

使用和管理 vRealize Automation Cloud Assembly

2022 年 10 月

vRealize Automation 8.0

您可以从 VMware 网站下载最新的技术文档:

<https://docs.vmware.com/cn/>。

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

**威睿信息技术（中国）有
限公司**
北京办公室
北京市
朝阳区新源南路 8 号
启皓北京东塔 8 层 801
www.vmware.com/cn

上海办公室
上海市
淮海中路 333 号
瑞安大厦 804-809 室
www.vmware.com/cn

广州办公室
广州市
天河路 385 号
太古汇一座 3502 室
www.vmware.com/cn

版权所有 © 2022 VMware, Inc. 保留所有权利。 [版权和商标信息](#)

目录

| | | |
|----------|----------------------------------------------------------------|-----------|
| 1 | vRealize Automation Cloud Assembly 是什么 | 6 |
| | vRealize Automation Cloud Assembly 如何工作 | 7 |
| 2 | 为您的组织设置 vRealize Automation Cloud Assembly | 9 |
| | vRealize Automation Cloud Assembly 用户角色是什么 | 9 |
| | 添加云帐户 | 12 |
| | 使用云帐户所需的凭据 | 13 |
| | 在 vRealize Automation 中创建 Microsoft Azure 云帐户 | 30 |
| | 在 vRealize Automation 中创建 Amazon Web Services 云帐户 | 30 |
| | 创建 Google Cloud Platform 云帐户 | 31 |
| | 创建 vCenter 云帐户 | 32 |
| | 创建 NSX-V 云帐户 | 33 |
| | 创建 NSX-T 云帐户 | 34 |
| | 创建 VMware Cloud on AWS 云帐户 | 36 |
| | 与其他应用程序集成 | 37 |
| | 如何使用 GitLab 和 GitHub 集成 | 37 |
| | 配置外部 IPAM 集成点 | 40 |
| | 如何升级到较新的 IPAM 集成软件包 | 42 |
| | 在 vRealize Automation Cloud Assembly 中配置 My VMware 集成 | 42 |
| | 在 Cloud Assembly 中配置 vRealize Orchestrator 集成 | 43 |
| | 如何在 vRealize Automation Cloud Assembly 中使用 Kubernetes | 45 |
| | 什么是 vRealize Automation Cloud Assembly 中的配置管理 | 52 |
| | 如何在 vRealize Automation Cloud Assembly 中创建 Active Directory 集成 | 57 |
| | 载入计划是什么 | 58 |
| | 将选定的计算机载入为单个部署 | 59 |
| | 将规则筛选的计算机作为单独部署载入 | 61 |
| | 高级配置 | 66 |
| | 如何配置 Internet 代理服务器 | 66 |
| | 如何使用 cloud-init 或 cloudbase-init 设置 Windows 模板 | 70 |
| | 如何使用 IPAM SDK 创建提供程序特定的外部 IPAM 集成软件包 | 70 |
| 3 | vRealize Automation Cloud Assembly 用例 | 72 |
| | WordPress 用例 | 72 |
| | 创建基础架构 | 73 |
| | 创建项目 | 80 |
| | 创建并扩展蓝图 | 81 |
| | VMware Cloud on AWS 用例 | 97 |

| | |
|----------------------------------------|-----|
| 配置基本 VMware Cloud on AWS workflow | 98 |
| 在 VMware Cloud on AWS 中配置隔离网络 | 109 |
| 提供程序特定的外部 IPAM 集成用例 | 113 |
| 在部署下载软件包之前，先在 Infoblox 应用程序中添加所需的可扩展属性 | 114 |
| 下载并部署外部 IPAM 提供程序软件包 | 116 |
| 为 IPAM 集成点创建运行环境 | 117 |
| 添加外部 IPAM 集成点 | 118 |
| 配置网络和网络配置文件以使用 IPAM 提供程序值 | 121 |
| 定义并部署使用 IPAM 提供程序范围分配的蓝图 | 123 |
| 对 IPAM 集成使用特定于 Infoblox 的属性 | 125 |

4 构建资源基础架构 127

| | |
|----------------------------------------------|-----|
| 如何添加云区域 | 127 |
| 了解有关云区域的更多信息 | 127 |
| 如何添加特定实例映射 | 129 |
| 了解有关特定实例映射的更多信息 | 129 |
| 如何添加映像映射 | 130 |
| 了解有关映像映射的更多信息 | 130 |
| 如何添加网络配置文件 | 132 |
| 了解有关网络配置文件的更多信息 | 133 |
| 使用网络和网络配置文件中的 IP 地址 | 137 |
| 使用网络和网络配置文件 | 138 |
| 使用负载均衡器设置 | 140 |
| 如何添加存储配置文件 | 142 |
| 了解有关存储配置文件的更多信息 | 142 |
| 如何使用标记 | 142 |
| 创建标记策略 | 145 |
| 在 vRealize Automation Cloud Assembly 中使用功能标记 | 146 |
| 在 vRealize Automation Cloud Assembly 中使用限制标记 | 146 |
| 标准标记 | 148 |
| vRealize Automation Cloud Assembly 如何处理标记 | 149 |
| 如何设置简单的标记结构 | 149 |
| 如何使用资源 | 150 |
| 计算资源 | 151 |
| 网络资源 | 151 |
| 安全资源 | 152 |
| 存储资源 | 153 |
| 计算机资源 | 153 |
| 卷资源 | 153 |
| 了解有关资源的更多信息 | 154 |

5 添加和管理项目 158

- 如何为我的开发团队添加项目 158
- 了解有关项目的更多信息 160
 - 使用项目标记和自定义属性 160
 - 项目在部署时的工作方式 161

6 设计部署 163

- 创建蓝图之前 164
- 用于创建蓝图的方法 164
- 如何从头开始创建简单的蓝图 166
 - 如何选择组件并将其添加到蓝图 166
 - 如何连接蓝图资源 167
 - 如何创建有效的蓝图代码 168
- 如何增强简单蓝图 170
 - 用户输入如何自定义蓝图 170
 - 如何设置组件部署顺序 175
 - 如何使用表达式使蓝图代码更具通用性 176
 - 如何在蓝图中自动初始化计算机 185
 - 如何在蓝图中启用远程访问 191
- 如何保存不同版本的蓝图 194
- 如何自定义已部署资源的名称 196
- 有哪些资源属性 198
- 有哪些蓝图代码示例 198
 - 蓝图中的 vSphere 组件示例 198
 - 可查看的蓝图 202
 - 网络、安全性和负载均衡器蓝图示例 209
 - Puppet 支持的具有用户名和密码访问权限的蓝图 213
- 如何使用商城 222
- 如何使用可扩展性扩展应用程序生命周期并实现自动化 223
 - 可扩展性操作订阅 223
 - 可扩展性工作流订阅 242
 - 了解有关可扩展性订阅的更多信息 248

7 管理部署 256

- 如何监控活动部署 257
- vRealize Automation Cloud Assembly 部署失败时可以执行哪些操作 258
- 如何管理已完成部署的生命周期 260
- 可以对部署运行哪些操作 262

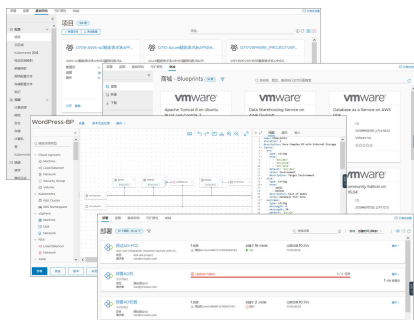
vRealize Automation Cloud Assembly 是什么

1

您可以使用 vRealize Automation Cloud Assembly 来连接到公有云提供商和私有云提供商，以将创建的计算机、应用程序和服务部署到您的资源。您和您的团队可以在支持迭代式工作流（从开发到测试再到生产）的环境中开发蓝图即代码。在置备时，可以部署到一系列云供应商。此服务是一个受管 VMware SaaS 和基于 NaaS 的框架。

vRealize Automation Cloud Assembly 概览包括以下基本功能。

- 在“基础架构”选项卡，可以添加和组织云供应商资源和用户。此选项卡还提供有关已部署的蓝图的信息。
- “商城”选项卡提供有助于生成蓝图库和访问支持 OVA 或 OVF 的 VMware Solution Exchange 蓝图和映像。
- “蓝图”选项卡是主要开发工作区。您可以使用画布和 YAML 编辑器来开发然后部署计算机和应用程序。
- “部署”选项卡显示了已置备资源的当前状态。您可以访问用于管理部署的详细信息和历史记录。



本章讨论了以下主题：

- [vRealize Automation Cloud Assembly 如何工作](#)

vRealize Automation Cloud Assembly 如何工作

vRealize Automation Cloud Assembly 是一项蓝图部署和开发服务。您和您的团队可以使用此服务将计算机、应用程序和服务部署到云供应商资源中。

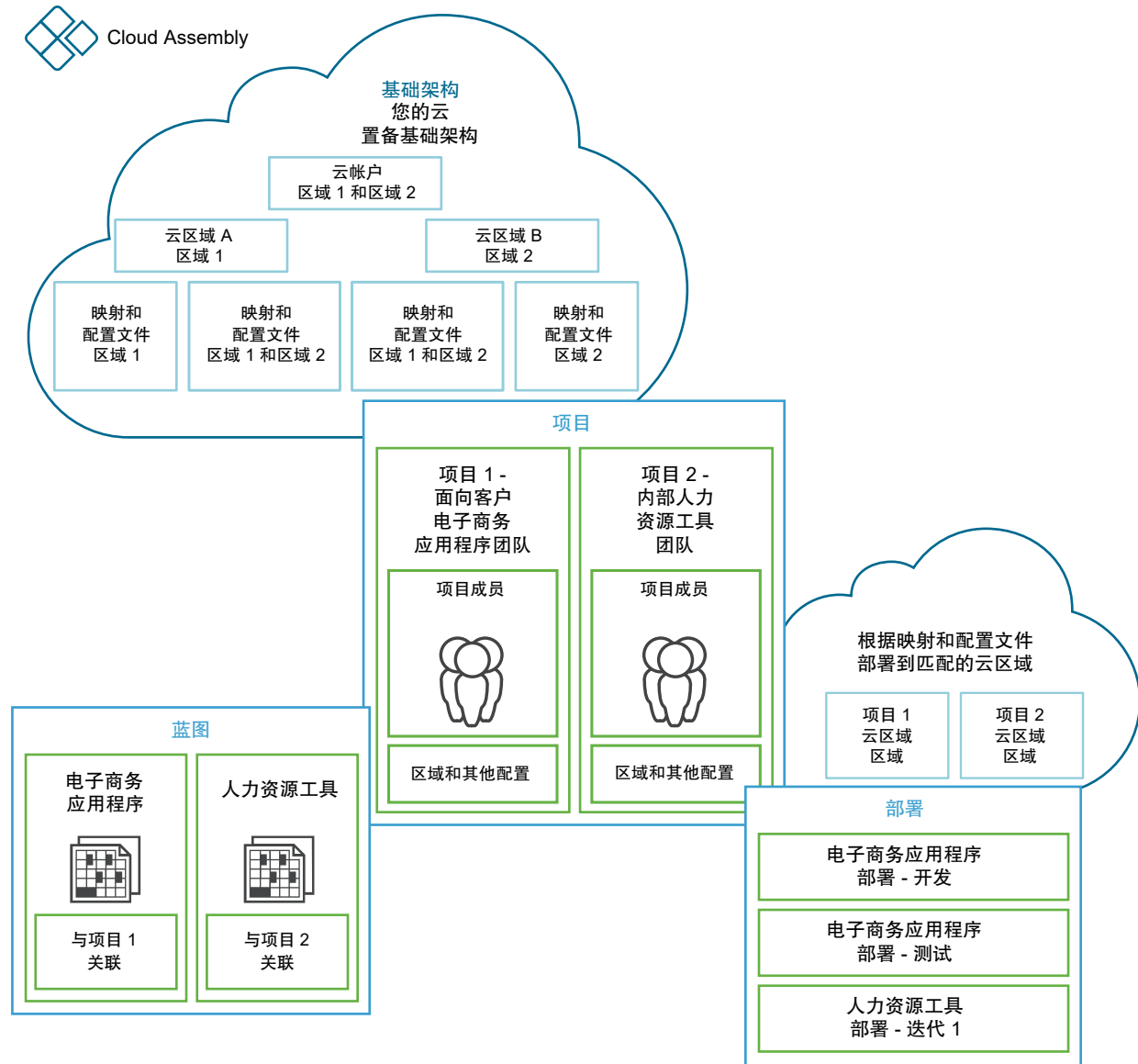
作为 Cloud Assembly 管理员（通常称为云管理员），您可以设置置备基础架构并创建用于对用户和资源进行分组的项目。

- 添加云供应商帐户。请参见[将云帐户添加到 vRealize Automation Cloud Assembly](#)。
- 确定哪些区域或数据存储是您希望开发人员部署到的云区域。请参见[了解有关 vRealize Automation Cloud Assembly 云区域的更多信息](#)。
- 创建策略以定义云区域。请参见[第 4 章 构建您的 vRealize Automation Cloud Assembly 资源基础架构](#)。
- 创建项目以将开发人员与云区域分组到一起。请参见[使用 vRealize Automation Cloud Assembly 项目标记和自定义属性](#)。

作为蓝图开发人员，您是一个或多个项目的成员。您可以创建蓝图并将其部署到与其中一个项目关联的云区域。

- 使用画布为项目开发蓝图。项目管理员可以使用商城从 VMware Solution Exchange 下载蓝图和支持映像。请参见[第 6 章 设计 vRealize Automation Cloud Assembly 部署和如何使用 vRealize Automation Cloud Assembly 商城](#)。
- 根据策略和限制将蓝图部署到项目云区域。
- 管理部署，包括删除未使用的应用程序。请参见[第 7 章 管理 vRealize Automation Cloud Assembly 部署](#)。

欢迎使用 vRealize Automation Cloud Assembly。有关如何定义基础架构然后创建并部署蓝图的示例，请参见[WordPress 用例](#)。



为您的组织设置 vRealize Automation Cloud Assembly

2

作为 Cloud Assembly 管理员，您必须了解用户角色，并设置与云帐户供应商和集成应用程序的连接。

当配置云帐户和集成时，您配置的是 Cloud Assembly 与这些目标系统之间的通信。

本章讨论了以下主题：

- [vRealize Automation Cloud Assembly 用户角色是什么](#)
- [将云帐户添加到 vRealize Automation Cloud Assembly](#)
- [将 vRealize Automation 与其他应用程序集成](#)
- [vRealize Automation Cloud Assembly 中的载入计划是什么](#)
- [vRealize Automation Cloud Assembly 环境的高级配置](#)

vRealize Automation Cloud Assembly 用户角色是什么

用户角色决定您在 vRealize Automation Cloud Assembly 中可以查看的内容和可以执行的操作。有些角色是在组织级别定义的，而有些角色特定于 vRealize Automation Cloud Assembly。

用户角色

由组织所有者在 vRealize Automation 控制台中为组织定义用户角色。有两种类型的角色：组织角色和服务角色。

组织角色是全局的，适用于组织中的所有服务。组织级别的角色是组织所有者或组织成员角色。

有关组织角色的详细信息，请参见《[管理 vRealize Automation](#)》。

vRealize Automation Cloud Assembly 服务角色（是特定于服务的权限）也在控制台的组织级别分配。

表 2-1. 服务角色

| 角色 | 说明 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------|
| Cloud Assembly 管理员 | 必须对整个用户界面和 API 资源具有读取和写入访问权限。这是唯一可以查看和执行所有操作的用户角色，包括添加云帐户、创建新项目以及分配项目管理员。 |
| Cloud Assembly 用户 | 不具有 Cloud Assembly 管理员角色的任何用户。 在 vRealize Automation Cloud Assembly 项目中，管理员将用户作为项目成员添加到项目中。管理员还可以添加项目管理员。下面定义了这两个角色的权限。 |

项目角色和权限

项目角色、项目管理员和项目成员在 vRealize Automation Cloud Assembly 中进行定义，可能会因项目而异。

在以下表中，定义了权限。请记住，云管理员对 UI 的所有区域具有完全权限。

项目管理员利用云管理员创建的基础架构来确保项目成员具有进行开发工作所需的资源。

表 2-2. 项目管理员权限

| 选项卡 | 节点或区域 | 查看 | 创建 | 修改/删除 |
|------|-------------|-----------|-----|-----------|
| 基础架构 | 配置 - 项目 | 是（仅限您的项目） | 否 | 是（仅限您的项目） |
| | 配置 - 云区域 | 否 | 否 | 否 |
| | 配置 - 特定实例映射 | 是 | 否 | 否 |
| | 配置 - 映像映射 | 是 | 否 | 否 |
| | 配置 - 网络配置文件 | 是 | 否 | 否 |
| | 配置 - 存储配置文件 | 是 | 否 | 否 |
| | 配置 - 标记 | 是 | 否 | 否 |
| | 资源 - 计算 | 是 | 否 | 否 |
| | 资源 - 网络 | 是 | 否 | 否 |
| | 资源 - 存储 | 是 | 否 | 否 |
| | 资源 - 计算机 | 是（仅限您的项目） | 是 | 是（仅限您的项目） |
| | 资源 - 卷 | | | |
| | 活动 - 请求 | 是（仅限您的项目） | 不适用 | 是（仅限您的项目） |
| | 活动 - 事件 | 是（仅限您的项目） | 不适用 | 是（仅限您的项目） |
| | 连接 - 云帐户 | 否 | 否 | 否 |
| | 连接 - 集成 | | 否 | 否 |

表 2-2. 项目管理员权限（续）

| 选项卡 | 节点或区域 | 查看 | 创建 | 修改/删除 |
|-----|-------------|-----------|-----------|-----------|
| | 连接 - 云代理 | | 否 | 否 |
| | 成本 - VMC 评估 | 是 | 否 | 否 |
| | 成本 - 私有云 | 是 | 否 | 否 |
| | 载入 | | 否 | 否 |
| 蓝图 | 蓝图 | 是（仅限您的项目） | 是（仅限您的项目） | 是（仅限您的项目） |
| 部署 | 部署 | 是（仅限您的项目） | 不适用 | 是（仅限您的项目） |

项目成员通常是创建和部署蓝图的开发人员。

表 2-3. 项目成员权限

| 选项卡 | 节点或区域 | 查看 | 创建 | 修改/删除 |
|------|-------------|----------------|-----|------------|
| 基础架构 | 配置 - 项目 | 是（仅限您属于其成员的项目） | 否 | 否 |
| | 配置 - 云区域 | 否 | 否 | 否 |
| | 配置 - 特定实例映射 | 是 | 否 | 否 |
| | 配置 - 映像映射 | 是 | 否 | 否 |
| | 配置 - 网络配置文件 | 是 | 否 | 否 |
| | 配置 - 存储配置文件 | 是 | 否 | 否 |
| | 配置 - 标记 | 是 | 否 | 否 |
| | 资源 - 计算 | 是 | 否 | 否 |
| | 资源 - 网络 | 是 | 否 | 否 |
| | 资源 - 存储 | 是 | 否 | 否 |
| | 资源 - 计算机 | 是（仅限由您部署的） | 是 | 是（仅限由您部署的） |
| | 资源 - 卷 | | | |
| | 活动 - 请求 | 是（仅限由您部署的） | 不适用 | 是（仅限由您部署的） |
| | 活动 - 事件 | 是（仅限由您部署的） | 不适用 | 是（仅限由您部署的） |
| | 连接 - 云帐户 | 否 | 否 | 否 |
| | 连接 - 集成 | | | |
| | 连接 - 云代理 | | | |
| | 成本 - VMC 评估 | 是 | 否 | 否 |

表 2-3. 项目成员权限（续）

| 选项卡 | 节点或区域 | 查看 | 创建 | 修改/删除 |
|-----|----------|----------------------------|-----------|-----------------------------------------|
| | 成本 - 私有云 | 是 | 否 | 否 |
| | 载入 | | | |
| 蓝图 | 蓝图 | 是（仅限您的项目） | 是（仅限您的项目） | 是（仅限您的项目） |
| 部署 | 部署 | 是（仅限您的部署，除非与所有项目成员共享项目部署。） | 不适用 | 是（仅限您的部署，除非与所有项目成员共享项目部署，并且您有权运行实施后操作。） |

将云帐户添加到 vRealize Automation Cloud Assembly

云帐户是已配置的权限，vRealize Automation Cloud Assembly 可使用此权限从区域或数据中心收集数据并将蓝图部署到这些区域。

收集的数据包括您稍后与云区域关联的区域。

稍后配置云区域、映射和配置文件时，可以选择与其关联的云帐户。

作为云管理员，您可以为项目创建云帐户，以便团队成员可以使用这些云帐户进行工作。将从云帐户收集网络 and 安全性、计算资源、存储以及标记内容等资源信息的数据。

注 如果云帐户具有已在区域中部署的关联计算机，则可以使用载入计划将这些计算机载入到 vRealize Automation Cloud Assembly 中进行管理。请参见 [vRealize Automation Cloud Assembly 中的载入计划是什么](#)。

如果移除部署中使用的云帐户，则属于该部署的资源将变为未受管。

在 vRealize Automation 中使用云帐户所需的凭据

要在 vRealize Automation 中配置和使用云帐户，请确认您具有以下凭据。

所需的云帐户凭据

| 要执行的操作... | 所需内容 ... |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 注册并登录到 vRealize Automation Cloud Assembly | VMware ID。 <ul style="list-style-type: none">■ 使用公司电子邮件地址设置 My VMware 帐户。 |
| 连接到 vRealize Automation 服务 | <p>对出站流量打开并具有透过防火墙对以下域的访问权限的 HTTPS 端口 443:</p> <ul style="list-style-type: none">■ *.vmwareidentity.com■ gaz.csp-vidm-prod.com■ *.vmware.com <p>有关端口和协议的详细信息，请参见 VMware 端口和协议。</p> <p>有关所需端口和协议的相关信息，请参见：</p> <ul style="list-style-type: none">■ “安装”帮助中的端口和协议■ “参考架构”帮助中的端口要求 |

| 要执行的操作... | 所需内容 ... |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 添加 Amazon Web Services (AWS) 云帐户 | <p>提供具有读取和写入权限的超级用户帐户。用户帐户必须是 AWS 标识与访问管理 (IAM) 系统中的电源访问策略 (PowerUserAccess) 的成员。</p> <ul style="list-style-type: none"> ■ 20 位访问密钥 ID 和相应的私有访问密钥 <p>如果您使用的是外部 HTTP Internet 代理，则必须针对 IPv4 对其进行配置。</p> <p>vRealize Automation 基于操作的可扩展性 (ABX) 和外部 IPAM 集成可能需要额外的权限。</p> <p>要允许 Auto Scaling 功能，建议具备以下 AWS 权限：</p> <ul style="list-style-type: none"> ■ Auto Scaling 操作： <ul style="list-style-type: none"> ■ autoscaling:DescribeAutoScalingInstances ■ autoscaling:AttachInstances ■ autoscaling>DeleteLaunchConfiguration ■ autoscaling:DescribeAutoScalingGroups ■ autoscaling>CreateAutoScalingGroup ■ autoscaling:UpdateAutoScalingGroup ■ autoscaling>DeleteAutoScalingGroup ■ autoscaling:DescribeLoadBalancers ■ Auto Scaling 资源： <ul style="list-style-type: none"> ■ * <p>提供所有 Auto Scaling 资源权限。</p> <p>要允许 AWS Security Token Service (AWS STS) 功能对 AWS 身份和访问支持临时、有限特权凭据，需要具备以下权限：</p> <ul style="list-style-type: none"> ■ AWS STS 资源： <ul style="list-style-type: none"> ■ * <p>提供所有 STS 资源权限。</p> <p>要允许 EC2 功能，需要具备以下 AWS 权限：</p> <ul style="list-style-type: none"> ■ EC2 操作： <ul style="list-style-type: none"> ■ ec2:AttachVolume ■ ec2:AuthorizeSecurityGroupIngress ■ ec2>DeleteSubnet ■ ec2>DeleteSnapshot ■ ec2:DescribeInstances ■ ec2>DeleteTags ■ ec2:DescribeRegions ■ ec2:DescribeVolumesModifications ■ ec2>CreateVpc ■ ec2:DescribeSnapshots ■ ec2:DescribeInternetGateways ■ ec2>DeleteVolume ■ ec2:DescribeNetworkInterfaces ■ ec2:StartInstances ■ ec2:DescribeAvailabilityZones ■ ec2:CreateInternetGateway ■ ec2:CreateSecurityGroup ■ ec2:DescribeVolumes ■ ec2>CreateSnapshot |

要执行的操作...

所需内容 ...

- ec2:ModifyInstanceAttribute
- ec2:DescribeRouteTables
- ec2:DescribeInstanceStatus
- ec2:DetachVolume
- ec2:RebootInstances
- ec2:AuthorizeSecurityGroupEgress
- ec2:ModifyVolume
- ec2:TerminateInstances
- ec2:DescribeSpotFleetRequestHistory
- ec2:DescribeTags
- ec2:CreateTags
- ec2:RunInstances
- ec2:DescribeNatGateways
- ec2:StopInstances
- ec2:DescribeSecurityGroups
- ec2:CreateVolume
- ec2:DescribeSpotFleetRequests
- ec2:DescribeImages
- ec2:DescribeVpcs
- ec2>DeleteSecurityGroup
- ec2>DeleteVpc
- ec2:CreateSubnet
- ec2:DescribeSubnets
- ec2:RequestSpotFleet

注 vRealize Automation 基于操作的可扩展性 (ABX) 或外部 IPAM 集成不需要 SpotFleet 请求权限。

- EC2 资源:

- *

提供所有 EC2 资源权限。

要允许弹性负载均衡功能，需要具备以下 AWS 权限：

- 负载均衡器操作：

- elasticloadbalancing:DeleteLoadBalancer
- elasticloadbalancing:DescribeLoadBalancers
- elasticloadbalancing:RemoveTags
- elasticloadbalancing:CreateLoadBalancer
- elasticloadbalancing:DescribeTags
- elasticloadbalancing:ConfigureHealthCheck
- elasticloadbalancing:AddTags
- elasticloadbalancing:CreateTargetGroup
- elasticloadbalancing>DeleteLoadBalancerListeners
- elasticloadbalancing:DeregisterInstancesFromLoadBalancer
- elasticloadbalancing:RegisterInstancesWithLoadBalancer
- elasticloadbalancing:CreateLoadBalancerListeners

- 负载均衡器资源：

- *

**要执行的操
作...****所需内容 ...**

提供所有负载均衡器资源权限。

可以启用以下 AWS 身份与访问管理 (IAM) 权限，但这些权限不是必需权限：

- iam:SimulateCustomPolicy
 - iam:GetUser
 - iam:ListUserPolicies
 - iam:GetUserPolicy
 - iam:ListAttachedUserPolicies
 - iam:GetPolicyVersion
 - iam:ListGroupsForUser
 - iam:ListGroupPolicies
 - iam:GetGroupPolicy
 - iam:ListAttachedGroupPolicies
 - iam:ListPolicyVersions
-

| 要执行的操作... | 所需内容 ... |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 添加 Microsoft Azure 云帐户 | <p>配置 Microsoft Azure 实例，并获取可以在其中使用订阅 ID 的有效 Microsoft Azure 订阅。</p> <p>按照 Microsoft Azure 产品文档中的如何：使用门户创建可访问资源的 Azure AD 应用程序和服务主体所述创建 Active Directory 应用程序。</p> <p>如果您使用的是外部 HTTP Internet 代理，则必须针对 IPv4 对其进行配置。</p> <p>记录以下信息：</p> <ul style="list-style-type: none"> ■ 订阅 ID <ul style="list-style-type: none"> 允许您访问 Microsoft Azure 订阅。 ■ 租户 ID <ul style="list-style-type: none"> 在 Microsoft Azure 帐户中创建的 Active Directory 应用程序的授权端点。 ■ 客户端应用程序 ID <ul style="list-style-type: none"> 用于访问 Microsoft Azure 个人帐户中的 Microsoft Active Directory。 ■ 客户端应用程序密钥 <ul style="list-style-type: none"> 生成的唯一的密钥，用于与客户端应用程序 ID 配对。 <p>创建和验证 Microsoft Azure 云帐户需要以下权限：</p> <ul style="list-style-type: none"> ■ Microsoft 计算 <ul style="list-style-type: none"> ■ Microsoft.Compute/virtualMachines/extensions/write ■ Microsoft.Compute/virtualMachines/extensions/read ■ Microsoft.Compute/virtualMachines/extensions/delete ■ Microsoft.Compute/virtualMachines/deallocate/action ■ Microsoft.Compute/virtualMachines/delete ■ Microsoft.Compute/virtualMachines/powerOff/action ■ Microsoft.Compute/virtualMachines/read ■ Microsoft.Compute/virtualMachines/restart/action ■ Microsoft.Compute/virtualMachines/start/action ■ Microsoft.Compute/virtualMachines/write ■ Microsoft.Compute/availabilitySets/write ■ Microsoft.Compute/availabilitySets/read ■ Microsoft.Compute/availabilitySets/delete ■ Microsoft.Compute/disks/delete ■ Microsoft.Compute/disks/read ■ Microsoft.Compute/disks/write ■ Microsoft 网络 <ul style="list-style-type: none"> ■ Microsoft.Network/loadBalancers/backendAddressPools/join/action ■ Microsoft.Network/loadBalancers/delete ■ Microsoft.Network/loadBalancers/read ■ Microsoft.Network/loadBalancers/write ■ Microsoft.Network/networkInterfaces/join/action ■ Microsoft.Network/networkInterfaces/read ■ Microsoft.Network/networkInterfaces/write ■ Microsoft.Network/networkInterfaces/delete ■ Microsoft.Network/networkSecurityGroups/join/action ■ Microsoft.Network/networkSecurityGroups/read ■ Microsoft.Network/networkSecurityGroups/write |

| 要执行的操作... | 所需内容 ... |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> ■ Microsoft.Network/networkSecurityGroups/delete ■ Microsoft.Network/publicIPAddresses/delete ■ Microsoft.Network/publicIPAddresses/join/action ■ Microsoft.Network/publicIPAddresses/read ■ Microsoft.Network/publicIPAddresses/write ■ Microsoft.Network/virtualNetworks/read ■ Microsoft.Network/virtualNetworks/subnets/delete ■ Microsoft.Network/virtualNetworks/subnets/join/action ■ Microsoft.Network/virtualNetworks/subnets/read ■ Microsoft.Network/virtualNetworks/subnets/write ■ Microsoft.Network/virtualNetworks/write ■ Microsoft 资源 <ul style="list-style-type: none"> ■ Microsoft.Resources/subscriptions/resourcegroups/delete ■ Microsoft.Resources/subscriptions/resourcegroups/read ■ Microsoft.Resources/subscriptions/resourcegroups/write ■ Microsoft 存储 <ul style="list-style-type: none"> ■ Microsoft.Storage/storageAccounts/delete ■ Microsoft.Storage/storageAccounts/listKeys/action ■ Microsoft.Storage/storageAccounts/read ■ Microsoft.Storage/storageAccounts/write ■ Microsoft Web <ul style="list-style-type: none"> ■ Microsoft.Web/sites/read ■ Microsoft.Web/sites/write ■ Microsoft.Web/sites/delete ■ Microsoft.Web/sites/config/read ■ Microsoft.Web/sites/config/write ■ Microsoft.Web/sites/config/list/action ■ Microsoft.Web/sites/publishxml/action ■ Microsoft.Web/serverfarms/write ■ Microsoft.Web/serverfarms/delete ■ Microsoft.Web/sites/hostruntime/functions/keys/read ■ Microsoft.Web/sites/hostruntime/host/read ■ Microsoft.web/sites/functions/masterkey/read |
| | <p>如果要将 Microsoft Azure 与基于操作的可扩展性配合使用，除了最小权限外，还需要以下权限：</p> <ul style="list-style-type: none"> ■ Microsoft.Web/sites/read ■ Microsoft.Web/sites/write ■ Microsoft.Web/sites/delete ■ Microsoft.Web/sites/config/read ■ Microsoft.Web/sites/config/write ■ Microsoft.Web/sites/config/list/action ■ Microsoft.Web/sites/publishxml/action ■ Microsoft.Web/serverfarms/write ■ Microsoft.Web/serverfarms/delete ■ Microsoft.Web/sites/hostruntime/functions/keys/read |

要执行的操
作...

所需内容 ...

- Microsoft.Web/sites/hostruntime/host/read
- Microsoft.Web/sites/functions/masterkey/read

如果要将 Microsoft Azure 与包含扩展的基于操作的可扩展性配合使用，还需要以下权限：

- Microsoft.Compute/virtualMachines/extensions/write
- Microsoft.Compute/virtualMachines/extensions/read
- Microsoft.Compute/virtualMachines/extensions/delete

| 要执行的操作... | 所需内容 ... |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 添加 Google Cloud Platform (GCP) 云帐户 | <p>Google Cloud Platform 云帐户与 Google Cloud Platform 计算引擎交互。</p> <p>创建和验证 Google Cloud Platform 云帐户需要项目管理员和所有者凭据。</p> <p>如果您使用的是外部 HTTP Internet 代理，则必须针对 IPv4 对其进行配置。</p> <p>必须启用计算引擎服务。在 vRealize Automation 中创建云帐户时，请使用在初始化计算引擎时创建的服务帐户。</p> <p>还需要以下计算引擎权限，具体取决于用户可以执行的操作：</p> <ul style="list-style-type: none"> ■ <code>roles/compute.admin</code> <p>用于完全控制所有计算引擎资源。</p> ■ <code>roles/iam.serviceAccountUser</code> <p>用于访问管理已配置为作为服务帐户运行的虚拟机实例的用户。授予对以下资源和服务的访问权限：</p> <ul style="list-style-type: none"> ■ <code>compute.*</code> ■ <code>resourcemanager.projects.get</code> ■ <code>resourcemanager.projects.list</code> ■ <code>serviceusage.quotas.get</code> ■ <code>serviceusage.services.get</code> ■ <code>serviceusage.services.list</code> ■ <code>roles/compute.imageUser</code> <p>提供列出和读取映像的权限，而无需对映像具有其他权限。在项目级别授予 <code>compute.imageUser</code> 角色，使用户能够列出项目中的所有映像。它还允许用户根据项目中的映像创建实例和永久磁盘等资源。</p> <ul style="list-style-type: none"> ■ <code>compute.images.get</code> ■ <code>compute.images.getFromFamily</code> ■ <code>compute.images.list</code> ■ <code>compute.images.useReadOnly</code> ■ <code>resourcemanager.projects.get</code> ■ <code>resourcemanager.projects.list</code> ■ <code>serviceusage.quotas.get</code> ■ <code>serviceusage.services.get</code> ■ <code>serviceusage.services.list</code> ■ <code>roles/compute.instanceAdmin</code> <p>提供创建、修改和删除虚拟机实例的权限。这包括创建、修改和删除磁盘以及配置受防护 VMBETA 设置的权限。</p> <p>对于管理虚拟机实例（但不是网络或安全设置或作为服务帐户运行的实例）的用户，将此角色授予包含实例的组织、文件夹或项目，或者授予单个实例。</p> <p>管理已配置为作为服务帐户运行的虚拟机实例的用户还需要 <code>roles/iam.serviceAccountUser</code> 角色。</p> <ul style="list-style-type: none"> ■ <code>compute.acceleratorTypes</code> ■ <code>compute.addresses.get</code> ■ <code>compute.addresses.list</code> ■ <code>compute.addresses.use</code> ■ <code>compute.autoscalers</code> ■ <code>compute.diskTypes</code> ■ <code>compute.disks.create</code> ■ <code>compute.disks.createSnapshot</code> ■ <code>compute.disks.delete</code> ■ <code>compute.disks.get</code> |

要执行的操作...

所需内容 ...

- compute.disks.list
- compute.disks.resize
- compute.disks.setLabels
- compute.disks.update
- compute.disks.use
- compute.disks.useReadOnly
- compute.globalAddresses.get
- compute.globalAddresses.list
- compute.globalAddresses.use
- compute.globalOperations.get
- compute.globalOperations.list
- compute.images.get
- compute.images.getFromFamily
- compute.images.list
- compute.images.useReadOnly
- compute.instanceGroupManagers
- compute.instanceGroups
- compute.instanceTemplates
- compute.instances
- compute.licenses.get
- compute.licenses.list
- compute.machineTypes
- compute.networkEndpointGroups
- compute.networks.get
- compute.networks.list
- compute.networks.use
- compute.networks.useExternallp
- compute.projects.get
- compute.regionOperations.get
- compute.regionOperations.list
- compute.regions
- compute.reservations.get
- compute.reservations.list
- compute.subnetworks.get
- compute.subnetworks.list
- compute.subnetworks.use
- compute.subnetworks.useExternallp
- compute.targetPools.get
- compute.targetPools.list
- compute.zoneOperations.get
- compute.zoneOperations.list
- compute.zones
- resourcemanager.projects.get
- resourcemanager.projects.list

| 要执行的操作... | 所需内容 ... |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> ■ serviceusage.quotas.get ■ serviceusage.services.get ■ serviceusage.services.list ■ roles/compute.instanceAdmin.v1 <p>用于完全控制计算引擎实例、实例组、磁盘、快照和映像。还提供对所有计算引擎网络资源的读取访问权限。</p> |
| | <p>注 如果在实例级别为用户授予此角色，则该用户无法创建新的实例。</p> <ul style="list-style-type: none"> ■ compute.acceleratorTypes ■ compute.addresses.get ■ compute.addresses.list ■ compute.addresses.use ■ compute.autoscalers ■ compute.backendBuckets.get ■ compute.backendBuckets.list ■ compute.backendServices.get ■ compute.backendServices.list ■ compute.diskTypes ■ compute.disks ■ compute.firewalls.get ■ compute.firewalls.list ■ compute.forwardingRules.get ■ compute.forwardingRules.list ■ compute.globalAddresses.get ■ compute.globalAddresses.list ■ compute.globalAddresses.use ■ compute.globalForwardingRules.get ■ compute.globalForwardingRules.list ■ compute.globalOperations.get ■ compute.globalOperations.list ■ compute.healthChecks.get ■ compute.healthChecks.list ■ compute.httpHealthChecks.get ■ compute.httpHealthChecks.list ■ compute.httpsHealthChecks.get ■ compute.httpsHealthChecks.list ■ compute.images ■ compute.instanceGroupManagers ■ compute.instanceGroups ■ compute.instanceTemplates ■ compute.instances ■ compute.interconnectAttachments.get ■ compute.interconnectAttachments.list ■ compute.interconnectLocations ■ compute.interconnects.get |

要执行的操作...

所需内容 ...

- compute.interconnects.list
- compute.licenseCodes
- compute.licenses
- compute.machineTypes
- compute.networkEndpointGroups
- compute.networks.get
- compute.networks.list
- compute.networks.use
- compute.networks.useExternallp
- compute.projects.get
- compute.projects.setCommonInstanceMetadata
- compute.regionBackendServices.get
- compute.regionBackendServices.list
- compute.regionOperations.get
- compute.regionOperations.list
- compute.regions
- compute.reservations.get
- compute.reservations.list
- compute.resourcePolicies
- compute.routers.get
- compute.routers.list
- compute.routes.get
- compute.routes.list
- compute.snapshots
- compute.sslCertificates.get
- compute.sslCertificates.list
- compute.sslPolicies.get
- compute.sslPolicies.list
- compute.sslPolicies.listAvailableFeatures
- compute.subnetworks.get
- compute.subnetworks.list
- compute.subnetworks.use
- compute.subnetworks.useExternallp
- compute.targetHttpProxies.get
- compute.targetHttpProxies.list
- compute.targetHttpsProxies.get
- compute.targetHttpsProxies.list
- compute.targetInstances.get
- compute.targetInstances.list
- compute.targetPools.get
- compute.targetPools.list
- compute.targetSslProxies.get
- compute.targetSslProxies.list
- compute.targetTcpProxies.get

| 要执行的操作... | 所需内容 ... |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> ■ compute.targetTcpProxies.list ■ compute.targetVpnGateways.get ■ compute.targetVpnGateways.list ■ compute.urlMaps.get ■ compute.urlMaps.list ■ compute.vpnTunnels.get ■ compute.vpnTunnels.list ■ compute.zoneOperations.get ■ compute.zoneOperations.list ■ compute.zones ■ resourceManager.projects.get ■ resourceManager.projects.list ■ serviceusage.quotas.get ■ serviceusage.services.get ■ serviceusage.services.list |
| 添加 NSX-T 云帐户 | <p>提供具有以下读取和写入权限的帐户：</p> <ul style="list-style-type: none"> ■ NSX-T 企业级管理员角色和访问凭据 ■ NSX-T IP 地址或 FQDN <p>管理员还需要访问 vCenter Server，如此页面上以下“基于 vCenter 的云帐户的 vSphere 代理要求”部分中所述。</p> |
| 添加 NSX-V 云帐户 | <p>提供具有以下读取和写入权限的帐户：</p> <ul style="list-style-type: none"> ■ NSX-V 企业级管理员角色和访问凭据 ■ NSX-V IP 地址或 FQDN <p>管理员还需要访问 vCenter Server，如此页面上以下“基于 vCenter 的云帐户的 vSphere 代理要求”部分中所述。</p> |
| 添加 vCenter 云帐户 | <p>提供具有以下读取和写入权限的帐户：</p> <ul style="list-style-type: none"> ■ vCenter IP 地址或 FQDN <p>管理员还需要访问 vCenter Server，如此页面上以下“基于 vCenter 的云帐户的 vSphere 代理要求”部分中所述。</p> |
| 添加 VMware Cloud on AWS (VMC) 云帐户 | <p>提供具有以下读取和写入权限的帐户：</p> <ul style="list-style-type: none"> ■ cloudadmin@vmc.local 帐户或 CloudAdmin 组中的任何用户帐户 ■ NSX 企业级管理员角色和访问凭据 ■ 对您组织的 VMware Cloud on AWS SDDC 环境的 NSX 云管理员访问权限 ■ 对您组织的 VMware Cloud on AWS SDDC 环境的管理员访问权限 ■ 您组织的 VMware Cloud on AWS 服务中的 VMware Cloud on AWS 环境的 VMware Cloud on AWS API 令牌 ■ vCenter IP 地址或 FQDN <p>管理员还需要访问您的目标 VMware Cloud on AWS SDDC 使用的 vCenter，它具有本页以下“基于 vSphere 的云帐户的 vCenter 代理要求”部分中列出的所有权限。</p> <p>有关创建和使用 VMware Cloud on AWS 云帐户所需的权限的详细信息，请参见 VMware Cloud on AWS 产品文档 中的管理 VMware Cloud on AWS 数据中心。</p> |

基于 vCenter 的云帐户的 vSphere 代理要求

下表列出了管理 VMware Cloud on AWS 和 vCenter 云帐户所需的权限。必须对 vCenter Server 中的所有集群启用这些权限，而不仅仅是托管端点的集群。

对于所有基于 vCenter Server 的云帐户（包括 NSX-V、NSX-T、vCenter 和 VMware Cloud on AWS），管理员必须拥有 vSphere 端点凭据，或者在 vCenter 中运行代理服务的凭据，从而提供对主机 vCenter Server 的管理访问权限。

有关 vSphere 代理要求的详细信息，请参见 [VMware vSphere 产品文档](#)。

表 2-4. vSphere 代理管理 vCenter Server 实例所需的权限

| 属性值 | 权限 |
|--------|--------------------------------------------------------------------------------------------------------------|
| 数据存储 | <ul style="list-style-type: none"> ■ 分配空间 ■ 浏览数据存储 ■ 低级别文件操作 |
| 数据存储集群 | 配置数据存储集群 |
| 文件夹 | <ul style="list-style-type: none"> ■ 创建文件夹 ■ 删除文件夹 |
| 全局 | <ul style="list-style-type: none"> ■ 管理自定义属性 ■ 设置自定义属性 |
| 网络 | 分配网络 |
| 权限 | 修改权限 |
| 资源 | <ul style="list-style-type: none"> ■ 将虚拟机分配给资源池 ■ 迁移已关闭电源的虚拟机 ■ 迁移已打开电源的虚拟机 |

表 2-4. vSphere 代理管理 vCenter Server 实例所需的权限（续）

| 属性值 | 权限 |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 内容库 | <p>要分配内容库的权限，管理员必须将该权限作为全局权限授予用户。有关相关信息，请参见位于 VMware vSphere 文档 内《vSphere 虚拟机管理》中的内容库权限的层次结构继承。</p> <ul style="list-style-type: none"> ■ 添加库项目 ■ 创建本地库 ■ 创建已订阅库 ■ 删除库项目 ■ 删除本地库 ■ 删除已订阅库 ■ 下载文件 ■ 逐出库项目 ■ 逐出已订阅库 ■ 探查订阅信息 ■ 读取存储 ■ 同步库项目 ■ 同步已订阅库 ■ 类型自检 ■ 更新配置设置 ■ 更新文件 ■ 更新库 ■ 更新库项目 ■ 更新本地库 ■ 更新已订阅库 ■ 查看配置设置 |
| 标记 | <ul style="list-style-type: none"> ■ 分配或取消分配 vSphere 标记 ■ 创建 vSphere 标记 ■ 创建 vSphere 标记类别 ■ 删除 vSphere 标记 ■ 删除 vSphere 标记类别 ■ 标记 vSphere 标记 ■ 编辑 vSphere 标记类别 ■ 修改类别的使用者字段 ■ 修改标记的使用者字段 |
| vApp | <ul style="list-style-type: none"> ■ 导入 ■ vApp 应用程序配置 <p>OVF 模板以及从内容库置备虚拟机时都需要使用 <code>vApp.Import</code> 应用程序配置。</p> <p>使用 <code>cloud-init</code> 执行云配置脚本时，需要 <code>vApp.vApp</code> 应用程序配置。此设置允许修改 vApp 的内部结构，例如产品信息和属性。</p> |
| 虚拟机 - 清单 | <ul style="list-style-type: none"> ■ 从现有项创建 ■ 新建 ■ 移动 ■ 移除 |

表 2-4. vSphere 代理管理 vCenter Server 实例所需的权限（续）

| 属性值 | 权限 |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 虚拟机 - 交互 | <ul style="list-style-type: none"> ■ 配置 CD 媒体 ■ 控制台交互 ■ 设备连接 ■ 关闭电源 ■ 打开电源 ■ 重置 ■ 挂起 ■ 工具安装 |
| 虚拟机 - 配置 | <ul style="list-style-type: none"> ■ 添加现有磁盘 ■ 添加新磁盘 ■ 移除磁盘 ■ 高级 ■ 更改 CPU 数目 ■ 更改资源 ■ 扩展虚拟磁盘 ■ 磁盘更改跟踪 ■ 内存 ■ 修改设备设置 ■ 重命名 ■ 设置注释 ■ 设置 ■ 交换文件位置 |
| 虚拟机 - 置备 | <ul style="list-style-type: none"> ■ 自定义 ■ 克隆模板 ■ 克隆虚拟机 ■ 部署模板 ■ 读取自定义规范 |
| 虚拟机 - 状态 | <ul style="list-style-type: none"> ■ 创建快照 ■ 移除快照 ■ 恢复到快照 |

配置 Microsoft Azure 以与 vRealize Automation Cloud Assembly 配合使用

要在 vRealize Automation Cloud Assembly 中创建 Microsoft Azure 云帐户，您必须收集一些信息并执行一些配置。

步骤

1 找到并记录您的 Microsoft Azure 订阅 ID 和租户 ID。

- 订阅 ID - 单击 Azure 门户左侧工具栏上的“订阅”图标可查看订阅 ID。
- 租户 ID - 单击 Azure 门户中的“帮助”图标，然后选择“显示诊断”。搜索租户并在找到后记录该 ID。

2 您可以创建新存储帐户和资源组以开始执行操作。或者，也可以稍后在蓝图中创建。

- 存储帐户 - 使用以下过程配置帐户。

- 1 在 Azure 门户中，找到侧栏上的“存储帐户”图标。确保选择正确的订阅，然后单击**添加**。还可以在 Azure 搜索字段中搜索存储帐户。
- 2 输入存储帐户的必填信息。您需要提供订阅 ID。
- 3 选择是使用现有资源组还是创建新资源组。记下您的资源组名称，因为稍后需要用到该名称。

注 保存您的存储帐户位置，因为稍后需要用到该位置。

3 创建虚拟网络。或者，如果您有合适的现有网络，则可以选择该网络。

如果要创建网络，必须选择“使用现有资源组”并指定在上一步中创建的组。此外，选择以前指定的相同位置。如果对象将使用的所有适用组件之间的位置不匹配，Microsoft Azure 将不会部署虚拟机或其他对象。

- a 在左侧面板中找到“虚拟网络”图标并单击，或搜索虚拟网络。确保选择正确的订阅，然后单击**添加**。
- b 输入新虚拟网络的唯一名称，并进行记录以供稍后使用。
- c 在**地址空间**字段中输入虚拟网络的相应 IP 地址。
- d 确保选择正确的订阅，然后单击**添加**。
- e 输入其余的基本配置信息。
- f 您可以根据需要修改其他选项，但对于大多数配置，可以保留默认值。
- g 单击**创建**。

4 设置 Azure Active Directory 应用程序，以便 vRA 可以进行身份验证。

- a 在 Azure 左侧菜单中找到 Active Directory 图标，然后单击该图标。
- b 单击**应用注册**，然后选择**添加**。
- c 键入符合 Azure 名称验证的应用程序的名称。
- d 保留“Web 应用/API”作为“应用程序类型”。
- e “登录 URL”可以是适合您使用情况的任何值。
- f 单击**创建**。

5 创建密钥以在 Cloud Assembly 中对应用程序进行身份验证。

- a 在 Azure 中单击应用程序的名称。
记下应用程序 ID，供稍后使用。
- b 单击下一个窗格中的**所有设置**，然后从设置列表中选择“密钥”。
- c 输入新密钥的描述并选择持续时间。
- d 单击**保存**并确保将该密钥值复制到安全位置，因为稍后将无法检索该密钥值。

- e 在左侧菜单中，选择应用程序的 **API 权限**，然后单击**添加权限**以创建新权限。
 - f 在“选择 API”页面上选择“Azure 服务管理”。
 - g 单击**委派的权限**。
 - h 在“选择权限”下，选择 `user_impersonation`，然后单击**添加权限**。
- 6 授权您的 Active Directory 应用程序连接到 Azure 订阅，以便可以部署和管理虚拟机。**
- a 在左侧菜单中，单击“订阅”图标，然后选择新订阅。
您可能需要单击名称文本才能使面板侧拉。
 - b 选择“访问控制 (IAM)”选项以查看对订阅的权限。
 - c 单击“添加角色分配”标题下的**添加**。
 - d 从“角色”下拉列表中选择“参与者”。
 - e 在“将访问权限分配给”下拉列表中，保留默认选择。
 - f 在“选择”框中键入应用程序的名称。
 - g 单击**保存**。
 - h 添加其他角色，使新应用程序具有“所有者”、“参与者”和“读者”角色。
 - i 单击**保存**。

后续步骤

您必须安装 **Microsoft Azure 命令行界面工具**。对于 **Windows** 和 **Mac** 操作系统，可免费使用这些工具。有关下载和安装这些工具的详细信息，请参见 **Microsoft 文档**。

安装命令行界面后，必须对新订阅进行身份验证。

- 1 打开终端窗口，然后键入您的 **Microsoft Azure** 登录信息。您将收到一个 URL 以及一个可供您进行身份验证的短代码。
- 2 在浏览器中，输入从设备上的应用程序收到的代码。
- 3 输入您的“身份验证代码”，然后单击**继续**。
- 4 选择您的 **Azure** 帐户并登录。

如果您有多个订阅，请确保使用 `azure account set <subscription-name>` 命令选择正确的订阅。

- 5 在继续操作之前，必须使用 `azure provider register microsoft.compute` 命令将 **Microsoft.Compute** 提供程序注册到新 **Azure** 订阅。

如果命令在首次运行时超时并生成错误，请重新运行。

完成配置后，可以使用 `azure vm image list` 命令检索可用的虚拟机映像名称。您可以选择所需映像，记录为其提供的 URN，并稍后在蓝图中使用。

在 vRealize Automation 中创建 Microsoft Azure 云帐户

作为云管理员，您可以为团队将在其中部署 vRealize Automation 蓝图的帐户区域创建 Microsoft Azure 云帐户。

要查看 Microsoft Azure 云帐户在 vRealize Automation 中的工作方式的示例用例，请参见 [WordPress 用例](#)。

前提条件

- 确认您具有所需的管理员凭据，并且已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有所需的用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 配置 Microsoft Azure 帐户与 vRealize Automation 配合使用。请参见 [配置 Microsoft Azure 以与 vRealize Automation Cloud Assembly 配合使用](#)。
- 如果您没有外部 Internet 访问，请配置一个 Internet 服务器代理。请参见 [如何配置 vRealize Automation 的 Internet 代理服务器](#)。

步骤

- 1 选择 **基础架构 > 连接 > 云帐户**，然后单击 **添加云帐户**。
- 2 选择 Microsoft Azure 帐户类型，然后输入凭据和其他值。
- 3 单击 **验证**。
将收集与该帐户相关联的帐户区域。
- 4 选择要将此资源置备到的区域。
- 5 为了提高效率，单击 **为选定区域创建云区域**。
- 6 如果需要添加标记以支持标记策略，请输入功能标记。请参见 [如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署和创建标记策略](#)。
- 7 单击 **保存**。

结果

该帐户将添加到 vRealize Automation 中，所选区域可用于指定的云区域。

后续步骤

为此云帐户创建基础架构资源。

在 vRealize Automation 中创建 Amazon Web Services 云帐户

作为云管理员，您可以为团队将在其中部署 vRealize Automation 蓝图的帐户区域创建 Amazon Web Services (AWS) 云帐户。

前提条件

- 确认您具有所需的管理员凭据，并且已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。
- 确认您具有所需的用户角色。请参见 [vRealize Automation Cloud Assembly](#) 用户角色是什么。
- 确认您具有所需的 AWS 管理员凭据。
- 如果您没有外部 Internet 访问，请配置一个 Internet 服务器代理。请参见[如何配置 vRealize Automation 的 Internet 代理服务器](#)。

步骤

- 1 选择**基础架构 > 连接 > 云帐户**，然后单击**添加云帐户**。
- 2 选择 AWS 帐户类型，然后输入凭据和其他值。
- 3 单击**验证**。
将收集与该帐户相关联的帐户区域。
- 4 选择要将此资源置备到的区域。
- 5 为了提高效率，单击**为选定区域创建云区域**。
- 6 如果需要添加标记以支持标记策略，请输入功能标记。请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署和创建标记策略](#)。
- 7 单击**添加**。

结果

该帐户将添加到 vRealize Automation 中，所选区域可用于指定的云区域。

后续步骤

为此云帐户配置基础架构资源。

在 vRealize Automation 中创建 Google Cloud Platform 云帐户

作为云管理员，您可以为团队将在其中部署 vRealize Automation 蓝图的帐户区域创建 Google Cloud Platform (GCP) 云帐户。

前提条件

- 确认您具有所需的管理员凭据，并且已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。
- 确认您具有所需的用户角色。请参见 [vRealize Automation Cloud Assembly](#) 用户角色是什么。
- 确认您有权访问 Google Cloud Platform JSON 安全密钥。
- 确认您拥有 Google Cloud Platform 实例所需的安全信息。这些信息大部分可以从您的实例或 Google 文档获取。

