒烟测试。
- 要简化部署过程的构建，各组织可使用命令行界面进一步使部署自动化，以允许其连续的版本系统或云置备门户生成并部署应用程序。

Application Services 中的标准化

借助 Application Services，您可以使用标准化的配置属性创建可重用的服务以满足 IT 合规性的严格要求。

Application Services 包含以下标准化的配置属性：

- 模型驱动架构，允许在应用程序蓝图中添加 IT 认证的虚拟机模板和中间件服务。
- 一个委派模型，用于在应用程序目录管理员、应用程序架构师和部署者之间替代配置名称值对，以便对应用程序和中间件服务的配置值实现标准化。

用户安全

用户必须至少已分配有一个应用程序用户角色，才能使用已注册租户的 SSO 凭据登录到 Application Services。

Application Services 通过 vRealize Automation 支持身份验证机制。在 vRealize Automation 中，为向 Application Services 注册的租户配置的身份存储 (IDP) 中的所有用户都可以使用 SSO 凭据登录（如果他们至少已分配有一个应用程序用户角色）。有关如何分配用户角色的信息，请参见 vRealize Automation 的《租户管理》。

主要概念

要部署应用程序，必须配置 CAL。对蓝图进行建模时，可以使用直接从云提供程序库中获取的虚拟机模板，以及立即可用应用程序基础架构组件和脚本的可扩展库。对应用程序部署拓扑进行建模后，可以创建依赖关系并编辑配置以完成执行计划。部署应用程序后，您也可以启动更新过程扩展群集节点，或者在新版本可用时更改已部署应用程序的配置或代码。

以下定义有助于您了解置备和更新工作流。

CAL 的主要组件。

云模板

云模板包含预定义的可重复使用计算机映像，该映像包括操作系统和数据。在云环境中创建虚拟机时，云模板中定义的规范和软件将应用于该虚拟机。

表 1-1 映射到支持的云环境的云模板

Application Services 环境	云环境
云模板	Application Services 模板
	vRealize Automation 蓝图
	Amazon 计算机映像 (AMI)

云提供程序

用于部署的云实例。可以为一个云提供程序类型定义多个云提供程序。

表 1-2 映射到支持的云环境的云提供程序

Application Services 环境	云环境
云提供程序	vCloud Director 组织
	vRealize Automation 6.1 业务组
	Amazon 区域

云提供程序类型

可进行部署的云基础架构的类型。Application Services 仅支持 vCloud Director、vRealize Automation 和 Amazon EC2。

部署环境

云提供程序中的环境，例如开发、测试、转储和生产环境。云提供程序可以具有多个部署环境。

表 1-3 映射到支持的云环境的部署环境

Application Services 环境	云环境
部署环境	vCloud Director 组织 vDC
	vRealize Automation 预留策略
	Amazon VPC

对于 vCloud Director，部署环境为定义的云提供程序映射到组织虚拟数据中心 (vDC)，并使用该 vDC 的资源。组织 vDC 为组织提供资源，并从提供者 vDC 进行分区。组织 vDC 提供可以存储、部署和操作虚拟系统的环境。组织 vDC 还为诸如软盘和 CD ROM 等虚拟介质提供存储。单个组织可以具有多个组织 vDC。

对于 vRealize Automation，部署环境映射到预留策略。如果未选择部署环境，则 vRealize Automation 会根据 vRealize Automation 蓝图中虚拟机的资源要求分配预留策略。

对于 Amazon EC2，部署环境映射到 Amazon 虚拟私有云 (VPC) 与区域的其中一个可用性区域的组合。

主要库组件。

库
包含逻辑模板（云模板的指针）的库。可以在多个应用程序中使用、在虚拟机上安装的可重复使用服务。可以在应用程序部署中执行额外的自定义任务的任务。

服务
可以在虚拟机上安装、在多个应用程序中重复使用的脚本式软件。

外部服务
在应用程序部署外部安装的应用程序服务，例如硬件负载平衡器或预安装数据库。应用程序和外部服务必须配置为相互协作。

逻辑模板
Application Services 中的预定义虚拟机定义。在云库和支持的服务中，逻辑模板可映射到实际云模板。逻辑模板允许应用程序蓝图保持云的不可知状态。

策略
一组用户定义的、用于控制应用程序生命周期操作的定义。例如，某个策略可以根据公司准则将特定软件的使用列入黑名单。通过使用策略实例，在特定部署环境中启用策略时，这些策略会强制执行。

自定义任务
在执行计划中，您可以添加自定义任务以执行额外的自定义任务，例如在应用程序部署中运行安全修补程序。可以在库中创建自定义任务，并将其添加到应用程序部署。Application Services 还在库中提供预定义任务，这些任务可用于配置 APT 存储库、YUM 存储库，在 Red Hat 网络中注册虚拟机或加入域。

操作系统
指定逻辑模板和服务的 IT 组织支持的操作系统。**操作系统**菜单中将显示操作系统列表，可以将操作系统添加到该列表。

标记
在蓝图编辑器中对逻辑模板和服务的列表进行组织，以增强可读性。**标记**菜单中将显示标记列表，可以将新标记添加到该列表。

应用程序置备的主要组件。

应用程序
逻辑部署单元，定义操作系统模板、应用程序组件及其分布在多个虚拟机间的从属服务之间的关系。

应用程序组件
用作组件（例如 EAR 文件、WAR 文件等）模板的自定义代码。这些代码是自定义脚本包，用于对某个节点或服务执行安装、配置、启动、更新、回滚和卸载操作。

应用程序蓝图
用于部署的应用程序的逻辑拓扑。蓝图会捕获具有逻辑节点的应用程序的结构、这些节点的相应服务和操作系统、依赖关系、默认配置以及网络和存储的拓扑要求。

node
在蓝图中定义的虚拟机。

群集节点
在蓝图中定义的虚拟机群集。

磁盘
要添加到相应虚拟机或节点的其他磁盘。

属性	为服务和应用程序组件配置名称-值对。脚本可使用这些变量在脚本中设置参数并运行各种配置。例如，可以设置 installation_path 属性值并配置安装脚本，从而在应用程序部署过程中使用此属性来指定安装服务时要使用的路径。
操作	要安装的服务和应用程序组件的安装、配置、启动、更新、回滚和卸载等脚本的生命周期阶段
存储器	云提供程序提供的空间，以将磁盘置于不同的存储中来满足性能要求。
部署配置文件	蓝图的部署设置集合，包括群集大小、CPU、内存、云模板和网络。
逻辑网络	作为网络的逻辑抽象而创建的实体。逻辑网络是用于指示应用程序节点之间的网络位置的云不可知抽象。逻辑网络在配置部署配置文件的过程中动态映射到云网络。
云网络	在云部署环境内定义的网络。
执行计划	任务计划，用于查看创建虚拟机的顺序以及安装、配置、启动和更新库和应用程序组件的操作脚本的顺序。部署执行计划中定义了代理执行任务的顺序。
卸载	从云中移除已部署的虚拟机。例如，您可以从云环境中卸载 vCloud Director vApp 和关联的虚拟机，vRealize Automation 中的 vCenter Server 虚拟机和关联的虚拟机，或已部署应用程序的 Amazon EC2 实例。卸载过程完成后，所有其他资源（例如这些虚拟机使用的存储和 IP 地址）将返回到其各自的池。

安装并配置 Application Services 2

Application Services 包括预定义示例应用程序、服务和虚拟机模板，帮助您了解基本概念并开始使用该产品。通过 Application Services 或 Amazon EC2 环境，完成安装 vCloud Director、配置预定义示例应用程序并将其部署到 vSphere 和 vRealize Automation 的任务。

前提条件

熟悉第 13 页，“Application Services 概览”中所述的 Application Services 置备工作流。

步骤

1 安装和设置 Application Services 虚拟设备。

请参见第 21 页，“准备安装 Application Services”和第 26 页，“启动 Application Services 设备”。

2 配置 Application Services 以使用代理。

当应用程序需要从企业防火墙外部下载文件时，请执行此任务。

请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。

3 登录到 Application Services Web 界面并熟悉产品功能。

请参见第 43 页，“登录 Application Services”和第 44 页，“使用 Application Services Web 界面”。

4 注册云提供程序。

注册云提供程序时，将 Application Services 映射到云环境中的以下组件之一。

云提供程序类型	参考主题
vCloud Director 实例和组织	第 75 页，“注册 vCloud Director 云提供程序和模板”
vRealize Automation 业务组	第 55 页，“注册 vRealize Automation 云提供程序和模板”
Amazon EC2 区域	第 85 页，“注册 Amazon EC2 云提供程序和模板”

5 在注册的云提供程序中创建部署环境。

在部署应用程序之前，将 Application Services 部署环境映射到云环境中的以下组件之一。

部署环境	参考主题
vCloud Director 中的组织 vDC	第 76 页，“创建 vCloud Director 部署环境”
vCloud Automation Center 中的预留策略	第 56 页，“创建 vRealize Automation 部署环境”
Amazon VPC 和关联的可用性区域	第 86 页，“创建 Amazon EC2 部署环境”

6 将云模板映射到逻辑模板。

将云模板映射到逻辑模板时，会将 Application Services 逻辑模板链接到云中的实际模板。

请参见第 115 页，“将逻辑模板添加到库”。

7 从部署配置文件向导中部署预定义示例应用程序。

请参见第 173 页，第 18 章“部署预定义库组件”和第 141 页，“创建部署配置文件”。

8 检查部署状态。

部署期间，将根据应用程序的依赖关系安装和配置组件。请参见第 157 页，“了解部署和更新过程”。

您可以使用用户界面来实时检查应用程序部署的状态。请参见第 156 页，“使用部署摘要页面”。

9 对部署失败进行故障排除。

如果遇到部署失败，可以检查特定于虚拟机的日志及部署日志，并对问题进行故障排除。

要访问特定于虚拟机的日志，请参见第 189 页，“查看应用程序的已部署虚拟机的详细信息和执行计划”。要解决该问题，请参见《Application Services 故障排除》文档。

安装 Application Services 3

要安装 Application Services 6.1，您可以在 vCloud Director 或 vSphere vCenter Server 中创建和部署虚拟设备。

本章讨论了以下主题：

- 第 21 页，“准备安装 Application Services”
- 第 26 页，“启动 Application Services 设备”
- 第 30 页，“对连接到 Application Services Web 界面出现的问题进行故障排除”
- 第 30 页，“解锁 darwin_user 帐户”
- 第 31 页，“重新启动 Application Services”
- 第 31 页，“将 Application Services 配置为对外部 URL 使用代理”
- 第 32 页，“将 Application Services 注册到 vRealize Automation”

准备安装 Application Services

在开始安装 Application Services 设备前，验证您的计算环境是否满足软硬件系统要求。

Application Services 要求 vCloud Director 使用特定的配置设置。对于之前安装的 vCenter Server 和 vCloud Director 服务器配置，验证这些服务器所使用的设置是否适用于 Application Services。

- Application Services 系统要求第 22 页，

  运行 Application Services 的虚拟设备必须满足特定的硬件和软件要求。此外，还必须为 Application Services 打开某些端口。

- 设置 vCenter Server 群集配置第 23 页，

  您必须设置 vCenter Server 群集配置的存储和网络以满足特定要求，以便可以在 Application Services 或 vCloud Director 中安装 vSphere。

- 为 vCloud Director 设置 Application Services 第 24 页，

  您必须在 vCloud Director 中配置虚拟设备，才能成功安装 Application Services。如果您有以前安装的 vCloud Director 版本，则必须检查配置设置，针对 Application Services 进行优化。

Application Services 系统要求

运行 Application Services 的虚拟设备必须满足特定的硬件和软件要求。此外，还必须为 Application Services 打开某些端口。

vCloud Director 磁盘空间和内存要求

Application Services 至少需要 20GB 的磁盘空间、2GB 内存和 1 个速度不低于 2GHz 的 vCPU。

小心 对于某些组织 vDC，vCloud Director 有时会基于 vCloud Director 设置默认将虚拟 CPU 设置为 0.24GHz。如果此设置是您环境中的默认设置，则必须为其中部署了 Application Services vApp 的组织 vDC 将 vCPU 速度设置为 2GHz。否则，Application Services 虚拟设备的性能将受到影响。

端口要求

您必须为 Application Services 虚拟机打开某些端口。

端口	连接类型
TCP 端口 8443	用于通过 HTTPS 连接 Application Services 用户界面的外部端口。
TCP 端口 8080	用于通过 HTTP 连接 Application Services 用户界面的可选端口。
TCP 端口 80	用于 `vami-lighttpd` 的外部端口。
TCP 端口 5671	用于 vFabric RabbitMQ 的外部端口。
TCP 端口 443	用于将 Application Services 用户界面连接到云环境的外部端口。
TCP 端口 22	用于外部 SSH 连接的可选端口。

有关您版本的最新端口要求的信息，请参见 vRealize Automation 文档中的支持列表，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。

Web 界面支持

Application Services 支持以下 Web 浏览器。有关您版本的最新版本要求的信息，请参见 vRealize Automation 文档中的支持列表，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。

- Internet Explorer
- Mozilla Firefox
- Chrome

虚拟化软件要求

要使用 Application Services，必须安装和设置以下 VMware 产品：

- vSphere ESXi。请参见 vSphere 文档，网址为：https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html。
- vCenter Server。请参见 vSphere 文档，网址为：https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html。
- vCloud Director 5.5。请参见 vCloud Director 文档，网址为：https://www.vmware.com/support/pubs/vcd_pubs.html。

- vRealize Automation 6.2。请参见 vRealize Automation 文档，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。

有关您版本的最新系统要求的信息，请参见 vRealize Automation 文档中的支持列表，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。

支持 Application Services 库中的虚拟机模板的操作系统

要创建在 Application Services 库中使用的自定义虚拟机模板，请验证是否在以下产品中安装了支持的操作系统：

- vCloud Director。请参见第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”。
- vRealize Automation。请参见第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”。
- Amazon EC2。请参见第 83 页，“创建 Amazon EC2 自定义模板的虚拟机要求”。

注意 不支持启用了联邦信息处理标准 (FIPS) 的 Linux 和 Windows 虚拟机模板。

设置 vCenter Server 群集配置

您必须设置 vCenter Server 群集配置的存储和网络以满足特定要求，以便可以在 Application Services 或 vCloud Director 中安装 vSphere。

前提条件

- 有关您版本的最新版本要求的信息，请参见 vRealize Automation 文档中的支持列表，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。
- 熟悉创建 vSphere 资源池和已启用 DRS 的 vCenter Server 群集的过程。请参见 vSphere 文档中心。
- 确认一个或多个主机正在运行 vSphere ESXi。
- 确认已具有启用了 DRS 的 vCenter Server 群集。
- 确认 vCenter Server 安装中的每台 vSphere ESXi 主机上都有 NTP 客户端在运行。

  同步部署 Application Services 的主机和要部署虚拟机的主机上的时间。如果时间不同步，则 Application Services 服务器在与置备的虚拟机进行通信时可能会出现问题。

  要同步主机上的时间，请针对 `Configuring Time` 系统下的每台 ESXi 主机，在其 vCloud Director 选项中设置一个 NTP 服务器。

步骤

1 创建一个启用 DRS 的群集。

2 检查配置设置是否满足数据存储和网络需求。

选项	操作
群集中的多台主机	■ 验证所有主机是否都至少具有一个共享数据存储，例如，共享的 LUN、NFS 等。 ■ 验证群集内的所有主机是否均已连接到至少一个公共网络。 ■ 要支持实时虚拟机迁移，请验证群集内的所有主机是否都相同。
网络或 SAN 存储器	■ 验证您的主机是否具有共享存储。 vCloud Director 和 Application Services 虽然可以使用本地存储，但使用共享存储可确保将来的可扩展性。 ■ 验证主机是否至少具有一个网络。

下一步

对于 vSphere，请部署 Application Services 设备。请参见第 24 页，“在 vSphere 中部署 Application Services 设备”。

对于 vCloud Director，请配置 vApp 设置以安装 Application Services。请参见第 24 页，“为 vCloud Director 设置 Application Services”。

在 vSphere 中部署 Application Services 设备

要在 vSphere 中安装 Application Services，必须部署一个 OVF 模板。

可以从 vSphere Client 计算机能够访问的任何本地文件系统或从远程 Web 服务器部署 OVF 模板。本地文件系统可以包括本地磁盘、可移动介质以及共享网络驱动器。

前提条件

- 确认已完成第 21 页，“准备安装 Application Services”中列出的针对 vSphere 的所有必备条件。
- 必须有一个 Application Services OVF 模板可用于部署。熟悉在 vSphere Client 中部署 OVF 模板的过程。请参见 vSphere 文档。

步骤

1 登录到 vSphere Client。

2 选择**文件 > 部署 OVF 模板**。

此时将打开部署 OVF 模板向导。

3 配置“群集”、“资源池”和“存储”设置。

4 将模板映射到 IP 地址池中的网络。

5 选中**部署后打开电源**复选框。

6 检查部署设置，然后单击**完成**。

Application Services 虚拟机将进行部署并在部署后打开电源。

下一步

初始化 Application Services 设备。请参见第 26 页，“启动 Application Services 设备”。

为 vCloud Director 设置 Application Services

您必须在 vCloud Director 中配置虚拟设备，才能成功安装 Application Services。如果您有以前安装的 vCloud Director 版本，则必须检查配置设置，针对 Application Services 进行优化。

前提条件

- 针对 Application Services 配置 vCenter Server 设置。请参见第 23 页，“设置 vCenter Server 群集配置”。
- 熟悉创建组织和目录的过程。请参见最新的 vCloud Director 文档。
- 验证 vCloud Director 管理员是否已将模板上载至目录。

  注意 如果模板上载到其他 vCloud 组织（与 Application Services 执行应用程序部署不同的组织）下的目录，则必须发布该目录和相应模板。验证该目录是否具有正确属性，以便用户可以访问。

- 验证您是否能直接连网到 Application Services 可使用的外部 IP 地址池。请联系您的 vCloud Director 管理员，以确定池中 IP 地址的数量。

步骤

◆ 在 vCloud Director 中，为 Application Services 分配一个单独组织。

下一步

验证您的 vCloud Director 设置。请参见第 25 页，“验证 vCloud Director 环境”。

要创建或更新自定义虚拟机模板，请参见第 69 页，“在 vCloud Director 中创建 Windows 虚拟机模板”和第 73 页，“在 vCloud Director 中创建 Linux 虚拟机模板”。

验证 vCloud Director 环境

如果安装有 vCloud Director 的早期版本，请确认 vApp 可以与虚拟机和外部网络 IP 地址进行通信。

Application Services 支持将虚拟机部署到直接连接的网络和 NAT 路由的网络。为已部署的虚拟机使用 IPPOOL 寻址，以便通过网络与 Application Services 服务器进行通信。

注意 不支持为直接连接的网络使用 DHCP 寻址。

前提条件

- 针对 Application Services 配置 vCenter Server 设置。请参见第 23 页，“设置 vCenter Server 群集配置”。
- 熟悉自定义 vCloud Director 的过程。请参见最新的 vCloud Director 文档以及 VMware 知识库文章 kb.vmware.com/kb/2005829 和 kb.vmware.com/kb/2034092。
- 验证是否安装了与浏览器一起使用的 vCloud Director VMRC 插件。有关兼容的 Web 浏览器的信息，请参见第 22 页，“Application Services 系统要求”。
- 验证是否为 Application Services 虚拟机打开了所需的 TCP 端口。请参见第 22 页，“Application Services 系统要求”。

步骤

1 在 vCloud Director 用户界面中，创建具有一个虚拟机的 vApp。

2 在映射到 Application Services 云提供程序的 vCloud 组织中，验证 vCloud 组织中的虚拟机具有允许虚拟机连接到 Application Services 设备的 vCloud 网络配置。

3 部署 vApp。

4 使用 VMRC 插件连接到虚拟机并验证其是否可以 Ping 外部 IP。

5 打开 TCP 端口，以便部署的虚拟机可以访问 Application Services 设备。

下一步

下载并部署 Application Services 设备。请参见第 25 页，“在 vCloud Director 中部署 Application Services 设备”。

在 vCloud Director 中部署 Application Services 设备

要在 Application Services 中安装 vCloud Director，必须下载该设备。

有关添加 vApp 的信息，请参见 vCloud Director 文档。

前提条件

确认已完成第 21 页，“准备安装 Application Services”中列出的针对 vCloud Director 的必备条件。

步骤

1 下载 Application Services 设备，并将两个文件保存在同一文件夹中而不更改文件名。

该设备包括以下文件：

`ApplicationDirector-版本号-*_OVF10.ovf`

`ApplicationDirector-版本号-*-system.vmdk`

2 使用该设备的内部版本号替换*版本号*。

3 登录到 vCloud Director 并选择要在其中部署 Application Services 的组织 vDC。

4 选择“目录”视图并单击 **vApp 模板**选项卡。

5 单击**上载** ( )。

6 在将 OVF 作为模板上载窗口中，填写请求的信息。

7 在任何证书警告弹出窗口中单击**确定**以继续上载该设备。

由于文件过大且受网络连接速度限制，上载过程预计需要很长时间。

8 右键单击已上载的模板，然后选择**添加到我的云**。

9 根据提示添加 vApp。

10 导航到**我的云**。

11 右键单击新添加的 vApp，然后单击**启动**。

下一步

初始化 Application Services 设备。请参见第 26 页，“启动 Application Services 设备”。

启动 Application Services 设备

您必须启动新添加的 vCloud Director vApp 或 vSphere 虚拟机以确认 Application Services 安装成功，并完成剩余的设置步骤。

步骤

1 配置 Application Services 设备第 27 页，

首次启动 Application Services 设备时，必须输入 Application Services 序列号并为 root 用户和 darwin_user 创建密码。

2 注册设备和 vRealize Automation 服务器第 27 页，

向 vRealize Automation 服务器注册 Application Services 设备。通过注册，您可以在 Application Services 中使用 vRealize Automation 用户、业务组和租户，并分配 Application Services 用户角色。

3 导入预定义的示例内容第 28 页，

注册 Application Services 设备和 vRealize Automation 服务器后，您可选择将预定义的示例内容安装到 vRealize Automation 租户中。

4 打开 Application Services Web 界面第 29 页，

对设备进行配置、将其注册到 vRealize Automation 服务器并（可选）导入预定义示例内容后，您便可以打开 Web 界面进行浏览和工作。

配置 Application Services 设备

首次启动 Application Services 设备时，必须输入 Application Services 序列号并为 root 用户和 darwin_user 创建密码。

您也可以使用 vCloud Director Web 控制台访问已打开电源的虚拟机。验证您的 Web 浏览器是否已安装 vmware-vmrc 插件的副本。请参见《vCloud API 编程》。

在启动 Application Services 设备的过程中，向 vRealize Automation 服务器和租户注册设备，然后选择是否将预定义的示例内容导入租户。

如果升级到新的 Application Services 版本，请参见 第 35 页，第 4 章“升级 Application Services”。

前提条件

- 验证是否安装了与浏览器一起使用的 vCloud Director VMRC 插件。有关兼容的 Web 浏览器的信息，请参见第 22 页，“Application Services 系统要求”。
- 验证 vSphere Client 是否满足 Web 浏览器要求。请参见 VMware vSphere 文档。
- 第 26 页，“启动 Application Services 设备”。

步骤

1 打开 Application Services 设备。

   a 从 vCloud Director“我的云”视图中，单击与 Application Services 设备对应的控制台列下的图像，然后在 VMRC 控制台内单击。

   这可能需要几秒钟时间，直至建立了到设备的连接且 vApp 控制台视图打开为止。

   b 从 vSphere Client 中，找到已打开电源的虚拟机，然后单击**控制台**选项卡。

2 在提示符下输入 Application Services 序列号，然后按 Enter。

3 输入 root 用户帐户的密码，然后按 Enter。

4 在提示符下，重新输入密码进行确认，然后按 Enter。

5 输入 darwin_user 帐户的密码，然后按 Enter。

   为 darwin_user 帐户选择一个您可以轻松记住的密码。当再次提示输入此密码时，您只有三次尝试输入正确密码的机会。

6 在提示符下，重新输入密码进行确认，然后按 Enter。

7 如果收到提示，请响应请求以启用 SSH 登录。

下一步

第 27 页，“注册设备和 vRealize Automation 服务器”。

注册设备和 vRealize Automation 服务器

向 vRealize Automation 服务器注册 Application Services 设备。通过注册，您可以在 Application Services 中使用 vRealize Automation 用户、业务组和租户，并分配 Application Services 用户角色。

前提条件

第 27 页，“配置 Application Services 设备”。

步骤

1 在提示符处输入 vRealize Automation 服务器 URL，并按 Enter。

例如，输入一个格式为 `https://vcac.it.mycompany.com` 或 `https://192.0.2.115:443` 的安全 URL。

2 输入 vRealize Automation 系统管理员帐户的用户名，并按 Enter。

3 输入 vRealize Automation 系统管理员帐户的密码，并按 Enter。

4 如果向 vRealize Automation 服务器注册失败，系统将提示您跳过注册或重试。

5 如果注册成功，系统会提示您将预定义的示例内容导入到 Application Services。输入 **Y**（代表“是”）或 **N**（代表“否”），并按 Enter。

输入	描述
Y	是，将预定义的示例内容导入到指定租户和业务组。导入后，租户中的所有业务组即可共享该内容。
N	否，不导入预定义的示例内容。

注意 在系统提示您设置管理员用户帐户密码之前，引导脚本安装和启动后台所需的软件进程需要不到一分钟的时间。在此期间，系统可能会出现不响应。

下一步

（可选）第 28 页，“导入预定义的示例内容”。

导入预定义的示例内容

注册 Application Services 设备和 vRealize Automation 服务器后，您可选择将预定义的示例内容安装到 vRealize Automation 租户中。

系统将提示您输入登录 CLI 所需的所有信息，然后才能导入预定义的示例内容。

前提条件

- 第 27 页，“注册设备和 vRealize Automation 服务器”。
- 确认已向租户用户帐户分配所有 Application Services 角色。有关向用户分配角色的信息，请参见《vRealize Automation 租户管理》。
- 确认租户用户属于业务组。

步骤

1 输入租户 URL 名称，然后按 Enter。

vRealize Automation 中的系统管理员在添加租户时指定 URL 名称。

2 输入可以访问租户的帐户的用户名，然后按 Enter。

用户帐户还必须已分配所有 Application Services 角色。

3 输入租户用户的密码，然后按 Enter。

4 输入包含租户用户的业务组的名称，然后按 Enter。

如果业务组名称包含空格，必须用双引号将整个名称括起来。

5 如果 CLI 登录失败，系统将提示您重试。输入 `Y` 或 `N`，然后按 Enter。

输入	描述
Y	重新尝试登录。然后系统会提示您执行步骤 1 到步骤 4。
N	不重新尝试登录。不导入预定义的示例内容。

6 如果 CLI 登录成功，系统将导入预定义示例内容并提示您按任意键继续。请按任意键。

下一步

第 29 页，“打开 Application Services Web 界面”。

打开 Application Services Web 界面

对设备进行配置、将其注册到 vRealize Automation 服务器并（可选）导入预定义示例内容后，您便可以打开 Web 界面进行浏览和工作。

前提条件

- 第 27 页，“注册设备和 vRealize Automation 服务器”。

步骤

1 输入 Application Services 管理员用户帐户的密码，然后按 Enter。

2 在提示符下，重新输入密码进行确认，然后按 Enter。

启动脚本会启动必要的服务并显示用于访问 Application Services 服务器的 URL。

3 使用支持的浏览器导航到 Application Services 服务器。

URL 格式为 `https://`*Application_Services_IP_or_hostname*`:8443/darwin/org/`*tenantURL*，其中 *tenantURL* 是租户 URL 名称。

4 以 vRealize Automation 租户管理员或已分配了一个或多个 Application Services 角色的租户用户的身份登录。用户必须至少属于租户中的一个业务组。

浏览器会打开 Web 界面。请参见第 44 页，“使用 Application Services Web 界面”。

注意 如果要关闭 vCloud Director 中部署的 Application Services 设备，请从 vCloud Director 用户界面执行此操作。请勿使用 vCenter Server 关闭 Application Services 设备。

下一步

如果 darwin_user 由于多次登录尝试失败而锁定，您需要解锁帐户才能继续。请参见第 30 页，“解锁 darwin_user 帐户”。

登录到 Application Services 并熟悉产品功能。请参见第 43 页，“登录 Application Services”和第 44 页，“使用 Application Services Web 界面”。

为您的云环境注册云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”、第 55 页，“注册 vRealize Automation 云提供程序和模板”或第 85 页，“注册 Amazon EC2 云提供程序和模板”。

对连接到 Application Services Web 界面出现的问题进行故障排除

一些情形会导致尝试访问 Application Services Web 界面时出现连接问题。

表 3-1 常见连接错误

错误	可能的原因	可能解决方案
Application Services 虚拟设备没有有效的 IP 地址或网络。	以下问题可能会导致网络连接错误： ■ 未正确配置组织 vDC 网络。 ■ 未在配置虚拟机向导中指定网络设置。 ■ 未在配置虚拟机向导中为 IP 分配设置指定 IP 地址。 ■ 静态 IP 地址不可用。 ■ DHCP 服务器不可用。	必须解决网络连接问题。
Application Services 虚拟设备在启动过程中暂时无法检索有效 IP 地址，或者 IP 地址在启动之后发生更改。	以下问题可能会导致网络连接错误： ■ 网络遇到问题。 ■ 静态 IP 地址不可用。 ■ DHCP 服务器不可用。 ■ IP 地址已显式修改。	使用 root 特权在设备虚拟机中运行以下命令： `/home/darwin/tools/darwin_util.sh -a AUTO`
vFabric tc Server 服务在 Application Services 虚拟设备中遇到错误。	虚拟设备未正确关闭并重新启动。 tc Server 服务未正确启动或重新启动。	请导航至 `/home/darwin/tcserver/darwin/logs/catalina.out` 中的日志文件查看错误详细信息，如有必要，请联系 VMware 技术支持。

解锁 darwin_user 帐户

使用 SSH 客户端登录 `darwin_user` 帐户时，如果在三次尝试后仍未提供正确的密码，则 `darwin_user` 帐户会被锁定。

前提条件

- 验证您是否已登录 Application Services 设备的 VMRC 控制台。

  已为 root 帐户禁用 SSH。

- 验证您是否具有 root 特权。

步骤

1 在 VMRC 控制台中，以 root 用户身份登录。

2 基于 shell 提示符检查失败登录尝试的次数。

```
faillog -u darwin_user
```

3 解锁失败的帐户。

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

重新启动 Application Services

在某些情况下，您可能必须重新启动 Application Services。

步骤

1 使用 SSH 客户端或 Application Services 控制台，登录到 vCloud Director 虚拟机。

2 重新启动 Application Services。

- 如果已使用 `darwin_user` 帐户登录，请输入 **`sudo service vmware-darwin-tcserver restart`**。
- 如果已使用 root 帐户登录，请输入 **`service vmware-darwin-tcserver restart`**。

将 Application Services 配置为对外部 URL 使用代理

即使仅使用 Application Services 在私有云中部署应用程序，某些部署也可能需要从企业防火墙外部访问 URL。例如，操作脚本可能涉及从开源网站下载应用程序位。对于这些情况，可以将 Application Services 配置为使用代理。

必须完成此任务才能创建服务和应用程序，或部署现有的预定义示例服务和应用程序。

Application Services 还包含一个名为 `darwin_global_noproxy.conf` 的代理文件，该文件不定义代理。如果部署环境不需要使用代理，您可以将此文件指定为 global_conf 属性值。Amazon EC2 部署环境不需要使用代理来部署应用程序。

代理支持现已从全局级别扩展到部署环境级别。有关详细信息，请参见第 56 页，“创建 vRealize Automation 部署环境”。

前提条件

- 以**应用程序架构师**和**应用程序目录管理员**身份登录到 Application Services。
- 验证您是否可以访问安装了 Application Services 的虚拟机，以及是否具有使用 darwin_user 帐户登录到操作系统的密码。该密码是在安装期间设置的。请参见第 26 页，“启动 Application Services 设备”。
- 熟悉创建新服务的过程。请参见第 103 页，“将服务添加到库”。

步骤

1 对于 vCloud Director 或 vRealize Automation，将 Application Services 虚拟设备配置为使用代理。

   a 以 **`darwin_user`** 身份登录到 Application Services 虚拟设备。

   b 输入 **`su`** 以切换到根用户。

   c 使用文本编辑器打开文件 `/home/darwin/tcserver/darwin/webapps/darwin/conf/darwin_global.conf`。

   d 更新代理 IP 和端口信息，然后保存文件。

   代理 URL 格式为 http://*代理:端口号*。

   必须通过 `yum update` 或 `install` 命令访问存储库的现有预定义应用程序或库服务会使用此文件中指定的代理。预定义服务和应用程序具有访问此代理所需的脚本。

2 以应用程序目录管理员身份登录到 Application Services，并为新服务或现有服务定义代理特定的属性和脚本。

   a 对于新服务，将服务添加到库。

   b 在新服务或现有服务中，将名为 global_conf 的属性添加到类型为“内容”的服务，然后将值定义为 **`https://DarwinServerIP:8443/darwin/conf/darwin_global.conf`**。

> **小心** 如果向 http_proxy、https_proxy 或 ftp_proxy 属性（其服务脚本将 `darwin_global.conf` 用作文件源）添加一个值，则服务脚本运行时，这些配置的属性会替代应用程序中的任何现有代理信息。

   c 将以下各行添加到需要使用代理的服务的每个操作脚本的开头：

```
# Import global conf
 . $global_conf
```

   d 保存该服务。

3 以应用程序架构师身份登录到 Application Services 并在应用程序蓝图中定义代理特定的属性和脚本，以将应用程序配置为在创建应用程序蓝图时使用代理。

   a 对于应用程序组件，将 global_conf 属性添加到类型为“内容”的应用程序组件，然后将值定义为 **`https://Darwin 服务器 IP:8443/darwin/conf/darwin_global.conf`**。

   b 将以下各行添加到需要使用代理的应用程序组件的每个操作脚本的开头：

```
# Import global conf
 . $global_conf
```

下一步

登录到 Application Services 并熟悉产品功能。请参见第 43 页，“登录 Application Services”和第 44 页，“使用 Application Services Web 界面”。

为您的云环境注册云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”、第 55 页，“注册 vRealize Automation 云提供程序和模板”或第 85 页，“注册 Amazon EC2 云提供程序和模板”。

将 Application Services 注册到 vRealize Automation

注册 Application Services 时，使用 CLI 在 Application Services 环境和 vRealize Automation 服务器之间建立连接，以便可以访问 vRealize Automation 服务目录。

通常，在启动 Application Services 设备的过程中进行注册。如果您需要重新建立注册或注册信息已更改，请使用此过程。

在 Application Services 和 vRealize Automation 之间建立连接后，您可以将部署配置文件发布到 vRealize AutomationApplication Services 目录，这样其他用户便可以请求该部署配置文件。您也可以在注册完成后将 Application Services 角色分配给用户和用户组。

> **注意** 您只能具有一个活动的注册。如果您尝试注册多次，则会收到一条错误消息。

前提条件

- 以**系统管理员**身份登录到 Application Services。
- 验证您是否知道 darwin_user 的密码。该密码是在安装期间设置的。请参见第 26 页，“启动 Application Services 设备”。
- 启动 Application Services CLI。请参见第 197 页，“远程启动 CLI”。
- 必须正确设置和配置 vRealize Automation IaaS 环境和端点。请参见《安装和配置》以及*《适用于虚拟平台的 IaaS 配置》*。

- 您必须具有可用的 SSO 管理员凭据。

步骤

1 在 roo shell 提示符下，将 Application Services 注册到 vRealize Automation 服务器。

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

componentRegistryUrl 是 vRealize Automation 自助目录的 URL。

componentRegistryUrl 的示例 URL 为 https://vcac-oci-071.eng.test.com。

注册过程可能需要大约 30 秒，之后会出现一条状态消息： `vCAC server registered successfully`。

2 （可选）查看 vRealize Automation 服务器信息。

```
list-vcac-info
```

3 （可选）如果 vRealize Automation 服务器信息有任何更改，请将 Application Services 注册到 vRealize Automation 服务器。

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

4 （可选）从 Application Services 中取消注册 vRealize Automation 服务器，以将该设备注册到另一个 vRealize Automation 服务器。

```
unregister-vcac-server
```

仅当 vRealize Automation 无法与 Application Services 搭配使用时，才取消注册该服务器。如果之后您将 Application Services 注册到其他 vRealize Automation 服务器，则现有的云提供程序和部署环境变为不可用，因为基础模板不再有效。如果新旧 vRealize Automation 服务器中的业务组不匹配，您也无法访问任何现有的 Application Services 内容。

下一步

将 Application Services 注册到 vRealize Automation 后，请注册云提供程序和模板。请参见第 55 页，“注册 vRealize Automation 云提供程序和模板”。

升级 Application Services 4

可以使用标准升级过程将 Application Services 安装升级到新版本。

可以从以前的 Application Services 版本升级到当前版本。要从之前版本升级，例如从 Application Director 6.0 或 6.0.1 升级到 Application Services 6.2，请按照升级路径，按从 6.0 依次到 6.0.1、6.1、6.2 的顺序升级。有关详细信息，请参见以前版本的文档。

注意 可以将 vCloud Application Director 6.0.1 迁移到 Application Services 6.1。然后将 Application Services 安装升级到下一版本。有关从 Application Director 6.0.1 迁移到 Application Services 6.1 的详细信息，请参见*《使用 Application Services》*文档的 6.1 版。

本章讨论了以下主题：

- 第 35 页，“升级 Application Services”
- 第 36 页，“回滚 Application Services 升级”

升级 Application Services

您可以下载并运行 Application Services 升级安装程序以升级您的 Application Services 安装。

需要具有 root 权限才能下载并运行 Application Services 升级安装程序。

注意 必须先升级到 vRealize Automation 6.2.x，然后才能升级 Application Services。

在升级过程中，安装和卸载操作会停止 Application Services 服务器。

前提条件

- 验证当前安装的 Application Services 版本是否为 6.1.x。
- 验证没有操作正在进行或在 Application Services 服务器上运行。
- 验证您已使用 root 权限登录 Application Services 服务器。
- 验证您已升级到 vRealize Automation 6.2.x。

步骤

1. 导航到 VMware 下载站点。
2. 下载名为 `ApplicationServices-`*`version-build`*`_Upgrade_Installer.tgz` 的升级文件。
3. 将 `ApplicationServices-`*`version-build`*`_Upgrade_Installer.tgz` 复制到安装了 vCloud Automation Center Application Services 6.1.x 的计算机。

4 创建 `ApplicationServices-*version-build*_Upgrade_Installer.tgz` 文件解压缩到的新目录，例如 `my_appservices_upgrade`。

5 将 `ApplicationServices-*version-build*_Upgrade_Installer.tgz` 文件解压缩到新目录，例如 `my_appservices_upgrade`。

```
tar xvfz ApplicationServices-version-build_Upgrade_Installer.tgz
```

6 将用户凭据切换到 root 才能具有运行升级安装程序的正确权限。

```
su root
```

7 从解压缩升级安装程序所在目录运行升级安装程序。

```
./install.sh
```

8 对所有提示做出响应。

Application Services 将升级到新版本。

下一步

升级完成后，必须执行以下任务：

- 重新启动 vRealize Automation 服务器以完成组件注册表刷新过程。
- 在使用 Application Services Web 用户界面之前清除浏览器缓存。

回滚 Application Services 升级

如果在升级 Application Services 安装期间或之后遇到问题，可以将安装回滚到开始升级过程的版本。

前提条件

执行升级过程，如第 35 页，“升级 Application Services”中所述。

步骤

1 将用户凭据切换到 root 才能具有运行卸载升级程序的正确权限。

```
su root
```

2 从解压缩升级安装程序所在目录运行卸载升级程序。

```
uninstall.sh
```

3 对所有提示做出响应。

4 回滚完成后，清除浏览器缓存，然后使用 Application Services Web 用户界面。

Application Services 将回滚到启动升级过程的版本。

设置用户和组 5

安装 Application Services 后，必须确定哪些用户或用户组可以在 Application Services 中使用哪些类型的对象，以及这些用户或用户组有权执行哪些任务。可使用 vRealize Automation 创建用户组和业务组，并在 Application Services 和 vRealize Automation 中分配用户角色、授权以及批准。

通过向特定用户和用户组分配角色可以选择性地分配管理权限。可以通过向同一业务组分配对象及其用户或组来限制对特定部署环境和云模板的访问。通过用户组，可以同时向多个用户分配角色和授权。通过业务组，可以将 Application Services 中的一组服务和资源与一组用户相关联。

用户角色指定用户可在 Application Services 中执行哪些功能。这些功能包括管理库、管理云提供程序和部署环境、创建应用程序以及部署应用程序。

Application Services 具有以下角色。

角色	函数
应用程序架构师	创建、修改和删除应用程序。
应用程序目录管理员	定义服务、模板、操作系统、任务和标记。
应用程序云管理员	定义资源和部署环境。
应用程序发布者和部署者	■ 将应用程序部署到 vRealize Automation 目录中。 ■ 创建、更新和发布服务、库项目以及操作。

必须以租户管理员身份登录到 vRealize Automation 来为 Application Services 分配角色。有关管理用户、用户组和业务组的详细信息，请参见 vRealize Automation 6.1 文档。

在 Application Services 中使用租户和业务组

6

Application Services 和 vRealize Automation 共享租户和业务组。租户是 vRealize Automation 部署中的组织单位。业务组属于租户，并将一组服务和资源与一组用户相关联。

将 Application Services 设备注册到 vRealize Automation 部署（也称为实例）以及该实例中的租户。Application Services 系统管理员可将设备注册到不同的租户，但同一时间只能注册到一个租户。

一个租户有多个业务组。租户中的每个业务组都具有成员和对象，如应用程序、服务和外部服务。用户可以成为多个业务组和多个租户的成员。每个对象均由某个业务组所拥有。只有属于业务组的用户可以编辑该组中的对象。

Application Services 中的应用程序、服务和其他对象可以是专用的，也可以是共享的。只有作为特定业务组成员的用户才能查看或编辑该组中的专用对象。租户中所有业务组中的所有用户均可查看共享对象。

作为其功能的一部分，用户角色还确定了用户可创建的对象。例如，已分配应用程序架构师角色的用户可创建应用程序以及其他对象。创建应用程序时，用户会分配业务组并指定应用程序是专用应用程序还是共享应用程序。此后，只有所属业务组的成员才能编辑该应用程序。如果应用程序是专用的，则只有所属业务组的成员才能进行查看。如果应用程序是共享的，则租户中所有业务组的所有成员均可查看该应用程序。无论应用程序是专用应用程序还是共享应用程序，只有所属业务组的成员才能对其进行编辑。

不同的共享级别决定了您可以对蓝图中的组件以及应用程序本身执行哪些操作。作为构建应用程序蓝图的已登录用户，您必须属于应用程序所属业务组才能编辑蓝图。逻辑模板、服务、外部服务、任务、项目和项目存储库均属于业务组，它们可以是专用的，也可以是共享的。如果您选择属于不同业务组的共享蓝图组件，则可以查看这些组件，但无法对其进行更改。仅当您属于组件所属业务组时才能对组件进行更改。共享应用程序时，必须显式共享应用程序蓝图中的所有组件，如逻辑模板、服务及外部服务。

将预定义的内容导入租户 7

可以将预定义的示例内容从 Application Services 设备导入已注册的 vRealize Automation 实例中的租户。

使用 `/home/darwin/tools/import_oob_content.sh` 脚本将预定义的示例内容从 Application Services 设备导入 vRealize Automation 实例。此脚本位于 Application Services 设备中。要将示例内容导入多个租户，必须对每个租户运行该脚本。

日志文件为 `/tmp/standalone_import.log` 和 `/tmp/standalone_import_all.log`。

要对脚本使用非英语字符输入，必须将环境变量 `LC_CTYPE` 设置为 **`en_US.UTF-8`**。

前提条件

- 确认 Application Services 设备已注册到 vRealize Automation 实例。
- 确认 vRealize Automation 实例的租户名称和租户用户凭据。
- 确认租户用户属于预定义导入的目标业务组，并具有所有 Application Services 角色。
- 使用 SSH 登录 Application Services 设备。

步骤

1 将当前目录设置为 `/home/darwin/tools`。

   您必须具有写入权限。

2 运行 `/import_oob_content.sh` 脚本。

   此脚本将提示您输入。

3 输入租户 URL 名称，然后按 Enter。

   vRealize Automation 中的系统管理员在添加租户时指定 URL 名称。

4 输入可以访问租户的帐户的用户名，然后按 Enter。

   用户帐户还必须已分配所有 Application Services 角色。

5 输入租户用户的密码，然后按 Enter。

6 输入包含租户用户的业务组的名称，然后按 Enter。

   如果业务组名称包含空格，必须用双引号将整个名称括起来。

7 如果 CLI 登录失败，系统将提示您重试。输入 **Y** 或 **N**，然后按 Enter。

Enter	描述
是	重新尝试登录。然后系统会提示您执行步骤 3 到步骤 6。
N	不重新尝试登录。不导入预定义的示例内容。

8 如果 CLI 登录成功，系统会导入预定义的示例内容。

下一步

可以使用 Darwin CLI `export-package` 命令创建导出软件包，并使用补充命令 `import-package` 将特定应用程序或服务从该软件包导入到其他 Application Services 设备上的租户。有关 `export-package` 和 `import-package` 命令的详细信息，请参见《使用 Application Services》。

使用 Application Services Web 界面 8

您可以使用 Application Services Web 界面注册云提供程序、维护虚拟机模板和服务的库、创建应用程序、部署应用程序、将部署配置文件发布至 vRealize Automation 服务目录、更新已部署的应用程序以及从云卸载已部署的应用程序。

注意 必须使用 CLI 注册 vRealize Automation 服务器，并在不同的实例间导入和导出软件包。

也可以使用 CLI 来部署应用程序、更新和卸载已部署的应用程序。请参见第 198 页，“常规 CLI 选项”。

本章讨论了以下主题：

- 第 43 页，“登录 Application Services”
- 第 44 页，“使用 Application Services Web 界面”

登录 Application Services

使用分配了 Application Services 用户角色的已注册租户的 SSO 凭据登录 Application Services Web 界面。

您登录后能否执行特定任务取决于分配给该用户帐户的角色。可以查看的对象取决于与该用户关联的业务组，以及这些对象是否也与该业务组相关联。可以使用任何 Application Services 用户角色登录。请参见第 37 页，第 5 章“设置用户和组”。

前提条件

- 验证 Application Services 设备是否具有已注册的 vRealize Automation 实例和租户。
- 验证 SSO 凭据是否包括对已注册租户的访问权限并且至少分配了一个 Application Services 用户角色。
- 验证您是否拥有 Application Services Web 界面的 URL。安装完成后，此 Web URL 显示在托管 Application Services 的虚拟机的控制台中。有关 URL 和启动 Application Services 的说明，请参见第 26 页，“启动 Application Services 设备”。
- 验证您的计算机上是否已安装支持的 Web 浏览器版本。请参见第 22 页，“Application Services 系统要求”。

步骤

1 打开 Web 浏览器并输入已注册租户的 Application Services URL。

   可以通过更改 URL 末尾的租户 ID 来访问已注册租户。如果不指定租户 ID，Application Services 将打开默认租户 `vsphere.local`。

   例如，您可以使用 URL `https://192.0.2.255:8443/darwin/org/dev` 访问已注册租户 `dev`。

2 输入已分配必要角色的用户帐户的凭据。

您可以访问已分配给您的角色的产品领域。例如，如果已向您的用户帐户分配了应用程序架构师角色，则登录 Application Services Web 界面时，“应用程序”页面会打开。如果没有向您的角色分配产品领域，则该用户界面可见，但被禁用。

下一步

熟悉 Application Services 功能。请参见第 44 页，“使用 Application Services Web 界面”。

使用 Application Services Web 界面

使用 Web 界面，可以访问几乎所有的可用功能。

Web 用户界面包括 Web 应用程序的标准功能。例如，使用搜索框，您可以选择与您要搜索的对象相关的筛选条件。如果搜索字符串包含下划线，则 Application Services 会忽略下划线，将其作为通配符。例如，如果一个应用程序称为 test_qe，另一个应用程序称为 testapp，则在搜索框中键入搜索字符串 **test_** 时，这两个应用程序均会显示在搜索结果中。

登录到 Application Services 设备时，根据与您的用户帐户关联的组，您可以访问相应的应用程序、部署和云环境。所有用户均可查看 Application Services 库中所包含的示例应用程序、服务、任务、逻辑模板、外部服务、操作系统以及标记。

Application Services 标题栏包含一个下拉菜单，其中列出了主类别“应用程序”、“部署”、“云”和“库”，“云”和“库”下又列出了一些子类别。从 Application Services 标题栏中，可以选择其他类别，以便从一个页面导航到另一个页面。

“应用程序”页面组织所有应用程序。在此页面中，您可以创建并编辑示例应用程序或自定义应用程序以部署到云环境。“部署”页面同时列出成功部署和失败部署。在此页面中，您可以打开部署摘要页面，在其中可以启动更新过程以扩展群集节点、修改配置以及卸载或删除部署的应用程序。通过“云”类别，您可以创建云提供程序和部署环境。通过“库”类别，可以为应用程序创建服务、外部服务、策略、逻辑模板、任务、操作系统、标记、项目存储库以及项目。还可以使用库中的示例服务和其他组件对应用程序蓝图进行建模。

使用 Application Services 按钮

Application Services 界面包含用于执行特定任务或多个任务的按钮。

表 8-1 Application Services 按钮

图标	名称	描述
	升级更新配置文件	在更新部署上可用。指示更新配置文件已升级到其他部署环境。
	添加关系	在蓝图中可用。可在蓝图中创建组件之间的关系。
	转换为节点数组	在蓝图中可用。创建群集节点。也可以使用该按钮将群集节点转换为单节点。
	添加	在蓝图中可用。在**网卡**选项卡中向节点添加网卡或在**属性**选项卡中添加应用程序组件属性。
	重置	在部署配置文件向导上可用。恢复到原始值。在更新部署配置文件向导、服务版本、逻辑模板版本和任务版本页面上也可用。
	查看任务信息	在执行计划中可用。可查看操作脚本文本、属性和日志。
	展开	在执行计划中可用。可显示群集节点中的所有虚拟机。
	折叠	在执行计划中可用。可最小化群集节点中的所有虚拟机。
	添加脚本任务	在执行计划中可用。向应用程序部署添加自定义任务。

表 8-1 Application Services 按钮 （续）

图标	名称	描述
	通知	任务未完成或用户执行不允许的任务时显示。
	展开	最大化部署摘要页面中的状态窗口。
	最小化	折叠部署摘要页面中的状态窗口。
	展开	最大化部署摘要页面的执行计划状态窗口中的置备任务。
	查看日志	部署后在部署详细信息页面或执行计划上可用。
	复制应用程序版本	创建新版本的现有应用程序。
	快速部署	在“应用程序”页面和应用程序版本卡上可用。可在不配置部署配置文件向导中任何元素的情况下部署应用程序。
	更新部署	启动更新过程以扩展群集节点或修改已部署应用程序的配置。
	卸载部署	从云环境中移除已部署应用程序。

了解 Application Services 图标

Application Services Web 界面包含表示功能可用性的各种图标。

表 8-2 Application Services 图标

图标	名称	描述
显示	组名称	显示在“应用程序”、“服务”、“外部服务”、“逻辑模板”和“部署”页面上，显示在拥有组件的组的名称旁边。
	公开共享	显示在“应用程序”、“服务”、“外部服务”、“逻辑模板”和“部署”页面上，针对给定租户内公开共享的组件。
	专享	显示在“应用程序”、“服务”、“外部服务”、“逻辑模板”和“部署”页面上，针对租户内业务组专用的组件。
	策略扫描成功	显示在“部署”页面上，针对策略扫描成功的部署或已部署应用程序。
	策略扫描失败	显示在“部署”页面上，针对策略扫描失败的部署或已部署应用程序。
	自动绑定占用	显示在蓝图画布中，显示在可与设置为“公开”的其他属性自动绑定的属性旁边。
	自动绑定公开	图标显示在蓝图画布中，显示在可用于自定义设置为“占用”的其他属性的属性旁边。
	额外配置	显示在部署配置文件向导和 vRealize Automation 部署的部署摘要页面的虚拟机详细信息状态窗口中。 在部署配置文件向导中，可以使用“额外配置”替代 vRealize Automation 蓝图自定义属性或添加到现有属性。

为 Application Services 环境设置应用程序置备

9

Application Services 使用 vRealize Automation 中的虚拟云基础架构功能将应用程序部署到 vSphere 环境。

在设置过程中，必须创建自定义 vRealize Automation 蓝图。vRealize Automation 蓝图包括置备虚拟机、云计算机或物理机的规范，用于确定计算机属性、置备计算机的方式及其策略和管理设置。请求计算机时，必须选择从中创建计算机的蓝图。vRealize Automation 会将在构建配置文件或蓝图中定义的一组自定义属性应用于该请求的计算机。在 vRealize Automation 界面中，您可以访问克隆蓝图或链接克隆蓝图并指定属性以替代 vCenter Server 模板中的多个属性。