- 如果您没有外部 Internet 访问，请配置一个 Internet 服务器代理。请参见[如何配置 vRealize Automation 的 Internet 代理服务器](#)。

步骤

- 1 选择**基础架构 > 连接 > 云帐户**，然后单击**添加云帐户**。
- 2 选择 Google Cloud Platform 帐户类型，并输入相应的凭据和相关信息。使用在初始化源 GCP 帐户计算引擎时创建的服务帐户。

如上面的**必备条件**部分中所述，可从在 [vRealize Automation 中使用云帐户所需的凭据](#) 了解凭据要求。要在 vRealize Automation 中成功创建云帐户，源 GCP 帐户必须启用计算引擎服务。

在 vRealize Automation 中，项目 ID 是 Google Cloud Platform 端点的一部分。您可以在创建云帐户时指定该 ID。在对特定于项目的专用映像收集数据期间，vRealize Automation GCP 适配器会查询 Google Cloud Platform API。

- 3 单击**验证**。
将收集与该帐户相关联的帐户区域。
- 4 选择要将此资源置备到的区域。
- 5 为了提高效率，单击**为选定区域创建云区域**。
- 6 如果需要标记以支持标记策略，请输入功能标记。请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署和创建标记策略](#)。
- 7 单击**添加**。

结果

该帐户将添加到 vRealize Automation 中，所选区域可用于指定的云区域。

后续步骤

为此云帐户创建基础架构资源。

在 vRealize Automation Cloud Assembly 中创建 vCenter 云帐户

可以为要部署 vRealize Automation Cloud Assembly 蓝图的帐户区域添加 vCenter 云帐户。

出于网络和安全目的，您可以将 NSX-T 或 NSX-V 云帐户与 vCenter 云帐户相关联。

前提条件

- 确认您具有所需的管理员凭据，并且已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您已正确配置端口和协议以支持云帐户。请参阅 [vRealize Automation 产品文档](#) 中的使用 vRealize Easy Installer 安装 vRealize Automation 中的 vRealize Automation 的端口和协议主题以及《vRealize Automation 参考架构指南》中的端口要求主题。

步骤

1 选择**基础架构 > 连接 > 云帐户**，然后单击**添加云帐户**。

2 选择 vCenter 帐户类型，然后输入 vCenter Server 主机 IP 地址。

3 输入您的 vCenter Server 管理员凭据，然后单击**验证**。

将收集与该帐户关联的数据中心。

4 在指定的 vCenter Server 上至少选择一个可用数据中心，以便置备此云帐户。

5 为了提高效率，创建一个云区域，以便置备到选定的数据中心。

您还可以根据组织的云策略，在单独的步骤中创建云区域。

有关云区域的信息，请参见[了解有关 vRealize Automation Cloud Assembly 云区域的更多信息](#)。

6 选择现有的 NSX 云帐户。

您可以现在或稍后编辑云帐户时选择 NSX 帐户。

有关 NSX-V 云帐户的信息，请参见在 [vRealize Automation Cloud Assembly 中创建 NSX-V 云帐户](#)。

有关 NSX-T 云帐户的信息，请参见在 [vRealize Automation Cloud Assembly 中创建 NSX-T 云帐户](#)。

7 如果要添加标记以支持标记策略，请输入功能标记。

您可以现在或稍后编辑云帐户时添加标记。有关标记的信息，请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署](#)。

8 单击**保存**。

结果

将添加云帐户，并且选定的数据中心可用于指定的云区域。收集的数据（例如计算机和卷）将列在“基础架构”选项卡的“资源”部分中。

后续步骤

为此云帐户配置其余基础架构资源。请参见第 4 章 [构建您的 vRealize Automation Cloud Assembly 资源基础架构](#)。

在 vRealize Automation Cloud Assembly 中创建 NSX-V 云帐户

出于网络和安全目的，您可以创建 NSX-V 云帐户并将其与 vCenter 云帐户相关联。

前提条件

- 确认您具有所需的管理员凭据，并且已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。

- 确认您具有与此 NSX 云帐户配合使用的 vCenter 云帐户。请参见在 [vRealize Automation Cloud Assembly 中创建 vCenter 云帐户](#)。
- 确认您已正确配置端口和协议以支持云帐户。请参阅 [vRealize Automation 产品文档](#) 中的使用 vRealize Easy Installer 安装 vRealize Automation 中的 vRealize Automation 的端口和协议主题以及《vRealize Automation 参考架构指南》中的端口要求主题。

步骤

- 1 选择**基础架构 > 连接 > 云帐户**，然后单击**添加云帐户**。
- 2 选择 NSX-V 帐户类型，然后输入 NSX-V 主机 IP 地址。
- 3 输入您的 NSX 管理员凭据，然后单击**验证**。
将收集与该帐户关联的资产。
如果 NSX 主机 IP 地址不可用，则验证将失败。
- 4 如果可用，请选择表示要与此 NSX-V 帐户关联的 vCenter 云帐户的 vCenter 端点。
- 5 如果要添加标记以支持标记策略，请输入功能标记。
可以稍后添加或移除功能标记。请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署](#)。
- 6 单击**保存**。

后续步骤

可以创建或编辑 vCenter 云帐户，并将其与此 NSX 云帐户关联。请参见在 [vRealize Automation Cloud Assembly 中创建 vCenter 云帐户](#)。

创建并配置一个或多个云区域，以便与此云帐户使用的数据中心配合使用。请参见了解有关 [vRealize Automation Cloud Assembly 云区域](#)的更多信息。

为此云帐户配置基础架构资源。请参见第 4 章 [构建您的 vRealize Automation Cloud Assembly 资源基础架构](#)。

在 vRealize Automation Cloud Assembly 中创建 NSX-T 云帐户

出于网络和安全目的，您可以创建 NSX-T 云帐户并将其与 vCenter 云帐户相关联。

为了促进部署中的容错能力和高可用性，每个 NSX-T 数据中心端点都代表由 3 个 NSX Manager 构成的集群。

- vRealize Automation 可以指向其中一个 NSX Manager。使用此选项时，一个 NSX Manager 将从 vRealize Automation 接收 API 调用。
- vRealize Automation 可以指向集群的虚拟 IP。使用此选项时，一个 NSX Manager 将取得 VIP 的控制权。该 Manager 将收到来自 vRealize Automation 的 API 调用。如果出现故障，集群中的另一个节点将接管 VIP，并收到来自 vRealize Automation 的 API 调用。

有关 VIP 配置的详细信息，请参见 [VMware NSX-T Data Center 文档](#) 内《NSX-T Data Center 安装指南》中的配置集群的虚拟 IP (VIP) 地址。

- vRealize Automation 可以指向负载平均器 VIP，以将调用负载均衡到 6 个 NSX Manager。使用此选项时，所有三个 NSX Manager 都将收到来自 vRealize Automation 的 API 调用。

您可以在第三方负载均衡器或 NSX-T 负载均衡器上配置 VIP。

对于规模较大的环境，请考虑使用此选项在 3 个 NSX Manager 之间拆分 vRealize Automation API 调用。

前提条件

- 确认您具有所需的管理员凭据，并且已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有与此 NSX 云帐户配合使用的 vCenter 云帐户。请参见在 [vRealize Automation Cloud Assembly 中创建 vCenter 云帐户](#)。
- 确认您已正确配置端口和协议以支持云帐户。请参阅 [vRealize Automation 产品文档](#) 中的使用 vRealize Easy Installer 安装 vRealize Automation 中的 vRealize Automation 的端口和协议主题以及《vRealize Automation 参考架构指南》中的端口要求主题。

步骤

- 1 选择**基础架构 > 连接 > 云帐户**，然后单击**添加云帐户**。
- 2 选择 NSX-T 帐户类型，然后输入 NSX-T 端点管理器实例或 VIP 的主机 IP 地址（请参见上文）。
- 3 输入您的 NSX 管理员凭据，然后单击**验证**。
将收集与该帐户关联的资产。
如果 NSX 主机 IP 地址不可用，则验证将失败。
- 4 （如果可用）选择表示要与此 NSX-T 云帐户关联的 vCenter 云帐户的 vCenter 端点。
- 5 如果要添加标记以支持标记策略，请输入功能标记。
可以稍后添加或移除功能标记。请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署](#)。
- 6 单击**保存**。

后续步骤

可以创建或编辑 vCenter 云帐户，以便与此 NSX 云帐户关联。请参见在 [vRealize Automation Cloud Assembly 中创建 vCenter 云帐户](#)。

创建并配置一个或多个云区域，以便与此云帐户使用的数据中心配合使用。请参见[了解有关 vRealize Automation Cloud Assembly 云区域的更多信息](#)。

为此云帐户配置基础架构资源。请参见第 4 章 [构建您的 vRealize Automation Cloud Assembly 资源基础架构](#)。

在 vRealize Automation 中创建 VMware Cloud on AWS 云帐户

作为云管理员，您可以为团队将在其中部署 vRealize Automation 蓝图的帐户区域创建 VMware Cloud on AWS 云帐户。

在 vRealize Automation 中，VMware Cloud on AWS 需要一些独特的配置过程。要针对 VMware Cloud on AWS 正确配置 vRealize Automation，包括为云帐户设置 API 令牌值以及为其云代理设置网关防火墙规则，请参见 [VMware Cloud on AWS 用例](#) 工作流。

前提条件

- 确认您具有所需的 VMware Cloud on AWS 管理员凭据，包括 vCenter 中目标 SDDC 的 VMware Cloud on AWS CloudAdmin 凭据，并确认已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 如果您没有外部 Internet 访问，请配置一个 Internet 服务器代理。请参见 [如何配置 vRealize Automation 的 Internet 代理服务器](#)。

步骤

- 1 选择 **基础架构 > 连接 > 云帐户**，单击 **添加云帐户** 并选择 VMware Cloud on AWS 帐户类型。
- 2 为您的组织添加 **VMC API 令牌** 以访问可用的 SDDC。

可以在链接的 **API 令牌** 页面上为您的组织创建新令牌或使用现有令牌。有关详细信息，请参见在 [vRealize Automation](#) 的示例工作流中创建 VMware Cloud on AWS 云帐户。

- 3 选择可用于部署的 SDDC。

NSX-V SDDC 不受支持，不会出现在列表中。

将根据 SDDC 自动填充 vCenter 和 NSX-T Manager IP 地址/FQDN 值。

- 4 输入指定 SDDC 的 vCenter 用户名和密码（如果不是默认值 cloudadmin@vmc.local）。
- 5 单击 **验证** 确认您对指定 vCenter 的访问权限，并检查 vCenter 是否正在运行。

将收集与该帐户关联的数据中心。

- 6 为了提高效率，创建一个云区域，以便置备到选定的 SDDC。

您还可以根据组织的云策略，在单独的步骤中创建云区域。

- 7 如果需要添加标记以支持标记策略，请输入功能标记。

可以稍后添加或移除功能标记。请参见 [如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署](#)。

- 8 单击 **保存**。

结果

将添加云帐户，并且选定的 SDDC 可用于指定的云区域。

后续步骤

要针对 VMware Cloud on AWS 正确配置 vRealize Automation，请参见 [VMware Cloud on AWS 用例](#)。

有关 vRealize Automation 之外的 VMware Cloud on AWS 的相关信息，请参见 [VMware Cloud on AWS 文档](#)。

将 vRealize Automation 与其他应用程序集成

集成使您能够将外部系统添加到 vRealize Automation。

集成包括 vRealize Orchestrator、配置管理和其他外部系统，例如 GitHub、Ansible、Puppet 和外部 IPAM 提供程序（如 Infoblox）。

注 如果您没有外部 Internet 访问权限，但您的集成需要该权限，则可以配置一个 Internet 服务器代理。请参见 [如何配置 vRealize Automation 的 Internet 代理服务器](#)。

如何在 vRealize Automation Cloud Assembly 中使用 GitLab 和 GitHub 集成

vRealize Automation Cloud Assembly 支持与 GitLab 和 GitHub 存储库集成，以便您可以在源代码控制下管理蓝图和操作脚本。此功能可简化对部署流程的审计和财务责任落实。

您必须具有相应的本地 Git 存储库并为指定的所有用户配置了访问权限，以便设置 Git 与 vRealize Automation Cloud Assembly 的集成。此外，您还必须在特定结构中保存蓝图，以便 Git 能够检测到它们。要创建与 GitLab 或 GitHub 的集成，请在 Cloud Assembly 中选择 **基础架构 > 连接 > 集成**，然后进行相应的选择。您需要目标存储库的 url 和令牌。

使用现有存储库配置 Git 集成时，与所选项目关联的所有蓝图都可供合格用户使用。您可以将这些蓝图用于现有部署，也可以将其用作新部署的基础。添加项目时，您必须选择与该项目在 Git 中的存储位置和存储方式相关的属性。

您可以直接从 vRealize Automation Cloud Assembly 将操作保存到 Git 存储库。您可以直接在 Git 中对操作脚本进行版本控制，也可以在 vRealize Automation Cloud Assembly 中创建版本。如果在 vRealize Automation Cloud Assembly 中创建操作的版本，则会自动将其作为版本保存到 Git 中。蓝图稍微复杂一些，因为不能从 vRealize Automation Cloud Assembly 将它们直接添加到 Git 集成中。您必须将它们直接保存到 Git 实例中，然后在使用 vRealize Automation Cloud Assembly 中的蓝图管理页面时可以从 Git 中检索它们。

开始前

您必须在特定结构中创建并保存蓝图，以便 GitLab 或 GitHub 检测到它们。

- 配置和存储蓝图，以便正确地与 GitLab 集成。只有有效蓝图才会导入到 GitLab。
 - 为蓝图创建一个或多个指定文件夹。
 - 所有蓝图都必须存储在 `blueprint.yaml` 文件中。
 - 确保蓝图的顶部包括 `name:` 和 `version:` 属性。

- 为适用的存储库提取 API 密钥。在 Git 帐户中，选择右上角的登录名，然后导航到“设置”菜单。选择**访问令牌**，并命名您的令牌，设置到期日期。然后，选择 API 并创建令牌。复制生成的值并保存它。

对于用于 Git 集成的所有蓝图，必须遵守以下准则。

- 每个蓝图必须驻留在单独的文件夹中。
- 所有蓝图都必须按 `blueprint.yaml` 格式命名。
- 所有蓝图 YAML 文件都必须使用 `name` 字段和 `version` 字段。
- 只会导入有效的蓝图。
- 如果您更新从 Git 导入的草稿蓝图，而且该蓝图的内容与最高版本中的不同，则在后续的同步不会更新草稿，并且会创建新版本。如果要更新蓝图并同时允许来自 Git 的后续同步，您必须在完成最终更改之后创建新版本。

■ 在 vRealize Automation Cloud Assembly 中配置 GitLab 蓝图集成

此过程说明了如何在 vRealize Automation Cloud Assembly 中配置 GitLab 集成，以便您可以使用存储库中的蓝图，并自动下载与指定项目关联的已保存蓝图。要将蓝图与 GitLab 配合使用，您必须创建与相应 GitLab 实例的连接，然后将所需蓝图保存到该实例。

■ 在 vRealize Automation Cloud Assembly 中配置 GitHub 集成

可以在 vRealize Automation Cloud Assembly 中集成 GitHub 云端存储库托管服务

在 vRealize Automation Cloud Assembly 中配置 GitLab 蓝图集成

此过程说明了如何在 vRealize Automation Cloud Assembly 中配置 GitLab 集成，以便您可以使用存储库中的蓝图，并自动下载与指定项目关联的已保存蓝图。要将蓝图与 GitLab 配合使用，您必须创建与相应 GitLab 实例的连接，然后将所需蓝图保存到该实例。

使用现有存储库配置 GitLab 集成时，与所选项目关联的所有蓝图将对合格用户可用。您可以将这些蓝图用于现有部署，也可以将其用作新部署的基础。添加项目时，必须选择与该项目在 GitLab 中的存储位置和存储方式相关的属性。

注 您无法从 vRealize Automation Cloud Assembly 将新的或已更新的蓝图推送到 Git 存储库。此外，您无法从 vRealize Automation Cloud Assembly 将新蓝图推送到存储库。要将蓝图添加到存储库，开发人员必须使用 Git 界面。

如果您更新从 Git 导入的草稿蓝图，而且该蓝图的内容与最高版本中的不同，则在后续的同步不会更新草稿，并且会创建新版本。如果要更新蓝图并同时允许来自 Git 的后续同步，您必须在完成最终更改之后创建新版本。

设置要与 GitLab 一起使用的蓝图并收集所需信息后，您必须设置与 GitLab 实例的集成。然后，您可以将指定的蓝图导入 GitLab。您可以在 <https://www.youtube.com/watch?v=h0vqp63Sdgg> 中查看此过程的视频演示。

前提条件

- 为适用的存储库提取 API 密钥。在 GitLab 帐户中，选择右上角的登录名，然后导航到“设置”菜单。选择“访问令牌”，并命名您的令牌，设置到期日期。然后，选择 API 并创建令牌。复制生成的值并保存它。

您必须具有相应的本地 Git 存储库并为指定的所有用户配置了访问权限，以便设置 Git 与 vRealize Automation Cloud Assembly 的集成。此外，您还必须在特定结构中创建并保存蓝图，以便 GitLab 检测到它们。

- 配置和存储蓝图，以便正确地与 GitLab 集成。只有有效蓝图才会导入到 GitLab。请参见[如何在 vRealize Automation Cloud Assembly 中使用 GitLab 和 GitHub 集成](#)。

步骤

- 1 在 vRealize Automation Cloud Assembly 中设置与 GitLab 环境的集成。
 - a 选择**基础架构 > 集成 > 新增**，然后选择 GitLab。
 - b 输入您的 GitLab 实例的 **URL**。对于软件即服务 GitLab 实例，大多数情况下，它是 `gitlab.com`。
 - c 输入指定 GitLab 实例的**令牌**（也称为 API 密钥）。有关从 GitLab 实例中提取令牌的信息，请参见上述必备条件。
 - d 添加相应的名称和说明。
 - e 单击**验证**以验证连接。
 - f 如果需要，添加功能标记。有关详细信息，请参见在 [vRealize Automation Cloud Assembly 中使用功能标记](#)。
 - g 单击**添加**。
- 2 配置 GitLab 连接，以便接受相应存储库中的蓝图。
 - a 选择**基础架构集成**，然后选择相应的 GitLab 集成。
 - b 选择**项目**。
 - c 选择**新建项目**，并为该项目创建一个名称。
 - d 在 GitLab 中输入**存储库**路径。通常情况下，这是附加到存储库名称的主帐户的用户名。
 - e 输入要使用的相应 GitLab **分支**。
 - f 如果适用，请输入**文件夹**名称。如果留空，则所有文件夹都可用。
 - g 输入相应的**类型**。如果适用，请输入文件夹名称。如果留空，则所有文件夹都可用。
 - h 单击**下一步**完成存储库的添加。
单击**下一步**时，会启动自动同步任务，可将蓝图导入到平台中。
同步任务完成后，会显示一条消息，指示已导入蓝图。

结果

现在，可以从 GitLab 检索蓝图。

在 vRealize Automation Cloud Assembly 中配置 GitHub 集成

可以在 vRealize Automation Cloud Assembly 中集成 GitHub 云端存储库托管服务

您需要有效的 GitHub 令牌才能在 vRealize Automation Cloud Assembly 中配置 GitHub 集成。有关创建和查找令牌的信息，请参见 [GitHub 文档](#)。

前提条件

- 您必须能够访问 GitHub。
- 配置和存储蓝图，以便正确地与 GitHub 集成。只有有效蓝图才会导入到 GitHub。请参见[如何在 vRealize Automation Cloud Assembly 中使用 GitLab 和 GitHub 集成](#)。

步骤

- 1 选择**基础架构 > 连接 > 集成**，然后单击**添加集成**。
- 2 选择“GitHub”。
- 3 在 GitHub 配置页面上输入必填信息。
- 4 单击**验证**以检查集成。
- 5 如果需要添加标记以支持标记策略，请输入功能标记。请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署和创建标记策略](#)。
- 6 单击**添加**。

结果

GitHub 可在 vRealize Automation Cloud Assembly 蓝图中使用。

后续步骤

现在，可以从 GitHub 检索蓝图。

在 vRealize Automation 中配置外部 IPAM 集成点

可以创建提供程序特定的外部 IPAM 集成点，以管理蓝图部署中使用的 IP 地址。使用外部 IPAM 集成点时，将从指定的 IPAM 提供程序（而非 vRealize Automation）获取 IP 地址并由其管理。

可以在 vRealize Automation 中创建提供程序特定的 IPAM 集成点，以管理蓝图部署和虚拟机的 IP 地址和 DNS 设置。

有关如何配置必备条件的信息，以及如何在示例工作流的上下文中创建提供程序特定的外部 IPAM 集成点的示例，请参见在[vRealize Automation 中添加外部 IPAM 集成点](#)。

有关如何创建所需资产以使外部 IPAM 合作伙伴和供应商能够将其 IPAM 解决方案与 vRealize Automation 集成的信息，请参见[如何使用 IPAM SDK 为 vRealize Automation 创建提供程序特定的外部 IPAM 集成软件包](#)。

前提条件

- 确认您具有云管理员凭据。请参见在[vRealize Automation 中使用云帐户所需的凭据](#)。

- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有外部 IPAM 提供程序（例如 Infoblox 或 Bluecat）的帐户，并且具有使用 IPAM 提供程序访问组织帐户的正确访问凭据。
- 确认您有权访问 IPAM 提供程序（例如 Infoblox 或 BlueCat）的已部署集成软件包。部署的软件包最初从 IPAM 提供程序或 vRealize Automation 商城以 .zip 形式下载，然后部署到 vRealize Automation。
- 确认您有权访问为 IPAM 提供程序配置的运行环境。
- 如果使用的是基于操作的可扩展性 (ABX) 内部部署嵌入式运行环境，请确认 vRealize Automation 网络中具有能够将出站流量传递到外部站点（如 gcr.io 和 storage.googleapis.com）的 HTTP 代理服务器。有关详细信息，请参见在 [vRealize Automation 8.x 中通过代理提取 Docker 映像 \(75180\)](#)。

步骤

- 1 选择**基础架构 > 连接 > 集成**，然后单击**添加集成**。

- 2 单击 **IPAM**。

- 3 在**提供程序**下拉列表中，选择已配置的 IPAM 提供程序软件包。

如果此列表为空，请单击**导入提供程序软件包**，导航到现有提供程序软件包 .zip 文件，然后选择该文件。如果没有此 .zip 文件，则可以从提供程序的网站或从 vRealize Automation 的**商城**选项卡获取该文件。

- 4 输入具有外部 IPAM 提供程序的帐户的管理员用户名和密码凭据以及所有其他（如果有）必填字段，如提供程序的主机名。

- 5 在**运行环境**下拉列表选择一个现有的运行环境，例如基于操作的可扩展性内部部署集成点。

运行环境支持 vRealize Automation 和 IPAM 提供程序之间的通信。

IPAM 框架仅支持基于操作的可扩展性 (ABX) 内部部署嵌入式运行环境。

注 如果使用 Amazon Web Services 或 Microsoft Azure 云帐户作为集成运行环境，请确保 IPAM 提供程序设备符合以下条件：可以通过 Internet 进行访问，不位于 NAT 或防火墙后面，并且具有可公开解析的 DNS 名称。如果 IPAM 提供程序不可访问，则 Amazon Web Services Lambda 或 Microsoft Azure 函数无法与其相连接，集成将失败。

- 6 单击**验证**。

- 7 当系统提示您信任来自外部 IPAM 提供程序的自签名证书时，单击**接受**。

接受自签名证书后，可以继续完成验证操作。

- 8 为此 IPAM 集成点输入一个名称，然后单击**添加**以保存新的 IPAM 集成点。

将模拟数据收集操作。将从外部 IPAM 提供程序收集网络和 IP 地址数据。

如何在 vRealize Automation 中升级到较新的 IPAM 集成软件包

您可以升级现有的外部 IPAM 集成点升级，以获取供应商特定的 IPAM 集成软件包的较新版本。

外部 IPAM 提供程序或 VMware 可以升级特定供应商的源 IPAM 集成软件包。例如，Infoblox 的外部 IPAM 集成软件包已多次升级。要保留使用命名的 IPAM 集成点的任何现有 vRealize Automation 基础架构设置，您可以编辑 IPAM 集成点以获取更新的 IPAM 集成软件包，而不是创建新的 IPAM 集成点。

前提条件

此过程假定您已创建外部 IPAM 集成点，并希望升级该集成点，以使用供应商的 IPAM 集成软件包的较新版本。

有关如何创建外部 IPAM 集成点的信息，请参见在 [vRealize Automation 中添加外部 IPAM 集成点](#)。

- 确认您具有云管理员凭据。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有外部 IPAM 提供程序的帐户，并且具有使用 IPAM 提供程序访问组织帐户的正确访问凭据。
- 确认您有权访问 IPAM 提供程序的已部署集成软件包。部署的软件包最初从 IPAM 提供程序网站或 vRealize Automation 商城以 .zip 形式下载，然后部署到 vRealize Automation。

有关如何下载和部署提供程序软件包 .zip 文件并使其在“IPAM 集成”页面上可作为提供程序值提供的信息，请参见 [下载并部署外部 IPAM 提供程序软件包以在 vRealize Automation Cloud Assembly 中使用](#)。

- 确认您有权访问为 IPAM 提供程序配置的运行环境。运行环境通常是一个基于操作的可扩展性 (ABX) 内部部署嵌入式集成点。

有关运行环境特性的信息，请参见在 [vRealize Automation 中为 IPAM 集成点创建运行环境](#)。

步骤

- 1 选择**基础架构 > 连接 > 集成 IPAM**，然后打开现有的 IPAM 集成点。
- 2 单击**管理提供程序**。
- 3 导航到更新的 IPAM 集成软件包并将其导入。
- 4 单击**验证**，然后单击**保存**。

在 vRealize Automation Cloud Assembly 中配置 My VMware 集成

您可以将 My VMware 与 vRealize Automation Cloud Assembly 集成，以支持 VMware 相关的操作和功能，例如访问 VMware Marketplace 以获取蓝图。

每个组织只能创建一个 My VMware 集成。

前提条件

您必须拥有对 My VMware 具有相应权限的用户帐户。

- 有关邀请用户加入 My VMware 帐户的信息，请参见[知识库文章 2070555](#)。

- 有关在 My VMware 帐户中分配用户权限的信息，请参见[知识库文章 2006977](#)。

步骤

- 1 选择**基础架构 > 连接 > 集成**，然后单击**添加集成**。
- 2 选择 My VMware。
- 3 在 My VMware 配置页面上输入所需的信息。
- 4 如果需要标记以支持标记策略，请输入功能标记。请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署和创建标记策略](#)。
- 5 单击**添加**。

结果

My VMware 可与蓝图配合使用。

后续步骤

将 My VMware 组件添加到所需的蓝图。

在 Cloud Assembly 中配置 vRealize Orchestrator 集成

您可以配置一个或多个 vRealize Orchestrator 集成，以便可以使用工作流作为可扩展性的一部分。

vRealize Automation 包含可用于可扩展性订阅的预配置 vRealize Orchestrator 实例。还可以从 vRealize Automation 云服务控制台访问嵌入式 vRealize Orchestrator 的客户端。

通过将 vRealize Orchestrator 集成到 vRealize Automation Cloud Assembly，您可以添加外部 vRealize Orchestrator 实例，并使用可扩展性订阅中包含的工作流库。有关详细信息，请参见[可扩展性工作流订阅](#)。

前提条件

- 确认您具有云管理员凭据。有关详细信息，请参见[vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 将 vRealize Orchestrator 7.x 实例迁移到版本 8.0。请参见《安装、配置和迁移 VMware vRealize Orchestrator》中的将独立 vRealize Orchestrator 迁移到外部 vRealize Orchestrator 8.0。

步骤

- 1 选择**基础架构 > 连接 > 集成**。
- 2 单击**添加集成**。
- 3 选择 vRealize Orchestrator。
- 4 在 vRealize Automation Cloud Assembly 中，输入 vRealize Orchestrator 实例的 URL。
- 5 要验证集成，请单击**验证**。
- 6 输入 vRealize Orchestrator 集成的名称。
- 7 （可选）输入 vRealize Orchestrator 集成的描述。

- 8 （可选）添加功能标记。有关功能标记的详细信息，请参见在 [vRealize Automation Cloud Assembly 中使用功能标记](#)。

注 可使用功能标记管理多个 vRealize Orchestrator 集成。请参见[使用项目限制管理多个 vRealize Orchestrator 集成](#)。

- 9 单击**添加**。

将保存 vRealize Orchestrator 集成。

后续步骤

要验证集成是否已配置且工作流是否已添加，请选择**可扩展性 > 库 > 工作流**。

使用项目限制管理多个 vRealize Orchestrator 集成

您可以使用项目限制来管理在工作流订阅中使用的 vRealize Orchestrator 集成。

vRealize Automation Cloud Assembly 支持集成多个可在工作流订阅中使用的 vRealize Orchestrator 服务器。可以使用软或硬项目限制管理由项目置备的蓝图中使用 vRealize Orchestrator 集成。有关项目限制的详细信息，请参见[使用 vRealize Automation Cloud Assembly 项目标记和自定义属性](#)。

前提条件

- 确认您具有云管理员凭据。有关详细信息，请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 在 vRealize Automation Cloud Assembly 中配置两个或更多 vRealize Orchestrator 集成。有关详细信息，请参见在 [Cloud Assembly 中配置 vRealize Orchestrator 集成](#)。
- 将功能标记添加到您的 vRealize Orchestrator 集成。有关功能标记的详细信息，请参见在 [vRealize Automation Cloud Assembly 中使用功能标记](#)。

步骤

- 1 导航到**基础架构 > 配置 > 项目**，然后选择您的项目。
- 2 选择**置备**选项卡。
- 3 在**可扩展性限制**文本框中输入 vRealize Orchestrator 集成的功能标记，然后将其设置为软或硬项目限制。
- 4 单击**保存**。

结果

当部署蓝图时，vRealize Automation Cloud Assembly 使用项目限制来管理在工作流订阅中使用的 vRealize Orchestrator 集成。

后续步骤

或者，也可以使用功能标记在云帐户级别管理多个 vRealize Orchestrator 集成。有关详细信息，请参见[使用云帐户功能标记管理多个 vRealize Orchestrator 集成](#)。

使用云帐户功能标记管理多个 vRealize Orchestrator 集成

您可以使用功能标记来管理在工作流订阅中使用的 vRealize Orchestrator 集成。

vRealize Automation Cloud Assembly 支持集成多个可在工作流订阅中使用的 vRealize Orchestrator 服务器。您可以通过向您的云帐户添加功能标记来管理在工作流订阅中使用的 vRealize Orchestrator 集成。

前提条件

- 确认您具有云管理员凭据。有关详细信息，请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 在 vRealize Automation Cloud Assembly 中配置两个或更多 vRealize Orchestrator 集成。有关详细信息，请参见在 [Cloud Assembly 中配置 vRealize Orchestrator 集成](#)。
- 将功能标记添加到您的 vRealize Orchestrator 集成。有关功能标记的详细信息，请参见在 [vRealize Automation Cloud Assembly 中使用功能标记](#)。

步骤

- 1 导航到**基础架构 > 连接 > 云帐户**。
- 2 选择您的云帐户。
- 3 输入要使用的 vRealize Orchestrator 集成的功能标记。

功能标记将自动转换为软限制。要在管理集成时使用硬限制，您必须使用项目限制。有关详细信息，请参见[使用项目限制管理多个 vRealize Orchestrator 集成](#)。

- 4 单击**保存**。

结果

当部署蓝图时，vRealize Automation Cloud Assembly 使用关联云帐户中的标记来管理在工作流订阅中使用的 vRealize Orchestrator 集成。

如何在 vRealize Automation Cloud Assembly 中使用 Kubernetes

可以将 Pivotal Container Service (PKS) 或 Red Hat OpenShift 与 vRealize Automation Cloud Assembly 相集成，以管理和部署 Kubernetes 资源。还可以在 vRealize Automation Cloud Assembly 中集成外部 Kubernetes 资源。

创建 PKS 或 OpenShift 集成后，适用的 Kubernetes 集群将在 vRealize Automation Cloud Assembly 中变得可用，您可以在 vRealize Automation Cloud Assembly 中添加和创建 Kubernetes 组件以支持集群和容器应用程序的管理。这些应用程序构成了可从 Service Broker 目录中获得的自助服务部署的基础。

- 在 [vRealize Automation Cloud Assembly 中配置 PKS 集成](#)

可以在内部部署和云中配置 PKS 资源连接，以在 vRealize Automation Cloud Assembly 中支持 Kubernetes 集成和管理功能。

- 在 [vRealize Automation Cloud Assembly 中使用 Kubernetes 群集和命名空间](#)

您可以在 vRealize Automation Cloud Assembly 中添加、查看和管理 Kubernetes 群集和命名空间（作为 Kubernetes 部署的基础）的配置。

- 在 [vRealize Automation Cloud Assembly](#) 中配置 [Kubernetes 区域](#)

通过 [Kubernetes 区域](#)，云管理员可以定义如何基于策略布置在 [vRealize Automation Cloud Assembly](#) 部署中使用的 [Kubernetes 集群](#)和命名空间。管理员可以使用此页面指定哪些集群可用于置备 [Kubernetes 命名空间](#)，以及可接受哪些属性用于集群。

- 在 [vRealize Automation Cloud Assembly](#) 中将 [Kubernetes 组件](#)添加到蓝图

将 [Kubernetes 组件](#)添加到 [vRealize Automation Cloud Assembly](#) 蓝图时，可以选择添加群集或允许用户在各种配置中创建命名空间。通常，此选择取决于您的访问控制要求、[Kubernetes 组件](#)的配置方式以及您的部署要求。

- 配合使用 [vRealize Automation Cloud Assembly](#) 可扩展性与 [Kubernetes](#)

[vRealize Automation Cloud Assembly](#) 提供一组标准事件主题，它们对应于与 [Kubernetes 集群部署](#)相关的典型操作。用户可以根据需要订阅这些主题，他们会在发生与已订阅主题相关的事件时收到通知。您还可以将 [vRO 工作流](#)配置为基于事件通知运行。

在 [vRealize Automation Cloud Assembly](#) 中配置 [PKS 集成](#)

可以在内部部署和云中配置 [PKS 资源连接](#)，以在 [vRealize Automation Cloud Assembly](#) 中支持 [Kubernetes 集成](#)和管理功能。

通过 [PKS 集成](#)，您可以管理内部部署和云中的 [PKS 实例](#)以及在 [PKS](#) 上置备的 [Kubernetes 群集](#)和外部群集。您必须创建 [Kubernetes 配置文件](#)并将其与项目相关联，以支持基于策略的资源放置。

前提条件

- 您必须使用 [UAA 身份验证](#)设置适当配置的 [Pivotal Container Service \(PKS\)](#) 服务器。
- 确认您具有云管理员凭据。有关详细信息，请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。

步骤

- 1 选择[基础架构 > 连接 > 集成](#)，然后单击[添加集成](#)。
- 2 选择“[VMware Enterprise PKS](#)”。
- 3 输入要创建的 [PKS 云帐户](#)的 IP 地址或 FQDN 以及 [PKS 地址](#)。
 - IP 地址是 [PKS 用户身份验证服务器](#)的 FQDN 或 IP 地址。
 - [PKS 地址](#)是主 [PKS 服务器](#)的 FQDN 或 IP 地址。
- 4 选择此 [PKS 服务器](#)是位于本地，还是位于公有云或私有云上。
- 5 输入 [PKS 服务器](#)的相应[用户名](#)和[密码](#)及其他相关信息。
- 6 如果使用标记以支持标记策略，请输入功能标记。请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署和创建标记策略](#)。
- 7 单击[添加](#)。

结果

您可以创建 Kubernetes 区域并将其分配给项目，也可以发现外部 Kubernetes 群集并将这些群集分配给项目。此外，还可以添加或创建 Kubernetes 命名空间，以便于管理大型组和组织中的群集。

后续步骤

创建或选择相应的 Kubernetes 区域，然后选择一个或多个群集或命名空间，并将其分配给项目。之后，您可以创建并发布蓝图，以使用户能够生成使用 Kubernetes 的自助服务部署。

在 vRealize Automation Cloud Assembly 中使用 Kubernetes 群集和命名空间

您可以在 vRealize Automation Cloud Assembly 中添加、查看和管理 Kubernetes 群集和命名空间（作为 Kubernetes 部署的基础）的配置。

您可以在**基础架构 > 资源 > Kubernetes** 页面上查看、添加和管理您有权访问的 Kubernetes 群集和命名空间。最常见的情况是，此页面可帮助管理部署的群集和命名空间。

- **群集：**群集是分布在一个或多个物理机上的一组 Kubernetes 节点。此页面显示已配置为可在 vRealize Automation Cloud Assembly 实例上使用的已置备但未部署的群集。您可以单击群集查看有关其当前状态的信息。部署群集时，包含指向 Kubconfig 文件的链接，此文件只能由云管理员进行访问。此文件授予对群集的完全管理员特权，包括命名空间列表。
- **命名空间：**命名空间是虚拟群集，它为管理员提供了一种分隔群集资源的方法。它们有助于在大型用户组和组织之间管理资源。作为基于角色进行访问控制的一种形式，云管理员可以让用户在请求部署时将命名空间添加到项目，然后从“Kubernetes 群集”页面管理这些命名空间。部署命名空间时，包含指向 Kubconfig 文件的链接，该文件让有效用户（如开发人员）能够查看和管理该命名空间的某些方面。

如果要配置新群集或现有群集，必须选择是使用主 IP 地址还是主节点主机名进行连接。

在 vRealize Automation Cloud Assembly 中使用 Kubernetes 群集

您可以使用此页面上的选项向 vRealize Automation Cloud Assembly 添加新群集、现有群集或外部群集。

- 1 选择**基础架构 > 资源 > Kubernetes**，并确认“群集”选项卡处于活动状态。

如果当前已为您的 vRealize Automation Cloud Assembly 实例配置任何群集，这些群集将显示在此页面上。

- 2 如果要添加新群集或现有群集，或者部署群集，请根据下表选择适当的选项。

| 选项 | 说明 | 详细信息 |
|-------|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| 部署 | 向 vRealize Automation Cloud Assembly 添加新群集 | 您必须指定要将此群集部署到的 PKS 云帐户以及所需计划和节点数。 |
| 添加现有项 | 配置现有群集以使用您的项目。 | 您必须指定 PKS 云帐户、要使用的群集以及适用于目标开发人员的相应项目。此外，您还需要指定共享范围。如果要全局共享，必须相应地配置 Kubernetes 区域和命名空间。 |
| 添加外部项 | 将可能与 PKS 没有关联的 vanilla Kubernetes 群集添加到 vRealize Automation Cloud Assembly。 | 您必须指定与群集关联的项目，输入所需群集的 IP 地址，然后选择连接到此群集所需的云代理和证书信息。 |

- 3 单击**添加**以使群集可在 vRealize Automation Cloud Assembly 中使用。

在 vRealize Automation Cloud Assembly 中使用 Kubernetes 命名空间

如果您是云管理员，则命名空间可帮助您对 Kubernetes 群集资源进行分组和管理。如果您是用户，则命名空间是 Kubernetes 群集内供您进行部署的区域。管理员和用户可以使用位于**基础架构 > 资源 > Kubernetes** 页面上的“命名空间”选项卡访问命名空间。

可通过多种方式将 Kubernetes 命名空间添加到 vRealize Automation Cloud Assembly 中的资源。以下过程概述了一种典型方法。

- 1 选择**基础架构 > 资源 > Kubernetes**，然后单击“命名空间”选项卡。
- 2 要添加新命名空间，请单击**新建命名空间**。要添加现有命名空间，请单击**添加命名空间**。
- 3 输入命名空间的**名称**和**描述**。

此时，您已添加一个可用于 Kubernetes 资源的命名空间，但此命名空间尚未与任何特定内容相关联。

- 4 指定要与此命名空间关联的**群集**。
- 5 单击**创建**将此命名空间添加到 vRealize Automation Cloud Assembly。

在 vRealize Automation Cloud Assembly 中配置 Kubernetes 区域

通过 Kubernetes 区域，云管理员可以定义如何基于策略布置在 vRealize Automation Cloud Assembly 部署中使用的 Kubernetes 集群和命名空间。管理员可以使用此页面指定哪些集群可用于置备 Kubernetes 命名空间，以及可接受哪些属性用于集群。

云管理员可以将 Kubernetes 区域与为 Cloud Assembly 配置的 PKS 云帐户关联，或者与尚未和项目关联的外部 Kubernetes 集群关联。

创建 Kubernetes 区域时，您可以为该区域分配提供程序特定的多个资源，这些资源将指示可以根据工作线程数、主节点、可用 CPU 数、内存和其他配置设置为新置备的集群设置哪些属性。对于 PKS 提供程序，这些提供程序与 PKS 计划相对应。管理员还可以向将用于布置新置备的 Kubernetes 命名空间的 Kubernetes 区域分配多个集群。管理员只能分配未载入或不受 CMX 管理，并通过预选集群提供程序进行置备的集群。管理员可以向单个项目分配多个 Kubernetes 区域，从而使它们全部可用于在此项目中发生的布置操作。

云管理员可以分多个级别分配优先级。

- 项目内的 Kubernetes 区域优先级。
- Kubernetes 区域内的资源优先级。
- Kubernetes 区域内的集群优先级。

云管理员还可以分多个级别分配标记：

- 每个 Kubernetes 区域的功能标记。
- 每个资源分配的标记。
- 每个集群分配的标记。

Service Broker 包含一个版本的 Kubernetes 区域页面，使 Service Broker 管理员能够访问现有的 Kubernetes 区域，以便他们可以为 Kubernetes 命名空间和从目录置备的集群创建放置策略。

前提条件

配置与合适 PKS 部署的集成。请参见在 [vRealize Automation Cloud Assembly](#) 中配置 PKS 集成

步骤

- 1 选择**基础架构 > 配置 > Kubernetes 区域**，然后单击**新建 Kubernetes 区域**。
- 2 输入要将此区域应用到的 PKS 集成**帐户**名称。
- 3 为 Kubernetes 区域添加**名称**和**描述**。
- 4 如果适用，添加功能标记。有关详细信息，请参见在 [vRealize Automation Cloud Assembly](#) 中使用**功能标记**。
- 5 单击**保存**。
- 6 单击“按需”选项卡，然后根据需要为要用于集群置备的区域添加 PKS 计划。

您可以选择一个或多个计划，并为其分配优先级。数字越小，优先级越高。优先级分配次于基于标记的选择。

- 7 单击“集群”选项卡，然后单击**添加**按钮将 Kubernetes 集群添加到区域。如果使用的是外部集群，则在选择时会将其自动载入到 vRealize Automation Cloud Assembly。

您可以在 vRealize Automation Cloud Assembly 中的“Kubernetes 集群”页面上向集群添加 Kubernetes 命名空间。

结果

Kubernetes 区域已配置，可与 vRealize Automation Cloud Assembly 部署配合使用。

后续步骤

向项目分配 Kubernetes 区域。

- 1 选择**基础架构 > 配置 > 项目**，然后选择要与 Kubernetes 区域关联的项目。
- 2 单击“项目”页面上的“Kubernetes 置备”选项卡。

- 3 单击**添加 Kubernetes 区域**并添加刚刚创建的区域。如果适用，您可以设置多个区域，还可以对区域设置优先级。
- 4 单击**保存**。

将区域分配给项目后，您可以使用“蓝图”页面，根据 Kubernetes 区域和项目配置来置备部署。

在 vRealize Automation Cloud Assembly 中将 Kubernetes 组件添加到蓝图

将 Kubernetes 组件添加到 vRealize Automation Cloud Assembly 蓝图时，可以选择添加群集或允许用户在各种配置中创建命名空间。通常，此选择取决于您的访问控制要求、Kubernetes 组件的配置方式以及您的部署要求。

要在 vRealize Automation Cloud Assembly 中将 Kubernetes 组件添加到蓝图，请单击“蓝图”，选择**新建**，然后在左侧菜单中找到并展开 Kubernetes 选项。然后，通过将所需选项（群集或 KBS 命名空间）拖动到画布做出选择。

将与项目关联的 Kubernetes 群集添加到蓝图是使 Kubernetes 资源可供有效用户使用的最简单方法。您可以在群集上使用标记来控制其部署位置，就像处理其他 Cloud Assembly 资源一样。在群集部署的分配阶段，您可以使用标记选择区域和 PKS 计划。

通过这种方式添加群集后，该群集将自动可供所有有效用户使用。

蓝图示例

第一个蓝图示例显示了通过标记控制的简单 Kubernetes 部署的蓝图。创建的 Kubernetes 区域包含两个部署计划，并在“新建 Kubernetes 区域”页面上进行配置。在此示例中，一个名为 `placement:tag` 的标记添加为该区域中的一个功能，并用于匹配蓝图上的类似限制。如果有多个区域配置了此标记，将选择具有最低优先级编号的区域。

```
formatVersion: 1
inputs: {}
resources:
  Cluster_provisioned_from_tag:
    type: Cloud.K8S.Cluster
    properties:
      hostname: 109.129.209.125
      constraints:
        -tag: 'placement tag'
        port: 7003
        workers: 1
        connectBy: hostname
```

第二个蓝图示例显示了如何使用名为 `$(input.hostname)` 的变量设置蓝图，以便用户可以在请求部署时输入所需的群集主机名。在群集部署的资源分配阶段，也可以使用标记选择区域和 PKS 计划。

```
formatVersion: 1
inputs:
  hostname:
    type: string
    title: Cluster hostname
resources:
  Cloud_K8S_Cluster_1:
```

```

type: Cloud.K8S.Cluster
properties:
  hostname: ${input.hostname}
  port: 8443
  connectBy: hostname
  workers: 1

```

如果要使用命名空间管理群集使用情况，可以在蓝图中设置一个名为 *name: \${input.name}* 的变量来替代用户在请求部署时输入的命名空间名称。对于此类部署，您需要创建一个如以下示例所示的蓝图：

```

1 formatVersion: 1
2 inputs:
3 name:
4   type: string
5   title: "Namespace name"
6 resources:
7   Cloud_K8S_Namespace_1:
8     type: Cloud.K8S.Namespace
9     properties:
10      name: ${input.name}

```

用户可以通过 kubeconfig 文件管理部署的群集，这些文件可从[基础架构 > 资源 > Kubernetes 群集](#)页面访问。在页面上找到所需群集对应的卡视图，然后单击 **Kubeconfig**。

配合使用 vRealize Automation Cloud Assembly 可扩展性与 Kubernetes

vRealize Automation Cloud Assembly 提供一组标准事件主题，它们对应于与 Kubernetes 集群部署相关的典型操作。用户可以根据需要订阅这些主题，他们会在发生与已订阅主题相关的事件时收到通知。您还可以将 vRO 工作流配置为基于事件通知运行。

以下主题可在 vRealize Automation Cloud Assembly 中的[可扩展性 > 库 > 事件主题](#)页面上进行订阅。要查看这些主题，请在“事件主题搜索”文本框中搜索 Kubernetes。

- Kubernetes 集群分配
- Kubernetes 集群置备后
- Kubernetes 集群移除后
- Kubernetes 集群置备
- Kubernetes 集群移除

单击这些主题之一以查看该主题的结构定义，其中显示了收集和传输的所有信息。您可以使用任何结构定义信息来设置各种通知和管理任务及报告任务。

您可以在[可扩展性 > 库 > 操作](#)页面上设置 CMX 相关操作的操作脚本。操作脚本可用于各种用途：例如创建 Kubernetes 集群置备的 DNS 记录。如果要创建 DNS 记录，可以使用操作脚本中具有 REST 命令的 Kubernetes 集群置备后主题中的 `masternodeips` 字段来创建 DNS 记录。

“订阅”页面定义事件主题和操作脚本之间的关系。您可以在 vRealize Automation Cloud Assembly 中的“订阅”页面上查看和管理这些组件。

什么是 vRealize Automation Cloud Assembly 中的配置管理

vRealize Automation Cloud Assembly 支持与 Puppet Enterprise 和 Ansible 开源集成，以便您可以对配置和偏差管理部署。

Puppet 集成

要集成基于 Puppet 的配置管理，必须在具有 vSphere 工作负载的公共或私有云上安装一个有效的 Puppet Enterprise 实例。必须在此外部系统与 vRealize Automation Cloud Assembly 实例之间建立连接。然后，可以通过将 Puppet 配置管理添加到相应的蓝图，使其供 vRealize Automation Cloud Assembly 使用。

vRealize Automation Cloud Assembly 蓝图服务 Puppet 提供程序在已部署的计算资源上安装、配置和运行 Puppet 代理。Puppet 提供程序同时支持 SSH 和 WinRM 连接，并具备以下必备条件：

- SSH 连接：
 - 要运行具有 NOPASSWD 的命令，用户名必须是超级用户或具有 sudo 权限的用户。
 - 针对给定用户禁用 requiretty。
 - cURL 必须在部署计算资源上可用。
- WinRM 连接：
 - PowerShell 2.0 必须在部署计算资源上可用。
 - 按照 vRealize Orchestrator 文档中的说明配置 Windows 模板。

DevOps 管理员负责管理与 Puppet Master 的连接，并将 Puppet 角色或配置规则应用于特定部署。执行以下部署后，配置为支持配置管理的虚拟机将注册到指定的 Puppet Master。

部署虚拟机时，用户可以添加或删除作为外部系统的 Puppet Master 或更新分配给 Puppet Master 的项目。最后，当虚拟机取消配置时，相应的用户可以从 Puppet Master 取消注册已部署的虚拟机。

Ansible 开源集成

设置 Ansible 集成时，请按照 Ansible 安装说明安装 Ansible 开源。有关安装的更多信息，请参见 Ansible 文档。

默认情况下，Ansible 启用主机密钥检查。如果重新安装 known_hosts 文件中的主机时使用不同的密钥，则会出现错误消息。如果主机未列在 known_hosts 文件中，则您必须在启动时提供密钥。您可以通过 /etc/ansible/ansible.cfg 或 ~/.ansible.cfg 文件中的以下设置禁用主机密钥检查：

```
[defaults]
host_key_checking = False
localhost_warning = False

[paramiko_connection]
record_host_keys = False

[ssh_connection]
#ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
ssh_args = -o UserKnownHostsFile=/dev/null
```

要避免主机密钥检查错误，请将 `host_key_checking` 和 `record_host_keys` 设置为 `False`，包括添加在 `ssh_args` 中设置的额外选项 `UserKnownHostsFile=/dev/null`。此外，如果清单最初为空，Ansible 会警告主机列表为空。这会导致 Playbook 语法检查失败。

通过 Ansible 保管库，可以加密文件而非纯文本形式存储敏感信息（如密码或密钥）。保管库使用密码进行加密。在 vRealize Automation Cloud Assembly 中，Ansible 使用保管库对主机的 ssh 密码等数据进行加密。它会假设已设置保管库密码的路径。

可以修改 `ansible.cfg` 文件，使用以下格式指定密码文件的位置。

```
vault_password_file = /path to/file.txt
```

还可以设置 `ANSIBLE_VAULT_PASSWORD_FILE` 环境变量，以便 Ansible 自动搜索密码。例如，
`ANSIBLE_VAULT_PASSWORD_FILE=~/.vault_pass.txt`

vRealize Automation Cloud Assembly 管理 Ansible 清单文件，因此必须确保 vRealize Automation Cloud Assembly 用户对清单文件具有 `rwX` 访问权限。

```
cat ~/var/tmp/vmware/provider/user_defined_script/$(ls -t ~/var/tmp/vmware/provider/
user_defined_script/ | head -1)/log.txt
```

如果要使用具有 vRealize Automation Cloud Assembly 开源集成的非 root 用户，则用户需要一组权限来运行 vRealize Automation Cloud Assembly 开源提供程序所使用的命令。必须在用户的 `sudoers` 文件中设置以下命令。

```
Defaults:myuser !requiretty
```

如果用户不属于未指定 `askpass` 应用程序的管理员组，请在用户的 `sudoers` 文件中设置以下命令。

```
myuser ALL=(ALL) NOPASSWD: ALL
```

如果在设置 Ansible 集成时遇到错误或其他问题，请参阅位于 Ansible 控制计算机上的 `'cat ~/var/tmp/vmware/provider/user_defined_script/$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ | head -1)'` 中的 `log.txt` 文件。

在 vRealize Automation Cloud Assembly 中配置 Puppet Enterprise 集成

vRealize Automation Cloud Assembly 支持与 Puppet Enterprise 集成配置管理集成。

将 Puppet Enterprise 作为外部系统添加到 Cloud Assembly 后，默认情况下可在所有项目中使用。可以将其限制到特定项目。

要添加 Puppet Enterprise 集成，您必须具有 Puppet 主节点名称以及主节点的主机名或 IP 地址。

您可以在以下位置找到 Puppet 日志，以防需要检查它们是否存在错误或出于信息目的。

| 说明 | 日志位置 |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 面向创建和安装相关事件的日志 | 这些日志位于已部署计算机上的 <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ head -1)/`。</code> 有关完整日志，请参考 log.txt 文件。有关详细的 Puppet 代理日志，请参阅 https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging |
| 面向 Puppet 删除和运行相关任务的日志 | 这些日志位于 PE 上的 <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ head -1)/`。</code> 有关完整日志，请参考 log.txt 文件。 |

步骤

- 1 选择**基础架构 > 连接 > 集成**，然后单击**添加集成**。
- 2 选择 Puppet。
- 3 在 Puppet 配置页面上输入所需的信息。
- 4 单击**验证**以检查集成。
- 5 单击**添加**。

结果

Puppet 可与蓝图配合使用。

后续步骤

将 Puppet 组件添加到所需的蓝图。

- 1 在蓝图菜单的“内容管理”标题下选择 Puppet，然后将 Puppet 组件拖动到画布上。
- 2 在右侧的窗格中输入“Puppet 属性”。

| 属性 | 说明 |
|--------|------------------------------------------------------------------|
| Master | 输入要与此蓝图一起使用的 Puppet 主计算机的名称。 |
| 环境 | 为 Puppet 主计算机选择环境。 |
| 角色 | 选择要与此蓝图一起使用的 Puppet 角色。 |
| 代理运行间隔 | 希望 Puppet 代理轮询 Puppet 主计算机以获取配置详细信息的频率，该配置详细信息将应用于与此蓝图相关的已部署虚拟机。 |

- 3 单击右侧窗格中的“代码”选项卡以查看 Puppet 配置属性的 YAML 代码。

在 vRealize Automation Cloud Assembly 中配置 Ansible 开源集成

vRealize Automation Cloud Assembly 支持与 Ansible 开源配置管理集成。配置集成后，可以将 Ansible 组件添加到新部署或现有部署中。

将 Ansible 开源与 vRealize Automation Cloud Assembly 集成时，可以将其配置为置备新计算机时按给定顺序运行一个或多个 Ansible playbook，以自动执行配置管理。可以在蓝图中为部署指定所需的 playbook。

设置 Ansible 集成时，必须指定 Ansible 开源主机，以及可为管理资源定义信息的清单文件路径。此外，还必须提供用于访问 Ansible 开源实例的名称和密码。稍后，在将 Ansible 组件添加到部署时，可以更新连接以使用基于密钥的身份验证。