重要事项 Application Services 仅支持置备到 vSphere 的 vRealize Automation 虚拟蓝图。

在 Application Services 中，必须注册云提供程序和模板。注册云提供程序时，将业务组中特定的 vRealize Automation 蓝图映射到 Application Services 中的云模板。此注册过程会使此 vRealize Automation 蓝图在 Application Services 库中可用。要使用此云模板部署应用程序，必须将云模板映射到逻辑模板。然后必须创建部署环境，并将此部署环境映射到适用的 vRealize Automation 预留策略。

熟悉与设置和配置应用程序置备的云环境相关的主要概念。请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”
- 第 49 页，“在 vRealize Automation 中创建虚拟机模板”
- 第 54 页，“更新 vRealize Automation 中现有的虚拟机模板”
- 第 55 页，“注册 vRealize Automation 云提供程序和模板”
- 第 56 页，“创建 vRealize Automation 部署环境”

创建 vRealize Automation 自定义模板的虚拟机要求

要创建自定义虚拟机模板以在 Application Services 中使用，请验证是否满足特定软件和虚拟化要求，以及特定 Linux 和 Windows 命令是否可用。

表 9-1 自定义模板的虚拟机要求

要求	描述
操作系统	有关您版本的最新版本要求的信息，请参见 vRealize Automation 文档中的支持列表，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。 ■ Red Hat Enterprise Linux ■ Windows Server ■ CentOS ■ SUSE Linux Enterprise ■ Ubuntu
JRE	安装 JRE 1.7.0。 ■ Windows。可以从 Application Services 虚拟设备中的软件包 `http://`*`Application_Director_IP`*`/agent/jre-1.7.0_51-win64.zip` 安装适用于 Windows 的 JRE。请参见第 49 页，“准备 vCenter Server Windows 虚拟机模板”。 ■ Linux。可以选择在模板准备期间安装适用于 Linux 的 JRE。请参见第 51 页，“准备 vCenter Server Linux 虚拟机模板”。
Windows 代理引导程序服务	从 Application Services 虚拟设备下载 http://*Application_Director_IP*`/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip` 文件。 要在基于 Windows 的虚拟机模板上安装代理引导程序服务，请参见第 49 页，“准备 vCenter Server Windows 虚拟机模板”。
vRealize Automation 客户机代理	对于 Windows 模板，请从 vRealize Automation Appliance 管理控制台安装页面下载以下文件。 `https://`*`vcac-va-hostname_or_IPaddress`*`:5480/installer/LinuxGuestAgentPkgs.zip` 该文件包括适用于 Windows 和 Linux 的 vRealize Automation 客户机代理，但您只需下载适用于 Windows 模板的文件。对于 Linux 模板，该文件会自动下载。 要在模板中安装 Windows 和 Linux 客户机代理，请参见第 49 页，“准备 vCenter Server Windows 虚拟机模板”和第 51 页，“准备 vCenter Server Linux 虚拟机模板”。
支持的 Windows 脚本运行	Application Services 支持 Windows CMD、PowerShell 2.0 和 BeanShell 1.3.0 脚本运行。
支持的 Linux 脚本运行	Application Services 支持 Bash 和 BeanShell 1.3.0 脚本运行。
Linux 命令	虚拟机上必须可使用以下 Linux 命令： ■ `wget` ■ `unzip` ■ `sha256sum` ■ `grep` ■ `sed` ■ `setsid` ■ `awk` ■ `ifconfig` ■ `apt-get` ■ `yum` ■ `chkconfig` ■ `dmidecode` ■ `perl`
可选服务	如果因故障排除或其他原因而计划使用 Linux `ssh` 登录或 Windows 远程桌面来远程访问虚拟机，则必须安装适用于 Linux 的 OpenSSH 服务器和客户端或适用于 Windows 的远程桌面服务 (RDS) 且它们正常运行。

重要事项 由于引导过程不得中断，因此请对虚拟机进行配置，以便在显示最终的操作系统登录提示之前，不会有任何原因导致虚拟机的引导过程暂停。例如，确认在虚拟机启动后没有进程或脚本提示用户交互。此要求仅适用于为 Application Services 库创建的虚拟机模板。

在 vRealize Automation 中创建虚拟机模板

要在 vCenter Server 中使用 vRealize Automation 蓝图启动克隆部署，您必须创建一个 vCenter Server 虚拟机模板。

克隆部署将根据 vCenter Server 虚拟机模板创建一个独立的完整虚拟机。链接的克隆部署将参考虚拟机快照。部署将根据虚拟机的 vCenter Server 快照创建一个独立的完整虚拟机。您可以从克隆部署中设置链接的克隆部署。请参见《适用于虚拟平台的 Iaas 配置》。

- 准备 vCenter Server Windows 虚拟机模板第 49 页，

  必须为 vCenter Server 蓝图准备 vRealize Automation Windows 虚拟机模板，才能创建克隆部署。Application Services 使用 vRealize Automation 蓝图将基于 Windows 的应用程序和服务部署到 vSphere 环境中。

- 准备 vCenter Server Linux 虚拟机模板第 51 页，

  必须为 vRealize Automation 蓝图准备 vCenter Server Linux 虚拟机模板，才能创建克隆部署。Application Services 使用 vRealize Automation 蓝图将基于 Linux 的应用程序和服务部署到 vSphere 环境中。

- 创建克隆部署的 vRealize Automation 蓝图第 52 页，

  创建自定义 vRealize Automation 蓝图时，可以允许克隆部署将 Application Services 基于 Windows 或基于 Linux 的应用程序和服务置备到 VMware vSphere 环境中。

- 创建并配置链接克隆部署的 vRealize Automation 蓝图第 53 页，

  创建自定义 vRealize Automation 蓝图时，可以允许链接克隆部署将 Application Services 基于 Windows 或基于 Linux 的应用程序和服务置备到 vSphere 环境中。链接克隆是基于快照的虚拟机的副本。它会占用小部分磁盘空间，可快速置备，在性能优先级不高时使用最为理想。

准备 vCenter Server Windows 虚拟机模板

必须为 vCenter Server 蓝图准备 vRealize Automation Windows 虚拟机模板，才能创建克隆部署。Application Services 使用 vRealize Automation 蓝图将基于 Windows 的应用程序和服务部署到 vSphere 环境中。

前提条件

- 遵循您版本的正确版本要求。请参见 vRealize Automation 文档中的支持列表，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。
- 验证是否已安装和配置 vCenter Server 5、5.1 或 5.5。
- 验证 vSphere Windows 虚拟机是否可用。请参见第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”。
- 验证 vSphere Windows 虚拟机是否尚未安装 vRealize Automation 5.2 或 6.0 客户机代理、或 Application Services 代理引导程序的早期版本。请参见 vSphere 文档。
- 熟悉如何转换成模板并创建操作系统的自定义规范。请参见 vSphere 文档。
- 移除现有的 vRealize Automation 代理服务和 Application Services 代理引导程序服务。请参见第 54 页，“更新 vRealize Automation 中现有的虚拟机模板”。
- 验证是否从网络配置文件中移除了所有网络配置项目。

步骤

1 登录到 vSphere Client，然后打开 Windows 虚拟机。

2 输入 Windows 管理员凭据，然后打开命令提示符。

3 从 http://*Application_Director_IP*/agent/jre-1.7.0_72-win64.zip 下载并安装支持的 Java SE 7 Runtime Environment。

   a 创建 `\opt\vmware-jre` 文件夹，然后将 JRE 文件解压缩到该文件夹。

   b 打开 PowerShell 命令窗口，然后输入 **`type \opt\vmware-jre\bin\java -version`** 以验证安装。

   将显示已安装的 Java 版本。

4 安装 vRealize Automation 客户机代理。

   a 将目录更改为 `C:\`。

   b 从 vRealize Automation Appliance 管理控制台安装页面 (`https://`*VCAC 虚拟设备主机名或 IP 地址*`:5480/installer/`) 下载文件 `GugentZip.zip`（32 位平台）或 `GugentZip_64.zip`（64 位平台）。

   c 右键单击下载的 ZIP 文件，并从弹出菜单中选择“取消阻止”。

   该选择将取消阻止 ZIP 文件内容。

   d 将安装文件解压缩到 `C:\`。

   例如，`C:\VRMGuestAgent\WinService.exe`。

5 从 http://*Application_Director_IP*`/agent/vmware-appdirector-agent-bootstrap-windows_`*VersionNumber*`.zip` 下载 Application Services 代理引导程序文件。

   a 右键单击下载的 ZIP 文件，并从弹出菜单中选择“取消阻止”。

   重要事项 如果不禁用该 Windows 安全功能，将无法使用 Application Services 代理引导程序文件。

   b 将 `vmware-appdirector-agent-bootstrap-windows_`*`version`*`.zip` 文件解压缩到 `\temp` 文件夹。

   c 运行 `rktools.exe` 实用程序，然后将 `NTRights.exe` 实用程序添加到 `\temp` 文件夹。

   在 `install.bat` 脚本中使用 `NTRights.exe` 实用程序以配置要使用 darwin 用户帐户运行的代理引导程序服务。该实用程序包含在 Windows Server 2003 Resource Kit Tools (`rktools.exe`) 中，可从 Microsoft 下载网站中下载。

   d （可选）如果 `NTRights.exe` 实用程序不可用，则在运行 `install.bat` 脚本后，手动配置要使用特定用户帐户运行的代理引导程序服务。

6 要安装 vRealize Automation 的代理引导程序服务，请完成以下步骤。

   a 打开 Windows CMD 控制台，然后导航到 `\temp` 文件夹。

   b 输入命令以安装代理引导程序。

   ```
   install.bat password=Password cloudProvider=vcac vcacServer=IaaS_Server_FQDN
   httpsMode=true
   ```

   `install.bat` 脚本将为代理引导程序服务创建名为 darwin 的用户帐户，并使用设置的密码。*密码*必须满足 Windows 的密码要求。

7 验证 **darwin_user** 用户是否存在。

   a 在命令提示符下输入 `lusrmgr.msc`。

   b 验证 **darwin_user** 用户是否存在且属于管理员组。

   c 将密码设置为永不过期。

   该设置确保模板在 30 天后仍然可用。

如果该用户不可用，则验证 Windows 服务器密码是否正确。

8 验证是否已安装 Application Services 和 VRM Agent 服务。

9 关闭 Windows 虚拟机。

10 为 Windows 虚拟机创建自定义规范。

使用自定义规范可以更改 Windows 操作系统属性，例如主机名称、网络设置和许可证设置。自定义客户机操作系统有助于防止在对具有相同设置的虚拟机部署重复主机名称时产生的问题。

a 选择**主页 > 自定义规范管理器**。

b 选择**新建**以创建自定义规范。

c 在 vSphere Client Windows 客户机自定义向导中，指定自定义规范的名称。

d 在 NetBIOS 部分中，单击**使用虚拟机名称**，然后在“管理员密码”部分中，输入 Windows 操作系统的管理员密码。

e 接受默认设置，然后单击**完成**。

基于 Windows 的 vRealize Automation 蓝图使用此自定义规范信息来创建克隆或链接部署。

11 在清单中，右键单击 Windows 虚拟机，然后选择**模板 > 转换成模板**。

vCenter Server 即会将该虚拟机标记为模板，并在“近期任务”窗格中显示该任务。

下一步

为克隆或链接克隆部署创建 vRealize Automation 蓝图。请参见第 52 页，“创建克隆部署的 vRealize Automation 蓝图”或第 53 页，“创建并配置链接克隆部署的 vRealize Automation 蓝图”。

准备 vCenter Server Linux 虚拟机模板

必须为 vRealize Automation 蓝图准备 vCenter Server Linux 虚拟机模板，才能创建克隆部署。Application Services 使用 vRealize Automation 蓝图将基于 Linux 的应用程序和服务部署到 vSphere 环境中。

前提条件

- 验证是否已安装和配置 vCenter Server。有关您版本的最新版本要求的信息，请参见 vRealize Automation 文档中的支持列表，网址为：https://www.vmware.com/support/pubs/vcac-pubs.html。
- 验证支持的 vSphere Linux 虚拟机是否可用。请参见第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”。
- 熟悉如何转换成模板并创建操作系统的自定义规范。请参见 vSphere 文档。
- 验证 vCenter Server 虚拟机是否满足 Application Services 虚拟机要求。请参见第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”。
- 验证是否从网络配置文件中移除了所有网络配置项目。

步骤

1 登录到 vSphere Client，然后打开 Linux 虚拟机。

2 以 root 特权登录虚拟机并打开一个终端。

3 将安装程序脚本 `http://`*`ApplicationServicesServerIP`*`/tools/preparevCACTemplate.sh` 下载到虚拟机模板。

可以使用 wget 或 scp 命令来下载该文件。

4 确保安装程序脚本具有执行权限。

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

5 运行安装程序脚本。

```
./preparevCACTemplate.sh
```

6 回答所有提示。

如果成功执行安装程序脚本，将显示一条成功消息。如果失败，安装程序脚本将显示一条错误消息并登录到控制台。

7 如果发生错误，请先解决错误，再重复步骤 4 至步骤 6。

8 关闭 Linux 虚拟机。

9 在清单中，右键单击 Linux 虚拟机并选择**模板 > 转换成模板**。

vCenter Server 即会将该虚拟机标记为模板，并在“近期任务”窗格中显示任务。

下一步

第 52 页，“创建克隆部署的 vRealize Automation 蓝图”或第 53 页，“创建并配置链接克隆部署的 vRealize Automation 蓝图”。

创建克隆部署的 vRealize Automation 蓝图

创建自定义 vRealize Automation 蓝图时，可以允许克隆部署将 Application Services 基于 Windows 或基于 Linux 的应用程序和服务置备到 VMware vSphere 环境中。

前提条件

- 验证是否已安装和配置 vRealize Automation 6.1 或更高版本。
- 验证是否已创建您计划与 Application Services 结合使用的 vRealize Automation 业务组。
- 验证 vRealize Automation 蓝图是否可使用 vCenter Server 虚拟机来创建克隆部署。

  对于基于 Windows 的 vRealize Automation 蓝图，必须有可用的自定义规范名称。请参见第 49 页，“准备 vCenter Server Windows 虚拟机模板”。
- 熟悉如何在 vRealize Automation 中创建蓝图。请参见《适用于虚拟平台的 Iaas 配置》。

步骤

1 以**租户管理员**身份登录到 vRealize Automation 控制台。

2 选择**新建蓝图 > 虚拟 > vSphere (vCenter)**。

3 完成**蓝图信息**选项卡上的信息。

4 在**构建信息**选项卡上，从**操作**下拉菜单中选择**克隆**选项。

5 单击**克隆自**浏览并选择创建以在 vRealize Automation 中使用的 vCenter Server 模板。

6 对于基于 Windows 的 vRealize Automation 蓝图，请在**自定义规范**文本框中输入自定义规范名称。

您已在 vCenter Server 虚拟机模板的准备期间创建了自定义规范。

7 根据需要向**属性**和**安全**选项卡添加信息，然后单击**确定**保存所做更改。

此时**蓝图**菜单中将显示 vRealize Automation 蓝图。

下一步

注册配置用于克隆部署的 vRealize Automation 蓝图以供在 Application Services 中使用。请参见第 55 页，“注册 vRealize Automation 云提供程序和模板”。

创建并配置链接克隆部署的 vRealize Automation 蓝图

创建自定义 vRealize Automation 蓝图时，可以允许链接克隆部署将 Application Services 基于 Windows 或基于 Linux 的应用程序和服务置备到 vSphere 环境中。链接克隆是基于快照的虚拟机的副本。它会占用小部分磁盘空间，可快速置备，在性能优先级不高时使用最为理想。

前提条件

- 验证是否已安装和配置 vRealize Automation 6.1 或更高版本。
- 验证是否已创建您计划与 Application Services 结合使用的 vRealize Automation 业务组。
- 验证 vCenter Server 蓝图是否可使用 vRealize Automation 虚拟机来创建链接克隆部署。

  对于基于 Windows 的 vRealize Automation 蓝图，必须有可用的自定义规范名称。请参见第 49 页，“准备 vCenter Server Windows 虚拟机模板”。
- 熟悉如何在 vRealize Automation 中创建蓝图。请参见《适用于虚拟平台的 Iaas 配置》。

步骤

1. 以**租户管理员**身份登录到 vRealize Automation 控制台。
2. 选择**新建蓝图 > 虚拟 > vSphere (vCenter)**，然后命名蓝图。例如，将蓝图命名为 **`vCAC_Clone`**。
3. 完成**蓝图信息**选项卡上的信息。
4. 在**构建信息**选项卡上，从**操作**下拉菜单中选择**克隆**选项。
5. 在**克隆自**下拉菜单中，选择创建以用于 vCenter Server 的 vRealize Automation 模板。
6. 对于基于 Windows 的 vRealize Automation 蓝图，请在**自定义规范**文本框中输入自定义规范名称。

   您已在 vCenter Server 虚拟机模板的准备期间创建了自定义规范，如必备条件部分所述。
7. 单击**确定**保存更改。
8. 选择蓝图，从下拉菜单中请求计算机，并接受默认设置。例如，选择名为 **`vCAC_Clone`** 的蓝图。

   请求信息派生自 vCenter Server 模板中定义的资源。

   计算机请求处理后，显示在**基础架构 > 计算机 > 受管计算机**菜单下。在计算机名称（如 vCAC_Clone）后附加一系列唯一的字母和数字，如 vCAC_Clone-VFF17Y0。
9. 登录到计算机（如 vCAC_Clone-VFF17Y0），然后停止 Application Services 和 vRealize Automation 服务，具体取决于您的操作系统是 Windows 还是 Linux。

服务	描述
vmware-appdirector-agent-service-vcac	适用于 Linux 的 Application Services 引导程序服务
vrm-agent	适用于 Linux 的 vRealize Automation 代理服务
AppDAgentBootstrap	适用于 Windows 的 Application Services 引导程序服务
VRMAgent 或 winservice	适用于 Windows 的 vRealize Automation 6.1 或更高版本的代理服务

10. 运行 Windows 脚本 `agent_reset.bat` 或 Linux 脚本 `agent_reset.sh` 可将运行时日志文件从计算机（如 vCAC_Clone-VFF17Y0）中移除。
11. 关闭计算机的电源，例如，关闭计算机 vCAC_Clone-VFF17Y0 的电源。

下一步

从蓝图中创建快照以进行链接克隆部署。请参见第 54 页，“从 vRealize Automation 蓝图创建快照”。

从 vRealize Automation 蓝图创建快照

创建链接克隆时，必须创建 vRealize Automation 虚拟机的快照。此快照需要的磁盘空间很小，所以链接克隆部署快于克隆部署。

有关创建虚拟机快照的信息，请参见第 54 页，“从 vRealize Automation 蓝图创建快照”。有关使用 vSphere Client 生成虚拟机模板快照的相关信息，请参见 VMware vSphere 文档。

前提条件

- 验证是否已安装和配置 vRealize Automation 6.1 或更高版本。
- 验证是否已创建您计划与 Application Services 结合使用的 vRealize Automation 业务组。
- 验证 vRealize Automation 蓝图是否可用于创建快照。请参见第 53 页，“创建并配置链接克隆部署的 vRealize Automation 蓝图”。

步骤

1 选择**基础架构** > **蓝图** > **蓝图**。

2 单击**新建蓝图** > **虚拟** > **vSphere (vCenter)**，然后命名蓝图。例如，将蓝图命名为 **`vCAC_LinkedClone`**。

3 完成**蓝图信息**选项卡上的信息。

4 在**内部版本信息**选项卡上，从**操作**下拉菜单中选择**链接克隆**选项。

5 选择 vCAC_Clone-VFF17Y0 虚拟机的快照。

   a 单击**克隆自**文本框旁边的**浏览**图标。

   此时将出现“选择快照”对话框。

   b 选择 **vCAC_Clone-VFF17Y0** 虚拟机并单击**下一步**。

   c 选择 **`vCAC_Clone-VFF17Y0_snapshot`** 快照并单击**下一步**。

   d 单击**确定**保存更改。

   vCAC_Clone-VFF17Y0_snapshot 将显示在**克隆自**文本框中。

6 对于基于 Windows 的 vRealize Automation 蓝图，请在**自定义规范**文本框中输入自定义规范名称。

   您已在 vCenter Server 虚拟机模板的准备期间创建了自定义规范。

7 根据需要配置**属性**和**操作**选项卡，然后单击**确定**以保存更改。

   新蓝图（例如 **`vCAC_LinkedClone`**）将显示在**蓝图**页面上。

下一步

注册配置用于链接克隆部署的 vRealize Automation 蓝图以供在 Application Services 中使用。请参见第 55 页，“注册 vRealize Automation 云提供程序和模板”。

更新 vRealize Automation 中现有的虚拟机模板

您必须先运行适用的命令来移除代理引导程序服务，然后才能更新现有 Linux 或 Windows 模板的内容或使用该内容创建新的模板。

对于 Linux 模板，`agent_reset.sh` 命令将重置 Application Services 代理引导程序状态并删除现有的运行时日志文件。您可以以 root 身份登录虚拟机并运行以下命令：

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

对于 Windows 模板，`agent_reset.bat` 命令会删除现有运行时日志文件。在 PowerShell 命令窗口中，键入以下命令。

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

对于 Windows 模板，您还可以移除现有 Application Services 代理引导程序和 vRealize Automation 6.0 或更高版本的客户机代理。在 PowerShell 命令窗口中，运行以下命令。

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

注册 vRealize Automation 云提供程序和模板

Application Services 将通过连接至 vRealize Automation 业务组来注册云提供程序。

vRealize Automation 业务组将用户与一组服务和资源关联，通常对应一个业务、部门或其他组织单位。每个业务组都可访问用于请求计算机的一个或多个 vRealize Automation 蓝图。这些蓝图可以属于一个或多个业务组。要请求计算机，用户必须属于至少一个业务组。vRealize Automation 蓝图中的计算机资源信息可从克隆或链接克隆的 vSphere 模板中手动添加或获取。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证是否已安装和配置 vRealize Automation 6.1 或更高版本。
- 验证是否已创建并配置好了计划与 Application Services 一起使用的 vRealize Automation 业务组。
- 验证是否已使用 CLI 将 Application Services 注册到 vRealize Automation。请参见第 32 页，“将 Application Services 注册到 vRealize Automation”。
- 验证业务组中至少有一个 vRealize Automation 蓝图可供 Application Services 云提供程序连接。该蓝图必须指向满足 Application Services 要求的虚拟机模板。请参见第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 云提供程序**。

2 单击**添加**图标 ( )。

3 完成云提供程序信息。

对于 vRealize Automation 6.0 或更高版本，当您将 Application Services 注册到 vRealize Automation 时，会为已登录的用户填充主机名称和业务组名称。

选项	描述
名称和描述	在任一字段中加上 vRealize Automation 业务组名称。这些字段的文本将显示在部署配置文件向导中的“云提供程序”列。
云提供程序类型	vCAC、vCloud 和 EC2 是支持的类型。
vCAC 基础架构 IP/主机名称/URL	显示已注册的 vRealize Automation 服务器的 IP 地址或主机 URL。该字段是只读的。 安全 vCAC IP 示例：https://192.0.2.115 和 https://192.10.1.113:443。 安全主机名称 URL 示例：https://vcac.it.vmware.com。
用户名和密码	为 vRealize Automation 中选定的业务组添加业务组管理员凭据。 当您输入用户名时，需要域名。用户名示例：myname@mycompany.com。
业务组	从**业务组**下拉菜单中选择业务组。该菜单包含已登录用户的业务组。
共享	显示云提供程序处于“专用”状态。该值始终为“专用”，而且是只读的。

4 要测试您输入的值是否正确，请单击**验证连接**。

5 单击**保存**以注册云提供程序。

如果云提供程序所需的条目无效或留空，系统将提示您进行更正。

6 要注册 vRealize Automation 蓝图，请单击**编辑**。

7 在“模板”部分中，单击 ✚。

8 选中一个或多个云模板旁边的复选框或者列表中的 vRealize Automation 蓝图，然后单击**确定**。

此时将仅显示全局蓝图和属于云提供程序的选定业务组的蓝图。

您注册的模板将添加到云模板列表中，从而在将逻辑模板映射至云模板时可从该模板列表中进行选择。

9 要更改 vRealize Automation 云提供程序的现有信息，请单击工具栏中的**编辑**，进行相应更改，然后单击**保存**。

10 单击**保存**。

最佳做法是在您编辑 vRealize Automation 云提供程序时刷新云模板信息、vRealize Automation 基础架构 IP 和主机名称。

注意 如果 vRealize Automation 环境中的 vRealize Automation 蓝图名称发生更改，则新蓝图名称不会显示在 Application Services 用户界面中。新蓝图名称仅会在云模板元数据中更新。

已注册云提供程序，并可在 Application Services 中使用其模板。

下一步

将 vRealize Automation 云模板映射至 Application Services 逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。

创建部署环境，以便将应用程序部署到 vRealize Automation 环境。请参见第 56 页，“创建 vRealize Automation 部署环境”。

创建 vRealize Automation 部署环境

您可以将部署环境映射至 vRealize Automation 预留策略，然后再将应用程序部署到该云环境中。如果未选择预留策略，vRealize Automation 将在部署时选择任何合适的预留策略。

一个 vRealize Automation 业务组可将多个预留映射至预留策略。预留包含一个池，其中具有内存、存储和网络等资源，这些资源可用于构建计算机。根据预留设置，您可以在指定的存储空间保存虚拟机并确定其可连接的网络。您可以使用预留策略来对类似资源进行分组，以创建定义的服务级别或制定用于特殊用途的特定资源类型。

您可以将带有预留的预留策略添加到 vRealize Automation 蓝图。从此蓝图中请求计算机时，虚拟机将保存到指定的存储空间并分配至已在预留中定义的网络。如果未将预留策略映射至蓝图，则 vRealize Automation 将根据 vRealize Automation 蓝图中虚拟机的资源要求来分配预留。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证是否已安装和配置 vRealize Automation 6.1 或更高版本。
- 验证是否已创建并配置好了计划与 Application Services 一起使用的 vRealize Automation 业务组。
- 验证您是否具有访问业务组的租户管理员特权。

  具有租户管理员特权的用户可以创建和维护业务组。有关业务组的详细信息，请参见《适用于虚拟平台的 IaaS 配置》。
- 验证您是否是业务组的业务组管理员。

- 验证是否在 vRealize Automation 中创建和配置了预留策略。Application Services 部署环境相当于 vRealize Automation 中的预留策略。
- 验证是否在 Application Services 中注册了一个 vRealize Automation 云提供程序。请参见第 55 页，“注册 vRealize Automation 云提供程序和模板”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

2 单击**新建**。

3 完成部署环境信息的填写。

选项	描述
部署环境名称和描述	在任一文本框中加上预留策略名称。这些文本框中的文本将显示在部署配置文件向导中的“部署环境”列中。
云提供程序	如果列表中未显示云提供程序，请取消对话框并选择**云 > 云提供程序**以添加 vRealize Automation 云提供程序。
预留策略	单击**选择**以从您选择的云提供程序所提供的预留策略列表中进行选择。 vRealize Automation 中名称包含 auto 的预留策略不会显示在预留策略列表中。重命名预留策略以便其显示在预留策略列表中。 如果您选择**无预留策略**，则 vRealize Automation 将根据资源可用性来分配预留。

4 选择预留策略并单击**确定**。

vRealize Automation 预留策略将映射至 Application Services 部署环境名称。

5 单击**保存**。

6 （可选）要更改 vRealize Automation 部署环境的现有信息，请单击工具栏中的**编辑**以进行更改。

对于某些部署，您还可以编辑代理设置的属性。您可通过更改这些代理设置来访问外部数据。单击**显示**编辑以下属性：

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

如果未编辑代理设置属性，则会考虑全局代理设置。

创建部署配置文件时，您可以选择此部署环境将应用程序部署到 vSphere 中。

下一步

您可以使用部署环境来映射外部服务实例、创建策略实例或注册解决方案实例。请参见第 57 页，“映射外部服务实例”、第 58 页，“创建策略实例”或第 61 页，“创建 Puppet 解决方案实例”。

映射外部服务实例

您可以将外部服务实例映射至部署环境，以便在将应用程序部署到该部署环境时始终使用该实例。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。

- 根据您的云提供程序，必须至少在 Application Services 中映射一个 vCloud Director、vRealize Automation 或 Amazon EC2 部署环境。请参见第 76 页，“创建 vCloud Director 部署环境”、第 76 页，“创建 vCloud Director 部署环境”、第 56 页，“创建 vRealize Automation 部署环境”或第 86 页，“创建 Amazon EC2 部署环境”。
- 如果您正在使用自定义外部服务，请验证库中的预配置外部服务或自定义外部服务是否可用。请参见第 109 页，“将外部服务添加到库”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

2 选择现有部署环境。

3 选择**外部服务实例**选项卡，然后单击**注册**以创建外部服务实例。

4 填写部署环境的外部服务实例信息。

选项	描述
名称和描述	在任一字段中加上外部服务名称。名称字段中的文本将显示在“部署环境”列下的部署配置文件向导中。
外部服务版本	如果外部服务未显示在下拉列表中，请取消对话框并选择**库 > 外部服务**将自定义外部服务添加到库。
服务提供程序	单击向下箭头可从与外部服务相关联的服务提供程序列表中进行选择。 **注意** 将自定义外部服务映射至部署环境时，会显示该选项。

在外部服务版本或提供程序规范版本中定义的属性将自动填充至“属性”部分。

5 单击**保存**。

6 单击外部服务实例名称旁的箭头可返回上一页。

创建策略实例

对于特定部署环境中的所有应用程序置备活动，策略实例允许现有策略定义生效。

注意 复合部署功能不支持策略合规性扫描。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 根据您的云提供程序，必须至少在 Application Services 中映射一个 vCloud Director、vRealize Automation 或 Amazon EC2 部署环境。请参见第 76 页，“创建 vCloud Director 部署环境”、第 76 页，“创建 vCloud Director 部署环境”、第 56 页，“创建 vRealize Automation 部署环境”或第 86 页，“创建 Amazon EC2 部署环境”。
- 如果使用自定义策略，请验证其在库中是否可用。请参见第 118 页，“将策略添加到库”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

2 选择现有部署环境。

3 选择**策略实例**选项卡，然后单击**创建策略实例**创建策略实例。

4 填写部署环境的策略实例信息。

选项	描述
名称和描述	在以下任意字段中包含策略名称。这些字段中的文本将显示在部署环境的策略实例部分中。
策略	如果策略不显示在下拉列表中，请取消对话框并选择**库 > 策略**，向库添加自定义策略。
版本	单击向下箭头可从策略关联的版本列表中进行选择。
严重	无论部署的当前合规性状况如何，选中此复选框可停止导致不合规部署的操作。 使用此选项之前，建议在部署环境中完全建立策略合规性。不严重的策略实例会标记部署的合规性状况，但不会停止操作。

注意 如果在策略实例中指定新的属性值，则新属性值将替代策略中定义的现有库值。

策略版本中定义的属性和脚本将自动填充到“属性”和“操作”部分中。

创建策略操作脚本时，策略实例将制作一个副本。对库中策略脚本所做的更改不会影响与该策略关联的现有策略实例。要使脚本更改生效，需要创建策略实例。

5 单击**保存**。

6 单击策略实例名称旁边的箭头可返回上一页。

注册 Puppet 解决方案实例

要注册解决方案实例（例如 Puppet Master），必须将 Puppet Master 配置为与 Application Services 服务器配合工作，并在 Application Services 的部署环境中创建 Puppet Master 解决方案实例。

对于注册过程，可以执行手动或自动选项。有关自动选项的说明，请参见 第 64 页，“注册 Puppet Master 解决方案实例”。

步骤

1 准备 Puppet Master 环境以进行注册第 59 页，

要在部署环境中注册 Puppet Master，必须准备 Puppet Master 以与 Application Services 服务器配合使用。

2 创建 Puppet 解决方案实例第 61 页，

必须在 Application Services 部署环境中创建解决方案实例（例如 Puppet Master），以便在该部署环境中部署使用基于 Puppet 的服务的应用程序时，始终使用 Puppet Master。

准备 Puppet Master 环境以进行注册

要在部署环境中注册 Puppet Master，必须准备 Puppet Master 以与 Application Services 服务器配合使用。

您也可以实现注册过程自动化，请参见 第 64 页，“注册 Puppet Master 解决方案实例”。

前提条件

- 确认已安装 Puppet Enterprise 3.0.1 到 3.2.3 版或 Puppet Open Source 3.2.4 到 3.6.2 版。
- 确认您已安装支持的最新版本的 Puppet 代理。有关代理安装说明，请参见 Puppet Labs 文档网站。
- 确保有一台可用的客户端除安装 Puppet Master 外还安装了 Puppet。
- 确保 Application Services 和 Puppet Master 服务器的系统时间同步。可以使用 NTP 服务器实现同步。这两个服务器之间的时差不能超过 60 秒。

步骤

1 导航到 Puppet Master 上的目录并创建一个名为 appd_nodes 的目录。

- 在 Puppet Enterprise 上，导航到 `/etc/puppetlabs/puppet/manifests/`。
- 在 Puppet Open Source 上，导航到 `/etc/puppet/manifests/`。

2 打开 `/etc/puppetlabs/puppet/manifests/site.pp` 或 `/etc/puppet/manifests/site.pp` 文件，并将命令 **`import 'appd_nodes/*.pp'`** 添加到文件的第一行。

3 在命令行中，创建 empty_manifest_file.pp 文件。

- 在 Puppet Enterprise 上，输入
  `touch /etc/puppetlabs/puppet/manifests/appd_nodes/empty_manifest_file.pp`
- 在 Puppet Open Source 上，输入
  `touch /etc/puppet/manifests/appd_nodes/empty_manifest_file.pp`

4 打开 MCollective 服务器配置文件，并添加 `plugin.ssl_serializer = yaml` 行（如果缺少）。

- 在 Puppet Enterprise 上，打开文件 `/etc/puppetlabs/mcollective/server.cfg`。
- 在 Puppet Open Source 上，打开文件 `/etc/mcollective/server.cfg`。

5 下载 appdintegn.rb 和 appdintegn.ddl 代理文件。

http://*Application_Services_IP*/artifacts/solutions/puppet/mcollective/agent/appdintegn.ddl

http://*Application_Services_IP*/artifacts/solutions/puppet/mcollective/agent/appdintegn.rb

6 将 appdintegn.rb 和 appdintegn.ddl 代理文件复制到 Puppet Master 上的 *LIBDIR*/mcollective/agent 目录。

LIBDIR 是 MCollective libdir 设置的值，通常设置为 `/opt/puppet/libexec/mcollective/`。

7 重新启动 MCollective 以查看新复制的代理文件。

- 在 Puppet Enterprise 上，输入 **`/etc/init.d/pe-mcollective restart`**。
- 在 Puppet Open Source 上，输入 **`/etc/init.d/mcollective restart`**。

8 在安装了 Puppet 的客户机上，生成一个证书。

`puppet certificate generate Name --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster`

*名称*是与 Puppet Master 对应的 Application Services 证书的字符串名称。*证书临时目录*是在 /tmp 文件夹中创建的本地目录，例如 `/tmp/appd_mc/credentials`。*CAPuppetMaster* 是 Puppet Master 的主机名称或 IP 地址。

9 （可选）如果未开启证书自动签名功能，请在 Puppet Master 上输入命令 **`sudo puppet cert sign Name`**。

10 将已签名的公共证书文件复制到 MCollective 授权客户端列表。

`cp CertDirectory/Name.pem AuthorizedClientsDirectory`

*证书目录*是 Puppet Master 证书目录的配置设置，*授权客户端目录*是 MCollective 服务器 plugin.ssl_client_cert_dir 的配置设置。

Puppet Enterprise 复制命令示例：`cp /etc/puppetlabs/puppet/ssl/ca/signed/vmware-appd.pem /etc/puppetlabs/mcollective/ssl/clients/`

11 在安装了 Puppet 的客户机上，从 Puppet Master 获取签名证书。

```
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 在安装了 Puppet 的客户机上，从 Puppet Master 获取公共证书。

- 在 Puppet Enterprise 上，输入 `puppet certificate find pe-internal-mcollective-servers --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster`。
- 在 Puppet Open Source 上，输入 `puppet certificate find mcollective-servers --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster`。

Application Services 将在 `/etc/puppetlabs/puppet/manifests/appd_nodes` 或 `/etc/puppet/manifests/appd_nodes` 目录中为其置备的每个虚拟机创建节点定义文件 *.pp。

下一步

在 Application Services 的部署环境中注册 Puppet Master。请参见第 61 页，“创建 Puppet 解决方案实例”。

创建 Puppet 解决方案实例

必须在 Application Services 部署环境中创建解决方案实例（例如 Puppet Master），以便在该部署环境中部署使用基于 Puppet 的服务的应用程序时，始终使用 Puppet Master。

您只能在每个部署环境中创建一个 Puppet Master 解决方案实例。多个部署环境的解决方案实例可以指向同一 Puppet Master。有关使用在 Puppet Master 服务器本地运行的 Ruby 脚本的信息，请参见第 64 页，“注册 Puppet Master 解决方案实例”。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 根据您的云提供程序，必须至少在 Application Services 中映射一个 vCloud Director、vRealize Automation 或 Amazon EC2 部署环境。请参见第 76 页，“创建 vCloud Director 部署环境”、第 76 页，“创建 vCloud Director 部署环境”、第 56 页，“创建 vRealize Automation 部署环境”或第 86 页，“创建 Amazon EC2 部署环境”。
- 确保将 Puppet Master 配置为与 Application Services 服务器一起使用。请参见第 59 页，“准备 Puppet Master 环境以进行注册”或第 64 页，“注册 Puppet Master 解决方案实例”。
- 如果使用自定义策略，请验证其在库中是否可用。请参见第 118 页，“将策略添加到库”。

步骤

1 登录到 Application Services Web 界面。

2 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

3 选择现有部署环境。

4 选择**解决方案实例**选项卡，然后单击**创建解决方案实例**。

5 完成部署环境的解决方案实例信息。

选项	描述
名称和描述	包含解决方案名称。 您可以在描述部分为解决方案添加属性或代理配置信息。
解决方案	从下拉列表中选择可用的 Puppet Master v1.0.0 解决方案。

此时将填充解决方案属性和相应的代理。

6 在“属性”部分输入所需的属性值。

属性	描述
puppet_server	Puppet Master 服务器的限定域名。 要让代理信任 Puppet Master 证书，请使用您安装 Puppet Master 时所设置的有效 DNS 名称之一。Application Services 服务器必须能够通过此 DNS 名称来联系 Puppet Master，除非同时提供了 puppet_server_ip 属性。
puppet_server_ip	Puppet Master 服务器 IP 地址。 如果在 Application Services 服务器或代理节点中无法解析 Puppet Master 服务器的主机名，则必须指定 IP 地址。
puppet_environment	分配给 Puppet 代理的 Puppet Master 环境的名称。此名称与 Puppet 代理的环境配置属性相对应。
agent_run_interval	Puppet 代理应用库的频率。默认的运行时间间隔为 30 分钟 (30m)。您能够以下列单位指定运行时间间隔。 ■ 秒。例如，30 或 30s。 ■ 分钟。例如，30m。 ■ 小时。例如，6h。 ■ 天。例如，5d。 ■ 年。例如，2y。 如果运行时间间隔为 0，则 Puppet 代理持续运行。
identity_private_key	Puppet Master 向 Application Services 服务器发放的证书的专用密钥。 Application Services 将使用该证书进行 Puppet Master 和 MCollective 身份验证。证书格式采用 PEM 编码，包括标头和结尾。示例标头格式为 '-----BEGIN RSA PRIVATE KEY--' 复制当您准备 Puppet Master 时所创建的文件 *证书临时目录*/private_keys/*名称*.pem 的内容，将其作为此属性的值。
identity_public_cert	与 identity_private_key 属性相对应的 X509 证书。 证书格式采用 PEM 编码，包括标头和结尾。示例标头格式为 '-----BEGIN CERTIFICATE--' 复制当您准备 Puppet Master 时所创建的文件 *证书临时目录*/certs/*名称*.pem 的内容，将其作为此属性的值。
ca_cert	证书颁发机构的 X509 证书，该颁发机构发放用于 Puppet Master 和 MCollective 身份验证的数字证书。 证书格式采用 PEM 编码，包括标头和结尾。示例标头格式为 '-----BEGIN CERTIFICATE--' 复制当您准备 Puppet Master 时所创建的文件 *证书临时目录*/certs/ca.pem 的内容，将其作为此属性的值。
mc_servers_shared_cert	MCollective 群集的共享服务器证书或公用密钥。 证书格式采用 PEM 编码，包括标头和结尾。示例标头格式为 '-----BEGIN CERTIFICATE-----' 或 '-----BEGIN PUBLIC KEY--' 复制当您准备 Puppet Master 时所创建的适用于 Puppet Enterprise 的文件 *证书临时目录*/certs/pe-internal-mcollective-servers.pem 或适用于 Puppet Open Source 的文件 *证书临时目录*/certs/mcollective-servers.pem 的内容，将其作为此属性的值。 示例结尾格式为 ' -----END CERTIFICATE-----' 或 '-----END RSA PUBLIC KEY--'

属性	描述
mc_messaging_server_password	与 MCollective 服务器配置文件中的配置设置 plugin.activemq.pool.1.password 相对应的密码。 例如，Puppet Enterprise 中的 MCollective 服务器配置文件位于 `/etc/puppetlabs/mcollective/server.cfg`。
identity_cert_name	身份证书的名称。 此名称是在准备 Puppet Master 过程中，生成与 Puppet Master 相对应的 Application Services 证书时所设置的字符串。
puppet_version	Puppet Enterprise 的 Puppet Enterprise 套装版本和 Puppet Open Source 的 Puppet 软件包版本。 例如，Puppet Enterprise 版本为 3.2.3，Puppet Open Source 版本为 3.6.2。
mc_messaging_server_username	与 MCollective 服务器配置文件中的配置设置 plugin.activemq.pool.1.user 相对应的用户名。 例如，Puppet Enterprise 中的 MCollective 服务器配置文件位于 `/etc/puppetlabs/mcollective/server.cfg`。
mc_messaging_server_port	与 MCollective 服务器配置文件中的配置设置 plugin.activemq.pool.1.port 相对应的端口。 例如，Puppet Enterprise 中的 MCollective 服务器配置文件位于 `/etc/puppetlabs/mcollective/server.cfg`。
node_manifest_dir	使用 Application Services 部署的虚拟机中节点定义清单所在的目录。 Puppet Master site.pp 必须能够从此目录中导入 *.pp 文件。
mc_appd_agent_ddl	MCollective Application Services 代理 DDL 文件的位置。 无需替代此值。
mc_client_setup_script	MCollective Client 设置 Beanshell 脚本的位置。 无需替代此值。
global_conf	用于为每个节点下载 Darwin 全局配置的 URL。
agent_post_install_wait	安装代理后要等待的时间（以秒为单位）。

7 在“代理”部分输入所需的值。

选项	描述
pe_installer_payload_base_path	托管操作系统和架构特定的 Puppet Enterprise tar 文件的基本 URI。不适用于 Windows。 脚本将根据 pe_installer_payload_base_path 以及计算机上操作系统和架构的当前版本来检测该 URI。可以通过为 pe_installer_payload 属性指定一个值来跳过此自动检测。自动检测假定特定目录结构将模拟位于 https://s3.amazonaws.com/pe-builds/ 的 Puppet Labs 所托管 tar 文件的结构。
installer_payload_base_path	pe_installer_payload_base_path 的 Windows 版本。
is_enterprise	指定注册的 Puppet Master 是值设置为 true 的 Puppet Enterprise，还是值设置为 false 的 Puppet Open Source。

8 在“代理”部分输入可选值。

选项	描述
pe_installer_payload	为 Puppet Enterprise 下载 tar 文件的链接。不适用于 Windows。 将该值留空，除非您希望明确提供与操作系统无关的 Puppet Enterprise tar 文件的完整 URI。如果未指定该值，脚本将根据 pe_installer_payload_base_path 以及计算机上当前的操作系统版本和架构来检测 URI。URI 示例为 https://s3.amazonaws.com/pe-builds/released/3.0.1/puppet-enterprise-3.0.1-all.tar.gz
installer_payload	pe_installer_payload 的 Windows 版本。

9 单击**保存**。

Application Services 将检查您输入的属性和代理值，如果缺少所需的值，则会显示一条错误消息。

10 单击**验证连接**以验证与服务器的连接。

11 单击解决方案实例名称旁的箭头可返回上一页。

12 （可选）在已安装 Puppet 的客户机上，要移除您从 Puppet Master 下载公共证书时创建的临时凭据目录，请输入以下命令。

```
rm -rf TempCredsDir
```

注册 Puppet Master 解决方案实例

您可以使用在 Puppet Master 服务器本地运行的 Ruby 脚本，在 Application Services 部署环境下注册 Puppet Master 解决方案实例。

有关与使用 Ruby 脚本等效的替代方式的信息，请参见第 61 页，“创建 Puppet 解决方案实例”。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 您必须具备安装有 Ruby 1.8.7、2.0.x 版（对于 Puppet 3.2 和更高版本）或 2.1.x 版（对于 Puppet 3.5 和更高版本）的 Puppet Master 服务器。

步骤

1 登录 Puppet Master 服务器。

2 从 Application Services 服务器 URL 下载 Ruby 脚本。

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

3 （可选）运行具有 **-h** 选项的 Ruby 脚本，以查看注册选项。

```
ruby RegisterWithAppD.rb -h

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

4 运行 Ruby 脚本，以向 Application Services 实例注册 Puppet Master 解决方案实例。

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

5 确认在脚本成功注册解决方案实例时可以收到通知。

6 登录到 Application Services Web 界面。

7 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

8 选择具有已注册解决方案实例的现有部署环境。

9 选择“解决方案实例”选项卡，然后单击解决方案实例。

10 如果您在 Puppet Master Open Source 上运行 Ruby 脚本，则执行下列步骤。

   a 单击工具栏中的“编辑”。

   b 在“代理”部分中，将 is_enterprise 属性值设置为 false。

   c 单击“保存”以完成操作。

11 单击**验证连接**以检查与服务器的连接。

会显示一条通知告诉您连接成功，并在“验证连接”按钮旁出现绿色选中标记。

下一步

将 Puppet 内容作为服务导入 Application Services 库。请参见第 108 页，“将 Puppet 服务导入库”。

为 vCloud Director 环境设置应用程序置备 10

要使用 Application Services 在 vCloud Director 环境中部署应用程序，必须安装 vCloud Director 5.1.2 或 5.5。

要设置应用程序置备，必须创建具有一个或多个用户帐户和一个组织 vDC 的 vCloud Director 组织，以便 Application Services 可以创建云提供程序并将其注册到组织 vDC。请参见第 24 页，“为 vCloud Director 设置 Application Services”。还必须在 vCloud Director 中创建自定义虚拟机模板。

在 Application Services 中，必须注册云提供程序和模板。注册云提供程序包括指定 vCloud Director 组织。注册云模板时，将 vCloud Director vApp 模板映射到 Application Services 库中的逻辑模板。然后必须创建部署环境，并将此部署环境映射到包含的 vCloud Director 组织 vDC。

熟悉与设置和配置应用程序置备的云环境相关的主要概念。请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”
- 第 69 页，“在 vCloud Director 中创建 Windows 虚拟机模板”
- 第 73 页，“在 vCloud Director 中创建 Linux 虚拟机模板”
- 第 74 页，“验证 vCloud Director 目录中的云模板配置”
- 第 75 页，“更新 vCloud Director 中的现有虚拟机模板”
- 第 75 页，“以 OVF 格式导出虚拟机模板”
- 第 75 页，“注册 vCloud Director 云提供程序和模板”
- 第 76 页，“创建 vCloud Director 部署环境”

创建 vCloud Director 自定义模板的虚拟机要求

在 Application Services 中，将逻辑模板映射到在 vCloud Director 中创建的云模板。这些云模板必须满足特定要求才能在 Application Services 中正常使用。

表 10-1 自定义模板的虚拟机要求

要求	描述
操作系统	支持以下操作系统： ■ Red Hat Enterprise Linux 6.1、6.3 和 6.5（32 位和 64 位） ■ Windows Server 2008 R2 Enterprise with Service Pack 1（64 位）和 Windows Server 2012 R2（32 位和 64 位） ■ CentOS 5.10、6.1、6.3、6.5（32 位和 64 位） ■ SUSE Linux Enterprise 11 SP2（32 位和 64 位） ■ Ubuntu 12.04.2（32 位和 64 位）
VMware Tools	必须安装 VMware Tools，且版本必须为 vCloud Director 5.1.2 或 5.5 以上，或 vCenter Server 5.0、5.1 或 5.5 以上。
CD/DVD 驱动器	vCloud Director 虚拟机上必须至少有一个 CD/DVD 驱动器可用。请参见 vCloud Director 文档。
JRE	可从其中一个软件包安装支持的 JRE，这些软件包在 Application Services 虚拟设备中可用。 ■ 适用于 Windows 的 JRE http://*Application_Services_IP*/`agent/jre-1.7.0_72-win64.zip` 请参见第 70 页，“在 vCloud Director 中配置 Windows 虚拟机模板”。 ■ 适用于 Linux 的 JRE ■ http://*Application_Services_IP*/`agent/jre-1.7.0_72-lin32.zip` ■ http://*Application_Services_IP*/`agent/jre-1.7.0_72-lin64.zip` 要在虚拟机模板上安装 JRE，请参见第 73 页，“在 vCloud Director 中创建 Linux 虚拟机模板”。
Linux 代理引导程序服务	从 VMware 产品下载站点 http://vmware.com/web/vmware/downloads 下载 Linux 代理引导程序软件包 `vmware-appdirector-agent-service`。从以下软件包之一安装 Linux 代理引导程序脚本： ■ http://*Application_Services_IP*/`agent/vmware-appdirector-agent-service_6.0.0.0-0_x86_64.rpm` ■ http://*Application_Services_IP*/`agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.rpm` ■ http://*Application_Services_IP*/`agent/vmware-appdirector-agent-service_6.0.0.0-0_amd64.deb` ■ http://*Application_Services_IP*/`agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.deb` 要在虚拟机模板上安装代理引导程序服务，请参见第 73 页，“在 vCloud Director 中创建 Linux 虚拟机模板”。 如果使用 `preparevCACTemplate.sh` 脚本，则下载这些文件。
vCloud Director 上的 Windows 代理引导程序服务	从 Application Services 虚拟设备下载 http://*Application_Director_IP*/`agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip` 文件。 要在基于 Windows 的虚拟机模板上安装代理引导程序服务，请参见第 70 页，“在 vCloud Director 中配置 Windows 虚拟机模板”。
支持的 Windows 脚本运行	Application Services 支持 Windows CMD 或 PowerShell 2.0 脚本运行。
支持的 Linux 脚本运行	Application Services 支持 Bash 脚本运行。

表 10-1 自定义模板的虚拟机要求 （续）

要求	描述
Linux 命令	虚拟机上必须可使用以下 Linux 命令： ■ `wget` ■ `md5sum` ■ `grep` ■ `sed` ■ `setsid` ■ `awk` ■ `ifconfig` ■ `apt-get` ■ `yum`
可选服务	如果因故障排除或其他原因而计划使用 Linux `ssh` 登录或 Windows 远程桌面来远程访问虚拟机，则必须安装适用于 Linux 的 OpenSSH 服务器和客户端或适用于 Windows 的远程桌面服务 (RDS) 且它们正常运行。

重要事项 由于引导过程不得中断，因此请对虚拟机进行配置，以便在显示最终的操作系统登录提示之前，不会有任何原因导致虚拟机的引导过程暂停。例如，确认在虚拟机启动后没有进程或脚本提示用户交互。此要求仅适用于为 Application Services 库创建的虚拟机模板。

在 vCloud Director 中创建 Windows 虚拟机模板

通过 Windows 虚拟机模板，您可以使用 Application Services 5.1.2 或 5.5 上的 vCloud Director 来创建应用程序蓝图，并部署基于 Windows 的服务和应用程序，如 .NET 应用程序。

创建 Windows 虚拟机模板前，必须设置 vCloud Director 环境。

- 创建 vCloud Director vApp 第 69 页，

  创建基于 Windows 的 vCloud Director vApp 时，必须在虚拟机中安装并配置 VMware Tools。

- 在 vCloud Director 中配置 Windows 虚拟机模板第 70 页，

  必须为 Windows 虚拟机配置 Java SE 7 Runtime Environment (JRE) 包和代理引导程序文件，Windows 虚拟机才能在 Application Services 中正常工作。