默认情况下，Ansible 使用 ssh 连接到物理计算机。如果使用的是在蓝图中通过 osType Windows 属性指定的 Windows 计算机，则 connection_type 变量将自动设置为 winrm。

Ansible 集成支持不使用 IP 地址的物理计算机。对于在公有云（如 AWS、Azure 和 GCP）上置备的计算机，仅当计算机连接到公共网络时，才会在所创建资源的 address 属性中填充计算机的公共 IP 地址。对于未连接到公共网络的计算机，Ansible 集成将从连接到该计算机的网络查找 IP 地址。如果连接了多个网络，Ansible 集成将查找 deviceIndex（即，连接到计算机的网卡 (NIC) 的索引）为最小的网络。如果蓝图中未指定 deviceIndex 属性，则集成将使用第一个连接的网络。

有关为 vRealize Automation Cloud Assembly 中的集成配置 Ansible 开源的更多详细信息，请参见[什么是 vRealize Automation Cloud Assembly 中的配置管理](#)。

前提条件

- Ansible 控制机必须使用 Ansible 版本 2.6.0 或更高版本。
- 用户必须具有 Ansible 清单文件所在目录的读取/写入权限。此外，如果已存在清单文件，用户也必须具有该文件的读取/写入权限。
- 如果使用的是具有 sudo 选项的非 root 用户，请确保在 sudoers 文件中设置以下内容：

```
Defaults:user_name !requiretty
```

和

```
username ALL=(ALL) NOPASSD: ALL
```

- 通过在 /etc/ansible/ansible.cfg 或 ~/.ansible.cfg 中设置 host_key_checking = False，确保禁用主机密钥检查。
- 通过将以下行添加到 /etc/ansible/ansible.cfg 或 ~/.ansible.cfg 文件，确保设置保管库密码：

```
vault_password_file = /path/to/password_file
```

保管库密码文件包含纯文本格式的密码，仅当蓝图或部署提供在 **ACM** 和节点之间使用的用户名和密码组合时，才会使用该密码，如以下示例中所示。

```
echo 'myStr0ng9@88w0rd' > ~/.ansible_vault_password.txt
echo 'ANSIBLE_VAULT_PASSWORD_FILE=~/.ansible_vault_password.txt' > ~/.profile      #
Instead of this way, you can also set it setting
'vault_password_file=~/.ansible_vault_password.txt' in either /etc/ansible/ansible.cfg or
~/.ansible.cfg
```

- 为避免在尝试运行 **playbook** 时出现主机密钥故障，建议您在 `/etc/ansible/ansible config` 中包含以下设置。

```
[paramiko_connection]
record_host_keys = False

[ssh_connection]
#ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
ssh_args = -o UserKnownHostsFile=/dev/null          # If you already have any
options set for ssh_args, just add the additional option shown here at the end.
```

步骤

- 1 选择**基础架构 > 连接 > 集成**，然后单击**添加集成**。
 - 2 单击“Ansible”。
- 此时将显示 Ansible 配置页面。
- 3 输入 Ansible 开源实例的主机名、清单文件路径和其他必填信息。
 - 4 单击**验证**以检查集成。
 - 5 单击**添加**。

结果

Ansible 可与蓝图一同使用。

后续步骤

将 Ansible 组件添加到所需蓝图。

- 1 在蓝图画布页面上，选择蓝图选项菜单上“配置管理”标题下的“Ansible”，然后将 Ansible 组件拖动到画布。
- 2 使用右侧面板配置适当的 Ansible 属性，例如，指定要运行的 **playbook**。

在 Ansible 中，用户可以为单个主机分配一个变量，然后在 **playbook** 中使用该变量。通过 Ansible 开源集成，可以在蓝图中指定这些主机变量。**hostVariables** 属性必须采用正确的 YAML 格式（Ansible 控制计算机要求采用此格式），且此内容将放置在以下位置：

```
parent_directory_of_inventory_file/host_vars/host_ip_address/vra_user_host_vars.yml
```


Ansible 清单文件的默认位置在 Cloud Assembly 的“集成”页面中添加的 Ansible 帐户中进行定义。Ansible 集成不会在蓝图中验证 hostVariable YAML 语法，但如果在格式或语法错误的情况下运行 playbook，Ansible 控制计算机将出现异常。

以下蓝图 YAML 代码段显示了 hostVariables 属性的示例用法。

```
Cloud_Ansible_1:
  type: Cloud.Ansible
  properties:
    host: '${resource.AnsibleLinuxVM.*}'
    osType: linux
    account: ansible-CAVA
    username: ${input.username}
    password: ${input.password}
    maxConnectionRetries: 20
    groups:
      - linux_vms
    playbooks:
      provision:
        - /root/ansible-playbooks/install_web_server.yml
    hostVariables: |
      message: Hello ${env.requestedBy}
      project: ${env.projectName}
```

如何在 vRealize Automation Cloud Assembly 中创建 Active Directory 集成

vRealize Automation Cloud Assembly 支持与 Active Directory 服务器集成，以便在置备虚拟机之前在 Active Directory 服务器内的指定组织单位 (OU) 中现成创建计算机帐户。

Active Directory 集成只支持 Active Directory 服务器的 LDAP 连接。

前提条件

- 如果要配置与 vCenter 内部部署的 Active Directory 集成，则必须为 Active Directory 集成配置可扩展性代理。选择**可扩展性 > 活动 > 集成**，然后选择**可扩展性操作内部部署**。
- 如果要在云中配置与 Active Directory 的集成，您必须拥有 Microsoft Azure 或 Amazon Web Services 帐户。
- 您的 Active Directory 服务器必须使用 LDAP 服务器连接。
- 您必须为项目配置适当的云区域以及映像映射和特定实例映射，才能与 Active Directory 搭配使用。
- 在将 Active Directory 集成与项目关联之前，必须在 Active Directory 上预先创建所需的 OU。

步骤

- 1 选择**基础架构 > 连接 > 集成**，然后选择**新建集成**。
- 2 单击 **Active Directory**。
- 3 在**摘要**选项卡上，输入适当的 LDAP 主机和环境名称。
- 4 输入 LDAP 服务器的用户名和密码。

- 5 为 Active Directory 中的所需用户和组输入适当的“基本 DN”。

注 每个 Active Directory 集成只能指定一个 DN。

- 6 单击**验证**以确保集成正常运行。
- 7 输入此集成的名称和说明。
- 8 单击**保存**。
- 9 单击**项目**选项卡，将项目添加到 Active Directory 集成。

在**添加项目**对话框中，必须选择项目名称和相对 DN（即在“摘要”选项卡上指定的“基本 DN”中存在的 DN）。

- 10 单击**保存**。

结果

现在，可以将具有 Active Directory 集成的项目关联到蓝图。使用此蓝图置备计算机时，该计算机将预转储在指定的 Active Directory 和组织单位中。

vRealize Automation Cloud Assembly 中的载入计划是什么

您可以使用工作负载载入计划来确定已从目标区域或数据中心的云帐户类型收集数据但尚未由 vRealize Automation Cloud Assembly 项目管理的计算机。

添加包含在 vRealize Automation Cloud Assembly 之外部署的计算机的云帐户时，这些计算机不会由 Cloud Assembly 进行管理，直到您载入它们。使用载入计划将非受管计算机载入到 vRealize Automation Cloud Assembly 中进行管理。您需要创建一个计划，将计算机填充到该计划中，然后运行该计划以导入这些计算机。使用载入计划，您可以创建蓝图，还可以创建一个或多个部署。

您可以在一个计划中载入一个或多个非受管计算机。您可以手动或使用筛选规则选择计算机。筛选规则根据计算机名称、状态、IP 地址和标记等条件选择要载入的计算机。

- 每小时可在单个载入计划中载入多达 3,500 个非受管计算机。
- 每小时可在多个载入计划中同时载入多达 17000 个非受管计算机。

可用于工作负载载入的计算机将在与特定云帐户类型和区域相关的**资源 > 计算机**页面中列出，并且在“来源”列中标记为 Discovered。只会列出已收集数据的计算机。在载入计算机后，它们会在“来源”列中显示为 Deployed。

系统会自动将运行工作负载载入计划的人员指定为计算机所有者。

载入示例

有关载入方法的示例，请参见[示例：将选定的计算机载入为 vRealize Automation Cloud Assembly 中的单个部署](#)和[示例：将规则筛选的计算机作为单独部署载入 vRealize Automation Cloud Assembly](#)。

载入事件订阅

运行计划时，将创建 Deployment Onboarded 事件。使用“可扩展性”选项卡中的选项，您可以订阅这些部署事件并对其执行操作。

示例：将选定的计算机载入为 vRealize Automation Cloud Assembly 中的单个部署

在本示例中，您将两台非受管计算机载入为单个 vRealize Automation Cloud Assembly 部署，并为计划中的所有计算机创建一个蓝图。

创建云帐户时，将对与该云帐户关联的所有计算机收集数据，然后在[基础架构 > 资源 > 计算机](#)页面中显示这些计算机。如果云帐户包含在 vRealize Automation Cloud Assembly 外部部署的计算机，您可以使用载入计划允许 vRealize Automation Cloud Assembly 管理计算机部署。

前提条件

- 确认您具有所需的用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 请参见 [vRealize Automation Cloud Assembly 中的载入计划是什么](#)。
- 创建并准备 vRealize Automation Cloud Assembly 项目。

此过程涉及基本 WordPress 用例中的一些步骤。请参见 [WordPress 用例](#)。

- 创建项目，在项目中添加用户并分配用户角色。请参见 [WordPress 用例：创建项目](#)。
- 为项目创建 Amazon Web Services 云帐户。请参见 [WordPress 用例：添加云帐户](#)。

此过程中的 Amazon Web Services 云帐户包含在将云帐户添加到 vRealize Automation Cloud Assembly 之前部署的计算机，以及由 vRealize Automation Cloud Assembly 以外的应用程序部署的计算机。

- 验证[计算机](#)页面是否包含要载入的计算机。请参见[计算机资源](#)。

步骤

- 1 转到[基础架构 > 载入](#)。
- 2 单击[新建载入计划](#)并输入示例值。

| 设置 | 示例值 |
|------|-------------------------------|
| 计划名称 | VC-sqa-deployments |
| 说明 | OurCo-AWS 云帐户的 AWS 计算机的示例载入计划 |
| 云帐户 | OurCo-AWS |
| 默认项目 | WordPress |

- 3 单击[创建](#)。

- 4 在计划的部署选项卡中，单击**选择计算机**，选择一台或多台计算机，然后单击**确定**。



- 5 选择**创建一个包含所有计算机的部署**，然后单击**创建**。
- 6 单击新部署名称旁边的复选框，然后单击**蓝图...**。
- 7 单击以 **Cloud Assembly** 格式创建蓝图。
- 8 输入蓝图名称，然后单击**保存**。



注 当您的载入计划使用 vSphere 计算机时，必须在载入过程完成后编辑蓝图。载入过程无法链接源 vSphere 计算机及其计算机模板，且生成的蓝图将在蓝图代码中包含 `imageRef: "no image available"` 条目。在 `imageRef` 字段中指定正确的模板名称之前，无法部署蓝图。为了在载入过程完成后更轻松地查找和更新蓝图，请使用部署的**蓝图配置**页面上的**蓝图名称**选项。记录自动生成的蓝图名称，或输入并记录您选择的蓝图名称。载入完成后，找到并打开蓝图，并将 `imageRef` 字段中的 `"no image available"` 条目替换为正确的模板名称。

- 9 单击部署名称复选框，单击**运行**，然后在**运行计划**页面中再次单击**运行**。
- 所选的 Amazon Web Services 计算机将作为单个部署与随附的蓝图一起载入。
- 10 通过单击**蓝图**选项卡，然后单击蓝图名称，打开并检查蓝图。

11 通过单击**部署**选项卡，然后单击部署名称，打开并检查部署。

示例：将规则筛选的计算机作为单独部署载入 vRealize Automation Cloud Assembly

在此示例中，您将使用筛选规则来载入状态为 On 且名称以字母 BG 开头的计算机。您还将为计划中的每台计算机创建单独的 vRealize Automation Cloud Assembly 蓝图和部署。

创建云帐户时，将对与该云帐户关联的所有计算机收集数据，然后在**基础架构 > 资源 > 计算机**页面中显示这些计算机。如果云帐户包含在 vRealize Automation Cloud Assembly 外部部署的计算机，您可以使用载入计划允许 vRealize Automation Cloud Assembly 管理计算机部署。

前提条件

- 确认您具有所需的用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 请参见 [vRealize Automation Cloud Assembly 中的载入计划是什么](#)。
- 创建并准备一个 vRealize Automation Cloud Assembly 项目，然后将一个或多个云帐户填充到该项目中。

这涉及到引导式设置过程中的一些基本步骤。

- 创建项目，在项目中添加用户并分配用户角色。请参见 [WordPress 用例：创建项目](#)。
- 在项目的指定区域中创建一个或多个云帐户。请参见 [WordPress 用例：添加云帐户](#)。
- 验证**计算机**页面是否包含要载入的计算机。请参见[计算机资源](#)。

步骤

- 1 转到**基础架构 > 载入**。
- 2 单击**新建载入计划**并输入值。

| 设置 | 示例值 |
|------|--------------------------------|
| 计划名称 | ob_rules_1 |
| 说明 | Machine onboarding with rules1 |
| 云帐户 | rs-aws |
| 默认项目 | rs-project |

新建载入计划



计划名称 ob_rules_1

描述

Machine onboarding with rules1

必备条件

添加云帐户，并为要载入的计算机所在的计算资源创建云区域。

创建至少具有一个用户的项目，并为该项目授予对云区域的访问权限。

云帐户

Q rs-aws



默认项目

Q rs-project



取消

创建

3 单击创建。

ob_rules_1

摘要 规则 计算机 部署

计划名称 ob_rules_1

描述 Machine onboarding with rules1

计划状态

上次运行时间 从不

源信息

云帐户 346test_vc_account

部署标记键

应用

目标配置

默认项目 Q 123

保存 运行 取消

4 单击规则选项卡，然后单击添加规则。

您可以创建一个或多个规则以根据特定计算机特性选择一组计算机进行载入。

ob_rules_1

摘要 规则 计算机 部署

使用规则将计算机添加到此计划。 ⓘ

添加规则 编辑 删除

| <input type="checkbox"/> | 名称 |
|--------------------------|----|
|--------------------------|----|

5 输入规则名称，例如 ob_rules_1。

添加 规则

创建 基于筛选器的规则，将用于在此计划中填充计算机。

规则名称 * ob_rules_1

6 添加筛选器以生成规则。

对于此示例，请使用**筛选器**下拉菜单上的**状态**和**名称**筛选器指定名称包含 BG* 且状态为 On 的所有计算机。

规则名称 * ob_rules_1

筛选...

| 属性 | 状态 | 地址 | 创建时间 | 标记 |
|----|------|---------------|------------|----|
| 任意 | ▶ 开启 | 10.184.68.223 | 2020年1月09日 | |
| 名称 | | | | |
| 状态 | | | | |
| 地址 | | | | |
| 标记 | | | | |

4 计算机

规则名称 * ob_rules_1

名称: BG* × 状态: 开启

7 单击保存。

虽然您可以创建其他规则，但此示例使用单个规则。

ob_rules_1

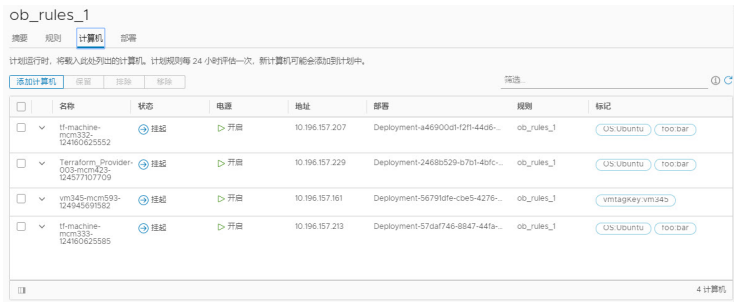
摘要 规则 计算机 部署

使用规则将计算机添加到此计划。 ⓘ

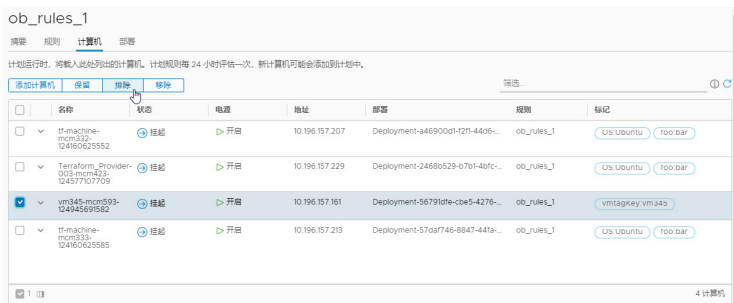
添加规则 编辑 删除

| <input type="checkbox"/> | 名称 | 状态 | 筛选器 |
|--------------------------|------------|------|-----------|
| <input type="checkbox"/> | ob_rules_1 | ✔ 确定 | Name: BG* |

- 8 单击**计算机**选项卡。在此示例中，选择了 4 台计算机，其中 3 台计算机的名称以字母 BG 开头，一台计算机的名称包含字母 BG。



- 9 通过选中对应的复选框并单击**排除**，移除名称不以 BG 开头的那台计算机。



- 10 单击**部署**选项卡。

名称以字母 BG 开头且状态为 On 的 3 台计算机已准备好进行部署。默认情况下，将为每台计算机创建单独的蓝图和部署。

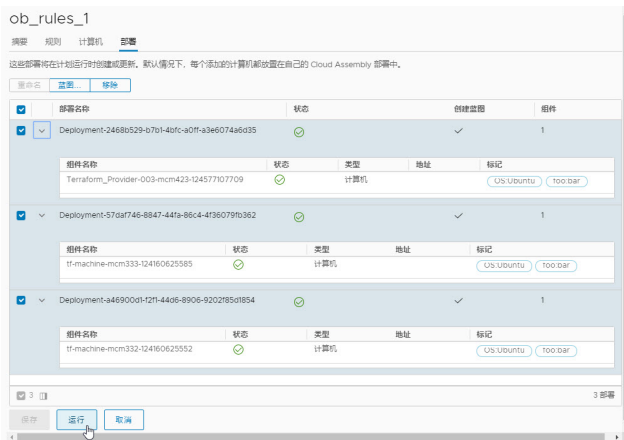


- 11 选中三个部署名称旁边的复选框，然后依次单击**蓝图**、以 **Cloud Assembly** 格式创建蓝图和**保存**。



注 当您的载入计划使用 vSphere 计算机时，必须在载入过程完成后编辑蓝图。载入过程无法链接源 vSphere 计算机及其计算机模板，且生成的蓝图将在蓝图代码中包含 `imageRef: "no image available"` 条目。在 `imageRef:` 字段中指定正确的模板名称之前，无法部署蓝图。为了在载入过程完成后更轻松地查找和更新蓝图，请使用部署的**蓝图配置**页面上的**蓝图名称**选项。记录自动生成的蓝图名称，或输入并记录您选择的蓝图名称。载入完成后，找到并打开蓝图，并将 `imageRef:` 字段中的 `"no image available"` 条目替换为正确的模板名称。

- 12 在**部署**页面中，选中三个部署名称旁边的复选框，然后单击**运行**。



13 当系统提示您确认时，单击**运行**以载入计算机。



将运行该计划，并将计算机载入到 vRealize Automation Cloud Assembly 中进行管理。将为每台计算机创建单独的蓝图和部署。

vRealize Automation Cloud Assembly 环境的高级配置

可以配置 vRealize Automation Cloud Assembly 环境以进一步支持项目配置、集成和部署。

有关管理方法（如使用用户和日志以及加入或退出客户体验计划）的相关信息和其他信息，请参见《[管理 vRealize Automation](#)》帮助。

如何配置 vRealize Automation 的 Internet 代理服务器

对于无法直接访问 Internet 的隔离网络上的 vRealize Automation 8.0.1 转发安装，可以使用 Internet 代理服务器，以便允许通过代理功能访问 Internet。Internet 代理服务器支持 HTTP 和 HTTPS。

要配置和使用公有云提供程序（如 Amazon Web Services (AWS)、Microsoft Azure 和 Google Cloud Platform (GCP)）以及外部集成点（如 IPAM、Ansible 和 Puppet），对于 vRealize Automation，您必须配置 Internet 代理服务器，以访问 vRealize Automation Internet 代理服务器。

vRealize Automation 包含一个可与 Internet 代理服务器通信的内部代理服务器。如果使用 `vracli proxy set ...` 命令对代理服务器进行了相应配置，则该服务器可与代理服务器通信。如果您尚未为组织配置 Internet 代理服务器，则 vRealize Automation 内部代理服务器会尝试直接连接到 Internet。

可以使用提供的 `vracli` 命令行实用程序将 vRealize Automation 设置为使用 Internet 代理服务器。有关如何使用 `vracli API` 的信息，可通过在 `vracli` 命令行中使用 `--help` 参数获得，例如 `vracli proxy --help`。

访问 Internet 代理服务器需要使用 vRealize Automation 中内置的基于操作的可扩展性 (ABX) 内部部署嵌入式控制。

注 不支持通过 Internet 代理访问 Workspace ONE Access（以前称为 VMware Identity Manager）。您无法使用 `vracli set vidm` 命令通过 Internet 代理服务器访问 Workspace ONE Access。

内部代理服务器要求将 IPv4 作为其默认 IP 格式。不要求对 TLS (HTTPS) 证书流量执行 Internet 协议限制、身份验证或中间人操作。

前提条件

- 确认您已有 HTTP 或 HTTPS 服务器，且可在能够将出站流量传递到外部站点的 vRealize Automation 网络中将其用作 Internet 代理服务器。必须针对 IPv4 配置连接。
- 确认目标 Internet 代理服务器已配置为支持 IPv4 作为其默认 IP 格式，而非 IPv6。
- 如果 Internet 代理服务器使用 TLS，并且需要与其客户端建立 HTTPS 连接，则必须先使用以下命令之一导入服务器证书，然后再设置代理配置。

```
■ vracli certificate proxy --set path_to_proxy_certificate.pem
```

```
■ vracli certificate proxy --set stdin
```

使用 stdin 参数执行交互式输入。

步骤

- 1 为 Kubernetes 使用的 pod 或容器创建代理配置。在此示例中，使用 HTTP 方案访问代理服务器。

```
vracli proxy set --host http://proxy.vmware.com:3128
```

- 2 显示代理配置。

```
vracli proxy show
```

结果将类似于以下内容：

```
{
  "enabled": true,
  "host": "10.244.4.51",
  "java-proxy-exclude": "/*.local|*.localdomain|localhost|10.244.*|192.168.*|172.16.*|kubernetes|sc2-rdops-vm06-dhcp-198-120.eng.vmware.com|10.192.204.9|*.eng.vmware.com|sc2-rdops-vm06-dhcp-204-9.eng.vmware.com|10.192.213.146|sc2-rdops-vm06-dhcp-213-146.eng.vmware.com|10.192.213.151|sc2-rdops-vm06-dhcp-213-151.eng.vmware.com",
  "java-user": null,
  "password": null,
  "port": 3128,
  "proxy-exclude": ".local,.localdomain,localhost,10.244.,192.168.,172.16.,kubernetes,sc2-rdops-vm06-dhcp-198-120.eng.vmware.com,10.192.204.9,.eng.vmware.com,sc2-rdops-vm06-dhcp-204-9.eng.vmware.com,10.192.213.146,sc2-rdops-vm06-dhcp-213-146.eng.vmware.com,10.192.213.151,sc2-rdops-vm06-dhcp-213-151.eng.vmware.com",
  "scheme": "http",
  "upstream_proxy_host": null,
  "upstream_proxy_password_encoded": "",
  "upstream_proxy_port": null,
  "upstream_proxy_user_encoded": "",
  "user": null,
  "internal.proxy.config": "dns_v4_first on \nhttp_port 0.0.0.0:3128\nlogformat squid %ts.%03tu %6tr %>a %Ss/%03>Hs %<st %rm %ru %[un %Sh/%<a %mt\naccess_log stdio:/tmp/logger squid\ncoredump_dir /&ncache deny all \nappend_domain .prelude.svc.cluster.local\nacl mylan src 10.0.0.0/8\nacl mylan src 127.0.0.0/8\nacl mylan src 192.168.3.0/24\nacl proxy-exclude dstdomain .local\nacl proxy-exclude dstdomain .localdomain\nacl proxy-exclude dstdomain localhost\nacl proxy-exclude dstdomain 10.244.\n\nacl proxy-exclude dstdomain 192.168.\n\nacl proxy-exclude dstdomain kubernetes\n\nacl proxy-exclude dstdomain 10.192.204.9\n\nacl proxy-exclude dstdomain .eng.vmware.com\n\nacl
```

```
proxy-exclude dstdomain 10.192.213.146\nacl proxy-exclude dstdomain
10.192.213.151\nalways_direct allow proxy-exclude\nhttp_access allow mylan\nhttp_access
deny all\n# End autogen configuration\n",
    "internal.proxy.config.type": "default"
}
```

注 如果您已为组织配置 Internet 代理服务器，则上例中将显示 "internal.proxy.config.type": "non-default"，而不是 'default'。为安全起见，不会显示密码。

注 如果使用 `-proxy-exclude` 参数，则必须编辑默认值。例如，如果要将 `acme.com` 添加为无法使用 Internet 代理服务器访问的域，请执行以下步骤：

- a 输入 `vracli proxy default-no-proxy` 以获取默认的 `proxy-exclude` 设置。这是自动生成的域和网络列表。
- b 编辑值以添加 `.acme.com`。
- c 输入 `vracli proxy set --proxy-exclude ...` 以更新配置设置。
- d 运行 `/opt/scripts/deploy.sh` 命令以重新部署环境。

3 （可选）排除 DNS 域、FQDN 和 IP 地址，使其无法通过 Internet 代理服务器进行访问。

始终使用 `parameter --proxy-exclude` 修改 `proxy-exclude` 变量的默认值。要添加域 `exclude.vmware.com`，请先使用 `vracli proxy show` 命令，然后复制 `proxy-exclude` 变量，并使用 `vracli proxy set ...` 命令添加域值，如下所示：

```
vracli proxy set --host http://proxy.vmware.com:3128 --proxy-exclude
"exclude.vmware.com,docker-
registry.prelude.svc.cluster.local,localhost,.local,.cluster.local,10.244.,192.,172.16.,sc-
rdops-vm11-dhcp-75-38.eng.vmware.com,10.161.75.38,.eng.vmware.com"
```

注 将元素添加到 `proxy-exclude`，而不是替换值。如果删除 `proxy-exclude` 默认值，vRealize Automation 将无法正常运行。如果发生这种情况，请删除代理配置，然后重新开始。

- 4 使用 `vracli proxy set ...` 命令设置 Internet 代理服务器后，可以使用 `vracli proxy apply` 命令更新 Internet 代理服务器配置，并使最新的代理设置处于活动状态。
- 5 如果尚未执行此操作，请运行以下命令以激活脚本更改：

```
/opt/scripts/deploy.sh
```

- 6 （可选）如果需要，请配置代理服务器以支持端口 22 上的外部访问。

要支持 Puppet 和 Ansible 等集成，代理服务器必须允许使用端口 22 访问相关主机。

示例： Squid 配置示例

相对于步骤 1，如果设置的是 Squid 代理，可以在 `/etc/squid/squid.conf` 中调整配置，根据以下示例进行调整：

```
acl localnet src 192.168.11.0/24

acl SSL_ports port 443

acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

http_access allow !Safe_ports
http_access allow CONNECT !SSL_ports
http_access allow localnet

http_port 0.0.0.0:3128

maximum_object_size 5 GB
cache_dir ufs /var/spool/squid 20000 16 256
coredump_dir /var/spool/squid
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern (Release|Packages(.gz)*)$ 0 20% 2880
refresh_pattern . 0 20% 4320

client_persistent_connections on
server_persistent_connections on
```

如何在 vRealize Automation 中使用 cloud-init 或 cloudbase-init 设置 Windows 模板

您可以将 vRealize Automation 环境配置为支持 cloud-init 和 cloudbase-init 脚本，以便在 Windows 部署中创建和使用自定义映像。

在映像映射和蓝图代码中支持云配置脚本。云配置脚本遵循 cloud-init 和 cloudbase-init 规则和格式。有关 cloud-init 和 cloudbase-init 的相关信息，请参见 <https://cloudbase.it/cloudbase-init>。

注 有关针对 Windows 配置 cloud-init 或 cloudbase-init 的信息，请参见以下 VMware 博客：

- [Windows Cloud-Init solution](#) 博客文章
- [Windows guest initialization with Cloudbase-Init in vCenter](#) 博客文章

有关为 Linux 配置 cloud-init 的相关信息，请参见博客文章为 [vSphere 构建 vRealize Automation Cloud Ready Ubuntu](#) 模板。

有关在 vRealize Automation 中使用 cloud-init 和云配置脚本的信息，请参见：

- [了解有关 vRealize Automation Cloud Assembly 中的映像映射的更多信息](#)
- [如何在 vRealize Automation Cloud Assembly 蓝图中自动初始化计算机](#)

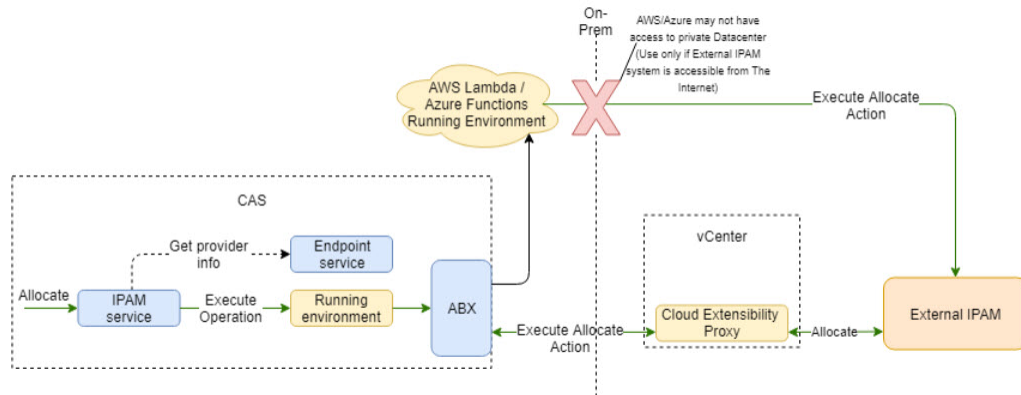
另请参见以下 cloud-init 和 cloudbase-init 供应商页面：

- <https://cloud-init.io>
- <https://cloudinit.readthedocs.io/en/latest/index.html>
- <https://cloudinit.readthedocs.io/en/latest/topics/examples.html#yaml-examples>
- <https://cloudbase.it/cloudbase-init>
- <https://cloudbase-init.readthedocs.io/en/latest/services.html>
- <https://cloudbase-init.readthedocs.io/en/latest/userdata.html#userdata>

如何使用 IPAM SDK 为 vRealize Automation 创建提供程序特定的外部 IPAM 集成软件包

外部 IPAM 供应商和合作伙伴可以下载并使用 IPAM SDK 创建 IPAM 集成软件包，以便 vRealize Automation 支持其提供程序特定的 IPAM 解决方案。

有关使用提供的 IPAM SDK 为 vRealize Automation 构建和部署自定义 IPAM 集成软件包的过程，请参见为 [VMware Cloud Assembly 创建和部署提供程序特定的 IPAM 集成软件包文档](#)。按照文档中所述，可以从 [VMware Solution Exchange](#) 商城下载 VMware vRealize Automation Third-Party IPAM SDK。



在花时间使用 IPAM SDK 创建供应商特定的 IPAM 集成软件包之前，请检查 vRealize Automation 是否已存在这样一个集成软件包。可以从以下位置检查提供程序特定的 IPAM 集成软件包：IPAM 提供程序的网站、VMware Solution Exchange 商城以及 vRealize Automation 的商城选项卡。

虽然提供程序特定的外部 IPAM 集成用例示例特定于供应商，但也包含有用的参考信息。

vRealize Automation Cloud Assembly 用例

3

这些用例说明了在 vRealize Automation Cloud Assembly 中创建资源基础架构、进行设计并将应用程序部署到该基础架构的示例。

这些用例仅提供示例值。您自己的环境结构和命名约定会有所不同。

本章讨论了以下主题：

- [WordPress 用例](#)
- [VMware Cloud on AWS 用例](#)
- [提供程序特定的外部 IPAM 集成用例](#)

WordPress 用例

此端到端 vRealize Automation Cloud Assembly 用例显示了创建基础架构并将 WordPress 站点部署到该基础架构的示例。

查看顺序设置，以了解使 WordPress 站点完工的过程。

请注意，用例中的值仅用作示例。不能在您的环境中逐字使用这些值。

请考虑在何处需要替换为您自己的值或从示例值外插值，以便满足您自己的云计算基础架构需求和部署需求。

步骤

1 [WordPress 用例：创建基础架构](#)

作为云管理员，您首先需要配置资源，随后工程团队可以在这些资源中开发和测试 WordPress 站点以及将其投入生产。

2 [WordPress 用例：创建项目](#)

项目指定具有置备权限的用户，并配置可能的置备量。

3 [WordPress 用例：创建并扩展蓝图](#)

作为开发人员，您可以采用通用 vRealize Automation Cloud Assembly 蓝图的形式定义 WordPress 站点，以便可以部署到任何云供应商。

WordPress 用例：创建基础架构

作为云管理员，您首先需要配置资源，随后工程团队可以在这些资源中开发和测试 WordPress 站点以及将其投入生产。

基础架构包含云目标，以及有关 WordPress 站点所需的计算机、网络 and 存储的定义。

步骤

1 WordPress 用例：添加云帐户

在此步骤，云管理员需要添加两个云帐户。项目需要在 AWS 上执行开发和测试工作，然后在 Azure 上投入生产。

2 WordPress 用例：添加云区域

在此步骤，云管理员将添加三个云区域，它们分别用于开发、测试和生产。

3 WordPress 用例：添加特定实例映射

在此步骤，云管理员将添加特定实例映射，以考虑可能因部署而异的容量需求。

4 WordPress 用例：添加映像映射

在此步骤中，云管理员为 Ubuntu 添加映像映射，以及 WordPress 服务器及其 MySQL 数据库服务器的主机。

5 WordPress 用例：添加网络配置文件

在此步骤，云管理员需要将网络配置文件添加到每个云区域。

6 WordPress 用例：添加存储配置文件

在此步骤，云管理员需要将存储配置文件添加到每个云区域。

WordPress 用例：添加云帐户

在此步骤，云管理员需要添加两个云帐户。项目需要在 AWS 上执行开发和测试工作，然后在 Azure 上投入生产。

步骤

- 1 转到**基础架构 > 连接 > 云帐户**。
- 2 单击**添加云帐户**，选择“Amazon Web Services”，然后输入值。

| 设置 | 示例值 |
|---------|----------------------|
| 访问密钥 ID | R5SDR3PXVV2ZW8B7YNSM |
| 私有访问密钥 | SZXAINXU4UHNAQ1E156S |
| 名称 | OurCo-AWS |
| 说明 | WordPress |
| 功能 | cloud:aws |

请记住，所有值仅是用例示例而已。您的帐户详细信息将与此不同。

- 要验证凭据，请单击**验证**。
- 单击**添加**。
- 编辑新添加的帐户**配置**，并允许置备到 **us-east-1** 和 **us-west-2** 区域。
- 单击**添加云帐户**，选择“Microsoft Azure”，然后输入值。

| 设置 | 示例值 |
|------------|----------------------------------------------|
| 订阅 ID | ef2avpf-dfdv-zxluguii-g4h0-i8ep2jwp4c9arbfe |
| 租户 ID | dso9wv3-4zgc-5nrcy5h3m-4skf-nnovp40wfxsro22r |
| 客户端应用程序 ID | bg224oq-3ptp-mbhi6aa05-q511-uflyjr2sttyik6bs |
| 客户端应用程序密钥 | 7uqxi57-0wtn-kymgf9wcj-t2l7-e52e4nu5fig4pmdd |
| 名称 | OurCo-Azure |
| 说明 | WordPress |
| 功能 | cloud:az |

- 要验证凭据，请单击**验证**。
- 单击**添加**。
- 编辑新添加的帐户**配置**，并允许置备到 **East US** 区域。

后续步骤

添加要让项目将 WordPress 站点部署到的云区域。请参见 [WordPress 用例：添加云区域](#)。

WordPress 用例：添加云区域

在此步骤，云管理员将添加三个云区域，它们分别用于开发、测试和生产。

云区域是项目为了支持 WordPress 站点而将计算机、网络 and 存储部署到的资源。

前提条件

添加云帐户。请参见 [WordPress 用例：添加云帐户](#)。

步骤

- 转到**基础架构 > 配置 > 云区域**。
- 单击**新建云区域**，并输入开发环境的值。

| 云区域设置 | 示例值 |
|-------|---------------------|
| 帐户/区域 | OurCo-AWS/us-east-1 |
| 名称 | OurCo-AWS-US-East |
| 说明 | WordPress |

| 云区域设置 | 示例值 |
|-------|---------|
| 放置策略 | 默认 |
| 能力标记 | env:dev |

请记住，所有值仅是用例示例而已。您的区域详细信息将与此不同。

- 3 单击 **计算资源**，并验证其中是否列出了预期的区域。
- 4 单击 **创建**。
- 5 分别使用用于测试环境的值和用于生产环境的值重复以上过程两次。

| 云区域设置 | 示例值 |
|-------|---------------------|
| 帐户/区域 | OurCo-AWS/us-west-2 |
| 名称 | OurCo-AWS-US-West |
| 说明 | WordPress |
| 放置策略 | 默认 |
| 能力标记 | env:test |

| 云区域设置 | 示例值 |
|-------|---------------------|
| 帐户/区域 | OurCo-Azure/East US |
| 名称 | OurCo-Azure-East-US |
| 说明 | WordPress |
| 放置策略 | 默认 |
| 能力标记 | env:prod |

后续步骤

通过添加特定实例映射，考虑不同大小的计算机部署。请参见 [WordPress 用例：添加特定实例映射](#)。

WordPress 用例：添加特定实例映射

在此步骤，云管理员将添加特定实例映射，以考虑可能因部署而异的容量需求。

特定实例映射通俗地称为“T 恤调整大小”。

前提条件

添加云区域。请参见 [WordPress 用例：添加云区域](#)。

步骤

- 1 转到 **基础架构 > 配置 > 特定实例映射**。每个云区域都必须能够容纳小型、中型和大型特定实例。

2 单击**新建特定实例映射**，并输入开发云区域的值。

| 设置 | 示例值 |
|------------|------------------------------------|
| 特定实例名称 | small |
| 帐户/区域 值 | OurCo-AWS/us-east-1 t2.micro |
| 帐户/区域 值 | OurCo-AWS/us-west-2 t2.micro |
| 帐户/区域 值 | OurCo-Azure/East US Standard_A0 |

请记住，所有值仅是用例示例而已。您的特定实例将与此不同。

3 单击**创建**。

4 分别使用用于中型特定实例的值和用于大型特定实例的值重复以上过程两次。

| 设置 | 示例值 |
|------------|------------------------------------|
| 特定实例名称 | medium |
| 帐户/区域 值 | OurCo-AWS/us-east-1 t2.medium |
| 帐户/区域 值 | OurCo-AWS/us-west-2 t2.medium |
| 帐户/区域 值 | OurCo-Azure/East US Standard_A3 |

| 设置 | 示例值 |
|------------|------------------------------------|
| 特定实例名称 | large |
| 帐户/区域 值 | OurCo-AWS/us-east-1 t2.large |
| 帐户/区域 值 | OurCo-AWS/us-west-2 t2.large |
| 帐户/区域 值 | OurCo-Azure/East US Standard_A7 |

后续步骤

通过添加映像映射，对操作系统进行计划。请参见 [WordPress 用例：添加映像映射](#)。

WordPress 用例：添加映像映射

在此步骤中，云管理员为 Ubuntu 添加映像映射，以及 WordPress 服务器及其 MySQL 数据库服务器的主机。

每个云区域都需要一个 Ubuntu 映像映射。

前提条件

添加云区域。请参见 [WordPress 用例：添加云区域](#)。

步骤

- 1 转到**基础架构 > 配置 > 映像映射**。
- 2 单击**新建映像映射**，然后输入 Ubuntu 服务器的值。

| 设置 | 示例值 |
|------------|-----------------------------------------------------------|
| 映像名称 | ubuntu-16 |
| 帐户/区域 值 | OurCo-AWS/us-east-1 ubuntu-16.04-server-cloudimg-amd64 |
| 帐户/区域 值 | OurCo-AWS/us-west-2 ubuntu-16.04-server-cloudimg-amd64 |
| 帐户/区域 值 | OurCo-Azure/East US azul-zulu-ubuntu-1604-923eng |

请记住，所有值仅是用例示例而已。您的映像会有变化。

- 3 单击**创建**。

后续步骤

添加网络。请参见 [WordPress 用例：添加网络配置文件](#)。

WordPress 用例：添加网络配置文件

在此步骤，云管理员需要将网络配置文件添加到每个云区域。

在每个配置文件中，管理员可以为 WordPress 计算机添加一个网络，并添加位于最终负载平衡器另一端的另一个网络。用户最终将通过第二个网络建立连接。

前提条件

添加云区域。请参见 [WordPress 用例：添加云区域](#)。

步骤

- 1 转到**基础架构 > 配置 > 网络配置文件**。

- 2 单击**新建网络配置文件**，然后创建用于开发云区域的配置文件。

| 网络配置文件设置 | 示例值 |
|----------|---------------------|
| 帐户/区域 | OurCo-AWS/us-east-1 |
| 名称 | devnets |
| 说明 | WordPress |
| 能力标记 | env:dev |

- 3 单击**网络**，然后单击**添加网络**。

- 4 选择 wpnet 和 appnet-public，然后单击**添加**。

请记住，所有值仅是用例示例而已。您的网络名称将与此不同。

- 5 单击**创建**。

此 Wordpress 示例不要求指定网络策略或网络安全设置。

- 6 重复上述过程两次，以创建用于 Wordpress 示例测试云区域的配置文件和用于生产云区域的配置文件。在每种情况下，都需要添加 wpnet 网络和 appnet-public 网络。

| 网络配置文件设置 | 示例值 |
|----------|---------------------|
| 帐户/区域 | OurCo-AWS/us-west-2 |
| 名称 | testnets |
| 说明 | WordPress |
| 能力标记 | env:test |

| 网络配置文件设置 | 值 |
|----------|---------------------|
| 帐户/区域 | OurCo-Azure/East US |
| 名称 | prodnets |
| 说明 | WordPress |
| 能力标记 | env:prod |

后续步骤

添加存储。请参见 [WordPress 用例：添加存储配置文件](#)。

WordPress 用例：添加存储配置文件

在此步骤，云管理员需要将存储配置文件添加到每个云区域。

管理员将快速存储放置到生产区域，而将一般存储放置到开发区域和测试区域。

前提条件

添加云区域。请参见 [WordPress 用例：添加云区域](#)。

步骤

- 1 转到**基础架构 > 配置 > 存储配置文件**。
- 2 单击**新建存储配置文件**，然后创建用于开发云区域的配置文件。
选择帐户/区域之后，将显示其他字段。

| 存储配置文件设置 | 示例值 |
|----------|------------------------|
| 帐户/区域 | OurCo-AWS/us-east-1 |
| 名称 | OurCo-AWS-US-East-Disk |
| 说明 | WordPress |
| 设备类型 | EBS |
| 卷类型 | 通用 SSD |
| 能力标记 | usage:general |

请记住，所有值仅是用例示例而已。

- 3 单击**创建**。
- 4 重复上述过程以创建用于测试云区域的配置文件。

| 存储配置文件设置 | 示例值 |
|----------|------------------------|
| 帐户/区域 | OurCo-AWS/us-west-2 |
| 名称 | OurCo-AWS-US-West-Disk |
| 说明 | WordPress |
| 设备类型 | EBS |
| 卷类型 | 通用 SSD |
| 能力标记 | usage:general |

- 5 重复上述过程以创建用于生产云区域的配置文件，生产云区域具有不同的设置，因为它是 **Azure** 区域。

| 存储配置文件设置 | 示例值 |
|----------|--------------------------|
| 帐户/区域 | OurCo-Azure/East US |
| 名称 | OurCo-Azure-East-US-Disk |
| 说明 | WordPress |

| 存储配置文件设置 | 示例值 |
|----------|------------|
| 存储类型 | 受管磁盘 |
| 磁盘类型 | 高级 LRS |
| 操作系统磁盘缓存 | 只读 |
| 数据磁盘缓存 | 只读 |
| 能力标记 | usage:fast |

后续步骤

创建项目以确定用户并定义置备设置。请参见 [WordPress 用例：创建项目](#)。

WordPress 用例：创建项目

项目指定具有置备权限的用户，并配置可能的置备量。

项目定义了用户设置和置备设置。

- 用户及其角色的权限级别
- 将部署置备到云区域时遵循的优先级
- 每个云区域的最大部署实例数

前提条件

添加云区域。请参见 [WordPress 用例：添加云区域](#)。

步骤

- 1 转到**基础架构 > 管理 > 项目**。
- 2 单击**新建项目**，然后输入名称“WordPress”。
- 3 单击**用户**，然后单击**添加用户**。
- 4 添加用户的电子邮件地址和角色。

要成功添加用户，VMware Cloud Services 管理员必须已向该用户授予 vRealize Automation Cloud Assembly 的访问权限。

请记住，此处显示的地址仅是用例示例而已。

- chris.ladd@ourco.com，成员
- kerry.mott@ourco.com，成员
- pat.tubb@ourco.com，管理员

- 5 单击**置备**，然后单击**添加云区域**。

6 添加用户可部署到的云区域。

| 项目云区域设置 | 示例值 |
|---------|---------------------|
| 云区域 | OurCo-AWS-US-East |
| 置备优先级 | 1 |
| 实例限制 | 5 |
| 云区域 | OurCo-AWS-US-West |
| 置备优先级 | 1 |
| 实例限制 | 5 |
| 云区域 | OurCo-Azure-East-US |
| 置备优先级 | 0 |
| 实例限制 | 1 |

7 单击**创建**。

8 转到**基础架构 > 配置 > 云区域**，然后打开在 [WordPress 用例：添加云区域](#)中创建的区域。

9 单击**项目**，然后验证 WordPress 项目是否可以置备到该区域。

10 检查在 [WordPress 用例：添加云区域](#)中创建的其他区域。

后续步骤

创建基本蓝图。

WordPress 用例：创建并扩展蓝图

作为开发人员，您可以采用通用 vRealize Automation Cloud Assembly 蓝图的形式定义 WordPress 站点，以便可以部署到任何云供应商。

用例蓝图包含一个 WordPress 应用程序服务器、一个 MySQL 数据库服务器和可部署到基于 AWS、Azure 或 vSphere 的云的支持组件。该蓝图从几个组件开始，然后随着您修改现有组件并添加更多组件逐渐增长。

WordPress 用例：创建基础架构中的示例包含由云管理员设置的基础架构：

- 两个云帐户，AWS 和 Azure。
- 三个云区域环境：
 - 开发 - OurCo-AWS-US-East
 - 测试 - OurCo-AWS-US-West
 - 生产 - OurCo-Azure-East-US
- 为每个区域配置特定实例映射且具有小型、中型和大型计算资源。
- 在每个区域中配置 Ubuntu 16 映像映射。
- 为每个区域配置包含内部子网和外部子网的网络配置文件：devnets、testnets 和 prodnets。
- 存储支持存档磁盘，通用存储用于开发和测试环境，快速存储用于生产环境。

- WordPress 项目包含全部三个云区域环境以及可以尝试此用例的用户。

前提条件

熟悉您的基础架构值。例如，此用例示例将 **AWS** 用于开发和测试，而将 **Azure** 用于生产。创建自己的蓝图时，可以替换为自己的值（通常由云管理员设置）。

步骤

1 WordPress 用例：创建基本蓝图

作为开发人员，您可以从仅包含最少 **WordPress** 组件（例如，仅包含一个应用程序服务器）的 vRealize Automation Cloud Assembly 蓝图开始。

2 WordPress 用例：测试基本蓝图

在开发过程中，通常从基本组件开始生成 vRealize Automation Cloud Assembly 蓝图，然后随着蓝图的增长进行部署和测试。

3 WordPress 用例：展开蓝图

在创建和测试基本的 vRealize Automation Cloud Assembly 蓝图后，将其扩展为可部署到开发、测试和最终生产环境的多层应用程序。

WordPress 用例：创建基本蓝图

作为开发人员，您可以从仅包含最少 **WordPress** 组件（例如，仅包含一个应用程序服务器）的 vRealize Automation Cloud Assembly 蓝图开始。

vRealize Automation Cloud Assembly 是一个基础架构即代码工具。您可以将组件拖动到设计画布中以开始。然后使用画布右边的代码编辑器填写详细信息。

代码编辑器允许您直接键入、剪切和粘贴代码。如果您不喜欢编辑代码，则可以在画布中选择一个资源，单击代码编辑器的**属性**选项卡，并在其中输入值。您输入的值将显示在代码中，就像直接键入它们一样。

前提条件

熟悉您的基础架构。此处显示的示例使用 **WordPress 用例：创建基础架构**中的基础架构值，但您可以将其替换为自己的基础架构值。

步骤

- 1 转到**蓝图**，然后单击**新建**。

- 2 将蓝图命名为 **Wordpress-BP**。

- 3 选择 **WordPress** 项目，然后单击**创建**。

- 4 从蓝图设计页面左侧的组件中，将两台云不可知的计算机拖动到画布中。

这两台计算机分别用作 **WordPress** 应用程序服务器 (**WebTier**) 和 **MySQL** 数据库服务器 (**DBTier**)。

- 5 在右侧，编辑计算机 **YAML** 代码以添加名称、映像、特定实例和限制标记：

```
resources:
  DBTier:
```

```

type: Cloud.Machine
properties:
  name: mysql
  image: 'ubuntu-16'
  flavor: 'small'
  constraints: - tag: env:dev
WebTier:
type: Cloud.Machine
properties:
  name: wordpress
  image: 'ubuntu-16'
  flavor: 'small'
  constraints: - tag: env:dev

```

6 将云不可知的网络拖动到画布中，并编辑其代码：

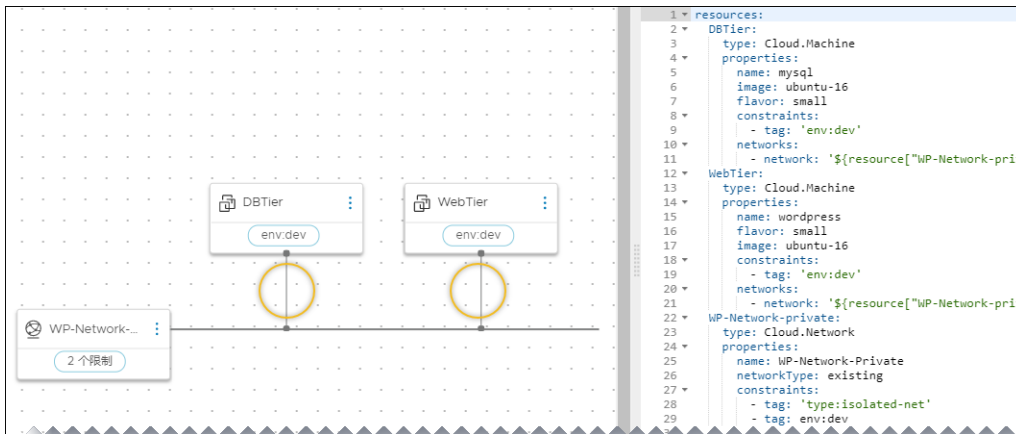
```

WP-Network-Private:
type: Cloud.Network
properties:
  name: WP-Network-Private
  networkType: existing
  constraints: - tag: 'type:isolated-net' - tag: 'env:dev'

```

7 将两台计算机连接到该网络：

在与网络块接触的行处单击并按住鼠标左键，拖动到计算机块，然后释放鼠标左键。



在编辑器中，您会发现网络代码将添加到两台计算机：

```

resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: 'small'
      constraints:
        - tag: env:dev

```

```

    networks: - network: '${resource["WP-Network-Private"].id}'
WebTier:
  type: Cloud.Machine
  properties:
    name: wordpress
    image: 'ubuntu-16'
    flavor: 'small'
    constraints:
      - tag: env:dev
    networks: - network: '${resource["WP-Network-Private"].id}'

```

8 添加用户输入提示。

在某些位置，用例基础架构已设置用于多个选项。例如：

- 用于开发、测试和生产的云区域环境
- 用于小型、中型和大型计算机的特定实例映射
- 适用于一般使用和快速使用的存储磁盘速度

您可以直接在蓝图中设置特定选项，但更好的方法是让用户在部署蓝图时选择选项。通过提示提供用户输入，您可以创建具有许多种部署方式的单个蓝图，而不必创建多个硬编码蓝图。

- a 在代码中创建 `inputs` 节，以便用户可以在部署时选择计算机大小和目标环境。定义可选择的值：

```
inputs:
  env:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
      - 'env:test'
    default: 'env:dev'
    title: Environment
    description: Target Environment
  size:
    type: string
    enum:
      - small
      - medium
      - large
    description: Size of Nodes
    title: Tier Machine Size
```

- b 在代码的 `resources` 节中，添加 `${input.input-name}` 代码以提示提供用户选择：

```
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: 'ubuntu-16'
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
  WP-Network-Private:
    type: Cloud.Network
    properties:
      name: WP-Network-Private
```

```
networkType: existing
constraints:
  - tag: 'type:isolated-net'
  - tag: '${input.env}'
```

- 9 最后，使用以下示例增强 WebTier 和 DBTier 代码。WP-Network-Private 代码不需要其他更改。请注意，增强包括对数据库服务器、数据库磁盘和部署时 cloudConfig 初始化脚本的登录访问。

| 组件 | 示例 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 其他 DBTier 输入 | <pre> username: type: string minLength: 4 maxLength: 20 pattern: '[a-z]+' title: Database Username description: Database Username userpassword: type: string pattern: '[a-z0-9A-Z@#]+\$' encrypted: true title: Database Password description: Database Password databaseDiskSize: type: number default: 4 maximum: 10 title: MySQL Data Disk Size description: Database Disk Size </pre> |
| DBTier 资源 | <pre> DBTier: type: Cloud.Machine properties: name: mysql image: ubuntu-16 flavor: '\${input.size}' constraints: - tag: '\${input.env}' networks: - network: '\${resource["WP-Network-Private"].id}' assignPublicIpAddress: true remoteAccess: authentication: usernamePassword username: '\${input.username}' password: '\${input.userpassword}' cloudConfig: #cloud-config repo_update: true repo_upgrade: all packages: - mysql-server runcmd: - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/ mysqlld.cnf - service mysql restart - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';" - mysql -e "FLUSH PRIVILEGES;" attachedDisks: [] </pre> |
| WebTier 资源 | <pre> WebTier: type: Cloud.Machine properties: name: wordpress flavor: '\${input.size}' </pre> |

| 组件 | 示例 |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <pre> image: ubuntu-16 constraints: - tag: '\${input.env}' networks: - network: '\${resource["WP-Network-Private"].id}' assignPublicIpAddress: true cloudConfig: #cloud-config repo_update: true repo_upgrade: all packages: - apache2 - php - php-mysql - libapache2-mod-php - php-mcrypt - mysql-client runcmd: - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/ latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1 - i=0; while [\$i -le 5]; do mysql --connect-timeout=3 -h \$ {DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break sleep 15; i=\$((i+1)); done - mysql -u root -pmysqlpassword -h \${DBTier.networks[0].address} -e "create database wordpress_blog;" - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/ html/mywordpresssite/wp-config.php - sed -i -e s/"define('DB_NAME', 'database_name_here');"/"define('DB_NAME', 'wordpress_blog');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define('DB_USER', 'username_here');"/"define('DB_USER', 'root');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define('DB_PASSWORD', 'password_here');"/"define('DB_PASSWORD', 'mysqlpassword');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e s/"define('DB_HOST', 'localhost');"/"define('DB_HOST', '\$ {DBTier.networks[0].address}');"/ /var/www/html/mywordpresssite/wp- config.php - service apache2 reload </pre> |