- 为 Windows 虚拟机模板启用 SID 更改和域加入第 71 页，

  您可以对一个 Windows 虚拟机模板使用“更改 SID”和启用“加入域”。使用“更改 SID”选项，所有已部署的虚拟机都可以获得一个唯一的安全标识符 (SID)。在准备 Windows 虚拟机模板时请启用该选项。使用“加入域”选项，已部署的虚拟机可以成为指定的 Windows Active Directory 域的成员。

- 将 Windows 虚拟机模板添加到 vCloud Director 目录第 72 页，

  请将 Windows 虚拟机模板添加到 vCloud Director 目录并对其进行注册，以便其能够与 Application Services 配合使用。

创建 vCloud Director vApp

创建基于 Windows 的 vCloud Director vApp 时，必须在虚拟机中安装并配置 VMware Tools。

前提条件

- 验证是否已安装和配置 vCloud Director 5.1.2 或 5.5。
- 验证计划与 Application Services 配合使用的 vCloud 组织已创建并已配置有一个用户帐户，该用户帐户具有创建和共享 vCloud Director 库模板的特权。
- 熟悉如何在 vCloud Director 中创建一个 vApp。请参见 vCloud Director 文档。

- 有关建议的存储和内存值以及远程桌面服务的设置说明，请参见“Windows Server 2008 R2 Enterprise Service Pack 1 的系统要求”。
- 熟悉如何安装和配置 VMware Tools。

步骤

1 登录 vCloud Director。

2 使用一个虚拟机创建 vApp。

   分配合适的虚拟存储和虚拟内存，以支持 Windows Server 2008 R2 Enterprise Service Pack 1 或 Windows Server 2012 R2 操作系统和您的应用程序。

3 在虚拟机中使用**完全安装**选项安装 Windows Server 2008 R2 Enterprise Service Pack 1 或 Windows Server 2012 R2。

4 在虚拟机中安装 VMware Tools。

下一步

在 Windows 虚拟机中安装支持的 Java SE 7 Runtime Environment 软件包和 Application Services 代理引导程序文件。请参见第 70 页，“在 vCloud Director 中配置 Windows 虚拟机模板”。

在 vCloud Director 中配置 Windows 虚拟机模板

必须为 Windows 虚拟机配置 Java SE 7 Runtime Environment (JRE) 包和代理引导程序文件，Windows 虚拟机才能在 Application Services 中正常工作。

代理引导程序是虚拟机第一次启动后在 Windows 上作为 NT 服务运行的一个守护进程。Application Services 代理引导程序使用 PowerShell 脚本来运行引导程序代码。

前提条件

- 验证是否已安装和配置 vCloud Director 5.1.2 或 5.5。
- 验证是否至少有一个 vApp 是在 vCloud Director 中创建的。请参见第 69 页，“创建 vCloud Director vApp”。

步骤

1 以属于本地管理员组的用户身份登录 Windows 虚拟机。

2 从 http://*Application_Director_IP*/agent/jre-1.7.0_51-win64.zip 下载并安装支持的 Java SE 7 Runtime Environment。

3 将 JRE 文件解压缩到 `C:\opt\vmware-jre\` 目录。

4 在 PowerShell 命令窗口中，导航到 `C:\opt\vmware-jre\bin` 目录并输入 **`java -version`** 以验证安装。

   将显示已安装的 Java 版本。

5 从 http://*Application_Director_IP*/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip 下载并安装 Application Services 代理引导程序文件。

6 将 `vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip` 文件解压缩到一个新文件夹。

7 右键单击 agent_bootstrap.ps1 文件查看其属性，并单击**取消阻止**以禁用文件的安全性。

   重要事项 如果不禁用该 Windows 安全功能，将无法使用 Application Services 代理引导程序文件。

8 将 `NTRights.exe` 实用程序添加到新创建的文件夹。

`install.bat` 命令将使用 `NTRights.exe` 实用程序以配置使用 darwin 用户帐户运行的代理引导程序服务。该实用程序包含在 Windows Server 2003 Resource Kit Tools (`rktools.exe`) 中，可从 Microsoft 下载网站中下载。

9 （可选）如果 `NTRights.exe` 实用程序不可用，请在运行 `install.bat` 命令后手动配置代理引导程序服务，以使用特定的用户帐户运行。

`install.bat` 命令会为代理引导程序服务创建一个名为 darwin 的用户帐户，并使用您设置的密码。*密码*必须满足 Windows 的密码要求。

10 要安装代理引导程序服务，请打开 Windows CMD 控制台，导航到该新文件夹，并输入 **`install.bat password=`*密码*` cloudProvider=vcd`**。

11 在命令行中输入 **`services.msc`**，打开 Application Services 代理引导程序服务。

12 在**登录**选项卡中，将**登录身份**选项设置为**此帐户**并输入登录凭据。

此用户帐户的用户名是 **`.\darwin`**。

13 在 PowerShell 命令窗口，完成以下与服务相关的任务。

a 输入 **`net start AppDAgentBootstrap`** 以验证服务成功启动。

b 输入 **`net stop AppDAgentBootstrap`** 以停止服务。

14 删除模板创建过程中生成的运行时日志文件。

`agent_reset.bat`

15 （可选）启用远程桌面以远程访问此虚拟机将来的实例。

您也可以将远程访问用于故障排除。

16 关闭 Windows 虚拟机并停止 vCloud Director 中的 vApp。

17 导航到 `C:\opt\vmware-appdirector\agent-bootstrap\agent_bootstrap.ps1`，将 $gcStatus 值改为 $True，并将 **`Write-Host "Skipping guest customization check for vCD 5.1.2"`** 命令添加到文件。

下一步

在将您的 vApp 保存到 vCloud Director 目录前，可以选择“更改 SID”和“加入域”功能，以允许 Windows 模板成为 Windows Active Directory 域中的成员。请参见第 71 页，“为 Windows 虚拟机模板启用 SID 更改和域加入”。

将 vApp 模板添加到 vCloud Director 目录，以便对其进行注册以在 Application Services 中使用。请参见第 72 页，“将 Windows 虚拟机模板添加到 vCloud Director 目录”。

为 Windows 虚拟机模板启用 SID 更改和域加入

您可以对一个 Windows 虚拟机模板使用“更改 SID”和启用“加入域”。使用“更改 SID”选项，所有已部署的虚拟机都可以获得一个唯一的安全标识符 (SID)。在准备 Windows 虚拟机模板时请启用该选项。使用“加入域”选项，已部署的虚拟机可以成为指定的 Windows Active Directory 域的成员。

如果需要部署必须加入企业托管域的大量 Windows 虚拟机，请使用带预配置域设置的 Windows 模板。对于小型部署或需要灵活加入不同域的部署，更有效的方法是使用不带预配置域设置的 Windows 模板，并将“加入域预定义任务”添加到部署的执行计划中。请参见第 182 页，“添加加入域预定义任务”。

前提条件

- 验证已正确创建 Windows vApp 并已正确配置虚拟机模板。请参见第 69 页，“创建 vCloud Director vApp”和第 70 页，“在 vCloud Director 中配置 Windows 虚拟机模板”。

- 验证 Windows 域名、域帐户和帐户密码均立即可用。
- 验证虚拟机可以连接到指定的域控制器。

  网络上必须存在 DHCP 服务，而且 DHCP 指定的 DNS 服务器能够解析在客户机自定义设置中指定的域名。

步骤

1 从源模板实例化一个 vApp 并找到关联的 Windows 虚拟机。
2 以属于本地管理员组的用户身份登录 Windows 虚拟机。
3 将所有的虚拟机网络接口配置为使用 DHCP。
   a 打开“网络”控制面板。
   b 选择**网络和共享中心 > 更改适配器设置**。
   c 打开每个网络连接并选择**属性**。
   d 将“Internet 协议版本 4/6”设置为**自动获得 IP 地址**和**自动获得 DNS 服务器地址**。
4 关闭 Windows 虚拟机并停止 vCloud Director 中的 vApp。
5 在 vCloud Director 中，打开虚拟机的属性并选择**客户机操作系统自定义**选项卡以启用客户机自定义。
6 在**客户机操作系统自定义**选项卡上选择**更改 SID**。
7 选择**加入域**并输入 Windows 域名、域帐户和帐户密码。
8 选择**添加到库**，从此 vApp 创建一个新的 vApp 模板。

部署过程中，Active Directory 将对登录凭据进行身份验证并允许 Windows 虚拟机成为域的成员。

下一步

将 vApp 模板添加到 vCloud Director 库。请参见第 72 页，“将 Windows 虚拟机模板添加到 vCloud Director 目录”。

将 Windows 虚拟机模板添加到 vCloud Director 目录

请将 Windows 虚拟机模板添加到 vCloud Director 目录并对其进行注册，以便其能够与 Application Services 配合使用。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证是否已安装和配置 vCloud Director 5.1.2 或 5.5。
- 验证 Windows 模板是否满足 Application Services 虚拟机要求。请参见第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”。
- 完成创建 Windows vApp 和配置虚拟机模板所需的准备任务。请参见第 69 页，“创建 vCloud Director vApp”和第 70 页，“在 vCloud Director 中配置 Windows 虚拟机模板”。

步骤

1 登录到 vCloud Director Web 用户界面。
2 导航到 Windows vApp 所在的 vCloud 组织位置。
3 选择 vApp 并单击**添加到库**以在库中创建一个 vApp 模板。

Windows 模板将添加到云模板的列表，您可以在将逻辑模板映射到云模板时从中进行选择。

下一步

注册 Windows 模板以供在 Application Services 中使用。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”。

在 vCloud Director 中创建 Linux 虚拟机模板

借助 Linux 虚拟机模板，可以在 vCloud Director 上使用 Application Services 来创建应用程序蓝图并部署基于 Linux 的服务和应用程序。

前提条件

vCloud Director

- 验证是否已安装和配置 vCloud Director 5.1.2 或 5.5。
- 确认计划与 Application Services 配合使用的 vCloud 组织已创建并已配置有一个用户帐户，该用户帐户具有创建和共享 vCloud Director 目录模板的特权。
- 熟悉如何在 vCloud Director 中创建一个 vApp。请参见 vCloud Director 文档。

  不支持创建多个虚拟机 vApp 模板。
- 验证 vApp 是否符合 Application Services 虚拟机的要求。请参见第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”。
- 熟悉如何安装和配置 VMware Tools。

步骤

1. 登录到 vCloud Director 并使用一个虚拟机创建 vApp。

   分配合适的虚拟存储和虚拟内存以支持 Linux 操作系统和您的应用程序。
2. 在虚拟机中安装支持的 Linux 操作系统。
3. 在虚拟机中安装 VMware Tools。
4. 以 root 特权登录虚拟机并打开一个终端。
5. 使用 YUM 包管理工具或 APT-GET 打包工具下载并安装支持的 JRE 包。

   如果已安装支持的 JRE 包，请跳过此步骤。

   a. 从 http://*Application_Director_IP*/agent/jre-1.7.0_51-lin-*ArchitectureName*.zip 下载 JRE 软件包，其中 *ArchitectureName* 是 32 或 64。

   b. 创建 `/opt/vmware-jre` 目录。

   c. 解压缩 JRE 包。

   ```
   unzip -d /opt/vmware-jre
   ```

   d. 输入 **`/opt/vmware-jre/bin/java -version`** 以验证安装

   将显示已安装的 Java 版本。

   e. 从 `/usr/java/default`、`/usr/java/latest` 和 `/usr/java/1.7.0_21` 目录删除现有 JRE 包。

6 下载并安装 Application Services 代理引导程序服务。

   a 将目录更改为 `/tmp/` 文件夹。

   b 对于基于 RPM 的虚拟机，请从 http://*Application_Director_IP*/agent/vmware-appdirector-agent-service_6.0.0.0-0_*ArchitectureName*.rpm 将代理引导程序下载到该文件夹，并在终端中输入 **`rpm -i vmware-appdirector-agent-service_6.0.0.0-0_`*`ArchitectureName`*`.rpm`** 以安装服务，其中 *ArchitectureName* 是 i386（32 位）或 x86_64（64 位）。

   c 对于基于 DEB 的虚拟机，请从 http://*Application_Director_IP*/agent/vmware-appdirector-agent-service_6.0.0.0-0_*ArchitectureName*.deb 将代理引导程序下载到该文件夹，并在终端中输入 **`dpkg -i vmware-appdirector-agent-service_6.0.0.0-0_`*`ArchitectureName`*`.deb`** 以安装服务，其中 *ArchitectureName* 是 i386（32 位）或 amd64（64 位）。

7 关闭 Linux 虚拟机并停止 vCloud Director 中的 vApp。

8 导航到 Linux vApp 所在的 vCloud 组织的位置。

9 选择 vApp 并单击**添加到库**以在库中创建一个 vApp 模板。

下一步

注册 Linux 模板以在 Application Services 中使用。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”。

验证 vCloud Director 目录中的云模板配置

在 Application Services 中使用模板之前，请验证云模板配置以避免部署失败。

如果您的云模板具有自定义脚本，Application Services 将会用自身的客户机自定义脚本覆盖该脚本。您需要使用客户机自定义脚本设置虚拟机，让它能与 Application Services 服务器通信，以便完成部署过程。

前提条件

- 验证是否已安装和配置 vCloud Director 5.1.2 或 5.5。
- 验证虚拟机模板已上载到 vCloud Director 并已在 vCloud Director 目录中注册为 vApp 模板。

  有关说明，请参见 vCloud Director 文档。

步骤

1 在 vCloud Director Web 界面中，手动实例化一个为云环境创建 vApp 副本的请求。

2 要验证客户机自定义过程是否成功，请打开 `/var/log/vmware-imc/customization.log` 中的日志文件检查其是否将自定义过程的完成状态显示为 `success`。

3 通过 vCloud Director 检查 IP 分配是否具有处于活动状态且正确的 IP 地址。

4 打开并检查 `/opt/vmware-appdirector/agent/logs/agent_bootstrap.log` 中的代理引导程序日志文件。

5 验证 Application Services 代理引导服务是否曾运行并尝试从 Application Services 服务器下载 Application Services 代理 JAR 文件。

   此时，尝试下载 JAR 文件将会失败。如果 Application Services 通过 Application Services 执行计划来部署 vApp，则下载过程会成功。

下一步

验证第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”中列出的项目是否显示在 vApp 中并正常运行。

更新 vCloud Director 中的现有虚拟机模板

要更新现有 Linux 或 Windows 模板的内容或使用该内容创建新的模板，您必须运行适用的命令来移除代理引导程序服务。

对于 Linux 模板，`agent_reset.sh` 命令将重置 Application Services 代理引导程序状态并删除现有的运行时日志文件。您可以以 root 身份登录虚拟机并运行以下命令：

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

对于 Windows 模板，`agent_reset.bat` 命令会删除现有运行时日志文件。在 PowerShell 命令窗口中，键入以下命令。

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

运行适用的命令后，可以将 vApp 实例作为新的 vApp 模板放回目录中。

以 OVF 格式导出虚拟机模板

使用 Application Services 可以创建 OVF 格式的虚拟机模板并将模板导出到 vCloud Director 目录。

如果计划以 OVF 格式将虚拟机模板导出到 vCloud Director 目录，请不要将自定义 OVF 属性添加到导出包。自定义 OVF 属性会禁用 vCloud Director 中的客户机操作系统自定义功能。这项功能禁用后，Application Services 就无法访问虚拟机模板。

注册 vCloud Director 云提供程序和模板

Application Services 通过连接到 vCloud Director 实例和组织来注册云提供程序。可供 vCloud Director 用户访问的目录可提供对虚拟机模板的访问。vCloud 组织的 vDC 映射到部署环境中。

有关设置 vCloud Director 以便与 Application Services 一起使用的信息，请参见第 21 页，“准备安装 Application Services”。

您不能删除链接到某个部署环境的云提供程序。删除未链接到部署环境的云提供程序时，将从逻辑模板中移除所有云模板映射。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证是否已安装和配置 vCloud Director 5.1.2 或 5.5。
- 创建符合 Application Services 要求的虚拟机模板。请参见第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”。
- 验证是否使用组织管理员特权创建并配置了计划与 Application Services 搭配使用的 vCloud 组织。
- 在 Application Services 中映射云提供程序的云模板时，请验证您要映射到的 vCloud Director 模板是否为单个虚拟机。不支持多个虚拟机 vApp 模板。

  具有多个虚拟机的模板无法向 Application Services 云提供程序进行注册。

步骤

1. 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 云提供程序**。
2. 单击**新建**。

3 完成云提供程序信息。

选项	描述
名称和描述	在任意一个字段中包含 vCloud Director 组织名称。这些字段的文本将显示在部署配置文件向导中的“云提供程序”列。
云提供程序类型	vCloud Director 5.1.2 或 5.5 版本类型受支持。
云 IP/主机名	云 IP 或主机名必须与 vCloud Director 实例的 IP 地址或主机名匹配。
组织名称	组织名称必须与 vCloud Director 中的组织名称匹配。
用户名和密码	组织用户的凭据，该用户具有该组织的组织管理员特权。该用户不能是 vCloud Director 系统管理员。 **注意** 如果用户名包含 @ 符号，则在提供登录凭据时将其替换为 %40。

4 要测试您输入的值是否正确，请单击**验证连接**。

5 单击**保存**以注册云提供程序。

如果云提供程序所需的条目无效或留空，则单击**保存**时，系统将提示您进行更正。

6 要注册 vCloud Director 虚拟机模板，请在工具栏中单击**编辑**。

7 在“模板”部分中，单击**新建**以注册模板。

8 在下拉菜单中选择目录，然后单击**获取模板**。

具有多个虚拟机的模板无法向 Application Services 云提供程序进行注册。

此时会在下拉菜单中显示具有单个虚拟机的云模板。

9 选中列表中一个或多个模板旁边的复选框，然后单击**确定**。

您注册的模板将添加到云模板列表中，从而在将逻辑模板映射至云模板时可从该模板列表中进行选择。

10 要更改 vCloud Director 云提供程序的现有信息，请单击工具栏中的**编辑**，进行相应更改，然后单击**保存**。

云提供程序已注册，并且其虚拟机模板及虚拟数据中心可以在 Application Services 中使用。

下一步

将这些 vCloud Director 云模板映射到 Application Services 逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。

创建部署环境，以便将应用程序部署到 vCloud Director 环境。请参见第 76 页，“创建 vCloud Director 部署环境”。

创建 vCloud Director 部署环境

必须将部署环境映射到 vCloud Director 组织 vDC，然后才能部署应用程序。

vCloud Director vDC 提供了一个其中可以存储、部署和操作虚拟系统的环境。例如，您可能具有单独的部署环境进行开发、测试、转储及生产。对于 vCloud Director，部署环境已映射到组织 vDC。

如果部署配置文件中或当前的部署中正使用某部署环境，则无法将其删除。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证 Application Services 中是否至少注册了一个 vCloud Director 云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”。

- 验证是否已在 vCloud Director 组织中创建和配置组织 vDC。Application Services 部署环境等同于 vCloud Director 中的组织 vDC。

  有关设置 vCloud Director 以便与 Application Services 一起使用的信息，请参见第 21 页，“准备安装 Application Services”。

- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

2 单击**新建**。

3 完成部署环境信息的填写。

选项	描述
部署环境名称和描述	在以下任意字段中包括 vDC 名称。“部署环境”列下的部署配置文件向导中将显示这些字段中的文本。
云提供程序	如果列表中未显示云提供程序，请取消对话框并选择**云 > 云提供程序**以添加 vCloud Director 云提供程序。
组织 vDC URI	单击**选择**从您所选的云提供程序所提供的 vDC 列表中进行选择。

4 选择组织 vDC 并单击**确定**。

  vCloud Director 组织 vDC 已映射到 Application Services 部署环境名称。

5 单击**保存**。

6 （可选）要更改 vCloud Director 部署环境的现有信息，请单击工具栏中的**编辑**，进行相应更改，然后单击**保存**。

您创建的部署环境已添加到部署环境列表中，在创建部署配置文件时可以从列表中进行选择。

下一步

您可以使用部署环境来映射外部服务实例、创建策略实例或注册解决方案实例。请参见第 57 页，“映射外部服务实例”、第 58 页，“创建策略实例”或第 61 页，“创建 Puppet 解决方案实例”。

为 Amazon EC2 环境设置应用程序置备 11

要在 Amazon EC2 环境中使用 Application Services 部署应用程序，必须设置一个或多个云环境。

在 Amazon EC2 云环境中设置应用程序置备包括注册 Amazon Web Services (AWS) 用户帐户以及在 Amazon 区域中创建和配置 VPC。通过 Application Services，可以创建云提供程序并将其链接至 Amazon 区域中的 VPC。必须在 Amazon EC2 环境中创建虚拟机模板。这些模板将在应用程序部署过程中用于创建虚拟机。

在 Application Services 中，必须注册云提供程序和模板。注册云提供程序包含指定 Amazon EC2 区域。注册云模板时，将把云环境中的 AMI 映射到 Application Services 库中的逻辑模板。然后必须创建部署环境并将此部署环境映射至 Amazon 虚拟私有云 (VPC) 和可用性区域的组合。

要将 Application Services 应用程序部署到 Amazon EC2，必须在 VPC 中创建被称为端点虚拟机的特殊虚拟机。端点虚拟机用作从企业网络中的 Application Services 设备连接到 Amazon VPC 环境的安全通道（也称为云通道）的目标。VPC 中的应用程序部署通过端点虚拟机与 Application Services 设备通信。可以使用 CLI 创建和管理云通道。请参见第 199 页，“管理云通道”。

图 11-1 Application Services 和 Amazon EC2 拓扑

熟悉与设置和配置应用程序置备的云环境相关的主要概念。请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 80 页，“为 Application Services 配置 Amazon EC2 环境”
- 第 83 页，“创建 Amazon EC2 自定义模板的虚拟机要求”
- 第 83 页，“创建 Amazon EC2 虚拟机模板或 AMI”
- 第 85 页，“注册 Amazon EC2 云提供程序和模板”
- 第 86 页，“创建 Amazon EC2 部署环境”

为 Application Services 配置 Amazon EC2 环境

设置 Amazon EC2 环境要求创建 VPC 作为 Amazon 区域中应用程序部署的目标位置。必须为要部署的 Application Services 应用程序配置此 VPC。

对于 Amazon EC2 的部署，外部网络上的网卡将接收弹性 IP 地址。此外，将为每个部署创建新的安全组以允许在部署的实例之间进行通信。此安全组允许从外部访问 80、8080、8081、8443 和 22 端口。如果需要打开其他端口，请使用 Amazon EC2 管理控制台找到新安全组并添加相应规则。

前提条件

- 设置 Amazon AWS 用户帐户。
- Amazon 为 Amazon EC2 实例的数量、帐户的弹性 IP 地址的数量以及 API 调用的次数定义了默认限制。联系 Amazon 支持可请求增加受限的实例、弹性 IP 地址或 API 调用的数量。有关 Amazon EC2 的详细信息，请参见“Amazon AWS 文档”。

步骤

1. 使用 AWS 用户帐户登录 AWS Management Console。
2. 在 AWS Identity and Access Management (IAM) 仪表板上，设置具有“管理员访问”或“超级用户访问”用户特权的新用户，并为此用户生成访问密钥。

   还可使用主 AWS 用户帐户设置访问密钥，而无需设置新的 IAM 用户。
3. 下载并保存此访问密钥以供 Application Services 稍后使用。
4. 要登录 Amazon EC2 中已部署的虚拟机，请生成名为 titan_keypair 的 SSH 密钥对，然后保存相应的专用密钥。
5. 创建 VPC：在 AWS Management Console 的 VPC 仪表板上，通过公共子网和专用子网创建 VPC。
   a. 使用向导选项通过公共子网和专用子网创建 VPC。

      这将创建两个子网，其实例具有 Internet 的访问权限。应用程序部署过程中，子网对于下载内容非常有用。
   b. 为子网指定可用性区域。

      可以为两个子网使用相同的可用性区域。
   c. 使用 titan_keypair SSH 密钥对。

      还可使用您自己的密钥对。
6. 在 VPC 中，创建名为 EndpointAccess 的安全组，然后设置入站和出站端口访问规则。

端口访问	源或目标
入站 22	0.0.0.0/0
全部入站	*SubnetofVPC*

端口访问	源或目标
全部出站	0.0.0.0/0
（可选）入站 2222	0.0.0.0/0

下一步

在 VPC 中创建 Linux 虚拟机。请参见第 81 页，“创建端点虚拟机”。

创建端点虚拟机

在您的 VPC 中，在 Amazon EC2 中创建名为端点虚拟机的 Linux 虚拟机，以便使用 Application Services 部署应用程序。

一个 Application Services 设备可用于多个端点虚拟机。但是，一个端点虚拟机只能指定为一个 Application Services 设备的 Amazon 区域、VPC 和可用性区域的端点。

前提条件

- 验证是否设置了 Amazon AWS 用户帐户。
- 验证对于应用程序部署 VPC 在 Amazon 区域中是否可用作目标位置。请参见第 80 页，“为 Application Services 配置 Amazon EC2 环境”。

步骤

1 在每个 VPC 中安装一个端点虚拟机，以便使用 Application Services 进行管理。

2 在外部子网上创建基于 Amazon 的 CentOS 虚拟机。

   a 使用 amazon/ami-vpc-nat-1.0.0-beta.i386-ebs AMI 或 amazon/ami-vpc-nat-1.0.0-beta.x86_64-ebs AMI。

   AMI 被预配置为用作 IP 伪装设备。

   b 为该虚拟机使用 EndpointAccess 安全组。

3 设置弹性 IP 地址以用于 VPC，并将该 IP 地址分配给端点虚拟机。

4 使用 OpenSSH 客户端登录具有弹性 IP 地址和专用密钥的端点虚拟机。

```
ssh -i PathToPrivateKeyFile ec2-user@ApplicationDirectorEndpointVM
```

5 在 CLI 中，打开 `/etc/ssh/sshd_config` 配置文件，添加 **`GatewayPorts yes`**、**`ClientAliveInterval 30`** 和 **`ClientAliveCountMax 3`** 行。