示例：基本蓝图代码已完成的示例

```

inputs:
  env:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
      - 'env:test'
    default: 'env:dev'
  title: Environment
  description: Target Environment
size:
  type: string
  enum:

```



```

    - small
    - medium
    - large
  description: Size of Nodes
  title: Tier Machine Size
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username
userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#&$]+'
  encrypted: true
  title: Database Password
  description: Database Password
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Database Disk Size
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu-16
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    remoteAccess:
      authentication: usernamePassword
      username: '${input.username}'
      password: '${input.userpassword}'
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all

    packages:
      - mysql-server

    runcmd:
      - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
      - service mysql restart
      - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
      - mysql -e "FLUSH PRIVILEGES;"
    attachedDisks: []
  WebTier:

```

```

type: Cloud.Machine
properties:
  name: wordpress
  flavor: '${input.size}'
  image: ubuntu-16
  constraints:
    - tag: '${input.env}'
  networks:
    - network: '${resource["WP-Network-Private"].id}'
      assignPublicIpAddress: true
  cloudConfig: |
    #cloud-config
    repo_update: true
    repo_upgrade: all

  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
    - php-mcrypt
    - mysql-client

  runcmd:
    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
    - i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
${DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php
    - service apache2 reload
WP-Network-Private:
type: Cloud.Network
properties:
  name: WP-Network-Private
  networkType: existing
  constraints:
    - tag: 'type:isolated-net'
    - tag: '${input.env}'

```

后续步骤

通过检查语法并部署蓝图来测试蓝图。

WordPress 用例：测试基本蓝图

在开发过程中，通常从基本组件开始生成 vRealize Automation Cloud Assembly 蓝图，然后随着蓝图的增长进行部署和测试。

为了确保部署按您希望的方式运作，可以多次测试和部署蓝图。您可以逐渐添加更多组件、重新测试和重新部署。

前提条件

创建基本蓝图。请参见 [WordPress 用例：创建基本蓝图](#)。

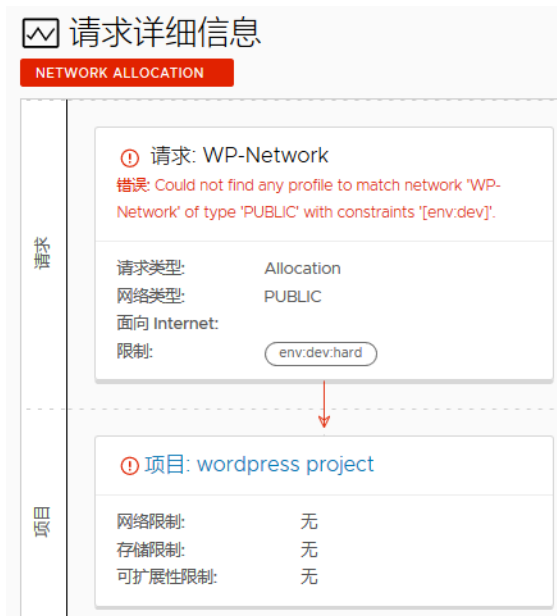
步骤

- 1 单击**蓝图**，然后打开 WordPress 的最佳实践蓝图。
基本蓝图将出现在设计画布和代码编辑器中。
- 2 要检查蓝图语法、放置位置和基本有效性，请单击左下角的**测试**。
- 3 键入输入值，然后单击**测试**。

| | | | |
|----------------------|----------|--------------------|---|
| Environment | env:dev | ▼ | ⓘ |
| Database Tier Size * | small | ▼ | ⓘ |
| Database Username * | ouradmin | | |
| Database Password * | •••••• | | |
| MySQL Data Disk Size | 4 | ⬆️⬆️⬆️⬆️⬆️⬆️⬆️⬆️⬆️ | ⓘ |

测试只是一个模拟，实际上并不部署虚拟机或其他资源。模拟可发现潜在问题，例如，未定义任何与蓝图中的硬性限制相匹配的资源功能。

测试包括指向**置备图**的链接，您可以在其中检查模拟的部署流并查看发生的任何错误。



成功的模拟不保证部署蓝图时不出错。

- 4 在蓝图通过模拟后，单击左下角的**部署**。
- 5 选择**选择新部署**。
- 6 将部署命名为 **WordPress for OurCo**，然后单击**下一步**。
- 7 键入输入值，然后单击**部署**。
- 8 要验证蓝图是否已成功部署，请在**部署**下进行查看。

如果某个部署失败，请单击其名称，然后单击**历史记录**选项卡以查看有助于进行故障排除的消息。

| 时间戳 | 状态 | 资源类型 | 资源名称 | 详细信息 |
|----------------------|------------------------|---------------|----------------------------|-----------------------------------------------------------|
| 2020年1月21日 上午9:41:32 | REQUEST_FINISHED | | | |
| 2020年1月21日 上午9:41:31 | COMPLETION_FINISHED | | | |
| 2020年1月21日 上午9:41:14 | COMPLETION_IN_PROGRESS | | | |
| 2020年1月21日 上午9:40:51 | CREATE_FINISHED | Cloud.Machine | Cloud_vSphere_Machine_1[1] | |
| 2020年1月21日 上午9:33:05 | CREATE_IN_PROGRESS | Cloud.Machine | Cloud_vSphere_Machine_1[1] | Request is in stage STARTED and substage RESOURCE_COUNTED |
| 2020年1月21日 上午9:31:05 | CREATE_IN_PROGRESS | Cloud.Machine | Cloud_vSphere_Machine_1[1] | |

某些历史记录条目的最右侧可能具有**置备图**链接。该图与模拟的图类似，您可以在其中检查置备过程中 vRealize Automation Cloud Assembly 决策点的流程图。

如需查看更多流程图，请单击**基础架构 > 活动 > 请求**。

- 9 要验证应用程序是否正常工作，请在浏览器中打开 **WordPress** 起始页面。
 - a 等待 **WordPress** 服务器完全创建并初始化。
初始化可能需要 30 分钟或更长时间，具体取决于环境。
 - b 要查找站点 FQDN 或 IP 地址，请转到**部署 > 拓扑**。

- c 在画布中，单击“WebTier”，然后在右侧的面板中查找 IP 地址。
- d 输入 IP 地址，作为 WordPress 起始页面完整 URL 的一部分。

在此用例中，完整 URL 是：

`http://{IP-address}/mywordpresssite`

或

`http://{IP-address}/mywordpresssite/wp-admin/install.php`

- 10 在浏览器中检查 WordPress 后，如果需要对应用程序进行其他处理，请进行蓝图更改并使用**更新现有部署**选项进行重新部署。
- 11 考虑对蓝图进行版本控制。如果更改导致部署失败，可以恢复到正常工作的版本。
 - a 在蓝图设计页面中，单击**版本**。
 - b 在“创建版本”页面中，输入 **WP-1.0**。
请勿在版本名称中输入空格。
 - c 单击**创建**。

要复查或恢复到某个版本，请在设计页面中单击**版本历史记录**选项卡。

- 12 基本部署现已准备就绪，可以通过增加应用程序服务器和数据库服务器的 CPU 和内存来尝试首次部署时增强。
将应用程序服务器和数据库服务器更新到中型节点大小。使用同一个蓝图，在部署时选择 **medium**，重新部署并重新验证应用程序。

后续步骤

通过添加更多组件，将蓝图扩展至适用于生产的应用程序。

WordPress 用例：展开蓝图

在创建和测试基本的 vRealize Automation Cloud Assembly 蓝图后，将其扩展为可部署到开发、测试和最终生产环境的多层应用程序。

要扩展蓝图，请添加以下增强功能。

- 用于集群应用程序服务器以增加容量的选项
- 应用程序服务器前面的公用网络和负载均衡器
- 具有存档存储的备份服务器

前提条件

创建基本蓝图并对其进行测试。请参见 [WordPress 用例：创建基本蓝图](#)和 [WordPress 用例：测试基本蓝图](#)。

步骤

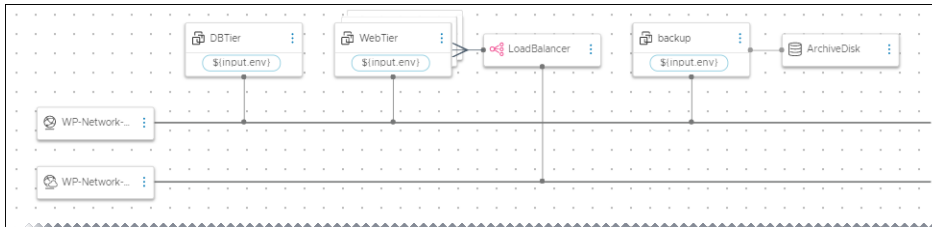
- 1 单击**蓝图**，然后打开 WordPress 的最佳实践蓝图。

基本蓝图将出现在设计画布和代码编辑器中。

- 2 使用代码示例和图作为指导进行添加和更改。

您可以使用 GUI 将新资源拖到画布（如负载均衡器），然后在代码编辑器中完成配置。

- a 添加 count 输入提示，以使 WordPress 应用程序服务器加入集群。
- b 添加云不可知负载均衡器。
- c 将负载均衡器连接到 WordPress 应用程序服务器集群。
- d 添加云不可知的备份计算机。
- e 将备份计算机连接到专用/内部网络。
- f 添加云不可知的公用/外部网络。
- g 将负载均衡器连接到公用网络。
- h 添加一个云不可知的存储卷，用作存档磁盘。
- i 将存档磁盘连接到备份计算机。
- j 为存储磁盘速度添加 archiveusage 输入提示。
- k 为存储磁盘大小添加 archiveDiskSize 输入提示。



- 3 部署、测试和更改的方式与为基本蓝图执行的操作相同。

您可以更新现有部署，甚至部署新实例，以便可以比较部署。

目的是获得可用于生产部署的实体、可重复的蓝图。

示例：已完成扩展的蓝图代码示例

```
inputs:
  env:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
      - 'env:test'
    default: 'env:dev'
    title: Environment
    description: Target Environment
```

```

size:
  type: string
  enum:
    - small
    - medium
    - large
  description: Size of Nodes
  title: Tier Machine Size
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username
userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#&$]+'
  encrypted: true
  title: Database Password
  description: Database Password
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Database Disk Size

count: type: integer default: 2 maximum: 5 minimum: 2 title: WordPress Cluster Size
description: WordPress Cluster Size (Number of Nodes) archiveDiskSize: type: number default:
4 maximum: 10 title: WordPress Archive Disk Size description: Archive Storage Disk Speed
archiveusage: type: string enum: - 'usage:general' - 'usage:fast' description: Archive
Storage Disk Speed title: Archive Disk Speed
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu-16
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    remoteAccess:
      authentication: usernamePassword
      username: '${input.username}'
      password: '${input.userpassword}'
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all

    packages:

```

```

- mysql-server

runcmd:
- sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
- service mysql restart
- mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
- mysql -e "FLUSH PRIVILEGES;"

attachedDisks: []
WebTier:
  type: Cloud.Machine
  properties:
    name: wordpress
    flavor: '${input.size}'
    image: 'ubuntu-16'
    count: '${input.count}'
    constraints:
      - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    storage: disks: - capacityGb: '${input.archiveDiskSize}' name: ArchiveDisk
  cloudConfig: |
    #cloud-config
    repo_update: true
    repo_upgrade: all

  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
    - php-mcrypt
    - mysql-client

  runcmd:
    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
    - i=0; while [ $i -le 10 ]; do mysql --connect-timeout=3 -h $
${DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php

```



```

- service apache2 reload

LoadBalancer: type: Cloud.LoadBalancer properties: name: myapp-lb network: '${resource["WP-
Network-Public"].id}' instances: - '${WebTier.id}' routes: - protocol: HTTP port: '80'
instanceProtocol: HTTP instancePort: '80' healthCheckConfiguration: protocol: HTTP port: '80'
urlPath: /mywordpresssite/wp-admin/install.php intervalSeconds: 6 timeoutSeconds: 5
unhealthyThreshold: 2 healthyThreshold: 2 internetFacing: true
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
    constraints:
      - tag: 'type:isolated-net'
      - tag: '${input.env}'

WP-Network-Public: type: Cloud.Network properties: name: WP-Network-Public networkType:
public constraints: - tag: 'type:public-net' - tag: '${input.env}' backup: type:
Cloud.Machine properties: name: backup flavor: '${input.size}' image: 'ubuntu-16' networks: -
network: '${resource["WP-Network-Private"].id}' constraints: - tag: '${input.env}'
attachedDisks: - source: '${ArchiveDisk.id}' ArchiveDisk: type: Cloud.Volume properties:
name: ArchiveDisk capacityGb: 5 constraints: - tag: '${input.archiveusage}' - tag: '$
{input.env}'

```

后续步骤

定义您自己的基础架构并创建自己的蓝图。

请参见第 4 章 [构建您的 vRealize Automation Cloud Assembly 资源基础架构](#) 和第 6 章 [设计 vRealize Automation Cloud Assembly 部署](#)。

VMware Cloud on AWS 用例

此 vRealize Automation Cloud Assembly 用例显示了定义资源基础架构和蓝图设置以部署到 VMware Cloud on AWS 环境的过程。

此过程要求云管理员已按照 [VMware Cloud on AWS 入门指南文档](#) 中的部署和管理软件定义的数据中心中所述配置组织的 VMware Cloud on AWS SDDC 数据中心。

请查看顺序设置，了解为 VMware Cloud on AWS 配置环境的过程。请注意，用例中的值仅用作示例。请考虑在何处需要替换为您自己的值或从示例值外插值，以便满足您自己的云计算基础架构需求和部署需求。

如何为 [Cloud Assembly 配置 VMware Cloud on AWS](#) 的 VMware 云计算管理技术营销中提供了类似工作流的详细视频。

步骤

1 在 vRealize Automation Cloud Assembly 中配置基本 VMware Cloud on AWS 工作流

此 vRealize Automation Cloud Assembly 用例展示了定义资源基础架构的过程以及部署到 VMware Cloud on AWS 环境的相应蓝图。

2 在 vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流中配置隔离网络

在此过程中，您将在 vRealize Automation Cloud Assembly 中为 VMware Cloud on AWS 部署添加一个隔离网络。

在 vRealize Automation Cloud Assembly 中配置基本 VMware Cloud on AWS 工作流

此 vRealize Automation Cloud Assembly 用例展示了定义资源基础架构的过程以及部署到 VMware Cloud on AWS 环境的相应蓝图。

在此过程中，您将配置支持将蓝图部署到现有 VMware Cloud on AWS 环境中的资源的基础架构。

前提条件

- 在 vRealize Automation Cloud Assembly 中创建和配置 VMware Cloud on AWS 云帐户之前，您必须属于现有 VMware Cloud on AWS SDDC 环境中的某个组织。有关配置 VMware Cloud on AWS 服务的信息，请参见 [VMware Cloud on AWS 文档](#)。
- 为了方便在 vCenter 中的现有 VMware Cloud on AWS 主机 SDDC 与 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 云帐户之间建立所需连接，您必须使用 VPN 或类似的网络连接方式提供网络连接并添加防火墙规则。请参见 [准备 VMware Cloud on AWS SDDC 连接到 vRealize Automation 中的 VMware Cloud on AWS 云帐户](#)。

步骤

1 准备 VMware Cloud on AWS SDDC 连接到 vRealize Automation 中的 VMware Cloud on AWS 云帐户

在 vRealize Automation Cloud Assembly 内部部署环境中使用 VMware Cloud on AWS 云帐户时，您必须创建网络连接才能支持 vCenter 中的 SDDC 和 vRealize Automation 中的任何 VMware Cloud on AWS 云帐户之间的通信。

2 在 vRealize Automation 的示例工作流中创建 VMware Cloud on AWS 云帐户

在此步骤中，将在 vRealize Automation 中创建一个 VMware Cloud on AWS 云帐户。

3 为 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署创建云区域

在此步骤中，您将创建一个云区域，以指定 CloudAdmin 用户在 vRealize Automation Cloud Assembly 中使用 VMware Cloud on AWS 时可以访问的计算资源。

4 为 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署配置网络和存储配置文件

在此步骤中，您将配置网络配置文件和存储配置文件，以便指定可供 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS CloudAdmin 用户使用的资源。

5 在 vRealize Automation Cloud Assembly 中创建一个项目以支持 VMware Cloud on AWS 部署

在此步骤中，您将定义一个 vRealize Automation Cloud Assembly 项目，该项目可用于控制 VMware Cloud on AWS 部署可用的资源。

6 在蓝图设计中定义 vCenter 计算机资源以支持 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署

在此步骤中，您将 vCenter 计算机资源拖动到设计画布上，并为 VMware Cloud on AWS 部署添加设置。

准备 VMware Cloud on AWS SDDC 连接到 vRealize Automation 中的 VMware Cloud on AWS 云帐户

在 vRealize Automation Cloud Assembly 内部部署环境中使用 VMware Cloud on AWS 云帐户时，您必须创建网络连接才能支持 vCenter 中的 SDDC 和 vRealize Automation 中的任何 VMware Cloud on AWS 云帐户之间的通信。

为了方便在 vCenter 中的现有 VMware Cloud on AWS 主机 SDDC 与 vRealize Automation 中的 VMware Cloud on AWS 云帐户之间建立所需连接，您必须使用 VPN 或类似的网络连接方式在这两个元素之间提供网络连接。

步骤

- 1 通过公用 Internet 或 AWS Direct Connect 配置 VPN 连接。

请参见 [VMware Cloud on AWS 文档](#) 中的《VMware Cloud on AWS 网络和安全》。

- 2 确认 vCenter Server FQDN 可在管理网络上的专用 IP 地址处解析。

请参见 [VMware Cloud on AWS 文档](#) 中的《VMware Cloud on AWS 网络和安全》。

- 3 配置所需的防火墙规则。

您必须在 SDDC 的 VMware Cloud on AWS 控制台中配置管理网关防火墙规则，以支持通信。这些规则必须位于**管理网关**防火墙规则区域中。通过使用 SDDC 控制台中**网络和安全**选项卡上的选项创建防火墙规则。

- 将使用 HTTPS (TCP 443) 服务的 ESXi 网络流量限制到 vRealize Automation 设备/服务器或 vRealize Automation 负载均衡器 VIP 的已发现 IP 地址。
- 将使用 ICMP（全部 ICMP）、SSO (TCP 7444) 和 HTTPS (TCP 443) 服务的 vCenter 网络流量限制到 vRealize Automation 设备/服务器或 vRealize Automation 负载均衡器 VIP 的已发现 IP 地址。
- 将使用 HTTPS (TCP 443) 服务的 NSX-T Manager 网络流量限制到 vRealize Automation 设备/服务器或 vRealize Automation 负载均衡器 VIP 的已发现 IP 地址。

下表汇总了所需的防火墙规则。

表 3-1. 必需的管理网关防火墙规则摘要

| 名称 | 源 | 目标 | 服务 |
|--------------|------------------|-------------|---------------|
| vCenter | 内部部署数据中心的 CIDR 块 | vCenter | 任意（所有流量） |
| vCenter ping | 任意 | vCenter | ICMP（所有 ICMP） |
| NSX Manager | 内部部署数据中心的 CIDR 块 | NSX Manager | 任意（所有流量） |

表 3-1. 必需的管理网关防火墙规则摘要（续）

| 名称 | 源 | 目标 | 服务 |
|---------------------|-------------------|-------------------|---------------|
| 内部部署到 ESXi ping | 内部部署数据中心的 CIDR 块 | 仅 ESXi 管理 | ICMP（所有 ICMP） |
| 内部部署到 ESXi 远程控制台和置备 | 内部部署数据中心的 CIDR 块 | 仅 ESXi 管理 | TCP 902 |
| 内部部署到 SDDC 虚拟机 | 内部部署数据中心的 CIDR 块 | SDDC 逻辑网络的 CIDR 块 | 任意（所有流量） |
| SDDC 虚拟机到内部部署 | SDDC 逻辑网络的 CIDR 块 | 内部部署数据中心的 CIDR 块 | 任意（所有流量） |

有关相关信息，请参见 [VMware Cloud on AWS 文档](#) 中的《VMware Cloud on AWS 网络和安全》和《VMware Cloud on AWS 操作指南》。

结果

配置所需的网关访问和防火墙规则后，可以继续创建 VMware Cloud on AWS 云帐户的过程。

在 vRealize Automation 的示例工作流程中创建 VMware Cloud on AWS 云帐户

在此步骤中，将在 vRealize Automation 中创建一个 VMware Cloud on AWS 云帐户。

有关相关信息，请参见 [VMware Cloud on AWS 文档](#)。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流程。

前提条件

- 此过程假设您具有所需的管理员凭据，包括 vCenter 中目标 SDDC 的 VMware Cloud on AWS CloudAdmin 凭据，并假设您已在端口 443 上启用 HTTPS 访问。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 此过程假设您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 为了方便在 vCenter 中的现有 VMware Cloud on AWS 主机 SDDC 与 vRealize Automation 中的 VMware Cloud on AWS 云帐户之间建立所需连接，您必须使用 VPN 或类似的网络连接方式提供网络连接和防火墙规则。请参见 [准备 VMware Cloud on AWS SDDC 连接到 vRealize Automation 中的 VMware Cloud on AWS 云帐户](#)。如果您使用的是外部 HTTP Internet 代理，则必须针对 IPv4 对其进行配置。
- 如果您没有外部 Internet 访问，请配置一个 Internet 服务器代理。请参见 [如何配置 vRealize Automation 的 Internet 代理服务器](#)。

步骤

- 1 选择**基础架构 > 连接 > 云帐户**。
- 2 单击**添加云帐户**，选择“VMware Cloud on AWS”，然后输入值。

下表提供了示例值和支持信息。

| 设置 | 示例值和说明 | 说明 |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VMC API 令牌 | <ol style="list-style-type: none"> 单击 VMC API 令牌 行末端的 i 帮助图标，然后在帮助文本框中单击 API 令牌 页面，以打开您组织的 我的帐户 页面上的 API 令牌 选项卡。 单击 生成令牌 以显示 生成新的 API 令牌 选项。 输入新令牌名称，例如 myinitials_mytoken。 将 令牌 TTL 设置为 永不过期。 如果创建设置为过期的令牌，则在令牌过期时，vRealize Automation 中的 VMware Cloud on AWS 操作将停止工作，并在您使用新令牌更新云帐户之前，这些操作将保持不工作状态。 在 定义范围 部分中，选择 所有角色。  单击 生成。 在生成的令牌页面中，单击 复制，然后单击 继续。 返回 新建云帐户 页面，将复制的令牌粘贴到 VMC API 令牌 行，然后单击 应用 API 令牌。  | <p>可以在链接的 API 令牌 页面上为您的组织创建新令牌或使用现有令牌。</p> <p>在 定义范围 部分中，API 令牌要求的最小角色为：</p> <ul style="list-style-type: none"> ■ 组织角色 <ul style="list-style-type: none"> ■ 组织成员 ■ 组织所有者 ■ 服务角色 - VMware Cloud on AWS <ul style="list-style-type: none"> ■ 管理员 ■ NSX Cloud 管理员 ■ NSX Cloud 审核员 <p>注 复制、下载或打印生成的令牌。离开此页面后，将无法检索生成的令牌。</p> <p>应用生成的令牌或提供的令牌连接到组织 VMware Cloud on AWS 订阅中的可用 SDDC 环境，并填充 SDDC 名称列表。</p> <p>如果 vRealize Automation 和 VMware Cloud on AWS 服务位于不同组织中，应切换到 VMware Cloud on AWS 组织，然后生成令牌。</p> <p>有关 API 令牌的更多信息，请参见 生成 API 令牌。</p> |
| SDDC 名称 | <p>对于此示例，请选择 Datacenter:Datacenter-abz。</p> <p>有效的 SDDC 名称将自动填充 vCenter 和 NSX-T FQDN 条目。如果云代理已部署到该 SDDC，则云代理值也会自动填充。</p> | <p>从 VMware Cloud on AWS 订阅中的可用 SDDC 列表中进行选择。SDDC 列表基于 VMware Cloud on AWS API 令牌。</p> <p>NSX-V SDDC 不受 vRealize Automation 支持，因此不显示在可用 SDDC 列表中。</p> |
| vCenter IP 地址/ FQDN | 地址会根据选择的 SDDC 自动填充。 | <p>输入指定 SDDC 中的 vCenter Server 的 IP 地址或 FQDN。</p> <p>IP 地址默认为专用 IP 地址。根据用于访问 SDDC 的网络连接类型，默认地址可能不同于指定 SDDC 中 NSX Manager Server 的 IP 地址。</p> |

| 设置 | 示例值和说明 | 说明 |
|------------------------|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NSX Manager IP 地址/FQDN | 地址会根据选择的 SDDC 自动填充。 | 确定指定 SDDC 中的 NSX Manager 的 IP 地址或 FQDN。 IP 地址默认为专用 IP 地址。根据用于访问 SDDC 的网络连接类型，默认地址可能不同于指定 SDDC 中 NSX Manager Server 的 IP 地址。 VMware Cloud on AWS 云帐户支持 NSX-T。 |
| vCenter 用户名和密码 | 系统会自动填充用户名 cloudadmin@vmc.local。 | 如果与默认用户名不同，则输入指定 SDDC 的 vCenter 用户名。 指定的用户需要 CloudAdmin 凭据。用户不需要 CloudGlobalAdmin 凭据。 输入用户密码。 |
| 验证 | 单击 验证 。 | 验证将确认您对指定 vCenter 的访问权限，并检查 vCenter 是否正在运行。 |
| 名称和说明 | 输入 OurCo-VMC 作为云帐户名称。 输入 Sample deployment for VMC 作为云帐户说明。 | |
| 允许置备到这些数据中心 | 此信息为只读。 | 列出指定 VMware Cloud on AWS SDDC 环境中的可用数据中心。 |
| 创建云区域 | 取消选中该复选框。对于此示例，将稍后在工作流中创建一个云区域。 | 请参见 了解有关 vRealize Automation Cloud Assembly 云区域的更多信息 。 |
| 能力标记 | 将此设置留空。此工作流不使用功能标记。 | 请根据您的组织的标记策略使用标记。请参见 如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署和创建标记策略 。 |

3 单击**添加**。

结果

将从 VMware Cloud on AWS SDDC 数据中心收集计算机和卷等资源的数据，并列在 vRealize Automation **基础架构**选项卡的**资源**部分中。

后续步骤

为 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署创建云区域。

为 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署创建云区域

在此步骤中，您将创建一个云区域，以指定 CloudAdmin 用户在 vRealize Automation Cloud Assembly 中使用 VMware Cloud on AWS 时可以访问的计算资源。

在 VMware Cloud on AWS 中，两个主要管理员凭据是 CloudGlobalAdmin 和 CloudAdmin。vRealize Automation Cloud Assembly 设计为支持 CloudAdmin 用户。部署到可供 VMware Cloud on AWS CloudAdmin 用户使用的资源。不要部署到需要 VMware Cloud on AWS CloudGlobalAdmin 凭据的资源。

云区域标识项目蓝图在其上部署计算机、网络和存储的计算资源。请参见[了解有关 vRealize Automation Cloud Assembly 云区域的更多信息](#)。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流。

前提条件

- 完成在 [vRealize Automation](#) 的示例工作流中创建 [VMware Cloud on AWS](#) 云帐户过程。
- 此过程假设您具有所需的管理员凭据，包括 vCenter 中目标 SDDC 的 VMware Cloud on AWS CloudAdmin 凭据。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。
- 此过程假设您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。

步骤

- 1 选择**基础架构 > 配置 > 云区域**。
- 2 单击**新建云区域**，并输入 VMware Cloud on AWS 环境的值。

| 设置 | 示例值 |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| 帐户/区域 | OurCo-VMC / Datacenter:Datacenter-abz 这是您在上一步在 vRealize Automation 的示例工作流中创建 VMware Cloud on AWS 云帐户中定义的云帐户和关联区域。 |
| 名称 | VMC_cloud_zone-1 |
| 说明 | 仅 VMware Cloud on AWS 资源 |
| 放置策略 | 默认 |
| 能力标记 | 将此设置留空。此工作流不使用功能标记。 |

- 3 单击**计算**选项卡。
- 4 如下面的区域 1 中所示，查找并选择可供 CloudAdmin 用户使用的计算资源。对于此示例，使用名为 Cluster 1/ Compute-ResourcePool 的资源。

Cluster 1/ Compute-ResourcePool 是 VMware Cloud on AWS 的默认计算资源。



- 5 如上面的区域 2 中所示，添加标记名称 `vmc_placements_abz`。

标记

已选择 1 个对象

添加标记

vmc_placements_abz X

输入新标记

移除标记

无标记 ⓘ

- 6 通过在筛选标记部分中输入 `vmc_placements_abz` 来筛选在此云区域中使用的计算资源。

- 7 单击保存。

| 名称 | 帐户/区域 | 类型 | 标记 |
|---------------------------------------------------------|------------------------------------------------------|----------------------|-------------------------|
| <input type="checkbox"/> ComputeClusterA | LK-TEST 测试表 A 中在中心部署的 Ubuntu / NSX62-Scale-DC | common.title.cluster | Cluster:ComputeClusterA |
| <input checked="" type="checkbox"/> ComputeClusterA-New | nsx-v 测试表 A 中在中心部署的 Ubuntu / NSX62-DataCenter | common.title.cluster | ComputeClusterA |
| <input type="checkbox"/> ComputeClusterA / Scale | 270_VC_account 测试表 A 中在中心部署的 Ubuntu / NSX62-Scale-DC | ResourcePool | ComputeClusterA |

对于此示例，只有名为 `Cluster 1/ Compute-ResourcePool` 的计算资源可供 `CloudAdmin` 用户使用。

后续步骤

为 [vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署配置网络和存储配置文件](#)。

为 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署配置网络和存储配置文件

在此步骤中，您将配置网络配置文件和存储配置文件，以便指定可供 `vRealize Automation Cloud Assembly` 中的 `VMware Cloud on AWS CloudAdmin` 用户使用的资源。

虽然还需要映像和特定实例值，但是它们对于 `VMware Cloud on AWS` 用户凭据没有任何独特之处。对于此示例，在定义蓝图时，您将使用特定实例值 `small` 和映像值 `ubuntu-16`。

有关映射和配置文件的一般信息，请参见第 4 章 [构建您的 vRealize Automation Cloud Assembly 资源基础架构](#)。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流。

前提条件

- 创建云区域。请参见为 [vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署创建云区域](#)。
- 此过程假设您具有所需的管理员凭据，包括 `vCenter` 中目标 `SDDC` 的 `VMware Cloud on AWS CloudAdmin` 凭据。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。

- 此过程假设您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。

步骤

1 为 VMware Cloud on AWS 部署定义网络配置文件。

- 选择**基础架构 > 配置 > 网络配置文件**，然后单击**新建网络配置文件**。

| 设置 | 示例值 |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 帐户/区域 | OurCo-VMC / Datacenter:Datacenter-abz 注 选择在 在 vRealize Automation 的示例工作流中创建 VMware Cloud on AWS 云帐户 中创建的 VMware Cloud on AWS 云帐户及其匹配的 SDDC 数据中心。 |
| 名称 | vmc-network1 |
| 说明 | 包含拥有 VMware Cloud on AWS CloudAdmin 凭据的蓝图管理员可以访问的网络。 |

- 单击**网络**选项卡，然后单击**添加网络**。
- 选择拥有 CloudAdmin 凭据的 VMware Cloud on AWS 用户可以部署到的网络，例如 sddc-cgw-network-1。

添加网络



| <input type="checkbox"/> | 名称 | 帐户/区域 | 区域 | 网络ID |
|-------------------------------------|---------------------|----------------------------------------------------------------|----|--------------------|
| <input checked="" type="checkbox"/> | ESO_PKS_VC01_VM_PKS | 1114VCあア7中表踏@墨 ホ CeÜBäãæðeoàùñUse / ESO_PKS_VC01_DC01 | | ESO_PKS_VC01_DVS01 |
| <input type="checkbox"/> | ESO_PKS_VC01_Mgmt | 1114VCあア7中表踏@墨 ホ CeÜBäãæðeoàùñUse / ESO_PKS_VC01_DC01 | | ESO_PKS_VC01_DVS01 |

2 保存网络配置文件。

3 为 VMware Cloud on AWS 部署定义存储配置文件。

针对 CloudAdmin 用户可访问的数据存储/群集配置一个存储配置文件。

- a 选择 **基础架构 > 配置 > 存储配置文件**，然后单击 **新建存储配置文件**。

| 设置 | 示例值 |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 帐户/区域 | OurCo-VMC / Datacenter:Datacenter-abz 选择在 vRealize Automation 的示例工作流程中创建 VMware Cloud on AWS 云帐户 中创建的 VMware Cloud on AWS 云帐户及其匹配的 SDDC 数据中心。 |
| 名称 | vmc-storage1 |
| 说明 | 包含拥有 VMware Cloud on AWS CloudAdmin 凭据的蓝图管理员可以部署到的数据存储群集。 |

- b 从 **数据存储/群集** 下拉菜单中，选择 **WorkloadDatastore** 数据存储。



对于 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS，存储策略必须使用 **WorkloadDatastore** 数据存储来支持 VMware Cloud on AWS 部署。

4 保存存储配置文件。

后续步骤

在 [vRealize Automation Cloud Assembly](#) 中创建一个项目以支持 **VMware Cloud on AWS** 部署。

在 vRealize Automation Cloud Assembly 中创建一个项目以支持 VMware Cloud on AWS 部署

在此步骤中，您将定义一个 vRealize Automation Cloud Assembly 项目，该项目可用于控制 VMware Cloud on AWS 部署可用的资源。

有关项目的信息，请参见 [vRealize Automation Cloud Assembly](#) 项目在部署时的工作方式。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流程。

前提条件

- 完成为 [vRealize Automation Cloud Assembly](#) 中的 **VMware Cloud on AWS** 部署配置网络和存储配置文件过程。
- 此过程假设您具有所需的管理员凭据，包括 vCenter 中目标 SDDC 的 VMware Cloud on AWS CloudAdmin 凭据。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。
- 此过程假设您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly](#) 用户角色是什么。

步骤

- 1 选择**基础架构 > 配置 > 项目**。
- 2 单击**新建项目**，然后输入项目名称 VMC_proj-1_abz。
- 3 单击**用户**，然后单击**添加用户**。

用户需要 CloudAdmin 凭据，才能访问其组织的 VMware Cloud on AWS 订阅。

- chris.gray@ourco.com，管理员
- kerry.white@ourco.com，成员

- 4 单击**置备**，然后单击**添加云区域**。
- 5 添加在前面的步骤中配置的云区域。

| 设置 | 示例值 |
|-------|---------------------------------------------------------------------------------------------------------------------------|
| 云区域 | VMC_cloud_zone-1 您在前面的步骤（为 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署创建云区域）中创建了此云区域。 |
| 置备优先级 | 1 |
| 实例限制 | 3 |

- 6 对于此示例，请忽略其他选项。

后续步骤

创建要在 VMware Cloud on AWS 环境中部署的蓝图。请参见在[蓝图设计中定义 vCenter 计算机资源以支持 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署](#)。

在蓝图设计中定义 vCenter 计算机资源以支持 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署

在此步骤中，您将 vCenter 计算机资源拖动到设计画布上，并为 VMware Cloud on AWS 部署添加设置。

创建一个可以将其部署到可用 VMware Cloud on AWS 资源的蓝图设计。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流。

前提条件

- 此过程假定您拥有蓝图设计器凭据。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 此过程假定您拥有 vCenter 中目标 SDDC (Datacenter:Datacenter-abz) 的 VMware Cloud on AWS CloudAdmin 凭据。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 按照前几节所述，配置资源基础架构和项目。

步骤

- 1 单击**设计**选项卡，然后单击**新建**。

| 设置 | 示例值 |
|----|---------------------------------------------------------------------------------------------------|
| 名称 | vmc-bp_abz |
| 说明 | 1 |
| 项目 | VMC_proj-1_abz 这是之前创建的项目，可支持您之前创建的云区域。该项目现在与云区域相关联，而云区域又与您之前创建的 VMware Cloud on AWS 云帐户/区域相关联。 |

- 2 将 vSphere 计算机资源拖动到画布上。
- 3 编辑计算机资源中的以下（粗体）蓝图资源代码。

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: ubuntu-1604
      cpuCount: 1
      totalMemoryMB: 1024
      folderName: Workloads
```

image 可以是适合您的部署需求的任何值。

您必须将 `folderName: Workloads` 语句添加到蓝图设计代码中才能支持 VMware Cloud on AWS 部署。`folderName: Workloads` 设置支持 VMware Cloud on AWS SDDC 环境中的 CloudAdmin 凭据，因此为必需项。

注意：虽然上述代码示例中显示的 `folderName: Workloads` 设置为必需项，但可以直接在蓝图设计代码中添加该设置（如上所示），也可以在关联的云区域或项目中添加。如果在这三个位置中的多个位置指定了该设置，则优先级如下所示：

- 项目设置覆盖蓝图设计设置和云区域设置。
- 蓝图设计设置覆盖云区域设置。

注意：您可以选择将 `cpuCount` 和 `totalMemoryMB` 设置替换为 `flavor`（调整大小）条目，如下所示：

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
```

```
properties:
  image: ubuntu-1604
  flavor: small
  folderName: Workloads
```

如果云区域的文件夹值设置为 **Workloads**，则无需在蓝图设计中设置 `folderName` 属性，除非您要替代云区域的文件夹值。

后续步骤

通过添加网络隔离，扩展此基本 VMware Cloud on AWS 工作流。请参见在 [vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流中配置隔离网络](#)。

在 vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流中配置隔离网络

在此过程中，您将在 vRealize Automation Cloud Assembly 中为 VMware Cloud on AWS 部署添加一个隔离网络。

定义您的 VMware Cloud on AWS 云帐户时，可以使用您在 VMware Cloud on AWS 服务中配置的 NSX-T 设置。有关在 VMware Cloud on AWS 服务中配置 NSX-T 设置的信息，请参见 [VMware Cloud on AWS 产品文档](#)。

vRealize Automation Cloud Assembly 支持带有 NSX-T 的 VMware Cloud on AWS。它不支持带有 NSX-V 的 VMware Cloud on AWS。

vRealize Automation Cloud Assembly 支持 VMware Cloud on AWS 部署的网络隔离。它不支持 VMware Cloud on AWS 的其他网络方法。

基本 VMware Cloud on AWS 工作流的这一扩展描述了以下创建隔离网络供在 vRealize Automation Cloud Assembly 蓝图中使用的方法：

- 配置基于网络的按需隔离。
- 配置基于安全组的按需隔离。

前提条件

此过程将扩展基本 VMware Cloud on AWS 工作流。它使用与您在 [VMware Cloud on AWS 用例](#) 工作流中配置的同云帐户和区域、云区域、项目和网络配置文件。

步骤

- 1 在 [vRealize Automation Cloud Assembly 中为 VMware Cloud on AWS 部署定义隔离网络](#)
可以使用以下任一过程为 VMware Cloud on AWS 部署配置网络隔离：
- 2 在蓝图中定义网络组件以支持 [vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的网络隔离](#)

在此步骤中，您将网络计算机组件拖动到 vRealize Automation Cloud Assembly 蓝图画布上，并将隔离网络部署的设置添加到目标 VMware Cloud on AWS 环境中。

在 vRealize Automation Cloud Assembly 中为 VMware Cloud on AWS 部署定义隔离网络

可以使用以下任一过程为 VMware Cloud on AWS 部署配置网络隔离：

- 在 vRealize Automation Cloud Assembly 中配置基于按需网络的隔离
- 在 vRealize Automation Cloud Assembly 中配置基于按需安全组的隔离

在 vRealize Automation Cloud Assembly 中配置基于按需网络的隔离

通过在 vRealize Automation Cloud Assembly 网络配置文件中指定和使用按需网络设置，您可以根据 VMware Cloud on AWS 部署需求配置网络隔离。

您可以使用安全组或按需网络设置来指定隔离网络。在此示例中，您可以通过在网络配置文件中指定按需网络设置来配置网络隔离。稍后，您可以在蓝图中访问网络，并在 VMware Cloud on AWS 部署中使用该蓝图。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流。

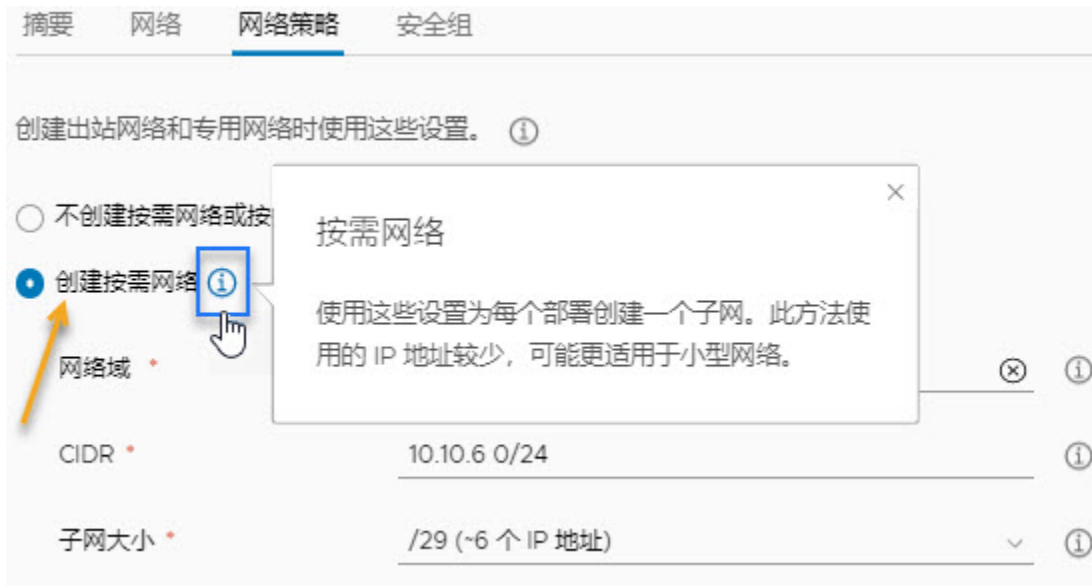
前提条件

- 完成在 vRealize Automation Cloud Assembly 中配置基本 VMware Cloud on AWS 工作流工作流。
- 请参见在 vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流中配置隔离网络。
- 此过程假设您具有所需的管理员凭据，包括 vCenter 中目标 SDDC 的 VMware Cloud on AWS CloudAdmin 凭据。请参见在 vRealize Automation 中使用云帐户所需的凭据。
- 此过程假设您具有云管理员用户角色。请参见 vRealize Automation Cloud Assembly 用户角色是什么。

步骤

- 1 打开在基本 VMware Cloud on AWS 工作流中使用的网络配置文件，例如 vmc-network1。请参见为 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署配置网络和存储配置文件。
- 2 您不需要在**网络**选项卡上进行任何选择。
- 3 单击**网络策略**选项卡。

- 4 选择**创建按需网络**选项，然后选择默认的网络域。指定适当的 CIDR 和子网大小。



- 5 单击**保存**。

使用此网络配置文件时，计算机将部署到默认网络域中的网络。通过使用专用或出站网络访问，将该网络与其他网络隔离。

后续步骤

在蓝图中配置网络组件。请参见在蓝图中定义网络组件以支持 [vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的网络隔离](#)

在 vRealize Automation Cloud Assembly 中配置基于按需安全组的隔离

通过在 vRealize Automation Cloud Assembly 网络配置文件中指定和使用按需安全组，您可以根据 VMware Cloud on AWS 部署需求配置网络隔离。

您可以使用安全组或按需网络设置来指定隔离网络。在此示例中，您可以通过在网络配置文件中指定按需安全组来配置网络隔离。稍后，您可以在蓝图中指定网络，并在 VMware Cloud on AWS 部署中使用该蓝图。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流。

前提条件

- 完成在 [vRealize Automation Cloud Assembly 中配置基本 VMware Cloud on AWS 工作流](#) 工作流。
- 请参见在 [vRealize Automation Cloud Assembly 的 VMware Cloud on AWS 工作流中配置隔离网络](#)。
- 此过程假设您具有所需的管理员凭据，包括 vCenter 中目标 SDDC 的 VMware Cloud on AWS CloudAdmin 凭据。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。

- 此过程假设您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。

步骤

- 1 打开在基本 VMware Cloud on AWS 工作流中使用的网络配置文件，例如 `vmc-network1`。请参见 [vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署配置网络和存储配置文件](#)。
- 2 选择在基本 VMware Cloud on AWS 工作流中使用的现有网络，例如 `sddc-cgw-network-1`。请参见 [vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署配置网络和存储配置文件](#)。
- 3 单击**网络策略**选项卡。
- 4 选择**创建按需安全组**选项。



- 5 单击**保存**。

使用此网络配置文件时，计算机将部署到选定的网络，并由新的安全组策略进行隔离。新的安全策略允许专用或出站网络访问。

后续步骤

在蓝图中配置网络组件。请参见[在蓝图中定义网络组件以支持 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的网络隔离](#)

在蓝图中定义网络组件以支持 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 的网络隔离

在此步骤中，您将网络计算机组件拖动到 vRealize Automation Cloud Assembly 蓝图画布上，并将隔离网络部署的设置添加到目标 VMware Cloud on AWS 环境中。

将网络隔离添加到您之前创建的蓝图中。该蓝图已经与支持部署到您的 VMware Cloud on AWS 环境的项目和云区域以及您为隔离配置的网络配置文件和网络相关联。

除非另有说明，否则在此过程中输入的步骤值仅适用于此示例工作流。

前提条件

- 完成在 [vRealize Automation Cloud Assembly](#) 中配置基于按需安全组的隔离或在 [vRealize Automation Cloud Assembly](#) 中配置基于按需网络的隔离过程。
- 此过程假定您拥有蓝图设计器凭据。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 此过程假定您拥有适用于 vCenter 中的目标 SDDC 的 VMware Cloud on AWS CloudAdmin 凭据。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。

步骤

- 1 打开您在上一个工作流中创建的蓝图。请参见在[蓝图设计中定义 vCenter 计算机资源以支持 vRealize Automation Cloud Assembly 中的 VMware Cloud on AWS 部署](#)。
- 2 从蓝图设计页面左侧的组件中，将网络组件拖动到画布上。
- 3 编辑网络组件 YAML 代码以指定网络类型 `private` 或 `outbound`，如粗体所示。

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: private
```

或

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: outbound
```

后续步骤

您已准备好部署或关闭蓝图。

提供程序特定的外部 IPAM 集成用例

您可以使用外部 IPAM 提供程序管理蓝图部署的 IP 地址分配。此示例过程介绍了如何在 [vRealize Automation Cloud Assembly](#) 中配置外部 IPAM 集成以及如何从该外部 IPAM 提供程序部署蓝图以获取其 IP 地址分配。此示例介绍了如何使用 Infoblox 作为外部 IPAM 提供程序配置 IPAM 集成。

在此过程中，将使用现有 IPAM 提供程序软件包（在本例中为 Infoblox 软件包）和现有运行环境构建提供程序特定的 IPAM 集成点。将配置现有网络并创建网络配置文件，以支持从外部 IPAM 提供程序分配 IP 地址。最后，将创建与网络和网络配置文件匹配的蓝图，并使用从外部 IPAM 提供程序获取的 IP 值部署联网计算机。

有关如何获取和配置 IPAM 提供程序软件包的信息以及如何配置访问云可扩展性代理的运行环境以支持 IPAM 提供程序集成的信息，作为参考信息包括在内。

请记住，显示的值为例值。不能在您的环境中逐字使用这些值。

请考虑在何处替换为您自己的值或根据示例值进行推断，以满足您组织的需求。

步骤

1 在 Infoblox 应用程序中添加所需的可扩展属性以与 vRealize Automation 集成

您必须先要在 Infoblox 中添加所需的可扩展性属性，然后才能从 Infoblox 网站或 VMware Marketplace 下载 Infoblox 提供程序软件包 (infoblox.zip)，并进行部署以与 vRealize Automation 集成。

2 下载并部署外部 IPAM 提供程序软件包以在 vRealize Automation Cloud Assembly 中使用

在 vRealize Automation Cloud Assembly 中定义外部 IPAM 集成点之前，您需要一个已配置的 IPAM 提供程序软件包。您可以从 IPAM 提供程序的网站或 vRealize Automation Cloud Assembly 商城下载提供程序软件包。

3 在 vRealize Automation 中为 IPAM 集成点创建运行环境

在 vRealize Automation 中定义外部 IPAM 集成点之前，需要创建或访问现有的运行环境以作为 IPAM 提供程序和 vRealize Automation 之间的中介。运行环境通常是 Amazon Web Services 或 Microsoft Azure 云帐户，或与云可扩展性代理相关联的基于操作的可扩展性内部部署集成点。

4 在 vRealize Automation 中添加外部 IPAM 集成点

vRealize Automation 支持与外部 IPAM 提供程序集成。可以使用提供程序特定的 IPAM 集成点获取并管理蓝图部署的 IP 地址和相关网络特性。

5 在 vRealize Automation Cloud Assembly 中配置网络和网络配置文件以使用 IPAM 提供程序值

可以将网络定义为使用从外部 IPAM 提供程序（而不是从 vRealize Automation Cloud Assembly 内部）获取并由其管理的 IP 地址值。

6 定义并部署使用 IPAM 提供程序范围分配的 vRealize Automation Cloud Assembly 蓝图

可以将蓝图定义为从外部 IPAM 提供程序获取 IP 地址分配并进行管理。

7 对 vRealize Automation 中的 IPAM 集成使用特定于 Infoblox 的属性和可扩展属性

对于包含 Infoblox 的外部 IPAM 集成的 vRealize Automation 项目，可以使用特定于 Infoblox 的属性。

在 Infoblox 应用程序中添加所需的可扩展属性以与 vRealize Automation 集成

您必须先要在 Infoblox 中添加所需的可扩展性属性，然后才能从 Infoblox 网站或 VMware Marketplace 下载 Infoblox 提供程序软件包 (infoblox.zip)，并进行部署以与 vRealize Automation 集成。

如果要为 Infoblox 与 vRealize Automation Cloud Assembly 集成创建外部 IPAM 集成点时，则此过程适用。

您必须使用您组织帐户的管理员凭据登录到 Infoblox 帐户，并预创建以下 Infoblox 可扩展属性，然后才能使用下载的 infoblox.zip:

- VMware NIC index
- VMware resource ID
- Tenant ID
- CMP Type
- VM ID
- VM Name

前提条件

- 确认您拥有 [Infoblox](#) 帐户，并且拥有对组织 Infoblox 帐户的正确访问凭据。
- 确认 Infoblox WAPI 版本受支持。IPAM 与 Infoblox 的集成依赖于 Infoblox WAPI 版本 2.7。支持 WAPI v2.7 的所有 Infoblox 设备均受支持。
- 查看对 [vRealize Automation](#) 中的 IPAM 集成使用特定于 Infoblox 的属性和可扩展属性。

步骤

- 1 使用管理员凭据登录到 Infoblox 帐户。

这些凭据是您在 vRealize Automation Cloud Assembly 中使用 [基础架构 > 连接 > 集成 >](#) 菜单顺序创建外部 IPAM 集成点时指定的管理员用户名和密码凭据。

- 2 按照 Infoblox 文档中所述的过程在 Infoblox 应用程序中创建所需的以下可扩展属性。

- VMware NIC index - 整数类型
- VMware resource ID - 字符串类型
- Tenant ID - 字符串类型
- CMP Type - 字符串类型
- VM ID - 字符串类型
- VM Name - 字符串类型

Infoblox 文档主题 [关于可扩展属性](#) 中的添加可扩展属性部分介绍了此过程。另请参见 [管理可扩展属性](#)。

后续步骤

添加所需的属性后，可以按照 [下载并部署外部 IPAM 提供程序软件包](#) 以在 vRealize Automation Cloud Assembly 中使用中所述继续执行下载和部署 Infoblox 软件包的过程。

下载并部署外部 IPAM 提供程序软件包以在 vRealize Automation Cloud Assembly 中使用

在 vRealize Automation Cloud Assembly 中定义外部 IPAM 集成点之前，您需要一个已配置的 IPAM 提供程序软件包。您可以从 IPAM 提供程序的网站或 vRealize Automation Cloud Assembly 商城下载提供程序软件包。

可以从以下位置获取提供程序特定的集成软件包：IPAM 提供程序的网站、VMware Solution Exchange 商城或者 vRealize Automation Cloud Assembly 的商城选项卡（如果可用）。

注 此示例使用 VMware 提供的 Infoblox 软件包 `Infoblox.zip`，该软件包可从 VMware Solution Exchange 商城以如下形式进行下载：

- [vRA Cloud Infoblox 插件版本 1.1](#) - 支持 vRealize Automation 8.1
- [vRA Cloud Infoblox 插件版本 1.0](#) - 支持 vRealize Automation 8.0.1
- [vRA Cloud Infoblox 插件版本 0.1](#) - 支持 vRealize Automation 8.0

IPAM 与 Infoblox 的集成依赖于 Infoblox WAPI 版本 2.7。支持 WAPI v2.7 的所有 Infoblox 设备均受支持。

有关如何为其他 IPAM 提供程序创建 IPAM 集成软件包的信息（如果商场中尚不存在该软件包），请参见[如何使用 IPAM SDK 为 vRealize Automation 创建提供程序特定的外部 IPAM 集成软件包](#)。

IPAM 提供程序软件包包含与元数据和其他配置一起打包的脚本。这些脚本包含与外部 IPAM 提供程序协调时 vRealize Automation Cloud Assembly 执行的操作所使用的源代码。示例操作包括 `Allocate an IP address for a virtual machine`、`Fetch a list of IP ranges from the provider` 和 `Update the MAC address of a host record in the provider`。

前提条件

- 确认您具有云管理员凭据。请参见在[vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有云管理员用户角色。请参见[vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有外部 IPAM 提供程序（例如 Infoblox 或 Bluecat）的帐户，并且具有使用 IPAM 提供程序访问组织帐户的正确访问凭据。
- 如果您要使用 Infoblox 作为外部 IPAM 提供程序，请确认已将所需的可扩展属性添加到 Infoblox 帐户，然后再继续操作。请参见在[Infoblox 应用程序中添加所需的可扩展属性以与 vRealize Automation 集成](#)。

步骤

- 1 导航到 VMware Solution Exchange 商城中的 [vRA Cloud Infoblox 插件版本 0.1](#) (vRealize Automation 8.0) 或 [vRA Cloud Infoblox 插件版本 1.0](#) (vRealize Automation 8.0.1) 软件包页面。
- 2 登录并下载插件软件包。
- 3 如果尚未执行此操作，请在 Infoblox 中添加所需的可扩展属性。请参见在[Infoblox 应用程序中添加所需的可扩展属性以与 vRealize Automation 集成](#)。

结果

现在，可以使用**集成 > 添加集成 > IPAM > 管理提供程序 > 导入软件包**菜单序列，对此软件包进行部署，如在 [vRealize Automation 中添加外部 IPAM 集成点](#) 中所述。

在 vRealize Automation 中为 IPAM 集成点创建运行环境

在 vRealize Automation 中定义外部 IPAM 集成点之前，需要创建或访问现有的运行环境以作为 IPAM 提供程序和 vRealize Automation 之间的中介。运行环境通常是 Amazon Web Services 或 Microsoft Azure 云帐户，或与云可扩展性代理相关联的基于操作的可扩展性内部部署集成点。

外部 IPAM 集成需要运行环境。定义 IPAM 集成点时，通过指定可用运行环境在 vRealize Automation Cloud Assembly 和 IPAM 提供程序之间创建连接。

在功能即服务 (FaaS) 提供程序（如 Amazon Web Services Lambda、Microsoft Azure Functions 或基于操作的可扩展性 (ABX) 内部部署嵌入式集成点）助力的运行环境中，IPAM 集成使用一组已下载的程序特定脚本或插件。运行环境用于连接到外部 IPAM 提供程序，例如 Infoblox。

注 Infoblox IPAM 集成点需要一个基于操作的可扩展性 (ABX) 内部部署嵌入式集成点。

每种类型的运行时环境都有优缺点：

- 基于操作的可扩展性 (ABX) 集成点
 - 免费，无其他供应商使用成本
 - 可以连接到位于不能公开访问的 NAT/Firewall 后面的内部部署数据中心内的 IPAM 供应商设备，例如 Infoblox
 - 与商业云供应商相比，速度慢且性能可靠性稍差
- Amazon Web Services
 - 有相关的供应商 FaaS 连接/使用成本
 - 无法连接到位于不能公开访问的 NAT/Firewall 后面的内部部署数据中心内的 IPAM 供应商设备
 - 快速且性能高度可靠
- Microsoft Azure
 - 有相关的供应商 FaaS 连接/使用成本
 - 无法连接到位于不能公开访问的 NAT/Firewall 后面的内部部署数据中心内的 IPAM 供应商设备
 - 快速且性能高度可靠

前提条件

- 确认您具有云管理员凭据。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有外部 IPAM 提供程序（例如 Infoblox 或 Bluecat）的帐户，并且具有使用 IPAM 提供程序访问组织帐户的正确访问凭据。

- 确认您有权访问 IPAM 提供程序（例如 Infoblox 或 BlueCat）的已部署集成软件包。部署的软件包最初从 IPAM 提供程序网站或 vRealize Automation Cloud Assembly 商城以 .zip 形式下载，然后部署到 vRealize Automation Cloud Assembly。

有关如何部署提供程序软件包 .zip 文件并使其在“IPAM 集成”页面上可作为提供程序值提供的信息，请参见[下载并部署外部 IPAM 提供程序软件包以在 vRealize Automation Cloud Assembly 中使用](#)。

步骤

- 1 要创建一个基于 FaaS 的内部部署可扩展性操作以用作 IPAM 集成运行环境，请选择**可扩展性 > 库 > 操作**。
- 2 单击**新建操作**，输入操作名称和描述，然后指定项目。
- 3 在 **FaaS 提供程序** 下拉菜单中，选择**内部部署**。
- 4 填写表单以定义可扩展性操作。



有关运行环境的相关信息，请参见在视频中播插的此 [Infoblox IPAM 插件 1.1 集成博客视频](#)（约 24 分钟）。

在 vRealize Automation 中添加外部 IPAM 集成点

vRealize Automation 支持与外部 IPAM 提供程序集成。可以使用提供程序特定的 IPAM 集成点获取并管理蓝图部署的 IP 地址和相关网络特性。

在此示例中，您将创建一个外部 IPAM 集成点，以支持使用外部 IPAM 提供程序访问组织的帐户。在此示例工作流中，IPAM 提供程序为 Infoblox，且提供程序特定的集成软件包已存在。虽然这些说明特定于 Infoblox 集成，但在为不同的外部 IPAM 提供程序创建 IPAM 集成时，也可以将其用作参考。

可以从以下位置获取提供程序特定的集成软件包：IPAM 提供程序的网站、[VMware Solution Exchange 商城](#)或者 vRealize Automation Cloud Assembly 的[商城](#)选项卡（如果可用）。

此示例使用 VMware 提供的 Infoblox 软件包 Infoblox.zip，该软件包可从 VMware Solution Exchange 商城以如下形式进行下载：

- [vRA Cloud Infoblox 插件版本 1.1](#) - 支持 vRealize Automation 8.1
- [vRA Cloud Infoblox 插件版本 1.0](#) - 支持 vRealize Automation 8.0.1
- [vRA Cloud Infoblox 插件版本 0.1](#) - 支持 vRealize Automation 8.0

前提条件

- 确认您具有云管理员凭据。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有外部 IPAM 提供程序的帐户，并且具有使用 IPAM 提供程序访问组织帐户的正确访问凭据。

- 确认您有权访问 IPAM 提供程序的已部署集成软件包。部署的软件包最初从 IPAM 提供程序网站或 VMware Solution Exchange 商城以 .zip 形式下载，然后部署到 vRealize Automation。
有关如何下载和部署提供程序软件包 .zip 文件并使其在“IPAM 集成”页面上可作为提供程序值提供的信息，请参见[下载并部署外部 IPAM 提供程序软件包以在 vRealize Automation Cloud Assembly 中使用](#)。
- 确认您有权访问为 IPAM 提供程序配置的运行环境。运行环境通常是一个基于操作的可扩展性 (ABX) 内部部署嵌入式集成点。
有关运行环境特性的信息，请参见在[vRealize Automation 中为 IPAM 集成点创建运行环境](#)。
- 在 Infoblox 应用程序中启用所需的可扩展属性。请参见在[Infoblox 应用程序中添加所需的可扩展属性以与 vRealize Automation 集成](#)。
- 如果您没有外部 Internet 访问权限，则可以配置一个 Internet 服务器代理。请参见[如何配置 vRealize Automation 的 Internet 代理服务器](#)。

步骤

- 1 选择**基础架构 > 连接 > 集成**，然后单击**添加集成**。
- 2 单击 **IPAM**。
- 3 在**提供程序**下拉列表中，从列表中选择已配置的 IPAM 提供程序软件包，例如 *Infoblox_hrg*。
如果此列表为空，请单击**导入提供程序软件包**，导航到现有提供程序软件包 .zip 文件，然后选择该文件。如果没有此提供程序 .zip 文件，则可以从 IPAM 提供程序的网站或从 vRealize Automation Cloud Assembly 的**商城**选项卡获取该文件。
有关如何在 vCenter 中部署提供程序软件包 .zip 文件并使其在“集成”页面上作为提供程序值提供的信息，请参见[下载并部署外部 IPAM 提供程序软件包以在 vRealize Automation Cloud Assembly 中使用](#)。
有关如何升级现有 IPAM 集成以使用最新版本的供应商 IPAM 集成软件包的信息，请参见[如何在 vRealize Automation 中升级到较新的 IPAM 集成软件包](#)。
- 4 输入具有外部 IPAM 提供程序的帐户的管理员用户名和密码凭据以及所有其他（如果有）必填字段，如提供程序的主机名。
在此示例中，您将通过以下步骤获取 Infoblox IPAM 提供程序的主机名：
 - a 在单独的浏览器选项卡中，使用您的 Infoblox 管理员凭据登录到 IPAM 提供程序帐户。
 - b 复制主机名 URL。
 - c 将您的主机名 URL 粘贴到“IPAM 集成”页面上的**主机名字段**中。

- 5 在**运行环境**下拉列表中，选择一个现有基于操作的可扩展性内部部署集成点，例如 *Infoblox_abx_intg*。

运行环境支持在 vRealize Automation 和外部 IPAM 提供程序之间进行通信。

注 如果使用 Amazon Web Services 或 Microsoft Azure 云帐户作为集成运行环境，请确保 IPAM 提供程序设备符合以下条件：可以通过 Internet 进行访问，不位于 NAT 或防火墙后面，并且具有可公开解析的 DNS 名称。如果 IPAM 提供程序不可访问，则 Amazon Web Services Lambda 或 Microsoft Azure 函数无法与其相连接，集成将失败。有关相关信息，请参见在 [vRealize Automation](#) 中为 IPAM 集成点创建运行环境。

IPAM 框架仅支持基于操作的可扩展性 (ABX) 内部部署嵌入式运行环境。

注 Infoblox IPAM 集成点需要一个基于操作的可扩展性 (ABX) 内部部署嵌入式集成点。

配置的云帐户或集成点允许在 vRealize Automation 和 IPAM 提供程序之间进行通信，在此示例 Infoblox 中通过关联的云可扩展性代理进行。可以选择已创建的提供程序，也可以创建一个提供程序。

有关如何创建运行环境的信息，请参见在 [vRealize Automation](#) 中为 IPAM 集成点创建运行环境。

- 6 单击**验证**。

由于此示例对运行环境使用基于内部部署操作的可扩展性集成，因此可以查看验证操作。

- a 单击**可扩展性**选项卡。

- b 单击**活动 > 操作运行**，然后从筛选器中选择**所有运行**或**集成运行**，以查看端点验证操作是否已启动且正在运行。

- 7 当系统提示您信任来自 IPAM 提供程序的自签名证书时，单击**接受**。

接受自签名证书后，可以继续完成验证操作。

- 8 输入此 IPAM 集成点的**名称**（例如 *Infoblox_Integration*）和**描述**（例如 *Infoblox IPAM with ABX integration for team HRG*）。

- 9 单击**添加**以保存新的外部 IPAM 集成点。

将模拟数据收集操作。将从 IPAM 提供程序收集网络 and IP 范围数据。可以按如下方式查看数据收集操作：

- a 单击**可扩展性**选项卡。

- b 单击**活动 > 操作运行**，并注意数据收集操作已启动且正在运行。可以打开并查看操作运行内容。

结果

提供程序特定的外部 IPAM 集成现在可用于网络和网络配置文件。

在 vRealize Automation Cloud Assembly 中配置网络和网络配置文件以使用 IPAM 提供程序值

可以将网络定义为使用从外部 IPAM 提供程序（而不是从 vRealize Automation Cloud Assembly 内部）获取并由其管理的 IP 地址值。

可以将网络定义为访问您在组织的外部 IPAM 提供程序帐户中定义的现有 IP 设置。此步骤将详述在上一步中创建的 Infoblox 提供程序集成。

在此示例中，将使用已从 vCenter 收集数据的现有网络配置网络配置文件。然后，将这些网络配置为从外部 IPAM 提供程序（在本例中为 Infoblox）获取 IP 信息。从 vRealize Automation Cloud Assembly 置备且可与此网络配置文件匹配的虚拟机从外部 IPAM 提供程序获取其 IP 以及其他与 TCP/IP 相关的设置。

有关网络的详细信息，请参见[网络资源](#)。有关网络配置文件的详细信息，请参见[如何添加 vRealize Automation Cloud Assembly 网络配置文件](#)和了解有关 vRealize Automation Cloud Assembly 中的网络配置文件的更多信息。

前提条件

此步骤顺序显示在 IPAM 提供程序集成工作流的上下文中。请参见[提供程序特定的外部 IPAM 集成用例](#)。

- 确认您具有云管理员凭据。请参见在 [vRealize Automation 中使用云帐户所需的凭据](#)。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有外部 IPAM 提供程序（例如 [Infoblox](#) 或 [Bluecat](#)）的帐户，并且具有使用 IPAM 提供程序访问组织帐户的正确访问凭据。在此示例工作流中，IPAM 提供程序为 Infoblox。
- 确认您具有 IPAM 提供程序的 IPAM 集成点。请参见在 [vRealize Automation 中添加外部 IPAM 集成点](#)。

步骤

- 1 要配置网络，请单击[基础架构 > 资源 > 网络](#)。
- 2 在[网络](#)选项卡上，选择要与 IPAM 提供程序集成点一起使用的现有网络。在此示例中，网络名称为 *net.23.117-only-IPAM*。

列出的网络已由 vRealize Automation Cloud Assembly 从组织中的 vCenter 进行数据收集。

- 3 要从外部 IPAM 提供程序获取值，请确认除[帐户/区域、名称和网络域](#)外，所有其他网络设置均为空，其中包括以下网络设置：
 - 域（请参见步骤 8 中的“注意”）
 - CIDR
 - 默认网关
 - DNS 服务器
 - DNS 搜索域
- 4 单击 [IP 范围](#) 选项卡，然后单击[添加 IPAM IP 范围](#)。
- 5 从[网络](#)菜单中，选择刚刚配置的网络，例如 *net.23.117-only-IPAM*。

- 6 从**提供程序**菜单中，选择在工作流的之前步骤中创建的 IPAM 集成点 *Infoblox_Integration*
- 7 从现在可见的**地址空间**下拉菜单中，选择一个列出的网络视图。

Infoblox 中的地址空间称为网络视图。

网络视图从 IPAM 提供程序帐户获取。此示例使用刚刚配置的网络子网（例如 *net.23.117-only-IPAM*）、在工作流的之前步骤中创建的集成点 *Infoblox_Integration*，以及名为 *default* 的地址空间。

列出的地址空间值从外部 IPAM 提供程序获取。

- 8 从可用于所选地址空间的已显示网络列表中，选择一个或多个网络，例如，选择 10.23.117.0/24。
- 对于此示例，选定网络的**域**和 **DNS 服务器**列值包含来自 Infoblox 的值。

注 如果在步骤 3 中选择的网络包含为 vRealize Automation Cloud Assembly 指定的“域”，然后从包含“域”值的外部 IPAM 提供程序地址空间选择一个网络，则外部 IPAM 提供程序网络中的“域”值优先于在 vRealize Automation Cloud Assembly 中指定的“域”。如果 IPAM IP 范围设置没有像上述那样在 Cloud Assembly 或外部 IPAM 提供程序中指定“域”值，置备将失败。

- 9 单击**添加**以保存网络的 IPAM IP 范围。

范围在 **IP 范围**表中可见。

- 10 单击 **IP 地址**选项卡。

使用外部 IPAM 提供程序中的新地址范围置备计算机后，**IP 地址**表中将显示一条新记录。

- 11 要将网络配置文件配置为使用该网络，请单击**基础架构 > 配置 > 网络配置文件**。

- 12 为网络配置文件命名，例如 *Infoblox-NP*，然后添加以下示例设置。

- “摘要”选项卡
 - 指定 vSphere 云帐户/区域。
 - 为网络配置文件添加功能标记，例如，名为 *infoblox_abx* 的标记。
请记录功能标记，因为还必须将其用作蓝图限制标记，才能在蓝图中进行置备关联。
- “网络”选项卡
 - 添加之前创建的网络，例如 *net.23.117-only-IPAM*。

- 13 单击**保存**以保存包含这些设置的网络配置文件。

结果

网络和网络配置文件设置现已配置为支持外部 IPAM 集成，并且可在蓝图中使用。

定义并部署使用 IPAM 提供程序范围分配的 vRealize Automation Cloud Assembly 蓝图

可以将蓝图定义为从外部 IPAM 提供程序获取 IP 地址分配并进行管理。

这是外部 IPAM 集成工作流的最后一步，在此步骤中，将定义并部署蓝图，用于将之前定义的网络和网络配置文件连接到您组织的 Infoblox 帐户，以从外部 IPAM 提供程序（而非 vRealize Automation Cloud Assembly）获取已部署虚拟机的 IP 地址分配并进行管理。

该工作流使用 Infoblox 作为外部 IPAM 提供程序，在某些步骤中，示例值为 Infoblox 所独有，但本意是该过程可应用于其他外部 IPAM 集成。

部署蓝图并启动虚拟机后，用于部署中每个虚拟机的 IP 地址将在**资源 > 网络**页面中显示为一个网络条目，在 IPAM 提供程序帐户的 IPAM 提供程序网络以及主机 vCenter 中每个已部署虚拟机的 vSphere Web Client 记录中显示为一条新主机记录。

前提条件

外部 IPAM 提供程序集成工作流的上下文中显示了此步骤顺序。请参见[提供程序特定的外部 IPAM 集成用例](#)。

- 确认您具有云管理员凭据。请参见在 [vRealize Automation](#) 中使用云帐户所需的凭据。
- 确认您具有云管理员用户角色。请参见 [vRealize Automation Cloud Assembly 用户角色是什么](#)。
- 确认您具有外部 IPAM 提供程序（例如 Infoblox 或 BlueCat）的帐户，并且具有使用 IPAM 提供程序访问组织帐户的正确访问凭据。
- 确认您对主机帐户具有管理员访问权限，并满足在主机 vCenter 中已部署虚拟机的 vSphere Web Client 记录中显示状态记录所需的任何角色要求。
- 确认您具有外部 IPAM 提供程序的 IPAM 集成点。请参见在 [vRealize Automation](#) 中添加外部 IPAM 集成点。
- 确认您已配置 vRealize Automation Cloud Assembly 网络和网络配置文件，以支持所需 IPAM 集成点的外部 IPAM 集成。请参见在 [vRealize Automation Cloud Assembly](#) 中配置网络和网络配置文件以使用 IPAM 提供程序值。
- 确认您的项目和云区域已添加标记，与 IPAM 集成点和网络或网络配置文件中的标记匹配。（可选）将项目配置为支持自定义资源命名。

有关项目和云区域的角色以及蓝图中其他基础架构元素的角色详细信息，请参见 [WordPress 用例](#)。有关标记的详细信息，请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署](#)。

有关使用项目中的设置自定义命名虚拟机的信息，请参见[如何使用 vRealize Automation Cloud Assembly 自定义已部署资源的名称](#)。

步骤

1 单击**蓝图 > 新建**，在**新建蓝图**页面中输入以下信息，然后单击**创建**。

- 名称 = ipam-bpa

- **描述** = Blueprint that uses Infoblox IPAM integration
 - **项目** = 123VC
- 2 对于此示例，将云平台无关的计算机组件和云平台无关的网络组件添加到蓝图画布并连接这两个组件。
 - 3 编辑蓝图代码，以将限制标记添加到与您添加到网络配置文件的功能标记匹配的网络组件。对于此示例，该标记值为 *infoblox_abx*。
 - 4 编辑蓝图代码，以指定网络分配类型为 *static*。

对于此示例，已知指定的 IP 地址 10.23.117.4 当前在为关联网络配置文件中的网络所选择的外部 IPAM 地址空间中可用。*static* 分配设置是必填项，而 *address* 值不是。如果要在特定地址处启动外部 IP 地址选择，可以这样做，但不是必需操作。如果未指定 *address* 值，外部 IPAM 提供程序将选择外部 IPAM 网络中的下一个可用地址。

- 5 根据以下示例验证蓝图代码。

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
      name: ipam
      constraints:
        - tag: infoblox_abx
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          assignment: static
          address: 10.23.117.4
          name: '${resource.Cloud_Network_1.name}'
```

注 默认情况下，Infoblox 集成会在 Infoblox 的默认 DNS 视图中创建一条 DNS 主机记录。

如果您的 Infoblox 管理员已创建自定义 DNS 视图，您可以覆盖默认集成行为，并使用计算机组件中的 Infoblox.IPAM.Network.dnsView 属性指定一个已命名的视图。例如，您可以将以下属性添加到 Cloud_Machine_1 组件，以在 Infoblox 中指定一个已命名的 DNS 视图。

```
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ubuntu
    flavor: small
    Infoblox.IPAM.Network.dnsView:<dns-view-name>
```

有关配置和使用 DNS 视图的信息，请参见 Infoblox 产品文档中的 [DNS 视图](#)。

- 6 单击蓝图页面上的**部署**，将部署命名为 *Infoblox-1*，然后在**部署类型**页面上单击**部署**。
- 7 在部署蓝图时，单击**可扩展性**选项卡，然后选择**活动 > 操作运行**，会看到 *Infoblox_AllocateIP_n* 可扩展性操作正在运行。
完成可扩展性操作并置备计算机后，*Infoblox_Update_n* 操作会将 MAC 地址传播到 Infoblox。
- 8 可以登录到 Infoblox 帐户并打开，以在关联的 10.23.117.0/24 网络中查看 IPAM 地址的新主机记录。还可以在 Infoblox 中打开“DNS”选项卡以查看新的 DNS 主机记录。
- 9 要验证是否正在置备虚拟机，请登录到主机 vCenter 和 vSphere Web Client，找到已置备的计算机并查看 DNS 名称和 IP 地址。
启动已置备的虚拟机后，MAC 地址将通过 *Infoblox_AllocateIP* 可扩展性操作传播到 Infoblox。
- 10 要在 vRealize Automation Cloud Assembly 中查看新的网络记录，请选择**基础架构 > 资源 > 网络**，然后单击以打开 **IP 地址**选项卡。
- 11 如果删除部署，则会释放部署中虚拟机的 IPAM 地址，这些 IP 地址将再次供外部 IPAM 提供程序进行其他分配。vRealize Automation Cloud Assembly 中此事件的可扩展性操作为 *Infoblox_Deallocate*。

对 vRealize Automation 中的 IPAM 集成使用特定于 Infoblox 的属性和可扩展属性

对于包含 Infoblox 的外部 IPAM 集成的 vRealize Automation 项目，可以使用特定于 Infoblox 的属性。

以下 Infoblox 属性可用于 Infoblox IPAM 集成。您可以在 vRealize Automation 中使用这些属性，以便在蓝图部署期间进一步控制 IP 地址的分配。这些属性的使用是可选的。

虽然以下所有 Infoblox 属性均可与适用于 vRealize Automation 8.0 的 [vRA Cloud Infoblox 插件版本 0.1](#) 软件包一起使用，但 `Infoblox.IPAM.Network.dnsView` 属性仅可与适用于 vRealize Automation 8.0.1 的 [vRA Cloud Infoblox 插件版本 1.0](#) 和适用于 vRealize Automation 8.1 的 [vRA Cloud Infoblox 插件版本 1.1](#) 软件包一起使用。

注 这些属性的使用未包括在[提供程序特定的外部 IPAM 集成用例示例](#)工作流中。

- `Infoblox.IPAM.createFixedAddress`

此属性让您能够在 Infoblox 内创建固定地址记录。可能的值为 `True` 和 `False`。默认情况下，将创建主机记录。默认值：`False`。

- `Infoblox.IPAM.Network.dnsView`

通过此属性，您可以在 Infoblox 中创建主机记录时使用 DNS 视图。默认值：`default`。

- `Infoblox.IPAM.Network.enableDns`

在 Infoblox 中分配 IP 时，还可以使用此属性创建 DNS 记录。可能的值为 `True` 和 `False`。默认值：`True`。

- `Infoblox.IPAM.Network.dnsSuffix`

通过此属性，您可以将 Infoblox 网络的 *domain* DHCP 选项替换为新选项。如果 Infoblox 网络未设置 *domain* DHCP 选项，或者必须覆盖 *domain* DHCP 选项，此功能将非常有用。默认值：无（空字符串）。

只有在将 `Infoblox.IPAM.Network.enableDns` 设置为 `True` 时，`Infoblox.IPAM.Network.dnsSuffix` 才适用。

您可以在 vRealize Automation Cloud Assembly 中使用以下方法之一指定 Infoblox 属性：

- 您可以使用 **基础架构 > 配置 > 项目** 页面中的 **自定义属性** 指定项目属性。使用此方法时，指定的属性将应用于在此项目范围内置备的所有计算机。
- 您可以在蓝图中的每个计算机组件上指定属性。下面的示例蓝图代码介绍如何使用 `Infoblox.IPAM.Network.dnsView` 属性：

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      Infoblox.IPAM.Network.dnsView: default
      image: ubuntu
      cpuCount: 1
      totalMemoryMB: 1024
      networks:
        - network: '${resource.Cloud_Network_1.id}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
      constraints:
        - tag: mk-ipam-demo
```

- 可以使用可扩展性订阅指定属性。

有关与此用例相关的 Infoblox 可扩展属性的相关信息，请参见在 [Infoblox 应用程序中添加所需的可扩展属性](#) 以与 [vRealize Automation](#) 集成。

构建您的 vRealize Automation Cloud Assembly 资源基础架构

4

在 vRealize Automation Cloud Assembly 资源基础架构中，您可以将云帐户区域定义为可在其中部署蓝图及其工作负载的区域。

此外，资源基础架构还涉及创建映像和计算机大小的通用映射，以及创建定义跨云帐户区域或数据中心的网络和存储功能的配置文件。

本章讨论了以下主题：

- 如何添加定义 vRealize Automation Cloud Assembly 目标放置区域或数据中心云区域的云区域
- 如何添加 vRealize Automation Cloud Assembly 特定实例映射以在 vRealize Automation Cloud Assembly 中创建通用计算机大小
- 如何添加 vRealize Automation Cloud Assembly 映像映射以创建通用操作系统
- 如何添加 vRealize Automation Cloud Assembly 网络配置文件
- 如何添加负责不同需求的 vRealize Automation Cloud Assembly 存储配置文件
- 如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署
- 如何使用 vRealize Automation Cloud Assembly 中的资源

如何添加定义 vRealize Automation Cloud Assembly 目标放置区域或数据中心的云区域

vRealize Automation Cloud Assembly 云区域是云帐户类型（例如 AWS 或 vSphere）中的一组资源。

特定帐户区域中的云区域是您的蓝图部署工作负载的位置。每个云区域均与一个 vRealize Automation Cloud Assembly 项目相关联。

选择**基础架构 > 配置 > 云区域**，然后单击**添加新区域**。

了解有关 vRealize Automation Cloud Assembly 云区域的更多信息

vRealize Automation Cloud Assembly 云区域是您的云帐户类型（例如 AWS 或 vSphere）中的部分。云区域特定于项目。

其他放置控制包括放置策略选项、能力标记和计算标记。

- 放置策略

放置策略有助于在指定的区域中选择主机。

- **default** - 在主机上随机放置计算资源。
- **binpack** - 将计算资源放置在负载最多但仍有足够资源运行给定计算资源的主机上。
- **spread** - 在主机之间均匀放置计算资源。

能力标记

蓝图包含有助于确定部署放置的限制标记。在部署过程中，蓝图限制标记将映射到云区域中匹配的能力标记，以确定哪些云区域可用于计算资源放置。

计算资源

您可以查看和管理可置备到该云区域的计算资源。

如果 vCenter 计算群集已启用 DRS，则云区域仅显示计算资源列表中的群集，不显示子主机。如果 vCenter 计算群集未启用 DRS，则云区域仅显示独立 ESXi 主机（如果存在）。

计算机标记有助于进一步控制放置。您可以使用标记来筛选可用的计算资源，以便仅列出与一个或多个标记匹配的计算资源，如以下示例中所示。

- 计算资源不包含任何标记，并且未使用任何筛选。



- 两个计算资源包含同一个标记，但未使用任何筛选。



- 两个计算资源包含同一个标记，并且标记筛选器与两个计算资源上使用的标记匹配。



■ 项目

您可以查看哪些项目已配置为支持将工作负载置备到该云区域。

创建云区域后，您可以验证其配置。

如何添加 vRealize Automation Cloud Assembly 特定实例映射以在 vRealize Automation Cloud Assembly 中创建通用计算机大小

在 vRealize Automation Cloud Assembly 特定实例映射中，您可以使用自然语言为特定的云帐户/区域定义目标部署大小。

特定实例映射表示对您的环境有意义的部署大小。一个示例可能是，对于已命名数据中心中的 vCenter 帐户，以及已命名区域中的 Amazon Web Services 帐户的 t2.nano 实例，small 表示 1 个 CPU 和 2 GB 内存，large 表示 2 个 CPU 和 8 GB 内存。

选择**基础架构 > 配置 > 特定实例映射**，然后单击**新建特定实例映射**。

了解有关 vRealize Automation Cloud Assembly 中的特定实例映射的更多信息

通过使用自然语言命名，特定实例映射对 vRealize Automation Cloud Assembly 中的特定云帐户/区域所使用的一组目标部署大小进行分组。

使用特定实例映射，您可以创建包含各个帐户区域的相似特定实例大小的命名映射。例如，名为 standard_small 的特定实例映射可能包含项目中部分或全部可用帐户/区域的相似特定实例大小（例如，1 个 CPU，2GB RAM）。生成蓝图时，可以选择适合您的需求的可用特定实例。

按部署意图组织项目的特定实例映射。

要简化蓝图创建，可以在添加新的云帐户时选择预配置选项。选择预配置选项时，将选择指定区域的组织最常用的特定实例映射和映像映射。

对于包含 vSphere 资源的蓝图中的映像映射，如果没有为 vSphere 云区域定义特定实例映射，则可以使用蓝图中的 vSphere 特定设置配置无限内存和 CPU。如果为 vSphere 云区域定义了特定实例映射，则特定实例映射将用作蓝图中 vSphere 特定配置的限制。

有关基本的特定实例映射示例，请参见 [WordPress 用例：添加特定实例映射](#)。

如何添加 vRealize Automation Cloud Assembly 映像映射以创建通用操作系统

在 vRealize Automation Cloud Assembly 映像映射中，您可以使用自然语言为特定的云帐户/区域定义目标部署操作系统。

选择**基础架构 > 配置 > 映像映射**，然后单击**新建映像映射**。

了解有关 vRealize Automation Cloud Assembly 中的映像映射的更多信息

映像映射使用自然语言命名对 vRealize Automation Cloud Assembly 中特定云帐户/区域使用的一组预定义目标操作系统规范进行分组。

诸如 Microsoft Azure 和 Amazon Web Services 等云供应商帐户使用映像将一组目标部署条件（包括操作系统和相关配置设置）分组到一起。基于 vCenter 和 NSX 的环境（包括 VMware Cloud on AWS）使用类似的分组机制定义一组操作系统部署条件。生成并最终部署和迭代蓝图时，您可以选择最符合需求的可用映像。

可以按相似操作系统设置、标记策略和功能部署意向组织项目的映像映射。

有关如何定义基本映像映射的示例，请参见 [WordPress 用例：添加映像映射](#)。

要简化蓝图创建，可以在添加新的云帐户时选择预配置选项。选择预配置选项时，将选择指定区域的组织最常用的特定实例映射和映像映射。

将映像信息添加到蓝图时，可以使用计算机组件 properties 部分中的 image 或 imageRef 条目。例如，如果要从快照克隆，请使用 imageRef 属性。

有关蓝图代码中的 image 和 imageRef 条目的示例，请参见第 6 章 [设计 vRealize Automation Cloud Assembly 部署](#)。

要分配内容库的权限，管理员必须将该权限作为全局权限授予用户。有关信息，请参见位于 [VMware vSphere 文档](#) 内的 vSphere 虚拟机管理中的[内容库权限的层次结构继承](#)。

同步云帐户/区域的映像

您可以运行映像同步，以确保正在**基础架构 > 配置 > 映像映射**页面上为给定云帐户/区域添加或移除的映像为最新映像。

- 1 通过选择**基础架构 > 连接 > 云帐户**来打开关联的云帐户/区域。选择现有的云帐户/区域。
- 2 单击**同步映像**按钮，然后等待操作完成。



- 3 操作完成后，单击**基础架构 > 配置 > 映像映射**。定义新映像映射或编辑现有映像映射，然后选择步骤 1 中的云帐户/区域。

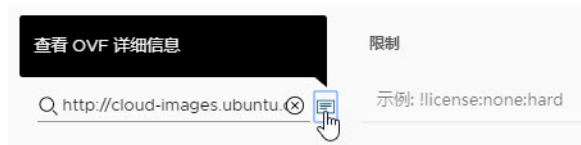
- 4 单击**映像映射**页面上的映像同步图标。



- 5 在**映像映射**页面上为指定云帐户/区域配置映像映射设置。

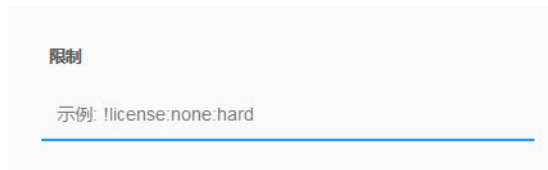
查看 OVF 详细信息

可以在 vRealize Automation Cloud Assembly 蓝图对象（如 vCenter 计算机组件和映像映射）中包含 OVF 规范。如果您的映像包含 OVF 文件，则无需打开文件即可发现其内容。将鼠标指针悬停在 OVF 上可显示 OVF 详细信息，包括其名称和位置。有关 OVF 文件格式的更多信息，请参见 [vcenter ovf:property](#)。



使用限制和标记细化映像选择

要在蓝图中进一步细化映像选择，可以添加一项或多项限制，以便对可部署的映像类型指定基于标记的限制。在创建或编辑映像映射配置时显示的**限制**示例为 `!license:none:hard`。提供的示例说明了基于标记的限制，在该示例中，仅当蓝图中不存在 `license:none` 标记时，才可以使用映像。如果添加标记（例如 `license:88` 和 `license:92`），则仅当 `license:88` 和 `license:92` 标记存在于蓝图中时，才能使用指定的映像。



使用云配置脚本控制部署

您可以在映像映射和/或蓝图中使用云配置脚本，来定义要在 vRealize Automation Cloud Assembly 部署中使用的自定义操作系统特性。例如，基于要将蓝图部署到公有云还是私有云，可以对映像应用特定的用户权限、操作系统权限或其他条件。云配置脚本遵循 `cloud-init` 格式（适用于基于 Linux 的映像）或 `cloudbase-init` 格式（适用于基于 Windows 的映像）。vRealize Automation Cloud Assembly 支持适用于 Linux 系统的 [cloud-init](#) 工具和适用于 Windows 系统的 [cloudbase-init](#) 工具。

对于 Windows 计算机，可以使用 `cloudbase-init` 支持的任何云配置脚本格式。

以下示例蓝图代码中的计算机资源使用包含云配置脚本的映像，该映像的内容显示在 `image` 条目中。

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: MyUbuntu16
```

```

https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ami-
ubuntu-16.04-1.10.3-00-15269239.ova
cloudConfig: |
  ssh_pwauth: yes
  chpasswd:
    list: |
      ${input.username}:${input.password}
    expire: false
  users:
    - default
    - name: ${input.username}
      lock_passwd: false
      sudo: ['ALL=(ALL) NOPASSWD:ALL']
      groups: [wheel, sudo, admin]
      shell: '/bin/bash'
  runcmd:
    - echo "Defaults:${input.username} !requiretty" >> /etc/sudoers.d/${input.username}

```

映像映射和蓝图包含云配置脚本时会出现什么情况

当包含云配置脚本的蓝图使用包含云配置脚本的映像映射时，这两个脚本将合并。合并操作首先处理映像映射脚本的内容，再处理蓝图脚本的内容，同时考虑脚本的格式是否为 #cloud-config。

- 对于采用 #cloud-config 格式的脚本，合并会按如下方式组合每个模块（例如 runcmd、users 和 write_files）的内容：
 - 对于内容为列表的模块，将合并映像映射和蓝图中的命令列表，不包括两个列表中相同的命令。
 - 对于内容为字典的模块，将合并命令，结果为两个字典的组合。如果两个字典中都存在相同的键，则将保留映像映射脚本字典中的键，而忽略蓝图脚本字典中的键。
 - 对于内容为字符串的模块，将保留映像映射脚本中的内容值，而忽略蓝图脚本中的内容值。
- 对于采用非 #cloud-config 格式的脚本，或者当一个脚本采用 #cloud-config 格式，而另一个脚本不采用此格式时，按以下方式将两个脚本组合在一起：先运行映像映射脚本，映像映射脚本完成后运行蓝图脚本。

有关相关信息，请参见[合并用户数据区域](#)。

有关配置和使用云配置脚本的更多信息

要在 vRealize Automation Cloud Assembly 中为 Windows 部署配置 cloud-init，请参见[如何在 vRealize Automation 中使用 cloud-init 或 cloudbase-init 设置 Windows 模板](#)。

有关使用云配置脚本的更多信息，请参见[如何在 vRealize Automation Cloud Assembly 蓝图中自动初始化计算机](#)和 VMware 博客文章[使用 Cloud-Init 自定义 Cloud Assembly 部署](#)。

如何添加 vRealize Automation Cloud Assembly 网络配置文件

vRealize Automation Cloud Assembly 网络配置文件描述要部署的网络的行为。

例如，网络可能需要面向 Internet，而不是仅供内部使用。网络及其配置文件是特定于云的。

选择[基础架构](#) > [配置](#) > [网络配置文件](#)，然后单击[新建网络配置文件](#)。

了解有关 vRealize Automation Cloud Assembly 中的网络配置文件的更多信息

网络配置文件定义了一组可用于特定区域中的云帐户或 vRealize Automation Cloud Assembly 中的数据中心的网络和网络设置。

您通常会定义网络配置文件以支持目标部署环境，例如，其中现有网络仅具有出站访问权限的小型测试环境，或需要一组安全策略的大型负载均衡生产环境。将网络配置文件视为特定于工作负载的网络特性的集合。

网络配置文件的内容

网络配置文件包含 vRealize Automation Cloud Assembly 中指定云帐户类型和区域的特定信息，其中包括以下设置：

- 网络配置文件的指定云帐户/区域和可选功能标记。
- 指定的现有网络及其设置。
- 定义网络配置文件按需方面及其他方面的网络策略。
- （可选）包含现有负载均衡器。
- （可选）包含现有安全组。

您可以根据网络配置文件确定网络 IP 管理功能。

网络配置文件功能标记与蓝图中的限制标记相匹配，以帮助控制网络选择。此外，分配给网络配置文件所收集网络的所有标记也与蓝图中的标记相匹配，以帮助在部署蓝图时控制网络选择。

功能标记是可选的。功能标记将应用于网络配置文件中的所有网络，但仅当将这些网络用作该网络配置文件的一部分时适用。对于不包含功能标记的网络配置文件，仅在网络标记上进行标记匹配。部署蓝图时，会应用匹配的网络配置文件中定义的网络和安全设置。

使用静态 IP 时，地址范围由 vRealize Automation 进行管理。对于 DHCP，IP 起始地址和结束地址由独立的 DHCP 服务器进行管理，而不是由 vRealize Automation 进行管理。使用 DHCP 或混合网络地址分配时，网络利用率值设置为零。按需网络的已分配范围基于网络配置文件中指定的 CIDR 和子网大小。要在部署中同时支持静态和动态地址分配，所分配的地址范围将一分为二：一个用于静态分配，另一个用于动态分配。

网络

网络（也称为子网）是 IP 网络的逻辑细分。网络会对云帐户、IP 地址或范围以及网络标记进行分组，以控制置备蓝图部署的方式和位置。配置文件中的网络参数定义了部署中的计算机如何通过 IP 第 3 层互相进行通信。网络可以具有标记。

可以将网络添加到网络配置文件，编辑网络配置文件使用的网络各个方面，以及从网络配置文件中移除网络。

■ 网络域或传输区域

网络域或传输区域是 vSphere vNetwork 分布式端口组 (dvPortGroup) 的分布式虚拟交换机 (dvSwitch)。传输区域是一个现有的 NSX 概念，类似于 *dvSwitch* 或 *dvPortGroup* 等术语。

使用 NSX 云帐户时，页面上的元素名称为**传输区域**，否则为**网络域**。

对于标准交换机，网络域或传输区域与交换机本身相同。网络域或传输区域定义 vCenter 内的子网边界。

传输区域控制 NSX 逻辑交换机可以访问的主机。它可以跨一个或多个 vSphere 集群。传输区域控制哪些集群及哪些虚拟机可以参与使用特定的网络。属于同一 NSX 传输区域的子网可用于相同的计算机主机。

■ 域

目标虚拟机的 vCenter Single Sign-On 域。域由 vCenter 管理员在 vSphere 配置期间配置。域确定 vCenter 中的本地身份验证空间。

■ IPv4 CIDR 和 IPv4 默认网关

vSphere 云帐户和蓝图中的 vSphere 计算机组件支持双 IPv6 和 IPv4 Internet 协议方法。例如，192.168.100.14/24 表示 IPv4 地址 192.168.100.14 及其关联的路由前缀 192.168.100.0，或者等效于其子网掩码 255.255.255.0，它具有 24 个前导 1 位。IPv4 块 192.168.100.0/22 表示从 192.168.100.0 到 192.168.103.255 的 1024 个 IP 地址。

■ IPv6 CIDR 和 IPv6 默认网关

vSphere 云帐户和蓝图中的 vSphere 计算机组件支持双 IPv6 和 IPv4 Internet 协议方法。例如，2001:db8::/48 表示从 2001:db8:0:0:0:0:0:0 到 2001:db8:0:ffff:ffff:ffff:ffff:ffff 的 IPv6 地址块。

按需网络不支持 IPv6 格式。

■ DNS 服务器和 DNS 搜索域

■ 支持公共 IP

选择此选项可将网络标记为公共网络。蓝图中具有 `network type: public` 属性的网络组件与标记为公共的网络匹配。将在蓝图部署期间进行进一步匹配以确定网络选择。

■ 区域的默认值

选择此选项可将网络标记为云区域的默认网络。在蓝图部署过程中，默认网络优先于其他网络。

■ 来源

标识网络源。

■ 标记

指定分配给网络的一个或多个标记。标记是可选项。标记匹配会影响可用于蓝图部署的网络。

网络项本身存在网络标记，而与网络配置文件无关。网络标记应用于其添加到的网络的每个实例，以及包含该网络的所有网络配置文件。网络可以实例化到任意数量的网络配置文件中。无论网络配置文件的驻留方式如何，网络标记在使用网络的任何地方都与该网络相关联。

部署蓝图时，蓝图网络组件中的限制标记与网络标记（包括网络配置文件功能标记）匹配。对于包含功能标记的网络配置文件，功能标记将应用于该网络配置文件的所有可用网络。部署蓝图时，会应用匹配的网络配置文件中定义的网络和安全设置。

网络策略

根据关联的云帐户，可以使用网络策略为 `outbound`、`private` 和 `routed` 网络类型以及按需安全组定义设置。当存在与该网络关联的负载均衡器时，也可以使用网络策略控制 `existing` 网络。

有关网络类型的信息，请参见在 [vRealize Automation Cloud Assembly 中使用网络和网络配置文件](#)。

以下按需选择的选项在[网络配置文件](#)屏幕帮助中进行了说明，并汇总如下。

- **不创建按需网络或按需安全组**

在指定 `existing` 或 `public` 网络类型时，可以使用此选项。需要 `outbound`、`private` 或 `routed` 网络的蓝图与此配置文件不匹配。

- **创建按需网络**

在指定 `outbound`、`private` 或 `routed` 网络类型时，可以使用此选项。

Amazon Web Services、Microsoft Azure、NSX、vSphere 和 VMware Cloud on AWS 支持此选项。

- **创建按需安全组**

在指定 `outbound` 或 `private` 网络类型时，可以使用此选项。

如果网络类型为 `outbound` 或 `private`，则会为匹配的蓝图创建新的安全组。

Amazon Web Services、Microsoft Azure、NSX 和 VMware Cloud on AWS 支持此选项。

网络策略设置可以特定于云帐户类型。这些设置在屏幕上的标志帮助中进行了说明，并汇总如下：

- **网络域或传输区域**

网络域或传输区域是 vSphere vNetwork 分布式端口组 (dvPortGroup) 的分布式虚拟交换机 (dvSwitch)。*传输区域*是一个现有的 NSX 概念，类似于 *dvSwitch* 或 *dvPortGroup* 等术语。

使用 NSX 云帐户时，页面上的元素名称为**传输区域**，否则为**网络域**。

对于标准交换机，网络域或传输区域与交换机本身相同。网络域或传输区域定义 vCenter 内的子网边界。

传输区域控制 NSX 逻辑交换机可以访问的主机。它可以跨一个或多个 vSphere 集群。传输区域控制哪些集群及哪些虚拟机可以参与使用特定的网络。属于同一 NSX 传输区域的子网可用于相同的计算机主机。

- **外部子网**

具有出站访问权限的按需网络需要具有出站访问权限的外部子网。该外部子网当在蓝图中请求时用于提供出站访问权限，它不控制网络布置。例如，该外部子网不影响专用网络的放置。

- **CIDR**

CIDR 表示法是 IP 地址及其关联的路由前缀的精简表示形式。CIDR 值指定了在置备期间用于创建子网的网络地址范围。[网络策略](#)选项卡上的此 CIDR 设置接受 IPv4 表示法，即以 `/nn` 结尾并包含介于 0-32 之间的值。

- **子网大小**

此选项指定要为使用此网络配置文件的部署中的每个隔离网络创建的按需网络大小（使用 IPv4 表示法）。子网大小设置可用于管理内部或外部 IP 地址。

按需网络不支持 IPv6 格式。

■ 分布式逻辑路由器

对于按需路由网络，使用 NSX-V 云帐户时必须指定分布式逻辑网络。

分布式逻辑路由器 (DLR) 用于在 NSX-V 上的按需路由网络之间路由东西向流量。仅当网络配置文件的帐户/区域值与 NSX-VCloud 帐户相关联时，此选项才可见。

■ IP 范围分配

此选项适用于支持 NSX 或 VMware Cloud on AWS 的云帐户，包括 vSphere。

可以选择以下三个选项之一来为部署网络指定 IP 范围分配类型：

■ 静态和 DHCP

默认选项，同时为推荐选项。此混合选项通过使用分配的 CIDR 和子网范围设置，将 DHCP 服务器池配置为使用 DHCP（动态）方法支持一半的地址空间分配，使用静态方法支持一半的 IP 地址空间分配。当连接到按需网络的一些计算机需要分配的静态 IP 地址，而一些计算机需要动态 IP 地址时，使用此选项。将创建两个 IP 范围。

在计算机连接到按需网络的部署（即，一些计算机分配有静态 IP，而另一些计算机由 NSXDHCP 服务器动态分配 IP）中，以及负载均衡器 VIP 为静态的部署中，此选项最为有效。

■ DHCP (动态)

此选项使用分配的 CIDR 在 DHCP 服务器上配置 IP 池。此网络的所有 IP 地址都动态分配。将为每个分配的 CIDR 创建一个 IP 范围。

■ 静态

此选项使用分配的 CIDR 静态分配 IP 地址。不需要为此网络配置 DHCP 服务器时使用此选项。将为每个分配的 CIDR 创建一个 IP 范围。

■ 网络资源 - 外部网络

外部网络也称为现有网络。这些网络已进行数据收集，可供选择。

■ 网络资源 - 第 0 层逻辑路由器

NSX-T 使用第 0 层逻辑路由器作为 NSX 部署外部网络的网关。第 0 层逻辑路由器为按需网络配置出站访问权限。

■ 网络资源 - Edge 集群

指定的 Edge 集群提供路由服务。Edge 集群用于为按需网络和负载均衡器配置出站访问权限。它可标识要在其中部署 Edge 设备的 Edge 集群或资源池。

■ 网络资源 - Edge 数据存储

指定的 Edge 数据存储用于置备 Edge 设备。此设置仅适用于 NSX-V。

负载均衡器

可以在网络配置文件中添加负载均衡器。可根据从源云帐户收集的数据信息使用列出的负载均衡器。

如果网络配置文件中任何负载均衡器上的标记与蓝图中负载均衡器组件上使用的标记相匹配，则会在部署过程中考虑负载均衡器。在部署蓝图时，将使用匹配网络配置文件中的负载均衡器。

有关详细信息，请参见在 [vRealize Automation Cloud Assembly](#) 的网络配置文件和蓝图设计中使用负载均衡器设置和 [vRealize Automation Cloud Assembly](#) 蓝图中的网络、安全性和负载均衡器示例。

安全组

部署蓝图时，蓝图的网络配置文件中的安全组将应用于已置备的计算机网卡。对于特定于 **Amazon Web Services** 的网络配置文件，网络配置文件中的安全组在“网络”选项卡上列出的网络所在的网络域 (VPC) 中可用。如果网络配置文件的“网络”选项卡上未列出任何网络，则会显示所有可用的安全组。

您可以使用安全组为 `private` 或 `outbound` 按需网络进一步定义隔离设置。安全组也适用于 `existing` 网络。

安全组将应用于部署中连接到与网络配置文件匹配的网络的所有计算机。由于蓝图中可能存在多个网络，每个网络都与不同的网络配置文件相匹配，因此您可以对不同的网络使用不同的安全组。

通过向现有安全组添加标记，将能够在蓝图 `Cloud.SecurityGroup` 组件中使用该安全组。安全组必须至少包含一个标记，否则无法在蓝图中使用。有关详细信息，请参见[安全资源](#)和 [vRealize Automation Cloud Assembly](#) 蓝图中的网络、安全性和负载均衡器示例。

有关网络配置文件、网络、蓝图和标记的更多信息

有关网络配置文件的详细信息，请参见此部分帮助中的其他主题以及 [WordPress 用例：添加网络配置文件](#)。

有关网络的详细信息，请参见[网络资源](#)。

有关蓝图中示例网络组件代码的示例，请参见 [vRealize Automation Cloud Assembly](#) 蓝图中的网络、安全性和负载均衡器示例。

有关示例网络自动化 workflow，请参见 [Cloud Assembly](#) 和 [NSX](#) 的网络自动化。

有关标记和标记策略的详细信息，请参见[如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署](#)。

在 vRealize Automation Cloud Assembly 中使用网络和网络配置文件中的 IP 地址

使用网络和网络配置文件设置，您可以控制在 [vRealize Automation Cloud Assembly](#) 蓝图和部署中使用网络 IP 地址的方式。

通过使用网络配置文件，您可以为包含静态、DHCP 或混合使用静态和 DHCP IP 地址设置的现有网络域定义子网。

您可以使用[网络策略](#)选项卡定义子网并指定 IP 地址设置。有关详细信息，请参见[了解有关 vRealize Automation Cloud Assembly 中的网络配置文件的更多信息](#)。

vRealize Automation Cloud Assembly 网络中的 IPv4 和 IPv6 支持

vRealize Automation Cloud Assembly 网络支持纯 IPv4 或双堆栈 IPv4 和 IPv6。当前不支持纯 IPv6。

虽然所有云帐户和集成类型均支持纯 IPv4，但仅 vSphere 云帐户及其端点支持双堆栈 IPv4 和 IPv6。

当前不支持将 IPv6 用于负载均衡器、NSX 按需网络或外部第三方 IPAM 提供程序。

外部 IPAM 提供程序支持

除了提供的内部 IPAM 支持外，您还可以使用外部 IPAM 提供程序为蓝图和部署中的网络动态或静态分配 IP 地址。

通过[基础架构 > 连接 > 添加集成 > IPAM](#) 菜单顺序创建的供应商特定的 IPAM 集成点支持外部 IPAM 提供程序（如 Infoblox）。

通过使用[网络策略 > 添加 IPAM IP 范围](#)页面上的[添加 IPAM IP 范围](#)选项，可以使用用于定义第三方 IPAM 提供程序地址信息的选项。

有关如何创建外部 IPAM 集成点的信息，请参见在[vRealize Automation 中配置外部 IPAM 集成点](#)。有关如何为特定 IPAM 供应商创建 IPAM 集成点的示例，请参见[提供程序特定的外部 IPAM 集成用例](#)。

在 vRealize Automation Cloud Assembly 中使用网络和网络配置文件

可以在 vRealize Automation Cloud Assembly 中使用网络和网络配置文件来帮助定义部署的网络置备行为。

在 vRealize Automation Cloud Assembly 中，您可以定义特定于云的网络配置文件。请参见[了解有关 vRealize Automation Cloud Assembly 中的网络配置文件的更多信息](#)。

网络类型

蓝图中的网络组件定义为以下 `networkType` 类型之一。

| 网络类型 | 定义 |
|-----------------------|------------------------------------------------------------------------------------------------------------------------|
| <code>existing</code> | 选择在底层云提供商（例如 vCenter、Amazon Web Services 和 Microsoft Azure）上配置的现有网络。outbound 按需网络需要现有网络。 可以在现有网络上定义一系列静态 IP 地址。 |
| <code>public</code> | 可以从 Internet 访问公共网络上的计算机。IT 管理委员会定义这些网络。对于允许通过公共网络传输网络流量的网络， <code>public</code> 网络的定义与 <code>existing</code> 网络的定义相同。 |
| <code>private</code> | 按需网络类型。 将网络流量限制为仅在已部署网络上的资源之间发生。它可防止入站和出站流量。在 NSX 中，它可以等同于按需 NAT 一对多。 |

| 网络类型 | 定义 |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| outbound | <p>按需网络类型。</p> <p>将网络流量限制为在部署中的计算资源之间发生，但也允许单向出站网络流量。在 NSX 中，它可以等同于具有外部 IP 的按需 NAT 一对多。</p> |
| routed | <p>按需网络类型。</p> <p>路由网络包含一个可路由的 IP 空间，该空间被划分为链接在一起的多个可用子网。置备路由网络和具有相同路由网络配置文件的虚拟机既可相互通信，也可与现有网络通信。</p> <p>路由网络是可用于 NSX-V 和 NSX-T 网络的按需网络类型。默认情况下，Microsoft Azure 和 Amazon Web Services 提供此连接。</p> <p>routed 网络只能在 Cloud.NSX.Network 网络组件中用于蓝图规范。</p> |

有关包含网络组件数据的已填充蓝图的示例，请参见 [vRealize Automation Cloud Assembly 蓝图中的网络、安全性和负载均衡器示例](#)。

网络方案

部署使用以下网络配置文件配置的蓝图时，以下是预期行为。

表 4-1. 网络方案

| 网络类型或方案 | 没有任何网络配置文件可用于云区域 | 多个网络配置文件可用于云区域 |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| 无网络 | <p>如果在蓝图中未指定任何网络，则将从与计算资源相同的置备区域中选择一个随机网络。</p> <p>标记为默认的网络优先。</p> <p>如果可用的置备区域中不存在任何网络，置备将失败。</p> | <p>从匹配的网络配置文件中选择一个网络。</p> <p>标记为默认的网络优先。</p> <p>如果没有任何网络配置文件符合条件，置备将失败。</p> |
| 现有网络 | <p>如果蓝图中的网络组件包含限制标记，则会使用这些限制来筛选可用网络的列表。蓝图网络组件中的限制标记将与网络标记进行匹配，如果存在网络配置文件限制标记，则还会与网络配置文件限制标记进行匹配。</p> <p>在筛选的网络列表中，从与计算资源相同的置备区域中选择单个网络。</p> <p>标记为默认的网络优先。</p> <p>根据限制进行筛选之后，如果置备区域中没有任何网络，置备将失败。</p> | <p>从匹配的网络配置文件中选择一个网络。</p> <p>标记为默认的网络优先。</p> <p>如果没有任何网络配置文件符合条件，置备将失败。</p> <p>可以使用网络限制根据预分配的标记筛选配置文件中的现有网络。</p> |

表 4-1. 网络方案（续）

| 网络类型或方案 | 没有任何网络配置文件可用于云区域 | 多个网络配置文件可用于云区域 |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 公共网络 | <p>如果网络具有限制，则使用这些限制来筛选设置了 <code>supports public IP</code> 属性的可用网络的列表。</p> <p>在筛选的网络列表中，从与计算资源相同的置备区域中选择一个随机网络。</p> <p>标记为默认的网络优先。</p> <p>根据限制进行筛选之后，如果置备区域中没有任何公共网络，置备将失败。</p> | <p>从匹配的网络配置文件中选择具有 <code>supports public IP</code> 属性的网络。</p> <p>标记为默认的网络优先。</p> <p>可以使用网络限制根据预分配的标记筛选配置文件中的现有公共网络。</p> |
| 专用网络 | 置备失败，因为专用网络需要网络配置文件中的信息。 | <p>将根据匹配的网络配置文件中的设置创建新的网络或新的安全组。</p> <p>可以使用网络限制标记来筛选网络配置文件和网络。</p> |
| 出站网络 | 置备失败，因为出站网络需要网络配置文件中的信息。 | <p>将根据匹配的网络配置文件中的设置创建新的网络或新的安全组。</p> <p>可以使用网络限制标记来筛选网络配置文件和网络。</p> |
| 按需路由网络 | 置备失败，因为路由网络需要网络配置文件中的信息。 | <p>对于 NSX-V，我们需要选择 DLR（分布式逻辑路由器）。</p> <p>对于 NSX-T 和 VMware Cloud on AWS，我们需要与专用网络和出站网络相似的按需设置。</p> |
| 使用现有网络或公共网络的 Wordpress 用例示例 | 按照现有网络或公共网络中所述进行置备。 | <p>有关使用现有网络和公共网络时的行为，请参见上述内容。</p> <p>请参见 WordPress 用例。</p> |
| 使用现有网络或公共网络和专用网络或出站网络的 Wordpress 用例示例 | 置备失败，因为网络需要网络配置文件中的信息。 | <p>有关专用网络和出站网络，请参见上述内容。</p> <p>请参见 WordPress 用例。</p> |
| 使用负载均衡器的 Wordpress 用例示例 | <p>置备失败，因为负载均衡器需要网络配置文件中的信息。</p> <p>如果存在现有负载均衡器，则可以进行置备。</p> | <p>根据网络配置文件配置创建新的负载均衡器。</p> <p>可以指定网络配置文件中已启用的现有负载均衡器。</p> <p>如果您请求现有负载均衡器，但网络配置文件中的所有负载均衡器均不满足限制，置备将失败。</p> <p>请参见 WordPress 用例。</p> |

在 vRealize Automation Cloud Assembly 的网络配置文件和蓝图设计中使用负载均衡器设置

可以根据您的 **NSX-V** 或 **NSX-T** 网络配置文件配置，在 vRealize Automation Cloud Assembly 蓝图中配置负载均衡器。

负载均衡器选项取决于蓝图的网络配置文件中的设置以及 **NSX-T** 或 **NSX-V** 源应用程序中的设置。

NSX-T 网络和负载均衡器选项

NSX-T 对每个第 1 层逻辑路由器支持一个负载均衡器服务。您的负载均衡器选项取决于蓝图中的负载均衡器组件与之关联的网络。

■ 按需出站网络

如果负载均衡器计算连接到按需出站网络，则会为按需网络的第 1 层路由器创建负载均衡器。

■ 按需专用网络

按需网络 VIP 必须与源外部网络的 VIP 关联。

如果负载均衡器计算资源连接到按需专用网络，将创建新的第 1 层路由器并将其连接到网络配置文件中指定的第 0 层路由器。负载均衡器随后将连接到第 1 层路由器。如果 VIP 位于外部网络中，则启用第 1 层路由器 VIP 通告。

■ 现有网络

如果负载均衡器连接到现有网络，将使用现有网络的第 1 层路由器创建负载均衡器。仅当没有任何负载均衡器服务连接到第 1 层路由器时，才会创建新的小型负载均衡器服务。如果负载均衡器服务已存在，则新的虚拟服务器会连接到该服务。如果现有网络未连接到第 1 层路由器，将创建新的第 1 层路由器并将其连接到网络配置文件中指定的第 0 层路由器，并且第 1 层路由器 VIP 通告未启用。

■ 具有新的安全组的按需网络

由于计算机连接到现有网络并且使用安全组创建隔离，因此，此选项类似于在现有网络中创建的负载均衡器。区别在于，为了启用数据路径，会将第 1 层上行链路端口 IP 添加到隔离安全组中。

如果不再使用，将销毁为按需网络创建的负载均衡器服务。销毁负载均衡器时，还会销毁监控器、池、应用程序配置文件和 VIP。不会销毁为外部网络创建的负载均衡器服务。

通过使用蓝图组件面板中可用的云不可知负载均衡器组件，可以在蓝图中指定 NSX-T 负载均衡器设置。

要了解有关在 NSX-T 网络部署中使用负载均衡器的更多信息，请参见 VMware 博客帖子 [vRA Cloud Assembly 负载均衡器与 NSX-T 配合使用深入探讨](#)。

NSX-V 网络和负载均衡器选项

基于使用新的网络子网还是安全组创建隔离网络，您可以将负载均衡器配置为双臂或单臂。

通过使用蓝图组件面板的 NSX 部分中可用的负载均衡器组件，可以在蓝图中指定 NSX-V 负载均衡器设置。

用于创建双臂负载均衡器的工作流是：

- 1 创建服务 Edge。
- 2 将服务 Edge 的上行链路接口连接到公用网络。
- 3 将下行链路接口连接到隔离（出站）网络。
- 4 为负载均衡器分配网络配置文件静态 IP 范围内的静态 IP 地址。
- 5 配置负载均衡器。
- 6 配置防火墙。

7 配置默认网关。

用于创建单臂负载均衡器的工作流是：

- 1 创建服务 Edge。
- 2 作为上行链路连接从预留中选择的网络。
- 3 为负载均衡器分配网络配置文件静态 IP 范围内的静态 IP 地址。
- 4 配置负载均衡器。
- 5 配置防火墙。
- 6 配置默认网关。

如何添加负责不同需求的 vRealize Automation Cloud Assembly 存储配置文件

vRealize Automation Cloud Assembly 存储配置文件描述要部署的存储的类型。

存储通常根据服务级别或成本、性能或用途（例如备份）等特性进行分析。

选择**基础架构 > 配置 > 存储配置文件**，然后单击**新建存储配置文件**。

了解有关 vRealize Automation Cloud Assembly 存储配置文件的更多信息

云帐户区域包含存储配置文件，云管理员可以使用存储配置文件为该区域定义存储。

存储配置文件包含磁盘自定义，以及用于按能力标记标识存储类型的方法。标记随后将与置备服务请求限制进行匹配，以在部署时创建所需的存储。

存储配置文件在特定于云的区域下进行组织。一个云帐户可能具有多个区域，每个区域下可能有多个存储配置文件。

可以进行不受供应商约束的放置。例如，将三个不同的供应商帐户可视化，且每个帐户分别具有一个区域。每个区域包含具有 **fast** 能力标记的存储配置文件。在置备时，无论资源由哪个供应商云提供，包含 **fast** 硬性限制标记的请求都将查找匹配的 **fast** 能力。随后在创建已部署的存储项期间，匹配将应用关联的存储配置文件中的设置。

注 不同的云存储可能具有不同的性能特性，但标记了该云存储的管理员仍将其视为 **fast** 存储。

添加到存储配置文件的能力标记不应标识实际资源目标。相反，能力标记描述存储类型。有关实际资源的更多信息，请参见[存储资源](#)。

如何使用标记来管理 vRealize Automation Cloud Assembly 资源和部署

标记是 vRealize Automation Cloud Assembly 的一个关键组件，它通过匹配功能和限制来辅助部署的放置。您必须有效地了解和实施标记，才能更好地利用 vRealize Automation Cloud Assembly。

从根本上说，标记是添加到 vRealize Automation Cloud Assembly 项目的标签。您可以创建适用于您的组织和实施的任何标记。但是标记的功能比标签多得多，因为它们控制 vRealize Automation Cloud Assembly 使用资源和基础架构的方式和位置以生成可部署服务。标记还支持在 Cloud Assembly 中进行管治。

标记结构

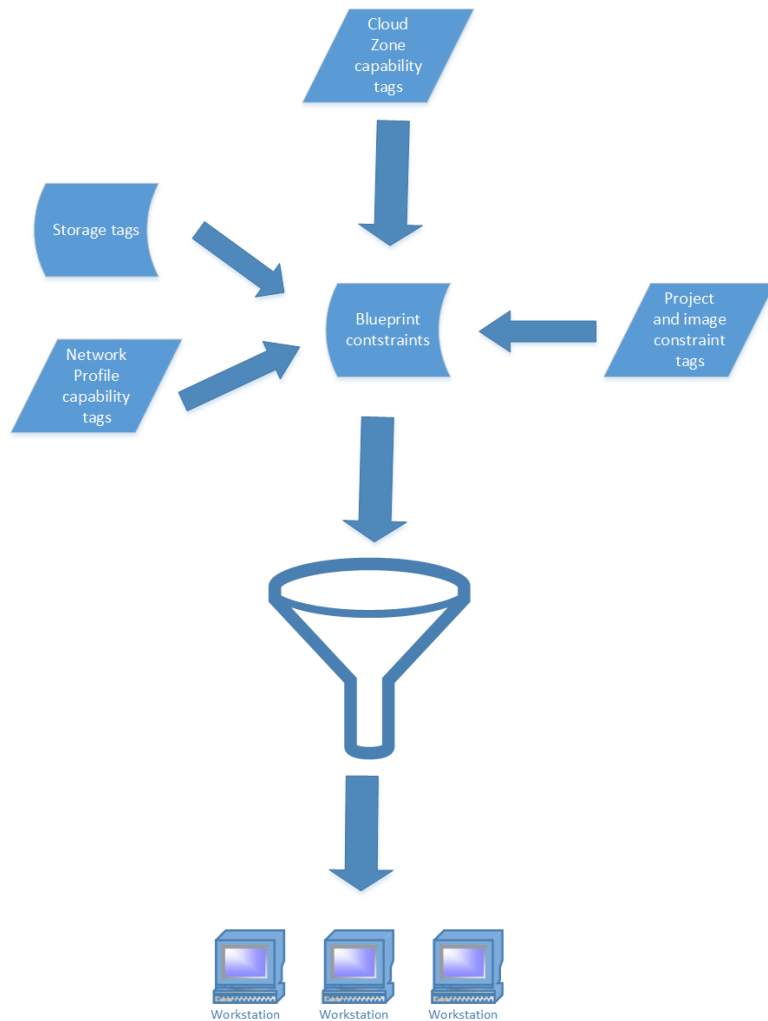
在结构上，标记必须遵循 `name:value` 的对约定，但在其他方面其结构在很大程度上是自由形式。在 vRealize Automation Cloud Assembly 中，所有标记看起来都是相同的，并且标记功能由上下文确定。

例如，基础架构资源上的标记主要用作功能标记，因为 vRealize Automation Cloud Assembly 使用它们将资源与部署进行匹配。其次，它们还会标识资源。

标记功能

vRealize Automation Cloud Assembly 中标记的主要功能是使用功能和限制来配置部署。放置在云区域、网络 and 存储配置文件以及各个基础架构资源上的功能标记定义了所需的部署功能。云管理员在项目上放置的限制标记使它们能够对这些项目执行一种形式的管治。这些限制标记将添加到以蓝图表示的其他限制中。

在置备期间，vRealize Automation Cloud Assembly 在蓝图中将这些功能与限制（也表示为标记）相匹配，以定义部署配置。这种基于标记的功能和限制功能用作 vRealize Automation Cloud Assembly 中部署配置的基础。例如，您可以使用标记使基础架构仅在特定区域的 PCI 资源上可用。



从辅助角度来看，标记还有助于搜索和识别存储和网络项目以及其他基础架构资源。

例如，假设您正在设置云区域，并且有许多可用的计算资源。如果您正确标记了计算资源，那么您可以使用“云区域”页面的“计算”选项卡上的搜索功能筛选与该特定云区域关联的资源。

此外，“管理标记”页面和资源配置页面中包含的搜索功能可用于按标记名称查找项目。为这些项目使用逻辑和人工可读标记是方便执行此搜索和识别功能的关键。

外部标记

vRealize Automation Cloud Assembly 也可能包含外部标记。这些标记是通过与 vRealize Automation Cloud Assembly 实例关联的云帐户自动导入的。这些标记可能是从 vSphere、AWS、Azure 或其他外部软件产品导入的。导入后，这些标记可与用户创建的标记相同的方式使用。

管理标记

您可以使用 vRealize Automation Cloud Assembly 中的“管理标记”页面来监控和管理标记库。您也可以在此页面上创建标记。此外，“管理标记”页面是唯一可以查看和标识外部标记的页面。



标记策略

为了最大程度地减少混淆，在 vRealize Automation Cloud Assembly 中创建标记之前，请设计适当的标记策略和标记约定，以便创建和使用标记的所有用户了解其含义以及使用方法。请参见[创建标记策略](#)。

创建标记策略

必须根据组织的 IT 结构和目标精心规划并实施相应的标记策略，以最大程度提高 Cloud Assembly 的功能并最大程度减少混淆。

虽然标记有几个常见的作用，但您的标记策略必须适合您的组织需求、结构和目标。

标记的最佳做法

有效标记策略的一些常规特征：

- 设计并实施与业务结构相关的标记的统一计划，并将此计划传达给所有适用的用户。计划必须支持您的部署需求，使用清晰的人工可读语言，并且所有适用的用户均可理解。
- 对标记使用简单、明确和有意义的名称和值。例如，存储和网络项的标记名称应该是清晰和连贯的，使用户能够轻松了解他们为已部署的资源选择或检查什么样的标记分配。
- 尽管可以使用不含值的名称创建标记，但最佳做法是为每个标记名称创建适用的值，这种做法更为合适，因为其他用户能够更清楚地了解标记的使用情况。

标记实施

绘制基本标记策略的主要注意事项。以下列表显示了在映射策略时需要考虑的典型注意事项。请注意，这些注意事项具有代表性，但并不是最终确定的。您可能具有与用例高度相关的其他注意事项。特定策略必须适合特定用例。

- 您将部署到多少个不同的环境。通常，您将创建标记来表示每个环境。
- 计算资源是如何构造并用于支持部署的。
- 您将部署到多少个不同的区域或位置。通常，您将在配置文件级别创建标记来表示每个不同的区域或位置。
- 有多少种不同的存储选项可用于部署，您希望如何描述它们的特征。这些选项应该用标记来表示。
- 对网络连接选项进行分类并创建标记以包含所有适用的选项。

- 典型的部署变量。例如，您将部署到多少个不同的环境。通常情况下，许多组织至少具有测试、开发和生产环境。您会想要创建和协调可以匹配的蓝图限制标记和云区域功能标记，以便轻松为一个或多个此类环境组建部署。
- 协调网络 and 存储资源上的标记，以便在使用这些标记的网络和存储配置文件环境中具有逻辑意义。资源标记可以对资源部署进行更为精细的控制。
- 将云区域和网络配置文件功能标记以及其他功能标记与蓝图限制标记进行协调。通常情况下，管理员将首先为云区域和网络配置文件创建功能标记，然后其他用户可以设计具有与这些功能标记匹配的限制的蓝图。

了解组织的重要注意事项之后，您可以规划适当的标记名称，以符合逻辑的方式处理这些注意事项。然后，创建策略大纲并将其提供给所有拥有创建或编辑标记特权的用户。

在开始此过程时，可以单独标记所有计算基础架构资源，这是一种有用的实施方法。如下所述，对与特定资源相关的标记名称使用逻辑类别。例如，可以将存储资源标记为 tier1、tier2 等。此外，还可以根据其操作系统（如 Windows、Linux 等）对计算资源进行标记。

标记资源之后，可以考虑采用最适合您需求的方法来为云区域和存储和网络配置文件创建标记。

在 vRealize Automation Cloud Assembly 中使用功能标记

在 vRealize Automation Cloud Assembly 中，可以使用功能标记定义用于基础架构组件部署的放置逻辑。功能标记功能更强大而且简洁，可对此类放置进行硬编码。

您可以在计算资源、云区域、映像和映像映射以及网络和网络配置文件上创建功能标记。用于创建这些资源的页面包含用于创建功能标记的选项。或者，也可以使用 vRealize Automation Cloud Assembly 中的“管理标记”页面创建功能标记。云区域和网络配置文件上的功能标记会影响这些区域或配置文件中的所有资源。存储或网络组件上的功能标记仅影响应用了这些标记的组件。

通常，功能标记可定义诸如计算资源的位置、网络的适配器类型或存储资源的层级别等内容。此外，还可定义环境位置或类型以及任何其他业务注意事项。与总体标记策略一样，应以逻辑方式组织功能标记。

vRealize Automation Cloud Assembly 在部署时将功能标记与云区域和蓝图的限制进行匹配。因此，在创建和使用功能标记时，您必须了解并计划创建适当的蓝图限制，以便按预期进行匹配。

例如，在 Wordpress 示例的“添加云区域”主题中，您为 OurCo-AWS-US-East 和 OurCo AWS-US-West 区域分别创建了开发标记和测试标记。这表示 OurCo-AWS-US-East 区域是一个开发环境，OurCo-AWS-US_West 区域是一个测试环境。通过将这些功能标记与相应的限制标记配对，可以将部署定向到所需的环境。

在 vRealize Automation Cloud Assembly 中使用限制标记

可以在 vRealize Automation Cloud Assembly 中向蓝图和其他各种组件添加限制标记，以与资源、云区域和配置文件中定义的功能匹配来生成适当的部署。

限制标记在 vRealize Automation Cloud Assembly 中的两个主要区域适用。第一个区域位于项目和映像中的配置侧。第二个区域位于蓝图中的部署侧。同时在两个区域中应用的限制将合并到蓝图中，从而形成一组部署要求。

限制标记在项目中如何工作

配置 Cloud Assembly 时，云管理员可以在项目和映像映射中应用限制标记。这样，云管理员可以直接在项目级别应用管治限制。在此级别添加的所有限制将应用到为适用项目请求的每个蓝图。

如果项目中的标记与蓝图中的标记冲突，将优先应用项目标记，从而使云管理员可以执行管治规则。例如，如果云管理员在项目创建 `location:london` 标记，而开发人员在蓝图中放置 `location:boston` 标记，将优先应用前一个标记，并且资源会部署到包含 `location:london` 标记的基础架构。

最多可以在项目中应用三项限制。项目限制可为硬性或软性。默认情况下，项目限制为硬性。使用硬性限制可以严格执行部署限制。如果不满足一项或多项硬性限制，部署将失败。软性限制提供了一种方法用于表示将选择的首选项（如果可用），但在不满足软性限制时，部署不会失败。

限制标记在蓝图中如何工作

在蓝图中，您可以将限制标记作为 YAML 代码添加到资源，以与云管理员在资源、云区域以及存储配置文件和网络配置文件中创建的相应能力标记匹配。此外，还有其他更复杂的选项可用于实施限制标记。例如，您可以使用变量在请求中填充一个或多个标记。这使您可以在请求时指定一个或多个标记。

使用 `tag` 标签在蓝图 YAML 代码中创建限制标记。项目中的限制标记将添加到蓝图中创建的限制标记。

vRealize Automation Cloud Assembly 支持简单字符串格式设置，以使在 YAML 文件中使用限制变得更容易：