6 （可选）如果出站 SSH 被您的公司防火墙阻止，则请您的防火墙管理员将备用端口设置为 2222 以用于 SSH 通信。

   将行 **`Port 2222`** 添加到 `/etc/ssh/sshd_config` 配置文件。

7 重新启动 SSH 守护进程。

```
sudo service sshd restart
```

8 创建 iptable 规则以将内部端口 80 访问权限重新路由到 8080。

```
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 注销端点虚拟机。

下一步

为 Amazon EC2 部署设置云通道以与 Application Services 服务器进行通信。请参见第 82 页，“创建云通道以连接到 Amazon EC2”。

创建云通道以连接到 Amazon EC2

必须创建和启用云通道实例以便 Amazon EC2 VPC 中的部署可以与 Application Services 服务器进行通信。

对于部署的每个 Amazon EC2 实例，从 Application Services 到端点虚拟机的网络应至少具有 1Mbps 的上载带宽。为了实现更快的下载速度，请将应用程序存储在 Amazon Simple Storage Service 上，而非通过云通道从 Application Services 设备进行下载。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证是否已正确设置和配置端点虚拟机。请参见第 80 页，“为 Application Services 配置 Amazon EC2 环境”。
- 验证端点虚拟机的弹性 IP 地址和专用 IP 地址是否立即可用。
- 验证端点虚拟机的专用密钥是否可用于建立从公司网络到端点虚拟机的云通道。
- 确定从运行 Application Services 的网络访问 Amazon EC2 是否需要代理服务器。

  代理服务器或网络必须允许访问网络外的标准 SSH 端口 22。
- 启动 Application Services CLI。请参见第 197 页，“远程启动 CLI”。

步骤

1. 使用 SSH 客户端将端点虚拟机的已下载专用密钥文件复制到 Application Services 设备并将该文件复制到 `/tmp` 目录。
2. 在 `roo` shell 中，创建安全的云通道实例。

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

   可以使用 `--sshPort` 参数指定 22 以外的端口。`--proxyUrl` 是您可以指定代理服务器用于连接到端点虚拟机的可选参数。
3. 启用安全的云通道连接。

   ```
   enable-cloud-tunnel --name TunnelName
   ```
4. 确认已建立安全的云通道连接。

   ```
   test-cloud-tunnel --name TunnelName
   ```

   验证云安全通道连接可能需要几分钟。

Application Services 将建立到 Amazon EC2 上的端点虚拟机的连接。

下一步

符合虚拟机要求并创建 Amazon EC2 AMI。请参见第 83 页，“创建 Amazon EC2 自定义模板的虚拟机要求”和第 83 页，“创建 Amazon EC2 虚拟机模板或 AMI”。

还可以管理 CLI 中的现有云通道连接。请参见第 199 页，“管理云通道”。

创建 Amazon EC2 自定义模板的虚拟机要求

可以使用受支持的 Linux 操作系统来创建适用于您应用程序的自定义虚拟机模板或 Amazon 计算机映像 (AMI)。还可以自定义现有 AMI 或虚拟机模板，并在 Application Services 中使用这些映像或模板。

表 11-1 自定义 AMI 的虚拟机要求

要求	描述
操作系统	CentOS 6.3
JRE	必须安装 JRE 1.7.0。可从其中一个软件包安装支持的首选 JRE，这些软件包在 Application Services 虚拟设备中可用： 适用于 Linux 的 JRE ■ http://*Application_Director_IP*/agent/jre-7u45-linux-amd64.rpm ■ http://*Application_Director_IP*/agent/jre-7u45-linux-i586.rpm 要在基于 RPM 的 AMI 上安装 JRE，请参见第 83 页，“创建 Amazon EC2 虚拟机模板或 AMI”。
Linux 代理引导程序服务	从 VMware 产品下载站点下载 Linux 代理引导程序包 vmware-appdirector-agent-service-ec2，网址为 http://vmware.com/web/vmware/downloads。从以下软件包之一安装 Linux 代理引导程序脚本： ■ 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_i386.rpm 要在基于 RPM 的 AMI 上安装代理引导程序服务，请参见第 83 页，“创建 Amazon EC2 虚拟机模板或 AMI”。 如果使用 preparevCACTemplate.sh 脚本，则下载这些文件。
支持的 Linux 脚本运行	Application Services 支持 Bash 脚本运行。
Linux 命令	虚拟机上必须可使用以下 Linux 命令： ■ wget ■ md5sum ■ grep ■ sed ■ setsid ■ awk ■ ifconfig
可选服务	如果因故障排除或其他原因而计划使用 Linux ssh 登录来远程访问虚拟机，则必须安装适用于 Linux 的 OpenSSH 服务器和客户端且它们正常运行。

重要事项 由于引导过程不得中断，因此请对虚拟机进行配置，以便在显示最终的操作系统登录提示之前，不会有任何原因导致虚拟机的引导过程暂停。例如，确认在虚拟机启动后没有进程或脚本提示用户交互。此要求仅适用于为 Application Services 库创建的虚拟机模板。

创建 Amazon EC2 虚拟机模板或 AMI

要在 Application Services 中使用 Amazon AMI，您必须通过自定义现有 AMI 来创建 AMI。

前提条件

- 熟悉 AWS Management Console 以及启动、配置和终止 Amazon EC2 实例或虚拟机的步骤。
- 验证是否具有访问 AWS Management Console 的正确凭据。
- 标识将要在其中运行 Application Services 应用程序的 AWS 区域，并确认该区域可以访问 AMI。

- 验证 Amazon EC2 AMI 是否满足 Application Services 虚拟机要求。请参见第 83 页，“创建 Amazon EC2 自定义模板的虚拟机要求”。

  有关使用 AMI 的详细信息，请参见 Amazon AWS 文档。

步骤

1 登录到 AWS 控制台并在相应的 Amazon 区域中找到支持 EBS 的 AMI，以部署应用程序。

2 使用 AMI 启动 Amazon EC2 实例，以 root 用户的身份登录，然后打开终端。

3 为基于 RPM 的 AMI 下载并安装支持的 JRE 软件包。

   a 在与 Application Services 处于同一网络的计算机上，从 http://*Application_Director_IP*/agent/jre-7u45-linux-*架构名称*.rpm 下载 JRE 软件包，其中*架构名称*是 i586（适用于 32 位）或 amd64（适用于 64 位）。

   b 将 JRE 软件包上载到 Amazon EC2 实例。

   c 在 Amazon EC2 实例中，输入 **`rpm -i jre-7u45-linux-`*架构名称*`.rpm`** 以安装软件包。

4 为基于 RPM 的 AMI 下载并安装 Application Services 代理引导程序服务。

   a 在与 Application Services 处于同一网络的计算机上，从 http://*Application_Director_IP*/agent/vmware-appdirector-agent-service-ec2_5.2.0.0-0_*架构名称*.rpm 下载代理引导程序服务，其中*架构名称*是 i386（适用于 32 位）或 x86_64（适用于 64 位）。

   b 将代理引导程序服务上载到 Amazon EC2 实例。

   c 在 Amazon EC2 实例中，输入 **`rpm -i vmware-appdirector-agent-service-ec2_5.2.0.0-0_`*架构名称*`.rpm`** 以安装服务。

5 （可选）在 CentOS 操作系统中添加另一个以太网设备 eth1。

   a 查找 `ifcfg-eth0` 文件。

   b 将文件内容复制到 `ifcfg-eth1` 文件中。

   c 在 `ifcfg-eth1` 文件中，将所有 `DEVICE=eth0` 字符串替换为 `DEVICE=eth1` 并保存更改。

   通过第二个以太网设备可为基于此 AMI 的虚拟机配置多个网卡。

6 （可选）清除 Linux 命令历史记录以保护您将创建的 AMI。

   a 删除 `.bash_history` 文件中保存的 Linux 命令历史记录。

   ```
   cat /dev/null > ~/.bash_history
   ```

   b 在当前的 Bash 登录会话中删除 Linux 命令历史记录。

   ```
   history -c
   ```

7 在 AWS 控制台中，停止 Amazon EC2 实例。

小心 如果终止实例，您将丢失所有更改。

8 要创建 Amazon EBS 支持的 AMI，请选择已停止的 Amazon EC2 实例，提供映像名称，然后启动转换为 EBS 支持的映像的过程。

   创建该 AMI 可能需要几分钟时间。

   AMI 可用后即可终止 Amazon EC2 实例。

下一步

将 Amazon EC2 区域映射到 Application Services 云提供程序并提供对 AMI 或虚拟机模板的访问权限。请参见第 85 页，“注册 Amazon EC2 云提供程序和模板”。

注册 Amazon EC2 云提供程序和模板

在 Application Services 中，注册云提供程序意味着使用在有效 AWS 帐户中建立的访问密钥连接到 Amazon EC2 区域。

指定云提供程序信息时，支持以下 Amazon 区域代码值。有关受支持的 Amazon 区域代码值的最新列表，请参见 Amazon AWS《区域和端点》文档。例如，请参见 http://docs.aws.amazon.com/general/latest/gr/rande.html#ec2_region。

表 11-2 常见的受支持 Amazon 区域代码值

Amazon 区域代码值	区域名称
ap-northeast-1	亚太（东京）地区
ap-southeast-1	亚太（新加坡）地区
ap-southeast-2	亚太（悉尼）地区
eu-west-1	欧州（爱尔兰）地区
sa-east-1	南非（圣保罗）地区
us-east-1	美国东部（北维吉尼亚州）地区
us-west-1	美国西部（北加利福尼亚州）地区
us-west-2	美国西部（俄勒冈州）地区

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证 AWS 用户帐户是否设置了访问密钥。
- 创建满足虚拟私有云 (VPC) 要求的虚拟机模板。请参见第 83 页，“创建 Amazon EC2 自定义模板的虚拟机要求”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 云提供程序**。

2 单击**新建**。

3 完成云提供程序信息。

选项	描述
名称和描述	在**描述**文本框中包括详细信息。该文本框中的文本会显示在部署配置文件向导的“云提供程序”列中。
云提供程序类型	Amazon EC2 是支持的类型。
Amazon 区域	指定 Amazon 区域代码值。例如，如果您的区域为美国东部（北维吉尼亚州），请输入 Amazon 区域代码 **us-east-1**。 尝试创建新的部署环境时，输入不受支持的值会阻止可用性通道显示内容。
代理 URL	如果您的公司网络需要代理服务器提供外部连接，请添加 Web 代理 URL 以支持虚拟私有云 (VPC) 服务器连接到 Amazon EC2。
用户名和密码	在 AWS 帐户上设置访问密钥。

4 要测试您输入的值是否正确，请单击**验证连接**。

5 单击**保存**。

如果所需的任何云提供程序条目无效或留空，在您单击**保存**时系统会提示您更正这些条目。

6 要注册 Amazon AMI，请单击**编辑**。

7 在“模板”部分中，单击**新建**以继续注册 AMI 或虚拟机模板。

8 从下拉菜单中选择类别，然后单击**获取模板**。

您可以选择 AWS 帐户所拥有的 AMI 或 AWS 帐户专用的 AMI。

下拉菜单中将显示您创建的或者 AWS 帐户专用的已筛选 AMI 或云模板。

9 选中列表中一个或多个模板旁边的复选框，然后单击**确定**。

您注册的模板将添加到云模板列表中，从而在将逻辑模板映射至云模板时可从该模板列表中进行选择。

10 要更改 Amazon EC2 云提供程序的现有信息，请单击**编辑**并进行更改。

11 单击**保存**。

Amazon EC2 区域已注册，其 AMI 可用于 Application Services 中。

下一步

创建部署环境，以便在云环境中部署应用程序。请参见第 86 页，“创建 Amazon EC2 部署环境”。

将这些 AMI 映射到 Application Services 逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。

创建 Amazon EC2 部署环境

将应用程序部署到云之前，必须先将部署环境映射到 Amazon EC2 VPC 和关联的可用性区域。

可以使用 VPC 置备 AWS 云的逻辑隔离区域，以在其中部署 Application Services 的应用程序。

如果部署环境当前正用于部署配置文件或当前部署，则无法将其删除。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证 Amazon EC2 环境是否已配置为与 Application Services 设备建立安全连接。请参见第 80 页，“为 Application Services 配置 Amazon EC2 环境”。
- 验证是否已在 Application Services 中至少注册了一个 Amazon 云提供程序。请参见第 85 页，“注册 Amazon EC2 云提供程序和模板”。
- 验证安全云通道连接是否可用。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

2 单击**新建**。

3 完成部署环境信息的填写。

选项	描述
部署环境和描述	在以下任意字段中包含 VPC 名称和可用性区域名称。“部署环境”列下的部署配置文件向导中将显示这些字段中的文本。
云提供程序	如果列表中未显示云提供程序，则取消对话框并选择**云 > 云提供程序**以添加 Amazon EC2 云提供程序。
虚拟私有云	单击**选择**查看区域中的可用 Amazon VPC 列表。

选项	描述
可用性区域	单击**选择**查看 Amazon 区域中的可用性区域列表。
云通道	选择连接 Application Services 服务器与端点虚拟机的云通道实例。此端点虚拟机必须位于已在此部署环境的虚拟私有云部分中设置的 Amazon EC2 VPC 中。单击**验证**以检查云通道的安全 SSH 连接状态。

4 单击**保存**。

5 要更改 Amazon EC2 部署环境的现有信息，请单击**编辑**并进行更改。

6 单击**保存**。

已将此部署环境添加到部署环境列表中，您可以在创建部署配置文件时进行选择。

下一步

您可以使用部署环境来映射外部服务实例、创建策略实例或注册解决方案实例。请参见第 57 页，“映射外部服务实例”、第 58 页，“创建策略实例”或第 61 页，“创建 Puppet 解决方案实例”。

开发 Application Services 组件 12

Application Services 提供了一个开放式框架以创建和开发可在虚拟机中安装的组件。

库中的 Application Services 组件包括预定义的示例服务（如 JBoss 和 MySQL）、预定义的任务（如配置 APT 存储库的脚本）以及蓝图应用程序组件（如 WAR 和 SQL_SCRIPT）。

在此信息中，组件包括服务、应用程序组件和自定义任务。所有这些组件都使用操作和属性作为其通用基础框架。

注意 Application Services 库中的示例组件只能在测试环境中使用。

熟悉有关开发可部署组件的主题中经常出现的主要概念。请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 89 页，“定义组件操作”
- 第 91 页，“配置组件属性”
- 第 97 页，“Application Services 预定义属性”
- 第 98 页，“安全组件属性”
- 第 99 页，“所需的组件属性”
- 第 99 页，“开发组件的最佳做法”

定义组件操作

每个组件都包含预定义的生命周期阶段或操作，以用于为服务或应用程序组件执行安装、配置、启动、更新、回滚和卸载脚本。

应用程序目录管理员必须为 INSTALL、CONFIGURE、START、UPDATE、ROLLBACK 或 TEARDOWN 中的至少一个生命周期阶段提供 Bash、Windows CMD、PowerShell 或 BeanShell 脚本。将这些脚本自定义为使用组件属性。

例如，要在虚拟机中部署 Apache Tomcat 服务器，可以添加以下脚本：

INSTALL	下载 Tomcat 服务器安装位，然后安装 Tomcat 服务。
CONFIGURE	设置 JAVA_OPTS、CATALINA_OPTS 和任何其他所需配置。
START	在 Tomcat 服务器中使用启动命令启动 Tomcat 服务。
UPDATE	使用更新脚本修改 Tomcat 服务的配置或使用负载平衡器更改群集大小以扩展已部署的应用程序并管理群集节点。

ROLLBACK 如果 Tomcat 服务的更新过程因性能或安全问题而失败，或更新过程成功但是部署的应用程序存在错误，则修改回滚脚本。

TEARDOWN 修改 Tomcat 服务的卸载脚本，例如卸载部署之前，在应用程序中执行特定的操作。

应用程序架构师可以使用声明对脚本进行参数化，例如将安装程序位置、安装路径和 Tomcat start 命令声明为脚本中的属性。使用参数可以使脚本变得通用。可以在不同的环境中部署服务，而无需修改这些通用脚本。

还可以从操作脚本中修改参数值。这些修改的属性可称为其他组件的属性值。请参见《*使用 Application Services 库服务*》指南。

为操作定义的脚本在 `/tmp/` 目录中运行。Linux 脚本位于 `/tmp/runId/ComponentName-LifecycleStageName`。*runId* 是每个部署的唯一作业标识符，在部署摘要页面的 任务详细信息 状态窗口上可用。Windows 脚本位于 `\Users\darwin\AppData\Local\Temp`。

注意 确认操作脚本运行期间不提示任何需要用户交互的进程。中断会暂停脚本，导致其无限期处于闲置状态，并最终失败。此外，如果 Windows CMD 脚本在非零退出状态下退出，则 Application Services 代理将停止部署，并将其标记为“失败的部署”。使用 `exit /b 0` 来指示成功状态，使用 `exit /b non-zero` 来指示错误状态。

请参见第 90 页，“支持的操作和自定义任务脚本”。

可以在应用程序部署中添加服务或自定义任务，然后定义组件操作。请参见第 103 页，“将服务添加到库”和第 122 页，“将自定义任务添加到库”。

支持的操作和自定义任务脚本

Application Services 支持为基于 Linux 的应用程序编写 Bash 或 BeanShell 脚本，支持为基于 Windows 的应用程序编写 Windows CMD、PowerShell 或 BeanShell 脚本。

要在蓝图画布中为生命周期阶段选择合适的操作脚本类型，请双击**脚本类型**列，然后从下拉菜单中选择脚本。对于自定义任务脚本，请选择一个或多个操作系统。您可以在下拉菜单中设置受支持的脚本。

根据选择的脚本类型，您可以编写代码并通过脚本中的变量访问从属属性。对于生命周期阶段的操作脚本，您也可以为相同组件中的每个生命周期使用相同操作系统系列中的不同脚本类型。例如，可以针对“安装”阶段使用 Windows CMD 脚本，针对“配置”阶段使用 PowerShell 脚本。请参见第 91 页，“属性的类型”。

当您编写操作脚本时，不同脚本类型的退出代码和返回代码也不同。应用程序架构师应该在脚本中设置适用于应用程序部署的合适退出代码。如果脚本缺少退出代码和返回代码，则脚本中运行的最后一个命令会变为退出状态。请参见第 157 页，“了解部署和更新过程”。

表 12-1 操作脚本退出代码和返回代码

脚本类型	描述
Bash	您可以在操所脚本中使用 `return 0` 或 `exit 0` 代码来指示成功状态。要指示错误状态，您可以使用 `return non-zero` 或 `exit non-zero`。
Windows CMD	不要在操作脚本中使用 `exit 0` 和 `exit non-zero` 代码。如果您在脚本中使用这些代码，则计算属性的任务处理将提前停止。使用 `exit /b 0` 来指示成功状态，使用 `exit /b non-zero` 来指示错误状态。
Windows PowerShell	您可以使用 `exit 0` 来指示成功状态，使用 `exit non-zero` 来指示错误状态。
BeanShell	您可以使用 `System.exit(0);` 来指示成功状态，使用 `System.exit(1);/non-zero` 来指示错误状态。

当您使用 Windows PowerShell 编写脚本时，不得在操作脚本中使用 `warning`、`verbose`、`debug` 和 `host` 调用。

配置组件属性

组件属性用于参数化脚本，以便 Application Services 可以将定义的属性作为环境变量传递到在虚拟机中运行的脚本。

运行生命周期阶段的脚本前，虚拟机中的 Application Services 代理将与 Application Services 服务器进行通信以解析属性。随后，该代理将继续通过这些属性创建脚本特定的变量并将其传递到脚本。

- 属性的类型第 91 页，

  Application Services 支持字符串属性、数组属性、内容属性、计算属性、布尔属性、单项选择属性、整数属性和双精度属性。

- 定义属性值第 95 页，

  应用程序目录管理员可以在库中定义服务和自定义任务的属性。

- 绑定到其他属性第 95 页，

  在一些部署方案中，组件需要其他组件的属性值以对其自身进行自定义。在 Application Services 中，此过程称为绑定到其他属性。

- 自动绑定到其他属性第 96 页，

  您可以通过在服务中使用自动绑定元数据设置或在节点、节点数组、服务和 WAR 应用程序组件中使用默认元数据来自动绑定到其他属性。

- 在生命周期阶段和组件之间传递属性值第 97 页，

  为每个组件定义的属性可在各个生命周期阶段的操作脚本中使用。

属性的类型

Application Services 支持字符串属性、数组属性、内容属性、计算属性、布尔属性、单项选择属性、整数属性和双精度属性。

注意 属性的名称区分大小写并且只能包含字母、数字、连字符 (-) 或下划线 (_) 字符。

字符串属性

字符串属性值可以是字符串，也可以是绑定到其他字符串属性的值。字符串值可以包含任何 ASCII 字符。对于绑定属性，使用蓝图画布中的**属性**选项卡可以选择相应的属性进行绑定。属性值随后则会作为原始字符串数据传递到操作脚本。

字符串属性示例	脚本语法	使用示例
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);`

数组属性

数组属性值可以是以 *[“值 1 ”， “值 2 ”， “值 3 ” …]* 形式定义的字符串数组，也可以是绑定到其他数组属性的值。定义数组属性的值时，必须将字符串数组用方括号括起来。对于字符串数组，数组元素中的值可以包含任何 ASCII 字符。要在数组属性值中正确编码反斜线字符，请再添加一个反斜线，例如 `["c:\\test1\\test2"]`。对于绑定属性，使用蓝图画布中的**属性**选项卡可以选择相应的属性进行绑定。

例如，假设某个负载平衡器虚拟机正在平衡一个应用程序服务器虚拟机群集的负载。在这种情况下，可为负载平衡器服务定义一个数组属性，并将该属性设置为应用程序服务器虚拟机的 IP 地址数组。

这些负载平衡器服务配置脚本将使用该数组属性在 Red Hat、Windows 和 Ubuntu 操作系统上配置相应的负载平衡方案。

数组属性示例	脚本语法	使用示例

operating_systems = ["Red Hat","Windows","Ubuntu"]	Bash - `${operating_systems[@]}`
	用于整个字符串数组
	`${operating_systems[N]}`
	用于单个数组元素
		`for (( i = 0 ; i < ${#operating_systems[@]}; i++ )); do`
		`  echo ${operating_systems[$i]}`
		`done`

	Windows CMD - `%operating_systems_N%`
	其中 *N* 表示元素在数组中的位置
		`for /F "delims== tokens=2" %%A in ('set operating_systems_') do (`
		`  echo %%A`
		`)`

	Windows PowerShell - `$operating_systems`
	用于整个字符串数组
	`$operating_systems[N]`
	用于单个数组元素
		`foreach ($os in $operating_systems){`
		`  write-output  $os`
		`}`

	BeanShell - `operating_systems[N]`
	其中 *N* 表示元素在数组中的位置
		`for(index=0;index < operating_systems.length; index++) {`
		`  print(operating_systems[index]);`
		`}`

内容属性

内容属性值是要下载内容的文件的 URL。Application Services 代理将从 URL 将内容下载到虚拟机并将虚拟机中本地文件的位置传递到脚本。

必须将内容属性定义为使用 HTTP 或 HTTPS 协议的有效 URL。例如，假设 Hyperic 项目托管在 Application Services 设备中，并且 URL 指向设备中的该位置。Application Services 代理将从指定的位置将项目下载到部署的虚拟机中。

字符串属性示例	脚本语法	使用示例
HQ_PACKAGE = "http://*Darwin 服务器 IP*/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%`

字符串属性示例	脚本语法	使用示例
	Windows PowerShell - `$HQ_PACKAGE`	`& c:\unzip.exe $HQ_PACKAGE`

	BeanShell - `HQ_PACKAGE`	`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();`
		`entry = zipIn.getNextEntry();`
		`}`
		`zipIn.close();`

计算属性

计算属性类型在进行定义时不允许有初始值。相反，计算属性类型从 INSTALL、CONFIGURE 或 START 生命周期脚本中获取值。分配的值会传播到后续可用生命周期和从属组件。

注意 在操作脚本中定义的计算属性不可用于 UPDATE、ROLLBACK 或 TEARDOWN 生命周期脚本。必须在各个生命周期脚本中设置计算属性的值。

字符串属性示例	脚本语法	使用示例
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";`

布尔属性

布尔属性类型在“值”下拉菜单中提供了“是”和“否”两个选项，但是单击页面的其他部分时，选择的值将更改为 True 或 False。Application Services 将显示“是”和“否”值以避免混淆，并提供与 vRealize Automation 控制台界面的一致性，但会将“是”更改为 True，将“否”更改为 False 以便在操作脚本中使用。

单项选择属性

单项选择属性可用于指定多个字符串值，用户只能从中选择一个值。

整数属性

整数属性类型接受零、正整数或负整数作为值。

双精度属性

双精度属性类型存储精确到四位小数的浮点值。

属性类型参考

属性类型参考显示了进行配置时可参考的属性类型。

属性类型	可绑定属性类型	自动绑定公开	自动绑定占用	接受初始属性值
字符串	字符串、数组、内容和计算	是	是	是
内容	不适用	不适用	不适用	是
数组	字符串、内容和计算	不适用	是	是
计算	不适用	是	不适用	不适用

数组属性也可以绑定到群集中预定义的节点数组属性 all。正则表达式形式的 all 属性是用于收集群集节点中给定属性所有值的一种方法。

有关绑定属性的详细信息，请参见第 95 页，“绑定到其他属性”。要了解自动绑定公开和自动绑定占用的概念，请参见第 96 页，“自动绑定到其他属性”。

绑定到其他类型属性时的字符串属性值示例。

属性类型示例	可绑定属性类型	绑定结果（A 绑定到 B）
字符串（属性 A）	字符串（属性 B="Hi"）	A="Hi"
字符串（属性 A）	内容（属性 B="http://my.com/content"）	A="http://my.com/content"
字符串（属性 A）	数组（属性 B=["1","2"]）	A="["1","2"]"
字符串（属性 A）	计算（属性 B="Hello"）	A="Hello"

绑定到其他类型属性时的数组属性值示例。

属性类型示例	可绑定属性类型	绑定结果（A 绑定到 B）
数组（属性 A）	字符串（属性 B="Hi"）	A="Hi"
数组（属性 A）	内容（属性 B="http://my.com/content"）	A="http://my.com/content"
数组（属性 A）	计算（属性 B="Hello"）	A="Hello"

定义属性值

应用程序目录管理员可以在库中定义服务和自定义任务的属性。

要允许自定义组件脚本，可以在蓝图中替代库的属性默认值，以便满足使用服务的特定应用程序的需求。还可以进一步替代部署配置文件中的属性值，以便符合服务的部署环境要求。

注意 属性值区分大小写。如果新属性值键入错误，则该值不会生效。

例如，要部署 Apache Tomcat 服务器，应用程序目录管理员可能已将服务中的 JVM HEAP_SIZE 值配置为 512MB。对于大型应用程序，应用程序架构师可以将该值重新定义为 1024MB。在生产部署环境中部署应用程序以处理大型负载时，部署者可以将该值替代为 2048MB。

默认情况下，属性值不可替代，但应用程序目录管理员可以选择启用可替代功能。对于可替代的属性，应用程序架构师可以在某些监管环境中强制指定属性的值，并对部署者禁用可替代功能。

在应用程序部署中添加服务或自定义任务时，可以定义属性值。请参见第 103 页，“将服务添加到库”和第 122 页，“将自定义任务添加到库”。

绑定到其他属性

在一些部署方案中，组件需要其他组件的属性值以对其自身进行自定义。在 Application Services 中，此过程称为绑定到其他属性。

应用程序目录管理员可以修改脚本中的属性定义。例如，WAR 组件可能需要 Apache Tomcat 服务器的安装位置。WAR 组件可以将 server_home 属性值设置为 Apache Tomcat 服务器 install_path 属性值。

组件的 Bash 脚本只能使用其自身的属性。除了将属性设置为硬编码值外，Application Services 还允许将属性绑定到蓝图中的其他属性。绑定到其他属性时，可以基于其他组件的属性值以及虚拟机属性（如 IP 地址）值自定义脚本。要将属性绑定到其他属性，请从“编辑属性”对话框的**蓝图值**下拉菜单中选择属性值。

对于单个虚拟机节点，**蓝图值**下拉菜单中的引用属性为 *NodeName*:*ComponentName*:*PropertyName*。

对于群集虚拟机节点，**蓝图值**下拉菜单中的引用属性为 all(*NodeName*:*ComponentName*:*PropertyName*)。当其他属性引用此群集属性时，它会从群集中的所有虚拟机获取 *PropertyName* 属性值。群集虚拟机的预定义 all(*NodeName*:node_array_index) 属性会获取群集中节点数组索引的集合。请参见第 98 页，“预定义的节点数组索引属性”。

对于群集节点和单个节点，self:*ComponentName*:*PropertyName* 值用于指示运行目标组件的虚拟机的组件属性。例如，如果在 Apache Tomcat 服务器中部署 WAR 组件，WAR server_home 属性可设置为 self:tomcat:install_path 以引用在当前虚拟机中运行的 Apache Tomcat 服务器。

节点级属性（如 IP 地址）将显示为 *NodeName*:ip 或 self:ip。这些属性专属于虚拟机，不属于虚拟机中的任何特定组件。

对于具有多个网卡的虚拟机的 IP 地址，Application Services 会提供 *NodeName*:NIC*x*_ip 或 self:NIC*x*_ip 属性。其中 *x* 反映网卡数。请参见第 98 页，“预定义 IP 地址属性”。

图 12-1 绑定到群集节点中的其他属性

自动绑定到其他属性

您可以通过在服务中使用自动绑定元数据设置或在节点、节点数组、服务和 WAR 应用程序组件中使用默认元数据来自动绑定到其他属性。

根据应用程序目录管理员在服务中定义的自动绑定元数据，自动绑定类型会显示为“占用”或“公开”，以允许其他属性在蓝图中绑定到该元数据。在蓝图画布中，“自动绑定公开”属性旁边显示“公开”图标，而“自动绑定占用”属性旁边显示“占用”图标。这些图标的工具提示将显示自动绑定类型和自动绑定标记信息。例如，webapps_dir 属性旁边的“占用”图标在工具提示中显示“自动绑定占用: Servlet 容器，DIR”信息。应用程序目录管理员在创建服务或编辑现有服务时，将设置服务自动绑定类型和标记参数。请参见第 103 页，“将服务添加到库”。

用于自动绑定的预配置组件

WAR 是预配置为自动绑定的唯一应用程序组件。您无法编辑预定义的自动绑定类型和标记参数。

向服务添加应用程序组件意味着也可以设置属性。在这种情况下，将自动绑定两次。应用程序组件上的源属性通过服务和父节点或节点数组上的目标属性进行设置。然后，服务上的源属性通过应用程序组件上的目标属性进行设置。

表 12-2 预配置的 WAR 应用程序组件

应用程序组件	属性	自动绑定类型	自动绑定标记
Linux WAR 文件	Context	公开	上下文
Linux WAR 文件	service_start	占用	Servlet 容器，启动
Linux WAR 文件	service_stop	占用	Servlet 容器，停止
Linux WAR 文件	webapps_dir	占用	Servlet 容器，DIR

部分节点和节点数组已预定义属性的自动绑定类型和标记参数。不能编辑这些参数。向节点或节点数组添加服务或应用程序组件后，将从服务或应用程序组件中获取源属性，从节点或节点数组中获取目标属性。

表 12-3 预定义的节点和节点数组属性

节点类型	属性	自动绑定类型	自动绑定标记
节点，节点数组	hostname	公开	Hostname
节点，节点数组	ip	公开	IP 地址
节点数组	node-array-index	公开	节点数组索引

如果存在依赖关系，源属性将从链接来源的组件中获取。目标属性从链接生成的组件、父服务或节点组件中获取。

属性兼容性

自动绑定取决于源属性和目标属性类型与节点类型之间的兼容性。

表 12-4 属性类型和节点类型兼容性

源属性类型	目标属性类型	节点类型	兼容
数组	字符串	节点	是
数组	计算	节点	是
数组	字符串	节点数组	是
数组	计算	节点数组	是
字符串	字符串	节点	是
字符串	计算	节点	是
字符串	字符串	节点数组	否
字符串	计算	节点数组	否

在生命周期阶段和组件之间传递属性值

为每个组件定义的属性可在各个生命周期阶段的操作脚本中使用。

对于计算属性，您可以修改属性值并将该值传递到操作脚本的下一个生命周期阶段。例如，如果组件 A 的 progress_status 值定义为 staged，则您应在 INSTALL 和 CONFIGURE 生命周期阶段的相应操作脚本中，将该值更改为 progress_status=installed。如果组件 B 绑定到组件 A，则操作脚本的生命周期阶段中的 progress_status 属性值与组件 A 相同。

在蓝图中定义组件 B 依赖于 A。这种依赖关系可定义在这两个组件之间传递正确的属性值，无论它们位于同一节点中还是跨不同节点。

例如，您可以使用支持的脚本更新操作脚本中的属性值。

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

注意 数组和内容属性不支持在生命周期阶段的操作脚本和组件之间传递已修改的属性值。

Application Services 预定义属性

Application Services 提供某些常用属性作为预定义的属性。这些属性适用于群集节点。

通常，大多数服务都需要正在其中运行的虚拟机的 IP 地址。对于群集节点，服务可能需要群集中所有虚拟机的 IP 地址。因此，可将虚拟机 IP 地址指定为预定义的属性。

预定义 IP 地址属性

节点可以包含多个网卡，每个网卡分配有已部署虚拟机中的一个 IP 地址。

以下 IP 地址显示在 *节点名称*:NIC*x*_ip 属性中，其中 -*x*- 是网卡编号。

在群集化的 Dukes Bank 应用程序示例中，负载平衡器节点属性显示为 load_balancer:NIC0_ip 和 load_balancer:NIC1_ip，因为该负载平衡器节点定义了两个网卡。在虚拟机中，无法保证 NIC0 和 NIC1 将分别分配到 eth0 和 eth1。网卡是网络接口蓝图中的逻辑名称。这些网卡会映射到逻辑网络，而逻辑网络会映射到特定云网络。属性 NIC0_ip 会如蓝图中的定义返回已分配到虚拟机的 IP 地址，而不是虚拟机中的 eth0 IP 地址。

还提供 load_balancer:ip 属性。此属性可引用第一个网卡和 NIC0 的 IP 地址。如果某个组件需要群集中所有虚拟机的 IP 地址，它会使用 all(*节点名称*:NIC*x*_ip) 属性。要引用当前虚拟机的 IP 地址，可以使用 self:ip 属性。此属性对于群集节点非常有用，因为组件可能需要了解自身所在的虚拟机，而不是群集中所有虚拟机的 IP 地址。要获取群集中特定虚拟机的属性，您可以使用 all(*节点名称*:node_array_index) 属性。请参见第 98 页，“预定义的节点数组索引属性”。

预定义的节点数组索引属性

对于群集节点，名为 node_array_index 的特殊属性会标识当前虚拟机在群集中的位置。

群集中第一个虚拟机的 node_array_index 属性值为 0，群集中第二个虚拟机的 node_array_index 属性值为 1，以此类推。

例如，在具有群集节点的部署中，如果第一个虚拟机为主虚拟机，它会将群集中的其他虚拟机作为辅助虚拟机进行管理。该主虚拟机必须以不同的方式进行配置。例如，如果组件的其中一个属性为 myPosition，则必须绑定到预定义的 self:node_array_index 属性。组件脚本可以验证 myPosition 属性的值是否为 0，如果是，则会执行其他配置。

对于群集节点，预定义的节点数组属性为 all(*NodeName*:node_array_index)。该属性提供了群集中所有虚拟机的索引，可用于确定群集大小。

预定义的磁盘布局信息属性

可变磁盘增强了存储的灵活性，并允许您将其他磁盘添加到节点中。

必须在服务定义中创建可绑定的属性并绑定到 disk_layout_info 属性，才能使用其他磁盘信息。disk_layout_info 属性中包含添加到应用程序蓝图的所有磁盘的信息。

例如，在任意应用程序蓝图中，磁盘信息由蓝图捕获为 Disk0、Disk1、Disk2，以此类推。

您可以获得以下信息：

- name
- sizeInGb
- metaTags
- seq
- mountPoint
- fileSystem

安全组件属性

这些属性用于配置部署的组件。在某些情况下，这些属性可用于存储敏感数据（如密码）。

例如，WAR 组件可能需要存储敏感数据（如密码）才能访问数据库。可以将这些属性标记为安全。安全属性值会被屏蔽，并在 Application Services 中显示为星号。

如果将安全属性更改为非安全属性，为了安全起见，Application Services 会重置属性值。必须为此属性设置新值。

重要事项 如果在脚本中使用 `echo` 命令或其他类似命令打印安全属性，则这些值会以纯文本的形式显示在日志文件中。日志文件中的这些值不会被屏蔽。

所需的组件属性

您可以在需要时设置属性。例如，如果在部署过程中没有为依赖属性的脚本定义这些属性，那么部署可能会失败。

例如，要运行 Apache Tomcat 服务器，Java 必不可少，并且必须设置 JAVA_HOME 属性值。

当某个属性被标记为必需时，必须至少在该属性的某一生命周期阶段内提供一个值，例如库、蓝图或部署配置文件。例如，应用程序目录管理员可以定义一个必需属性，将该属性标记为可替代的，并且在库中不为该属性设置任何值。应用程序架构师必须在蓝图中为此属性提供一个值，或者在部署配置文件中将该属性标记为可替代的。如果应用程序架构师没有为此属性设置值，那么部署者必须在部署配置文件中为此属性设置一个值，然后才能部署该应用程序。

开发组件的最佳做法

熟悉库中的示例组件，因为其中包含了一些如何定义属性和操作脚本的示例。

注意 库中的示例组件只能在测试环境中使用。

在 Application Services 中开发组件时，请遵循以下最佳做法。

- 一些安装程序可能需要访问 tty 控制台。从 `/dev/console` 重定向输入。

  例如，预定义 RabbitMQ 服务使用其安装脚本中的 `./rabbitmq_rhel.py --setup-rabbitmq < /dev/console` 命令。

- 如果组件使用多个生命周期阶段，可以在 INSTALL 生命周期阶段中更改属性值。新值将发送到下一个生命周期阶段。操作脚本可以在部署过程中计算属性值，以便将该值提供给其他从属脚本。

  注意 对于使用多个生命周期阶段的组件，无法更改其内容属性值。

  例如，在群集化的 Dukes Bank 示例应用程序中，JBossAppServer 服务将在安装生命周期阶段的过程中计算 JVM_ROUTE 属性。JBossAppServer 服务使用此属性配置生命周期。然后，Apache 负载平衡器服务会将其 JVM_ROUTE 属性绑定到 all(appserver:JbossAppServer:JVM_ROUTE) 属性以获取 node0 和 node1 的最终计算值。

  如果某一组件需要使用另一个组件的属性值才能成功完成应用程序部署，则必须在应用程序蓝图中声明这些明确的依赖关系。

- 为了使脚本无中断地运行，必须将返回值设置为零 (0)。

  此设置允许代理捕获所有属性并将其发送到 Application Services 服务器。

管理 Application Services 库 13

使用标准逻辑模板、示例服务、任务脚本、操作系统、外部服务、策略和标记的预填充可扩展库，应用程序架构师可以快速创建多层企业应用程序的蓝图。云管理员可以为部署者启用关于部署环境的预定义策略定义，以便在部署过程中将其应用于多层应用程序。

重要事项 Application Services 包含多个预定义逻辑模板、服务和外部服务。应用程序目录管理员可以使用或修改这些预定义模板和服务，如此一来便无需自己进行创建。最佳做法是创建服务副本以保留原始服务供日后参考。

库中的预定义服务、任务、操作系统和标记可用于 Application Services 中的所有用户组。对于默认组以外的每个组，则必须添加逻辑模板。

可以使用 Application Services 库创建自定义任务，这些任务是可以添加到部署配置文件中的执行计划的自定义脚本。

熟悉有关管理库的主要概念。请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 101 页，“将操作系统添加到库”
- 第 102 页，“将标记添加到库”
- 第 103 页，“将服务添加到库”
- 第 108 页，“将 Puppet 服务导入库”
- 第 109 页，“将外部服务添加到库”
- 第 115 页，“将逻辑模板添加到库”
- 第 118 页，“将策略添加到库”
- 第 122 页，“将自定义任务添加到库”

将操作系统添加到库

您可能需要将操作系统添加到 Application Services 所包含的预安装操作系统列表中。

您可以在库中创建操作系统定义，并指定逻辑模板所使用的操作系统或特定服务所支持的操作系统。Application Services 支持 Linux 和 Windows 操作系统。

注意 无法删除库中的操作系统。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。

- 熟悉“操作系统”页面上的预安装操作系统。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 操作系统**。

2 单击**新建**。

3 设置操作系统的名称和描述。

最佳做法是使用操作系统名称和版本号，并将位信息添加到描述中。

4 从下拉菜单中选择架构。

选项	描述
x86_32	用于 32 位操作系统。
i386	用于 32 位操作系统。
x86_64	用于 64 位操作系统。

5 从下拉菜单中选择操作系统系列，然后单击**保存**。

在为服务创建操作脚本时，根据指定操作系统系列确定脚本类型。针对基于 Linux 的应用程序，可以使用 Bash 或 BeanShell 脚本，而针对基于 Windows 的应用程序，则可以使用 Windows CMD、PowerShell 或 BeanShell 脚本。

6 单击**创建操作系统版本**创建操作系统版本。

您可以为一个操作系统创建多个版本。

7 填写操作系统版本信息并单击**保存**。

操作系统名称保持不变。您可以指定主要和次要版本，并添加操作系统的服务包详细信息。

将新操作系统添加到“操作系统”页面。在创建其他组件以添加到库或应用程序蓝图时，您可以选择该操作系统。

将标记添加到库

创建逻辑模板、服务或外部服务时，您可以将一个或多个标记与组件关联。

注意 无法从库中删除标记。

标记类型不可互换。例如，将标记类型从服务器类型更改为属性描述符可能会导致蓝图画布中的渲染问题，或在您尝试使用服务时生成错误消息。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉“标记”页面上列出的预安装标记。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 标记**。

2 单击**新建**。

3 在文本框中输入标记的名称和描述。

4 从下拉菜单中选择标记类型。

属性描述符标记类型显示为服务属性的**自动绑定标记**下拉菜单中的选项。创建服务或逻辑模板时，服务器类型标记显示为下拉菜单中的选项。

5 单击**保存**。

新创建的标记在“标记”页面上列出。将服务、任务或逻辑模板添加到库或应用程序蓝图时，可以选择标记。

将服务添加到库

服务由用于安装、配置、启动和更新应用程序所需的软件服务的脚本组成。可以将自定义服务添加到 Application Services 库。

步骤

1 在库中创建服务版本第 103 页，

使用 Application Services 库创建软件服务版本。Application Services 库还包含预定义服务。

2 定义服务版本属性第 104 页，

Application Services 将定义的属性作为环境变量传递到在虚拟机中运行的脚本。服务版本属性可定义服务的脚本中使用的变量。

3 将操作脚本添加到服务版本中第 106 页，

应用程序目录管理员必须为至少一个生命周期阶段提供操作脚本。为了部署应用程序，您可以创建安装、配置、启动、回滚和卸载脚本，还可以创建更新脚本以更新现有部署。将这些脚本自定义为使用组件属性。

4 维护服务版本第 107 页，

您可以从现有的服务版本复制或编辑服务。

5 从库中删除服务第 108 页，

可从服务页面中删除服务。

在库中创建服务版本

使用 Application Services 库创建软件服务版本。Application Services 库还包含预定义服务。

有时，与其创建新服务，您可能更愿意编辑预定义服务中的脚本和变量。最佳做法是创建预定义服务的副本，然后再进行更改。请参见第 107 页，“维护服务版本”。

可以在组中公开共享或专享服务。在组件级别支持共享，但不支持在各个版本级别共享。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否必须向操作系统和描述性标记列表中添加项目。请参见第 101 页，“将操作系统添加到库”和第 102 页，“将标记添加到库”。
- 如果计划添加启用了 SELinux 的操作系统，请验证正在安装的应用程序是否启用了可允许模式或特定异常适合正在安装的应用程序，以避免任何安装失败。
- 如果某项服务在蓝图中使用，或作为预安装服务包含在逻辑模板中，则无法删除该服务。删除服务之前，请先删除包含预安装服务的引用蓝图或逻辑模板。
- 要查看有关如何命名或描述服务的示例，请选择**库 > 服务**查看预定义服务。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 服务**。

2 单击**新建**。

3 设置服务名称，添加描述，然后单击**保存**。

使用与软件组件（与操作脚本相关）对应的名称，并附加限定符以便跟踪。例如，如果在特定操作系统上使用该服务，则可以将服务名称设置为 *ServiceName*_Windows 或 *ServiceName*_Linux。

4 单击**创建服务版本**创建服务版本。

可以为服务创建多个版本。

此时将打开用于创建服务版本的页面。

5 完成服务版本信息。

服务版本名称保持不变。

a 指定主要、次要或微发布版本，以及版本是否具有限定符。

例如，可以使用 1.0、1.5 或 1.0.1-Linux 等版本号。

b （可选）在“描述”部分，如果要为特定的操作系统创建服务版本，请描述使用的操作系统版本和任何需要的适用配置。

6 为应用程序创建部署蓝图时，设置标记以对显示的服务列表进行组织

注意 只有指定为“服务器类型”的标记才会显示在下拉菜单中。

可以添加多个标记。

7 要创建列表中未出现的标记，请单击**取消**，然后选择**库 > 标记**。

8 （可选）在“支持的操作系统”部分，如果该服务中使用的脚本只能在特定操作系统中运行，请在此处选择这些操作系统。

在蓝图编辑器中，Application Services 会阻止将服务添加到模板中，除非模板包含其中一个操作系统 。如果该服务可以在任何操作系统中使用，请将此字段留空。

9 要使用列表中未出现的操作系统名称，请单击**取消**，然后选择**库 > 操作系统**创建操作系统名称。

10 （可选）在“支持的组件”部分中，如果该服务中只能运行特定类型的应用程序组件，请在此处指定这些组件。

例如，vFabric tc Server 实例中只能运行 WAR 和 JAR 组件。数据库服务器中只能运行 SQL 脚本。在应用程序蓝图中，选择的组件限制了可以向该服务添加哪些应用程序组件。如果可以向该服务添加组件，请将此字段留空。

11 创建或编辑逻辑模板时，选中**在模板中预安装**复选框以在“包含的服务”部分中列出该服务。

选中该复选框指示该服务已安装在模板中。

如果没有选中该复选框，则该服务将显示在为应用程序创建部署蓝图时可包含的服务列表中。

应用程序蓝图的“操作系统模板”部分中还显示包含预安装服务的逻辑模板，这些逻辑模板可作为模板的一部分使用。

下一步

配置服务版本属性。请参见第 104 页，“定义服务版本属性”。

定义服务版本属性

Application Services 将定义的属性作为环境变量传递到在虚拟机中运行的脚本。服务版本属性可定义服务的脚本中使用的变量。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 要查看关于如何定义服务的示例，请选择**库 > 服务**查看预定义的服务。

- 验证服务版本在 Application Services 中是否可用。请参见第 103 页，“在库中创建服务版本”。

步骤

1 打开现有服务版本，然后单击**编辑**。

2 针对“属性名称”和“描述”，定义一个变量并添加可选描述。

属性名称不能以数字开头。

添加并定义变量后，您可以创建服务版本的安装、配置、启动、更新、回滚或卸载脚本。

3 要更改类型，请从下拉菜单中选择一种属性类型。

无法将类型添加到菜单。

4 在脚本运行时，输入值来替代此属性。

例如，对于名为 http_port 的属性，可以在该字段中键入 **`80`**。

注意 对于 vCloud Director 或 vRealize Automation，如果向 http_proxy、https_proxy 或 ftp_proxy 属性（其服务脚本将 `darwin_global.conf` 文件用作文件源）添加一个值，则脚本运行时，这些属性会在已部署的应用程序中替代任何现有的代理信息。Amazon EC2 不需要使用代理来部署应用程序。

如果将属性类型设置为“计算”，则无法键入值。

如果选中**必需**复选框而未选中**在蓝图中可替代**复选框，则该属性必须具有一个值。

5 从“自动绑定类型”下拉菜单中，选择自动绑定类型。

自动绑定类型描述分配给某个属性以用于在蓝图画布中自动绑定的角色。

选项	描述
无	该属性不会自动绑定。
占用	该属性值设置了自动绑定。
公开	该属性可用于自定义使用自动绑定的其他属性值。

注意 属性不能具有“占用”和“公开”角色。

“占用”或“公开”的自动绑定类型需要为该属性至少定义一个自动绑定标记。“占用”属性上的自动绑定标记必须是“公开”属性上的标记的子集。标记集不必相等。

6 定义属性类型。

定义的属性类型会影响您可以设置的自动绑定类型。不允许使用以下属性类型和自动绑定类型的组合。

属性类型	不能将自动绑定类型设置为
数组	公开
计算	占用
内容	占用或公开
单一选择	占用或公开

7 从“自动绑定标记”下拉菜单中，单击**新建**选择一个自动绑定类型，以便向属性添加属性描述符标记。

注意 只有被指定为属性描述符的标记才会显示在下拉菜单中。

您可以添加多个自动绑定标记。

例如，如果将 Apache 2.2.0 服务的 appsrv_routes 属性的自动绑定类型分配为“占用”，将 vFabric tc Server 2.1.0 服务的 JVM_ROUTE 属性的自动绑定类型分配为“公开”。appsrv_routes 属性会使用 JVM_ROUTE 属性值来对自身进行自定义。为允许自动绑定与这些服务属性进行连接，您可以将 Servlet 容器和路由标记分别添加到 appsrv_routes 和 JVM_ROUTE 属性。创建包含 vFabric tc Server 和 Apache 服务的应用程序时，如果这两个节点之间存在依赖关系，则属性会在蓝图编辑器中相互自动绑定。

8 针对部署应用程序所需的属性，选中**必需**复选框。

如果有任何所需属性值留空，则会提示您在执行应用程序部署之前填充这些属性值。

如果属性类型为“计算”，则**必需**复选框不可用。如果属性类型为“内容”，则默认情况下**必需**复选框处于选中状态，而“受保护”复选框不可用。

例如，对于使用 Apache JServ 协议 (AJP) 的 http_proxy_port 属性，必须将值设置为 **8009**。

9 选中**安全模式**复选框，以便输入您定义的密码或隐藏其他属性的值。

如果属性从安全模式变为非安全模式，Application Services 出于安全考虑会重置属性值。您必须为该属性设置新的密码值。

例如，示例 MySQL 服务中的 db_password 属性受保护。

10 选中**在蓝图中可替代**复选框以允许用户（例如应用程序架构师）替代应用程序蓝图中的属性值。

如果属性类型为“计算”，则**在蓝图中可替代**复选框不可用。

如果某属性不可在蓝图中替代，则无法将自动绑定类型设置为“占用”。

例如，应用程序目录管理员可以将 vFabric tc Server 服务配置为具有 512MB 大小的 JVM 堆。但是对于大型部署，应用程序架构师可将此设置更改为 1024MB。

11 单击**删除**以从“属性”部分中移除所选行。

12 （可选）单击**重置**以恢复为原始属性值。

13 单击**上载**以将属性值导入选定的 CSV 文件。

14 保存您的服务属性定义。

下一步

创建该服务版本的操作脚本。请参见第 106 页，“将操作脚本添加到服务版本中”。

将操作脚本添加到服务版本中

应用程序目录管理员必须为至少一个生命周期阶段提供操作脚本。为了部署应用程序，您可以创建安装、配置、启动、回滚和卸载脚本，还可以创建更新脚本以更新现有部署。将这些脚本自定义为使用组件属性。

无需为所有生命周期阶段添加脚本。如果不需要某个特定阶段，可将其忽略。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 如果计划使用从外部网站下载软件的脚本，请验证用于部署应用程序的虚拟机是否具有对外部网络的访问权限。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 要查看关于如何向服务添加脚本的示例，请选择**库 > 服务**查看预定义的服务。

- 验证是否为服务版本定义了属性。请参见第 104 页，“定义服务版本属性”。

步骤

1 打开现有服务版本，然后单击**编辑**。

2 从下拉菜单中，选择操作脚本的脚本类型。

   您可以使用 Bash 或 BeanShell 脚本为基于 Linux 的应用程序编写脚本，还可以使用 Windows CMD、PowerShell 或 BeanShell 脚本为基于 Windows 的应用程序编写脚本。

3 在“脚本”列中，单击超链接以打开“编辑脚本”对话框。

   您可以写入脚本或将脚本复制到对话框中。

   应用程序目录管理员可以将服务的安装和配置参数化。为服务定义的属性可以在脚本内部使用。

4 在“脚本类型”字段中单击向下箭头，为生命周期阶段选择操作脚本类型。

5 要插入定义的属性，请在“选择属性”中单击向下箭头以插入列表。

6 （可选）

   应用程序部署期间，请在脚本成功运行后选中**重新引导**复选框以重新启动虚拟机。

7 单击**确定**。

8 在“重新引导”列中选中复选框，以便代理引导程序可以在操作脚本成功运行后重新启动虚拟机。

   重新启动虚拟机后，代理将继续进入服务版本中定义的下一个生命周期阶段。

9 单击**重置**清除脚本。

   此操作不会移除生命周期阶段中的行。

10 创建完服务版本后，单击**保存**。

11 单击**确定**。

将您创建的服务添加到页面中。

下一步

创建现有服务版本的副本或编辑服务。请参见第 107 页，“维护服务版本”。

维护服务版本

您可以从现有的服务版本复制或编辑服务。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 如果计划编辑添加到应用程序蓝图的服务中包含的属性和脚本，请先熟悉第 103 页，“将服务添加到库”中介绍的任务。
- 如果计划使用从外部网站下载软件的脚本，请验证用于部署应用程序的虚拟机是否具有对外部网络的访问权限。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 创建服务的副本以保留原始版本以供将来参考，然后编辑复制的版本。
- 您必须是拥有服务版本的业务组中的成员。

  如果您不属于拥有服务版本的业务组，则**复制**和**编辑**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 服务**。

2 复制现有的服务版本。

   a 打开一个服务版本，然后在工具栏中单击**复制**。

   复制服务时，会保留所有属性、操作脚本、操行系统、描述性标记的列表以及受支持的应用程序组件。您无法更改现有服务版本的名称。

   b 更改版本号，更新描述，然后单击**保存**。

   版本号和描述可将新的服务和父版本区分开来。

3 要修改变量或更新操作脚本，请在工具栏中单击**编辑**，进行更改，然后单击**保存**。

下一步

对于不用于构建应用程序蓝图模型的服务，可将其删除。请参见第 108 页，“从库中删除服务”。

从库中删除服务

可从服务页面中删除服务。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证此服务未在蓝图中使用，或此服务未作为预安装服务包含在逻辑模板中。

  删除包含预安装服务的引用蓝图或逻辑模板。
- 您必须是拥有服务版本的业务组中的成员。

  如果您不属于拥有相应服务版本的业务组，则**删除**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 服务**。

2 打开服务。

3 选择**删除**以移除现有的服务版本，并确认您的选择。

4 单击服务名称旁边的箭头返回到服务页面。

5 删除服务并确认您的选择。

将 Puppet 服务导入库

Puppet 服务可用于构建应用程序蓝图。必须将 Puppet 模块和定义的资源作为服务导入 Application Services 库。

您可以同时导入多个 Puppet 模块。

导入 Puppet 模块后，这些服务将具有名为“Puppet 服务”的特殊标记。新导入的服务没有操作脚本。Application Services 会将其生命周期活动委派到相应部署环境中已注册的 Puppet Master 实例。

无法更新已导入的 Puppet 内容。您可以移除该服务或更改现有的 Puppet 服务版本，然后将更新的 Puppet 内容导入库。对于缺少版本信息的 Puppet 模块，默认版本为 1.0.0 - 未知。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 启动 Application Services CLI。请参见第 197 页，“远程启动 CLI”。

- 用于运行 CLI 的远程计算机必须安装有 Puppet 模块并具有可用的 Puppet CLI。

  也可以使用 Puppet Master 计算机在两台不同的计算机上维护 Puppet 模块。

- 验证您要导入的 Puppet 服务在 Puppet Master 上是否可用。

步骤

◆ 登录 CLI 程序后，在 `roo` shell 中输入以下命令以导入 Puppet 模块，例如 Apache。