```
[!tag_key[:tag_value][:hard|:soft]
```

默认情况下，vRealize Automation Cloud Assembly 将创建具有硬性执行的明确限制。在应用程序的剩余部分，标记值为可选，但建议提供标记值。

以下 WordPressWithMySQL 示例显示了 YAML 限制标记，这些标记表示计算资源的特定位置信息。

```
name: "wordpressWithMySQL"
components:
  mysql:
    type: "Compute"
    data:
      name: "mysql"
      # ... skipped lines ...
  wordpress:
    type: "Compute"
    data:
      name: "wordpress"
      instanceType: small
      imageType: "ubuntu-server-1604"
      constraints:
        - tag: "!location:eu:hard"
        - tag: "location:us:soft"
        - tag: "!pci"
      # ... skipped lines ...
```

有关如何使用蓝图的更多信息，请参见 [WordPress 用例：创建并扩展蓝图](#)。

硬性限制和软性限制在项目以及蓝图中如何工作

项目和蓝图中的限制都可为硬性或软性。上述代码片段显示了硬性限制和软性限制的示例。默认情况下，所有限制均为硬性。使用硬性限制可以严格执行部署限制。如果不满足一项或多项硬性限制，部署将失败。软性限制用于表示将应用的首选项（如果可用），但在不满足软性限制时，不会失败。

如果有一系列硬性限制和软性限制针对特定资源类型，则软性限制还可以用作 **Tie Breaker**。也就是说，如果多个资源满足硬性限制，则使用软性限制来选择在部署中使用的实际资源。

例如，最多可以采用网络项、存储项和可扩展性项的任意组合在项目中指定三项限制。此外，还可以选择每项限制为硬性还是软性。假设您创建具有 `location:boston` 标记的硬性存储限制。如果项目中的所有存储均不满足该限制，则任何相关的部署都将失败。

注 在项目和蓝图中，`failOnConstraintMergeConflict` 标志用于修改限制的行为。将该标志设置为 **true** 时，如果项目限制和蓝图限制之间存在冲突，请求将失败。如果该标志不存在或设置为 **false**，则项目限制的优先级高于蓝图限制。

标准标记

vRealize Automation Cloud Assembly 对部分部署应用标准标记，以支持对已部署的资源进行分析、监控和分组。

标准标记在 vRealize Automation Cloud Assembly 中具有唯一性。与其他标记不同，用户在配置部署的过程中不使用标准标记，并且不应用任何限制。这些标记会在置备期间自动应用到 **AWS**、**Azure** 和 **vSphere** 部署。这些标记存储为系统自定义属性，并且会在置备之后添加到部署中。

下面显示了标准标记的列表。

表 4-2. 标准标记

| 说明 | 标记 |
|-------------------------------|-------------------------------------------------------------------------------------------------------------|
| 组织 | <code>org: orgID</code> |
| 项目 | <code>project: projectID</code> |
| 请求者 | <code>requester: username</code> |
| 部署 | <code>deployment: deploymentID</code> |
| 蓝图参考（如果适用） | <code>blueprint: blueprintID</code> |
| 蓝图中的组件名称 | <code>blueprintResourceName: CloudMachine_1</code> |
| 放置限制：在蓝图或请求参数中应用，或者通过 IT 策略应用 | <code>constraints: key:value:soft</code> |
| 云帐户 | <code>cloudAccount: accountID</code> |
| 区域或配置文件，如果适用 | <code>zone: zoneID</code> 、 <code>networkProfile: profileID</code> 、 <code>storageProfile: profileID</code> |

vRealize Automation Cloud Assembly 如何处理标记

在 vRealize Automation Cloud Assembly 中，标记表示能力和限制，这些能力和限制决定置备过程中如何将资源分配给置备的部署以及分配到何处。

在解析标记时，vRealize Automation Cloud Assembly 使用特定顺序和层次结构来创建置备的部署。了解此过程的基础知识有助于高效地实施标记以创建可预测的部署。

以下列表汇总了能力和限制标记处理的高级别操作和顺序：

- 云区域按多个标准进行筛选，包括可用性和配置文件：区域所属地区的配置文件中的标记将在此时进行匹配。
- 区域能力标记和计算能力标记用于按硬性限制筛选其余云区域。
- 在筛选的区域中，将按优先级选择云区域。如果多个云区域具有相同优先级，则通过使用云区域能力和计算能力的组合匹配软性限制来对这些云区域进行排序。
- 选择云区域后，将通过匹配一系列筛选器（包括蓝图中表示的硬性限制和软性限制）来选择主机。

如何设置简单的标记结构

本主题介绍逻辑 vRealize Automation Cloud Assembly 标记策略的基本方法和选项。您可以使用这些示例作为实际部署的起点，也可以计划更符合需求的不同策略。

通常，云管理员是负责创建和维护标记的主要人员。

本主题引用 vRealize Automation Cloud Assembly 文档中其他主题所述的 **WordPress** 用例来说明如何向某些关键项目添加标记。本主题还介绍 **WordPress** 用例中显示的标记示例的可能替代方法和扩展。

有关 **WordPress** 用例的详细信息，请参见 [WordPress 用例](#)。

WordPress 用例介绍如何在云区域以及存储配置文件和网络配置文件上放置标记。这些配置文件类似于有组织的资源包。在配置文件上放置的标记应用于该配置文件中的所有项目。还可以创建标记并将其放置在存储资源和各个网络项目以及计算资源上，但这些标记仅应用于其放置到的特定资源。设置标记时，最佳做法通常是从标记计算资源开始并在以后再向配置文件和云区域添加标记。此外，还可以使用这些标记来筛选云区域的计算资源列表。

例如，虽然可以如此用例中所示在存储配置文件上放置标记，但也可以在各个存储策略、数据存储和存储帐户上放置标记。借助这些资源上的标记，可以对存储资源的部署方式进行更精细控制。在为准备部署进行的处理过程中，这些标记将作为配置文件标记后的下一个级别进行处理。

例如，您可以通过网络配置文件上放置标记 `region: eastern` 来配置典型客户场景。此标记将应用于该配置文件中的所有资源。然后可以在配置文件中的 `pri` 网络资源上放置标记 `networktype:pri`。具有东部和 `pri` 限制的蓝图将创建对东部地区使用此 `pri` 网络的部署。

步骤

1 以符合逻辑的适当方式标记计算基础架构资源。

必须以符合逻辑的方式标记计算资源，这点特别重要，以便可以在“创建云区域”页面的“计算”选项卡中使用搜索功能来查找计算资源。使用此搜索功能，可以快速筛选与云区域关联的计算资源。如果在配置文件级别标记存储和网络，可能不需要标记各个存储资源和网络资源。

- a 选择**资源 > 计算**，以查看已为 vRealize Automation Cloud Assembly 实例导入的计算资源。
- b 根据需要选择每个计算资源并单击**标记**，以向资源添加标记。如果需要，可以向每个资源添加多个标记。
- c 根据需要对存储资源和网络资源重复上一步。

2 创建云区域能力标记和网络配置文件能力标记。

可以为云区域和网络配置文件使用相同的标记，也可以为每个项目创建唯一的标记（如果这对您的实现更有意义）。

对于网络配置文件，可以在整个配置文件上放置标记，也可以在配置文件中的子网上放置标记。在配置文件级别应用的标记将应用于该配置文件中的所有组件，例如子网。子网上的标记仅应用于其放置到的特定子网。在处理标记的过程中，配置文件级别标记的优先级高于子网级别标记。

请参见 [WordPress 用例：添加云区域](#) [WordPress 用例：添加网络配置文件](#)，了解向云区域或网络配置文件添加标记的信息。

在此示例中，我们创建三个简单标记，这些标记在 vRealize Automation Cloud Assembly 云区域标记和网络配置文件标记的用例文档中通篇出现。这些标记标识配置文件组件的环境。

- zone:test
- zone:dev
- zone:prod

3 为存储组件创建存储配置文件标记。

通常，存储标记标识存储项目的性能级别（例如，第 1 层或第 2 层），或者标识存储项目的特性（例如 pci）。

有关向存储配置文件添加标记的信息，请参见 [WordPress 用例：添加存储配置文件](#)。

- usage:general
- usage:fast

结果

创建基本标记结构后，可以开始使用该标记结构，并根据需要添加或编辑标记以优化并扩展标记功能。

如何使用 vRealize Automation Cloud Assembly 中的资源

云管理员可以复查通过数据收集公开的 vRealize Automation Cloud Assembly 资源。云管理员可以使用功能标记对资源进行标记，以影响 vRealize Automation Cloud Assembly 蓝图的部署位置。

计算资源

云管理员可以查看通过数据收集公开的计算资源。在 vRealize Automation Cloud Assembly 中进行置备期间，云管理员可以选择直接将标记应用到资源，以标记适用于相应用途的功能。

网络资源

在 vRealize Automation Cloud Assembly 中，云管理员可以查看和编辑从项目的云帐户和集成中进行数据收集的网络资源。

添加云帐户后，数据收集会发现云帐户的网络和安全性信息，并使该信息可用于网络配置文件和其他选项。

网络是可用网络域或传输区域的特定于 IP 的组件。如果您是 Amazon Web Services 或 Microsoft Azure 用户，请将网络视为子网。

vRealize Automation Cloud Assembly **网络**页面包含如下信息：

- 在您的云帐户（例如，在 vCenter、NSX-V 或 Amazon Web Services 中）网络域外部定义的网络和负载均衡器。
- 由云管理员部署的网络和负载均衡器。
- 由云管理员定义或修改的 IP 范围和其他网络特性。
- 提供商特定外部 IPAM 集成中特定地址空间的外部 IPAM 提供程序 IP 范围。

有关网络的详细信息，请参见以下信息、**网络**页面上各种设置的标志帮助以及[了解有关 vRealize Automation Cloud Assembly 中的网络配置文件的更多信息](#)。

网络

您可以查看和编辑网络及其特性，例如添加标记或移除对公共 IP 访问的支持。您还可以在可用网络中定义新的 IP 范围并管理现有的 IP 范围。

对于现有网络，您可以通过选中网络的复选框并选择**管理 IP 范围**或**标记**来更改 IP 范围和标记设置。否则，您可以选择网络本身以编辑其信息。

标记提供了一种将相应网络（或网络配置文件）与蓝图中的网络组件相匹配的方法。网络标记将应用到该网络的每个实例，而不管网络可能驻留在哪个网络配置文件中。网络可以实例化到任意数量的网络配置文件中。无论网络配置文件的驻留方式如何，网络标记在使用网络的任何地方都与该网络相关联。蓝图与一个或多个网络配置文件相匹配后，蓝图中的其他组件会出现网络标记匹配。

IP 范围

选择 IP 范围，以定义或更改组织中特定网络的起始和结束 IP 地址。

默认网关不能包含在 IP 范围内。子网 IP 范围不能包含子网网关值。

如果为特定 IPAM 提供程序使用外部 IPAM 集成，则可以添加 IPAM IP 范围。此过程在[在 vRealize Automation Cloud Assembly 中配置网络和网络配置文件以使用 IPAM 提供程序值的整体外部 IPAM 集成](#)工作流程上下文中进行了介绍。

IP 地址

显示组织使用的已定义 IP 地址的状态（例如可用或已分配）。

负载均衡器

显示组织中帐户/区域云帐户的可用负载均衡器的相关信息。您可以打开并显示为每个可用负载均衡器配置的设置。还可以添加和移除负载均衡器的标记。

安全资源

添加云帐户后，数据收集会发现云帐户的网络和安全性信息，并使该信息可用于网络配置文件和其他选项。

安全组和防火墙规则支持网络隔离。

安全组

您可以查看在 vRealize Automation Cloud Assembly 中创建的按需安全组以及在源应用程序中创建的现有安全组，例如通过数据收集公开的 NSX-T 和 Amazon Web Services。

可以查看可用的安全组，并为选定的安全组添加或移除标记。还可以编辑所创建的按需安全组。云管理员可以打开和编辑安全组。蓝图作者可以将一个或多个安全组分配给计算机网卡，以控制蓝图部署的网络规则和其他安全方面。

在安全组组件的蓝图代码中使用防火墙规则时，

底层云帐户端点（如 NSX-V、NSX-T 或 Amazon Web Services 应用程序）中的现有安全组由 vRealize Automation Cloud Assembly 收集数据。现有安全组将在**来源**列中显示并归类为 Discovered。在 vRealize Automation Cloud Assembly 的蓝图或网络配置文件中创建的按需安全组将在**来源**列中显示并归类为 Managed by Cloud Assembly。只能看到可在您的组织中使用的安全组。

如果直接在源应用程序（例如源 NSX 应用程序）而非 vRealize Automation Cloud Assembly 中编辑现有安全组，则在 vRealize Automation Cloud Assembly 中不会显示更新，直到您在 vRealize Automation Cloud Assembly 中使用关联的云帐户或集成点运行数据收集。

云管理员可以将一个或多个标记分配给现有安全组，以便能够在蓝图中使用该安全组。安全组必须至少包含一个标记，否则无法在蓝图中使用。蓝图作者可以使用蓝图中的 Cloud.SecurityGroup 组件，通过使用标记限制来分配现有安全组。在蓝图中，必须将 Cloud.SecurityGroup 安全组组件应用于计算机网卡，以控制蓝图部署中的网络规则和其他安全方面。

有关管理员如何向现有安全组添加标记或如何从源应用程序中的安全组收集标记数据的信息，请参见[了解有关 vRealize Automation Cloud Assembly 中的网络配置文件的更多信息](#)。

有关蓝图作者如何在蓝图安全组件中使用标记的示例，请参见[vRealize Automation Cloud Assembly 蓝图中的网络、安全性和负载均衡器示例](#)。

如需蓝图代码示例，请参见[vRealize Automation Cloud Assembly 蓝图中的网络、安全性和负载均衡器示例](#)中的“对计算机网卡应用了限制标记的现有安全组”示例。

有关在网络配置文件中安全组的详细信息，请参见[了解有关 vRealize Automation Cloud Assembly 中的网络配置文件的更多信息](#)。

在网络配置文件和蓝图中定义安全组的方法

您可以通过以下任一方法利用安全组功能：

- 网络配置文件中指定的现有安全组

您可以将现有安全组添加到网络配置文件中。当蓝图使用该网络配置文件时，其计算机将成组并成为安全组的成员。此方法不要求您将安全组组件添加到蓝图。

- 与蓝图中的计算机组件关联的安全组组件

您可以将安全组组件拖放到蓝图中，并使用蓝图中现有安全组组件上和数据收集资源中现有安全组上的限制标记将安全组组件绑定到计算机网卡。

- NSX-T 应用程序中指定的 NSX-T 标记

- 您可以使用在蓝图中的网络组件上指定为限制的 NSX-T 标记，网络组件将通过该标记连接到蓝图中的计算机网卡。通过 NSX-T 标记，可以使用从预定义的 NSX-T 标记（即从 NSX-T 源端点收集的数据）来动态分组计算机。在 NSX-T 中创建 NSX-T 标记时，使用逻辑端口。

存储资源

云管理员可以使用存储资源及其功能，这些资源和功能是通过关联的云帐户进行 vRealize Automation Cloud Assembly 数据收集发现的。

存储资源功能通过通常源自源云帐户的标记公开。通过使用 vRealize Automation Cloud Assembly，云管理员可以选择将其他标记直接应用到存储资源。在置备时，附加标记可能会为匹配目的标记特定功能。

存储资源的功能会在 vRealize Automation Cloud Assembly 存储配置文件的定义中可见。请参见[了解有关 vRealize Automation Cloud Assembly 存储配置文件的更多信息](#)。

计算机资源

在 vRealize Automation 中，所有用户都可以查看通过数据收集公开的计算机资源。

项目中的所有计算机都将显示在“计算机”列表中。

与项目中的云帐户关联的未受管计算机将与受管计算机一样显示在此列表中。“来源”列指示计算机状态。

- 已发现 - 尚未载入的计算机。
- 已部署 - 已从 vRealize Automation 载入或置备的计算机。

您可以使用工作负载载入计划将未受管计算机引入 vRealize Automation 管理。

有关使用载入计划将未受管计算机引入 vRealize Automation 管理的信息，请参见[vRealize Automation Cloud Assembly 中的载入计划是什么](#)。

卷资源

在 vRealize Automation Cloud Assembly 中，所有用户都可以查看卷资源。

vRealize Automation Cloud Assembly 显示两个来源的卷或逻辑驱动器：

- 通过源云帐户数据收集发现的卷

■ 与由 vRealize Automation Cloud Assembly 置备的工作负载关联的卷

您可以根据卷或逻辑驱动器查看容量和功能。该列表还会公开源自源云帐户的功能标记或添加到 vRealize Automation Cloud Assembly 本身中的功能标记。

了解有关 vRealize Automation Cloud Assembly 中的资源的更多信息

vRealize Automation Cloud Assembly 可以公开有关数据收集资源的其他信息，例如成本。

vRealize AutomationvRealize Automation Cloud Assembly 中数据收集的工作原理

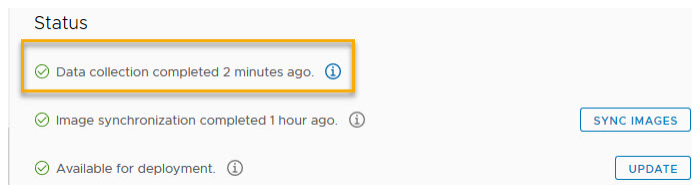
执行初步数据收集后，将每隔 10 分钟自动执行一次资源数据收集。数据收集时间间隔不可配置，且无法手动启动数据收集。

您可以在现有云帐户页面的“状态”部分中发现有关该云帐户的资源数据收集和映像同步的信息。方法是：选择**基础架构 > 连接 > 云帐户**，然后在所选的现有云帐户上单击**打开**。

您可以打开现有的云帐户，并在其页面的**状态**部分中查看其关联的端点版本。如果关联的端点已升级，则会在数据收集期间发现新的端点版本，并反映在云帐户页面的**状态**部分中。

资源数据收集

数据收集每隔 10 分钟执行一次。每个云帐户都会显示其数据收集上次完成的时间。

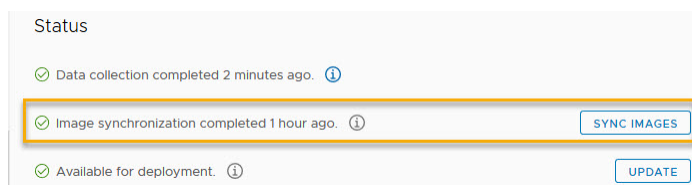


映像数据收集

映像同步每隔 24 小时执行一次。您可以为某些云帐户类型启动映像同步。要启动映像同步，请打开云帐户（**基础架构 > 云帐户**，然后选择并打开现有云帐户），然后单击**同步映像**按钮。没有用于 NSX 云帐户的映像同步选项。

注 映像在内部分类为公共映像或专用映像。公共映像是共享的，不特定于特定的云订阅或组织。专用映像不进行共享，并且特定于特定订阅。公共映像和专用映像每 24 小时自动同步一次。通过云帐户页面上的选项，您可以触发专用映像的同步。

云帐户页面将显示映像同步的上次完成时间。



为了促进部署中的容错能力和高可用性，每个 NSX-T 数据中心端点都代表由 3 个 NSX Manager 构成的集群。有关相关信息，请参见在 [vRealize Automation Cloud Assembly 中创建 NSX-T 云帐户](#)。

云帐户和载入计划

创建云帐户时，将对与该云帐户关联的所有计算机收集数据，然后在**基础架构 > 资源 > 计算机**页面中显示这些计算机。如果云帐户包含在 vRealize Automation Cloud Assembly 外部部署的计算机，您可以使用载入计划允许 vRealize Automation Cloud Assembly 管理计算机部署。

有关添加云帐户的信息，请参见[将云帐户添加到 vRealize Automation Cloud Assembly](#)。

有关载入非受管计算机的信息，请参见[vRealize Automation Cloud Assembly 中的载入计划是什么](#)。

什么是 vRealize Automation Cloud Assembly 部署成本

项目管理员和云管理员通常负责管理成本。vRealize Automation Cloud Assembly 成本管理可帮助您了解各个部署的成本影响，以便帮助您管理资源。

必须在 vRealize Operations 中将成本配置为使用 vRealize Automation 并启用，才能查看成本核算。对 vRealize Operations 配置 vRealize Automation 时，请确保这两个应用程序都设置为同一时区。要在 vRealize Operations 中配置时区，请启用 SSH 并登录到每个 vRealize Operations 节点，编辑 `$ALIVE_Base/user/conf/analytics/advanced.properties` 文件，然后添加 `timeZoneUseInMeteringCalculation = <time zone>`。

随着时间的推移，部署成本在部署卡视图上显示为“月累计成本”，每月月初重置为零。部署详细信息中提供了组件成本细目。在部署级别提供此信息可告知云管理员相关信息，但它还有助于成员了解其工作可能对预算和长期开发产生的影响。

由于类似的业务原因，您可能需要了解整个项目的总成本。项目级别的成本反映项目团队所有部署的全部成本。要查看完整的项目成本，请选择**基础架构 > 项目 > 选择项目 > 成本**。结果将按部署显示总成本和成本细目。

注 项目成本仅包括私有云工作负载的成本。如果项目包含属于公有云的部署，则这些部署的成本不会包括在项目成本中。

如何计算成本

您在部署级别看到的计算资源和存储资源的初始成本基于行业标准基准费率，然后根据时间计算。将成本费率应用于主机，服务会计算 CPU 和内存费率。服务器每 24 小时重新计算一次成本。

您也可以在**基础架构 > 集成 > vROPs 端点 >**的“vROPs 端点”页面上随时手动刷新成本服务器。在 vCenter Server 部分中，单击**同步**。使用**同步**选项手动刷新成本服务器时，将为组织中所有项目重新计算成本。根据您的组织有多少个项目，此过程可能会耗费大量资源，并且需要一些时间。

有关所支持资源的列表，请参见[vRealize Automation Cloud Assembly 中的计费组件类型列表](#)。

如何自定义计算的费率

使用一段时间内的基准值后，云管理员可能会发现您的实际成本已打折。您可以调整用于计算成本的费率，使其更好地反映您的业务实践。

如果进行调整，则计算更改将在下次服务运行计算时反映出来。服务器每 24 小时重新计算值。

vRealize Automation Cloud Assembly 中的计费组件类型列表

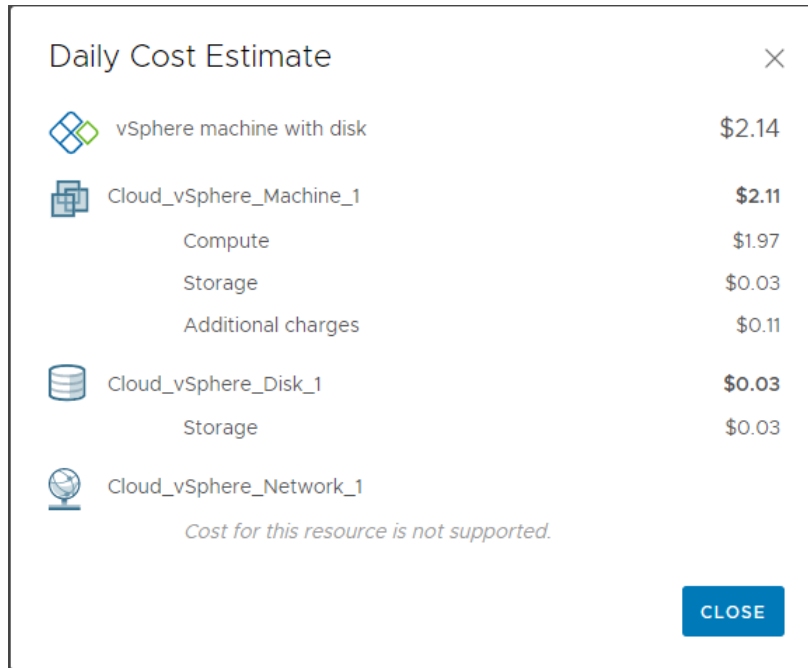
vRealize Automation Cloud Assembly 提供了以下蓝图组件类型的基准成本信息。

表 4-3. 计费组件类型

| 蓝图组件类型 | 服务名称/对象类型 | 蓝图资源类型 | 注释 |
|---------|-------------|-----------------------|----------------------------------------|
| 云不可知 | 计算机 | Cloud.Machine | 如果为不可知计算机配置了 vSphere，则可以查看部署成本。 |
| | 磁盘 | Cloud.Volume | 如果将不可知磁盘连接到配置了 vSphere 的虚拟机，则可以查看部署成本。 |
| vSphere | vSphere 计算机 | Cloud.vSphere.Machine | 使用特定于云的蓝图的部署。 |
| | vSphere 磁盘 | Cloud.vSphere.Disk | 使用连接到虚拟机的云特定蓝图进行部署。 |

如何估算部署成本

在部署目录项之前，您可以使用前期成本进行部署的成本估算。



部署的前期成本是在部署给定目录项之前基于资源分配为该目录项得出的每日成本估算。部署目录项后，您可以在**部署和基础架构 > 项目**选项卡上以前期成本总计的形式查看当月至今成本。针对私有云资源（如 vSphere 计算机和 vSphere 磁盘、Cloud Assembly 目录项及为私有云配置 vCenter 的云不可知项）支持前期成本。

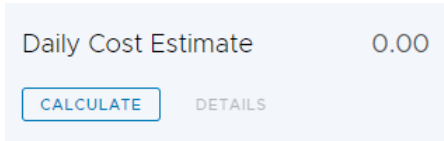
注 针对公有云资源或非 vSphere 计算机或磁盘私有云资源不支持前期成本成本。

前提条件

要在 vRealize Automation Cloud Assembly 中查看前期成本，必须配置 vRealize Operations 集成端点：启用成本核算并预设货币。

步骤

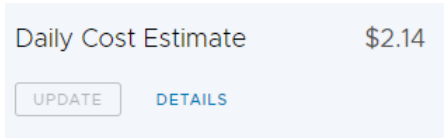
- 1 从“目录”中选择目录项，然后单击**请求**。



Daily Cost Estimate 0.00

CALCULATE DETAILS

- 2 输入目录项请求的详细信息，然后单击**计算**。



Daily Cost Estimate \$2.14

UPDATE DETAILS

- 3 （可选）单击**详细信息**以在“每日成本估算”窗口中查看成本细目。

后续步骤

如果每日成本估算可接受，请单击**提交**以继续执行部署请求。

如何确定部署看起来成本高昂的原因

管理成本时，项目管理员和云管理员可能会发现部署成本高昂。

部署的成本由其资源计算。有时，单个部署中的资源组合可能会增加成本，从而使部署对于您的业务需求来说成本高昂。因此，如果您的项目包含一个或多个成本高昂的部署，则总项目成本也会增加。

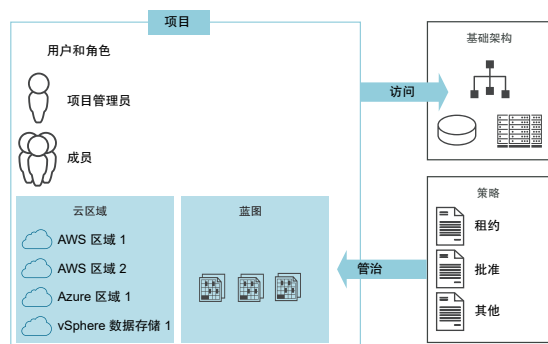
确定部署是否对于您自己的业务需求成本高昂且独一无二。您可以通过打开部署的详细信息并单击**成本**来查看部署的成本细目。要修改部署的成本，您必须编辑其蓝图的资源组成并重新部署。

添加和管理 vRealize Automation Cloud Assembly 项目

5

项目控制哪些用户有权访问 vRealize Automation Cloud Assembly 蓝图，并控制蓝图将部署到何处。您可以使用项目来组织和管治用户可执行的操作，以及用户可在云计算基础架构中部署蓝图的云区域。

云管理员设置项目，他们可以向其中添加用户和云区域。任何要创建和部署蓝图的用户都必须是至少一个项目的成员。



本章讨论了以下主题：

- 如何为我的 vRealize Automation Cloud Assembly 开发团队添加项目
- 了解有关 vRealize Automation Cloud Assembly 项目的更多信息

如何为我的 vRealize Automation Cloud Assembly 开发团队添加项目

您可以创建一个项目，向其添加成员和云区域，以便项目成员可以将其蓝图部署到关联的区域。作为 vRealize Automation Cloud Assembly 管理员，您可以为开发团队创建一个项目。然后，您可以分配项目管理员，也可以作为项目管理员进行操作。

创建蓝图时，首先需要选择要与该蓝图相关联的项目。该项目必须先存在，您才能创建蓝图。

确保您的项目满足开发团队的业务需求。

- 项目是否提供支持团队目标的资源。有关基础架构资源和项目如何支持蓝图的示例，请参见 [WordPress 用例](#)。

此过程基于创建仅包含基本配置的初始项目。当您的开发团队创建和部署蓝图时，您可能要修改项目。您可以添加限制、自定义属性和其他选项，以提高部署效率。请参见 [了解有关 vRealize Automation Cloud Assembly 项目的更多信息](#) 中包含的文章。

前提条件

- 确认您配置了云区域。请参见第 4 章 构建您的 vRealize Automation Cloud Assembly 资源基础架构。
- 对于添加为此项目的云区域的区域，确认您为其配置了映射和配置文件。请参见第 4 章 构建您的 vRealize Automation Cloud Assembly 资源基础架构。
- 确认您拥有执行此任务所需的权限。请参见 vRealize Automation Cloud Assembly 用户角色是什么。
- 确定将指定为项目管理员的用户。要了解项目管理员可以在 vRealize Automation Cloud Assembly 中执行的操作，请参见 vRealize Automation Cloud Assembly 用户角色是什么。
- 如果要将 Active Directory 组添加到项目，请确认您已为组织配置了 Active Directory 组。请参见《管理 vRealize Automation》中的在 vRealize Automation 中编辑组角色分配。如果这些组未同步，则当您尝试将它们添加到项目时，这些组不可用。

步骤

- 1 选择**基础架构 > 配置 > 项目**，然后单击**新建项目**。
- 2 输入项目名称。
- 3 单击**用户**选项卡。
 - a 要使项目成员执行的部署仅可由所有者访问，请禁用**部署共享**。
 - b 添加具有已分配角色的用户。
- 4 单击**置备**选项卡，然后添加一个或多个云区域。

云区域必须包含支持用户所部署蓝图的资源。
- 5 单击**创建**。
- 6 要使用项目云区域测试您的项目，请单击“项目”页面上的**测试配置**。

模拟将针对项目云区域资源运行标准化的假设部署测试。如果失败，您可以查看详细信息，并更正资源配置。

后续步骤

开始使用蓝图。请参见第 6 章 设计 vRealize Automation Cloud Assembly 部署。

了解有关 vRealize Automation Cloud Assembly 项目的更多信息

项目是蓝图和资源之间的连接器。您对它们的工作方式以及如何使它们为您工作了解得越多，您的 vRealize Automation Cloud Assembly 开发和部署过程就会越有效。

使用 vRealize Automation Cloud Assembly 项目标记和自定义属性

作为管理员，当项目的要求与 vRealize Automation Cloud Assembly 蓝图不同时，您可以添加项目级别的监管限制或自定义属性。除了限制标记外，您还可以添加资源标记，可以在置备过程中将这些标记添加到已部署资源以便能够管理资源。

什么是项目资源标记

项目资源标记作为标准化标识标记运行，可用于管理已部署的资源并确保合规性。

在项目中定义的资源标记将添加到作为该项目的一部分部署的所有组件资源。然后，您可以使用标准标记通过其他应用程序管理资源。

例如，作为云管理员，您希望使用诸如 CloudHealth 的应用程序来管理成本。您将 `costCenter:eu-cc-1234` 标记添加到专用于开发欧盟人力资源工具的项目中。当项目团队从该项目部署时，该标记将添加到已部署的资源中。然后，您可以配置成本计算工具来标识和管理包含此标记的资源。具有其他成本中心的其他项目将具有与此键匹配的替代值。

什么是项目限制标记

项目限制作为管治定义运行。项目限制是一个 `key:value` 标记，用于定义部署请求在项目云区域中使用或避开的资源。

部署过程将查找与项目限制匹配的网络和存储的标记，然后根据匹配的标记进行部署。

可扩展性限制用于指定要与可扩展性工作流配合使用的 vRealize Orchestrator 集成实例。

配置项目限制时，可以考虑以下格式。

- **key:value** 和 **key:value:hard**。如果必须在具有匹配的能力标记的资源上置备蓝图，请使用此标记，任一格式皆可。如果找不到匹配的标记，部署过程将失败。例如，由项目成员部署的蓝图必须在符合 PCI 的网络上置备。您将使用 `security:pci`。如果在项目云区域中找不到任何网络，部署将失败，以确保没有任何不安全部署。
- **key:value:soft**。如果您首选匹配的资源但希望部署过程继续而不失败并且可以接受具有不匹配标记的资源，请使用此标记。例如，您希望项目成员将其蓝图部署到成本更低的存储，但不希望存储可用性干扰项目成员的部署能力。您将使用 `tier:silver:soft`。如果项目云区域中不存在具有 `tier:silver` 标记的存储，蓝图仍会部署到其他存储资源。
- **!key:value**。如果要避免部署到具有匹配的标记的资源，请使用此标记，指定为硬性或软性皆可。

重要的是，项目限制标记具有比蓝图限制标记更高的优先级，而且项目限制标记会在部署时替代蓝图限制标记。如果您的蓝图不允许出现这种情况，可以在该蓝图中使用

`failOnConstraintMergeConflict:true`。例如，您的项目有一个网络 `loc:london` 限制，但蓝图是 `loc:mumbai`，您希望部署失败并显示限制冲突消息，而不希望项目位置优先，则可以添加类似下例的属性。

```
constraints:
  - tag: 'loc:mumbai'
failOnConstraintMergeConflict:true
```

如何使用项目自定义属性

可以使用项目自定义属性进行报告、触发和填充可扩展性操作和工作流，以及替代蓝图级别属性。

通过将自定义属性添加到部署中，您可以使用用户界面中的值或使用 API 检索该属性来生成报告。

可扩展性还可以将自定义属性用于可扩展性订阅。

您可能想要针对某个项目更改蓝图的特定属性值。您可以提供替代名称和值作为自定义属性。

vRealize Automation Cloud Assembly 项目在部署时的工作方式

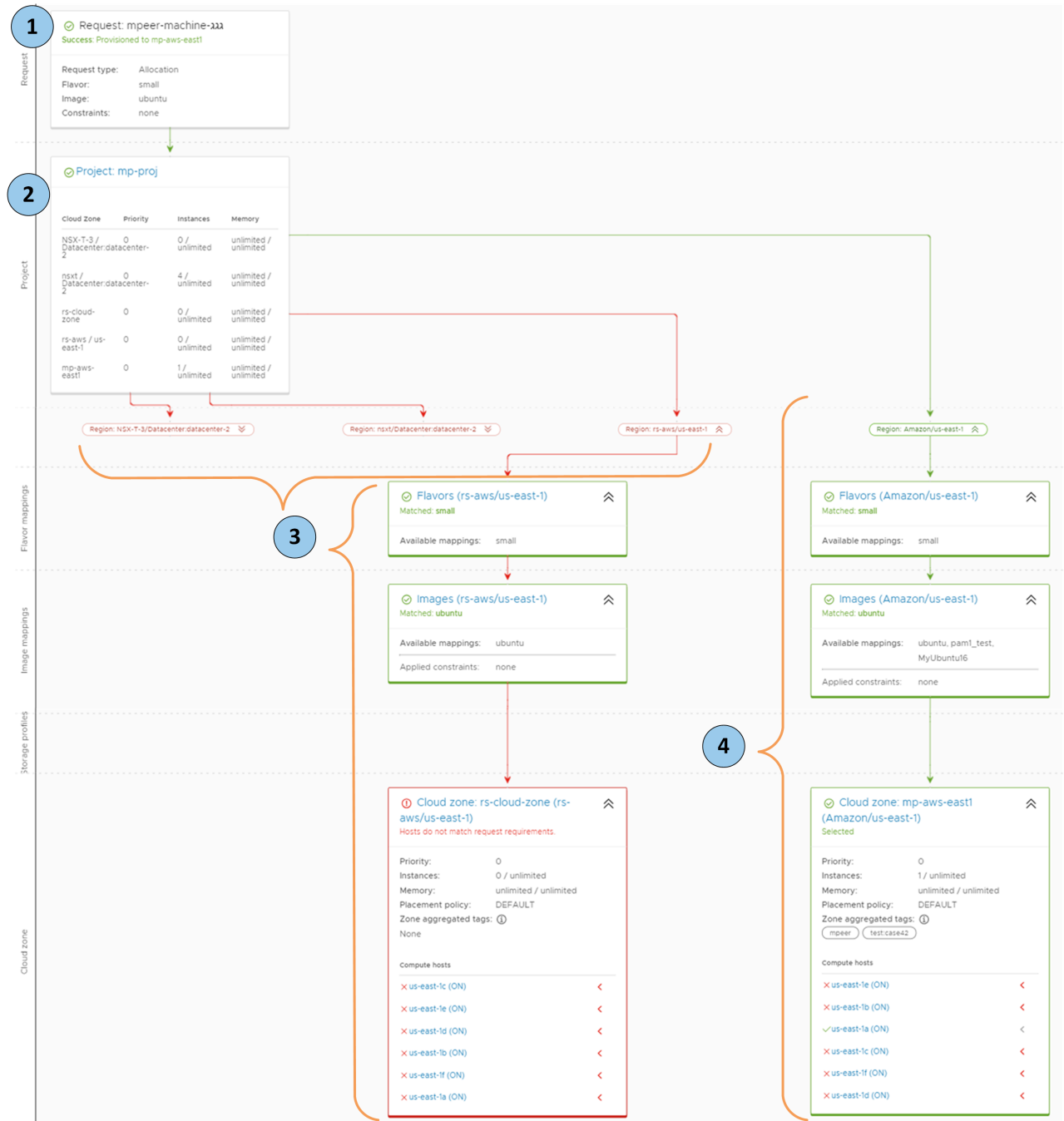
项目控制用户对云区域的访问和用户对已置备资源的所有权。无论您是云管理员还是蓝图开发人员，您都必须了解项目在部署时的工作方式，以便您能够管理您的部署并对任何问题进行故障排除。

作为为各种团队设置项目的云管理员，您必须了解项目如何确定蓝图组件的部署位置。了解这一点可帮助您创建支持蓝图开发人员的项目并对失败的部署进行故障排除。

创建蓝图时，首先将其与项目关联。在部署时，将根据项目云区域评估蓝图要求，以查找最佳部署位置。

以下工作流说明了该流程。

- 1 您可以提交蓝图部署请求。
- 2 该项目将评估蓝图和项目要求，例如特定实例、映像和限制标记。将要求与项目云区域进行比较，以找到支持这些要求的区域。
- 3 这些区域没有支持该请求的资源。
- 4 此云区域支持请求要求，并且蓝图将部署到此云区域帐户区域。



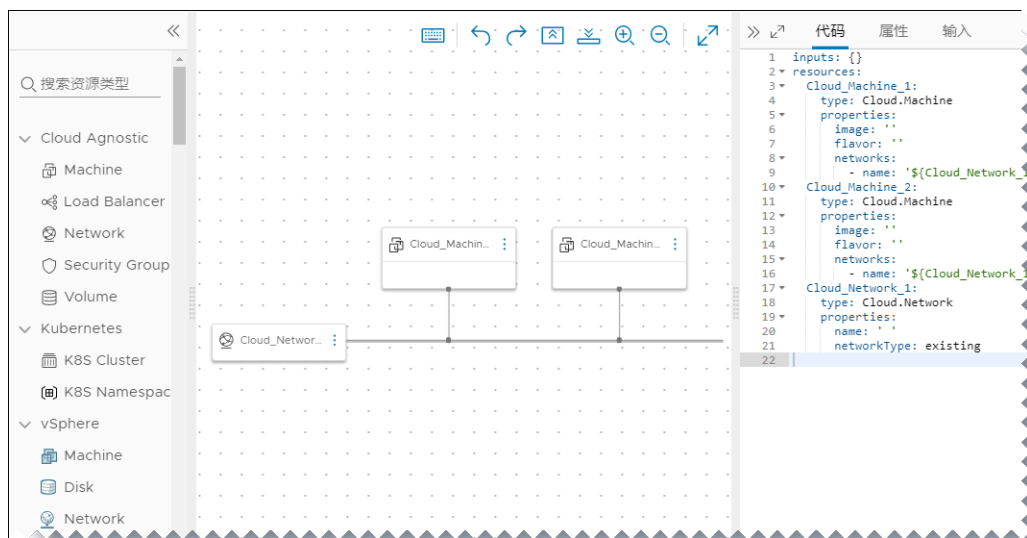
设计 vRealize Automation Cloud Assembly 部署

6

部署从蓝图开始，这些规范定义了您通过 vRealize Automation Cloud Assembly 在云资源上创建的计算机、应用程序和服务。

作为蓝图开发人员，您可以设计面向特定云供应商的蓝图，或者将蓝图设为云不可知。分配给项目的云区域决定您可以采用的方法。请向云管理员咨询，以确保您了解构成云区域的资源类型。

请注意，vRealize Automation Cloud Assembly 蓝图创建是一个基础架构即代码过程。您可以在设计画布中添加并连接组件以开始。然后使用画布右边的代码编辑器填写详细信息。在代码编辑器中，可以直接键入代码或将属性值输入表单。



本章讨论了以下主题：

- 创建蓝图之前
- 用于创建蓝图的方法
- 如何从头开始创建简单的 vRealize Automation Cloud Assembly 蓝图
- 如何增强简单的 vRealize Automation Cloud Assembly 蓝图
- 如何保存不同版本的 vRealize Automation Cloud Assembly 蓝图
- 如何使用 vRealize Automation Cloud Assembly 自定义已部署资源的名称

- 有哪些 vRealize Automation 资源属性
- 有哪些蓝图代码示例
- 如何使用 vRealize Automation Cloud Assembly 商城
- 如何使用可扩展性扩展应用程序生命周期并实现自动化

创建蓝图之前

您可以在任何时候创建 vRealize Automation Cloud Assembly 蓝图，但首先需要定义云资源基础架构才能部署蓝图。

- [第 4 章 构建您的 vRealize Automation Cloud Assembly 资源基础架构](#)

此外，您必须创建包含这些基础架构资源作为云区域的 vRealize Automation Cloud Assembly 项目。

- [使用 vRealize Automation Cloud Assembly 项目标记和自定义属性](#)

用于创建蓝图的方法

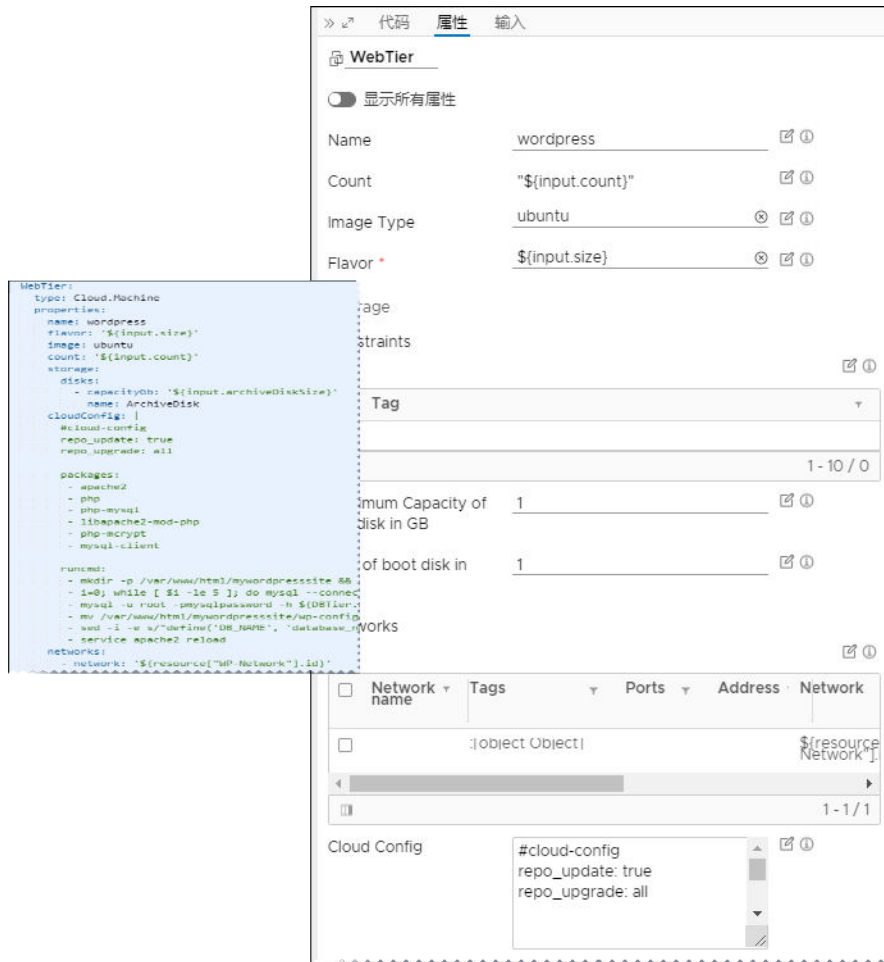
vRealize Automation Cloud Assembly 创建蓝图并将蓝图另存为代码，这使您可以轻松设计和重用蓝图。

您可以从空白画布生成蓝图，也可以利用现有代码。

vRealize Automation Cloud Assembly 蓝图设计页面

要从头开始创建蓝图，请转到[蓝图](#)并单击**新建**。将组件拖动到画布中，连接这些组件，并在代码编辑器中为组件完成配置。

代码编辑器允许您直接键入、剪切、复制和粘贴代码。如果您不喜欢编辑代码，则可以在设计画布中选择一个资源，单击代码编辑器的**属性**选项卡，并在其中输入值。您输入的属性值将显示在代码中，就像直接键入它们一样。



请注意，您可以将代码从一个蓝图复制并粘贴到另一个蓝图。

蓝图克隆

要克隆蓝图，请转到[蓝图](#)，选择一个源，然后单击**克隆**。可以克隆蓝图来基于源创建一个副本，然后将克隆分配给新项目或将其用作新应用程序的起始代码。

上载和下载

vRealize Automation Cloud Assembly 商城提供了完成的蓝图以快速启动您的工作。请参见[如何使用 vRealize Automation Cloud Assembly 商城](#)。

此外，您还可以采用对您的站点有效的任何方式上载、下载和共享蓝图 YAML 代码。您甚至可以使用外部编辑器和开发环境修改蓝图代码。

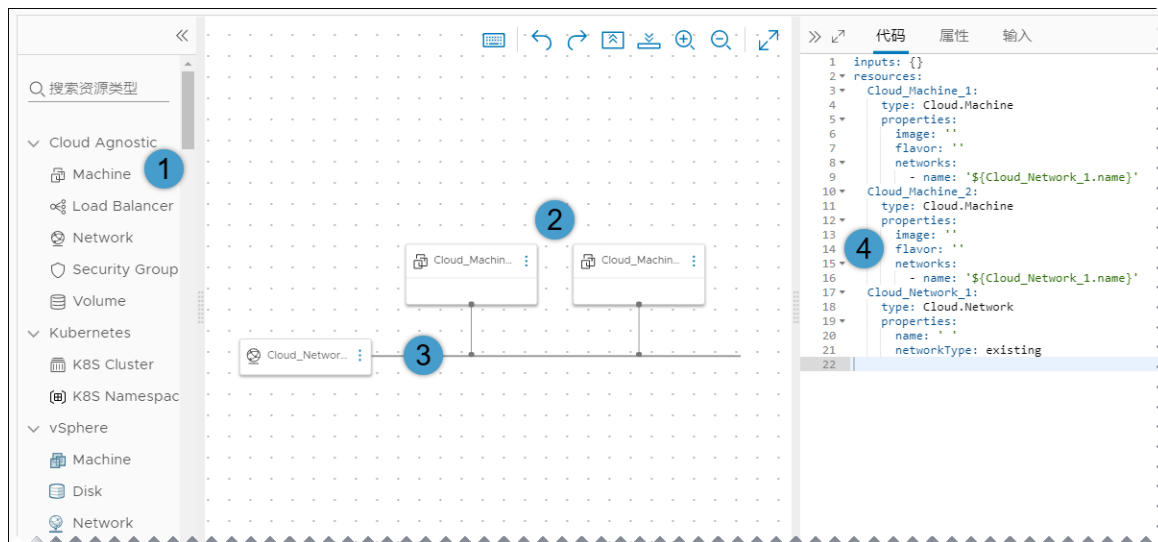
注 验证共享蓝图代码的一个典型方法是，在蓝图设计页面上的 vRealize Automation Cloud Assembly 代码编辑器中对其进行检查。

| 蓝图 250 页 | | | | | | |
|-------------------------------------|----------------------------|-----|---------------------------------|-----------------------|----------------------|-------|
| + 新建 上一步 同步存储库 克隆 部署 下载 删除 | | | | | | |
| Q 筛选... | | | | | | |
| <input type="checkbox"/> | 名称 | 源控制 | 项目 | 上次更新日期 | 更新者 | 已发布版本 |
| <input checked="" type="checkbox"/> | vSphere-With-Disk-Attached | | Human Resources Tool Project | 2020年1月21日 下午4:00:38 | sestervii@vmware.com | 0 共 0 |
| <input type="checkbox"/> | code stream | | 0709-AWS-w2测试表ホアA中匠6器停8道Ü8àùH | 2020年1月21日 下午3:52:28 | sestervii@vmware.com | 0 共 0 |
| <input type="checkbox"/> | vSphere-With-Disk-Attached | | test-AD-project | 2020年1月21日 下午12:48:57 | sestervii@vmware.com | 0 共 0 |
| <input type="checkbox"/> | DB | | 0709-AWS-w2测试表ホアA中匠6器停8道Ü8àùH | 2020年1月21日 上午11:34:36 | pmartini@vmware.com | 0 共 0 |
| <input type="checkbox"/> | WordPress-BP | | 0709-AWS-w2测试表ホアA中匠6器停8道Ü8àùH | 2020年1月20日 下午3:25:36 | canli@vmware.com | 0 共 0 |
| <input type="checkbox"/> | git BPP | | Azure Project测试表ホアA中匠6器停8道Ü8àùH | 2020年1月20日 下午3:18:40 | canli@vmware.com | 0 共 0 |

如何从头开始创建简单的 vRealize Automation Cloud Assembly 蓝图

可以使用设计页面为要置备的计算机或应用程序创建 vRealize Automation Cloud Assembly 蓝图规范。

- 1 查找组件。
- 2 将组件拖动到画布中。
- 3 连接组件。
- 4 通过编辑蓝图代码来配置组件。



从设计页面中，还可以更改蓝图名称和版本、恢复到某个蓝图版本以及克隆或部署蓝图。

如何选择 vRealize Automation Cloud Assembly 组件并将其添加到蓝图

vRealize Automation Cloud Assembly 组件是蓝图构建块。在设计页面中，您可以使用云不可知的组件或特定于云供应商的组件。

组件将显示在设计页面的左侧供您选择。

云不可知组件

可以将云不可知的组件部署到任何云供应商。在置备时，部署使用匹配的特定于云的资源。例如，如果希望蓝图同时部署到 AWS 云区域和 vSphere 云区域，请使用云不可知的组件。

云供应商组件

Amazon Web Services 组件、Microsoft Azure 组件和 VMware vSphere 组件只能部署到匹配的 AWS 云区域、Azure 云区域或 vSphere 云区域。

可以将云不可知的组件添加到包含特定于云供应商的组件的蓝图。请注意项目云区域在供应商方面提供的支持。

配置管理组件

配置管理组件依赖集成的应用程序。例如，Puppet 组件可以监控并强制执行其他组件的配置。

如何在 vRealize Automation Cloud Assembly 中连接蓝图资源

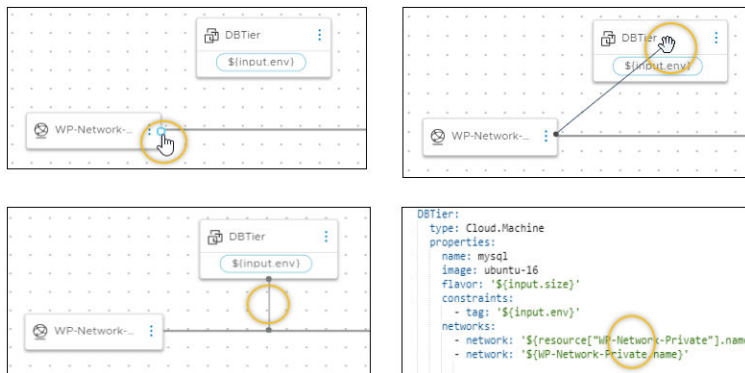
使用图形设计画布连接 vRealize Automation Cloud Assembly 蓝图资源。

如果要连接的资源兼容，则可以连接这些资源。例如：

- 将负载均衡器连接到计算机集群。
- 将计算机连接到网络。
- 将外部存储连接到计算机。

要进行连接，请将鼠标指针悬停在资源边缘上以显示连接气泡。之后，单击并拖动气泡到目标资源，然后释放鼠标按钮。

在代码编辑器中，源资源的其他代码将显示在目标资源的代码中。



资源之间的实线表示资源必须在同一位置结束。尽管可以在画布上添加连接，但如果存在有冲突的布置限制标记，则部署将失败。例如，如果连接以下资源：一个硬性限制到测试 us-west-1 云区域，而另一个硬性限制到生产 us-east-1 云区域，则部署将失败。

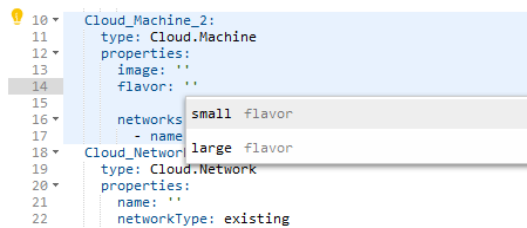
如何创建有效的 vRealize Automation Cloud Assembly 蓝图代码

在画布中添加并连接 vRealize Automation Cloud Assembly 组件时，系统只会创建起始代码。要完全配置这些组件，请编辑代码。

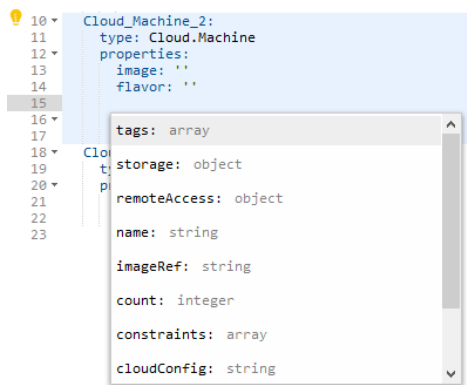
在代码编辑器中，可以直接键入代码或将属性值输入表单。为了帮助直接创建代码，vRealize Automation Cloud Assembly 编辑器包括语法完成和错误检查功能。

编辑器提示 示例

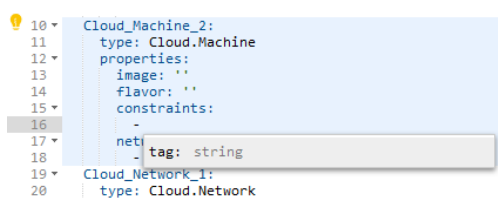
可用值



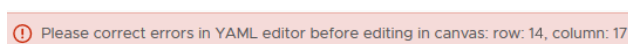
允许的属性



子属性



语法错误



编辑器提示 示例

按 Ctrl+F
可执行搜索

```

1 inputs: {}
2 resources:
3   Cloud_Machine_1:
4     type: Cloud.Machine
5     properties:
6       image: ''
7       flavor: ''
8     networks:
9       - name: '${Cloud_Network_1.name}'
10  Cloud_Machine_2:
11    type: Cloud.Machine
12    properties:
13      image: ''
14      flavor: 'small'
15      constraints:
16      networks:
17        - name: '${Cloud_Network_1.name}'

```

可选参数

插入可选参数

- + attachedDisks
- + autoScaleConfiguration
- + cloudConfig
- + cloudConfigSettings

```

1 inputs: {}
2 resources:
3   Cloud_Machine_1:
4     type: Cloud.Machine
5     properties:
6       image: ''
7       flavor: ''
8     networks:
9       - name: '${Cloud_Network_1.name}'
10  Cloud_Machine_2:
11    type: Cloud.Machine
12    properties:
13      image: ''
14      flavor: 'small'
15      constraints:
16      networks:
17        - name: '${Cloud_Network_1.name}'

```

结构定义帮助 对于所有自定义属性，还可以参阅 [VMware {code} 站点上的整合资源结构定义。](#)

cloudConfig

类型

string

When provisioning an instance, machine cloud-init startup instructions from user data fields. Sample cloud config instructions:

```

#cloud-config
repo_update: true
repo_upgrade: all
packages:
- httpd
- mariadb-server

runcmd:
- [ sh, -c, "amazon-linux-extras install -y
- systemctl start httpd
- sudo systemctl enable httpd

```

```

Tier:
type: Cloud.Machine
properties:
  name: mysql
  image: ubuntu-16
  flavor: '${input.size}'
  constraints:
    - tag: '${input.env}'
  networks:
    - name: '${resource["WP-Network-Private"]}'
    - name: '${WP-Network-Private.name}'
  remoteAccess:
    authentication: usernamePassword
    username: '${input.username}'
    password: '${input.userpassword}'
  cloudConfig:
    #cloud-config
    repo_update: true
    repo_upgrade: all

    packages:
    - mysql-server

    runcmd:
    - sed -e '/bind-address/ s/^#/#/' -i
    - service mysql restart
    - mysql -e "GRANT ALL PRIVILEGES ON *.*
    - mysql -e "FLUSH PRIVILEGES;"
  attachedDisks: []
bTier:
type: Cloud.Machine

```

如何增强简单的 vRealize Automation Cloud Assembly 蓝图

存在高级的 vRealize Automation Cloud Assembly 蓝图代码的可能性，可以使简单的蓝图到达一个新的水平。

此处描述的技术需要对基础架构代码有一定的了解。幸运的是，vRealize Automation Cloud Assembly 代码是人类可读的，并且相当容易遵循。

用户输入如何自定义 vRealize Automation Cloud Assembly 蓝图

作为蓝图开发人员，您可以使用输入参数，以便用户可以在请求时进行自定义选择。

用户提供输入时，您不再需要保存多个仅有细微差别的蓝图副本。此外，输入还可以为实施后操作准备蓝图，因为目录用户在更新部署时会看到提供新输入的提示。

小心 某些属性更改会导致重新创建资源。例如，更改 `Cloud.Service.Azure.App.Service` 下的 `connection_string.name` 会删除现有资源并创建一个新资源。

设计输入以支持实施后更改时，请决定是否允许输入删除和重新创建资源。要了解哪些属性会重新创建资源，请参见 [VMware {code} 上的 vRealize Automation 蓝图结构](#) 中的整合资源结构。

以下输入介绍了如何为 MySQL 数据库服务器创建一个蓝图，用户可以在其中将该蓝图部署到不同的云资源环境中，并且每次都应用不同的容量和凭据。

| | | |
|----------------------|----------|---------|
| Environment | env.dev | ▼ ⓘ |
| Database Tier Size * | small | ▼ ⓘ |
| Database Username * | ouradmin | |
| Database Password * | ••••• | |
| MySQL Data Disk Size | 4 | ⬆️ ⬆️ ⓘ |

如何定义蓝图输入参数

在蓝图代码中添加一个 `inputs` 部分，您可以在其中设置可选择的值。

在以下示例中，可以选择计算机大小、操作系统和集群服务器数量。

```
inputs:
  wp-size:
    type: string
    enum:
      - small
      - medium
    description: Size of Nodes
```

```

    title: Node Size
  wp-image:
    type: string
    enum:
      - coreos
      - ubuntu
    title: Select Image/OS
  wp-count:
    type: integer
    default: 2
    maximum: 5
    minimum: 2
    title: Wordpress Cluster Size
    description: Wordpress Cluster Size (Number of nodes)

```

如果您不擅长编辑代码，可以单击代码编辑器的**输入**选项卡，然后在其中输入设置。以下示例显示了前面提到的 MySQL 数据库的一些输入。

The screenshot shows the 'Inputs' tab in the vRealize Automation Cloud Assembly interface. It displays a table of blueprint inputs and an 'Edit Blueprint Input' dialog for the 'size' input.

| Name | Title | Type | Default Value |
|------------------|----------------------|--------|---------------|
| size | Tier Machine Size | string | |
| username | Database Username | string | |
| userpassword | Database Password | string | **** |
| databaseDiskSize | MySQL Data Disk Size | number | 4 |

Edit Blueprint Input: size

Name *

Title

Description

Type

Encrypted ☐

如何引用蓝图输入参数

然后，在 `resources` 部分中，使用 `${input.property-name}` 语法引用输入参数。

如果属性名称包含空格，请使用方括号和双引号分隔，而不是使用点表示形式：`${input["property name"]}`

重要说明 在蓝图代码中，除了指示输入参数外，不能使用 `input` 一词。

```
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      flavor: '${input.wp-size}'
      image: '${input.wp-image}'
      count: '${input.wp-count}'
```

输入属性的列表

| 属性 | 说明 |
|-------------|-------------------------------------------------------------------------------------------|
| const | 与 <code>oneOf</code> 一起使用。与友好标题关联的实际值。 |
| default | 输入的预填充值。 默认值必须为正确的类型。请勿输入单词作为整数的默认值。 |
| description | 输入的用户帮助文本。 |
| encrypted | 是否对用户键入的输入进行加密（ <code>true</code> 或 <code>false</code> ）。 密码通常是加密的。 |
| enum | 允许值的下拉菜单。 请使用以下示例作为格式指南。 <div> <pre>enum: - value 1 - value 2</pre> </div> |
| format | 设置所需的输入格式。例如，(25/04/19) 支持日期-时间。 允许在 vRealize Automation Service Broker 自定义表单中使用日期选择器。 |
| items | 声明数组中的项目。支持数字、整数、字符串、布尔或对象。 |
| maxItems | 数组中可选择的最大项目数。 |
| maxLength | 字符串允许使用的最大字符数。 例如，要将字段限制在 25 个字符以内，请输入 <code>maxLength: 25</code> 。 |
| maximum | 数字或整数的最大允许值。 |
| minItems | 数组中可选择的最小项目数。 |
| minLength | 字符串允许使用的最小字符数。 |
| minimum | 数字或整数的最小允许值。 |

| 属性 | 说明 |
|------------|--------------------------------------------------------------------------------------------------------------------------|
| oneOf | 允许用户输入表单为不太友好的值 (const) 显示友好名称 (title)。如果设置默认值，请设置 const ，而不是 title 。 适用于字符串、整数和数字类型。 |
| pattern | 正则表达式语法中字符串输入的允许字符。 例如，'[a-z]+' 或 '[a-zA-Z0-9@#\$%]+' |
| properties | 声明对象的 key:value 属性块。 |
| readOnly | 仅用于提供表单标签。 |
| title | 与 oneOf 一起使用。 const 值的友好名称。在部署时，标题显示在用户输入表单上。 |
| type | 数字、整数、字符串、布尔或对象的数据类型。 |
| writeOnly | 在表单中隐藏星号后面的击键。不能与 enum 一起使用。在 vRealize Automation Service Broker 自定义表单中显示为密码字段。 |

其他示例

包含枚举的字符串

```
image:
  type: string
  title: Operating System
  description: The operating system version to use.
  enum:
    - ubuntu 16.04
    - ubuntu 18.04
  default: ubuntu 16.04

shell:
  type: string
  title: Default shell
  Description: The default shell that will be configured for the created user.
  enum:
    - /bin/bash
    - /bin/sh
```

包含最小值和最大值的整数

```
count:
  type: integer
  title: Machine Count
  description: The number of machines that you want to deploy.
  maximum: 5
  minimum: 1
  default: 1
```

对象数组

```
tags:
  type: array
  title: Tags
  description: Tags that you want applied to the machines.
  items:
    type: object
    properties:
      key:
        type: string
        title: Key
      value:
        type: string
        title: Value
```

包含友好名称的字符串

```
platform:
  type: string
  oneOf:
    - title: AWS
      const: platform:aws
    - title: Azure
      const: platform:azure
    - title: vSphere
      const: platform:vsphere
  default: platform:aws
```

包含模式验证的字符串

```
username:
  type: string
  title: Username
  description: The name for the user that will be created when the machine is provisioned.
  pattern: ^[a-zA-Z]+$
```

显示为密码的字符串

```
password:
  type: string
  title: Password
  description: The initial password that will be required to logon to the machine.
  Configured to reset on first login.
  writeOnly: true
```

显示为文本区域的字符串

```
ssh_public_key:
  type: string
  title: SSH public key
  maxLength: 256
```

布尔

```
public_ip:
  type: boolean
  title: Assign public IP address
  description: Choose whether your machine should be internet facing.
  default: false
```

如何在 vRealize Automation Cloud Assembly 中设置组件部署顺序

部署 vRealize Automation Cloud Assembly 蓝图时，一个组件可能要求先提供另一个组件。

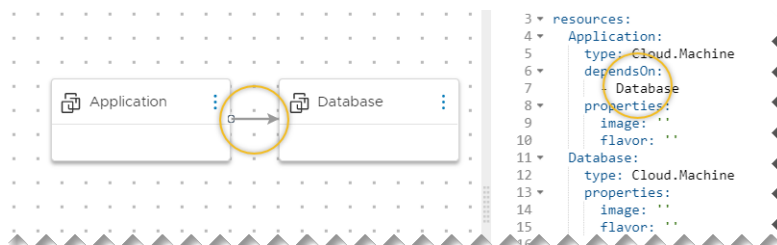
如何创建显式依赖关系

有时，一个组件要求先部署另一个组件。例如，可能需要先部署数据库服务器，然后才能创建应用程序服务器并将其配置为访问数据库服务器。

显式依赖关系设置部署时的生成顺序，或者设置是执行缩减操作还是扩大操作。您可以使用图形设计画布或代码编辑器添加显式依赖关系。

- 设计画布选项—从依赖组件开始绘制连接，然后在要先部署的组件处结束。
- 代码编辑器选项—将 `dependsOn` 属性添加到依赖组件，并标识要先部署的组件。

显式依赖关系会在画布中创建一个实心箭头。



如何创建隐式依赖关系或属性绑定

有时，一个组件的属性要求在另一个组件的属性中找到值。例如，备份服务器可能需要正在备份的数据库服务器的操作系统映像，因此数据库服务器必须先存在。

隐式依赖关系也称为属性绑定，它通过等待所需属性可用再部署依赖组件来控制生成顺序。您可以使用代码编辑器添加隐式依赖关系。

- 编辑依赖组件，添加一个标识必须先存在的组件和属性的属性。

隐式依赖关系或属性绑定会在画布中创建一个虚线箭头。



如何使用表达式使 vRealize Automation Cloud Assembly 蓝图代码更具通用性

要提高灵活性，您可以向 vRealize Automation Cloud Assembly 蓝图代码添加表达式。

表达式使用 `${expression}` 构造，如以下示例中所示。

这些示例已经过修剪，仅显示重要行。未经编辑的整个蓝图显示在最后。

示例

在部署时，允许用户粘贴在远程访问所需的加密密钥中：

```

inputs:
  sshKey:
    type: string
    maxLength: 500
resources:
  frontend:
    type: Cloud.Machine
    properties:
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'

```

要部署到 VMware Cloud on AWS，请将文件夹名称设置为所需的 Workload 名称：

```

inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'

```


部署时，使用与所选环境匹配的全小写 `env` 标记对计算机进行标记：

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
resources:
  frontend:
    type: Cloud.Machine
    properties:
      constraints:
        - tag: '${"env:" + to_lower(input.environment)}'
```

将前端群集中的计算机数设置为 1（小型）或 2（大型）。请注意，大型群集是通过清除过程设置的：

```
inputs:
  envsize:
    type: string
    enum:
      - Small
      - Large
resources:
  frontend:
    type: Cloud.Machine
    properties:
      count: '${input.envsize == "Small" ? 1 : 2}'
```

通过绑定到在网络资源中找到的属性，将计算机连接到同一 Default 网络：

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing
```

对提交到 API 的访问凭据进行加密:

```
resources:
  apitier:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        #cloud-config
        runcmd:
          - export apikey=${base64_encode(input.username:input.password)}
          - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com
```

发现 API 计算机的地址:

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        runcmd:
          - echo ${resource.apitier.networks[0].address}
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
```

完整蓝图

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
  sshKey:
    type: string
    maxLength: 500
  envsize:
    type: string
    enum:
      - Small
      - Large
  resources:
    frontend:
      type: Cloud.Machine
      properties:
        folderName: '${input.environment == "VMC" ? "Workload" : ""}'
        image: ubuntu
```

```

    flavor: medium
    count: '${input.envsize == "Small" ? 1 : 2}'
    remoteAccess:
      authentication: publicPrivateKey
      sshKey: '${input.sshKey}'
    cloudConfig: |
      packages:
        - nginx
      runcmd:
        - echo ${resource.apitier.networks[0].address}
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'
    networks:
      - name: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu
      flavor: small
      cloudConfig: |
        #cloud-config
      runcmd:
        - export apikey=${base64_encode(input.username:input.password)}
        - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com
    remoteAccess:
      authentication: publicPrivateKey
      sshKey: '${input.sshKey}'
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'
    networks:
      - name: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'

```

vRealize Automation Cloud Assembly 蓝图表达式语法

vRealize Automation Cloud Assembly 蓝图表达式语法展现了表达式的所有功能。

如何使用表达式使 vRealize Automation Cloud Assembly 蓝图代码更具通用性介绍的示例中仅对语法进行了部分描述。

文字

支持以下文字。

- 布尔 (true 或 false)

- 整数
- 浮点
- 字符串

反斜杠转义双引号、单引号和反斜杠本身：

" 转义为 \"

' 转义为 \'

\ 转义为 \\

只需要在以相同类型的引号括起来的字符串中转义引号，如以下示例所示。

```
"I am a \"double quoted\" string inside \"double quotes\"."
```

- Null

环境变量

环境名称：

- orgId
- projectId
- projectName
- deploymentId
- deploymentName
- blueprintId
- blueprintVersion
- blueprintName
- requestedBy（用户）
- requestedAt（时间）

语法：

```
env.ENV_NAME
```

示例：

```
${env.blueprintId}
```

资源变量

使用资源变量可以绑定到来自其他资源的资源属性。

语法：

```
resource.RESOURCE_NAME.PROPERTY_NAME
```

示例：

- `${resource.db.id}`
- `${resource.db.networks[0].address}`
- `${resource.app.id}`（返回非集群资源的字符串 - 未指定计数。返回集群资源的数组。）
- `${resource.app[0].id}`（返回集群资源的第一个条目。）

资源自身变量

仅允许对支持分配阶段的资源使用资源自身变量。资源自身变量只有在分配阶段完成后才可用（或具有值集）。

语法：

`self.property_name`

示例：

`${self.address}`（返回在分配阶段分配的地址。）

请注意，对于名为 `resource_x` 的资源，`self.property_name` 和 `resource.resource_x.property_name` 相同，并且都视为自引用。

集群计数索引

语法：

`count.index`

示例：

`${count.index == 0 ? "primary" : "secondary"}`（返回集群资源的节点类型。）

限制：

不支持使用 `count.index` 进行资源分配。例如，以下容量表达式在输入时创建的磁盘数组中引用位置时会失败。

```
inputs:
  disks:
    type: array
    minItems: 0
    maxItems: 12
    items:
      type: object
      properties:
        size:
          type: integer
          title: Size (GB)
          minSize: 1
          maxSize: 2048
resources:
  Cloud_vSphere_Disk_1:
```

```

type: Cloud.vSphere.Disk
properties:
  capacityGb: '${input.disks[count.index].size}'
  count: '${length(input.disks)}'

```

条件

语法:

- 相等运算符为 == 和 !=。
- 关系运算符为 < > <= 和 >=。
- 逻辑运算符为 && || 和 !。
- 条件使用模式:

condition-expression ? true-expression : false-expression

示例:

```
${input.count < 5 && input.size == 'small'}
```

```
${input.count < 2 ? "small" : "large"}
```

算术运算符

语法:

运算符为 + - / * 和 %。

示例:

```
${(input.count + 5) * 2}
```

字符串串联

语法:

`${'ABC' + 'DEF'}` 的评估结果为 ABCDEF。

运算符 [] 和 .

在统一处理 [] 和 . 运算符方面, 表达式遵循 ECMAScript。

因此, `expr.identifier` 等效于 `expr["identifier"]`。identifier 用于构造其值为 identifier 的文字, 然后将 [] 运算符与该值一起使用。

示例:

```
${resource.app.networks[0].address}
```

此外, 当属性包含空格时, 请使用方括号和双引号分隔, 而不是使用点表示形式。

不正确:

```
input.operating system
```

正确:

```
input["operating system"]
```

映射构造

语法:

```
${{'key1':'value1', 'key2':input.key2}}
```

数组构造

语法:

```
${['key1','key2']}
```

示例:

```
${[1,2,3]}
```

函数

语法:

```
${function(arguments...)}
```

示例:

```
${to_lower(resource.app.name)}
```

表 6-1. 函数

| 函数 | 说明 |
|------------------------------|-------------------------------|
| abs(number) | 数字绝对值 |
| floor(number) | 返回小于或等于参数且等于数学整数的最大（最接近正无穷大）值 |
| ceil(number) | 返回大于或等于参数且等于数学整数的最小（最接近负无穷大）值 |
| to_lower(str) | 将字符串转换为小写形式 |
| to_upper(str) | 将字符串转换为大写形式 |
| contains(array, value) | 检查数组是否包含值 |
| contains(string, value) | 检查字符串是否包含值 |
| join(array, delim) | 使用分隔符连接字符串数组并返回一个字符串 |
| split(string, delim) | 使用分隔符拆分字符串并返回字符串数组 |
| slice(array, begin, end) | 返回从开始索引到结束索引的数组片 |
| reverse(array) | 反向数组条目 |
| starts_with(subject, prefix) | 检查主题字符串是否以前缀字符串开头 |
| ends_with(subject, suffix) | 检查主题字符串是否以后缀字符串结尾 |

表 6-1. 函数（续）

| 函数 | 说明 |
|--------------------------------------|---------------------------------------------------|
| replace(string, target, replacement) | 将包含目标字符串的字符串替换为目标字符串 |
| substring(string, begin, end) | 返回从开始索引到结束索引的字符串的子字符串 |
| format(format, values...) | 返回使用 Java Formatter 类格式的带格式字符串和值。 |
| keys(map) | 返回映射的键 |
| values(map) | 返回映射的值 |
| merge(map, map) | 返回合并映射 |
| length(string) | 返回字符串长度 |
| length(array) | 返回数组长度 |
| max(array) | 返回数字数组中的最大值 |
| min(array) | 返回数字数组中的最小值 |
| sum(array) | 返回数字数组中所有值的总和 |
| avg(array) | 返回数字数组中所有值的平均值 |
| digest(value, type) | 返回使用受支持类型（md5、sha1、sha256、sha384、sha512）的值的摘要 |
| to_string(value) | 返回值的字符串表示形式 |
| to_number(string) | 将字符串解析为数字 |
| not_null(array) | 返回第一个非 null 条目 |
| base64_encode(string) | 返回 base64 编码值 |
| base64_decode(string) | 返回解码的 base64 值 |
| now() | 以 ISO-8601 格式返回当前时间 |
| uuid() | 返回随机生成的 UUID |
| from_json(string) | 解析 json 字符串 |
| to_json(value) | 将值序列化为 json 字符串 |
| json_path(value, path) | 使用 XPath for JSON 根据值评估路径。 |
| matches(string, regex) | 检查字符串是否与正则表达式匹配 |
| url_encode(string) | 使用 url 编码规范对字符串进行编码 |
| trim(string) | 移除前导空格和尾随空格 |

如何在 vRealize Automation Cloud Assembly 蓝图中自动初始化计算机

您可以通过 vSphere 自定义规范或直接运行命令在 vRealize Automation Cloud Assembly 中应用计算机初始化。

蓝图代码中的属性可以按名称引用 vSphere 自定义规范。或者，您也可以向蓝图中添加 cloudConfig 部分，并在此部分中嵌入特定命令。

尝试组合嵌入式命令和自定义规范初始化时，请务必小心。两者未正式兼容，因此一起使用时可能会产生不一致或不需要的结果。

有关自定义规范如何影响 cloudConfig 部分中的命令的示例，请参见[如何部署具有静态 IP 地址的 Linux 计算机](#)。

vRealize Automation Cloud Assembly 蓝图中的 vSphere 自定义规范

自定义规范允许您在部署时（部署到基于 vSphere 的云区域时），应用客户机操作系统设置。

自定义规范必须存在于 vSphere 中您部署到的目标上。

直接编辑蓝图代码。以下示例指向 vSphere 上 WordPress 主机的 cloud-assembly-linux 自定义规范。

```
resources:
  WebTier:
    type: Cloud.vSphere.Machine
    properties:
      name: wordpress
      cpuCount: 2
      totalMemoryMB: 1024
      imageRef: 'Template: ubuntu-18.04'
      customizationSpec: 'cloud-assembly-linux'
      resourceGroupName: '/Datacenters/Datacenter/vm/deployments'
```

自定义规范与初始化命令

如果您希望置备体验符合您当前在 vSphere 中执行的操作，那么继续使用自定义规范可能是最佳方法。但是，要扩展到混合或多云置备，一种更中立的方法是在蓝图的 cloudConfig 部分中嵌入初始化命令。

有关蓝图中的 cloudConfig 部分的详细信息，请参见[vRealize Automation Cloud Assembly 蓝图中的配置命令](#)。

vRealize Automation Cloud Assembly 蓝图中的配置命令

可以将 CloudConfig 部分添加到 vRealize Automation Cloud Assembly 蓝图代码中，在其中添加在部署时运行的计算机初始化脚本。

- Linux - 初始化命令遵循开放的 [cloud-init](#) 标准。

- Windows - 初始化命令使用 [Cloudbase-init](#)。

注 Linux [cloud-init](#) 和 Windows [Cloudbase-init](#) 不共享相同的语法。一个操作系统的 cloudConfig 部分在其他操作系统的计算机映像中不起作用。

您可以使用初始化命令在创建实例时自动应用数据或设置，以自定义用户、权限、安装或任何其他基于命令的操作。示例包括：

- 设置主机名
- 生成并设置 SSH 私钥
- 安装软件包

在 vRealize Automation Cloud Assembly 中，您还可以在配置基础架构时预先将初始化命令添加到计算机映像。引用源映像的所有蓝图都将进行相同的初始化。

重要说明 您可能有一个映像映射和一个蓝图，这两者都包含初始化命令。在部署时，这些命令将合并，vRealize Automation Cloud Assembly 会运行合并的命令。

当两个位置中出现相同的命令但包含不同的参数时，系统只会运行映像映射命令。

有关更多详细信息，请参见[了解有关 vRealize Automation Cloud Assembly 中的映像映射的更多信息](#)。

以下示例 cloudConfig 部分取自适用于基于 Linux 的 MySQL 服务器的 [WordPress 用例：创建基本蓝图](#) 蓝图代码。

为确保正确解释命令，请始终包括管道线字符 cloudConfig: |，如下所示。

如果 cloud-init 脚本行为异常，请在故障排除时检查 /var/log/cloud-init-output.log 中捕获的控制台输出。有关 cloud-init 的详细信息，请参见 [cloud-init 文档](#)。

```
cloudConfig: |
  #cloud-config
  repo_update: true
  repo_upgrade: all
  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
    - php-mcrypt
    - mysql-client
  runcmd:
    - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
    - i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
```

```
mywordpresssite/wp-config.php
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_PASSWORD',
'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define( 'DB_HOST',
'localhost' );"/"define( 'DB_HOST', '${DBTier.networks[0].address}' );"/ /var/www/html/
mywordpresssite/wp-config.php
    - service apache2 reload
```

如何部署具有静态 IP 地址的 Linux 计算机

部署到 vSphere 时，静态 IP 地址要求 vRealize Automation Cloud Assembly 生成 vSphere 自定义规范，这可能会干扰 cloud-init 命令。

问题

- vRealize Automation Cloud Assembly 蓝图包括 `assignment: static`，以将静态 IP 地址应用于 vSphere 虚拟机。
- 该蓝图还包含 `cloudConfig` 部分，其中包括使用 `cloud-init` 运行的初始化命令。
- 要为虚拟机提供静态 IP，vRealize Automation Cloud Assembly 应动态生成要应用的 vSphere 自定义规范。
- 每次应用自定义规范时，最后一个操作将重新启动虚拟机。
- 自定义规范不知道 `cloud-init` 命令正在运行，因此重新启动会中断这些命令。
- `Cloud-init` 命令仅在首次引导时运行，并且在中断时不会自动恢复。
- 生成的虚拟机仅完成部分配置。

解决办法

创建一个包含 `cloud-init` 定时禁用的计算机模板。然后，根据模板部署计算机，以便在执行 `cloud-init` 之前进行自定义规范和重新启动操作。

示例过程 — Ubuntu 18.04

以下步骤适用于 Ubuntu 18.04。您可能需要进行调整，并采用此处列出的适用于其他 Linux 版本或产品的观点。

- 1 创建虚拟机，并使用所需的版本更新和软件包使其保持最新。

请注意，其他 Linux 产品可能未预安装 `cloud-init`，但 Ubuntu 18.04 已经预安装了 `cloud-init`。

- 2 重新配置 `cloud-init`，将数据源设置为 OVF。

```
sudo dpkg-reconfigure cloud-init
```

- 3 编辑以下文件。

```
/etc/cloud/cloud.cfg
```

- a 通过添加以下行，启用传统客户机操作系统自定义 (GOSC)。

```
disable_vmware_customization: true
```

- b 请确保已启用网络配置。删除或注释掉禁用设置（如果存在）。

```
network:
  # config: disabled
```

或者，检查以下目录中的所有配置文件。

```
/etc/cloud/cloud.cfg.d/*
```

删除所有包含 `network: {config: disabled}` 设置的文件。

- 4 编辑以下文件。

```
/usr/lib/tmpfiles.d/tmp.conf
```

- 注释掉该设置，以防止临时目录被清除。

```
# D /tmp 1777 root root -
```

- 5 编辑以下文件。

```
/lib/systemd/system/open-vmtools.service
```

- 在 `[Unit]` 部分下添加以下行，将 `open-vmtools` 配置为在 `dbus.service` 之后启动。

```
After=dbus.service
```

- 6 创建新的空文件以禁用 `cloud-init`。

```
sudo touch /etc/cloud/cloud-init.disabled
```

- 7 创建 `re_init.sh` 脚本。在针对自定义规范暂停的 `cron` 作业延迟后，脚本将重新启用并初始化 `cloud-init`。

```
sudo rm -rf /etc/cloud/cloud-init.disabled
sudo cloud-init init
sleep 20
sudo cloud-init modules --mode config
sleep 20
sudo cloud-init modules --mode final
```

- 8 添加脚本的运行权限。

```
sudo chmod +x re_init.sh
```

- 9 创建一个 `cron` 作业，该文件将在启动时睡眠 90 秒后运行。键入 `crontab -e`，然后输入以下内容：

```
@reboot ( sleep 90 ; sh /script_path/delay_init.sh )
```

如果自定义规范和重新启动需要更长时间才能完成，您可以应用超过 90 秒的时间。

- 10 创建用于清理模板的 `cleaner.sh` 脚本。将 `cloudadmin` 替换为您在安装操作系统期间设置的用户。

示例脚本特定于 Ubuntu。要为其 Linux 产品创建脚本，请确保包含突出显示的必需部分。

```
#!/bin/bash

# Add usernames to add to /etc/sudoers for passwordless sudo
users=("ubuntu" "cloudadmin")

for user in "${users[@]}"
do
cat /etc/sudoers | grep ^$user
RC=$?
if [ $RC != 0 ]; then
bash -c "echo \"$user ALL=(ALL) NOPASSWD:ALL\" >> /etc/sudoers"
fi
done

#grab Ubuntu Codename
codename="$(lsb_release -c | awk {'print $2'})"

#Stop services for cleanup
service rsyslog stop

#clear audit logs
if [ -f /var/log/audit/audit.log ]; then
cat /dev/null > /var/log/audit/audit.log
fi
if [ -f /var/log/wtmp ]; then
cat /dev/null > /var/log/wtmp
fi
if [ -f /var/log/lastlog ]; then
cat /dev/null > /var/log/lastlog
fi

#cleanup persistent udev rules
if [ -f /etc/udev/rules.d/70-persistent-net.rules ]; then
rm /etc/udev/rules.d/70-persistent-net.rules
fi

#cleanup /tmp directories
rm -rf /tmp/*
rm -rf /var/tmp/*

#cleanup current ssh keys
#rm -f /etc/ssh/ssh_host_*

#cat /dev/null > /etc/hostname

#cleanup apt
apt-get clean

#Clean Machine ID

truncate -s 0 /etc/machine-id
rm /var/lib/dbus/machine-id
```

```
ln -s /etc/machine-id /var/lib/dbus/machine-id

#Clean Cloud-init
cloud-init clean --logs --seed

#cleanup shell history
history -w
history -c
```

- 11 添加模板清理脚本的运行权限。

```
sudo chmod +x cleaner.sh
```

- 12 在 Ubuntu 18.04 中，清理脚本需要 root 特权。编辑以下文件。

```
/etc/ssh/sshd_config
```

- a 请确保您可以切换到 root 用户。

```
PermitRootLogin yes
```

- b 为 root 用户设置密码。

```
sudo passwd root
```

- 13 运行清理脚本。

```
sudo ./script_path/cleaner.sh
```

- 14 （可选）为安全起见，请恢复步骤 12 以防止继续进行 root 登录。

- 15 关闭虚拟机，然后使用 vSphere 将其转换为模板。

模板更新

每次更新模板时，cron 作业都会运行。如果更新所需的时间超过延迟（例如 90 秒），则需要重新添加 /etc/cloud/cloud-init.disabled 文件，并在关闭模板前重新运行清理脚本。否则，在首次引导时不会禁用 cloud-init，自定义规范重新启动将中断 cloud-init 命令。

故障排除

如果您怀疑 vSphere 自定义规范阻止 cloud-init 完成，请暂时禁用自定义规范，并确定 cloud-init 是否可按预期完成。要暂时禁用自定义规范，请使用 `customizeGuestOS: false` 属性。

```
properties:
  image: ubuntu
  cpuCount: 1
  totalMemoryMB: 8192
  customizeGuestOS: false
```

如何使 vRealize Automation Cloud Assembly 部署等待初始化

有时，在继续进行 vRealize Automation Cloud Assembly 部署之前，需要完全启动虚拟机。

例如，如果部署仍在安装软件包和启动 Web 服务器的计算机，则可能会导致快速用户试图在应用程序可用之前访问该应用程序的情况。

使用此功能时，请注意以下注意事项。

- 此功能使用 `cloud-init` `phone_home` 模块，并在部署 Linux 计算机时可用。
- 由于 `Cloudbase-init` 限制，`phone_home` 不适用于 Windows。
- `phone_home` 可以像显式依赖关系一样影响部署顺序，但在计时和处理选项方面具有更大的灵活性。请参见[如何在 vRealize Automation Cloud Assembly 中设置组件部署顺序](#)。
- `phone_home` 需要蓝图中的 `cloudConfig` 命令。
- 您的创造力是一个因素。`CloudConfig` 部分可能包括操作之间的嵌入等待时间，可以与 `phone_home` 协同使用。
- 如果计算机模板中已包含 `phone_home` 模块设置，则基于蓝图的 `phone_home` 将无法正常工作。
- 计算机必须具有返回 vRealize Automation Cloud Assembly 的出站通信访问权限。

要在 vRealize Automation Cloud Assembly 中使用 `phone_home` 等待计算机初始化，请将 `cloudConfigSettings` 部分添加到蓝图中：

```
cloudConfigSettings:
  phoneHomeShouldWait: true
  phoneHomeTimeoutSeconds: 600
  phoneHomeFailOnTimeout: true
```

| 属性 | 说明 |
|--------------------------------------|--------------------------------------------------------------------------------|
| <code>phoneHomeShouldWait</code> | 是否等待初始化（ <code>true</code> 或 <code>false</code> ）。 |
| <code>phoneHomeTimeoutSeconds</code> | 何时决定是否继续部署，即使初始化仍在运行。默认值为 10 分钟。 |
| <code>phoneHomeFailOnTimeout</code> | 是否在超时后继续部署（ <code>true</code> 或 <code>false</code> ）。请注意，即使继续进行，部署仍可能因各种原因而失败。 |

如何在 vRealize Automation Cloud Assembly 蓝图中启用远程访问

要远程访问 vRealize Automation Cloud Assembly 已部署的计算机，请在部署之前将属性添加到该计算机的蓝图中。

对于远程访问，您可以配置以下身份验证选项之一。

注 在需要复制密钥的情况下，您还可以在蓝图中创建一个 `cloudConfig` 部分，以便在置备时自动复制密钥。此处没有详细说明，但[如何在 vRealize Automation Cloud Assembly 蓝图中自动初始化计算机](#)提供了有关 `cloudConfig` 的一般信息。

在 vRealize Automation Cloud Assembly 置备时生成密钥对

如果您没有自己的公钥-私钥对可用于远程访问身份验证的，则可以让 vRealize Automation Cloud Assembly 生成密钥对。

使用以下代码作为指导。

- 1 在 vRealize Automation Cloud Assembly 中，在置备之前，请将 `remoteAccess` 属性添加到蓝图 中，如示例中所示。

`username` 是可选的。如果省略它，系统会生成一个随机 ID 作为 `username`。

示例：

```
type: Cloud.Machine
properties:
  name: our-vm2
  image: Linux18
  flavor: small
  remoteAccess: authentication: generatedPublicPrivateKey username: testuser
```

- 2 在 vRealize Automation Cloud Assembly 中，从蓝图中置备计算机，并使其进入启动状态。
置备过程会生成密钥。

- 3 在 **部署 > 拓扑** 属性中找到密钥名称。

- 4 使用云提供商接口（如 vSphere Client）访问已置备的计算机命令行。

- 5 授予对私钥的读取权限。

```
chmod 600 key-name
```

- 6 转到 vRealize Automation Cloud Assembly 部署，选择计算机，然后单击 **操作 > 获取私钥**。

- 7 将私钥文件复制到您的本地计算机。

典型的本地文件路径是 `/home/username/.ssh/key-name`。

- 8 打开远程 SSH 会话，然后连接到已置备的计算机。

```
ssh -i key-name user-name@machine-ip
```

为 vRealize Automation Cloud Assembly 提供您自己的公钥-私钥对

许多企业会创建和分发自己的公钥-私钥对以进行身份验证。

使用以下代码作为指导。

- 1 在本地环境中，获取或生成您的公钥-私钥对。

现在，只需在本地生成并保存密钥。

- 2 在 vRealize Automation Cloud Assembly 中，在置备之前，请将 `remoteAccess` 属性添加到蓝图 中，如示例中所示。

`sshKey` 包括可在公钥文件 `key-name.pub` 中找到的长字母数字。

`username` 是可选的，创建后可用在登录时使用。如果省略它，系统会生成一个随机 ID 作为 `username`。

示例：

```
type: Cloud.Machine
properties:
  name: our-vm1
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: ssh-rsa Iq+5aQgBP3ZNT4o1baP5Ii+dstIcowRRkyobbfpA1mj9ts1f
qGxvU66PX9IeZax5hZvNWFgjw6ag+Z1zndOLhVdVoW49f274/mIRild7Uuw...
    username: testuser
```

- 3 在 vRealize Automation Cloud Assembly 中，从蓝图中置备计算机，并使其进入启动状态。
- 4 使用云供应商客户端访问置备的计算机。
- 5 将公钥文件添加到计算机上的主文件夹中。使用您在 `remoteAccess.sshKey` 中指定的密钥。
- 6 确认您的本地计算机上存在私钥文件副本。

密钥通常是不带 `.pub` 扩展名的 `/home/username/.ssh/key-name`。

- 7 打开远程 SSH 会话，然后连接到已置备的计算机。