```
import-puppet-manifest --typeFilter "^apache$" --targetGroup pubs
```

此命令中的参数区分大小写。如果您使用了错误的大小写，则该命令将被忽略。

参数	描述
typeFilter	将导入类型限制为与指定的搜索表达式相匹配。 有关表达式语法的详细信息，请参见 Puppet 文档或键入 **`puppet man resource_type`** 以查看手册页。例如，要导入 mysql 模块，请键入 **`import-puppet-manifests --typeFilter "^mysql$"`**。
osfamily	指定导入类型的操作系统。选项为 WINDOWS 和 LINUX。 Linux 为默认的操作系统。
puppetPath	指定 Puppet 二进制文件的位置。
共享	指示项目为共享还是专用。 可以在命令中定义 true 或 false。默认设置 false 指示该项目为执行导入操作的用户组的专用项目。
targetGroup	为导入的清单指定业务组。这是拥有该服务的业务组。

此命令将 apache Puppet 模块导入库。

下一步

从蓝图画布的“Puppet 服务”部分将 Puppet 内容添加到新的或现有应用程序蓝图中，并部署应用程序。请参见第 126 页，“创建应用程序”和第 141 页，“设置和配置部署配置文件”。

将外部服务添加到库

在 Application Services 库中，外部服务是在应用程序部署外部安装的服务。需要配置外部服务和应用程序使它们相互协作。

步骤

1 在库中创建外部服务第 110 页，

   借助 Application Services，您可以创建基本外部服务或高级外部服务。

2 定义外部服务属性第 111 页，

   Application Services 将定义的属性作为环境变量传递到在虚拟机中运行的脚本。外部服务版本属性定义了在外部服务脚本中使用的变量。

3 为高级外部服务版本创建提供程序规范第 113 页，

   高级外部服务需要您创建提供程序规范。此规范是一个或多个脚本的集合，可定义如何在各种服务提供程序环境（如 VMware Data Director 或 Amazon RDS）中配置、更新、回滚和卸载外部服务。提供程序规范还包含可定义以连接到提供程序及提供程序特定参数的属性。

4 维护外部服务版本第 114 页，

   可以从现有版本中复制基本或高级外部服务，或修改此服务。

5 删除外部服务第 115 页，

要管理您的外部服务，可以删除重复或过期的服务。

在库中创建外部服务

借助 Application Services，您可以创建基本外部服务或高级外部服务。

基本外部服务不提供对多个服务提供程序的支持。您必须定义逻辑模板，并可以选择为 CONFIGURE、UPDATE、ROLLBACK 和 TEARDOWN 生命周期阶段添加脚本。某些基本外部服务的示例包括安装了应用程序架构的现有数据库、SAAS 应用程序（如 Workday）或单点登录 (SSO) 服务器。

高级外部服务包括一些提供程序规范的选项，这些选项可添加对外部服务的多个提供程序的支持。在提供程序规范中定义高级外部服务属性值后，这些值会在将外部服务实例映射到部署环境时填充在该实例中。例如，可以通过 VMWare Data Director 或 Amazon RDS 置备和配置外部 Oracle 数据库。应用程序中的外部负载平衡器可以通过 F5 负载平衡器实例或软件负载平衡器（如 Apache 服务器）实现。

您可以公开共享外部服务或在组中专享外部服务。在组件级别支持共享，但不支持在各个版本级别共享。共享外部服务时，不会共享外部服务实例配置文件。每个组均必须在其部署环境中定义外部服务实例配置文件。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否必须向操作系统和描述性标记列表中添加项目。请参见第 101 页，“将操作系统添加到库”和第 102 页，“将标记添加到库”。
- 要查看关于如何命名或描述外部服务的示例，请选择**库 > 外部服务**查看预定义的外部服务。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 外部服务**。

2 单击**新建**。

3 设置外部服务的名称，添加描述，然后单击**保存**。

使用与您使用的服务器或数据库相对应的名称

4 单击**创建外部服务版本**可创建外部服务版本。

可以为外部服务创建多个版本。

5 在对话框中，选择**基本**或**高级**，然后单击**确定**。

6 填写外部服务版本信息。

外部服务版本名称应保持一致。

a 指定主要、次要或微发布版本，以及版本是否具有限定符。

例如，可以使用版本号，如 1.0、1.5 或 1.0.1-Dev。

b （可选）在“描述”部分，如果要为特定的数据库创建外部服务版本，请描述使用的数据库版本和任何适用的所需配置。

7 单击**新建**可从下拉菜单中设置标记。

标记会对外部服务进行分组，为应用程序创建部署蓝图时，可以在其对应的组中查看这些服务。

可以添加多个标记。

8 要创建列表中未出现的标记，请单击**取消**，然后选择**库 > 标记**。

9 （可选）在“支持的组件”部分中，如果在此高级外部服务中只能运行特定类型的应用程序组件，请单击**新建**在此处指定这些组件。

例如，在数据库服务器中只能运行 SQL 脚本。选择的组件限制了可在应用程序蓝图中向此高级外部服务添加哪些应用程序组件。 如果可以向高级外部服务添加组件，请将此字段留空。

10 为外部服务指定操作系统。

a 对于基本外部服务，请在“逻辑模板”部分的下拉菜单中选择基于 Windows 或 Linux 的模板。

系统将填充与该逻辑模板关联的受支持操作系统。

b 对于高级外部服务，如果只能在特定的操作系统中运行在高级外部服务中使用的脚本，请单击**新建**在“支持的操作系统”部分中添加这些操作系统。

在蓝图编辑器中，Application Services 会阻止将外部服务添加到模板中，除非模板包含其中一个操作系统。如果可以在任意操作系统中使用高级外部服务，请将此字段留空。

11 要使用列表中未出现的操作系统名称，请单击**取消**，然后选择**库 > 操作系统**创建操作系统名称。

下一步

配置外部服务版本属性。请参见第 111 页，“定义外部服务属性”。

定义外部服务属性

Application Services 将定义的属性作为环境变量传递到在虚拟机中运行的脚本。外部服务版本属性定义了在外部服务脚本中使用的变量。

如果使用预配置或自定义的外部服务部署应用程序，则无法在部署过程中编辑该外部服务的属性定义或提供程序规范属性。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 要查看关于如何定义外部服务的示例，请选择**库 > 外部服务**查看预定义的服务。
- 验证外部服务版本在 Application Services 中是否可用。请参见第 110 页，“在库中创建外部服务”。

步骤

1 打开现有外部服务版本并单击**编辑**。

2 针对“属性名称”和“描述”，定义一个变量并添加可选描述。

属性名称不能以数字开头。

3 要更改类型，请从下拉菜单中选择一种属性类型。

无法将类型添加到菜单。

4 在脚本运行时，输入值来替代此属性。

例如，对于名为 http_port 的属性，可以在该字段中键入 **`80`**。

注意 对于 vCloud Director 或 vRealize Automation，如果向 http_proxy、https_proxy 或 ftp_proxy 属性（其服务脚本将 `darwin_global.conf` 文件用作文件源）添加一个值，则脚本运行时，这些属性会在已部署的应用程序中替代任何现有的代理信息。Amazon EC2 不需要使用代理来部署应用程序。

如果将属性类型设置为“计算”，则无法键入值。

如果选中**必需**复选框而未选中**在蓝图中可替代**复选框，则该属性必须具有一个值。

5 从“自动绑定标记”下拉菜单中，单击**新建**选择一个自动绑定类型，以便向属性添加属性描述符标记。

注意 只有被指定为属性描述符的标记才会显示在下拉菜单中。

您可以添加多个自动绑定标记。

例如，如果将 Apache 2.2.0 服务的 appsrv_routes 属性的自动绑定类型分配为“占用”，将 vFabric tc Server 2.1.0 服务的 JVM_ROUTE 属性的自动绑定类型分配为“公开”。appsrv_routes 属性会使用 JVM_ROUTE 属性值来对自身进行自定义。为允许自动绑定与这些服务属性进行连接，您可以将 Servlet 容器和路由标记分别添加到 appsrv_routes 和 JVM_ROUTE 属性。创建包含 vFabric tc Server 和 Apache 服务的应用程序时，如果这两个节点之间存在依赖关系，则属性会在蓝图编辑器中相互自动绑定。

6 从“自动绑定类型”下拉菜单中，选择自动绑定类型。

自动绑定类型描述分配给某个属性以用于在蓝图画布中自动绑定的角色。

选项	描述
无	该属性不会自动绑定。
占用	该属性值设置了自动绑定。
公开	该属性可用于自定义使用自动绑定的其他属性值。

注意 属性不能具有“占用”和“公开”角色。

“占用”或“公开”的自动绑定类型需要为该属性至少定义一个自动绑定标记。“占用”属性上的自动绑定标记必须是“公开”属性上的标记的子集。标记集不必相等。

7 针对部署应用程序所需的属性，选中**必需**复选框。

如果有任何所需属性值留空，则会提示您在执行应用程序部署之前填充这些属性值。

如果属性类型为“计算”，则**必需**复选框不可用。如果属性类型为“内容”，则默认情况下**必需**复选框处于选中状态，而“受保护”复选框不可用。

例如，对于使用 Apache JServ 协议 (AJP) 的 http_proxy_port 属性，必须将值设置为 **8009**。

8 选中**安全模式**复选框，以便输入您定义的密码或隐藏其他属性的值。

如果属性从安全模式变为非安全模式，Application Services 出于安全考虑会重置属性值。必须为此属性设置新值。

9 从“替代”下拉菜单中，选择一个属性替代选项。

选项	描述
蓝图	应用程序蓝图中的属性可替代。 将外部服务实例注册到部署环境时，云管理员无法查看该属性。
部署环境	云管理员可以替代部署环境中的属性。 该属性显示在应用程序蓝图画布中，并且用户可以自动绑定到某个属性。应用程序蓝图中的属性值不可编辑。
不可替代	属性不可替代。

10 选中**在蓝图中可替代**复选框以允许用户（例如应用程序架构师）替代应用程序蓝图中的属性值。

如果属性类型为“计算”，则**在蓝图中可替代**复选框不可用。

如果某属性不可在蓝图中替代，则无法将自动绑定类型设置为“占用”。

例如，应用程序目录管理员可以将 vFabric tc Server 服务配置为具有 512MB 大小的 JVM 堆。但是对于大型部署，应用程序架构师可将此设置更改为 1024MB。

11 单击**删除**以从“属性”部分中移除所选行。

12 （可选）单击**重置**以恢复为原始属性值。

13 单击**上载**以将属性值导入选定的 CSV 文件。

14 保存外部服务属性定义。

下一步

指定外部服务定义对部署没有影响，除非在部署环境中创建外部服务实例以在部署环境下的所有部署中启用外部服务定义。请参见第 57 页，“映射外部服务实例”。

为自定义外部服务版本创建提供程序规范。请参见第 113 页，“为高级外部服务版本创建提供程序规范”。

为高级外部服务版本创建提供程序规范

高级外部服务需要您创建提供程序规范。此规范是一个或多个脚本的集合，可定义如何在各种服务提供程序环境（如 VMware Data Director 或 Amazon RDS）中配置、更新、回滚和卸载外部服务。提供程序规范还包含可定义以连接到提供程序及提供程序特定参数的属性。

可以在**库 > 外部服务**页面上查看有关如何为外部服务创建提供程序规范的示例。

只能更改外部服务的组成员资格设置。然后新的成员资格设置会传播到外部服务的服务提供程序规范。

用户无法单独更新服务提供程序规范的组成员资格。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 如果计划使用从外部网站下载软件的脚本，请验证用于部署应用程序的虚拟机是否具有对外部网络的访问权限。
- 验证是否已为外部服务版本定义属性。请参见 Application Services。请参见第 110 页，“在库中创建外部服务”。

步骤

1 打开现有服务版本，然后单击**创建提供程序规范版本**。

2 键入提供程序规范名称，然后选择**创建版本**。

3 指定主要、次要或微发布版本，以及版本是否具有限定符。

   提供程序规范名称保持不变。

4 （可选）在“描述”部分，如果要为特定配置创建提供程序规范，请描述使用的配置和适用要求。

5 在“逻辑模板”部分，如果将支持的操作系统设为基于 Windows 或基于 Linux 的操作系统，则下拉菜单中将会提供相应的逻辑模板。

6 添加适用属性。

7 从下拉菜单中，选择操作脚本的脚本类型。

   您可以使用 Bash 或 BeanShell 脚本为基于 Linux 的应用程序编写脚本，还可以使用 Windows CMD、PowerShell 或 BeanShell 脚本为基于 Windows 的应用程序编写脚本。

8 在“脚本”列中，单击超链接以打开“编辑脚本”对话框。

   您可以写入脚本或将脚本复制到对话框中。

   应用程序目录管理员可以将服务的安装和配置参数化。为服务定义的属性可以在脚本内部使用。

9 在“脚本类型”字段中单击向下箭头，为生命周期阶段选择操作脚本类型。

10 要插入定义的属性，请在“选择属性”中单击向下箭头以插入列表。

11 （可选）

应用程序部署期间，请在脚本成功运行后选中**重新引导**复选框以重新启动虚拟机。

12 单击**确定**。

13 在“重新引导”列中选中复选框，以便代理引导程序可以在操作脚本成功运行后重新启动虚拟机。

重新启动虚拟机后，代理将继续进入服务版本中定义的下一个生命周期阶段。

14 单击**重置**清除脚本。

此操作不会移除生命周期阶段中的行。

15 创建完提供程序规范版本后，单击**保存**。

您创建的提供程序规范版本已添加到高级外部服务中。

下一步

指定外部服务定义对部署没有影响，除非在部署环境中创建外部服务实例以在部署环境下的所有部署中启用外部服务定义。请参见第 57 页，“映射外部服务实例”。

向您的应用程序蓝图添加外部服务。请参见第 132 页，“将外部服务添加到高级蓝图”。

更新外部服务或复制现有外部服务。请参见第 114 页，“维护外部服务版本”。

维护外部服务版本

可以从现有版本中复制基本或高级外部服务，或修改此服务。

可以从**库 > 外部服务**页面查看有关如何定义外部服务的示例。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证外部服务版本在 Application Services 中是否可用。请参见第 110 页，“在库中创建外部服务”。
- 创建外部服务的副本以保留原始版本以供将来参考，然后编辑复制的版本。
- 您必须是外部服务版本所属业务组的成员。

  如果您不属于外部服务版本所属业务组，则**复制**和**编辑**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 外部服务**。

2 复制现有的外部服务版本。

   a 打开一个外部服务版本，然后在工具栏中单击**复制**。

   复制外部服务时，会保留所有属性、操行系统、描述性标记的列表、受支持的应用程序组件以及提供程序规范脚本和属性。您无法更改现有外部服务版本的名称。

   b 更改版本号，更新描述，然后单击**保存**。

   版本号和描述可将新的外部服务和父版本区分开来。

3 修改现有属性。

   a 单击工具栏中的**编辑**。

   b 进行更改，然后单击**保存**。

4 更新提供程序规范。

   a 打开提供程序规范版本。

   b 单击工具栏中的**编辑**。

   c 进行更改，然后单击**保存**。

5 复制现有的提供程序规范。

   a 打开提供程序规范版本。

   b 单击工具栏中的**复制**。

   c 更改版本号，更新描述，然后单击**保存**。

   版本号和描述可将新的提供程序规范和父版本区分开来。

下一步

删除在应用程序蓝图中不使用的外部服务的旧版本。请参见第 108 页，“从库中删除服务”。

删除外部服务

要管理您的外部服务，可以删除重复或过期的服务。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 确认蓝图中未使用外部服务。请参见第 132 页，“将外部服务添加到高级蓝图”。
- 验证您是否为拥有外部服务版本的业务组中的成员。

  如果您不属于拥有外部服务版本的组，则**删除**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 外部服务**。

2 打开现有外部服务。

3 单击**删除**以移除现有的服务版本，并确认您的选择。

4 单击外部服务名称旁的箭头以返回“外部服务”页面。

5 删除外部服务并确认您的选择。

将逻辑模板添加到库

对应用程序蓝图进行建模时，可以向自定义或示例逻辑模板中添加兼容服务和脚本。可以将逻辑模板映射到 vCloud Director、vRealize Automation 或 Amazon EC2 中的实际云模板。逻辑模板允许应用程序蓝图保持云的不可知状态。

作为逻辑模板定义的一部分，您可以介绍哪些服务已与操作系统一起安装在模板中。通常，在 IT 组织中，模板中会安装一些性能监控代理或病毒扫描程序。并且，例如，vFabric tc Server 可能已预安装在逻辑模板中以加速部署。如果部署逻辑模板时始终使用某个特定服务，则可以预安装该服务以避免每次部署时都要添加该服务。

您可以公开共享逻辑模板或在组中专享逻辑模板。在组件级别支持共享，但不支持在各个版本级别共享。

时常保存更改。如果 Application Services 用户界面会话不活动，将在 30 分钟后过期。如果会话过期，未保存的所有更改将会丢失。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。

- 在 vCloud Director、vRealize Automation 或 Amazon EC2 中创建满足在 Application Services 中正常运行要求的云模板。请参见第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”、第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”或第 83 页，“创建 Amazon EC2 自定义模板的虚拟机要求”。
- 验证 Application Services 中是否至少注册了一个云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”、第 55 页，“注册 vRealize Automation 云提供程序和模板”或第 85 页，“注册 Amazon EC2 云提供程序和模板”。
- 如果应用程序需要从企业防火墙外部访问 URL，请将适用的服务和应用程序组件配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 验证是否必须向操作系统和描述性标记列表中添加项目。请参见第 101 页，“将操作系统添加到库”和第 102 页，“将标记添加到库”。
- 如果计划在模板中预安装应用程序或服务，请在将其添加到模板之前创建和配置服务。请参见第 103 页，“将服务添加到库”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 逻辑模板**。

2 单击**新建**。

3 设置逻辑模板的名称，添加描述，并单击**保存**。

要跟踪所使用的云模板或操作系统，请包含云模板或操作系统的名称。

4 单击**创建 LT 版本**创建逻辑模板版本。

可以为一个逻辑模板创建多个版本。

此时会打开创建逻辑模板版本的页面。

5 填写模板版本信息。

选项	描述
名称	逻辑模板版本名称保持不变。
版本	添加限定符来指示主要和次要版本，如 **`1.2.1`** 或 **`1.0.1-CentOS56-32bit`**。
描述	添加有关逻辑模板的详细信息。例如，介绍您所使用的脚本或者所需的磁盘大小。
标记	基于逻辑模板所提供的功能对逻辑模板进行分类。为应用程序创建部署蓝图时，Application Services 基于您所看到的标记组织模板。 **注意** 只有指定为“服务器类型”的标记才会显示在下拉菜单中。 可以添加多个标记。 要使用列表中没有的标记，请单击**取消**，然后选择**库 > 标记**创建标记。
支持的操作系统	指定逻辑模板中安装的操作系统。该操作系统信息在应用程序蓝图中使用以限制可以添加到该模板的服务。并非所有服务都在所有操作系统上运行。例如，如果指定 Ubuntu 操作系统，则在蓝图中使用此模板并尝试添加与 Ubuntu 不兼容的服务时，Application Services 会阻止您添加该服务。 要使用列表中未出现的操作系统名称，请单击**取消**，然后选择**库 > 操作系统**创建操作系统名称。

6 将云模板映射到逻辑模板。

可以向一个逻辑模板版本添加多个云模板或者为不同的云选择不同的云模板。即使您使用相同的云提供程序，也可能需要在部署时从不同的云模板中选择以便进行不同的模板配置。

不保存重复的云模板、逻辑模板映射和空行。

例如，对于多个云模板，可以使用相同的逻辑模板。如果部署到生产环境，则可以选择具有大量磁盘空间的云模板。对于测试或转储环境，可以选择具有少量磁盘空间的云模板。

a 在“云提供程序名称”列，单击向下箭头选择云提供程序。

此时将会显示创建的云提供程序列表。

b 在“云模板”列中，单击向下箭头指定云模板。

与用户属于相同组的云模板在下拉菜单中显示。如果云模板的列表为空，则说明现有云模板不属于您的组，或者云模板未注册。

7 （可选）要将多个云模板映射到一个逻辑模板，请重复执行步骤 6。

8 （可选）定义预安装的服务。

a 在“服务名称”列，单击向下箭头选择预安装的服务。

b 要使用列表中没有的服务，请单击**取消**，并选择**库 > 服务**创建服务及其包含的操作脚本。

如果在创建蓝图后有任何预安装服务添加到逻辑模板，则不会将新的预安装服务添加到节点。

9 （可选）将新的预安装服务添加到节点。

a 将具有预安装服务的逻辑模板添加到应用程序蓝图。

b 将服务和组件传输到新的逻辑模板。

c 重新创建任何适用的依赖关系并删除旧模板。

10 完成创建模板时，单击**保存**。

创建的逻辑模板将添加到逻辑模板页面。该模板还会显示在为应用程序创建部署蓝图时可以包括的逻辑模板列表中。

下一步

复制现有逻辑模板版本，编辑或删除模板。请参见第 117 页，“维护逻辑模板版本”。

维护逻辑模板版本

要将现有逻辑模板映射到其他云模板或者添加预安装的服务，可以编辑逻辑模板版本。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 在 vCloud Director、vRealize Automation 或 Amazon EC2 中创建满足在 Application Services 中正常运行要求的云模板。请参见第 68 页，“创建 vCloud Director 自定义模板的虚拟机要求”、第 48 页，“创建 vRealize Automation 自定义模板的虚拟机要求”和第 83 页，“创建 Amazon EC2 自定义模板的虚拟机要求”。
- 如果应用程序需要从企业防火墙外部访问 URL，请将适用的服务和应用程序组件配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 验证是否必须向操作系统和描述性标记列表中添加项目。请参见第 101 页，“将操作系统添加到库”和第 102 页，“将标记添加到库”。
- 如果计划在模板中预安装应用程序或服务，请创建并配置服务。请参见第 103 页，“将服务添加到库”。

- 验证您是否为拥有逻辑模板版本的业务组中的成员。

  如果您不属于拥有逻辑模板版本的业务组，则**复制**和**编辑**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 逻辑模板**。

2 从现有逻辑模板版本复制逻辑模板。

   a 打开逻辑模板版本，然后在工具栏中单击**复制**。

     所有的云模板、服务、操作系统和用于标识应用程序组件的描述性标记列表将保留。您无法更改现有逻辑模板版本的名称。

   b 更改版本号、更新描述，并根据需要更新云提供程序、云模板或服务。

     版本号和描述可将新的逻辑模板与父版本区分开来。

     最佳做法是创建模板副本，以便保留原始模板供日后参考。

   c 单击**保存**。

3 要更改逻辑模板的现有云提供程序、云模板或服务，请单击工具栏中的**编辑**，执行更改，然后单击**保存**。

下一步

如果有早期版本的逻辑模板，可以将其从库中删除。请参见第 118 页，“从库中删除逻辑模板”。

从库中删除逻辑模板

可以从逻辑模板页面删除逻辑模板。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证应用程序蓝图中的逻辑模板和引用该模板的所有其他对象是否已删除。
- 验证您是否为拥有逻辑模板版本的业务组中的成员。

  如果您不属于拥有逻辑模板版本的业务组，则**删除**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 逻辑模板**。

2 打开逻辑模板。

3 单击**删除**可移除现有逻辑模板版本并确认选择。

4 单击逻辑模板名称旁边的箭头可返回至“逻辑模板”页面。

5 删除逻辑模板并确认选择。

将策略添加到库

您可以创建策略定义，以便捕获评估应用程序部署合规性状态的自定义规则集。

Application Services 包含以下预定义策略定义。

- 黑名单服务 - 检查是否在蓝图节点或逻辑模板中使用了不需要的服务。
- 内存策略 - 将节点内存值与定义的最小和最大内存值进行比较。
- CPU 策略 - 将节点 CPU 计数值与指定的最小和最大内存值进行比较。
- 最大虚拟机计数策略 - 检查部署中所有节点内虚拟机实例的总数（不包括用于外部服务的临时虚拟机）。

- 强制服务 - 检查蓝图节点是否缺少指定的强制服务。

策略对其租户中的所有业务组可见，但对其他租户中的业务组不可见。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 要查看如何命名或定义策略，请选择**库 > 策略**查看预定义策略。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 策略**。

2 单击**新建**。

3 设置策略名称，添加描述，然后单击**保存**。

   使用与策略定义关联的名称，以便进行跟踪。例如，如果该策略检查特定服务在部署过程中是否始终应用于应用程序，您可以在描述中列出服务名称。

4 单击**创建策略版本**创建策略版本。

   您可以为一个策略创建多个版本。

   此时将打开创建策略版本的页面。

5 填完策略版本信息。

   策略版本名称保持不变。

   a 指定主要、次要或微发布版本，以及版本是否具有限定符。

      例如，您可以使用诸如 1.0、1.5 或 1.0.1-CPU-Policy 等版本号。

   b （可选）在“描述”部分中，描述策略中的定义。如果该策略基于前期的版本，请详细说明两个策略版本之间的差异。

6 针对“属性名称”和“描述”，定义一个变量并添加可选描述。

   属性名称不能以数字开头。

7 要更改类型，请从下拉菜单中选择“字符串”或“数组”属性类型。

   无法将类型添加到菜单。

8 在脚本运行时，键入值来替代此属性。

   例如，对于名为 max_cpu_count 的属性，您可以在此字段中键入 **10** 来指定应用程序中允许的最大 CPU 数。

9 选中**安全模式**复选框，以便输入您定义的密码或隐藏其他属性的值。

   如果属性从安全模式变为非安全模式，Application Services 出于安全考虑会重置属性值。您必须为该属性设置新的密码值。

10 单击**删除**以从“属性”部分中移除所选行。

11 （可选）单击**重置**以恢复为原始属性值。

12 完成之后，单击**确定**并保存更改。

将创建的策略添加到该页面。

下一步

添加策略定义脚本，以便评估部署的合规性状态。请参见第 120 页，“创建策略定义脚本”。

创建策略定义脚本

每个策略定义中的生命周期阶段都有一个 SCAN 脚本，以对部署的合规性状况进行评估。对部署执行操作（卸载过程除外）之前或当用户明确启动部署策略扫描时，Application Services 会先调用策略扫描操作脚本。

扫描操作脚本包括一个如 Application Services REST API 规范中所定义的部署模型。扫描操作脚本也会接收蓝图中所使用的其他组件以进行部署。

您必须在特定部署环境中创建策略实例以启用策略。如果在部署过程中发生策略冲突，则将对其进行标记并在合规性视图摘要页面中查看冲突详细信息。

注意 Java Script 是唯一支持编写策略定义脚本的语言。

策略定义中定义的策略属性将作为单独变量提供给脚本。脚本可通过声明与属性同名的变量对其进行访问。

脚本输入变量	描述	
var min_cpu_count	与 min_cpu_count 属性相对应，并将脚本的值设置为“占用”。	
eventPayload	包括进行策略合规性评估的部署的详细信息。 对于常规属性，必须添加 var eventPayload 以访问 eventPayload 对象。 策略脚本会将 eventPayload 变量视为具有以下属性的 java 对象：	
	DeploymentProfile 类型的 deploymentProfile，如 V2 API 中所定义	表示捕获最新部署详细信息的部署配置文件。更新时，该文件包括更新配置文件中的所有更改。
	Blueprint 类型的蓝图，如 V2 API 中所定义	表示从 deploymentProfile 中引用的实际蓝图对象。
	***ListLogicalTemplate* 类型的 logicalTemplates，其中 LogicalTemplate 如 V2 API 中所定义**	表示从应用程序蓝图中的各个节点引用的逻辑模板列表。
	***ListServiceVersion* 类型的 serviceVersions，其中 ServiceVersion 如 V2 API 中所定义**	表示从应用程序蓝图中的各个节点引用的服务版本列表。

预计策略脚本将输出下列属性以传递合规性评估的结果。脚本必须将其声明为变量。

脚本输出变量	描述	
complianceResult	字符串类型，必需设置的变量。如果脚本无法设置该变量，则将合规性结果视为“错误”。 变量的有效值为：	
	Compliant	指示部署符合正在评估的策略。
	Non_Compliant	指示部署违反了正在评估的策略。
	错误	指示无法生成评估结果。
complianceMessage	字符串类型。 该可选值提供了策略冲突原因的高度概括。值可以是任意字符串，但不能超过 2048 个字符。	

脚本可以通过 Java Script 中可用的标准 println 功能创建日志消息。Application Services 已捕获该日志，它对于诊断策略脚本中的错误或提供策略评估结果的详细信息十分有用。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否已至少在库中创建一个策略。请参见第 118 页，“将策略添加到库”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 策略**。

2 打开策略以添加策略定义脚本。

3 在“脚本”列中，单击超链接以打开“编辑脚本”对话框。

   您可以参考现有的预定义策略定义并在对话框中创建脚本。

4 单击**确定**。

下一步

指定策略定义不会对部署产生任何影响，除非在部署环境中创建策略实例以在该部署环境下的所有部署中启用该策略定义。请参见第 58 页，“创建策略实例”。

维护策略版本

您可以编辑策略，以添加或删除策略定义、编辑操作脚本或更改扫描参数。编辑策略定义的操作不影响从该策略创建的现有策略实例。更改只对新策略实例生效。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 要查看策略定义和操作的示例，请选择**库 > 策略**查看预定义策略。
- 验证 Application Services 库中是否有可用的策略版本。请参见第 118 页，“将策略添加到库”。
- 创建策略的副本来保留原始策略以供日后参考，然后编辑该复制的版本。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 策略**。

2 复制现有策略版本。

   a 打开策略版本，单击工具栏中的**复制**。

      复制策略时，操作脚本和所有属性都会保留。无法更改现有策略版本的名称。

   b 更改版本号，更新描述，然后单击**保存**。

      版本号和描述可将新策略与父版本区分开来。

3 修改现有策略属性和操作脚本。

   a 单击工具栏中的**编辑**。

   b 进行更改，然后单击**保存**。

下一步

删除重复或过期的策略。请参见第 122 页，“删除策略”。

删除策略

如果策略定义不适用于您的应用程序部署，可以从 Application Services 库删除相应的策略。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证该策略在部署环境中不包含现有的策略实例。请参见第 76 页，“创建 vCloud Director 部署环境”、第 56 页，“创建 vRealize Automation 部署环境”或第 86 页，“创建 Amazon EC2 部署环境”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 策略**。

2 打开现有策略。

3 删除现有的策略版本并确认您的选择。

4 单击策略名称旁的箭头以返回“策略”页面。

5 删除该策略并确认您的选择。

将自定义任务添加到库

使用 Application Services，可以创建自定义任务，从而在应用程序部署中执行自定义任务，如运行安全修补程序。

Application Services 目录包含要配置 APT 或 YUM 存储库的预定义任务、将计算机注册到 Red Hat 网络的脚本，或启用虚拟机以连接到指定 Windows Active Directory 域的脚本。可以将这些自定义任务添加到部署配置文件中的执行计划。在某些情况下，与其创建自定义任务，您可能更愿意编辑预定义任务。

时常保存更改。如果 Application Services 用户界面会话不活动，将在 30 分钟后过期。如果会话过期，未保存的所有更改将会丢失。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否必须向操作系统和描述性标记列表中添加项目。请参见第 101 页，“将操作系统添加到库”和第 102 页，“将标记添加到库”。
- 如果任务中的脚本需要从企业防火墙外部访问 URL，请配置适用的服务和应用程序组件以使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 任务**。

2 单击**新建**。

3 设置自定义任务名称，添加描述，然后单击**保存**。

   最佳做法是使用与脚本所执行任务对应的名称。

4 单击**创建任务版本**创建自定义任务版本。

   可以为自定义任务创建多个版本。

   此时将显示用于创建自定义任务版本的页面。

5 完成自定义任务版本信息。

选项	描述
名称	任务版本名称保持不变。
描述	添加有关自定义任务的详细信息。例如，描述将脚本添加到执行计划中的节点时该脚本所执行的操作。
版本	可以指定主要版本、次要版本或微版本，按需决定是否需要限定符。例如，可以使用 1.0 或 1.5 等唯一版本号。
支持的操作系统	如果该任务中使用的脚本只能在特定操作系统中运行，请在此处选择这些操作系统。在执行计划中，Application Services 会防止自定义任务显示在“目录任务名称”列表中，除非其中包含这些操作系统之一。必须至少在文本框中添加一个操作系统。 要创建列表中未出现的操作系统名称，请单击**取消**，然后选择**库 > 操作系统**。

6 在“属性”部分，定义用于自定义任务的变量。

选项	描述
属性名称	单击“属性名称”列的第一行以定义变量。例如，可以创建用于配置存储库的自定义任务，或者创建在服务或应用程序组件的部署任务成功完成后发送通知电子邮件的自定义电子邮件任务。
描述	包含有关自定义任务的详细信息。
类型	要更改类型，请从下拉菜单中选择一种类型。可用的属性类型包括字符串、内容和数组。无法将类型添加到菜单。
值	键入自定义任务运行时要替代此属性的值。例如，对于名为 remove_all 的属性，可以在此文本框中键入 `true` 或 `false`。
安全	选中您所定义的密码的复选框或要隐藏其属性值的其他属性的复选框。例如，JBoss 服务中的 JBOSS_JMX_PWD 属性是安全的。 如果属性从安全模式变为非安全模式，Application Services 出于安全考虑会重置属性值。必须为此属性设置新值。
“删除”按钮	从“属性”部分中移除选定的行。

7 在“脚本编辑器”部分，为自定义任务添加脚本。

目录管理员可以将服务的安装和配置参数化。为服务定义的属性可以在脚本内部使用。

a 展开“编辑脚本”对话框，从下拉菜单中设置脚本类型以编写脚本或复制脚本。

b 应用程序部署期间，请在脚本成功运行后选中**重新引导**复选框以重新启动虚拟机。

c 单击“选择属性”中的向下箭头，插入列表以添加自定义任务属性。

d 完成后，单击**确定**。

8 完成创建自定义任务之后，单击**保存**。

创建的自定义任务将添加到“任务”页面。

下一步

可以在执行计划中添加自定义任务，并将其部署到部署环境中。请参见第 146 页，“检查执行计划并添加自定义任务”。修改自定义任务以满足您当前的需求。请参见第 123 页，“维护自定义任务版本”。

维护自定义任务版本

您可以配置脚本或重新定义自定义任务版本的现有属性。要保留原始的自定义任务版本，您可以创建副本并修改复制的版本。

注意 如果在部署配置文件中使用自定义任务来自定义应用程序部署，则无法删除该自定义任务。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否必须向操作系统和描述性标记列表中添加项目。请参见第 101 页，“将操作系统添加到库”和第 102 页，“将标记添加到库”。
- 如果任务中的脚本需要从企业防火墙外部访问 URL，请配置适用的服务和应用程序组件以使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 任务**。

2 从现有自定义任务版本中复制自定义任务。

   a 打开自定义任务版本，然后在工具栏中单击**复制**。

   这将保留所有属性、脚本和操作系统。您无法更改现有自定义任务版本的名称。

   b 更改版本号，更新说明，并根据需要更新操作系统。

   版本号和描述将新的自定义任务和父版本区分开来。

   c 单击**保存**。

3 要更改现有属性或现有脚本，请在工具栏中单击**编辑**，进行更改，然后单击**保存**。

创建应用程序 14

Application Services 提供了一种拖放画布，应用程序架构师可以用来进行应用程序蓝图建模。使用此蓝图，可以创建用于云部署的应用程序。

可以将以下项拖至蓝图。

- 库中的逻辑模板

  可以使用 Application Services 示例模板或创建自定义模板。将包含操作系统映像以及可能预安装和配置了服务的云模板映射到 Application Services 逻辑模板。

- 库中的应用程序基础架构组件和脚本

  包含的示例服务是可重复使用的软件组件。Application Services 包含遵循服务最佳实践的安装和配置脚本。

- 应用程序组件

  这些应用程序在服务上运行。配置应用程序组件后，可以将其部署到兼容的服务或操作系统映像上。

共享

Application Services 依赖于 vRealize Automation

已登录的 Application Services 用户属于一个或多个业务组。一个或多个业务组属于单个租户。一个或多个租户

注意 有关从 Application Services 中删除应用程序部署的信息，请参见第 194 页，“从 Application Services 中删除应用程序部署”。

熟悉有关创建应用程序的主题中经常出现的主要概念。请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 126 页，“创建应用程序”
- 第 126 页，“创建应用程序版本”
- 第 127 页，“应用程序蓝图建模”
- 第 128 页，“配置应用程序蓝图”
- 第 129 页，“创建高级蓝图”
- 第 133 页，“复制应用程序版本”
- 第 133 页，“复制应用程序”
- 第 134 页，“删除应用程序版本”

创建应用程序

使用 Application Services，可对应用程序部署进行建模、创建依赖关系以及编辑应用程序配置。

您可以将应用程序指定为供整个租户共享或仅供所属业务组专用。共享应用程序时，必须显式共享应用程序蓝图中的所有组件，如逻辑模板、服务及外部服务。与该应用程序关联的部署配置文件不会共享。这些文件始终为所属业务组专用。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。
- 要了解如何给应用程序命名以及在描述文本框中输入哪些内容，请参见应用程序页面上已经预定义的应用程序。要查看这些应用程序，您必须以默认业务组成员用户身份登录。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击 ✚ **创建新应用程序**。

3 在**名称**文本框中输入应用程序名称。

4 （可选）在**描述**文本框中输入应用程序的描述。

5 从**业务组**下拉菜单中选择应用程序所属的业务组。

6 在**共享**下拉菜单中，选择应用程序的共享状态。

只有所属业务组的成员可以访问专用应用程序。租户中所有业务组的成员都可以访问共享应用程序。

7 单击**保存**。

创建应用程序版本

您可以为一个应用程序创建多个应用程序版本。当您创建应用程序版本时，应用程序名称保持不变，但是不会从现有应用程序中复制蓝图内容。您必须为新应用程序版本创建蓝图。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。
- 要了解如何给应用程序命名以及在描述文本框中输入哪些内容，请参见应用程序页面上已经预定义的应用程序。要了解这些应用程序，您必须以默认组成员用户身份登录。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 打开一个应用程序。

3 单击 ✚**创建应用程序版本**。

4 在**版本**文本框中输入该应用程序的主次版本。

例如，您可能还会向主要和次要版本添加限定符，如 **`1.0.0-SNAPSHOT`**。

5 （可选）在**描述**文本框中输入该应用程序版本的描述。

6 单击**保存**。

该应用程序版本将显示在原始应用程序的“应用程序版本”部分。

下一步

为该应用程序版本的应用程序蓝图建模。请参见第 127 页，“应用程序蓝图建模”。

应用程序蓝图建模

应用程序蓝图可对安装依赖关系、配置更改和可编辑的 Application Services 脚本进行精细控制。在蓝图中生成了执行计划，您可以对其进行修改并用于在支持的云环境中部署应用程序。

可在蓝图画布中进行应用程序建模并创建应用程序。蓝图画布左侧是 Application Services 库中的逻辑模板。右侧是库和应用程序组件中可用的服务的列表。您可以选择逻辑模板并将其拖动到蓝图画布的中心，以开始进行应用程序建模。

时常保存更改。如果 Application Services 用户界面会话不活动，将在 30 分钟后过期。如果会话过期，未保存的所有更改将会丢失。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。
- 验证 Application Services 中是否至少创建了一个应用程序版本。请参见第 126 页，“创建应用程序版本”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 选择现有的应用程序，然后打开一个应用程序版本。

3 单击**创建蓝图**创建应用程序蓝图。

   此时将显示用于应用程序蓝图建模的画布。

4 选择一个或多个逻辑模板并将其拖动到画布以创建节点。

   例如，要创建一个三层应用程序，可将操作系统模板列表中的三个项目拖动到画布上，或分别拖动操作系统模板列表中的一个模板、数据库服务器列表中的一个项目以及应用程序服务器列表中的一个项目。这些列表的名称与模板关联的标记相对应。

5 （可选）选择画布上的一个节点，然后在蓝图下方的**详细信息**选项卡上更改节点名称。

   如果节点名称包含除字母、数字、下划线或连字符以外的其他字符，则该字符将被替换为连字符。

   例如，对于一个三层应用程序，可将三个节点分别重命名为 Application_Server、Database_Server 和 Load_Balancer。

6 （可选）如果计划将应用程序部署到 vCloud Director 或 vCloud Automation Center，请在蓝图下方的**详细信息**选项卡上设置主机名。

   该主机名可以充当其网络中虚拟机计算机名称的标识符。如果主机名文本框留空，则 Application Services 将生成包含随机字符的主机名。

   主机名必须遵循以下命名约定：

   - 不得超过 15 个字符。
   - 必须以字母开头。
   - 可以包含字母、数字或连字符，但不能以连字符结尾。
   - 不能与应用程序蓝图中的其他节点具有相同的主机名。

- 可在主机名结尾包含 ${random} 表达式以生成唯一字符。

  例如，部署应用程序后，Apache${random} 主机名可能会生成 Apache9INOIK3YT 等字符。您可以在部署摘要页面中查看包含唯一字符的主机名。

对于群集节点，主机名不得超过 15 个字符（包含附加的节点数组索引）。例如，称为 AppServer 的群集节点在群集中包含虚拟机主机名，如 AppServer-1 和 AppServer-2 等。

7 （可选）要更改已部署虚拟机的默认 CPU 数量或内存量，请选择画布上的相关节点，然后在蓝图下方的**详细信息**选项卡上对这些值进行编辑。

蓝图下方显示的选项卡与所选节点对应。

蓝图中的 CPU 和内存值可能与在 Amazon EC2 中创建的虚拟机的对应值不匹配。Amazon EC2 允许虚拟机中包含特定的 CPU 和内存组合，因此它将使用您在蓝图中指定的值来确定最接近的匹配值。

8 选择一个或多个服务或应用程序组件并将其拖动到节点。

例如，您可以将 MySQL 服务拖动到数据库服务器节点、将 JBoss 服务拖动到应用程序服务器节点，以及将 Apache 服务拖动到负载平衡器节点。

如果服务或应用程序组件与特定节点不兼容，则无法将其拖放到节点上。例如，您可以将称为 SQL SCRIPT 的应用程序组件拖动到 MySQL 服务，但无法将 SQL SCRIPT 组件拖动到 JBoss 服务。

应用程序目录管理员会在设置库服务支持的操作系统和组件时创建兼容性限制。应用程序目录管理员还可以向操作系统和标记列表添加库中可用的操作系统和标记。例如，库中的 MySQL 服务已列出支持的组件，如 SQL SCRIPT。只能将 SQL SCRIPT 应用程序组件类型添加到 MySQL 服务。

此外，MySQL 服务已将支持的操作系统设置为 CentOS32 6.3.0、CentOS64 6.3.0、CentOS32 6.4.0、CentOS64 6.4.0 和 Ubuntu32 12.4.2。可将 MySQL 服务添加到包括以上任一操作系统的逻辑模板。

注意 如果在创建蓝图后将任何预安装的服务添加到逻辑模板，则不会将新的预安装服务添加到节点中。在这种情况下，必须重新创建节点并添加预安装的服务。

您可以将应用程序组件 SCRIPT 和 Other 添加到节点或任何服务中。

9 单击**保存**。

下一步

配置已添加到应用程序蓝图的服务和应用程序组件。请参见第 128 页，“配置应用程序蓝图”。

配置应用程序蓝图

您可以配置服务和应用程序组件的属性及操作脚本以自定义您的应用程序部署。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。
- 如果应用程序需要从企业防火墙外部访问 URL，请将适用的服务和应用程序组件配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 如果计划自定义您的应用程序，请熟悉绑定到其他属性的基本概念。请参见第 95 页，“绑定到其他属性”。
- 确保应用程序蓝图中存在服务和应用程序组件。请参见第 127 页，“应用程序蓝图建模”。

步骤

1 选择一个服务或应用程序组件，并编辑该蓝图下**详细信息**和**操作**选项卡上的信息。

在应用程序蓝图上，只能对应用程序目录管理员指定为可替代的属性进行更改。

在**操作**选项卡上，对脚本的访问可以贯穿组件生命周期的所有阶段，这些阶段包括安装、配置、启动、更新、回滚和卸载阶段。应用程序目录管理员可以通过**库 > 服务**编辑服务脚本。

2 要编辑某个属性，请在表中单击相应属性。

如果您有一个单节点或群集节点，则必须至少在该属性生命周期的一个阶段内提供所需属性的值。例如，要运行 Apache Tomcat 服务器，Java 必不可少，并且必须设置 JAVA_HOME 属性值。

此时将显示“编辑属性”对话框。

3 （可选）要将某个属性绑定到其他属性或项目，请从“编辑属性”对话框内的**蓝图值**下拉菜单中选择属性值或项目名称。

绑定到其他属性可允许您根据其他节点的运行时属性（如 IP 地址）值自定义脚本。绑定到项目可允许您直接使用组件，而无需了解其物理位置。

4 完成应用程序蓝图的创建后，请单击**保存**。

Application Services 将检查您创建的应用程序拓扑并显示一个消息框，其中列出所有错误。例如，如果您所选的属性类型与脚本类型不兼容，或者某个服务或组件缺少必需脚本，那么您将看到此消息。必须先更正某些错误，然后才能保存应用程序。

下一步

创建部署配置文件。请参见第 141 页，“创建部署配置文件”。

创建高级蓝图

通过 Application Services，您可以利用组件、群集节点和多个网络之间的依赖关系创建高级应用程序蓝图。

图 14-1 群集化的 Windows 应用程序示例

- 创建组件之间的依赖关系第 130 页，

  在蓝图中添加依赖关系，以定义执行部署任务时必须遵循的顺序。创建某个项目（如服务或应用程序组件）到另一个服务或应用程序组件的依赖关系链接可保证创建第一个项目的任务成功完成后才会开始第二个任务。

- 将节点指定为群集第 130 页，

  对于扩展部署，您可能需要针对特定节点部署多个虚拟机或一个群集并使用负载平衡器管理它们。

- 为节点定义多个网卡第 131 页，

  在大多数的部署中，部分服务器将部署到 DMZ 区域，部分服务器将部署到受防火墙保护的网络。

- 添加和管理可变磁盘布局第 131 页，

  可变磁盘布局增强了存储的灵活性，并允许您将其他磁盘添加到节点中。也可以在创建应用程序蓝图时添加磁盘。这些磁盘在置备过程中动态创建并添加到节点。

- 将外部服务添加到高级蓝图第 132 页，

  如果应用程序需要外部服务才能正常工作，可以将基本或高级外部服务添加到蓝图。

创建组件之间的依赖关系

在蓝图中添加依赖关系，以定义执行部署任务时必须遵循的顺序。创建某个项目（如服务或应用程序组件）到另一个服务或应用程序组件的依赖关系链接可保证创建第一个项目的任务成功完成后才会开始第二个任务。

步骤

1 要创建服务或应用程序组件之间的依赖关系，请单击画布顶部工具栏中的**添加依赖关系** ( )。

2 选择第一个组件，然后再选择其依赖的组件。

  例如，通常只有在服务器已启动并且正在运行的情况下才能配置负载平衡器，因此可以添加从 Apache 服务到 WAR 组件的依赖关系。

  显示一条蓝色点线，指向从属组件。

3 完成时，请单击画布上方的工具栏中的**保存**。

示例：创建 JBoss 到 MySQL 的依赖关系

例如，可以创建 JBoss 服务到 MySQL 服务的依赖关系链接。

显示的蓝线指示应用程序服务器 JBoss 依赖于已创建且已配置的 MySQL 数据库。

下一步

部署应用程序。请参见第 141 页，第 16 章“部署应用程序”。

将节点指定为群集

对于扩展部署，您可能需要针对特定节点部署多个虚拟机或一个群集并使用负载平衡器管理它们。

前提条件

熟悉绑定到其他属性、节点数组索引属性以及定义组件操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。

步骤

1 要指定虚拟机群集，请单击节点中的**转换为节点数组**图标 ( )。

2 在蓝图下，设置群集大小。

3 绑定到 node_array_index 等属性以识别运行当前脚本的虚拟机。

  可以通过将属性绑定到 all(node:ip) 来查找群集中所有虚拟机的 IP 地址。

4 （可选）如果其他属性引用群集属性，请定义组件属性以访问群集节点中属性值的数组。

5 单击画布上方工具栏中的**保存**。

示例：指定负载平衡器

要了解使用群集的方式，请参见群集化的 Dukes Bank 示例应用程序。在负载平衡器节点中，单击 **Apache_LB** 并注意 http_node_ips 属性引用 all(appserver:ip)。https_node_ips 属性引用指定群集大小内每个节点的所有 IP 地址。

下一步

部署应用程序。请参见第 141 页，第 16 章“部署应用程序”。

为节点定义多个网卡

在大多数的部署中，部分服务器将部署到 DMZ 区域，部分服务器将部署到受防火墙保护的网络。

在群集化的 Duke's Bank 示例应用程序中，负载平衡器节点是应从公用网络访问的唯一节点。数据库节点和 Appserver 节点必须部署在受防火墙保护的专用网络中。负载平衡器节点还必须有权访问数据库节点和 AppServer 节点。

在 Application Services 中，要解决这种情况，可以在负载平衡器上定义两个网卡。每个网卡都必须指定一个逻辑网络名称。部署时，逻辑网络会映射到实际的云网络。创建虚拟机时，虚拟机的网卡数量派生自节点。

在 Dukes Bank 示例应用程序中，负载平衡器节点具有两个网卡，即指向 ServiceNetwork 的 NIC0 和指向 MgmtNetwork 的 NIC1。数据库节点和 Appserver 节点具有一个指向 ServiceNetwork 的网卡。部署时，ServiceNetwork 可以映射到受防火墙保护的云网络，而 MgmtNetwork 可以映射到公共云网络。

前提条件

虚拟机中具有多个网卡时，请先熟悉预定义 IP 地址属性概念。请参见第 98 页，“预定义 IP 地址属性”。

步骤

1 要添加多个网卡，请选择一个节点，然后单击该节点下方的**网卡**选项卡。

   最多可以向一个节点添加 10 个网卡。

2 要添加网卡，请单击**添加**图标 (✚)，然后指定逻辑网络名称。

3 （可选）要与其他节点处于同一网络，请从下拉菜单中选择网络名称。

4 单击画布的其他部分或按 Enter 键完成网卡的添加。

5 单击画布上方工具栏中的**保存**，保存您的更改。

下一步

部署应用程序。请参见第 141 页，第 16 章“部署应用程序”。

添加和管理可变磁盘布局

可变磁盘布局增强了存储的灵活性，并允许您将其他磁盘添加到节点中。也可以在创建应用程序蓝图时添加磁盘。这些磁盘在置备过程中动态创建并添加到节点。

还可以尝试将磁盘置于其他数据存储中。一旦用户为蓝图中的节点定义了可变磁盘布局，应用程序部署人员即可进一步自定义该部署：

- 将各个磁盘映射到特定的数据存储
- 设置每个额外的磁盘的磁盘大小

额外的磁盘属于虚拟机节点的一部分。使用这些磁盘的服务和使用该服务的用户都可以定义这些磁盘。因此，在服务与用户之间进行了一些假定。

例如，如果某项 Oracle 服务至少需要一个数据磁盘和一个重做日志磁盘才能运行，则用户必须将两个磁盘添加到该节点，否则，服务脚本将无法正确运行。

注意 不能为同一节点数组中的各节点指定不同的磁盘设置。

数据库可以利用可变磁盘布局。在 Application Services 中添加了新 Oracle 11g 数据库服务作为一项现成服务。此服务已修改，能够利用可变磁盘布局。

前提条件

计划在虚拟机中添加多个磁盘时，请熟悉预定义的磁盘布局信息属性。请参见第 98 页，“预定义的磁盘布局信息属性”

步骤

1 选择节点，然后单击蓝图下面的**磁盘**选项卡中的**新建**。

将磁盘添加到应用程序蓝图时，请考虑使用以下选项：

表 14-1 磁盘选项

选项	描述
名称	节点中唯一的磁盘名称。
挂载路径	要挂载磁盘的路径。如果挂载路径未指定或磁盘未格式化，Application Services 将不挂载磁盘。
文件系统	可以指定的文件系统，例如 ext3 或 NTFS。
大小 (GB)	要置备的磁盘的大小。
标记	指定能够用来确定磁盘与存储之间的映射的字符串。
描述	指定磁盘的更多信息。

2 （可选）分配带有多个磁盘标记的每项额外的任务。

每个磁盘标记可用于磁盘布局信息中的多个磁盘。磁盘标记用于指示磁盘的用途。有四种预定义的标记，这些标记已针对数据库服务优化：

表 14-2 磁盘标记

标记	描述
存档	存储数据库存档文件。
二进制	存储二进制文件，例如 Oracle 二进制文件。
数据	存储数据文件，例如 Oracle 表空间文件。
日志	存储数据库联机日志。

3 完成时，请单击画布上方的工具栏中的**保存**。

下一步

部署应用程序。请参见第 141 页，第 16 章“部署应用程序”

将外部服务添加到高级蓝图

如果应用程序需要外部服务才能正常工作，可以将基本或高级外部服务添加到蓝图。

前提条件

- 熟悉基本或高级外部服务的基本概念。请参见第 109 页，“将外部服务添加到库”。

- 了解定义服务的基本概念。请参见第 104 页，“定义服务版本属性”。
- 熟悉如何创建节点之间的依赖关系。请参见第 130 页，“创建组件之间的依赖关系”。

步骤

1 打开现有应用程序蓝图。

2 选择一个外部服务并将其拖放至蓝图画布。

   例如，您可以将 vPostgres 外部服务添加到蓝图中。

3 （可选）在蓝图中创建适用组件之间的依赖关系。

   显示一条蓝色点线，指向从属组件。

4 单击要编辑的属性。

   您可以像编辑或定义库中任何其他服务的属性一样编辑或定义外部服务属性。

5 单击**保存**。

下一步

第 141 页，第 16 章“部署应用程序”。

复制应用程序版本

复制现有应用程序版本时，会复制除部署配置文件以外的所有蓝图内容，且应用程序名称保持不变。只能编辑应用程序版本和版本描述。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。
- 如果您不熟悉将组件添加到应用程序蓝图、创建组件间的依赖关系或修改服务和脚本的相关流程，请参见第 126 页，“创建应用程序”。
- 验证您是否为拥有相应应用程序版本的业务组的成员。

  如果您不属于拥有相应应用程序版本的业务组，则**复制**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 打开应用程序，并选择应用程序版本。

3 单击工具栏中的**复制**按钮 ( ) 以复制应用程序版本。

4 更改应用程序版本，添加描述，然后单击**保存**。

   此时将打开复制的应用程序版本。

下一步

创建部署配置文件。请参见第 141 页，“创建部署配置文件”。

复制应用程序

通过复制应用程序，您可以保留应用程序版本的所有组件、设置新的应用程序名称并将应用程序版本更改为独立的应用程序。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。

- 如果您不熟悉将组件添加到应用程序蓝图、创建组件间的依赖关系或修改服务和脚本的相关流程，请参见第 126 页，“创建应用程序”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 从“应用程序”页面指向应用程序卡视图。

3 单击**复制** ( ) 以复制应用程序。

4 从下拉菜单中选择现有的应用程序版本。

5 为应用程序设置新名称并添加描述。

   您可能会在执行计划中加上有关配置的属性和服务、依赖关系、节点配置或自定义任务的详细信息。

6 单击**确定**。

   新的应用程序将显示在“应用程序”页面上。

下一步

创建部署配置文件。请参见第 141 页，“创建部署配置文件”。

删除应用程序版本

如果应用程序版本已弃用，您可以将其从设备中删除。

当删除应用程序版本时，也会随之删除蓝图。无法删除带有部署配置文件的应用程序版本。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。
- 如果您不熟悉将组件添加到应用程序蓝图、创建组件间的依赖关系或修改服务和脚本的相关流程，请参见第 126 页，“创建应用程序”。
- 验证您是否为拥有相应应用程序版本的业务组的成员。

  如果您不属于拥有相应应用程序版本的业务组，则**删除**按钮处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 打开应用程序，并选择应用程序版本。

3 单击**删除**和**确定**以确认。

应用程序版本已删除，并显示应用程序版本列表。

使用项目 15

项目管理允许您的组织使用构建文件和其他类型的软件项目来部署应用程序，而不必考虑其物理位置。

项目是一个脚本或构建过程的输出。借助项目管理，应用程序蓝图可通过类型和名称指定项目，而不是通过位置或唯一标识符。项目管理将监视项目的物理位置和标识，并在部署期间提供所需项目。

在项目管理中，应用程序目录管理员创建项目存储库规范并将其映射到项目存储库实例。项目存储库规范将标识可包含无限数量项目的库。它具有可标识项目存储库实例及其所含项目的属性。Application Services 为 Jenkins 和 YUM 提供现成的项目存储库规范，还可以定义其他项目存储库规范。

项目存储库规范具有可从项目存储库实例获取项目的代理端和服务器端脚本。脚本可生成元数据并将其与项目一同存储。它还可以跟踪并搜索项目。脚本生命周期包括解析项目、下载项目和查找跟踪 ID。您可以编写自己的 Bash、Windows Cmd、Windows Powershell、BeanShell 和 JavaScript 脚本。

应用程序架构师和其他应用程序管理员创建项目规范，将其映射到项目存储库实例，例如实际的 Jenkins 构建项目，然后将项目规范绑定到服务属性、外部服务或应用程序组件，例如应用程序蓝图中的 WAR 或 JAR 文件。项目规范仅通过名称、描述和业务组标识项目。您选择的部署环境将决定实际的项目存储库实例，而您使用的部署配置文件将指定实际项目。在每个部署环境中，您可以将一个项目映射到一个存储库实例。

本章讨论了以下主题：

- 第 135 页，“创建项目存储库规范”
- 第 137 页，“创建项目存储库实例”
- 第 138 页，“创建项目规范”
- 第 138 页，“将项目规范映射到项目存储库实例”
- 第 139 页，“将项目绑定到应用程序蓝图”
- 第 140 页，“使用项目部署应用程序”

创建项目存储库规范

项目存储库规范是项目存储库实例的模板。

项目存储库规范用于定义访问项目存储库实例所用的属性并标识所含项目。规范中包含在部署期间使用属性值来解析项目存储库实例和项目的操作脚本。

前提条件

- 以**应用程序架构师**或**应用程序目录管理员**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 项目存储库**。

2 单击“添加”图标 ( )。

3 输入项目存储库规范的名称和描述。

4 输入用于标识存储库中物理项目的跟踪标识符。

使用唯一字符串作为构建 ID、作业 ID、提交 ID、版本号或其他标识符。系统使用此标识符来查找同一存储库中同一项目的不同版本。可以在操作脚本中使用此标识符。

5 从**业务组**下拉菜单中选择业务组。

只有业务组用户可以访问通过此规范创建的项目存储库实例。

6 添加存储库访问属性。

存储库访问属性用于指定查找项目存储库实例所需的服务器 URL 以及任何其他信息。

a 单击存储库访问属性的“添加”图标 ( )。

您也可以使用“存储库访问属性”列表中的空行（如果有）。

b 在**属性名称**文本框中输入属性名称。

c （可选）在**描述**文本框中输入描述。

d 从**类型**下拉菜单中选择属性类型。

类型	描述
字符串	指定通过此规范创建的存储库的唯一标识符。最多接受 1024 个字母数字字符。
整数	使用正值或负值指定一个整数。
内容	指定要通过脚本下载为项目的对象的 URL。
数组	以“value1”、“value2”、...形式指定数组
计算	指定要由脚本提供的计算值
布尔	指定 YES 或 NO。
双精度	指定双精度整数。
单一选择	定义单一选择语句。

e （可选）在**值**文本框中输入属性值。

仅当要将相同的值应用于通过此规范创建的所有项目存储库实例时，才在此处输入值。

f （可选）选中**受保护**复选框。

选中后，通过屏蔽字符隐藏属性的值。

7 添加项目标识符属性。

a 单击项目标识符属性的“添加”图标 ( )。

您也可以使用“项目标识符属性”列表中的空行（如果有）。

b 在**属性名称**文本框中输入属性名称。

c （可选）在**描述**文本框中输入描述。

d 从**类型**下拉菜单中选择属性类型。

类型	描述
字符串	指定通过此规范创建的存储库的唯一标识符。最多接受 1024 个字母数字字符。
整数	使用正值或负值指定一个整数。
内容	指定要通过脚本下载为项目的对象的 URL。
数组	以“value1”、“value2”、...形式指定数组
计算	指定要由脚本提供的计算值
布尔	指定 YES 或 NO。
双精度	指定双精度整数。
单一选择	定义单一选择语句。

e （可选）在**值**文本框中输入属性值。

仅当要将相同的值应用于项目存储库实例中的所有项目时，才在此处输入值。

f （可选）选中**受保护**复选框。

选中后，通过屏蔽字符隐藏属性的值。

8 创建脚本以解析项目的位置。

a 从**脚本类型**下拉菜单中为 RESOLVE_ARTIFACT 脚本选择脚本类型。

b 单击**脚本**文本框，编辑脚本。

c 在编辑窗口中输入脚本文本。

可以使用编辑窗口顶部的下拉菜单更改脚本类型或选择要插入的属性。

d 单击“确定”关闭编辑窗口并保存脚本。

脚本文本将显示在“脚本”文本框中。

9 完成之后，单击**保存**以保存所做的更改。

下一步

第 137 页，“创建项目存储库实例”。

创建项目存储库实例

通过将物理存储库与项目存储库规范相关联来创建项目存储库实例。

前提条件

- 以**应用程序云管理员**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。
- 验证是否存在合适的项目存储库规范。
- 验证是否存在合适的部署环境。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**云 > 部署环境**。

2 单击要使用的部署环境。

此时将显示部署详细信息窗口。

3 单击**项目存储库实例**选项卡。

4 单击“添加”图标 ( )。

此时将显示“新建项目存储库实例”页面。

5 从**项目**下拉菜单中选择一个项目存储库规范。

6 在**名称**文本框中输入名称。

7 （可选）在**描述**文本框中输入描述。

8 在“属性”部分输入所有必填属性值。

9 单击**保存**。

下一步

第 138 页，“创建项目规范”。

创建项目规范

项目规范用于描述位于一个或多个项目存储库实例中的逻辑项目。项目规范不提供任何物理详细信息。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 项目**。

2 单击项目的“添加”图标 ( )。

3 在**名称**文本框中输入项目名称。

4 （可选）在**描述**文本框中输入描述。

5 从**业务组**下拉菜单中选择业务组。

只有业务组的用户可以访问该项目。

6 完成之后，单击**保存**。

下一步

第 138 页，“将项目规范映射到项目存储库实例”。

将项目规范映射到项目存储库实例

在将项目规范映射到项目存储库实例时，您将指定部署期间在何处查找项目。

您还可以更改项目存储库实例到项目规范的当前映射。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。
- 验证是否存在合适的项目规范。请参见第 138 页，“创建项目规范”。
- 验证是否存在合适的项目存储库实例。请参见第 137 页，“创建项目存储库实例”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**库 > 项目**。

2 单击项目存储库的“添加”图标 ( )。

此时将显示项目的“存储库”窗口。

3 从**项目名称**下拉菜单中选择项目名称。

已映射到项目规范的任何项目存储库实例都会显示在“项目存储库实例”列表中。

4 单击项目存储库实例的“添加”图标 ( )。

列表中会显示一个具有下拉菜单的新行。

5 从下拉菜单中选择一个项目存储库实例。

所选实例的属性值将显示在“项目标识符”表中。

6 （可选）输入一个新的属性值以替换当前值（如果需要）。

7 （可选）更改项目存储库实例到项目规范的当前映射。

   a 在**项目名称**下拉菜单中选择项目规范。

   b 选择要更改的项目存储库实例。

   c 选择一个不同的项目存储库实例。

8 单击“确定”。

下一步

第 139 页，“将项目绑定到应用程序蓝图”

将项目绑定到应用程序蓝图

将项目绑定到应用程序蓝图时，实际上是将项目规范绑定到蓝图中应用程序组件、服务或者外部服务的属性。

前提条件

- 以**应用程序架构师**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。
- 验证是否存在具有项目存储库实例的项目规范的合适映射。请参见第 138 页，“将项目规范映射到项目存储库实例”。
- 验证是否存在应用程序和应用程序蓝图。请参见第 126 页，“创建应用程序”和第 128 页，“配置应用程序蓝图”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 打开应用程序、应用程序版本，然后打开蓝图。

3 选择现有的或者添加新的应用程序组件、服务或外部服务。

4 单击**属性**选项卡。

5 找到要绑定到项目的属性。

例如，JAR 文件有一个 jar_file 属性，您可以指定可下载文件的 URL。或者，您可以生成并绑定一个唯一的文本字符串，将其作为软件包名称，供用户从 yum 存储库安装软件包。

6 单击属性的“编辑”按钮。

此时将显示“编辑属性”对话框。

7 从**蓝图值**下拉菜单中选择带有项目绑定标记的项目名称。

8 完成之后，单击**保存**。

9 系统提示保存蓝图更改时，单击**确定**。

下一步

第 140 页，“使用项目部署应用程序”。

使用项目部署应用程序

您可以将应用程序部署为使用一个或多个项目作为应用程序组件。

为部署配置文件选择的部署环境决定要使用的项目存储库实例。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。
- 验证是否存在具有项目存储库实例的部署环境。请参见第 137 页，“创建项目存储库实例”。
- 验证是否存在应用程序和应用程序版本。
- 验证是否存在具有项目绑定的应用程序蓝图。请参见第 139 页，“将项目绑定到应用程序蓝图”。
- 验证是否存在部署配置文件。
- 验证是否有权访问部署配置文件的应用程序属性。请参见第 145 页，“配置应用程序属性”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 依次打开应用程序、应用程序版本和部署配置文件。

3 打开“部署配置文件”向导中的“应用程序属性”页面。

4 单击映射的对象类型所对应的选项卡。例如，单击**应用程序组件**或**服务**显示相应的属性。

5 在**新值**文本框中找到项目存储库和项目名称。

这些值具有 `Repository:` 和 `Artifact:` 前缀。

6 单击**新值**文本框中的名称。

此时将显示**属性值**对话框。

7 （可选）您可以为项目存储库规范中定义的跟踪标识符输入值。

对于 Jenkins 内部版本，现成项目存储库规范的跟踪标识符为 JOB_ID，对于 Yum 内部版本则为 VERSION。其他规范可以具有其他跟踪标识符。您可以将该文本框留空，在以后进行部署时再输入跟踪标识符，方法是返回此处输入跟踪标识符，或者使用应用程序上的“快速部署”。

8 （可选）您也可以替代部署的项目管理。在“键入您自己的值”文本框中指定应用程序组件的 URL。这样，将优先使用该 URL。

9 单击“确定”。

部署应用程序 16

Application Services 简化了混合云环境中多层企业应用程序的部署并进行自动部署。

使用 Application Services，您可以为一个生命周期创建不同的部署配置文件，例如开发、测试和生产。您可以将部署配置文件添加到复合部署配置文件、绑定其属性并按顺序将其一起部署。部署配置文件设置保存在 Application Services 中。可以重新使用保存的部署配置文件以将某个应用程序版本部署到受支持的云环境中。

如果更新已部署的应用程序，则 Application Services 会根据已修改的值创建更新配置文件。该更新配置文件包含从已定义的更新生命周期阶段生成的更新脚本。

要熟悉与部署应用程序相关的主要概念，请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 141 页，“设置和配置部署配置文件”
- 第 151 页，“使用部署配置文件部署”
- 第 153 页，“将部署配置文件发布到 vCloud Automation Center 服务目录”
- 第 155 页，“请求 vRealize Automation 服务目录项”
- 第 156 页，“使用部署摘要页面”
- 第 156 页，“使用复合部署摘要页面”
- 第 157 页，“了解部署和更新过程”
- 第 158 页，“了解部署故障”

设置和配置部署配置文件

通过部署配置文件，您可以配置可用于特定部署环境的云模板、网络和应用程序属性值等设置。

在部署配置文件中，您可以查看执行计划、将自定义任务添加到执行计划、查看部署配置文件设置并在部署应用程序前进行相关更改。

创建部署配置文件

为每个部署环境创建不同的部署配置文件。在部署配置文件中，如果为属性启用了**在部署时替代**选项，则可以输入或替代特定部署的应用程序属性。

例如，对于特定部署环境，您可以将数据库端口更改为 3307。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。

- 验证 Application Services 中是否至少创建了一个应用程序。请参见第 125 页，第 14 章“创建应用程序”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

   此时将显示应用程序版本列表。

3 选择应用程序版本，然后单击“添加”图标 ( )。

   此时会显示选择菜单。

4 从菜单中选择“创建部署配置文件”。

   该菜单还包含“创建复合部署配置文件”，用于创建现有部署配置文件的群集。

5 输入部署配置文件的名称。

   您可以命名该配置文件，以便使用该名称指示使用的环境类型、特定的替代属性或群集节点配置。

   例如，对于称为 MyTimecard 的应用程序，您可以命名配置文件 myTimecard-QA，指示此配置文件用于 QA 环境而非生产环境。

6 如果为共享应用程序创建部署配置文件，请从下拉菜单中选择业务组。

   不能为专用应用程序选择业务组。

7 单击**部署**。

此时将显示部署配置文件向导，同时突出显示“部署环境”页面。

下一步

映射应用程序部署的逻辑模板和网络模板。请参见第 142 页，“配置部署环境”。

配置部署环境

您可以检索部署环境中可用的云模板和网络的列表，然后将其映射到逻辑模板和逻辑网络。您还可以在 Application Services 中配置自定义属性，以替代 vRealize Automation 蓝图自定义属性（将预定义的磁盘映射到存储）或添加到现有属性。

“虚拟机模板”部分中列出的节点与该应用程序的组件相对应，如应用程序蓝图中所示。如果应用程序中包含的自定义外部服务需要运行脚本，虚拟机模板将映射到临时虚拟机。

“网络”部分列出了在蓝图中定义的逻辑网络。您选择并映射的网络取决于部署环境。部署配置文件还包含 vCloud Director 和 Amazon EC2 网络的网络列表。对于 vCloud Director，Application Services 支持外部网络和 vCloud 路由网络（无论是否采用 DHCP）。如果网络列表为空，请联系您的 vCloud Director 管理员。对于 Amazon EC2，Application Services 支持 NAT 路由网络、公用网络和专用网络。NAT 路由网络和公用网络可以访问 Internet。非 NAT 路由的专用网络只能访问 Application Services 设备。要部署到 Amazon EC2，您必须谨慎决定置于外部网络中的虚拟机。外部网络中的每个网卡都将获得一个弹性 IP 地址，该地址将该接口置于 Internet 中。请仅在确实需要时才将网卡置于外部网络中。

您可以通过映射云模板来连接到外部云网络。要连接到外部云网络，请执行以下操作：

- 确保网络配置文件的类型为外部。
- 确保将其分配给网络路径或创建预留。

从下拉列表中选择云网络名称，并选择一个指向节点或群集中定义的网卡的网络映射。单击云网络名称旁的图标可以查看网络详细信息。

可以选择定义 vRealize Automation 蓝图以包含单个自定义属性或分组为构建配置文件的自定义属性集合。在创建虚拟机后，这些自定义属性会应用到虚拟机。Application Services 允许您替代 vRealize Automation 蓝图自定义属性或添加到现有属性。例如，要替代现有 vRealize Automation 网络信息，您可以在应用程序蓝图中的特定节点的自定义属性中指定网络或静态 IP 地址。只要创建虚拟机，就会应用定义的该自定义属性。

自定义属性为键-值对。您可以将这些属性定义为“键=值”。

有关定义和使用蓝图上的自定义属性的相关信息，请参见*《自定义属性参考》*。

注意 请不要使用 Application Services 预留属性和内部属性作为自定义属性。

请参见第 145 页，“Application Services 预留属性和内部属性”。

注意 在**额外配置**选项可用时，建议您在 vRealize Automation 中将自定义属性添加到计算机蓝图中，而不是使用**额外配置**选项添加它们。从 vRealize Automation 目录请求应用程序置备时，在 vRealize Automation 中添加到计算机蓝图的自定义属性将显示为节点属性。不过，您可以使用**额外配置**选项来添加不作为计算机蓝图的一部分显示的其他自定义属性，包括不属于蓝图的关联构建配置文件或业务组的自定义属性。

前提条件

- 以**应用程序云管理员**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否至少有一个云模板已映射到蓝图中使用的每个逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。
- 验证 Application Services 中是否至少创建了一个应用程序。请参见第 125 页，第 14 章“创建应用程序”。
- 根据您的云环境，必须至少有一个 vCloud Director 或 Amazon EC2 网络可用于部署环境。请参见第 76 页，“创建 vCloud Director 部署环境”或第 86 页，“创建 Amazon EC2 部署环境”。
- 将您的云模板注册到 Application Services 云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”、第 55 页，“注册 vRealize Automation 云提供程序和模板”和第 85 页，“注册 Amazon EC2 云提供程序和模板”。
- 熟悉 Application Services 部署的自定义属性和预留自定义属性。请参见 vRealize Automation *《自定义属性参考》*。
- 验证是否已创建部署配置文件。请参见第 141 页，“创建部署配置文件”。

步骤

1 从列表中选择部署环境，然后单击**映射详细信息**。

   您可以查看“部署环境”部分的相应列表，单击标题栏，然后从下拉菜单中选择**云 > 部署环境**。

2 在“外部服务”部分中，将外部服务映射到部署环境中的外部服务实例。

   如果外部服务实例的列表为空，则将不在部署环境中注册现有实例。如果您有访问权限，请至少注册一个外部服务实例，或者让您的云管理员为您注册一个实例。

   关联的外部服务实例在下拉菜单中显示。

3 在“虚拟机模板”部分中，将逻辑模板映射到云环境中的云模板。

   如果云模板的列表为空，则说明现有云模板不属于您的组，或者云模板未注册。如果您拥有访问权限，请登录到具有现有云模板的组，或者在 Application Services 中注册至少一个云模板。还可以让您的云管理员为您注册一个云模板。

   对于 vRealize Automation，与部署环境具有相同预留策略的云模板在下拉菜单中显示。

   与用户属于相同组的云模板在下拉菜单中显示。

4 （可选）对于 vRealize Automation 部署，单击**额外配置**图标 ( ) 以将自定义属性添加到应用程序蓝图中的每个节点。

请不要使用 Application Services 预留和内部属性。

a 定义自定义属性以将管理网络映射到 vCenter Server 直接网络，将服务网络映射到示例群集化的 Dukes Bank 应用程序的 vCenter Server 路由网络。

vCenter Server 直接网络为 network1，vCenter Server 路由网络为 network2。

b 在 appserver 节点行中，单击**额外配置**图标 ( )，在“appserver”对话框的“额外配置信息”中键入 **`virtualmachine.network0.name=network2`**，然后单击**保存**。

c 在数据库节点行中，单击**额外配置**图标（ ），在“数据库”对话框的“额外配置信息”中键入 **`virtualmachine.network0.name=network2`**，然后单击**保存**。

d 在“load_balancer”节点行中，单击**额外配置**图标 ( )，在“load_balancer”对话框的“额外配置信息”中键入 **`virtualmachine.network0.name=network1 virtualmachine.network1.name=network2`**，然后单击**保存**。

5 对于 vCloud Director 和 Amazon EC2 部署，请在“网络”部分中为目录中的每个逻辑网络选择一个支持的云网络。

例如，对于负载平衡器，如果要将应用程序部署到测试环境中，您可能会为两个负载平衡器网络网卡选择一个内部网络。为生产环境创建部署配置文件时，您可能会为一个负载平衡器网卡选择一个内部网络，为另一个负载平衡器网卡选择一个外部网络。

6 在“磁盘”部分中，将每个单独的磁盘映射到特定存储。

在以下情况下，“磁盘”部分可能为空：

- 相应的应用程序蓝图可能未定义额外的磁盘。在这种情况下，请将该部分保留为空，然后继续进行部署。
- 选择了非 vRealize Automation 部署环境。在这种情况下，您可以安全地忽略警告消息选定的部署环境不允许在虚拟机中自定义磁盘。在蓝图中定义的磁盘不会被置备。如果操作脚本取决于蓝图中的自定义磁盘设置，则部署可能会失败 (`Selected Deployment Environment does not allow customization of disks in the VM. Disks defined in the blueprint will not be provisioned. The deployment might fail if the action scripts depend on customized disk settings in the blueprint`)。然后继续进行部署。

可变磁盘布局允许您将磁盘置于特定数据存储中，从而获得最高性能，并将成本降至最低。

例如，在快速数据存储中创建操作系统磁盘以获得更好的性能，在缓慢数据存储中创建一个存档磁盘以降低成本。

7 完成选择后，单击**下一步**。

此时将显示**应用程序属性**选项卡。

下一步

定义节点、服务和应用程序组件的适用属性值。请参见第 145 页，“配置应用程序属性”。

Application Services 预留属性和内部属性

一些预留属性和内部属性在创建自定义属性时不可用。如果使用这些属性，您将收到错误消息。

预留属性和内部属性

预留属性	内部属性
agent.download.url	Hostname
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	
APPD_REQUEST_ID	

配置应用程序属性

可以为节点属性（如主机名或 vCPU）定义新值。还可以为在应用程序蓝图中选中**可在部署时替代**复选框的应用程序组件和服务属性定义新值。

对于成功部署，会从库、蓝图或部署配置文件为所需节点属性分配一个值。系统将定义节点属性，例如内存分配和 CPU 数量，但是您可以替代它们。例如，某项所需服务的 JVM 堆大小可能为 512 MB。但是对于大型部署，您可以替代该设置并将大小更改为 1024MB。

您可以设置主机名，以便可以在 vCloud Director 或 vRealize Automation 部署中轻松识别虚拟机。要生成唯一的字符，您可以将 ${random} 表达式附加在主机名末尾。

群集节点的群集大小还可以使用新值定义。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证 Application Services 中是否至少创建了一个应用程序。请参见第 125 页，第 14 章“创建应用程序”。
- 验证是否已配置**部署环境**选项卡。请参见第 142 页，“配置部署环境”。

步骤

1 验证您在蓝图中设置的属性是否在组件选项卡中正确显示。

2 （可选）单击**服务**、**应用程序组件**或**节点**选项卡。

3 选择特定服务、应用程序组件或节点属性。

4 在表中，单击特定行的“新值”列，键入要在部署配置文件中使用的值。

对于到 Amazon EC2 的部署，必须将 `global_conf` 属性值更改为 **`https://DarwinServerIP:8443/darwin/conf/darwin_global_noproxy.conf`**。

5 要恢复为初始值，请单击**重置值**按钮 ( )。

6 （可选）在“磁盘映射”部分，键入新的整数值以增加或减小磁盘大小。

7 完成更改后，单击**下一步**。

此时将显示**执行计划**选项卡。

下一步

查看执行计划中的置备任务和依赖关系。请参见第 146 页，“检查执行计划并添加自定义任务”。

检查执行计划并添加自定义任务

系统根据应用程序蓝图生成部署执行计划。部署应用程序之前，可以检查执行计划并添加自定义任务，以在应用程序部署中执行额外的自定义任务。

执行计划中的蓝色点线定义部署任务运行的特定顺序。

主机和代理引导程序置备任务在每个节点的组件旁边显示。对于部署到 vRealize Automation 环境中的应用程序，除主机和代理引导程序任务外，还将显示网络引导程序置备任务。这些置备任务显示的进程在代理为每个组件执行安装和设置任务之前运行。如果部署失败，您可以看到用于进行故障排除的置备任务日志。不能在执行部署计划中的主机、代理引导程序或网络引导程序置备任务之间添加自定义任务。

如果应用程序中包含的外部服务需要运行脚本，则执行计划中将显示临时虚拟机。脚本在主机和代理引导程序置备任务中成功运行后，Application Services 将移除这些虚拟机。如果应用程序在移除临时虚拟机之前出现故障，则必须标识包含外部服务虚拟机的节点，然后将其从您的云环境中移除。

蓝图有助于为所有部署环境中的某个应用程序生成一个通用执行计划。有时，您必须为每个部署环境自定义执行计划。例如，如果某个应用程序部署到生产部署环境中，您可能需要在部署之后发送一封电子邮件。在测试部署环境中，可能不需要执行此类检查。您可以创建一个电子邮件自定义任务，以在服务或应用程序组件的部署任务成功完成时发送通知电子邮件。可以在部署配置文件中将该任务添加到执行计划中，这会部署到生产部署环境中。

小心 确认自定义任务运行期间不提示任何需要用户交互的进程。任何中断都会暂停该任务，导致其无限期保留在闲置状态。您可以在一小时后取消应用程序部署，否则 Application Services 将在三天后使部署失败并保留在闲置状态。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证 Application Services 中是否至少创建了一个应用程序。请参见第 125 页，第 14 章“创建应用程序”。
- 确认已配置**应用程序属性**选项卡。请参见第 145 页，“配置应用程序属性”。
- 确认至少在 Application Services 目录中创建一个自定义任务。请参见第 122 页，“将自定义任务添加到库”。
- 确认为应用程序中的所有必需节点属性分配值，以便能够成功部署。
- 根据您的云环境，必须至少有一个 vCloud Director 或 Amazon EC2 网络可用于部署环境。请参见第 76 页，“创建 vCloud Director 部署环境”或第 86 页，“创建 Amazon EC2 部署环境”。

- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。

步骤

1 检查执行计划中的置备任务、组件和依赖关系。

2 如果节点已加入群集，请单击**展开群集**按钮 ( ) 以展开该节点。

   如果未扩展群集节点，自定义任务将仅添加到群集中的第一台虚拟机。如果应用程序架构师修改了群集节点的节点，现有自定义任务将仅应用于群集中的第一台虚拟机。部署过程中部署人员应进行检查，确认自定义任务应用于群集中的第一台虚拟机还是应用于所有虚拟机，然后执行恰当的步骤。

3 对于具有关联脚本的服务和组件，请单击组件或服务名称旁边的向下箭头，以查看脚本或脚本中使用的变量定义。

4 （可选）选择**添加脚本任务**按钮 ( ) 并将自定义任务拖至节点。

   拖动**添加脚本任务**按钮时，您可以看到定位点 ( )，这些定位点指示能够插入自定义任务的位置。对于群集节点，请将自定义任务添加到每个节点。

   例如，您可以将一个或多个自定义任务添加到应用程序服务器、数据库服务器或负载平衡器节点。

   将自定义任务拖动到节点后，将打开“添加自定义任务”对话框。

5 （可选）在**库任务名称**下拉菜单中选择任务。

   此时将显示该节点的操作系统支持的自定义任务。例如，如果 CentOS 6.3 操作系统支持自定义任务，但该节点的操作系统为 Ubuntu 12.4.2，该任务将不在菜单中列出。

   选择自定义任务时，任务、脚本和属性详细信息将在对话框中显示。

6 （可选）要替代**属性**选项卡中的属性值，请单击该属性。

   例如，在发送电子邮件自定义任务中，其中一个属性为收件人的电子邮件地址。您可以将属性电子邮件地址值设置为收件人的电子邮件地址。

   a 在“编辑属性”对话框中，键入属性的新值或从下拉菜单中选择一个现有属性，以将该属性绑定到应用程序蓝图中的其中一个属性。

   b 单击**保存**。

7 （可选）在“添加自定义任务”对话框中，检查自定义任务的脚本和属性详细信息，然后单击**确定**。

8 单击**下一步**以检查部署配置文件设置。

9 单击**保存**。

10 单击**确定**。

此时将列出与应用程序版本对应的部署配置文件。

下一步

使用部署配置文件部署应用程序。请参见第 151 页，“使用单个部署配置文件进行部署”。

使用现有部署配置文件

可以重用应用程序版本的现有部署配置文件。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。

- 将已上载的 CentOS 6.3 32 位模板注册到云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”、第 55 页，“注册 vRealize Automation 云提供程序和模板”或第 85 页，“注册 Amazon EC2 云提供程序和模板”。
- 验证是否至少有一个云模板已映射到蓝图中使用的每个逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。
- 验证部署配置文件在 Application Services 中是否可用。请参见第 141 页，“设置和配置部署配置文件”。
- 根据您的云环境，必须至少有一个 vCloud Director 或 Amazon EC2 网络可用于部署环境。请参见第 76 页，“创建 vCloud Director 部署环境”或第 86 页，“创建 Amazon EC2 部署环境”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

   此时将显示应用程序版本列表。

3 选择应用程序版本，然后单击现有部署配置文件。

   此时会打开部署配置文件向导。

4 （可选）如果您已在蓝图中创建了新节点，请将这些新节点映射到云模板。

   即使已更新蓝图值，现有部署配置文件中保存的属性替代项在新值列中仍然保持替代状态。部署配置文件中将不再显示已删除的属性。

下一步

完成用于部署应用程序的部署任务。请参见第 141 页，“创建部署配置文件”和第 151 页，“使用单个部署配置文件进行部署”。

创建复合部署配置文件

复合部署配置文件通过共享属性绑定将多个部署配置文件加入到一个可部署单元。

前提条件

以**应用程序发布者和部署者**身份登录到 Application Services。

步骤

1 设置复合部署配置文件第 148 页，

   通过复合部署配置文件，您可以一起加入多个部署配置文件。结果为应用程序的复合部署。

2 选择部署配置文件第 149 页，

   您可以选择一个或多个部署配置文件，并确定部署的顺序。

3 绑定属性第 150 页，

   您可以绑定关联应用程序中所用节点、服务和应用程序组件的属性。

设置复合部署配置文件

通过复合部署配置文件，您可以一起加入多个部署配置文件。结果为应用程序的复合部署。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证您是否为相应业务组的成员。
- 验证应用程序是否存在。请参见第 133 页，“复制应用程序”。

- 验证应用程序是否具有应用程序版本。请参见第 126 页，“创建应用程序版本”。
- 验证应用程序版本的一个或多个部署配置文件是否存在。请参见第 141 页，“创建部署配置文件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

   此时将显示应用程序版本列表。

3 选择应用程序版本，然后单击“添加”图标 ( )。

   此时会显示一个菜单。

4 从菜单中选择“创建复合部署配置文件”。

5 输入复合部署配置文件的名称。

6 选择业务组，然后单击**部署**。

   此业务组必须包含要包括到复合部署配置文件中的部署配置文件。

此时将显示“添加部署配置文件”页面。

下一步

新的复合部署配置文件已设置完毕，但此文件是空的。您需要选择部署配置文件。

选择部署配置文件

您可以选择一个或多个部署配置文件，并确定部署的顺序。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否存在多个部署配置文件。请参见第 141 页，“创建部署配置文件”。
- 第 148 页，“设置复合部署配置文件”。

步骤

1 从**应用程序**下拉菜单中选择应用程序。

   该应用程序必须包含要包括到复合部署配置文件中的部署配置文件。

2 从**应用程序版本**下拉菜单中选择应用程序版本。

3 从**部署配置文件**下拉菜单中选择部署配置文件。

   该菜单只显示与复合部署配置文件位于同一业务组中的部署配置文件。

4 单击**添加**。

   该部署配置文件将显示在选择表中。

5 重复 步骤 1 到 步骤 4 以添加其他部署配置文件。

   如果多次添加同一部署配置文件，则系统会在每个重复项别名的后面附加下划线和递增序列号。

6 （可选）更改部署配置文件的部署顺序。

默认情况下，部署配置文件以自上而下的顺序进行部署。可以通过更改序号来更改顺序。对部署配置文件进行重新排序不会更改其物理位置。

a 单击部署配置文件的序号进行更改。

b 键入数字，或者单击向上箭头或向下箭头。

7 （可选）单击部署配置文件的“查看详细信息”链接以显示信息，例如其应用程序和部署环境。

必须关闭“查看详细信息”窗格才能继续。

8 单击**可视化**以图形形式显示部署配置文件。

“查看详细信息”链接也会显示在部署配置文件的图形表示上。

9 单击**下一步**。

此时会显示“绑定属性”页面。

下一步

您已选择部署配置文件并对其进行排序。现在，您可以绑定与这些部署配置文件连接的属性。

绑定属性

您可以绑定关联应用程序中所用节点、服务和应用程序组件的属性。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 第 149 页，“选择部署配置文件”。

步骤

1 单击“添加”图标 ( )。

2 在“属性”列中单击**选择**。

此时将显示**选择属性**对话框。

3 从**部署配置文件**下拉菜单中选择已添加到复合部署配置文件的部署配置文件之一。

4 选择**服务**或**应用程序组件**单选按钮。

选择会因部署配置文件的不同而有所不同。

5 从相应的下拉菜单中选择一项。

您所做的选择将决定可供选择的属性。

6 从**属性**下拉菜单中选择一个属性。

7 单击**确定**。

8 单击“绑定到属性”列中的**选择**。

9 从**部署配置文件**下拉菜单中选择已添加到复合部署配置文件的部署配置文件之一。

该菜单将显示要在当前部署配置文件之前部署的部署配置文件。这些部署配置文件的订单号小于当前部署配置文件。

10 选择**节点**、**服务**或**应用程序组件**单选按钮。

选择会因部署配置文件的不同而有所不同。

11 从**绑定到属性**下拉菜单中选择一个属性。

12 单击**确定**。

13 根据需要重复此过程以绑定其他属性。

14 单击**保存**。

使用部署配置文件部署

保存部署配置文件或复合部署配置文件后，可以部署关联的一个或多个应用程序。

使用单个部署配置文件进行部署

可以通过 Application Services 用户界面部署应用程序。

您还可以从命令行界面部署应用程序。请参见第 200 页，“使用 CLI 部署和更新应用程序”。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证 Application Services 中是否至少有一个部署配置文件可用。请参见第 141 页，“创建部署配置文件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

此时将显示应用程序版本列表。

3 选择应用程序版本并创建部署配置文件或使用现有配置文件。

4 按照“部署配置文件”向导中的提示操作，并进行任何更改。

5 查看部署配置文件设置。

您可以先对部署配置文件进行必要的更改，然后再部署应用程序。

6 单击**部署**。

此时将显示部署摘要页面，该页面大约每 30 秒刷新一次，以显示部署状态。

7 （可选）单击工具栏中的**刷新**可更新部署的实时状态。

下一步

了解当应用程序部署到云时后台运行的进程。请参见第 157 页，“了解部署和更新过程”。

检查正在进行的部署的状态。请参见第 156 页，“使用部署摘要页面”。

使用复合部署配置文件进行部署

您可以从 Application Services 用户界面，使用复合部署配置文件，以选定的顺序和绑定的属性来部署多个应用程序。

您还可以从命令行界面部署应用程序。请参见第 200 页，“使用 CLI 部署和更新应用程序”。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证复合部署配置文件在 Application Services 中是否可用。请参见第 148 页，“创建复合部署配置文件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

此时将显示应用程序版本列表。

3 选择应用程序版本，并创建复合部署配置文件或使用现有配置文件。

您必须拥有一个或多个部署配置文件才能创建复合部署配置文件。

4 查看选定的部署配置文件和属性绑定。

在部署复合部署配置文件所代表的应用程序之前，您可以对其进行更改。

5 单击**部署**。

此时将显示复合部署摘要页面，该页面大约 30 秒刷新一次，以显示每个部署配置文件的部署状态。

6 （可选）单击工具栏中的**刷新**可更新部署的实时状态。

7 单击“部署名称”列中的一个名称，可显示相应部署配置文件的部署摘要页面。

单击返回箭头可返回到复合部署摘要页面。

下一步

了解当应用程序部署到云时后台运行的进程。请参见第 157 页，“了解部署和更新过程”。

检查正在进行的部署的状态。请参见第 156 页，“使用部署摘要页面”。

快速部署应用程序

从云环境中卸载已部署应用程序时，可以快速重新部署该应用程序，而无需配置部署配置文件向导中的元素。

通过快速部署，还可以在部署配置文件向导以外更新所需属性和可替代属性并部署应用程序蓝图。

快速部署应用程序时，可使用该应用程序的最新版本和关联的部署配置文件进行部署。要快速部署应用程序的早期版本，可打开该应用程序，然后快速部署特定的应用程序版本。

注意 对于 vRealize Automation 部署，如果向应用程序添加其他网卡并使用现有部署配置文件对该应用程序进行快速部署，则会收到一条错误消息。必须打开部署配置文件向导，然后选择**映射详细信息**将云网络映射到新添加的网卡。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证 Application Services 中是否至少有一个部署配置文件可用。请参见第 141 页，“创建部署配置文件”。
- 您必须是关联部署配置文件所属的业务组的成员。

  如果您不是拥有关联部署配置文件的业务组的成员，则**快速部署**按钮 ( ) 将处于禁用状态。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 从“应用程序”页面中选择要部署的应用程序。

3 选择要“快速部署”的应用程序版本。

4 单击**快速部署** ( ) 启动部署进程。

如果未选择应用程序版本，则默认情况下将部署最新的应用程序版本。

5 在**目标**下拉菜单中，选择关联的部署配置文件。

此时将显示部署所需的属性和部署时的可替代属性。

6 （可选）为所需属性或可替代属性设置新值，然后单击**部署**。

不会在部署配置文件中保存所需属性或可替代属性的修改值以供未来的部署使用。

此时将显示部署摘要页面，其中显示正在进行中的部署的状态。

下一步

通过部署摘要页面上的状态窗口跟踪部署状态。请参见第 156 页，“使用部署摘要页面”。

将部署配置文件发布到 vCloud Automation Center 服务目录

使用 Application Services 可以将部署配置文件或复合部署配置文件发布到 vRealize Automation 服务目录。vRealize Automation 用户可以请求目录项开始使用该环境中的应用程序。

发布单个部署配置文件

您可以将 Application Services 部署配置文件发布到 vRealize Automation 服务目录。vRealize Automation 用户可以使用该环境中的应用程序请求启动目录项。

您无法从 vRealize Automation 服务目录删除已发布的部署配置文件。

如果发布的部署在其蓝图中具有已映射项目，则可以从 vRealize Automation 服务目录替代此类蓝图中常用的跟踪标识符和内容 URL。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证部署配置文件在 Application Services 中是否可用。请参见第 141 页，“设置和配置部署配置文件”。
- 必须正确设置和配置 vRealize Automation IaaS 环境和授权。请参见 vRealize Automation *《安装和配置》*和*《适用于虚拟平台的 IaaS 配置》*文档。
- 验证如果部署配置文件中存在自定义任务，是否定义了所有必需的属性值，以便 SSO 用户能够请求 vRealize Automation 服务目录中的项，或者快速部署最新的应用程序版本及关联部署配置文件。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

此时将显示应用程序版本列表。

3 选择一个应用程序版本，并单击现有的部署配置文件或创建一个部署配置文件。

此时会打开部署配置文件向导。

4 （可选）如果您已在蓝图中创建了新节点，请将这些新节点映射到云模板。

即使已更新蓝图值，现有部署配置文件中保存的属性替代项在新值列中仍然保持替代状态。部署配置文件中将不再显示已删除的属性。

5 查看部署配置文件设置。

您可以在发布之前对部署配置文件进行更改。

6 单击**发布**。

7 在“发布”对话框中，为部署配置文件的名称附加更多信息，然后单击**确定**。

例如，您可以将 Publish_vCAC 附加到 ClusteredDukesBankDP 部署配置文件。在“描述”部分，添加一条可选描述，例如将使用该部署配置文件的 Dev、QE 或 IT 环境。

已发布的名称和描述会显示在 vRealize Automation 服务目录中。此名称和描述与部署配置文件的名称和描述不同。

8 在 vRealize Automation 中登录您的租户。

此时会打开 vRealize Automation 控制台。

9 选择**管理 > 目录管理 > 目录项**并打开目录中已发布的项。

10 从“服务”下拉菜单中向该目录项分配一个已授权的服务，然后单击**更新**。

此时会显示目录项列表。

11 （可选）在右侧导航中，要向目录项授权服务，请选择**授权**并打开授权。

a 在“详细信息”选项卡中，键入授权的名称，然后单击**下一步**。

b 在“项目和批准”选项卡中，从下拉菜单中将已发布的项分配给已授权的目录项。

c 在“授权的资源操作”部分，键入**应用程序部署详细信息**资源操作并从下拉菜单中选择该操作。

12 单击**更新**。

下一步

请求 vRealize Automation 目录项以用于开发。请参见第 155 页，“请求 vRealize Automation 服务目录项”。

发布复合部署配置文件

可以将 Application Services 复合部署配置文件发布到 vRealize Automation 服务目录。vRealize Automation 用户可以请求目录项开始使用该环境中的应用程序。

您无法从 vRealize Automation 服务目录删除已发布的部署配置文件。

如果发布的部署在其蓝图中具有已映射项目，则可以从 vRealize Automation 服务目录替代此类蓝图中常用的跟踪标识符和内容 URL。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证复合部署配置文件在 Application Services 中是否可用。请参见第 141 页，“设置和配置部署配置文件”。
- 必须正确设置和配置 vRealize Automation IaaS 环境和授权。请参见 vRealize Automation *《安装和配置》*和*《适用于虚拟平台的 IaaS 配置》*文档。
- 如果任何部署配置文件中存在自定义任务，请务必定义该部署配置文件中所有必需的属性值，以便 SSO 用户能够请求 vRealize Automation 服务目录中的项，或者快速部署最新的应用程序版本及关联的部署配置文件。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

此时将显示应用程序版本列表。

3 选择应用程序版本，然后单击现有复合部署配置文件或创建一个复合部署配置文件。

必须有一个或多个可用部署配置文件夹才能创建复合部署配置文件。

4 查看部署配置文件和属性绑定。

在发布复合部署配置文件之前可以对其进行更改。

5 单击**发布**。

6 在“发布”对话框中，为复合部署配置文件的名称附加更多信息，添加描述，然后单击**确定**。

例如，在“描述”部分添加关于要部署的应用程序的可选描述。

已发布的名称和描述会显示在 vRealize Automation 服务目录中。此名称和描述与部署配置文件的名称和描述不同。

7 在 vRealize Automation 中登录您的租户。

此时会打开 vRealize Automation 控制台。

8 选择**管理 > 目录管理 > 目录项**并打开目录中已发布的项。

9 从“服务”下拉菜单中向该目录项分配一个已授权的服务，然后单击**更新**。

此时会显示目录项列表。

10 要向目录项授权服务，请在右侧导航中选择**授权**并打开授权。

a 在“详细信息”选项卡中，键入授权的名称，然后单击**下一步**。

b 在“项目和批准”选项卡中，从下拉菜单中将已发布的项分配给已授权的目录项。

c 在“授权的资源操作”部分，键入**应用程序部署详细信息**资源操作并从下拉菜单中选择该操作。

11 单击**更新**。

下一步

请求 vRealize Automation 目录项以用于开发。请参见第 155 页，“请求 vRealize Automation 服务目录项”。

请求 vRealize Automation 服务目录项

请求服务目录项时，会将该项置备到指定的云环境中。

vRealize Automation 中的典型用户无法直接访问 Application Services 环境。因此，该用户可以访问 vRealize Automation 服务目录（即一种自助门户），并请求 Application Services 应用程序部署。

前提条件

- 验证您能否以应用程序发布者和部署者身份登录到 vRealize Automation IaaS 环境中。有关如何安装和配置 vRealize Automation 以及对目录项设置授权的信息，请参见 vRealize Automation 文档。
- 验证是否至少一个 vRealize Automation 目录项拥有授权且为其分配了服务。请参见第 153 页，“发布单个部署配置文件”。
- 以**应用程序发布者和部署者**身份登录到 vRealize Automation 控制台。

步骤

1 选择**管理 > 目录管理 > 目录项**。

2 查找已分配授权服务的目录项。

3 单击**目录**选项卡并导航到该服务。

4 单击服务图标上的**请求**并完成描述。

例如，您可以为了开发和测试目的请求一个项目。

5 单击**下一步**以查看该项的属性。

如果某个部署在其蓝图中使用项目映射，且该蓝图包含一个跟踪标识符或内容 URL 作为属性值，则可以在服务目录中替代该值。

您可以编辑节点属性（如 CPU、内存、群集大小和主机名），包括在节点的计算机蓝图上定义的自定义属性。

6 单击**提交**以请求目录项。

部署完成之后，Application Services 在 vRealize Automation 中创建**项目**选项卡中显示的资源。

下一步

选择 vRealize Automation 资源跟踪进度。您也可以使用 Application Services 部署摘要页面实时查看状态。请参见第 156 页，“使用部署摘要页面”。

使用部署摘要页面

Application Services 提供了一个图形用户界面，可用于在部署摘要页面上实时检查应用程序部署的状态。

您还可以使用 vCloud Director 和 vRealize Automation 用户界面检查已部署虚拟机的状态。要查看 Amazon VPC 及关联的可用性区域中 Application Services 部署的状态，请参见 Amazon AWS 文档。

运行某个部署时，在部署摘要页面上该部署的整体部署状态、更新或卸载过程都会显示在工具栏中。部署完成后，状态栏将变成红色或绿色，这取决于部署内的任务是成功还是失败。在任务状态窗口上方，任务时间轴包含用于显示应用程序部署的启动时间的时间戳、所有后续的更新部署（如扩展的部署或要修改配置的更新）或者显示是否已运行卸载过程。

图 16-1 部署摘要页面

您可以在部署摘要页面中展开窗口以查看应用程序部署的详细信息和状态。请参见第 187 页，“查看应用程序的部署任务和蓝图详细信息”和第 189 页，“查看应用程序的已部署虚拟机的详细信息和执行计划”。

使用复合部署摘要页面

Application Services 提供了一个图形用户界面，您可以在复合部署摘要页面上实时检查应用程序部署的状态。

您还可以使用 vCloud Director 和 vRealize Automation 用户界面检查已部署虚拟机的状态。要查看 Amazon VPC 及关联的可用性区域中 Application Services 部署的状态，请参见 Amazon AWS 文档。

运行部署时，在复合部署摘要页面上的工具栏中会显示部署的整体状态。此外，该页面还会显示部署的顺序、状态、部署名称、部署配置文件名称、部署环境、应用程序、应用程序版本、启动进程的登录用户、开始时间以及结束时间。

“部署名称”列中的名称带有指向相应部署配置文件的部署摘要页面的链接。单击某个名称可打开相应部署配置文件的部署摘要页面。有关部署摘要页面的信息，请参见第 156 页，“使用部署摘要页面”。

部署完成后，根据部署内的任务是成功还是失败，状态栏将相应变成绿色或红色。

了解部署和更新过程

将应用程序部署到云或更新云中已部署应用程序时，后台将运行多个进程。将创建虚拟机，并在虚拟机中置备软件。了解部署和更新过程十分重要，这样才能轻松确定任何部署故障并进行故障排除。

在 Application Services 中，可以查看执行计划中的部署过程。主机和代理引导程序置备任务显示在每个节点的组件的旁边。对于部署到 vRealize Automation 环境中的应用程序，除主机和代理引导程序任务外，还将显示网络引导程序置备任务。

将应用程序部署到云和更新已部署应用程序的过程包括以下步骤。使用复合部署配置文件，对每个部署的应用程序重复这些步骤。

1 Application Services 运行策略评估以确定部署过程的合规性。如果存在标记为“严重”的策略违反实例，部署或更新过程将停止。

2 Application Services 通过实例化部署配置文件中映射的云模板来置备虚拟机。

   要使更新过程扩展已部署应用程序，Application Services 可通过实例化部署配置文件中映射的 vCloud Director 目录或 vRealize Automation 中的云模板来置备扩展的群集虚拟机。扩展的群集虚拟机使用的模板与将应用程序部署到云时最初应用的模板相同。更新已部署应用程序时无法更改模板。

   更新配置文件过程的其余部分与部署过程相同。

3 Application Services 请求云服务为部署中的所有虚拟机建立网络连接和接收 IP 地址。分配了 IP 地址后，虚拟机将重新启动，以确保设置过程正常完成。主机名从应用程序蓝图派生，并作为节点名称进行分配。

   如果主机名未分配，则主机名将使用逻辑名称并会附加一串随机生成的字符。

4 每个虚拟机中包含的引导程序脚本从 Application Services 服务器下载代理到虚拟机。代理为在 Java 虚拟机中运行的 JAR 文件。引导程序脚本必须安装在物理模板上。

5 引导程序脚本启动代理进程。

6 代理通过 Application Services 服务器进行身份验证。

7 每个虚拟机中的代理从 Application Services 下载执行计划到虚拟机。

8 代理按照部署执行计划中指定的顺序为各个组件执行安装和设置任务。

9 对于每个脚本，代理会等待从属任务成功完成，然后将所有内容下载到虚拟机目录 `/tmp/runid/content/ComponentName/PropertyName`，而将任务脚本下载到目录 `/tmp/runid/TaskName`。代理根据服务器发送的参数值运行任务。任务完成后，代理会通知服务器任务的状态。

   如果脚本以非零退出状态完成，则代理会将该任务标记为“失败”。否则，代理会将脚本标记为“完成”并继续执行下一个任务。任务失败时，整个部署将停止并标记为“失败的部署”，并且不会再运行其他任务。可在**详细信息**选项卡上查看失败原因。所有任务都通过后，会将部署标记为“部署成功”。

   注意 要使脚本无中断地运行，必须将返回值设为零 (0)。通过设置该值，可使代理捕获所有计算属性并将其发送到 Application Services 服务器。

要进行故障排除，可以访问部署详细信息或查看执行计划中包含的置备任务日志。针对在部署各部分中使用的安装、配置、启动、更新、回滚和卸载脚本，都将在日志文件中捕获任务信息。这些日志可捕获发送到 `stdout` 和 `stderr` 日志文件的所有信息。

了解部署故障

将应用程序部署到云中或在云中更新已部署的应用程序时，会创建虚拟机并在虚拟机中置备软件。在这些过程中，可能会出现部署失败，从而影响您的部署。

失败可能会出现在多个部署阶段。在 Application Services 中，您可以通过查看执行计划来确定出现故障的位置。您可以使用**恢复部署**选项（位于“部署视图”中的**执行计划**面板）来恢复失败的部署。

表 16-1 部署阶段

部署阶段	失败种类
置备前	■ 策略验证 ■ 模板验证 ■ DE 可访问性验证
置备	出现引导程序失败的原因如下： ■ 耗尽最多虚拟机 ■ OS 安装 ■ IP 设置 ■ 存储问题 出现代理引导程序失败的原因如下： ■ 模板中的防火墙问题 ■ 模板中的网关问题 ■ 由于网络和防火墙问题而导致从代理到 Application Services 的通信超时
执行前	vRealize Automation 网络引导程序问题。
执行	■ 安装或配置阶段内出现的内容下载失败 ■ 安装或配置初始阶段内出现的脚本失败
执行后	当外部服务为部署的一部分时，会出现取消置备失败。

在恢复操作中无法添加或删除自定义任务。由于自定义任务失败而导致的部署失败无法恢复。

解决部署失败问题

在应用程序置备的任何阶段出现部署失败问题，都需要卸载应用程序部署。由于部署失败频繁发生，而且会导致严重问题，因此必须予以修复以免需要重新部署。借助“恢复部署”操作，您可以通过修复错误情况来重新启动失败的部署。

预置备失败

预置备失败出现在部署的早期阶段。如果部署在此阶段失败，将会显示一条错误消息。更正错误之后，在同一部署中触发恢复部署操作。然后，您才可以重新启动预置备过程。

在预置备失败的情况下，错误通常是由于策略和模板验证所致。

置备失败

主机置备失败或代理引导程序失败可能会导致置备失败。此时，主机或代理引导程序节点中会出现一个红色图标。如果由于 Application Services 代理安装问题或任何与代理相关的问题而导致其中一个虚拟机中的引导程序失败，系统会卸载新置备的虚拟机，然后重新置备它。部署恢复后，系统会卸载未正常工作的虚拟机并重新置备它。

注意 不支持网络引导程序失败。

执行

执行失败可能是由于内容下载失败或脚本失败所致。启用恢复选项后，您可以编辑失败组件的属性和脚本。恢复部署时，将首先执行组件的第一个脚本，如安装、配置或启动脚本。恢复虚拟机或脚本时，可能不会重新安装或启动安装过程，因为它已安装。您必须修改脚本，使虚拟机进入空白状态才能开始安装。

更新应用程序部署 17

更新现有应用程序部署时，可创建更新配置文件以捕获该更新所需的更改的新值。您也可以回滚更新来恢复到上一个有效的更新并继续启动其他更新过程。

重要事项 使用复合部署配置文件部署的应用程序不支持恢复、任何类型的更新或辅助卸载。但是，这类应用程序支持快速卸载。

您可以多次部署保存的更新配置文件以更新现有部署。借助 vCloud Application Director 6.0 和 6.0.1 以及 Application Services 6.1，您可以对 vCloud Director 和 vRealize Automation 中的现有部署多次启动更新过程，例如，缩小和扩大一个节点的群集，或者修改现有服务和应用程序组件的配置。

在扩展现有应用程序的节点群集时，请确保云中有足够的资源来支持应用程序中的其他节点。

如果部署包括外部服务，且在外部服务实例中修改了配置，则部署卡上将显示通知图标。在您启动更新或回滚过程前，请单击该图标并更新外部服务属性。

重要事项 vCloud Application Director 6.0 和 6.0.1 以及 Application Services 6.1 不支持更新 Amazon EC2 中的现有部署。

为帮您确定应用程序中已更改和受影响的属性及其依赖关系，当您更新节点的群集大小时，在更新向导和检查页面上的蓝图窗口中，Application Services 将缩小或扩大的节点突出显示为“已更改”，并将从属属性突出显示为“受影响”。包含从属属性的组件将突出显示为受影响的组件。

当更新属性以修改配置时，在更新向导和检查页面上的蓝图窗口中，Application Services 将属性突出显示为“已更改”，并将从属属性突出显示为“受影响”。包含新属性值的组件将突出显示为已更改的组件。包含从属属性的组件将突出显示为受影响的组件。

当您回滚更新时，在回滚向导和检查页面上的蓝图窗口中，Application Services 将突出显示已更改和回滚到的属性。包含新属性值的组件将突出显示为回滚到的组件。包含从属属性的组件将突出显示为已更改的组件。

本章讨论了以下主题：

- 第 162 页，“启动更新过程以扩大部署”
- 第 164 页，“启动更新过程以缩小部署”
- 第 166 页，“启动更新过程以修改配置”
- 第 168 页，“使用现有的更新配置文件”
- 第 168 页，“升级更新配置文件”
- 第 170 页，“回滚更新过程”
- 第 171 页，“了解运行自定义任务更新”
- 第 171 页，“对失败的扩展部署更新过程进行故障排除”

- 第 172 页，“对失败的修改配置更新过程进行故障排除”

启动更新过程以扩大部署

利用 Application Services，可以为现有部署创建更新配置文件（例如扩大节点的群集）以提高性能并维持可扩展应用程序部署的需求。扩大已部署应用程序的群集节点时，只能对应用程序蓝图中已建模为群集的节点的群集大小进行配置。

在扩大已部署应用程序的更新过程中，基于更新设置，将创建虚拟机并在新虚拟机上运行所需的操作脚本。在多层应用程序中，如果某节点依赖于扩大的群集节点而不是外部服务，则必须在从属节点上运行更新脚本。

例如，在部署的群集化的 Dukes Bank 应用程序中可以扩展 AppServer 节点以处理更多负载。在更新过程中，AppServer 安装、配置和启动脚本将在新扩大的虚拟机上运行。由于 Apache_LB 服务的 http_node_ips 和 appsrv_routes 属性依赖于 AppServer 节点，AppServer 群集大小的更改会影响 Apache_LB 服务并且会启动更新脚本运行。

您可在初始应用程序部署时为从属服务或应用程序组件定义 UPDATE 生命周期阶段脚本。还可以在更新过程中添加或修改更新脚本。在更新过程中配置更新脚本时，脚本将保存在更新配置文件中以供日后部署。

注意 您无法在更新过程中修改安装、配置或启动脚本。您只能配置更新脚本。

您也可以在命令行界面中启动更新过程来扩大已部署的应用程序。请参见第 200 页，“使用 CLI 部署和更新应用程序”。使用 Application Services 中的 REST API 可自动扩大已部署应用程序。请参见*《使用 Application Services REST API》*文档。

扩展部署的更新过程有时可能会失败。扩展更新失败后需要进行清理操作以删除新虚拟机。手动取消置备新创建的虚拟机是一项繁琐的任务，对于外部服务用户比较困难。要避免人工操作，可以考虑自动删除虚拟机。将以下标记设置为 True 可在扩展失败后启用自动清理虚拟机功能：

- VM_CLEANUP_AFTER_UPDATE_FAILURE

可以在部署之后设置此标记，并在稍后扩展或者回滚。如果不想自动取消置备虚拟机，请将此标记设置为 False。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 部署的应用程序必须至少包含一个群集节点。请参见第 130 页，“将节点指定为群集”。
- 如果计划添加自定义任务，则验证 Application Services 库中是否至少创建了一个自定义任务。请参见第 122 页，“将自定义任务添加到库”。
- 验证初始部署是否已成功部署到云环境。

  您不能从失败的部署或者在扩展操作失败后扩展群集节点。
- 请联系您的云管理员获取有关部署环境存储空间限制的信息。

步骤

1. 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。
2. 选择已成功部署的应用程序部署。
3. 从工具栏的**操作**下拉菜单中，选择**更新**。

   此时会打开“更新配置文件”页面。
4. 选择**创建更新配置文件**。

   此时将打开“更新配置文件”对话框。

5 在“更新类型”下拉菜单中，选择**扩大**。

6 为扩展的更新过程命名，添加可选描述并单击**确定**。

在描述中，可以添加有关此更新中所包括更改的信息。

7 （可选）单击蓝图映像以检查服务或应用程序组件之间突出显示的依赖关系。

记录所有从属组件，以便能够在更新脚本不存在时创建更新脚本或者修改现有更新脚本。

8 在**节点**选项卡中，增加一个或多个群集节点的群集大小值。

9 （可选）在**服务**或**应用程序组件**选项卡中，为属性绑定到群集节点的所有可用从属组件创建更新脚本。

如果定义了 UPDATE 生命周期阶段脚本，则脚本和相关联的服务或应用程序组件会一起出现。如果未定义这些脚本，请为更新过程创建相应的脚本。

例如，如果某个节点的 all(node_array:ip) 属性绑定到一个群集节点，则该节点必须运行更新脚本。

10 （可选）如果代理必须在更新脚本成功运行后重新启动虚拟机，请选中**重新引导**复选框，然后单击**下一步**。

扩大的群集节点和从属组件的更新任务会显示在执行计划中。原始执行计划不会在更新过程中出现。

11 （可选）将 APPD_UPDATE_PROPS 属性添加到更新脚本以查看所有已更改属性的列表。

更新脚本将保存到更新配置文件中并且可用于多个更新。不能将 APPD_UPDATE_PROPS 用作属性名称或者属性名称的限定符。

例如，示例 MySQL 服务更新脚本包含 APPD_UPDATE_PROPS 属性以更新数据库端口和密码。

12 （可选）将 APPD_PREV 属性添加到更新脚本可以查看属性的前一个值。

更新脚本将保存到更新配置文件中并且可用于多个更新。不能将 APPD_PREV 用作属性名称或者属性名称的限定符。

例如，示例 MySQL 服务更新脚本包含 APPD_PREV 属性以查看数据库密码的值。

13 （可选）单击**展开群集**按钮 ( ) 以展开群集节点、选择自定义任务并将任务拖动到每个节点。

您可以在“添加自定义任务”对话框中配置任务属性并保存更改。

14 在执行计划中，查看扩大的群集节点，如果适用则更新脚本，然后单击**下一步**。

执行计划中的蓝色点线定义部署任务运行的特定顺序。

单击服务更新脚本旁的向下箭头可查看脚本详细信息或脚本中使用的变量定义。

15 查看更新中的已修改属性和操作。

具有已定义更新脚本的从属属性会突出显示。

16 单击**更新**部署更新的应用程序。

更新过程会向云部署扩大的更新。

下一步

您可以在部署摘要页面查看部署的状态。请参见第 156 页，“使用部署摘要页面”。

了解当已部署应用程序在云中更新时后台所发生的各个进程。请参见第 157 页，“了解部署和更新过程”。

启动更新过程以缩小部署

利用 Application Services，可以为现有部署创建更新配置文件（例如缩小一个或多个节点的群集）以释放未使用的资源并在可扩展的应用程序中容纳不断变化的负载。缩小已部署应用程序的群集节点时，只能对应用程序蓝图中已建模为群集的节点的群集大小进行配置。

在缩小已部署应用程序的更新过程中，基于更新设置，将在从属组件上运行更新脚本，随后运行用于移除节点的卸载操作脚本。在多层应用程序中，如果某节点依赖于缩小的群集节点，则必须在从属节点上运行更新脚本。

例如，在部署的群集化的 Dukes Bank 应用程序中可以缩小 appServer 节点以降低负载。在更新过程中，将在从属的 load_balancer 节点上运行更新脚本。卸载操作脚本会从部署中移除 appServer_2 和 appServer_3 节点。此外，操作脚本还将调用云提供程序 API 来移除云环境中的节点。

在缩小期间，您可以将自定义任务附加到缩小的节点并执行更新。但是，当重用该缩小更新配置文件时，节点没有任何与之关联的自定义任务。这是因为，在首次缩小期间，自定义任务信息会随缩小的节点一起删除，因此不能用于缩小重用。

在初始应用程序部署过程中，可为相关的服务或应用程序组件定义 UPDATE 和 TEARDOWN 生命周期阶段脚本。更新脚本将在从属的节点上运行，卸载脚本将在缩小的节点上运行。还可以在更新过程中添加或修改更新脚本和卸载脚本。在更新过程中配置更新脚本或卸载脚本时，脚本将保存在更新配置文件中以供日后部署。

注意 您无法在更新过程中修改安装、配置或启动脚本。您只能配置更新脚本和卸载脚本。

使用 Application Services 中的 REST API 可自动缩小已部署应用程序。请参见*《使用 Application Services REST API》*文档。

缩小部署的更新过程有时可能会失败。缩小更新失败后需要进行清理操作以重新尝试取消置备新虚拟机。手动取消置备新创建的虚拟机是一项繁琐的任务，对于外部服务用户比较困难。要避免人工操作，可以考虑自动删除虚拟机。如果第一次尝试删除失败，则将以下标记设置为 True 可重新尝试清理虚拟机：

- UPDATE_RETRY_VM_DEPROVISIONING_AFTER_FAILURE_FLAG

将以下标记设置为 True 可在延迟时间（单位为毫秒）内重新尝试删除虚拟机：

- UPDATE_RETRY_VM_DEPROVISIONING_AFTER_FAILURE_DELAY_INTERVAL

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 部署的应用程序必须至少包含一个群集节点。请参见第 130 页，“将节点指定为群集”。
- 如果计划添加自定义任务，则验证 Application Services 库中是否至少创建了一个自定义任务。请参见第 122 页，“将自定义任务添加到库”。
- 验证初始部署是否已成功部署到云环境。

  您不能从失败的部署或者在扩展操作失败后扩展群集节点。
- 请联系您的云管理员获取有关部署环境存储空间限制的信息。

步骤

1. 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。
2. 选择已成功部署的应用程序部署。
3. 从工具栏的**操作**下拉菜单中，选择**更新**。

   此时会打开“更新配置文件”页面。

4 选择**创建更新配置文件**。

此时将打开“更新配置文件”对话框。

5 在“更新类型”下拉菜单中，选择**缩小**。

6 为扩展的更新过程命名，添加可选描述并单击**确定**。

在描述中，可以添加有关此更新中所包括更改的信息。

7 （可选）单击蓝图映像以检查服务或应用程序组件之间突出显示的依赖关系。

记录所有从属组件，以便能够在更新脚本不存在时创建更新脚本或者修改现有更新脚本。

8 在**节点**选项卡中，减小群集的大小值。

Application Services 将选择最后创建的节点。

例如，如果群集中包含 appserver_0、appserver_1、appserver_2 和 appserver_3 节点，并且您将群集大小减小到两个节点，则会选择 appserver_2 和 appserver_3 节点以供移除。

9 （可选）单击**选择要移除的节点**按钮可列出群集中的所有节点并手动选择要移除的节点。

10 （可选）在**服务**或**应用程序组件**选项卡中，为属性绑定到群集节点的所有可用从属组件创建更新脚本。

如果定义了 UPDATE 和 TEARDOWN 生命周期阶段，则脚本和相关联的服务或应用程序组件会一起出现。

例如，如果某个负载平衡器节点的 all(node_array:ip) 属性绑定到群集节点，则该节点必须运行更新脚本。

11 （可选）如果代理必须在更新脚本成功运行后重新启动虚拟机，请选中**重新引导**复选框，然后单击**下一步**。

缩小的群集节点和从属组件中的更新任务会显示在执行计划中。原始执行计划不会在更新过程中出现。

12 （可选）将 APPD_UPDATE_PROPS 属性添加到更新脚本以查看所有已更改属性的列表。

更新脚本将保存到更新配置文件中并且可用于多个更新。不能将 APPD_UPDATE_PROPS 用作属性名称或者属性名称的限定符。

例如，示例 MySQL 服务更新脚本包含 APPD_UPDATE_PROPS 属性以更新数据库端口和密码。

13 （可选）将 APPD_PREV 属性添加到更新脚本可以查看属性的前一个值。

更新脚本将保存到更新配置文件中并且可用于多个更新。不能将 APPD_PREV 用作属性名称或者属性名称的限定符。

例如，示例 MySQL 服务更新脚本包含 APPD_PREV 属性以查看数据库密码的值。

14 （可选）单击**展开群集**按钮 ( ) 以展开群集节点、选择自定义任务并将任务拖动到每个节点。

您可以在“添加自定义任务”对话框中配置任务属性并保存更改。

15 查看缩小的群集节点并更新执行计划中的脚本，然后单击**下一步**。

执行计划中的蓝色点线定义部署任务运行的特定顺序。

单击服务更新脚本旁的向下箭头可查看脚本详细信息或脚本中使用的变量定义。

16 查看更新中的已修改属性和操作。

具有已定义更新脚本的从属属性会突出显示。

17 单击**更新**部署更新的应用程序。

更新过程会向云部署缩小的更新。

下一步

您可以在部署摘要页面查看部署的状态。请参见第 156 页，“使用部署摘要页面”。

了解当已部署应用程序在云中更新时后台所发生的各个进程。请参见第 157 页，“了解部署和更新过程”。

启动更新过程以修改配置

您可以启动更新过程来修改已部署应用程序（如 Tomcat 和 MYSQL）或应用程序组件（如 WAR 和 SQL）中现有服务的配置和代码。修改已部署应用程序的配置时，只能配置应用程序的属性值。

在初始部署之后，如果修改现有的应用程序组件或服务，或修改拥有从属属性的组件，则必须在所有受影响的组件和从属组件上运行更新脚本。您可在初始应用程序部署中或更新过程中为服务或应用程序组件定义 UPDATE 和 ROLLBACK 生命周期阶段脚本。

例如，在群集化的 Dukes Bank 应用程序中，如果您修改 MYSQL 服务的 db_port 属性，则在部署更新过程中，更新脚本会在 MYSQL 服务上运行。Appserver 节点上的 Dukes_Bank_App 应用程序组件也会运行更新脚本，因为 db_port 属性已绑定到 MYSQL 服务的 db_port 属性。

注意 您无法在更新过程中修改安装、配置或启动脚本。您只能配置更新或回滚脚本。

您也可以在命令行界面中启动更新过程来修改配置以及服务或应用程序组件的代码。请参见第 200 页，“使用 CLI 部署和更新应用程序”。使用 Application Services 中的 REST API 可自动修改已部署应用程序的配置。请参见*《使用 Application Services REST API》*文档。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 如果计划添加自定义任务，则验证 Application Services 库中是否至少创建了一个自定义任务。请参见第 122 页，“将自定义任务添加到库”。
- 验证初始部署是否已成功部署到云环境。

  不能从失败的部署中启动更新过程来修改现有服务的配置和代码以扩展群集节点。
- 部署的应用程序必须至少具有一个部署时可替代的服务属性或应用程序组件属性。

步骤

1. 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。
2. 选择已成功部署的应用程序部署。
3. 从工具栏的**操作**下拉菜单中，选择**更新**。

   此时会打开“更新配置文件”页面。
4. 选择**创建更新配置文件**。

   此时将打开“更新配置文件”对话框。
5. 在“更新类型”下拉菜单中，选择**配置**。
6. 为配置更新过程命名，添加描述并单击**确定**。

   在描述中，可以添加有关此更新中所包括更改的信息。
7. 单击蓝图映像以检查服务或应用程序组件之间突出显示的依赖关系。

   记录所有从属组件，以便能够在更新脚本不存在时创建更新脚本或者修改现有更新脚本。

8 在**服务**或**应用程序组件**选项卡中，修改适用的属性并更新脚本。

如果定义了 UPDATE 和 ROLLBACK 生命周期阶段，则脚本和相关联的服务或应用程序组件将一起显示。您必须为已更改的组件和从属组件创建更新脚本。

就群集节点而言，如果修改了服务或应用程序组件的属性，则这些更改将应用到该群集的所有节点。

例如，在群集化的 Dukes Bank App 中，如果您更改了 initialize_db_script 应用程序组件中的 db_password 属性，则 Dukes_Bank_App 中的 db_password 属性会依赖于 initialize_db_script 应用程序组件。两者都会运行更新脚本，因为 initialize_db_script 是已更改的组件，Dukes_Bank_App 是受影响的组件。Dukes_Bank_App 的更新脚本将在 Appserver 群集的所有节点上运行。

9 （可选）如果代理必须在更新脚本成功运行后重新启动虚拟机，请选中**重新引导**复选框，然后单击**下一步**。

已更改组件和受影响组件的更新任务将显示在执行计划中。原始执行计划不会在更新过程中出现。

10 （可选）将 APPD_UPDATE_PROPS 属性添加到更新脚本以查看所有已更改属性的列表。

更新脚本不会保存，且该脚本仅适用于该特定更新过程。不能将 APPD_UPDATE_PROPS 用作属性名称或者属性名称的限定符。

例如，示例 MySQL 服务更新脚本包含 APPD_UPDATE_PROPS 属性以更新数据库端口和密码。

11 （可选）将 APPD_PREV 属性添加到更新脚本可以查看属性的前一个值。

更新脚本不会保存，且该脚本仅适用于该特定更新过程。不能将 APPD_PREV 用作属性名称或者属性名称的限定符。

例如，示例 MySQL 服务更新脚本包含 APPD_PREV 属性以查看数据库密码的值。

12 （可选）单击**展开群集**按钮 ( ) 以展开群集节点、选择自定义任务并将任务拖动到每个节点上。

如果修改群集节点上的服务属性或应用程序组件属性，则更改将应用于所有节点。您可以在“添加自定义任务”对话框中配置任务属性并保存更改。

13 检查更新执行计划中已修改的群集节点和更新脚本。

执行计划中的蓝色点线定义了部署任务将运行的特定顺序。

单击配置更新脚本旁的向下箭头以查看脚本详细信息或脚本中使用的变量定义。

14 检查回滚执行计划中已修改的群集节点和更新脚本，然后单击**下一步**。

执行计划中的蓝色点线定义了部署任务将运行的特定顺序。

单击配置更新脚本旁的向下箭头以查看脚本详细信息或脚本中使用的变量定义。

15 查看更新中更改及受影响的组件。

已修改属性和从属属性会突出显示。

16 单击**更新**部署应用程序的已修改配置。

更新过程会将已配置的部署部署到云中。

下一步

检查部署摘要页面中的部署状态。请参见第 156 页，“使用部署摘要页面”。

了解当已部署应用程序在云中更新时后台所发生的进程。请参见第 157 页，“了解部署和更新过程”。

使用现有的更新配置文件

部署者可以使用现有的更新配置文件对之前在同一云环境中部署的应用程序进行多次更新。更新配置文件会保存初次更新过程中所使用的公用属性和脚本。

您可以保存更新配置文件并只能将其应用到部署的一个应用程序版本。

CLI 不支持保存新的或现有的更新配置文件。如果使用 CLI 应用现有的更新配置文件，则无法在启动更新过程修改配置或扩展群集节点时替代脚本或属性。

使用 CLI 部署应用程序时，如果选择某个更新配置文件，则该更新配置文件名称的自动完成选项将不会显示。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证 Application Services 中是否至少有一个部署已成功更新。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 选择已成功更新的部署。

   此时将打开部署摘要页面。

3 在部署摘要上方的工具栏中，选择**部署视图 > 更新视图**。

   更新摘要页面上会显示与部署关联的所有详细信息，如可用更新配置文件、应用程序名称和版本。

4 找到要使用的更新配置文件类型并单击。

   该更新配置文件可用于扩大、缩小或配置更新过程。每个更新配置文件都会显示该配置文件用于更新（无论成功与否）部署的次数。

   此时会打开部署配置文件向导。

下一步

完成更新部署的任务。请参见第 162 页，“启动更新过程以扩大部署”、第 164 页，“启动更新过程以缩小部署”和第 166 页，“启动更新过程以修改配置”。

升级更新配置文件

部署者可以升级现有更新配置文件，以更新任何跨多个部署环境的部署。您可以将同一类型的更新过程应用到多个源自同一应用程序版本的部署。

例如，部署者可以将与安全相关的配置更新应用到应用程序的一个或多个组件，该应用程序位于源自同一应用程序版本的不同部署上。

不支持通过 REST API 跨不同部署环境升级更新配置文件。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 选择已成功更新的部署。

   此时将打开部署摘要页面。

3 在部署摘要上方的工具栏中，选择**部署视图 > 更新视图**。

更新摘要页面上会显示与部署关联的所有详细信息，如可用更新配置文件、应用程序名称和版本。

4 找到要升级的现有更新配置文件。

5 要将更新配置文件升级到一个或多个部署环境，请单击**升级更新配置文件**图标。

此时将打开“管理更新升级”对话框，其中显示了可用的部署环境列表。

6 在“可用”列中，选择部署环境，然后单击**添加到所选**箭头。

该部署环境会显示在“选定”列中。

7 （可选）要从“选定”列中撤销某个部署环境，请选择该部署环境，然后单击**从所选中移除**箭头。

该部署环境会显示在“可用”列中。

8 查看“选定”列中的部署环境，然后单击**提交**。

更新配置文件成功升级到选定部署环境后，更新配置文件摘要卡上会显示绿色的“竖起大拇指”图标。

通知图标 ( ) 会显示在单个或多个部署环境中的部署上。例如，如果群集化的 Dukes Bank 更新配置文件已升级，则通知会显示在选定部署环境内的所有现有群集化的 Dukes Bank 应用程序部署上。

下一步

将升级的更新配置文件应用到属于其中一个选定部署环境的部署。请参见第 169 页，“接受或放弃升级的更新配置文件”。

接受或放弃升级的更新配置文件

部署者可单击通知以查看升级的更新配置文件的组件，并确定是接受还是放弃更改。如果接受更改，则 Application Services 会在应用这些更改之前检查升级的更新配置文件和部署之间的兼容性。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否至少有一个更新配置文件已升级至一个或多个部署环境。请参见第 168 页，“升级更新配置文件”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 找到带有通知图标 ( ) 的部署并单击该图标。

此时会打开更新摘要页面。

3 单击**应用**。

Application Services 会检查升级的更新配置文件和当前更新后部署之间的兼容性。如果存在严重的不兼容问题，您可以取消或放弃通知。如果不兼容问题并不严重（例如部署中存在同名的更新配置文件），您可以继续应用，也可以取消或放弃通知。

4 在向导中查看“应用程序属性”选项卡和“执行计划”选项卡中的新值或配置，并选择**应用更新**。

5 （可选）如果您不希望应用升级的更新配置文件中的新值或配置，请选择**放弃更新**。

升级的更新配置文件会被放弃，通知图标 ( ) 也会从部署中删除。

下一步

您可以选择其他包含通知的部署，继续查看和应用，或者放弃升级的更新配置文件。

回滚更新过程

凭借 Application Services，如果修改配置的更新过程失败，或更新部署发生性能或可用性问题，您可以使用回滚功能将该更新还原到时间轴中的上次成功状态。

回滚脚本在其属性正在发生更改的组件和其他从属节点上运行。例如，在部署的群集化的 Dukes Bank App 中，当您将 MySQL 端口数的配置从 3306 修改为 3307 时，回滚脚本将在 mysql 服务和 dukes_bank_app 组件上运行。如果此更新过程失败，可以回滚配置更新以恢复到之前的端口值 3306。在回滚向导中，可以查看当前值 3307 和回滚到的值 3306。回滚脚本在 mysql 服务和 dukes_bank_app 组件上运行以完成此过程。

在初始应用程序部署期间为从属服务或应用程序组件定义 ROLLBACK 生命周期阶段脚本。回滚更新过程以修改配置时，也可以添加或修改回滚脚本。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 验证初始部署是否已成功部署到云环境。

步骤

1. 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。
2. 选择要回滚到之前成功状态的应用程序部署。
3. 从工具栏的**操作**下拉菜单中，选择**回滚**。
4. 单击蓝图映像以检查服务或应用程序组件之间突出显示的依赖关系。

   记录所有从属组件，以便能够在回滚脚本不存在时创建回滚脚本或者修改现有回滚脚本。
5. 从**服务**或**应用程序组件**选项卡中，修改相应属性和回滚脚本。

   如果定义了 ROLLBACK 生命周期阶段，则脚本将与关联的服务或应用程序组件一起显示。您必须为更改和回滚到的组件创建回滚脚本。

   就群集节点而言，如果修改了服务或应用程序组件的属性，则这些更改将应用到该群集的所有节点。

   如果更新了由外部服务实例中的更改触发的部署的配置属性，则无法回滚这些更新属性。
6. （可选）如果在回滚脚本成功运行后代理必须重新启动虚拟机，则选中**重新引导**复选框，然后单击**下一步**。

   在执行计划中会显示已更改和回滚到的组件的回滚任务。在回滚过程中不显示原始执行计划。
7. 检查修改的群集节点和执行计划中的回滚脚本，然后单击**下一步**。

   执行计划中的蓝色点线定义了部署任务将运行的特定顺序。

   单击配置回滚脚本旁边的向下箭头来查看脚本详细信息或脚本中使用的变量定义。
8. 检查更新中的已更改和回滚到的组件。

   突出显示已更改和回滚到的属性。
9. 单击**回滚**。

回滚过程将部署恢复到云中的上次成功状态。

下一步

检查部署摘要页面中的部署状态。请参见第 156 页，“使用部署摘要页面”。

了解运行自定义任务更新

通过运行自定义任务更新，您可以在现有部署的虚拟机上运行一系列任意的自定义任务。

通过运行自定义任务更新，您可以执行以下任务。

- **在虚拟机级别操作**。允许您同时在组件和服务的若干个组件上操作。
- **在不更改属性的情况下执行脚本**。允许您执行脚本，而无需更改现有部署中的任何服务或组件属性。

要通过 REST API 触发运行自定义任务更新，必须指定以下信息。

- 需要执行的所有任务。
- 每个任务的输入参数。
- 在其上执行每个任务的节点。
- 节点中每个任务的执行依赖关系。例如，可能有一个或多个任务会等待其他任务执行并完成。

运行自定义任务更新在启动时会显示在部署概览页面中。部署时间轴有一个节点表示运行自定义任务更新的当前执行状况。

运行自定义任务支持回滚操作。自定义任务无法执行或要恢复之前所做的更改时，必须执行另一个自定义任务以恢复更改。这是因为 Application Services 无法为任意自定义任务生成回滚脚本。

引入了两个新的 REST API 来管理运行自定义任务。

- 创建和调度运行自定义任务更新配置文件
- 更新和调度运行自定义任务更新配置文件

有关 REST API 的详细信息，请参见*《使用 Application Services REST API》*。

对失败的扩展部署更新过程进行故障排除

扩展部署更新过程失败时，已知解决方案和建议可以为您提供帮助。

问题

扩展部署更新过程失败。

原因

您可能正尝试启动先前扩展失败的扩展部署的更新过程。

解决方案

1 云管理员必须删除部署的所有群集节点的新虚拟机。

2 云管理员必须删除缩小更新过程中未移除的虚拟机。

3 针对先前失败的部署，启动另一个扩大更新过程。

   Application Services 会预填充之前采用的成功的属性值。新值则来自之前失败的部署。

   例如，如果您启动了更新过程，将 AppServer1 群集节点从 1 个节点扩展到 2 个，将 AppServer2 群集节点从 1 个节点扩展到 3 个，但 AppServer2 节点的更新过程失败。在第二个更新过程中，AppServer1 的值填充为 2，而对于 AppServer2 节点，您可以指定大于前一个值的值。

4 如果部署失败且虚拟机被移除，则启动另一个缩小更新过程。

5 启动另一个更新过程来扩大所有失败的节点，以便更新脚本在所有从属组件（如负载平衡器）上运行，从而使组件有效。

对失败的修改配置更新过程进行故障排除

您可以在修改配置的更新过程失败时使用某些解决方案。

问题

修改配置的更新过程失败。

原因

针对上一次失败的更新，您可以通过重置失败的属性（如端口号）并尝试继续操作来试图启动修改配置的更新过程。

解决方案

- 针对上一次失败的更新，启动另一个修改配置的更新过程。

  Application Services 会预填充之前采用的成功的属性值。新值则来自之前失败的部署。

  当您启动其他修改配置的更新过程时，必须手动更新所有失败的属性。

- 创建或修改更新脚本以忽略失败的属性。

  如果您不希望更新所有失败的属性，则可以通过编写更新脚本来忽略失败的属性并继续更新过程。

  不会保存修改的更新脚本，且该脚本仅适用于该特定更新过程。

部署预定义库组件 18

Application Services 库包括预定义示例服务、逻辑模板、应用程序组件和任务。

示例应用程序包含基于 Windows 或 Linux 的逻辑模板、服务和专为此应用程序配置的应用程序组件。在预定义应用程序中只能配置服务的操作脚本，然后将其部署到支持的云环境中。创建用于部署示例应用程序的部署配置文件时，可将兼容的自定义任务添加到执行计划中。

重要事项 预定义示例模板、应用程序组件、服务和任务只能在测试环境中使用。如果计划在生产环境中使用示例模板，请确保在部署前将最新的安全修补程序应用到此模板中的操作系统。

分配给 Application Services 默认组的所有用户帐户都可以访问和部署示例应用程序。

为库中的示例模板和服务渲染的所有图标都基于模板或服务的名称。例如，官方的 **MySQL** 图标显示在库中 MySQL 服务的旁边。如果服务或模板的名称唯一，则将显示通用图标。

熟悉有关部署示例库组件的主题中经常出现的主要概念。请参见第 16 页，“主要概念”。

本章讨论了以下主题：

- 第 173 页，“使用示例模板”
- 第 174 页，“可用于部署的示例应用程序”
- 第 179 页，“更新已部署的示例应用程序”
- 第 181 页，“示例 Application Services 库任务”
- 第 186 页，“示例库服务”

使用示例模板

Application Services 提供已安装操作系统并具有用于部署应用程序的所有必需的库的 CentOS 和 Ubuntu 示例模板。

例如，要使用 CentOS 示例模板，请从 VMware 产品下载站点下载以下文件。

- CentOS 6.3 32 位 OVF
- CentOS 6.3 32 位 VMDK

有关可用示例模板的列表，请参见 Application Services 自述文件。

将 CentOS 6.3 32 位 OVF 软件包上载到 vCloud Director 目录或 vRealize Automation。有关 vCloud Director 目录或 vRealize Automation 的详细信息，请参见相应文档。

注意 对于 CentOS 32 位示例模板，未启用物理地址扩展 (PAE)，因此最多可以为虚拟机分配 3.25GB 的 RAM。已为 Ubuntu 32 位示例模板启用 PAE，因此可以为虚拟机分配 4GB 以上的 RAM。

将 libpython 软件包添加到 Ubuntu 模板

必须添加 libpython XML 软件包，才能使示例 Ubuntu 模板正常工作。

步骤

1 从 Ubuntu 镜像站点下载 python 软件包。

```
wget http://UbuntuMirrorSite/libpython2.7_2.7.3-0ubuntu3.4_amd64.deb
```

2 将 python 软件包安装到 Ubuntu 模板。

```
dpkg -i libpython2.7_2.7.3-0ubuntu3.4_amd64.deb
```

为 Ubuntu 虚拟机启用 SSH 连接

Ubuntu 虚拟机必须具有生成的 SSH 主机密钥才能正常运行。

前提条件

验证是否安装了与浏览器一起使用的 vCloud Director VMRC 插件。有关兼容的 Web 浏览器的信息，请参见第 22 页，“Application Services 系统要求”。

步骤

1 将 Ubuntu OVF 软件包上载到 vCloud Director 目录。

2 在 vCloud Director VMRC 控制台中单击。

3 登录到 Ubuntu 虚拟机。

4 启用 SSH 连接。

```
ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
"ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key"
```

5 重新启动 SSH 服务器。

```
service ssh restart
```

此时将启用与 Ubuntu 虚拟机的 SSH 连接。

可用于部署的示例应用程序

您可以使用预定义的单层或三层应用程序来部署简单的 Web 应用程序或复杂的电子邮件套件。示例应用程序的应用程序蓝图包含逻辑模板、服务、外部服务和应用程序组件，部署到云之后，它们都将发挥全部功能。

应用程序架构师可以使用 Application Services 示例应用程序作为建立自定义应用程序模型和创建自定义应用程序的基础。

部署 Nanotrader 应用程序

Nanotrader 应用程序示例是一个三层 Web 应用程序，该应用程序可使用 vFabric Web Server、vFabric tc Server、vFabric RabbitMQ 和 vFabric SQLFire 组件来部署 NanoTrader 应用程序。

仅能在测试环境中使用预定义的示例应用程序。

前提条件

- 将已上载的 CentOS 6.3 32 位模板注册到云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”、第 55 页，“注册 vRealize Automation 云提供程序和模板”或第 85 页，“注册 Amazon EC2 云提供程序和模板”。

- 将云模板映射到 CentOS63 32 位逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。
- 以**应用程序架构师**和**应用程序发布者和部署者**身份登录到 Application Services。
- 了解创建部署配置文件和部署应用程序的基本概念。请参见第 141 页，第 16 章“部署应用程序”。
- 验证 `spring-nanotrader-asynch-services.war`、`spring-nanotrader-services.war`、`spring-nanotrader-web.tgz` 和 `spring-nanotrader-web.war` 文件是否已下载到 HTTPD 服务器。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 选择 **Nanotrader**。

3 单击屏幕顶部的“蓝图”屏幕截图。

4 在蓝图编辑器中，选择服务和 SCRIPT 组件，并添加 URL 值以下载适用的文件。

服务和组件	描述
vFabric_Web_Server	`spring-nanotrader-web.tgz` 文件用于 deployment_archive 属性
vFabric_tc_Server	`spring-nanotrader-web.war` 文件用于 external_template 属性
spring-nanotrader-services	`spring-nanotrader-services.war` 文件用于 war_file 属性

5 单击**保存**。

6 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

7 打开您配置的 Nanotrader 应用程序。

8 创建部署配置文件并部署应用程序。

9 要访问部署的应用程序，请打开支持的 Web 浏览器，并键入 **`http://`*`Load_Balancer_IP`*`/#login`** URL。

   其中，*Load_Balancer_IP* 是已部署的负载平衡器的 IP 地址。

10 创建 NanoTrader 帐户以设置用户名和密码。

下一步

启动更新过程以扩展或修改 Nanotrader 应用程序的配置。请参见第 179 页，“启动更新过程以扩展示例应用程序”和第 180 页，“启动更新过程以修改示例应用程序中的配置”。

部署群集化的 DotShoppingCart 应用程序

群集化的 DotShoppingCart 是一个基于 Windows 的三层示例应用程序，该应用程序将 Microsoft SQL Server 用作数据库服务器、将 Microsoft IIS Server 用作应用程序服务器，以及将 Apache HTTP Server 用作负载平衡器。

Create_DB 脚本会在 SQL Server 数据库中创建并初始化 DotShoppingCart 数据库。DotShoppingCart_App 脚本会在 Microsoft IIS 服务器中安装 DotShoppingCart 应用程序并使用 DotShoppingCart 数据库。可以将 Microsoft IIS AppServer 节点定义为群集，以便扩展到多个节点以处理更高的负载。Apache HTTP 服务器用于处理负载平衡。

仅能在测试环境中使用预定义的示例应用程序。

前提条件

- 向云提供程序注册 Windows Server 2008 R2 SP1 vApp 模板。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”。
- 将云模板映射到 Windows Server 2008 R2 SP1 逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。

- 以**应用程序架构师**和**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否已下载 DotShoppingCart 3.0（开放源版本）软件包 `OpenSourceEdition.V3.zip` 文件。必须将 Web 服务器配置为使用 HTTP 协议传输文件。标识用于在 Web 浏览器中访问文件的 URL。
- 了解创建部署配置文件和部署应用程序的基本概念。请参见第 141 页，第 16 章“部署应用程序”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 选择**群集化的 DotShoppingCart**。

3 单击屏幕顶部的“蓝图”屏幕截图。

4 在蓝图编辑器中，选择 DotShoppingCart_App 应用程序组件，并为 DSC_ZIP 属性值键入有效的 URL。

   部署期间，AppServer Windows 虚拟机中的 Application Services 代理必须能够访问 `OpenSourceEdition.V3.zip` 文件的有效 URL。

5 （可选）如果计划创建多个 AppServer 节点，请选择 **AppServer** 并增加群集大小。

6 创建部署配置文件并部署应用程序。

7 要访问部署的应用程序，请打开支持的 Web 浏览器，然后键入 **`http://`*`Load_Balancer_IP`*`:8081`** URL。

   其中，*Load_Balancer_IP* 是已部署的负载平衡器的 IP 地址。

下一步

启动更新过程以扩展或修改群集化的 DotShoppingCart 应用程序配置。请参见第 179 页，“启动更新过程以扩展示例应用程序”和第 180 页，“启动更新过程以修改示例应用程序中的配置”。

部署群集化的 Dukes Bank 应用程序

群集化的 Dukes Bank 应用程序是一个三层 Application Services 示例应用程序，该应用程序将 MySQL 或 Microsoft SQL Server 用作其数据库、将 JBoss Server 用作其应用程序服务器，而将 Apache HTTP 服务器用作可以部署到测试环境的负载平衡器。

initialize_db_script 脚本可在 MySQL 或 SQL Server 中创建并初始化 dukes_db 数据库。Dukes_Bank_App EAR 应用程序组件使用在 JBoss 服务器中部署的 dukes_db 数据库。可以将 JBoss appserver 节点定义为群集，以便扩展到多个节点以处理更高的负载。Apache HTTP 服务器用于处理负载平衡。

仅能在测试环境中使用预定义的示例应用程序。

前提条件

- 向云提供程序注册已上载的 CentOS 6.3 32 位 Linux 模板或 Windows Server 2008 R2 SP1 vApp 模板。请参见第 173 页，“使用示例模板”和第 75 页，“注册 vCloud Director 云提供程序和模板”。
- 将云模板映射到 CentOS63 32 位或 Windows Server 2008 R2 SP1 逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。
- 以**应用程序架构师**和**应用程序发布者和部署者**身份登录到 Application Services。
- 对于基于 Linux 的群集化的 Dukes Bank，如果系统需要代理才能访问 Internet，请验证您的代理连接是否已正确配置。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 对于基于 Windows 的群集化的 Dukes Bank，请验证 Windows 虚拟机是否具有一个可以在模板中使用的有效 JRE 安装路径。
- 了解创建部署配置文件、部署应用程序和更新已部署应用程序的基本概念。请参见第 141 页，第 16 章“部署应用程序”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击**群集化的 Dukes Bank 应用程序**。

3 选择应用程序版本。

选项	操作
基于 Linux 的群集化的 Dukes Bank	选择版本 2.1.0。
基于 Windows 的群集化的 Dukes Bank	选择版本 3.0.0。

4 单击屏幕顶部的“蓝图”屏幕截图。

5 在蓝图编辑器中，选择 JBossAppServer 服务并键入相应的属性值。

- 对于基于 Windows 的群集化的 Dukes Bank，请更新 JAVA_INSTALL_DIR 属性。
- 对于基于 Linux 的群集化的 Dukes Bank，请更新 JAVA_HOME 属性。

部署期间，JBossAppServer 服务中的 Application Services 代理应可以访问安装的 JRE。

6 对于基于 Linux 的群集化的 Dukes Bank，如果云模板位于没有代理的专用网络中，则在部署应用程序时，应向部署配置文件中的各个虚拟机添加 YUM 配置任务，然后设置 repository_url 属性以使用在专用网络中托管的 CentOS 6.3 32 位存储库。

7 创建部署配置文件并部署应用程序。

8 要访问部署的应用程序，请打开支持的 Web 浏览器，然后键入 **`http://`*`Load_Balancer_IP`*`:8081/bank/main.faces`** URL。

其中，*Load_Balancer_IP* 是已部署的负载平衡器的 IP 地址。

9 使用登录凭据 **`200`** 和密码 **`foobar`** 可访问应用程序。

下一步

启动更新过程以扩展或修改群集化的 Dukes Bank 3.0.0 或 2.1.0 应用程序版本的配置。请参见第 179 页，“启动更新过程以扩展示例应用程序”和第 180 页，“启动更新过程以修改示例应用程序中的配置”。

部署 jPetStore 应用程序

jPetStore 1.0.0 是单层示例 Web 存储应用程序，可使用 SQLFire 作为可部署到测试环境的数据库在 vFabric tc Server 上部署 jPetStore App WAR 文件。

要部署 jPetStore 应用程序的蓝图，您必须下载 VMware vFabric SQLFire 1.0.0 安装程序 (`vFabric_SQLFire_10_Installer.jar`) 文件，并使用 SQLFire 服务属性 `SQLFire10_Installer` 指向该文件。

仅能在测试环境中使用预定义的示例应用程序。

前提条件

- 向云提供程序注册 jPetStore 1.0.0 的已上载 CentOS 6.3 32 位模板。请参见第 173 页，“使用示例模板”和第 75 页，“注册 vCloud Director 云提供程序和模板”。
- 将云模板映射到 CentOS63 32 位磁盘逻辑模板 jPetStore 1.0.0。请参见第 115 页，“将逻辑模板添加到库”。
- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 如果系统需要代理才能访问 Internet，请验证您的代理连接是否已正确配置。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 了解创建部署配置文件和部署应用程序的基本概念。请参见第 141 页，第 16 章“部署应用程序”。

步骤

1 从 VMware 下载页面 (https://my.vmware.com/web/vmware/downloads) 将 vFabric SQLFire 1.0.0 安装程序包 (`vFabric_SQLFire_10_Installer.jar`) 下载到您自己的 `httpd` 服务器或 Application Services 服务器上的 `httpd` 服务器。

   如果使用的是 Application Services 服务器上的 `httpd` 服务器，请将 `vFabric_SQLFire_10_Installer.jar` 文件下载到 `/opt/vmware/darwin/htdocs/artifacts/services/sqlfire` 文件夹中。

   要下载此安装程序包，您必须输入客户帐户凭据。

2 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

3 单击 **jPetStore** 应用程序以打开 jPetStore 1.0.0 版。

4 单击**查看蓝图**图像以打开“jPetStore v1.0.0 - 蓝图”页面。

   a 单击 jPetStore 蓝图图像中的 **SQLFire**。

   b 单击包含**详细信息**、**属性**和**操作**选项卡的行中的**属性**选项卡。

   c 单击包含 `SQLFire10_Installer` 属性名称的行中的“编辑”( ) 图标。

   d 在编辑 SQLFire 的属性页面中使用以下选项之一指定一个蓝图值。

      - 单击**使用库值**以接受默认库值 `http://${darwin.content.server.ip}/artifacts/services/sqlfire/vFabric_SQLFire-1.0.jar`。此库值指向 Application Services 服务器上的 httpd 服务器。

        注意 要使用此库值，从 VMware 下载页面下载的 `vFabric_SQLFire_10_Installer.jar` 必须位于 `/opt/vmware/darwin/htdocs/artifacts/services/sqlfire`，并且名称为 `vFabric_SQLFire-1.0.jar`。

      - 单击**键入您自己的值**文本框，然后输入另一个 httpd 位置上的 `vFabric_SQLFire_10_Installer.jar` 文件路径。

   e 单击**保存**以关闭编辑 SQLFire 的属性页面。

5 单击**部署**以部署此蓝图。

6 创建部署配置文件。

7 部署应用程序。

8 要访问已部署的应用程序，请打开受支持的 Web 浏览器并在地址栏中输入 http://jPetStore_VM_IP:8080/jpetstore-1.0.0。

部署 Radiant CMS 应用程序

Radiant CMS 是一个单节点应用程序，用于部署 Ruby On Rails 服务和 MySQL 数据库，并安装配置为使用 MySQL 数据库的 Radiant 应用程序。

仅能在测试环境中使用预定义的示例任务。

前提条件

- 将已上载的 CentOS 6.3 32 位模板注册到云提供程序。请参见第 75 页，“注册 vCloud Director 云提供程序和模板”、第 55 页，“注册 vRealize Automation 云提供程序和模板”或第 85 页，“注册 Amazon EC2 云提供程序和模板”。
- 将云模板映射到 CentOS63 32 位逻辑模板。请参见第 115 页，“将逻辑模板添加到库”。
- 以**应用程序发布者和部署者**身份登录到 Application Services。

- 验证 Application Services 是否已配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 了解创建部署配置文件和部署应用程序的基本概念。请参见第 141 页，第 16 章“部署应用程序”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击 **Radiant CMS**。

此应用程序的所有服务和应用程序组件都是预定义的，无需另行配置。

3 单击屏幕顶部的“蓝图”屏幕截图。

4 创建部署配置文件并部署应用程序。

5 要访问此应用程序，在支持的 Web 浏览器中键入 Radiant CMS 虚拟机 IP 地址 http://*VM_IP*/ 然后使用 Radiant 默认的 admin 用户名和 Radiant 密码登录。

更新已部署的示例应用程序

更新现有示例应用程序部署会启动该更新所需的新更改值的捕获过程。

您可以扩展节点的群集或修改现有服务和应用程序组件的配置及代码，从而为单层或三层示例应用程序启动更新过程。

启动更新过程以扩展示例应用程序

您可以通过扩展节点的群集多次对预定义的示例应用程序部署启动更新过程。扩大或缩小已部署应用程序的群集节点时，只能对应用程序蓝图中已建模为群集的节点的群集大小进行配置。

您可以扩展以下预定义的示例应用程序。

- Nanotrader 应用程序
- 群集化的 DotShoppingCart 应用程序
- 群集化的 Dukes Bank 3.0.0 或 2.1.0 应用程序版本

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 部署的应用程序必须至少包含一个群集节点。请参见第 130 页，“将节点指定为群集”。
- 验证初始部署是否已成功部署到云环境。

  您不能从失败的部署或者在扩展操作失败后扩展群集节点。
- 请联系您的云管理员获取有关部署环境存储空间限制的信息。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 选择一个成功部署的预定义的示例应用程序。

3 从工具栏的**操作**下拉菜单中，选择**更新**。

此时会打开“更新配置文件”页面。

4 选择**创建更新配置文件**。

此时将打开“更新配置文件”对话框。

5 在“更新类型”下拉菜单中，选择**扩大**或**缩小**。

6 为扩展的更新过程命名，添加可选描述并单击**确定**。

在描述中，可以添加有关此更新中所包括更改的信息。

7 从**节点**选项卡中，增加一个或多个群集节点的群集大小值，并单击**下一步**。

8 查看执行计划中的扩展群集节点，然后单击**下一步**。

执行计划中的蓝色点线定义部署任务运行的特定顺序。

9 查看更新中的已修改属性。

受影响的属性会突出显示。

10 单击**更新**部署更新的应用程序。

下一步

您可以在部署摘要页面查看部署的状态。请参见第 156 页，“使用部署摘要页面”。

启动更新过程以修改示例应用程序中的配置

您可以启动更新过程以便修改已部署示例应用程序中现有服务和应用程序组件的配置和代码。修改已部署应用程序的配置时，只能配置应用程序的属性值。

您可以修改以下预定义示例应用程序的配置和代码。

- Nanotrader 应用程序
- 群集化的 DotShoppingCart 应用程序
- 群集化的 Dukes Bank 3.0.0 或 2.1.0 应用程序版本

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 熟悉定义和配置组件属性与操作的基本概念。请参见第 89 页，第 12 章“开发 Application Services 组件”。
- 验证初始部署是否已成功部署到云环境。

  不能从失败的部署中启动更新过程来修改现有服务的配置和代码以扩展群集节点。
- 部署的应用程序必须至少具有一个部署时可替代的服务属性或应用程序组件属性。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 选择成功部署的预定义的示例应用程序。

3 从工具栏的**操作**下拉菜单中，选择**更新**。

此时会打开“更新配置文件”页面。

4 选择**创建更新配置文件**。

此时将打开“更新配置文件”对话框。

5 在“更新类型”下拉菜单中，选择**配置**。

6 为配置更新过程命名，添加描述并单击**确定**。

在描述中，可以添加有关此更新中所包括更改的信息。

7 在**服务**选项卡中，为您的示例应用程序修改适用的属性。

示例应用程序	服务版本	属性
Nanotrader	vFabric_Web_Server 5.1.1	http_port、deployment_archive 和 webserver_conf_file
	vFabric_tc_Server 2.7.1	port、db_ip、db_port 和 jdbc_url
	vFabric_SQLFire_Server 1.0.3	schema_file 和 dataload_file
群集化的 DotShoppingCart	Apache_LB 2.2.22	http_port
	SQL_Server_2008 1.0.0	SA_PWD
群集化的 Dukes Bank 3.0.0	Apache_LB 2.2.22	http_port 和 http_proxy_port
	JBossAppServer 5.1.0	JBOSS_JMX_USER 和 JBOSS_JMX_PWD
	SQL_Server_2008 1.0.0	SA_PWD
	initialize_db_script	db_username 和 db_password
	Dukes_Bank_App	JAR_FILE、EAR_FILE、db_password 和 db_user
群集化的 Dukes Bank 2.1.0	Apache_LB 2.2.22	http_port 和 http_proxy_port
	JBossAppServer 5.1.0	JBOSS_JMX_USER 和 JBOSS_JMX_PWD
	MySQL 5.0.0	db_port 和 db_root_password
	initialize_db_script	db_username、db_password、init_db_username 和 init_db_password
	Dukes_Bank_App	JAR_FILE、EAR_FILE、db_port、db_password 和 db_user

8 查看更新中更改及受影响的组件。

已修改属性和从属属性会突出显示。

9 单击**更新**部署应用程序的已修改配置。

下一步

您可以在部署摘要页面查看部署的状态。请参见第 156 页，“使用部署摘要页面”。

示例 Application Services 库任务

在部署过程中，可将预定义任务添加到执行计划。在应用程序部署过程中，必须执行或可以选择执行这些任务。

例如，必须添加和配置执行计划中 RHN 存储库的预定义任务才可安装或更新 YUM 存储库，从而使基于 Red Hat Linux 的应用程序正常运行。

此外，还可以创建自定义任务，将其添加到预定义的应用程序，然后将其部署到云。要使自定义任务在“任务”页面中可用，必须将其添加到库。请参见第 122 页，“将自定义任务添加到库”。

- 添加加入域预定义任务第 182 页，

  通过“加入域”预定义任务，Windows 虚拟机可以在部署过程中加入 Active Directory 域。该预定义任务会启动 Windows 域管理器 `Netdom.exe` 实用程序，以自动执行此操作。

- 添加 APT 存储库配置预定义任务第 183 页，

  APT 存储库配置预定义任务是用于更新 APT 存储库的脚本，以便在 Ubuntu 或其他基于 DEB 的操作系统上安装或更新软件。

- 添加 YUM 存储库配置预定义任务第 184 页，

  YUM 存储库配置预定义任务是用于更新 YUM 存储库的脚本，以便在 CentOS 或其他基于 RPM 的操作系统上安装或更新软件。

- 添加 RHN 注册预定义任务第 185 页，

  RHN 注册预定义任务是一个脚本，用于更新 YUM 存储库，从而安装 Red Hat Enterprise Linux 或其他具备 Red Hat 网络与 YUM 的 Red Hat 操作系统。

添加加入域预定义任务

通过“加入域”预定义任务，Windows 虚拟机可以在部署过程中加入 Active Directory 域。该预定义任务会启动 Windows 域管理器 `Netdom.exe` 实用程序，以自动执行此操作。

利用该预定义任务，您无需再手动为 Windows 虚拟机模板配置静态域设置，并且您可以自定义该任务。该预定义任务需要额外的重新启动周期，才能完成加入 Active Directory 域过程。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证 Application Services 是否已配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。
- 对于多个部署，请先手动为 Windows 虚拟机模板启用“更改 SID”和“域加入”选项，然后再部署应用程序。请参见第 71 页，“为 Windows 虚拟机模板启用 SID 更改和域加入”。

步骤

1. 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。
2. 单击基于 Windows 的应用程序的名称。

   此时将显示应用程序版本列表。
3. 选择应用程序版本。
4. 创建部署配置文件。
5. 在部署配置文件向导中，按照**执行计划**的提示进行操作。
6. 如果节点已加入群集，请单击**展开群集** ()。

   如果群集节点未扩展，预定义任务仅添加到群集中的第一个虚拟机。
7. 单击**添加脚本任务** () 并拖动蓝图中的预定义任务。

   拖动预定义任务时，会看到光标 ( )，指示您可以在此处拖放预定义任务。

   将预定义任务拖放到节点中后，会显示“添加自定义任务”对话框。
8. 从**库任务名称**下拉菜单中选择预定义任务。

   此时将显示支持的操作系统、预定义任务详细信息、脚本和属性详细信息。
9. 在**属性**选项卡上，配置属性。
   a. 选择 domain_name 属性，在“编辑属性”对话框中为 Windows 域键入一个新名称，然后单击**保存**。
   b. 选择 domain_user 属性，在“编辑属性”对话框中指定可以加入 Active Directory 的域用户的名称，然后单击**保存**。
   c. 选择 domain_password 属性，在“编辑属性”对话框中键入域用户密码，然后单击**保存**。

d （可选）选择 apply_ou 属性，在“编辑属性”对话框中将新值更改为 **Yes** 以允许 Active Directory 域中的特定组织单位加入 Windows 虚拟机，然后单击**保存**。

e （可选）选择 domain_ou 属性，在“编辑属性”对话框中指定 Active Directory 域中的组织单位，然后单击**保存**。

10 单击**确定**。

“加入域”预定义任务会添加到执行计划。

11 查看部署配置文件设置并部署应用程序。

下一步

了解是否向 Application Services 库添加自定义任务。请参见第 122 页，“将自定义任务添加到库”。

添加 APT 存储库配置预定义任务

APT 存储库配置预定义任务是用于更新 APT 存储库的脚本，以便在 Ubuntu 或其他基于 DEB 的操作系统上安装或更新软件。

可以配置 APT 存储库配置属性以添加新的存储库或移除现有的所有存储库。如果需要多个存储库，则可创建多个任务并通过逐一连续添加任务在执行计划中将这些任务链接起来。

仅能在测试环境中使用预定义的示例任务。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 添加需要 APT 用于安装或更新软件包的任何服务或应用程序之前，请验证是否将预定义的任务添加到执行部署计划。
- 验证 Application Services 是否已配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

此时将显示应用程序版本列表。

3 选择应用程序版本。

4 创建部署配置文件。

5 在部署配置文件向导中，按照**执行计划**的提示进行操作。

6 如果节点已加入群集，请单击**展开群集** ()。

如果群集节点未扩展，预定义任务仅添加到群集中的第一个虚拟机。

7 单击**添加脚本任务** () 并拖动蓝图中的预定义任务。

拖动预定义任务时，会看到光标 ( )，指示您可以在此处拖放预定义任务。

将预定义任务拖放到节点中后，会显示“添加自定义任务”对话框。

8 从**库任务名称**下拉菜单中选择预定义任务。

此时将显示支持的操作系统、预定义任务详细信息、脚本和属性详细信息。

9 在**属性**选项卡上，配置属性。

   a 选择 repository_name 属性，键入新的唯一值以用于在“编辑属性”对话框中标识存储库，然后单击**保存**。

   b 选择 source_str 属性，在“编辑属性”对话框的 **URL 值**文本框中键入 **`http://site.example.com/debian distribution component1 component2 ...`**，然后单击**保存**。

   Ubuntu URL 示例为 deb http://us.archive.ubuntu.com/ubuntu/ lucid main。

   c 选择 remove_all_repos 属性，然后在“编辑属性”对话框中定义相应的值。

   将值设置为 **`true`** 可在添加新配置之前移除其他所有存储库。此外，可接受默认的 `false` 值以添加新的存储库。

10 单击**保存**。

11 单击**确定**。

   APT 存储库配置预定义任务将添加到执行计划。

12 查看部署配置文件设置并部署应用程序。

下一步

确定是否向 Application Services 库添加自定义任务。请参见第 122 页，“将自定义任务添加到库”。

添加 YUM 存储库配置预定义任务

YUM 存储库配置预定义任务是用于更新 YUM 存储库的脚本，以便在 CentOS 或其他基于 RPM 的操作系统上安装或更新软件。

可以配置 YUM 存储库配置属性以添加新的存储库或移除现有的所有存储库。如果需要多个存储库，则可创建多个任务并通过逐一连续添加任务在执行计划中将这些任务链接起来。

仅能在测试环境中使用预定义的示例任务。

前提条件

- 使用分配有 Application Services 角色的用户帐户登录到 Application Services。
- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 在您添加需要 YUM 来安装或更新软件包的任何服务或应用程序之前，请验证是否已将预定义任务添加到执行部署计划。
- 验证 Application Services 是否已配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

   此时将显示应用程序版本列表。

3 选择应用程序版本。

4 创建部署配置文件。

5 在部署配置文件向导中，按照**执行计划**的提示进行操作。

6 如果节点已加入群集，请单击**展开群集** ( )。

   如果群集节点未扩展，预定义任务仅添加到群集中的第一个虚拟机。

7 单击**添加脚本任务** ( ) 并拖动蓝图中的预定义任务。

拖动预定义任务时，会看到光标 ( )，指示您可以在此处拖放预定义任务。

将预定义任务拖放到节点中后，会显示“添加自定义任务”对话框。

8 从**库任务名称**下拉菜单中选择预定义任务。

此时将显示支持的操作系统、预定义任务详细信息、脚本和属性详细信息。

9 在**属性**选项卡上，配置属性。

a 选择 repository_name 属性，键入新的唯一值以用于在“编辑属性”对话框中标识存储库，然后单击**保存**。

b 选择 repository_url 属性，在“编辑属性”对话框中键入 URL 值，然后单击**保存**。

有关 CentOS 6.3 32 位 URL 的示例，请参见 http://vault.centos.org/6.3/os/i386/。

c 选择 remove_all_repos 属性，然后在“编辑属性”对话框中定义相应的值。

d 将值设置为 **`true`** 可在添加新配置之前移除其他所有存储库。

e 接受默认的 **`false`** 值可添加新的存储库。

10 单击**保存**。

11 单击**确定**。

YUM 存储库配置预定义任务会添加到执行计划。

12 查看部署配置文件设置并部署应用程序。

下一步

考虑向 Application Services 库添加自定义任务。请参见第 122 页，“将自定义任务添加到库”。

添加 RHN 注册预定义任务

RHN 注册预定义任务是一个脚本，用于更新 YUM 存储库，从而安装 Red Hat Enterprise Linux 或其他具备 Red Hat 网络与 YUM 的 Red Hat 操作系统。

预定义任务使用以计算机名称 VMware_AppDirector_*$RANDOM* 提供的凭据在 Red Hat 网络中注册虚拟机，其中 *$RANDOM* 是使虚拟机注册唯一的短字符串。

仅能在测试环境中使用预定义的示例任务。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 在您添加需要 YUM 来安装或更新软件包的任何服务或应用程序之前，请验证是否已将预定义任务添加到执行部署计划。
- 验证 Application Services 是否已配置为使用代理。请参见第 31 页，“将 Application Services 配置为对外部 URL 使用代理”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**应用程序**。

2 单击应用程序的名称。

此时将显示应用程序版本列表。

3 选择应用程序版本。

4 创建部署配置文件。

5 在部署配置文件向导中，按照**执行计划**的提示进行操作。

6 如果节点已加入群集，请单击**展开群集** ( )。

如果群集节点未扩展，预定义任务仅添加到群集中的第一个虚拟机。

7 单击**添加脚本任务** ( ) 并拖动蓝图中的预定义任务。

拖动预定义任务时，会看到光标 ( )，指示您可以在此处拖放预定义任务。

将预定义任务拖放到节点中后，会显示“添加自定义任务”对话框。

8 从**库任务名称**下拉菜单中选择预定义任务。

此时将显示支持的操作系统、预定义任务详细信息、脚本和属性详细信息。

9 在**属性**选项卡上，配置属性。

a 选择 rhn_username 属性，输入用于在 Red Hat 网络中注册虚拟机的用户名值，然后单击**保存**。

b 选择 rhn_password 属性，输入用于在 Red Hat 网络中注册虚拟机的密码值，然后单击**保存**。

10 单击**确定**。

已将 RHN 注册预定义任务添加到执行计划。

11 查看部署配置文件设置并部署应用程序。

下一步

如果拥有自定义任务，可以将其添加到 Application Services 库。请参见第 122 页，“将自定义任务添加到库”。

示例库服务

Application Services 的库中包含在多个应用程序中可重复使用的预定义组件（如服务）。这些服务可供 Application Services 中的所有业务组使用。

库服务

在 Application Services 标题栏上，可以单击下拉菜单并选择**库 > 服务**查看可用的示例服务。“库”菜单还包括标准逻辑模板、任务、操作系统、标记、策略和外部服务。

应用程序架构师可以创建应用程序蓝图，然后向适用节点添加示例服务并对其进行配置。也可以在部署预定义的应用程序时配置示例服务。

在应用程序蓝图中，这些示例服务分为应用程序服务器、数据库服务器、Web 服务器、Windows 服务、监控、Puppet 服务以及其他服务。

注意 仅在测试环境中使用预定义的示例库服务。

管理部署 19

部署应用程序时，会向 Application Services 中的“部署”页面添加一个项目。

可以使用“部署”页面查看以下项目：

- 所有部署的列表
- 单个部署的部署详细信息，如云提供程序、部署环境、部署配置文件、应用程序的类型和版本或更新过程
- 包含策略和可用更新的部署

还可以启动策略扫描、启动更新过程、从云中卸载已部署的应用程序以及从 Application Services 中移除应用程序部署记录。

注意 策略扫描在复合部署中无法运行。

从部署摘要页面可以导航到更新与合规性摘要页面。

本章讨论了以下主题：

- 第 187 页，“查看应用程序的部署任务和蓝图详细信息”
- 第 189 页，“查看应用程序的已部署虚拟机的详细信息和执行计划”
- 第 190 页，“启动策略扫描”
- 第 191 页，“从云中卸载应用程序”
- 第 192 页，“从 vCloud Automation Center 缩小部署”
- 第 192 页，“从 vCloud Automation Center 扩大部署”
- 第 193 页，“从 vCloud Automation Center 卸载应用程序”
- 第 194 页，“从 Application Services 中删除应用程序部署”
- 第 195 页，“取消部署或更新过程”
- 第 196 页，“查看策略合规性摘要”

查看应用程序的部署任务和蓝图详细信息

可以在部署摘要页面上查看特定应用程序部署或更新过程的进度、成功或失败的相关详细信息。可以查看所分配的 IP 地址、所选云网络以及所运行的每个安装、配置和启动或更新脚本的日志。

部署摘要页面显示部署的整体状态。摘要页面上列出的每个部署都有一个启动部署时的应用程序蓝图和部署配置文件快照。如果您更改了实际的应用程序蓝图或部署配置文件，这些更改不影响特定部署中显示的蓝图或部署。请参见第 156 页，“使用部署摘要页面”。复合部署摘要页面显示其所有应用程序部署的详细信息，并允许您随时对其部署摘要页面进行访问，如第 156 页，“使用复合部署摘要页面”中所述。

所有 Application Services 角色均可查看其业务组中的部署。

前提条件

验证是否已部署应用程序或已启动更新过程。请参见第 151 页，“使用单个部署配置文件进行部署”或第 161 页，第 17 章“更新应用程序部署”。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 单击部署的名称。

此时将显示带有状态窗口的部署摘要页面。在虚拟机详细信息状态窗口中，您可以快速访问每个虚拟机，并且能够登录在 vCloud Director 上部署的虚拟机。

3 要查看任务、依赖任务或正在获取 IP 地址的虚拟机的状态，请单击**展开**图标 ( ) 以在部署摘要页面中打开适当的窗口。

4 单击**最小化**图标 ( ) 以查看完整的部署状态摘要。

5 要了解整体部署状态，请查看任务详细信息状态窗口。

6 要查找部署任务的名称，请查看任务详细信息状态窗口中的“任务摘要”部分。

“任务摘要”部分列出了部署或更新配置文件、用户角色、策略合规性信息、部署的开始和结束时间、上次更新应用程序的时间、包含日志的子文件夹的运行 ID 编号以及分配给应用程序名称的值的相关详细信息。

要查找 vCloud Director、vCloud Automation Center 或 Amazon EC2 中的部署的相关信息，您需要使用 vApp、计算机或实例名称。还可以从 vCloud Director、vCloud Automation Center 和 Amazon EC2 用户界面检查部署的进度。

7 有关部署环境、云提供程序名称、云提供程序说明、主机 IP 地址、vCloud Director 组织名称、业务组和预留策略或者 Amazon VPC 以及关联的可用性区域和用户名详细信息，请查看任务详细信息状态窗口中的“部署位置”部分。

本部分中的部署信息是部署时应用程序蓝图和配置文件的快照。如果您更改了实际的应用程序蓝图，这些更改将不影响特定部署中显示的蓝图。云提供程序和部署信息反映为应用程序部署创建的云提供程序映射和部署环境详细信息。

8 要查看应用程序蓝图名称和版本信息，请查看任务详细信息状态窗口中的“应用程序详细信息”部分。

9 要查看蓝图和部署配置文件中的属性替代，请展开蓝图状态窗口。

此状态窗口显示的快照反映部署时蓝图中包含的设置和定义。如果您更改了实际的应用程序蓝图，这些更改将不影响特定部署中显示的蓝图。您可以删除某个部署而不影响应用程序中的组件或服务。

a 选择一个服务或应用程序组件，然后单击**属性**选项卡。

b 对于更新过程，**属性**选项卡中的“上一个值”列显示上一更新过程的库、蓝图或部署配置文件值。

同一选项卡中的“新值”列显示在当前更新过程中添加的值。

10 查看展开的蓝图状态窗口中失败的任务。

如果某项任务在服务或应用程序组件上失败，该服务或应用程序组件上将显示任务失败图标 ( )。任务失败时，整个部署将停止并标记为“失败的部署”，并且不会再运行其他任务。所有其余任务上将显示任务未运行图标。

如果执行更新部署，则将突出显示已更新的节点应用程序组件和属性。如果更新部署失败，应用程序组件将标记为“失败”，并用红色突出显示受影响的属性。

下一步

检查执行计划中提供的虚拟机详细信息和任务。请参见第 189 页，“查看应用程序的已部署虚拟机的详细信息和执行计划”。

查看应用程序的已部署虚拟机的详细信息和执行计划

在部署摘要页面中，可以展开虚拟机详细信息状态窗口获取特定于虚拟机的信息。还可以展开执行计划状态窗口查看基于在应用程序中定义的依赖关系所运行的置备任务。

可提供 vCloud Director、vRealize Automation 或 Amazon EC2 中虚拟机的相关信息（如节点名称、逻辑模板和云模板）、内存分配、CPU 数量以及网络连接详细信息。还可以查看主机名以轻松识别 vCloud Director 或 vRealize Automation 部署中的虚拟机。

前提条件

验证应用程序是否已部署或更新过程是否已启动。请参见第 151 页，“使用单个部署配置文件进行部署”或第 161 页，第 17 章“更新应用程序部署”。

步骤

1 要查找虚拟机的 IP 地址，请查看虚拟机详细信息状态窗口。

   分配 IP 地址时，各个虚拟机的 IP 地址将显示在 IP 地址列中。

   例如，在包含负载平衡器节点的三层应用程序中，您可能会找到负载平衡器的 IP 地址，并将该 IP 地址提供给您的最终用户。

2 在虚拟机详细信息状态窗口的虚拟机表中找到特定于虚拟机的代理引导程序日志。

   每个虚拟机的代理引导程序日志将在“日志”列中进行更新。如果引导程序进程失败，则部署将标记为“失败的部署”并在任务详细信息窗口中显示原因。

3 展开窗口，然后单击“云模板”列中的省略号按钮 (...) 检索来自 vCloud Director、vRealize Automation 或 Amazon EC2 中特定于虚拟机的云模板详细信息。

   此信息包括诸如磁盘大小、CPU、内存之类的信息。

4 展开窗口，然后单击“网络信息”列中的省略号 (...) 检索来自 vCloud Director 或 Amazon EC2 的网络详细信息。

5 要查看为各个节点定义以替代 vRealize Automation 蓝图中属性的自定义属性，请单击“额外配置”图标 ( )。

6 对于正在进行的部署，单击页面右上角的**刷新**可更新状态。

   此页面大约每 30 秒刷新一次。

7 对于具体的任务详细信息（如开始时间、结束时间和上次更新时间），请展开执行计划状态窗口。

   对于每个任务，在获取 IP 地址并引导代理后，将对组件进行部署、安装和配置。上述过程将根据执行计划中任务之间的箭头显示的应用程序依赖关系执行。

8 检查每个任务的状态。

   根据任务的状态，会在任务的旁边显示图标。

图标	描述
	任务尚未启动或没有运行。
	任务正在进行中。任务成功完成后会显示该状态图标。此图标也用于显示主机、代理引导程序和网络引导程序置备任务的整体完成状态。
	展开并查看每个置备任务的详细信息。

图标	描述
	任务需要重新引导。
	任务失败。任务失败时会显示该状态图标。此图标也用于显示主机、代理引导程序或网络引导程序置备任务的整体失败状态。
	展开并查看一个或多个失败的置备任务的日志。
	任务正等待其中一个依赖关系完成运行

9 有关组件或操作脚本及其属性的详细信息，请展开执行计划状态窗口，单击任务旁边的箭头图标 (▼)，然后选择**查看组件属性**。

此时列出的详细信息为部署时蓝图所包含的设置和定义。

10 要查看操作脚本，请展开执行计划状态窗口，单击任务旁边的箭头图标 (▼)，然后选择**查看操作脚本**。

11 要访问虚拟机日志文件，请单击任务旁边的箭头图标 (▼)，然后选择**查看虚拟机日志**。

12 要查看脚本的属性值，请单击任务旁边的箭头图标 (▼)，然后选择**查看组件属性**。

13 有关更新部署的详细信息，请展开执行计划状态窗口并检查更新的节点。

对于扩展部署，执行计划会显示已修改的群集节点和受影响节点的更新脚本。主机、代理引导程序和网络引导程序置备任务仅显示在扩大的节点上。从属节点不包含任何置备任务。

对于要修改已部署应用程序配置的更新过程，执行计划会显示更改节点和受影响节点的更新脚本。

下一步

要从云中卸载已部署的应用程序，请参见第 191 页，“从云中卸载应用程序”。

要从 Application Services 中删除部署记录，请参见第 194 页，“从 Application Services 中删除应用程序部署”。

启动策略扫描

可以对部署启动策略扫描以评估在与部署对应的部署环境下定义的所有策略实例。启动部署或更新过程时，Application Services 的第一步是运行策略评估。

策略扫描只能针对活动部署执行。例如，如果更新过程失败，则必须将当前部署的状态还原为活动状态，然后再重试策略扫描。

如果部署环境中用于策略实例的策略已更新，则用户必须删除现有策略实例，然后基于更新后的策略创建策略实例。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证策略或策略实例是否已更改。
- 验证具有已弃用策略定义的部署在 Application Services 中是否可用。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 在搜索文本框中输入要扫描的现有部署的名称。

3 从搜索结果列表中，选择要扫描的部署。

此时将打开部署摘要页面。

4 在部署摘要上方的工具栏中，选择**操作 > 扫描**。

5 确认扫描部署过程。

此时将打开合规性摘要页面，显示整体合规性状态和策略详细信息。时间轴中将显示策略扫描和启动扫描时的时间戳。

下一步

跟踪部署扫描的策略合规性详细信息。请参见第 196 页，“查看策略合规性摘要”。

从云中卸载应用程序

在 Application Services 中，您可以启动卸载过程，以从云环境中移除 vCloud Director vApp 和关联的虚拟机、vRealize Automation 虚拟机和 vCenter Server 中关联的虚拟机或 Amazon EC2 实例。

您可以从 Application Services 用户界面或命令行界面卸载已部署的应用程序。请参见第 202 页，“使用 CLI 可卸载部署”。在用户界面中，您可以为服务版本和应用程序组件定义一个 TEARDOWN 生命周期阶段脚本，以移除应用程序和关联的虚拟机或更新的应用程序。

如果安装一个或多个属于应用程序的虚拟机后部署失败，或未使用应用程序，则您可以使用 Application Services 卸载该应用程序。应用程序中的所有虚拟机均已从云中的主机移除。

从云中卸载部署不会从 Application Services 中的“部署”页面移除部署记录。要从“部署”页面删除部署记录，请参见第 194 页，“从 Application Services 中删除应用程序部署”。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证属于该应用程序的虚拟机是否仍存在于云中。例如，如果您的环境具有在特定天数后删除虚拟机的策略，则虚拟机可能已经删除。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 在搜索文本框中输入要卸载的部署的名称。

3 从搜索结果列表中，选择要卸载的部署。

   此时将打开部署摘要页面。

4 在部署摘要上方的工具栏中，选择**操作 > 卸载**。

   - 在“卸载部署”对话框中，选择**快速卸载**以移除此部署的虚拟机，然后单击**卸载**。
   - 在“卸载部署”对话框中，选择**辅助卸载**打开卸载向导，然后单击**卸载**。

   对于快速卸载过程，TEARDOWN 生命周期阶段脚本不会运行。

   复合应用程序部署仅支持快速卸载过程，并且其所有应用程序部署必须同时卸载。

   对于辅助卸载过程，卸载向导会显示您可以配置的 TEARDOWN 生命周期阶段脚本。您还可以查看脚本中任务的运行顺序，并在运行脚本以移除此部署的虚拟机之前查看脚本变更。

5 （可选）如果卸载过程失败，则重复此过程。

任务时间轴将在任务状态窗口上方显示卸载过程的状态。从云中的主机成功移除虚拟机后，状态会在“任务摘要”部分和整体部署状态中显示。

启动卸载过程后，即使卸载过程失败且虚拟机存在于云环境中，您也无法更新已部署的虚拟机。请让您的云管理员从云中卸载该部署。

下一步

要从“部署”页面删除部署记录，请参见第 194 页，“从 Application Services 中删除应用程序部署”。

从 vCloud Automation Center 缩小部署

可以从 vRealize Automation 控制台请求缩小现有部署中的一个或多个群集节点。您只能对应用程序蓝图中建模为群集的节点的群集大小进行配置。

如果某个部署使用的蓝图具有已映射项目，则不能缩小该部署。有关项目管理的详细信息，请参见第 135 页，第 15 章“使用项目”。

前提条件

- 以**租户管理员**身份登录到 vRealize Automation 控制台。
- 验证您至少拥有从 vRealize Automation 访问一个包含群集节点的已部署应用程序的权限。

步骤

1 选择**项目** > **应用程序部署**。

2 在搜索文本框中输入部署的名称，然后单击放大镜图标 ( ) 或按 Enter 键。

3 从搜索结果列表中选择部署。

   此时将显示“项目详细信息”页面。

4 从**操作**菜单中选择**缩小**。

   此时将显示“缩小”对话框的**请求信息**选项卡。

5 在**描述**文本框中输入描述。

   此必填描述将帮助您在提交后监视请求的进度。

6 （可选）在**原因**文本框中输入请求的原因。

7 单击**下一步**。

8 通过单击**缩小量**文本框中的上下箭头输入要缩小的群集数。

9 单击**提交**。

   此时将显示“请求”确认框。

10 单击**确定**。

下一步

您可以在 vRealize Automation 控制台的**请求**选项卡上监视缩小请求的状态。

从 vCloud Automation Center 扩大部署

可以从 vRealize Automation 控制台请求扩大现有部署中的一个或多个群集节点。您只能对应用程序蓝图中建模为群集的节点的群集大小进行配置。

如果某个部署使用的蓝图具有已映射项目，则不能扩大该部署。有关项目管理的详细信息，请参见第 135 页，第 15 章“使用项目”。

前提条件

- 以**租户管理员**身份登录到 vRealize Automation 控制台。
- 验证您至少拥有从 vRealize Automation 访问一个包含群集节点的已部署应用程序的权限。

步骤

1 选择**项目** > **应用程序部署**。

2 在搜索文本框中输入部署的名称，然后单击放大镜图标 ( ) 或按 Enter 键。

3 从搜索结果列表中选择部署。

此时将显示“项目详细信息”页面。

4 从**操作**菜单中选择**扩大**。

此时将显示“扩大”对话框的**请求信息**选项卡。

5 在**描述**文本框中输入描述。

此必填描述将帮助您在提交后监视请求的进度。

6 （可选）在**原因**文本框中输入请求的原因。

7 单击**下一步**。

8 通过单击**扩大量**文本框中的上下箭头输入要扩大的群集数。

9 单击**提交**。

此时将显示“请求”确认框。

10 单击**确定**。

下一步

您可以在 vRealize Automation 控制台的**请求**选项卡上监视扩大请求的状态。

从 vCloud Automation Center 卸载应用程序

您可以从 vRealize Automation 控制台请求卸载标准应用程序和复合应用程序及其虚拟机。

您也可以从 Application Services 用户界面或命令行界面卸载已部署的应用程序。请参见第 191 页，“从云中卸载应用程序”和第 202 页，“使用 CLI 可卸载部署”。在 Application Services 用户界面中，您可以为服务版本和应用程序组件定义一个 TEARDOWN 生命周期阶段脚本，以移除应用程序和关联的虚拟机或更新的应用程序。

如果在 vCenter Server 和 vRealize Automation 中安装一个或多个属于应用程序的虚拟机后部署失败，或未使用应用程序，则您可以使用 vRealize Automation 卸载该应用程序。应用程序中的所有虚拟机均会从其主机中移除。

通过 vRealize Automation 进行部署卸载不会从 Application Services 的“部署”页面移除部署记录。要从部署页面删除部署记录，请参见第 194 页，“从 Application Services 中删除应用程序部署”。

前提条件

- 以**租户管理员**身份登录到 vRealize Automation 控制台。
- 验证您至少可以访问部署在 vRealize Automation 中的一个应用程序。
- 验证属于该应用程序的虚拟机是否仍存在于云中。例如，如果您的环境具有在特定天数后删除虚拟机的策略，则虚拟机可能已经删除。

步骤

1 选择**项目 > 应用程序部署**。

2 在搜索文本框中输入要卸载的部署的名称，然后单击放大镜图标 ( ) 或按 Enter 键。

3 从搜索结果列表中选择要卸载的部署。

此时将显示“项目详细信息”页面。

4 从**操作**菜单中选择**破坏**。

此时将显示“破坏”对话框的**请求信息**选项卡。

5 在**描述**文本框中输入描述。

此必填描述将帮助您在提交后监视请求的进度。

6 （可选）在**原因**文本框中输入请求的原因。

7 单击**下一步**。

8 在**卸载部署**选项卡上选择**选择拆卸选项**单选按钮。

选项	描述
辅助卸载	破坏所有部署有应用程序的虚拟机并执行 TEARDOWN 生命周期阶段脚本。
快速卸载	破坏所有部署有应用程序的虚拟机，但不执行 TEARDOWN 生命周期阶段脚本。

9 单击**提交**。

此时将显示“请求”确认框。

10 单击**确定**。

下一步

您可以在 vRealize Automation 控制台的**请求**选项卡上监视卸载请求的状态。

要从“部署”页面删除部署记录，请参见第 194 页，“从 Application Services 中删除应用程序部署”。

从 Application Services 中删除应用程序部署

从云中删除应用程序后，或者如果您不再需要特定部署的详细信息，那么可以从 Application Services 的“部署”页面中删除该部署。

从 Application Services 的“部署”页面中删除部署时，不会从云中删除已部署的应用程序、vApp、用于 vCloud Director 的虚拟机、vRealize Automation 虚拟机以及 vCenter Server 中的关联虚拟机或者 Amazon EC2 实例。要从云中删除应用程序，请参见第 191 页，“从云中卸载应用程序”。

前提条件

以**应用程序发布者和部署者**身份登录到 Application Services。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 在搜索文本框中输入要删除的部署的名称。

3 从搜索结果列表中，选择要删除的部署。

此时将打开部署摘要页面。

4 在部署摘要上方的工具栏中，选择**操作 > 删除**。

5 单击**删除部署**。

6 如果未先从云中卸载已部署的应用程序就从“部署”页面中删除了部署，则必须使用 vCloud Director、vRealize Automation 或 Amazon EC2 删除云中的残留组件。

选项	描述
vCloud Director	删除 vApp 及关联的虚拟机。
vRealize Automation	删除 vRealize Automation 虚拟机以及 vCenter Server 中的关联虚拟机。
Amazon EC2	停止部署实例，删除与部署对应的安全组，并释放分配给部署实例的弹性 IP 地址。

部署记录将从“部署”页面中删除。

取消部署或更新过程

在某些情况下，如果某个部署或更新过程无限期处于正在进行状态，而不显示通过或失败部署状态，您可以停止该部署或更新过程，并在云环境中停止置备。

停止部署或更新过程时，状态将更改为“正在停止”且一直保持该状态，直到 Application Services 成功停止部署或更新过程。之后状态将更改为“已停止”。Application Services 不删除部署或更新过程。

您不能停止或恢复复合部署。

注意 如果部署状态未在合理时间内更改为 STOPPED，该过程可能会挂起。您可以通过再次单击**停止**强制停止部署。请仅在部署状态经过不合理的长时间后仍未更改为 STOPPED 时强制停止该过程。有关相关信息，请参见*《Application Services 故障排除》*。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否有正在进行的部署或更新过程。

步骤

1 在搜索文本框中输入要取消的部署或更新过程的名称。

2 从搜索结果列表中，选择部署或更新过程。

此时将打开部署摘要页面。

3 在部署摘要上方的工具栏中单击**停止**。

4 确认您的选择。

注意 如果部署状态在合理的时间内未更改为 STOPPED，该过程可能会挂起，在这种情况下，您可以通过再次单击**停止**强制硬停止。再次单击**停止**会立即强制停止部署，而不会完成停止进程所使用的正常事件序列。系统将提示您通过单击**确定**确认此操作。请仅在部署状态经过不合理的长时间后仍未更改为 STOPPED 时强制停止该过程。

如果正常停止部署，当前运行的进程也将正常停止。如果强制停止，部署将突然停止，当前运行的进程将继续运行。在任一情况下，都不会清理关联的虚拟机。

下一步

正常停止部署后，可以恢复部署、从云中卸载部署或从 vRealize Automation 中删除应用程序部署记录。请参见第 187 页，第 19 章“管理部署”。

对于更新过程，您可以通过启动另一个更新过程来继续与部署进行交互。请参见第 161 页，第 17 章“更新应用程序部署”。

查看策略合规性摘要

您可以查看部署的整体合规性状态和关联的策略详细信息。

在部署期间，策略定义中的 SCAN 生命周期脚本会评估部署的合规性状态。“整体合规性状态”图标变红或变绿取决于部署中策略定义的冲突或合规性。“合规性摘要”页面上显示的状态是启动部署时应用程序蓝图的快照。合规与违反的策略定义以及关键与非关键的定义的数量会出现在整体合规性状态中。关联的部署配置文件位于整体状态旁边。

在整体合规性状态上方，策略扫描时间轴包含一个时间戳，显示了启动部署扫描和任何后续部署合规性扫描的时间。例如，后续扫描会检查目录中已更新的策略定义是否与现有部署相符。

前提条件

- 以**应用程序发布者和部署者**身份登录到 Application Services。
- 验证是否已将策略实例应用到该部署。

步骤

1 在 Application Services 标题栏中，单击下拉菜单，然后选择**部署**。

2 在“搜索”文本框中输入具有策略定义的部署的名称。

3 从搜索结果列表中，选择部署。

   此时将打开部署摘要页面。

4 在部署摘要上方的工具栏中，选择**部署视图 > 合规性视图**。

   此时将显示整体合规性状态。表中将列出详细的策略状态。

5 在“策略”列中，单击违反合规性的策略名称以在合规性日志中查看详细信息。

   根据合规性日志中的信息，您可以修复策略定义中的冲突并部署应用程序。

   例如，内存限制策略显示应用程序部署中有两台虚拟机（共 10 台）违反了策略。您可以查看策略的“详细信息”列和“合规性日志”部分，以标识冲突并修复应用程序。

下一步

如果部署正在进行中，则从部署摘要页面跟踪部署状态。请参见第 156 页，“使用部署摘要页面”。

使用 Application Services CLI 20

Application Services CLI 是基于 Spring Roo 的客户端，可使用 REST API 通过 HTTPS 与 Application Services 服务器进行通信。

admin 用户使用 Application Services CLI 在 vRealize Automation 服务器中注册 Application Services 设备。具有应用程序发布者和部署者角色的用户帐户可以使用 CLI 从云中部署或卸载应用程序。有关用户角色的信息，请参见第 37 页，第 5 章“设置用户和组”。

注意 **admin** 用户的密码为首次启动设备时设置的管理员密码。

Application Services 用户名和密码必须符合 ISO-8859-1

虽然 Application Services 用户界面支持在用户名和密码值中使用 unicode 字符集，但 Application Services CLI 仅支持 ISO-8859-1 字符集。如果要使用 Application Services CLI，请使用符合 ISO-8859-1 字符集的 Application Services 用户名和密码值。

本章讨论了以下主题：

- 第 197 页，“远程启动 CLI”
- 第 198 页，“常规 CLI 选项”
- 第 199 页，“管理云通道”
- 第 200 页，“使用 CLI 部署和更新应用程序”
- 第 202 页，“使用 CLI 可卸载部署”

远程启动 CLI

您可以从远程计算机启动 Application Services CLI。

最佳做法是远程运行 CLI 以降低服务器负载并避免在 Application Services 设备上共享 CLI。保护远程计算机与 CLI 的连接。

有些 CLI 命令使用的业务组或部署的名称可包含非英语字符。要显示这些字符，必须在完全国际化的 shell 客户端或者 Putty 或 iTerm2 等终端界面中运行 CLI。此外，将环境变量 LC_CTYPE 设置为 `en_US.UTF-8` 以启用非英语字符输入。

如果使用 **`su - root`** 登录 CLI，系统会自动设置该变量。如果忽略连字符，而使用 **`su root`** 登录，则必须使用以下命令设置该变量。

```
export LC_CTYPE=en_US.UTF-8;
```

前提条件

- 验证您是否知道 Application Services 设备的密码。
- 验证是否在远程计算机上安装了 Java JRE 1.7。
- 请确保远程计算机可使用 HTTPS 连接到 Application Services 设备。

步骤

1 从 Application Services 服务器 (http://*DarwinServerIP*/tools/darwin-cli.jar) 将 darwin-cli.jar 文件下载到远程计算机上具有写入权限的文件夹中。

   将 *DarwinServerIP* 替换为 Application Services 服务器 IP 地址。

   CLI 在会话过程中会创建日志文件。

2 打开命令提示符并启动客户端。

   ```
   java -jar /PathToJarFolder/darwin-cli.jar
   ```

   此时将显示 Darwin CLI 横幅以及 appd> CLI 提示符。

3 登录到 Application Services 服务器。

   ```
   login --serverUrl https://DarwinServerIP:8443/darwin --username UserName --password password
   --tenantId tenantid
   ```

   如果通过登录命令或者允许您添加密码的命令运行 `--password` 参数，您的密码将以纯文本形式保存在当前目录的 `darwin-cli-history.log` 文件中。默认情况下，CLI 会删除日志文件。为了提高安全性，请删除该日志文件。

   如果忽略 --password 参数，系统将提示您输入密码。系统不会将密码保存为纯文本。

   `--tenantId` 参数用于登录到租户名 *tenantid* 标识的租户。如果忽略 `--tenantId` 参数，租户默认为 vsphere.local。

命令行提示符将显示您的用户名、域和租户。

常规 CLI 选项

通过 Application Services 命令行界面选项，您可以部署应用程序、更新已部署的应用程序或从云中卸载应用程序。

登录 CLI 程序后，在 `roo` shell 提示符下按 **Tab** 键可显示可用命令选项列表。对命令选项的访问权限取决于所分配的用户角色。有关用户角色及其允许的功能的信息，请参见第 37 页，第 5 章“设置用户和组”。

如果使用多个以空格分隔的单词，请使用引号将单词括起来。通常在您输入命令后，系统除了指示命令是否成功外，还会显示许多详细信息。

注意 请勿使用在 CLI 中不可用的命令选项。

表 20-1 常规 CLI 命令

CLI 命令	描述
`help`	列出使用信息。
`cliversion`	显示 CLI 版本信息。
`login`	将当前用户登录到 `roo` shell。
`logout`	注销当前用户，但不关闭 `roo` shell 提示符。 您可以注销再以其他用户身份登录。

表 20-1 常规 CLI 命令（续）

CLI 命令	描述
`status`	指示用户是否已登录。如果用户已登录，则将显示用户名。
`exit`	退出 CLI 程序。

管理云通道

可以执行创建、更新、启用或删除 Application Services 设备和 Amazon EC2 VPC 中的端点虚拟机之间的安全云通道连接等操作。

以**应用程序云管理员**身份登录到 Application Services。

请参见第 82 页，“创建云通道以连接到 Amazon EC2”。

表 20-2 使用 CLI 管理云通道

CLI 命令	描述
`create-cloud-tunnel --name` *`TunnelName`* `--description` *`"TunnelDescription"` `--enabled false --externalAddress` *`EndpointVMElasticIP`* `--sshPort 22 --internalAddress` *`EndpointVMPrivateIP`* `--proxyUrl` *`ProxyURL`* `--username` *`UserName`* `--privateKeyPath` *`PrivateKeyFilePath`*	创建安全云通道。 例如，可以使用以下命令创建名为 EC2 Tunnel 的云通道。 `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`
`update-cloud-tunnel --name` *`PreviousTunnelName`* `--new-name` *`NewTunnelName`* `--description` *`"TunnelDescription"`* `--externalAddress` *`EndpointVMElasticIP`* `--sshPort 22 --internalAddress` *`EndpointVMPrivateIP`* `--proxyUrl` *`ProxyURL`* `--username` *`UserName`* `--privateKeyPath` *`PrivateKeyFilePath`*	更改现有云通道的各个参数的值。 对于大多数参数，此命令仅在云通道禁用时正常运行。
`list-cloud-tunnels --name` *`TunnelName`*	检索所有可用云通道的列表或者特定云通道（指定时）。 要查看特定云通道的详细信息，请使用命令 `list-cloud-tunnels --name EC2 Tunnel`。
`enable-cloud-tunnel --name` *`TunnelName`*	启用现有云通道。 此命令可启动用于建立安全云通道连接的后台操作。
`disable-cloud-tunnel --name` *`TunnelName`*	禁用现有云通道。 此命令可启动用于断开安全云通道连接的后台操作。
`delete-cloud-tunnel --name` *`TunnelName`*	删除现有云通道。 云通道必须先禁用再删除。 移除 Application Services 设备与 Amazon EC2 环境之间的云通道连接。
`test-cloud-tunnel --name` *`TunnelName`*	验证云通道安全 SSH 连接并报告当前状态。 启用或禁用云通道可在后台启动连接或断开连接操作。此命令可报告该后台操作的状态。此外，该命令还会检查部署期间的连接是否可在 Application Services 上正常运行。 在启用或禁用云通道后使用此命令可确认请求是否成功。

使用 CLI 部署和更新应用程序

要部署应用程序，必须以应用程序发布者和部署者身份登录。

部署应用程序之前，验证您的部署配置文件是否完整、已保存且不存在验证错误。还需使用应用程序的名称。在启动更新过程以扩展群集节点之前，请先验证已部署应用程序是否包含群集节点。

您可以快速部署应用程序、检查应用程序部署的状态、为已部署应用程序启动扩大过程，或使用 CLI 修改已部署应用程序的服务和应用程序组件配置。还可以使用 Application Services 用户界面执行以上所有任务。请参见第 152 页，“快速部署应用程序”、第 156 页，“使用部署摘要页面”、第 162 页，“启动更新过程以扩大部署”或第 166 页，“启动更新过程以修改配置”。

还可以从 vCloud Director、vRealize Automation 和 Amazon EC2 用户界面查看已部署虚拟机的状态。

表 20-3 在 CLI 中部署或更新应用程序

CLI 命令	描述
`deploy-application --destination` *`ApplicationName-ApplicationVersion-DeploymentProfileName`*	部署应用程序蓝图。 例如，要部署具有 prod-dep 部署配置文件的 2.1.0 版群集化的 Dukes Bank 应用程序，可使用命令 `deploy-application --destination "Clustered Dukes Bank App-2.1.0-prod-dep"`。 输入 **`deploy`** 并按 Tab 键可显示该命令的可用选项。如果列表没有为使用空格分隔的多个单词显示引号，请使用引号将单词括起来。
`deployment-status --deploymentName` *`DeploymentName`*	显示最新部署的状态。 如果应用程序部署失败，可以收集特定于虚拟机的日志文件以对该问题进行故障排除。请参见《VMware vCloud Automation Center Application Services 故障排除》指南。

表 20-3 在 CLI 中部署或更新应用程序 （续）

CLI 命令	描述

deploy-application --destination *ApplicationNameApplicationVersion-DeploymentProfileName* --propertiesFile *FileName*.xml	部署具有新属性值的最新应用程序版本。
	可以创建为所需属性和部署时可替代的属性定义新属性值的属性文件。以下代码示例为 Apache_LB 和 AppServer 节点的 global_conf、http_node_port 和 http_server_port 属性定义了新值。
	`<?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>`
	例如，要部署具有 staging-dep 部署配置文件和 `new.props` 属性文件的 2.1.0 版群集化的 Dukes Bank 应用程序，可使用命令 `deploy-application --destination " Clustered Dukes Bank App-2.1.0-DP_MAN_VCD" --propertiesFile ~/new.props.xml`
	输入 **deploy** 并按 Tab 键可显示该命令的可用选项。如果列表没有为使用空格分隔的多个单词显示引号，请使用引号将单词括起来。

表 20-3 在 CLI 中部署或更新应用程序 （续）

CLI 命令	描述

`update-scaleout --deploymentName` *`DeploymentName`* `--propertiesFile` *`FileName`*`.xml`	启动更新过程以扩展已部署应用程序。
	您必须创建为节点定义新群集大小值的属性文件。以下代码示例为 AppServer1 和 DBServer1 节点定义了群集大小值。
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<ns1:scaleout-properties`
	`xmlns:ns1="http://www.test.com/darwin/schema/beans/api">`
	`<cluster-node>`
	`<name>appserver</name>`
	`<scale-out-by>1</scale-out-by>`
	`</cluster-node>`
	`</ns1:scaleout-properties>`
	例如，要扩大具有 `scaleout` 属性文件的 2.1.0 版群集化的 Dukes Bank 应用程序，可使用命令 `update-scaleout --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-6-cc0a3b20-43a0-4a22-be3e-49d4f31ab8e8" --propertiesFile ~/scaleout.xml`
	输入 **`update`** 并按 Tab 键可显示该命令的可用选项。如果列表没有为使用空格分隔的多个单词显示引号，请使用引号将单词括起来。

`update-config --deploymentName` *`DeploymentName`* `--configUpdatePropertiesFile` *`ConfigUpdatePropertiesFileName`*`.xml`	启动更新过程可修改已部署应用程序中的现有服务或应用程序组件的配置。
	必须创建配置更新属性文件才能配置现有服务。以下代码示例修改了 AppServer 节点 vFabric tc Server 服务中的 global_conf 和 db_port 属性。
	`<?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>`
	`</ns1:config-update-properties>`
	例如，要修改具有 `configupdate` 属性文件的 2.1.0 版群集化的 Dukes Bank 应用程序中的配置，可使用命令 `update-config --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-2-19d63535-673e-4766-b380-de4e6ec3676a" --configUpdatePropertiesFile ~/configupdate.xml`
	输入 **`update`** 并按 Tab 键可显示该命令的可用选项。如果列表没有为使用空格分隔的多个单词显示引号，请使用引号将单词括起来。

使用 CLI 可卸载部署

要卸载部署，您必须以应用程序发布者和部署者身份登录。

从云中卸载已部署的应用程序之前，请验证是否具有已部署应用程序的部署名称。在 Application Services 用户界面中可以监控卸载过程的状态。请参见第 191 页，“从云中卸载应用程序”。

表 20-4 移除 CLI 中的部署

CLI 命令	描述
`teardown --deploymentName` *`DeploymentName`*	从 vCloud Director 中卸载已部署的应用程序。 例如，`teardown --name appd-TestApp-1.0.0-admin-3-a99309aa-aa5b-45c2-b6a0-bac4a421178e` 命令可从云中移除应用程序。 输入 **`teardown`** 并按 Tab 键可显示该命令的可用选项。如果列表没有为使用空格分隔的多个单词显示引号，请使用引号将单词括起来。

使用 CLI 导入和导出功能 21

您可以跨 Application Services 实例导入和导出预配置的应用程序蓝图及其关联的部署配置文件、服务、外部服务、策略、项目存储库以及可用的自定义任务，以帮助您进一步自定义应用程序。

可以在 Application Services 6.1 的不同实例之间导入和导出软件包，可以把 Application Services 5.0、5.2 和 6.0 中的软件包导入到 6.1。

您可以从 Application Services 远程计算机启动 CLI。如果使用完全国际化的 shell 客户端或终端界面，您还可以运行包含非英文字符的 CLI 命令。请参见第 197 页，“远程启动 CLI”。

您必须以应用程序架构师和应用程序目录管理员身份登录，才能导入和导出 Application Services 软件包。

如果导入或导出软件包大于实例上的可用内存，则会显示一条错误消息通知您重新分配内存。例如，您可以使用 `java -Xmx6000m -jar darwin-cli.jar` 命令为实例分配 6GB 的内存。

- 使用 CLI 导出命令第 205 页，

  CLI 导出命令创建由应用程序及其关联的蓝图和部署配置文件、外部服务、策略、项目存储库、服务、逻辑模板和可用的自定义任务组成的软件包，以在不同的 Application Services 实例之间导出。

- 使用 CLI 导入命令第 207 页，

  CLI 导入命令可导入软件包，该软件包涵盖应用程序及其关联的蓝图和部署配置文件、外部服务、策略、项目存储库、服务、逻辑模板以及不同 Application Services 实例之间可用的自定义任务。

使用 CLI 导出命令

CLI 导出命令创建由应用程序及其关联的蓝图和部署配置文件、外部服务、策略、项目存储库、服务、逻辑模板和可用的自定义任务组成的软件包，以在不同的 Application Services 实例之间导出。

导出软件包具有两种格式，压缩存档文件格式和非压缩文件格式。压缩格式为默认格式，其文件类型为 `.zip`。非压缩格式是一个 XML 文件，可以是任何文件类型。import-package 命令接受上述两种格式。

您无法导出名称中包含制表符、下划线、换行符或回车符的对象。

导出应用程序、服务、外部服务、脚本任务或策略版本时，将默认移除所有安全属性，以避免将密码等敏感信息导出到其他 Application Services 实例中。

运行导入命令时，会将整个软件包导入到指定的服务器中。无法选择性地从导出软件包导入元素。

export-package 命令

CLI export-package 命令具有以下格式。

```
export-package --exportFilePath 路径名 --fromGroup obgname --CommandOption [[--CommandOption] ...]
[--uncompressed]
```

下表介绍了 export-package 命令所需的组件。

CLI 命令组件	描述
`export-package`	export-package 命令名称。
`--exportFilePath` *路径名*	指定要创建的导出软件包的路径名称。例如，`--exportFilePath /home/dev/joomla.zip` 指定 `joomla.zip` 导出软件包的位置和名称。导出软件包的文件扩展名为 `.xml` 时，必须使用 `--uncompressed` 选项。
`--fromGroup` *busgroupname*	指定要导出的对象所属的业务组。不包括策略。租户中的任何人都可以访问策略，因此所属业务组将被忽略。
--CommandOption	指定要导出的对象的类型、名称和版本。例如，命令选项 `--applicationVersion distmaps:1.0.0` 指定应用程序名称为 `distmaps`，版本为 `1.0.0`。export-package 命令可以有多个命令选项，但必须至少有一个命令选项。

下表介绍了导出一个或多个对象时可以使用的命令选项。

CLI 命令选项	描述

`--applicationVersion` *名称*:*版本 ID*
`--serviceVersion` *名称*:*版本 ID*
`--scriptTaskVersion N` *名称*:*版本 ID*
`--externalServiceVersion` *名称*:*版本 ID*
`--policyVersion` *名称*:*版本 ID*
`--repoVersion` *名称*:`1.0.0`
	指定要导出的对象和版本。
	■ *名称*是要导出的应用程序、服务、脚本任务、外部服务、策略或项目存储库的名称。名称不区分大小写。如果名称包含空格，请用双引号括起。
	■ *版本 ID* 是对象的版本标识符，格式为 *主版本*.*次版本*.*微版本*{-*标识符*}，例如 1.0.0-beta。
	您可以通过用逗号分隔版本名称，使用单个导出命令导出多个相同类型的对象。如果版本名称列表中的某个版本名称包含空格，请用双引号将整个列表括起。
	`export-package --exportFilePath pkgname.zip --fromGroup Dev --applicationVersion "aname1:1.0.0,aname two:1.0.0,aname3:1.0.0"`
	您也可以使用同一导出命令导出多个类型的对象。
	`export-package --exportFilePath pkgname.zip --fromGroup Dev --applicationVersion aname:1.0.0 --serviceVersion sname:1.0.0 --scriptTaskVersion stname:1.0.0`
	使用 ALL 关键字可导出指定业务组中所有版本的对象类型。
	`export-package --exportFilePath pkgname.zip --fromGroup Dev --applicationVersion ALL --serviceVersion ALL --scriptTaskVersion ALL`

`--uncompressed`	以非压缩格式导出软件包。对于文件扩展名为 `.xml` 的导出软件包必需。

导出依赖关系

导出某个对象时，还会导出其依赖关系。对于应用程序等高级别的对象，依赖关系可以包括服务、虚拟机模板、脚本任务等。导出命令会验证顶级对象所属的业务组，但不会验证从属对象所属的业务组，因其可能属于不同的业务组。该命令只是将从属对象作为顶级对象的一部分导出。例如，假设某一应用程序属于“开发”业务组，但是其中的一个服务属于“生产”业务组，则“开发”组的成员可以导出包含该服务的应用程序。但是，“生产”业务组的成员只能导出该服务，不能导出该应用程序。当然，同时属于这两个业务组的成员既可以同时导出这两个对象，或者单独导出其中一个对象。

导出要求

下表显示了导出或导入对象时，必须属于所属业务组的情况。

导出的对象	业务组要求
应用程序 服务 外部服务 虚拟机模板	以下任务要求在所属业务组中具有成员身份。 ■ 导出对象。 ■ 将专用对象作为依赖关系导出。 以下任务允许在租户的任何业务组中具有成员身份。 ■ 查看共享应用程序或服务。 ■ 将共享对象作为依赖关系导出。
脚本任务 （始终共享）	以下任务要求在所属业务组中具有成员身份。 ■ 导出对象。 以下任务允许在租户的任何业务组中具有成员身份。 ■ 查看对象。
策略 （始终共享）	以下任务允许在租户的任何业务组中具有成员身份。 ■ 导出对象。
项目存储库 （始终共享）	以下任务要求在所属业务组中具有成员身份。 ■ 导出对象。

使用 CLI 导入命令

CLI 导入命令可导入软件包，该软件包涵盖应用程序及其关联的蓝图和部署配置文件、外部服务、策略、项目存储库、服务、逻辑模板以及不同 Application Services 实例之间可用的自定义任务。

您仅可以导入使用导出命令创建的软件包。无法导入由导出命令以外的任何方式更改或创建的软件包。

您必须属于导入软件包中每个对象所属的业务组，才能导入对象。

运行 import-package 命令时，会将整个软件包导入目标服务器。无法选择性地从软件包导入对象。

无法将同名的外部服务导入不同的业务组，如果外部服务由其他业务组所有，则无法将该外部服务作为依赖关系导入。

- 确保源设备中包含部署配置文件的应用程序是共享应用程序，或者由目标设备中的目标组专有。不支持将部署配置文件导入目标组之外的任何业务组拥有的专用应用程序。
- 确保导入期间未更改创建部署配置文件时所使用的蓝图，蓝图中稍有任何变化都将导致节点和组件无法映射，不支持这种使用情形。

您可以将部署配置文件作为共享应用程序的依赖关系导入，或者作为目标业务组拥有的专用应用程序的依赖关系导入。另外，还请确保导入期间未更改创建部署配置文件时所使用的蓝图，因为稍有变化都将导致节点和组件无法映射。请先保存您已添加到执行部署计划的自定义任务，然后再导入部署配置文件。请参见第 146 页，“检查执行计划并添加自定义任务”。

要成功完成导入过程，软件包中的对象不能包含非 ASCII 字符。

import-package 命令

CLI import-package 命令的格式如下。

`import-package --importFilePath` *路径名* `--targetGroup` *obgname* `--ConflictResolutionAction` *操作*

下表介绍了 import-package 命令所需的组成部分。

CLI 命令组件	描述
`import-package`	import-package 命令名称。
`--importFilePath` *路径名*	指定导入软件包的路径名称。例如，`--importFilePath /home/dev/joomla.zip` 指定了 `joomla.zip` 导入软件包的位置和名称。

CLI 命令组件	描述
`--targetGroup` *busgroupname*	指定拥有要导入对象的业务组。不包括策略。租户中的任何人都可以访问策略，因此所属业务组将被忽略。
`--ConflictResolutionAction` *操作*	指定 CHECK、SKIP、OVERWRITE 或 IMPORTASNEW 冲突解决方案操作。一个 import-package 命令必须指定一个冲突解决方案操作。

下表介绍了通过导入软件包导入对象时要使用的冲突解决方案操作。

冲突解决方案操作	描述
`CHECK`	列出导入软件包中的所有对象，并指示对象是否存在于目标上。通过比较对象名称和版本标识符确定匹配项。版本标识符采用 *major.minor.micro.qualifier* 的形式，例如 `2.0.1.beta`。
`SKIP`	如果对象不存在于目标上，请从软件包复制该对象。如果对象存在于目标上，则不要复制该对象。复制的对象属于 **`--TargetGroup`** 指定的拥有业务组。
`OVERWRITE`	如果对象存在于目标上，请使用软件包内容覆盖对象内容。将不存在于目标上的对象复制到 **`--TargetGroup`** 指定的拥有业务组。已登录用户必须属于软件包中拥有对象的业务组才能导入对象。
`IMPORTASNEW`	如果对象存在于目标上，请使用新名称将该对象从软件包复制到目标。新对象属于 **`--TargetGroup`** 指定的拥有业务组。必须通过 `--suffix` 选项为重命名的对象指定后缀。

下表介绍了可用的命令选项。

CLI 命令选项	描述
`--shared`	将软件包中的新项目与租户中所有业务组的成员共享。如果省略，请使软件包中的新项目专用于拥有业务组的成员。
`--suffix` *sfxtext*	将下划线和 *sfxtext* 所表示的指定文本添加到每个作为匹配项从软件包中复制的对象的末尾。例如，如果指定 **`--suffix NEW`** 并从软件包复制 `apache:1.0.0`，则对象的新名称为 `apache _NEW`，版本为 `1.0.0`。拥有新对象的业务组是 **`--targetGroup`** 指定的目标组。

示例：SKIP 和 OVERWRITE

```
import-package --importFilePath /home/dev/dukes.zip --conflictResolutionAction SKIP --targetGroup
Development
import-package --importFilePath /home/dev/dukes.zip --conflictResolutionAction OVERWRITE --
targetGroup Development
```

索引

符号

${random} **127**

A

Active Directory 域
  加入 **71**
  添加预定义任务 **182**
Amazon EC2
  部署配置文件 **141**
  网络 **142**
  应用程序置备 **79**
  注册云提供程序 **85**
  自定义 AMI **83**
Amazon EC2 AMI 模板, 创建 **83**
AMI, 创建 **83**
安全设置, 基于用户组 **16**
安全组件, 属性 **98**
安装, 准备 **21**
安装必备条件 **21**
安装代理引导程序, vCenter Server Windows 虚拟机模板 **49**
Application Director OVF, 部署 **24**
Application Services
  创建应用程序 **126**
  打开 Web 界面 **29**
  概览 **13**
  管理库 **101**
  架构原则 **14**
  启动设备 **26**
  升级到新版本 **35**
  设置 **19**
  使用 **9**
  输入序列号 **27**
  术语 **16**
  添加自定义任务 **122**
  Web 界面 **43**
  正在重新启动 **31**
Application Services 设备
  部署 **25**
  创建 darwin_user 密码 **27**
  配置 **27**
  向 vRealize Automation 服务器注册 **27**
  注册 vCloud Application Center 服务器 **26**
Application Services 中的对象, 共享 **39**
Application Services 组件, 开发 **89**
APT 存储库配置, 添加预定义任务 **183**

B

绑定, 属性 **95**
绑定属性 **128**
Bash, 支持的脚本 **90**
BeanShell, 支持的脚本 **90**
标记, 添加到库 **102**
标准化配置, 属性 **15**
部署
  查看任务状态 **187**
  查看摘要 **187**
  更新 **189**
  管理 **187**
  回滚更新过程 **170**
  扩大 **162**
  扩大节点群集 **192**
  启动更新过程以修改 **166**
  取消 **195**
  删除 **194**
  升级更新配置文件 **168**
  缩小 **164**
  缩小节点群集 **192**
  现有的更新配置文件 **168**
  修改配置 **166**
  应用程序更新 **161**
  运行 ID **187**
部署过程, 强制停止部署 **195**
部署环境
  创建 vCloud Director **76**
  为 Amazon EC2 创建 **86**
部署配置文件
  部署应用程序 **151**
  导入 **207**
  发布 **153**
  设置 **141**
  在复合部署配置文件中使用 **148**
部署配置文件向导 **151**
部署失败
  查看执行计划 **158**
  恢复 **158**
  恢复失败的部署 **158**
部署示例应用程序 **175**
部署摘要, 查看 **189**
部署摘要页面 **156**

部署状态, 查看 **189**

C

操作
  生命周期阶段 **89**
  组件生命周期阶段 **89**
操作系统
  Application Services 库 **101**
  自定义模板 **48**
  自定义模板的 **68, 83**
策略
  删除 **122**
  添加到库 **118**
策略版本
  编辑 **121**
  导出 **205**
  复制 **121**
策略定义脚本
  变量 **120**
  创建 **120**
策略配置文件, 定义 **118**
策略扫描, 启动 **190**
策略扫描时间轴, 查看 **196**
策略实例
  创建 **58**
  更新 **190**
CentOS, 示例模板 **173**
CHECK 选项, 导入 **207**
撤销更新配置文件 **168**
重新引导
  代理引导程序 **106**
  虚拟机 **106**
创建
  操作脚本 **106**
  Linux 模板 **51**
  Windows 模板 **49**
  Windows vApp **69**
  云通道 **82**
磁盘布局, 预定义的属性 **98**
CLI
  部署应用程序 **200**
  部署状态 **200**
  常规命令 **198**
  创建用户和组 **197**
  导出 **205**
  导入 **205**
  更新过程 **200**
  管理用户和组 **197**
  将 ISO-8859-1 字符集用于用户名和密码 **197**
  使用快速部署 **200**
  卸载部署 **202**
  注册 Application Services **32**
CLI（命令行界面） **199**
CLI), 导入 **207**
存储库规范, 创建 **135**
存储选项, 配置 **23**

D

代理引导程序
  安装 **70**
  重新引导 **113**
  下载 **70**
代理引导程序服务
  自定义模板 **48**
  自定义模板的要求 **83**
单层应用程序, 示例 **174**
单个节点, 绑定 **95**
darwin 用户帐户已锁定, 解锁 **30**
darwin_user 帐户, 解锁 **30**
登录, 默认组 **43**
DHCP, 配置 **71**
DotShoppingCart, 更新过程 **179**
端点虚拟机, 创建 **86**
端口, 要求 **22**
对照表, 设置 Application Services **19**
多个网卡, 定义 **131**
多云, 支持 **15**

E

额外配置 **142**

F

防火墙, 从防火墙外部访问 URL **31**
访问密钥 **80**
复合部署配置文件
  部署使用 **151**
  部署应用程序 **151**
  创建 **148**
  对部署配置文件进行排序 **149**
  发布 **154**
  设置 **148**
  选择部署配置文件 **149**
复合部署属性, 绑定属性 **150**
服务
  编辑 **107**
  定义 **104**
  复制 **107**
  删除 **108**
  添加到库 **103**
  现成 **103**
服务版本
  创建 **103**
  导出 **205**
服务目录, 请求项目 **155**
服务属性, 定义 **104**

G

高级蓝图, 创建 **129**
高级外部服务, 创建 **110**
高级应用程序蓝图 **129**
更新, 缩小示例应用程序 **179**
更新部署 **156**
更新的信息 **11**
更新过程
  回滚 **170**
  扩大示例应用程序 **179**
  Nanotrader **179**
  群集化的 DotShoppingCart **179**
  群集化的 Dukes Bank **179**
  取消 **195**
  以扩展示例应用程序 **179**
更新配置文件
  升级 **168**
  用于部署 **168**
global_conf **145**
共享选项, 导入 **207**
故障
  对修改配置进行故障排除 **172**
  扩展部署故障排除 **171**
故障排除
  扩展部署更新 **171**
  连接问题 **30**
  修改配置更新 **172**

H

合规性, 查看策略定义 **196**
合规性摘要, 查看 **196**
环境变量, 在脚本中使用 **91**
回滚部署 **161**

I

IMPORTASNEW 选项, 导入 **207**
IP 地址, 预定义的属性 **98**

J

架构原则, Application Services **14**
脚本
  Bash **90**
  BeanShell **90**
  PowerShell **90**
  Windows CMD **90**
  支持的操作 **90**
  支持的自定义任务 **90**
加入域
  启用 **71**
  添加预定义任务 **182**
基本外部服务, 创建 **110**
基本应用程序蓝图 **127**
节点群集
  扩大 **192**
  缩小 **192**
节点数组索引, 预定义的属性 **98**
解决方案实例
  创建 **61**
  注册 Puppet Master **64**
解锁, darwin_user 帐户 **30**
计算, 属性 **91**
JRE
  安装 **70**
  下载 **70**
  自定义模板要求 **83**
JRE 要求
  自定义模板 **48**
  自定义模板的 **68**
角色, 分配 **37**

K

可变磁盘 **131**
可变磁盘布局 **142**
客户机, 自定义 **74**
克隆, 部署 **49**
克隆部署 **52**
可用性区域
  Amazon EC2 模板 **85**
  为部署环境定义 **86**
库
  Application Services **15**
  操作系统 **101**
  创建高级外部服务 **110**
  创建基本外部服务 **110**
  创建外部服务 **110**
  导入 Puppet 服务 **108**
  定义外部服务属性 **111**
  管理 **101**
  删除外部服务 **115**
  添加标记 **102**
  添加策略 **118**
  添加外部服务 **109**
  维护外部服务版本 **114**
快照, 创建 **54**
库服务
  定义属性 **104**
  示例组件 **186**
扩大部署, 启动更新过程 **162**
扩展部署, 对失败的更新进行故障排除 **171**

L

蓝图, vRealize Automation **49**
蓝图详细信息, 查看 **187**
连接 Amazon EC2 **82**

链接克隆
部署 **49, 53, 54**
创建快照 **54**
连接问题, 故障排除 **30**
了解
部署过程 **157**
更新过程 **157**
libpython 软件包, 下载 **174**
临时虚拟机, 外部服务 **142**
Linux 模板, 创建 **73, 83**
Linux 虚拟机, 更新模板 **54**
Linux 虚拟机模板, 更新 **75**
逻辑模板
编辑 **117**
复制 **117**
删除 **118**
虚拟机 **115**

M

默认组, 登录 **43**
模板
导出 **75**
JRE 要求 **83**
OVF 格式 **24, 75**
示例 **173**
vCloud Automation Center **55**
vCloud Director **75**
为 Windows 配置 **70**
虚拟机 **15, 74**
注册 Amazon EC2 **85**
注册虚拟机 **85**

N

Nanotrader
部署 **174**
更新过程 **179**
修改示例应用程序中的配置 **180**
内存, 磁盘空间和内存大小 **22**
内容, 属性 **91**

O

OOB（即开即用）, 服务 **15**

P

配置
Amazon EC2 **80**
Linux 模板 **51**
vCloud Application Director **82**
Windows 模板 **49**
配置文件
设置应用程序部署 **141**
应用程序部署 **141, 147**
PowerShell, 支持的脚本 **90**
Puppet 服务, 导入库 **108**
Puppet Master
创建实例 **61**
注册 **59**
注册解决方案实例 **64**
准备 **59**

Q

启动 CLI 程序, 远程 **197**
群集化的 DotShoppingCart
部署 **175**
修改示例应用程序中的配置 **180**
群集化的 Dukes, 修改示例应用程序中的配置 **180**
群集化的 Dukes Bank, 更新过程 **179**
群集化的 Dukes Bank 应用程序, 部署 **176**
群集节点, 指定 **130**

R

Radiant CMS, 部署 **178**
任务时间轴 **156**
任务状态, 查看 **187, 189**
任务状态窗口 **156**
REST API
创建用户和组 **197**
管理用户和组 **197**
RHN 注册, 添加预定义任务 **185**

S

三层应用程序, 示例 **174**
升级的更新配置文件
放弃 **169**
接受 **169**
删除, 服务 **108**
设备, 启动 Application Services **26**
升级
使用升级安装程序后回滚 **36**
运行升级安装程序 **35**
生命周期阶段, 操作 **89**
设置
服务属性 **145**
节点属性 **145**
应用程序组件属性 **145**
置备环境 **47**
主机名称 **145**
设置主机名 **127**
示例, 任务 **173**
示例服务, 库服务 **186**
示例模板
CentOS **173**
使用 **173**
示例应用程序
部署 **174**
部署 DOTShoppingCart **175**
部署 Radiant CMS **178**
部署库组件 **173**

部署群集化的 Dukes Bank 应用程序 **176**
单层应用程序 **174**
服务 **173**
更改代码 **180**
更新已部署的应用程序 **179**
jPetStore **177**
扩展应用程序的更新过程 **179**
库任务 **181**
三层应用程序 **174**
添加 APT 存储库配置任务 **183**
添加 RHN 注册任务 **185**
添加 YUM 存储库配置任务 **184**
添加“加入域”任务 **182**
虚拟机模板 **173**
示例应用程序, 修改配置 **180**
属性
安全组件 **98**
必需 **99**
node **142**
计算 **91**
内部 **145**
内容 **91**
数组 **91**
特殊更新属性 **162, 164, 166**
预定义 **97**
预留 **145**
自定义 **142**
字符串 **91**
属性值
定义 **95**
解析 **97**
术语 **16**
数组, 属性 **91**
SID, 启用 **71**
SKIP 选项, 导入 **207**
搜索条件, Web 界面 **44**
SQLFire
部署 jPetStore 应用程序 **177**
下载 vFabric_SQLFire-1.0.jar **177**
SSH 连接, 启用 Ubuntu VM **174**
缩小部署, 启动更新过程 **164**

T

弹性 IP 地址 **80**
提供程序规范
编辑 **114**
创建 **113**

U

Ubuntu, 示例模板 **173**
URL, 从防火墙外部访问 **31**

V

vCenter Server, 虚拟机模板 **49**
vCenter Server Windows 虚拟机模板
安装代理引导程序 **49**
下载代理引导程序 **49**
自定义规范 **49**
vCloud Application Director
将策略添加到库 **118**
添加逻辑模板 **115**
vCloud Application Services, 安装 **21**
vCloud Automation Center
部署配置文件 **141**
发布到 **153**
发布到服务目录 **153**
将复合部署配置文件发布到 **154**
蓝图 **47**
模板 **55**
请求目录项 **155**
授权 **153**
SSO **153**
向 Application Services 设备注册 **26**
业务组 **55**
应用程序置备 **47**
预留 **56**
预留策略 **56**
云提供程序 **55**
注册 **32**
注册 Application Services **32**
vCloud Director
创建部署环境 **76**
创建自定义模板 **68**
模板 **75**
配置 **24**
网络 **142**
验证设置 **25**
应用程序置备 **67**
云提供程序 **75**
注册云提供程序和模板 **75**
vCloud Director 目录, 添加 Windows 模板 **72**
VMware Tools **69**
VMware Tools 要求, 自定义模板的 **68**
VMware vCenter 群集
单个主机 **23**
多个主机 **23**
配置 **23**
vRealize Automation
gugent **49, 51**
客户机代理 **49, 51**
蓝图 **49**
网络 **142**
自定义模板 **48**

W

外部 URL, 从防火墙外部访问 **31**
外部 URL 的代理, 配置 **31**
外部服务
  编辑 **114**
  创建 **110**
  复制 **114**
  蓝图 **132**
  删除 **115**
  添加 **132**
  添加到库 **109**
  映射 **57**
外部服务版本, 导出 **205**
外部服务实例 **56**
外部服务属性, 定义 **111**
网卡, 定义多个 **131**
网络带宽 **82**
网络选项, 配置 **23**
Web 界面
  Application Services **43**
  打开 Application Services **29**
  搜索条件 **44**
Web 浏览器, 支持的版本 **22**
违反, 查看策略定义 **196**
Windows
  添加虚拟机模板 **72**
  虚拟机模板 **69, 71**
Windows CMD, 支持的脚本 **90**
Windows 模板, 配置 **70**
Windows 虚拟机, 更新现有模板 **54**
Windows 虚拟机模板, 更新 **75**

X

现成, 服务 **103**
项目
  绑定到应用程序蓝图 **139**
  创建 **138**
  描述 **135**
  使用应用程序部署 **140**
项目存储库, 创建实例 **137**
项目存储库规范, 描述 **135**
项目存储库实例
  创建 **137**
  描述 **135**
  映射到项目规范 **138**
项目管理, 描述 **135**
项目规范
  创建 **138**
  描述 **135**
  映射到项目存储库规范 **138**
现有部署配置文件 **147**
下载代理引导程序, vCenter Server Windows 虚拟机模板 **49**
卸载
  请求 **193**
  应用程序 **191**
系统要求 **22**
修改配置, 对失败的更新进行故障排除 **172**
虚拟机
  查看详细信息 **189**
  重新引导 **113**
  端点 **81, 82**
  Linux **80**
  模板 **74**
  NAT **80**
  删除 **193**
  卸载 **193**
  自定义模板要求 **68**
虚拟机端点, 与 Application Services 配合使用 **79**
虚拟机模板
  创建 Amazon EC2 **83**
  对于 Windows **69**
  更新 **54, 75**
  启用 SID **71**
  启用加入域 **71**
  vCenter Server **49**
虚拟数据中心 (VCD), 创建 **76**
虚拟数据中心 (vDC) **76**
虚拟私有云 (VPC)
  为部署环境定义 **86**
  注册 **85**

Y

业务组
  vCloud Automation Center **55**
  在 Application Services 中使用 **39**
依赖关系, 创建 **130**
引导程序代理 Linux
  安装 **73**
  下载 **73**
引导程序要求, 自定义模板的 **83**
映射, 逻辑模板 **142**
应用程序
  版本 **126**
  部署 **141, 151**
  部署配置文件 **141, 147, 152**
  查看部署详细信息 **187**
  创建 **125**
  从云中删除虚拟机 **191**
  导出 **205**
  发布到服务目录 **153**
  复制 **133**
  共享 **126**
  快速部署 **152**
  配置 **128**

设置部署配置文件 **141**
使用复合部署配置文件进行部署 **151**
使用项目部署 **140**
卸载 **191**
应用程序版本
复制 **133**
删除 **134**
应用程序部署
更新 **161**
回滚更新过程 **170**
扩大 **162**
启动更新过程以修改 **166**
删除 **194**
升级更新配置文件 **168**
缩小 **164**
修改配置 **166**
应用程序蓝图, 创建 **129**
应用程序置备
Amazon EC2 **79**
vCloud Automation Center **47**
vCloud Director **67**
应用程序组件, 管理项目 **139**
用户
创建 **197**
管理 **197**
设置 **37**
用户界面
按钮 **44**
图标 **45**
用户角色, 与租户和业务组配合使用 **39**
用户帐户
解锁 **30**
设置 **37**
预定义的示例内容, 导入 **28, 41**
预定义的属性
磁盘布局 **98**
IP 地址 **98**
节点数组索引属性 **98**
预定义服务, 添加到库 **103**
预留, vCloud Automation Center **56**
预留策略, vCloud Automation Center **56**
YUM 存储库配置, 添加预定义任务 **184**
云抽象层 (CAL), Application Services **13**
云逻辑模板, 添加到库 **115**
云提供程序
vCloud Automation Center **55**
注册 **85**
云通道
管理 **199**
与 Application Services 配合使用 **79**
云中的应用程序, 删除 **191**

Z

置备任务
代理引导程序 **146, 157, 189**
网络引导程序 **146, 157**
主机 **146, 157, 189**
执行计划 **151**
注册
解决方案实例 **59**
Puppet Master **59**
Puppet Master 解决方案实例 **64**
为 Application Services 创建 **27**
注册解决方案
手动 **59**
自动 **59**
主机名称 **127, 145, 157**
自定义规范, vCenter Server Windows 虚拟机模板 **49**
自定义模板
操作系统 **48, 68**
代理引导程序服务 **48**
代理引导程序要求 **83**
JRE 要求 **48, 68, 83**
VMware Tools 要求 **68**
虚拟机要求 **68**
自定义模板的代理引导程序服务要求 **68, 83**
自定义任务
编辑 **123**
导出 **205**
复制 **123**
删除 **123**
添加 **122, 146**
自动绑定
标记 **104, 111**
标记信息 **96**
公开 **104, 111**
公开属性 **96**
type **96, 104, 111**
预配置组件 **96**
占用 **104, 111**
占用属性 **96**
字符串, 属性 **91**
组
创建 **197**
管理 **197**
设置 **37**
租户
将预定义的示例内容导入 **28**
在 Application Services 中使用 **39**
组件
开发 **89**
最佳做法 **99**
组件生命周期阶段, 操作 **89**
组件属性
必需 **99**
配置 **91**