```
ssh -i key-name user-name@machine-ip
```

为 vRealize Automation Cloud Assembly 提供 AWS 密钥对

通过将 AWS 密钥对名称添加到蓝图，您可以远程访问 vRealize Automation Cloud Assembly 部署到 AWS 的计算机。

请注意，AWS 密钥对是区域特定的。如果您将工作负载置备到 `us-east-1`，则密钥对必须存在于 `us-east-1` 中。

使用以下代码作为指导。此选项仅适用于 AWS 云区域。

```
type: Cloud.Machine
properties:
  image: Ubuntu
  flavor: small
  remoteAccess: authentication: keyPairName keyPair: cas-test
constraints:
  - tag: 'cloud:aws'
```

向 vRealize Automation Cloud Assembly 提供用户名和密码

通过将用户名和密码添加到蓝图，可以方便地远程访问 vRealize Automation Cloud Assembly 部署的计算机。

虽然安全性较低，但使用用户名和密码远程登录可能就是您的情况所需要的全部。请注意，某些云供应商或配置可能不支持这个不太安全的选项。

- 1 在 vRealize Automation Cloud Assembly 中，在置备之前，请将 `remoteAccess` 属性添加到蓝图中，如示例中所示。

设置您希望在登录时使用的用户名和密码。

示例：

```
type: Cloud.Machine
properties:
  name: our-vm3
  image: Linux18
  flavor: small
  remoteAccess: authentication: usernamePassword username: testuser password: admin123
```

- 2 在 vRealize Automation Cloud Assembly 中，从蓝图中置备计算机，并使其进入启动状态。
- 3 转到您的云供应商界面，并访问已置备的计算机。
- 4 在已置备的计算机上，创建或启用该帐户。
- 5 从本地计算机中，打开到已置备计算机 IP 地址或 FQDN 的远程会话，并像往常一样使用用户名和密码登录。

如何保存不同版本的 vRealize Automation Cloud Assembly 蓝图

作为蓝图开发人员，您可以安全地捕获工作蓝图的快照，然后再冒险进行更改。

在部署时，您可以选择任一版本进行部署。

如何捕获蓝图版本

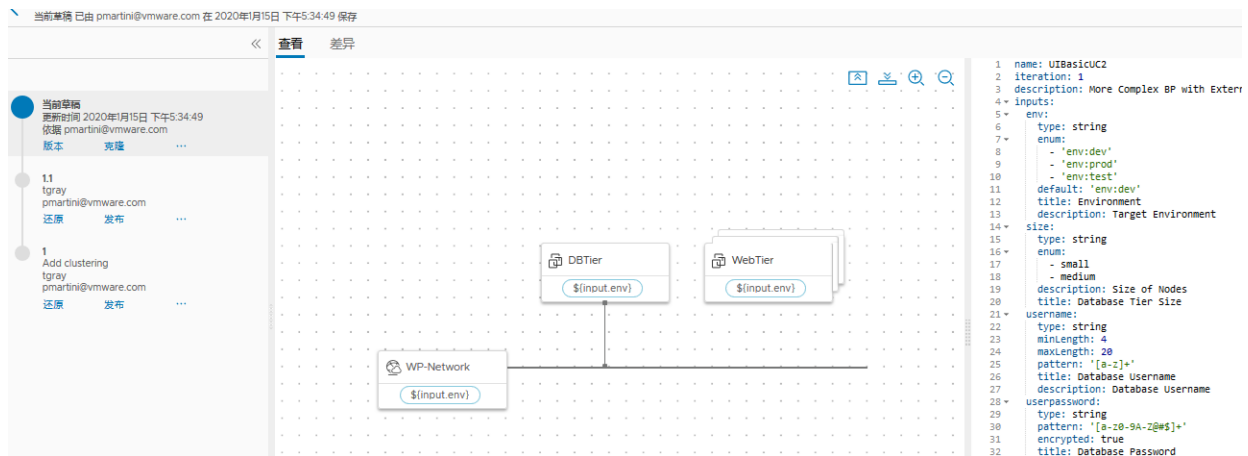
从设计页面中，单击**版本**，并提供一个名称。

该名称必须为字母数字，不含空格，并且仅允许使用句点、连字符和下划线作为特殊字符。

如何还原较旧版本

在设计页面中，单击**版本历史记录**。

在左侧选择较旧版本以在画布和代码编辑器中对其进行检查。找到所需的版本时，单击**还原**。还原将覆盖当前草稿，而不移除任何命名版本。



如何向 vRealize Automation Service Broker 用户发布版本

在设计页面中，单击**版本历史记录**。

在左侧选择一个版本并单击**发布**。对当前草稿进行版本化之前，您无法发布该草稿。

发布一个蓝图的多个版本时，vRealize Automation Service Broker 将使用最新的一个版本。

如何比较蓝图版本

更改和版本累积时，您可能需要确定它们之间的差异。

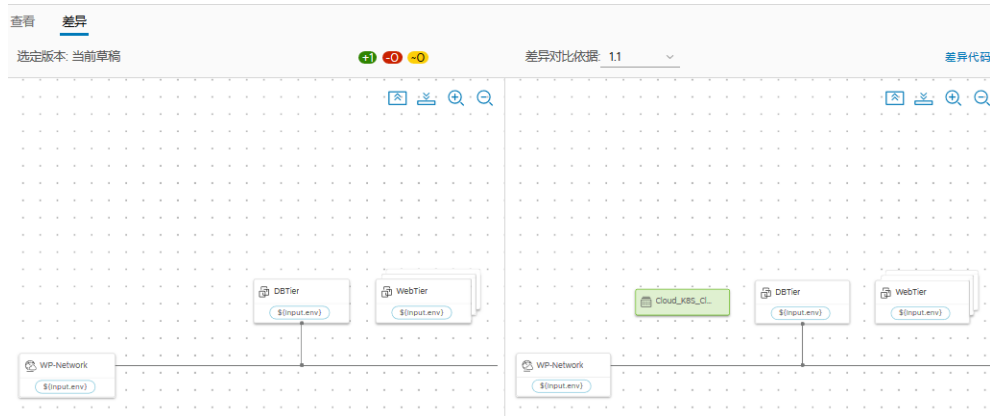
在“版本历史记录”视图中，选择一个版本，并单击**差异**。然后，从**差异对比**下拉列表中选择要比较的另一个版本。

请注意，您可以在查看代码差异或视觉拓扑差异之间切换。

图 6-1. 代码差异



图 6-2. 视觉拓扑差异



如何克隆蓝图

尽管与保存版本不同，但从设计页面中选择**操作 > 克隆**会创建当前蓝图的副本，供其他开发之用。

如何使用 vRealize Automation Cloud Assembly 自定义已部署资源的名称

作为云或项目管理员，您有一个针对环境中资源的规定命名约定，并且您希望部署的资源遵循这些约定，而无需用户交互。您可以为 vRealize Automation Cloud Assembly 项目中的所有部署创建一个命名模板。

例如，您的主机命名约定是为资源添加 *projectname-sitecode-costcenter-whereDeployed-identifier* 前缀。您可以为每个项目的计算机配置自定义命名模板。一些模板变量在部署时从系统中提取，其他一些模板变量基于项目自定义属性。

所有资源名称必须是唯一的。使用数字递增属性可确保唯一性。对于所有部署（包括由 vRealize Automation Cloud Assembly 命名的部署），数字都采用递增的形式。随着系统变得更强健，编号可能看起来是随机的，但仍确保唯一性。

除了此处提供的示例外，还可以添加用户名、使用的映像、其他内置选项和简单字符串。在生成模板时，会提供有关可能选项的提示。

请注意，用例中的某些值仅用作示例。不能在您的环境中逐字使用这些值。请考虑在何处需要替换为您自己的值或从示例值外插值，以便满足您自己的云计算基础架构需求和部署管理需求。

前提条件

- 确认您知道要用于项目中部署的命名约定。
- 此过程假定您具有或可以创建一个简单蓝图，用于测试自定义主机前缀命名。

步骤

- 1 选择**基础架构 > 项目**。
- 2 选择现有项目或新建一个项目。

- 3 在**置备**选项卡上，找到“自定义属性”部分，并为站点代码和成本中心值创建属性。

在这里，您将在此处看到的值替换为与您的环境相关的值。

自定义属性
指定应添加到此项目中的所有请求的自定义属性。 ①

| 定义自定义属性 | 名称 | 值 |
|---------|------------|-------------|
| | siteCode | BGL |
| | costCenter | IT-research |

自定义命名
指定要用于在此项目中置备的计算机、网络、安全组和磁盘的命名模板。

模板: `${project.name}-${resource.siteCode}-${resource.costCenter}` ①
Hint: Avoid conflicting names by generating digits in names: `${#####}`

- a 创建名为 **siteCode** 且值为 **BGL** 的自定义属性。
 - b 添加另一个名为 **costCenter** 且值为 **IT-research** 的自定义属性。
- 4 找到“自定义命名”部分，然后添加以下模板。

```
${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}
```

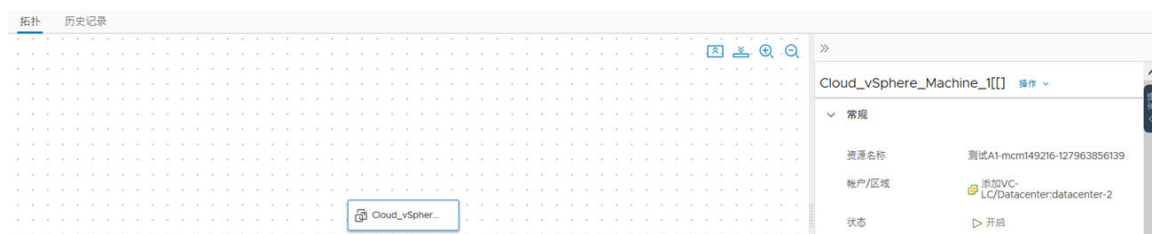
您可以在字符串中复制内容，但如果这是第一个命名模板，请考虑在生成模板时使用提示文本和快速选择。

- 5 部署与项目关联的蓝图，以验证自定义名称是否应用于资源。

- a 单击**蓝图**选项卡，然后单击与项目关联的蓝图。
- b 部署该蓝图。

此时将打开**部署**选项卡，其中显示正在进行的部署。

- c 部署完成后，单击部署名称。
- d 在**拓扑**选项卡上，请注意您的自定义名称是右侧窗格中的资源名称。



- 6 如果部署了测试蓝图以验证命名约定，则可以删除该部署。

后续步骤

为其他项目创建自定义命名模板。

有哪些 vRealize Automation 资源属性

在 vRealize Automation 基础架构即代码编辑器中，可以通过单击或悬停鼠标指标，查看语法和代码完成帮助。要查看完整的蓝图资源属性集（有时称为自定义属性），请参阅整合的资源结构定义。

该结构定义可从 VMware {code} 站点获得。访问链接，然后单击**模型**，即会列出可用于蓝图的资源对象。

- [VMware {code} 上的 vRealize Automation 资源类型结构定义](#)

有哪些蓝图代码示例

vRealize Automation Cloud Assembly 中的蓝图代码在组合与应用方面几乎无限制。

通常，成功代码的示例是进行进一步开发的最佳起点。遵照示例时，请进行替换，以便在资源名称、值等方面应用您的站点设置。

vRealize Automation Cloud Assembly 蓝图中的 vSphere 计算机示例

以下基本示例定义了 vRealize Automation Cloud Assembly 蓝图中的 vSphere 资源。

| 资源 | 示例蓝图 |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 具有 CPU、内存和操作系统的 vSphere 虚拟机 | <pre>resources: demo-machine: type: Cloud.vSphere.Machine properties: name: demo-machine cpuCount: 1 totalMemoryMB: 1024 image: ubuntu</pre> |
| 具有数据存储资源的 vSphere 计算机 | <pre>resources: demo-vsphere-disk-001: type: Cloud.vSphere.Disk properties: name: DISK_001 type: 'HDD' capacityGb: 10 dataStore: 'datastore-01' provisioningType: thick</pre> |

| 资源 | 示例蓝图 |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 具有连接磁盘的 vSphere 计算机 | <pre> resources: demo-vsphere-disk-001: type: Cloud.vSphere.Disk properties: name: DISK_001 type: HDD capacityGb: 10 dataStore: 'datastore-01' provisioningType: thin demo-machine: type: Cloud.vSphere.Machine properties: name: demo-machine cpuCount: 2 totalMemoryMB: 2048 imageRef: >- https://bintray.com/vmware/photon/ download_file?file_path=2.0%2FRC%2Fova%2Fphoton- custom-hw11-2.0-31bb961.ova attachedDisks: - source: '\${demo-vsphere-disk-001.id}' </pre> |
| 链接克隆映像中的 vSphere 计算机（附加正斜杠和快照名称） | <pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: imageRef: 'demo-machine/snapshot-01' cpuCount: 1 totalMemoryMB: 1024 </pre> |
| 在 vCenter 中位于特定文件夹的 vSphere 计算机 | <pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: name: demo-machine cpuCount: 2 totalMemoryMB: 1024 imageRef: ubuntu resourceGroupName: 'myFolder' </pre> |

| 资源 | 示例蓝图 |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 具有多个网卡的 vSphere 计算机 | <pre> resources: demo-machine: type: Cloud.Machine properties: image: ubuntu flavor: small networks: - name: '\${network-01.name}' deviceIndex: 0 - name: '\${network-02.name}' deviceIndex: 1 network-01: type: Cloud.vSphere.Network properties: name: network-01 network-02: type: Cloud.vSphere.Network properties: name: network-02 </pre> |
| 从快照映像生成的 vSphere 计算机 | <pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: imageRef: 'demo-machine/snapshot-01' cpuCount: 1 totalMemoryMB: 1024 </pre> |
| 在 vCenter 中附加了标记的 vSphere 计算机 | <pre> resources: demo-machine: type: Cloud.Machine properties: flavor: small image: ubuntu tags: - key: env value: demo </pre> |
| 具有自定义规范的 vSphere 计算机 | <pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: name: demo-machine image: ubuntu flavor: small customizationSpec: Linux </pre> |

| 资源 | 示例蓝图 |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 具有 vSphere 网络组件和静态 IP 地址的 vSphere 计算机 | <pre>resources: demo-network: type: Cloud.vSphere.Network properties: name: demo-network demo-machine: type: Cloud.vSphere.Machine properties: image: ubuntu flavor: small networks: - name: demo-network assignment: static</pre> |
| 启用远程访问的 vSphere 计算机 | <pre>inputs: username: type: string title: Username description: Username default: testUser password: type: string title: Password default: VMware@123 encrypted: true description: Password for the given username resources: demo-machine: type: Cloud.vSphere.Machine properties: flavor: small imageRef: >- https://cloud-images.ubuntu.com/ releases/16.04/release-20170307/ubuntu-16.04- server-cloudimg-amd64.ova cloudConfig: ssh_pwauth: yes chpasswd: list: \${input.username}:\${input.password} expire: false users: - default - name: \${input.username} lock_passwd: false sudo: ['ALL=(ALL) NOPASSWD:ALL'] groups: [wheel, sudo, admin] shell: '/bin/bash' runcmd: - echo "Defaults:\${input.username} ! requiretty" >> /etc/sudoers.d/\${input.username}</pre> |

已记录 vRealize Automation Cloud Assembly 蓝图示例

此示例包含一组完整的评论，让您能够查看 vRealize Automation Cloud Assembly 蓝图中各部分的结构和用途。

```
# *****
#
# This WordPress blueprint is enhanced with comments to explain its
# parameters.
#
# Try cloning it and experimenting with its YAML code. If you're new to
# YAML, visit yaml.org for general information.
#
# The blueprint deploys a minimum of 3 virtual machines and runs scripts
# to install packages.
#
# *****
#
# -----
# Blueprints need a descriptive name and version if
# source controlled in git.
# -----
name: WordPress Blueprint with Comments
formatVersion: 1
version: 1
#
# -----
# Inputs create user selections that appear at deployment time. Inputs
# can set placement decisions and configurations, and are referenced
# later, by the resources section.
# -----
inputs:
#
# -----
# Choose a cloud endpoint. 'Title' is the visible
# option text (oneOf allows for the friendly title). 'Const' is the
# tag that identifies the endpoint, which was set up earlier, under the
# Cloud Assembly Infrastructure tab.
# -----
platform:
  type: string
  title: Deploy to
  oneOf:
    - title: AWS
      const: aws
    - title: Azure
      const: azure
    - title: vSphere
      const: vsphere
  default: vsphere
#
# -----
# Choose the operating system. Note that the Cloud Assembly
# Infrastructure must also have an AWS, Azure, and vSphere Ubuntu image
```

```

# mapped. In this case, enum sets the option that you see, meaning there's
# no friendly title feature this time. Also, only Ubuntu is available
# here, but having this input stubbed in lets you add more operating
# systems later.
# -----
osimage:
  type: string
  title: Operating System
  description: Which OS to use
  enum:
    - Ubuntu
#
# -----
# Set the number of machines in the database cluster. Small and large
# correspond to 1 or 2 machines, respectively, which you see later,
# down in the resources section.
# -----
dbenvsize:
  type: string
  title: Database cluster size
  enum:
    - Small
    - Large
#
# -----
# Dynamically tag the machines that will be created. The
# 'array' of objects means you can create as many key-value pairs as
# needed. To see how array input looks when it's collected,
# open the blueprint and click TEST.
# -----
Mtags:
  type: array
  title: Tags
  description: Tags to apply to machines
  items:
    type: object
    properties:
      key:
        type: string
        title: Key
      value:
        type: string
        title: Value
#
# -----
# Create machine credentials. These credentials are needed in
# remote access configuration later, in the resources section.
# -----
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username

```

```

userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#\$]+'
  encrypted: true
  title: Database Password
  description: Database Password
#
# -----
# Set the database storage disk size.
# -----
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Size of database disk
#
# -----
# Set the number of machines in the web cluster. Small, medium, and large
# correspond to 2, 3, and 4 machines, respectively, which you see later,
# in the WebTier part of the resources section.
# -----
clusterSize:
  type: string
  enum:
    - small
    - medium
    - large
  title: Wordpress Cluster Size
  description: Wordpress Cluster Size
#
# -----
# Set the archive storage disk size.
# -----
archiveDiskSize:
  type: number
  default: 4
  maximum: 10
  title: Wordpress Archive Disk Size
  description: Size of Wordpress archive disk
#
# -----
# The resources section configures the deployment of machines, disks,
# networks, and other objects. In several places, the code pulls from
# the preceding interactive user inputs.
# -----
resources:
#
# -----
# Create the database server. Choose a cloud agnostic machine 'type' so
# that it can deploy to AWS, Azure, or vSphere. Then enter its property
# settings.
# -----
DBTier:
  type: Cloud.Machine

```

```

    properties:
#
# -----
# Descriptive name for the virtual machine. Does not become the hostname
# upon deployment.
# -----
        name: mysql
#
# -----
# Hard-coded operating system image to use. To pull from user input above,
# enter the following instead.
# image: '${input.osimage}'
# -----
        image: Ubuntu
#
# -----
# Hard-coded capacity to use. Note that the Cloud Assembly
# Infrastructure must also have AWS, Azure, and vSphere flavors
# such as small, medium, and large mapped.
# -----
        flavor: small
#
# -----
# Tag the database machine to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with a site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
        constraints:
            - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Also tag the database machine with any free-form tags that were created
# during user input.
# -----
        tags: '${input.Mtags}'
#
# -----
# Set the database cluster size by referencing the dbenvsize user
# input. Small is one machine, and large defaults to two.
# -----
        count: '${input.dbenvsize == "Small" ? 1 : 2}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
        networks:
            - name: '${resource.WP_Network.name}'
              network: '${resource.WP_Network.id}'
#
# -----
# Enable remote access to the database server. Reference the credentials
# from the user input.

```

```

# -----
  remoteAccess:
    authentication: usernamePassword
    username: '${input.username}'
    password: '${input.userpassword}'
#
# -----
# You are free to add custom properties, which might be used to initiate
# an extensibility subscription, for example.
# -----
  ABC-Company-ID: 9393
#
# -----
# Run OS commands or scripts to further configure the database machine,
# via operations such as setting a hostname, generating SSH private keys,
# or installing packages.
# -----
  cloudConfig: |
    #cloud-config
    repo_update: true
    repo_upgrade: all
    packages:
      - mysql-server
    runcmd:
      - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
      - service mysql restart
      - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
      - mysql -e "FLUSH PRIVILEGES;"
  attachedDisks: []
#
# -----
# Create the web server. Choose a cloud agnostic machine 'type' so that it
# can deploy to AWS, Azure, or vSphere. Then enter its property settings.
# -----
  WebTier:
    type: Cloud.Machine
    properties:
#
# -----
# Descriptive name for the virtual machine. Does not become the hostname
# upon deployment.
# -----
    name: wordpress
#
# -----
# Hard-coded operating system image to use. To pull from user input above,
# enter the following instead:
# image: '${input.osimage}'
# -----
    image: Ubuntu
#
# -----
# Hard-coded capacity to use. Note that the Cloud Assembly
# Infrastructure must also have AWS, Azure, and vSphere flavors
# such as small, medium, and large mapped.

```

```

# -----
#     flavor: small
#
# -----
# Set the web server cluster size by referencing the clusterSize user
# input. Small is 2 machines, medium is 3, and large defaults to 4.
# -----
#     count: '${input.clusterSize== "small" ? 2 : (input.clusterSize == "medium" ? 3 : 4)}'
#
# -----
# Set an environment variable to display object information under the
# Properties tab, post-deployment. Another example might be
# {env.blueprintID}
# -----
#     tags:
#       - key: cas.requestedBy
#         value: '${env.requestedBy}'
#
# -----
# You are free to add custom properties, which might be used to initiate
# an extensibility subscription, for example.
# -----
#     ABC-Company-ID: 9393
#
# -----
# Tag the web server to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with your site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
#     constraints:
#       - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
#     networks:
#       - name: '${resource.WP_Network.name}'
#         network: '${resource.WP_Network.id}'
#
# -----
# Run OS commands or scripts to further configure the web server,
# with operations such as setting a hostname, generating SSH private keys,
# or installing packages.
# -----
#     cloudConfig: |
#       #cloud-config
#       repo_update: true
#       repo_upgrade: all
#       packages:
#         - apache2
#         - php
#         - php-mysql

```

```

- libapache2-mod-php
- php-mcrypt
- mysql-client
runcmd:
- mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
- i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
- mysql -u root -pmysqlpassword -h ${resource.DBTier.networks[0].address} -e
"create database wordpress_blog;"
- mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
- sed -i -e s/"define('DB_NAME', 'database_name_here');"/"define('DB_NAME',
'wordpress_blog');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define('DB_USER', 'username_here');"/"define('DB_USER', 'root');"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define('DB_PASSWORD',
'password_here');"/"define('DB_PASSWORD', 'mysqlpassword');"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define('DB_HOST',
'localhost');"/"define('DB_HOST', '${resource.DBTier.networks[0].address}');"/ /var/www/html/
mywordpresssite/wp-config.php
- service apache2 reload
#
# -----
# Create the network that the database and web servers connect to.
# Choose a cloud agnostic network 'type' so that it can deploy to AWS,
# Azure, or vSphere. Then enter its property settings.
# -----
WP_Network:
  type: Cloud.Network
  properties:
#
# -----
# Descriptive name for the network. Does not become the network name
# upon deployment.
# -----
name: WP_Network
#
# -----
# Set the networkType to an existing network. You could also use a
# constraint tag to target a specific, like-tagged network.
# The other network types are private or public.
# -----
networkType: existing
#
# *****
#
# VMware hopes that you found this commented blueprint useful. Note that
# you can also access an API to create blueprints, or query for input
# schema that you intend to request. See the following Swagger
# documentation.

```



```
#
# www.mgmt.cloud.vmware.com/blueprint/api/swagger/swagger-ui.html
#
# *****
```

vRealize Automation Cloud Assembly 蓝图中的网络、安全性和负载均衡器示例

这些蓝图代码示例说明了一些基本网络、安全和负载均衡器配置。

有关详细网络 and 安全性实施场景的示例，请参见如下等 VMware 博客：

- [vRealize Automation Cloud Assembly 负载均衡器与 NSX-T 配合使用深入探讨](#)
- [使用 Cloud Assembly 和 NSX 实现网络自动化 - 第 1 部分](#)（包括使用 NSX-T 和 vCenter 云帐户和网络 CIDR）
- [使用 Cloud Assembly 和 NSX 实现网络自动化 - 第 2 部分](#)（包括使用现有的出站网络类型）
- [使用 Cloud Assembly 和 NSX 实现网络自动化 - 第 3 部分](#)（包括使用现有的按需安全组）
- [使用 Cloud Assembly 和 NSX 实现网络自动化 - 第 4 部分](#)（包括使用现有的按需负载均衡器）

有关所有蓝图结构定义选项的完整摘要，请参见 [vRealize Automation 资源类型结构定义](#)。

有关网络类型的详细信息，请参见在 [vRealize Automation Cloud Assembly 中使用网络和网络配置文件](#)。

| 资源场景 | 示例蓝图代码 |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 具有多个网卡的 vSphere 计算机 | <pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: image: ubuntu flavor: small networks: - name: '\${network-01.id}' deviceIndex: 0 - name: '\${network-02.id}' deviceIndex: 1 Cloud_vSphere_Network_1: type: Cloud.vSphere.Network properties: networkType: existing name: network-01 Cloud_vSphere_Network_2: type: Cloud.NSX.Network properties: networkType: existing name: network-02 </pre> |
| 使用内部 IP（而非公共 IP）的公有云计算机 | <pre> resources: wf_proxy: type: Cloud.Machine properties: image: ubuntu 16.04 flavor: small constraints: - tag: 'platform:vsphere' networks: - name: '\${resource.wf_net.id}' assignPublicIpAddress: false </pre> |
| 使用 NSX 网络组件类型的 NSX-V 或 NSX-T 路由网络 | <pre> Cloud_NSX_Network_1: type: Cloud.NSX.Network properties: networkType: routed </pre> |
| 为出站网络标记 NSX 逻辑交换机 有关此方案的详细信息，请参见社区博客帖子使用 Cloud Assembly 在 NSX 中创建标记。 | <pre> Cloud_NSX_Network_1: type: Cloud.NSX.Network properties: networkType: outbound tags: - key: app value: opencart </pre> |

| 资源场景 | 示例蓝图代码 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>对计算机网卡应用了限制标记的现有安全组</p> <p>要使用现有安全组，请为 Cloud.SecurityGroup 组件中的 securityGroupType 属性输入 <i>existing</i>。要创建按需安全组，请为 Cloud.SecurityGroup 组件中的 securityGroupType 属性输入 <i>new</i>。</p> <p>可以为 Cloud.SecurityGroup 组件分配标记，以使用标记限制分配现有安全组。无法在蓝图中使用不包含标记的安全组。</p> | <pre>formatVersion: 1 inputs: {} resources: allowSsh_sg: type: Cloud.SecurityGroup properties: securityGroupType: existing constraints: - tag: allowSsh compute: type: Cloud.Machine properties: image: centos flavor: small networks: - network: '\${resource.prod-net.id}' securityGroups: - '\${resource.allowSsh_sg.id}' prod-net: type: Cloud.Network properties: networkType: existing</pre> |

资源场景

示例蓝图代码

包含 1 臂负载均衡器的按需网络

```

inputs: {}
resources:
  mp-existing:
    type: Cloud.Network
    properties:
      name: mp-existing
      networkType: existing
  mp-wordpress:
    type: Cloud.vSphere.Machine
    properties:
      name: wordpress
      count: 2
      flavor: small
      image: tiny
      customizationSpec: Linux
      networks:
        - network: '${resource["mp-private"].id}'
  mp-private:
    type: Cloud.NSX.Network
    properties:
      name: mp-private
      networkType: private
      constraints:
        - tag: nsxt
  mp-wordpress-lb:
    type: Cloud.LoadBalancer
    properties:
      name: wordpress-lb
      internetFacing: false
      network: '${resource.mp-existing.id}'
      instances: '${resource["mp-wordpress"].id}'
      routes:
        - protocol: HTTP
          port: '80'
          instanceProtocol: HTTP
          instancePort: '80'
          healthCheckConfiguration:
            protocol: HTTP
            port: '80'
            urlPath: /index.pl
            intervalSeconds: 60
            timeoutSeconds: 30
            unhealthyThreshold: 5
            healthyThreshold: 2

```

包含负载均衡器的现有网络

```

formatVersion: 1
inputs:
  count:
    type: integer
    default: 1
resources:
  ubuntu-vm:
    type: Cloud.Machine
    properties:
      name: ubuntu
      flavor: small
      image: tiny
      count: '${input.count}'
      networks:

```

资源场景

示例蓝图代码

```

- network: '$
{resource.Cloud_NSX_Network_1.id}'
Provider_LoadBalancer_1:
  type: Cloud.LoadBalancer
  properties:
    name: OC-LB
    routes:
      - protocol: HTTP
        port: '80'
        instanceProtocol: HTTP
        instancePort: '80'
        healthCheckConfiguration:
          protocol: HTTP
          port: '80'
          urlPath: /index.html
          intervalSeconds: 60
          timeoutSeconds: 5
          unhealthyThreshold: 5
          healthyThreshold: 2
        network: '$
{resource.Cloud_NSX_Network_1.id}'
        internetFacing: false
        instances: '${resource["ubuntu-vm"].id}'
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing
    constraints:
      - tag: nsxt24prod

```

Puppet 支持的具有用户名和密码访问权限的蓝图

在本示例中，您将 Puppet 配置管理添加到在具有用户名和密码访问权限的 vCenter 计算资源上部署的蓝图。

此过程显示的示例是关于如何创建 Puppet 支持的可部署资源，该资源需要用户名和密码身份验证。用户名和密码访问权限意味着，用户必须从计算资源手动登录到 Puppet 主计算机，才能调用 Puppet 配置管理。

或者，您可以配置远程访问身份验证，从而在蓝图中设置配置管理，以便计算资源能在 Puppet 主计算机上进行身份验证。启用远程访问后，计算资源会自动生成密钥以满足密码身份验证。仍需要使用有效的用户名。

有关如何在 vRealize Automation Cloud Assembly 蓝图中配置不同 Puppet 场景的更多示例，请参见 [AWS Puppet 配置管理蓝图示例](#)和 [vCenter Puppet 配置蓝图示例](#)。

前提条件

- 在有效的网络上设置 Puppet Enterprise 实例。
- 使用集成功能将 Puppet Enterprise 实例添加到 vRealize Automation Cloud Assembly。请参见在 [vRealize Automation Cloud Assembly 中配置 Puppet Enterprise 集成](#)
- 设置 vSphere 帐户和 vCenter 计算资源。

步骤

1 将 Puppet 配置管理组件添加到所需蓝图的画布上的 vSphere 计算资源中。

- a 选择**基础架构 > 管理 > 集成**。
- b 单击**添加集成**，然后选择 Puppet。
- c 在 Puppet 配置页面上输入相应的信息。

| 配置 | 说明 | 示例值 |
|---------------------------------|------------------------------------------------------------------------------------------------------|------------------------------------|
| 主机名 | Puppet 主计算机的主机名或 IP 地址 | Puppet-Ubuntu |
| SSH 端口 | 用于在 vRealize Automation Cloud Assembly 和 Puppet 主计算机之间进行通信的 SSH 端口。（可选） | 不适用 |
| Autosign 密钥 | Puppet 主计算机上配置的共享密钥，节点应提供此共享密钥以支持 autosign 证书请求。 | 特定于用户 |
| 位置 | 指示 Puppet 主计算机是位于私有云还是公有云上。 注 仅当部署计算资源与 Puppet 主计算机之间存在连接时，才支持跨云部署。 | |
| Cloud proxy | 公有云帐户（如 Microsoft Azure 或 Amazon Web Services）不需要此配置。如果您使用的是基于 vCenter 的云帐户，请为您的帐户选择相应的 cloud proxy。 | 不适用 |
| 用户名 | Puppet 主计算机的 SSH 和 RBAC 用户名。 | 特定于用户。YAML 值为 “\${input.username}” |
| 密码 | Puppet 主计算机的 SSH 和 RBAC 密码。 | 特定于用户。YAML 值为 “\${input.password}” |
| Use sudo commands for this user | 选择对 procidd 使用 sudo 命令。 | true |
| 名称 | Puppet 主计算机名称。 | PEMasterOnPrem |
| 说明 | | |

2 将用户名和密码属性添加到 Puppet YAML，如以下示例中所示。

3 确保 Puppet YAML 的 remoteAccess 属性值设置为 authentication: username and password，如以下示例中所示。

示例：vCenter 用户名和密码 YAML 代码

以下示例显示了用于在 vCenter 计算资源上添加用户名和密码身份验证的具有代表性的 YAML 代码。

```
inputs:
  username:
    type: string
    title: Username
```

```

    description: Username to use to install Puppet agent
    default: puppet
password:
    type: string
    title: Password
    default: VMware@123
    encrypted: true
    description: Password for the given username to install Puppet agent
resources:
  Puppet-Ubuntu:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      imageRef: >-
        https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ubuntu-16.04-server-
cloudimg-amd64.ova
      remoteAccess:
        authentication: usernamePassword
        username: '${input.username}'
        password: '${input.password}'
  Puppet_Agent:
    type: Cloud.Puppet
    properties:
      provider: PEMasterOnPrem
      environment: production
      role: 'role::linux_webserver'
      username: '${input.username}'
      password: '${input.password}'
      host: '${Puppet-Ubuntu.*}'
      useSudo: true
      agentConfiguration:
        certName: '${Puppet-Ubuntu.address}'

```

AWS Puppet 配置管理蓝图示例

有多个选项可用于配置蓝图，以支持对 AWS 计算资源进行基于 Puppet 的配置管理。

使用用户名和密码对 AWS 进行 Puppet 管理

| 示例... | 示例蓝图 YAML |
|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>对任何受支持的 Amazon 计算机映像进行云配置身份验证。</p> | <pre>inputs: username: type: string title: Username default: puppet password: type: string title: Password encrypted: true default: VMware@123 resources: Webserver: type: Cloud.AWS.EC2.Instance properties: flavor: small image: centos cloudConfig: #cloud-config ssh_pwauth: yes chpasswd: list: \${input.username}:\${input.password} expire: false users: - default - name: \${input.username} lock_passwd: false sudo: ['ALL=(ALL) NOPASSWD:ALL'] groups: [wheel, sudo, admin] shell: '/bin/bash' ssh-authorized-keys: - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6/+vGbmKXoRpX dmettem@dmettem-m01.vmware.com runcmd: - echo "Defaults:\${input.username} !requiretty" >> /etc/sudoers.d/\${input.username} Puppet_Agent: type: Cloud.Puppet properties: provider: PEOonAWS environment: production role: 'role::linux_webserver' host: '\${Webserver.*}' osType: linux username: '\${input.username}' password: '\${input.password}' useSudo: true</pre> |
| <p>使用现有用户对自定义 Amazon 计算机映像进行云配置身份验证。</p> | <pre>inputs: username: type: string title: Username default: puppet</pre> |

示例...

示例蓝图 YAML

```

password:
  type: string
  title: Password
  encrypted: true
  default: VMware@123
resources:
  Webserver:
    type: Cloud.AWS.EC2.Instance
    properties:
      flavor: small
      image: centos
      cloudConfig: |
        #cloud-config
        runcmd:
          - sudo sed -e 's/. *PasswordAuthentication no.*/
PasswordAuthentication yes/' -i /etc/ssh/sshd_config
          - sudo service sshd restart
  Puppet_Agent:
    type: Cloud.Puppet
    properties:
      provider: PEOAWS
      environment: production
      role: 'role::linux_webserver'
      host: '${Webserver.*}'
      osType: linux
      username: '${input.username}'
      password: '${input.password}'
      useSudo: true

```

使用 generatedPublicPrivateKey 对 AWS 进行 Puppet 管理

示例...

示例蓝图 YAML

使用 generatedPublicPrivateKey
访问权限对 AWS 进行
remoteAccess.authentication 身份
验证

```

inputs: {}
resources:
  Machine:
    type: Cloud.AWS.EC2.Instance
    properties:
      flavor: small
      imageRef: ami-a4dc46db
      remoteAccess:
        authentication: generatedPublicPrivateKey
  Puppet_Agent:
    type: Cloud.Puppet
    properties:
      provider: puppet-BlueprintProvisioningITSuite
      environment: production
      role: 'role::linux_webserver'
      host: '${Machine.*}'
      osType: linux
      username: ubuntu
      useSudo: true
      agentConfiguration:
        runInterval: 15m
        certName: '${Machine.address}'
      useSudo: true

```

vCenter Puppet 配置蓝图示例

有多种方式可配置蓝图，以支持对 vCenter 计算资源进行基于 Puppet 的配置管理。

vSphere 上的 Puppet，具有用户名和密码身份验证

以下示例显示了 vSphere OVA 上 Puppet 的示例 YAML 代码，该 Puppet 具有用户名和密码身份验证。

表 6-2.

| 示例... | 示例蓝图 YAML |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>vSphere OVA 上 Puppet 的 YAML 代码，该 Puppet 具有用户名和密码身份验证。</p> | <pre>inputs: username: type: string title: Username default: puppet password: type: string title: Password encrypted: true default: VMware@123 resources: Puppet_Agent: type: Cloud.Puppet properties: provider: PEonAWS environment: dev role: 'role::linux_webserver' username: '\${input.username}' password: '\${input.password}' useSudo: true host: '\${Webserver.*}' osType: linux agentConfiguration: runInterval: 15m certName: '\${Machine.address}' Webserver: type: Cloud.vSphere.Machine properties: cpuCount: 1 totalMemoryMB: 1024 imageRef: >- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova cloudConfig: #cloud-config ssh_pwauth: yes chpasswd: list: \${input.username}:\${input.password} expire: false users: - default - name: \${input.username} lock_passwd: false sudo: ['ALL=(ALL) NOPASSWD:ALL'] groups: [wheel, sudo, admin] shell: '/bin/bash' ssh-authorized-keys: - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX dmettem@dmettem-m01.vmware.com runcmd: - echo "Defaults:\${input.username}"</pre> |
| <p>vSphere OVA 上 Puppet 的 YAML 代码，该 Puppet 对计算资源进行用户名和密码身份验证。</p> | <pre>inputs: username: type: string title: Username default: puppet</pre> |

表 6-2. (续)

| 示例... | 示例蓝图 YAML |
|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <pre> password: type: string title: Password encrypted: true default: VMware@123 resources: Puppet_Agent: type: Cloud.Puppet properties: provider: PEonAWS environment: dev role: 'role::linux_webserver' username: '\${input.username}' password: '\${input.password}' useSudo: true host: '\${Webserver.*}' osType: linux agentConfiguration: runInterval: 15m certName: '\${Machine.address}' Webserver: type: Cloud.vSphere.Machine properties: cpuCount: 1 totalMemoryMB: 1024 imageRef: >- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova cloudConfig: #cloud-config ssh_pwauth: yes chpasswd: list: \${input.username}:\${input.password} expire: false users: - default - name: \${input.username} lock_passwd: false sudo: ['ALL=(ALL) NOPASSWD:ALL'] groups: [wheel, sudo, admin] shell: '/bin/bash' ssh-authorized-keys: - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX dmettem@dmettem-m01.vmware.com runcmd: - echo "Defaults:\${input.username} </pre> |
| <p>vCenter 上 Puppet 的 YAML 代码，该 Puppet 对计算资源启用了远程访问密码身份验证。</p> | <pre> inputs: username: type: string title: Username description: Username to use to install Puppet agent default: puppet password: type: string title: Password default: VMware@123 encrypted: true </pre> |

表 6-2. (续)

| 示例... | 示例蓝图 YAML |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <pre>description: Password for the given username to install Puppet agent resources: Puppet-Ubuntu: type: Cloud.vSphere.Machine properties: flavor: small imageRef: >- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova remoteAccess: authentication: usernamePassword username: '\${input.username}' password: '\${input.password}' Puppet_Agent: type: Cloud.Puppet properties: provider: PEMasterOnPrem environment: production role: 'role::linux_webserver' username: '\${input.username}' password: '\${input.password}' host: '\${Puppet-Ubuntu.*}' useSudo: true agentConfiguration: certName: '\${Puppet-Ubuntu.address}'</pre> |

vSphere 上的 Puppet，具有生成的 PublicPrivateKey 身份验证

表 6-3.

| 示例... | 示例蓝图 YAML |
|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| vSphere OVA 上 Puppet 的 YAML 代码，具有针对计算资源的生成的 PublicPrivateKey 身份验证。 | <pre> inputs: {} resources: Machine: type: Cloud.vSphere.Machine properties: flavor: small imageRef: >- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova remoteAccess: authentication: generatedPublicPrivateKey Puppet_Agent: type: Cloud.Puppet properties: provider: puppet-BlueprintProvisioningITSuite environment: production role: 'role::linux_webserver' host: '\${Machine.*}' osType: linux username: ubuntu useSudo: true agentConfiguration: runInterval: 15m certName: '\${Machine.address}' - echo "Defaults:\${input.username} </pre> |

如何使用 vRealize Automation Cloud Assembly 商城

要快速启动资源库，请从 vRealize Automation Cloud Assembly 商城下载文件。

商城提供了在 [VMware Solution Exchange](#) 上管理的已完成蓝图和开放虚拟化映像。标记有 cloud assembly 的 Solution Exchange 文件会显示在 vRealize Automation Cloud Assembly “商城”选项卡下。

如何访问商城

在 vRealize Automation Cloud Assembly 中，选择**基础架构 > 连接 > 集成**。单击**添加集成**，单击 **My VMware**，并提供您的 My VMware 帐户凭据。

如何下载和使用商城蓝图文件

在**商城**选项卡中，单击**获取**，并接受蓝图 EULA。然后，您可以将蓝图添加到 vRealize Automation Cloud Assembly 项目，或仅下载它。您可以在**蓝图**选项卡中上载蓝图。

对于基于项目的示例，假设您是大数据工作的项目管理员。为了帮助您的团队，您可以找到添加到团队项目中的商城 Hadoop 蓝图。然后，自定义您的资源环境的蓝图，并发布它。之后，将蓝图导入 vRealize Automation Service Broker 目录，以便您的团队可以部署它。

如何下载和使用商城映像文件

在**商城**选项卡中，单击**获取**，并接受 OVF 或 OVA 映像 EULA。之后，您可以下载 OVF 或 OVA 映像，并在蓝图代码中引用该映像。

继续前一示例，您的团队可能需要访问 Hadoop 本身的版本。您可以下载 Hadoop OVF，并将它添加到云帐户资源，例如 vCenter Server Content Library。然后，您可以更新需要指向 OVF 映像的任何蓝图代码。

如何使用可扩展性扩展应用程序生命周期并实现自动化

可以将可扩展性操作或 vRealize Orchestrator 工作流与可扩展性订阅一起使用来扩展应用程序生命周期。

借助 vRealize Automation Cloud Assembly 可扩展性，您可以使用订阅为事件分配可扩展性操作或 vRealize Orchestrator 工作流。发生指定的事件时，订阅将启动该操作或工作流以使其运行，并通知所有订阅者。

可扩展性操作

可扩展性操作是小型的轻量级代码脚本，用于指定操作和操作执行方式。可以从预定义的 vRealize Automation Cloud Assembly 操作模板或从 ZIP 文件导入可扩展性操作。您还可以使用操作编辑器为您的可扩展性操作创建自定义脚本。在一个脚本中将多个操作脚本链接到一起时，就创建了一个操作流。通过使用操作流，可以创建操作序列。有关使用操作流的信息，请参见[操作流是什么](#)。

vRealize Orchestrator 工作流

通过将 vRealize Automation Cloud Assembly 与现有的 vRealize Orchestrator 环境相集成，您可以在可扩展性订阅中使用工作流。

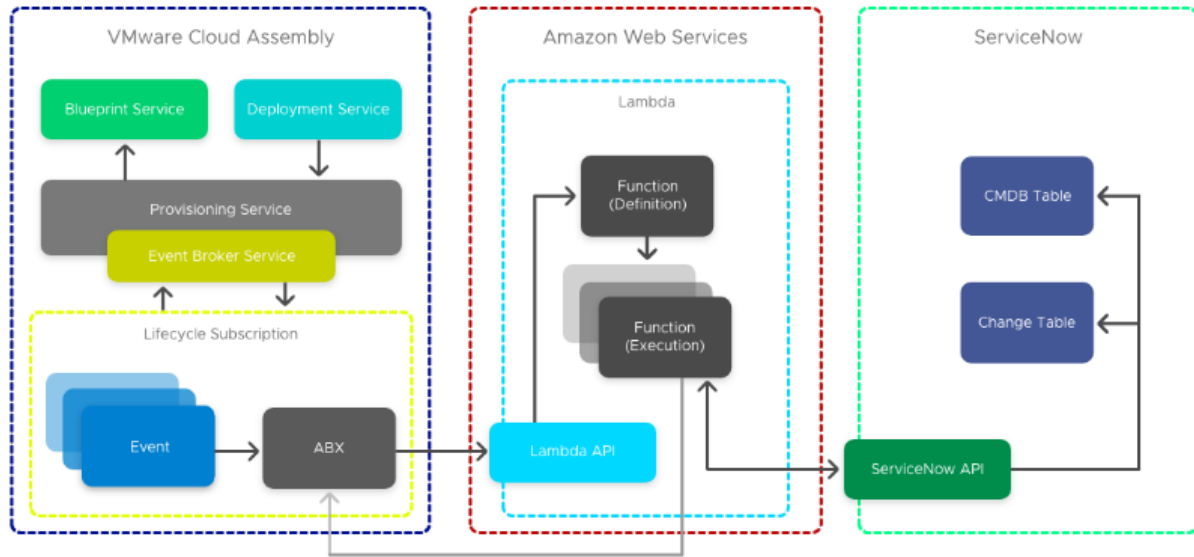
可扩展性操作订阅

可以为 vRealize Automation Cloud Assembly 订阅分配可扩展性操作，以扩展应用程序生命周期。

注 以下订阅只是一些用例示例，未涵盖所有可扩展性操作功能。

如何使用可扩展性操作将 Cloud Assembly 与 ServiceNow 集成

使用可扩展性操作，可以将 vRealize Automation Cloud Assembly 与企业 ITSM（例如 ServiceNow）集成。

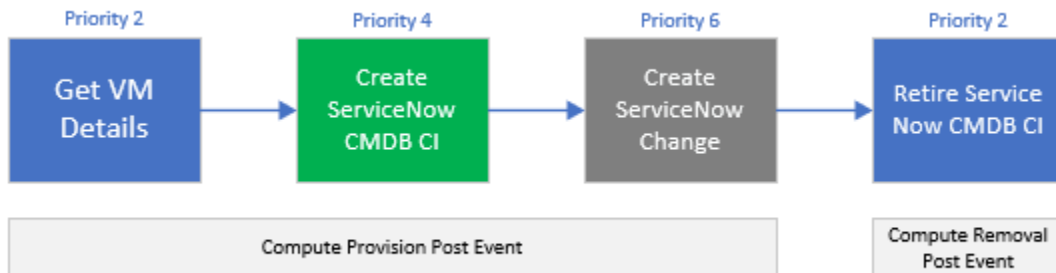


企业用户通常将其云计算管理平台与 IT 服务管理 (IT Service Management, ITSM) 和配置管理数据库 (Configuration Management Database, CMDB) 平台集成以实现合规性。按照此示例，可以使用可扩展性操作脚本将 vRealize Automation Cloud Assembly 与适用于 CMDB 和 ITSM 的 ServiceNow 集成。

注 还可以使用 vRealize Orchestrator 工作流将 ServiceNow 与 vRealize Automation Cloud Assembly 集成。有关使用工作流集成 ServiceNow 的信息，请参见[如何使用 vRealize Orchestrator 工作流将适用于 ITSM 的 Cloud Assembly 与 ServiceNow 集成](#)。

要创建此集成，需要使用四个可扩展性操作脚本。在置备期间，前三个脚本在发生计算资源置备后事件时按顺序启动。第四个脚本在发生计算资源移除后事件时触发。

有关事件主题的详细信息，请参阅[随 vRealize Automation Cloud Assembly 提供的事件主题](#)。



获取虚拟机详细信息

“获取虚拟机详细信息”脚本可获取创建 CI 所需的其他负载详细信息，以及存储在 Amazon Web Services Systems Manager 参数存储 (SSM) 中的标识令牌。此外，此脚本还可以使用其他属性更新 `customProperties` 供以后使用。

创建 ServiceNow CMDB CI

“创建 ServiceNow CMDB CI”脚本将 ServiceNow 实例 URL 作为输入传递，并将实例存储在 SSM 中以满足安全要求。此脚本还会读取 ServiceNow CMDB 唯一记录标识符响应 (`sys_id`)。此脚本将该响应作为输出传递，并在创建期间写入自定义属性 `serviceNowSysId`。此值用于在销毁实例时将 CI 标记为已注销。

注 可能需要向 vRealize Automation services Amazon Web Services 角色分配其他权限，以允许 Lambda 访问 SSM 参数存储。

创建 ServiceNow 变更

此脚本通过将 ServiceNow 实例 URL 作为输入传递，并将 ServiceNow 凭据存储在 SSM 中以满足安全要求，完成 ITSM 集成。

创建 ServiceNow 变更

“注销 ServiceNow CMDB CI”脚本根据在创建脚本中创建的自定义属性 `serviceNowSysId` 提示 ServiceNow 停止并将 CI 标记为已注销。

前提条件

- 配置此集成之前，筛选具有条件蓝图属性的所有事件订阅：

```
event.data["customProperties"]["enable_servicenow"] == "true"
```

注 此属性存在于需要 ServiceNow 集成的蓝图上。

- 已安装了 Python 应用程序。

有关筛选订阅的详细信息，请参见[创建可扩展性订阅](#)。

步骤

- 从虚拟机打开命令提示符。
- 运行“获取虚拟机详细信息”脚本。

```
from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    baseUri = inputs['url']
    casToken = client.get_parameter(Name="casToken",WithDecryption=True)

    url = baseUri + "/iaas/login"
    headers = {"Accept":"application/json","Content-Type":"application/json"}
    payload = {"refreshToken":casToken['Parameter']['Value']}
```

```

results = requests.post(url,json=payload,headers=headers)

bearer = "Bearer "
bearer = bearer + results.json()["token"]

deploymentId = inputs['deploymentId']
resourceId = inputs['resourceIds'][0]

print("deploymentId: " + deploymentId)
print("resourceId:" + resourceId)

machineUri = baseUrl + "/iaas/machines/" + resourceId
headers = {"Accept":"application/json","Content-Type":"application/json",
"Authorization":bearer }
resultMachine = requests.get(machineUri,headers=headers)
print("machine: " + resultMachine.text)

print( "serviceNowCPUCount: " + json.loads(resultMachine.text)["customProperties"]
["cpuCount"] )
print( "serviceNowMemoryInMB: " + json.loads(resultMachine.text)["customProperties"]
["memoryInMB"] )

#update customProperties
outputs = {}
outputs['customProperties'] = inputs['customProperties']
outputs['customProperties']['serviceNowCPUCount'] = int(json.loads(resultMachine.text)
["customProperties"]["cpuCount"])
outputs['customProperties']['serviceNowMemoryInMB'] = json.loads(resultMachine.text)
["customProperties"]["memoryInMB"]
return outputs

```

3 运行 CMDB 配置项创建操作。

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):

    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    table_name = "cmdb_ci_vmware_instance"
    url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'name': inputs['customProperties']['serviceNowHostname'],
        'cpus': int(inputs['customProperties']['serviceNowCPUCount']),
        'memory': inputs['customProperties']['serviceNowMemoryInMB'],
        'correlation_id': inputs['deploymentId'],
        'disks_size': int(inputs['customProperties']['provisionGB']),
        'location': "Sydney",
        'vcenter_uuid': inputs['customProperties']['vcUuid'],
        'state': 'On',
        'sys_created_by': inputs['__metadata']['userName'],

```

```

        'owned_by': inputs['__metadata']['userName']
    }
    results = requests.post(
        url,
        json=payload,
        headers=headers,
        auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
    )
    print(results.text)

    #parse response for the sys_id of CMDB CI reference
    if json.loads(results.text)['result']:
        serviceNowResponse = json.loads(results.text)['result']
        serviceNowSysId = serviceNowResponse['sys_id']
        print(serviceNowSysId)

    #update the serviceNowSysId customProperty
    outputs = {}
    outputs['customProperties'] = inputs['customProperties']
    outputs['customProperties']['serviceNowSysId'] = serviceNowSysId;
    return outputs

```

4 运行创建操作脚本。

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    table_name = "change_request"
    url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'short_description': 'Provision CAS VM Instance'
    }
    results = requests.post(
        url,
        json=payload,
        headers=headers,
        auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
    )
    print(results.text)

```

结果

vRealize Automation Cloud Assembly 已成功与 ITSM ServiceNow 集成。

后续步骤

如果需要，可以使用 CMDB 配置项注销操作注销 CI:

```
from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    tableName = "cmdb_ci_vmware_instance"
    sys_id =inputs['customProperties']['serviceNowSysId']
    url = "https://" + inputs['instanceUrl'] + "/api/now/"+tableName+"/"+{0}".format(sys_id)
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'state': 'Retired'
    }

    results = requests.put(
        url,
        json=payload,
        headers=headers,
        auth=(inputs['username'], inputs['password'])
    )
    print(results.text)
```

有关如何使用可扩展性操作在 vRealize Automation Cloud Assembly 中集成 ServiceNow 的更多信息，请参见为 [ServiceNow 集成使用基于操作的可扩展性扩展 Cloud Assembly](#)。

如何使用可扩展性操作在置备期间标记虚拟机

您可以将可扩展性操作与订阅配合使用，以自动执行和简化标记虚拟机操作。

作为云管理员，您可以使用可扩展性操作和可扩展性订阅来创建使用指定的输入和输出自动标记的部署。针对包含“标记虚拟机”订阅的项目创建新部署时，部署事件将触发并运行“标记虚拟机”脚本，并且会自动应用标记。这样可以节省时间并提高效率，同时还可以简化部署管理。

前提条件

- 对云管理员凭据的访问权限。
- Lambda 函数的 Amazon Web Services 角色。

步骤

- 1 导航到 **可扩展性 > 库 > 操作 > 新建操作**，然后使用以下参数创建操作。

| 参数 | 说明 |
|------|-----------------------------|
| 操作名称 | 可扩展性操作名称，最好以“TagVM”作为前缀或后缀。 |
| 项目 | 要针对其测试可扩展性操作的项目。 |

| 参数 | 说明 |
|-------|--------|
| 操作模板 | 标记虚拟机 |
| 运行时 | Python |
| 脚本源代码 | 编写脚本 |

2 输入 **Handler** 作为主函数。

3 添加标记输入以用于测试可扩展性操作。

例如，resourceNames = ["DB_VM"], target = world。

4 要保存操作，单击**保存**。

5 要测试操作，单击**测试**。

6 要退出操作编辑器，单击**关闭**。

7 导航到**可扩展性 > 订阅**。

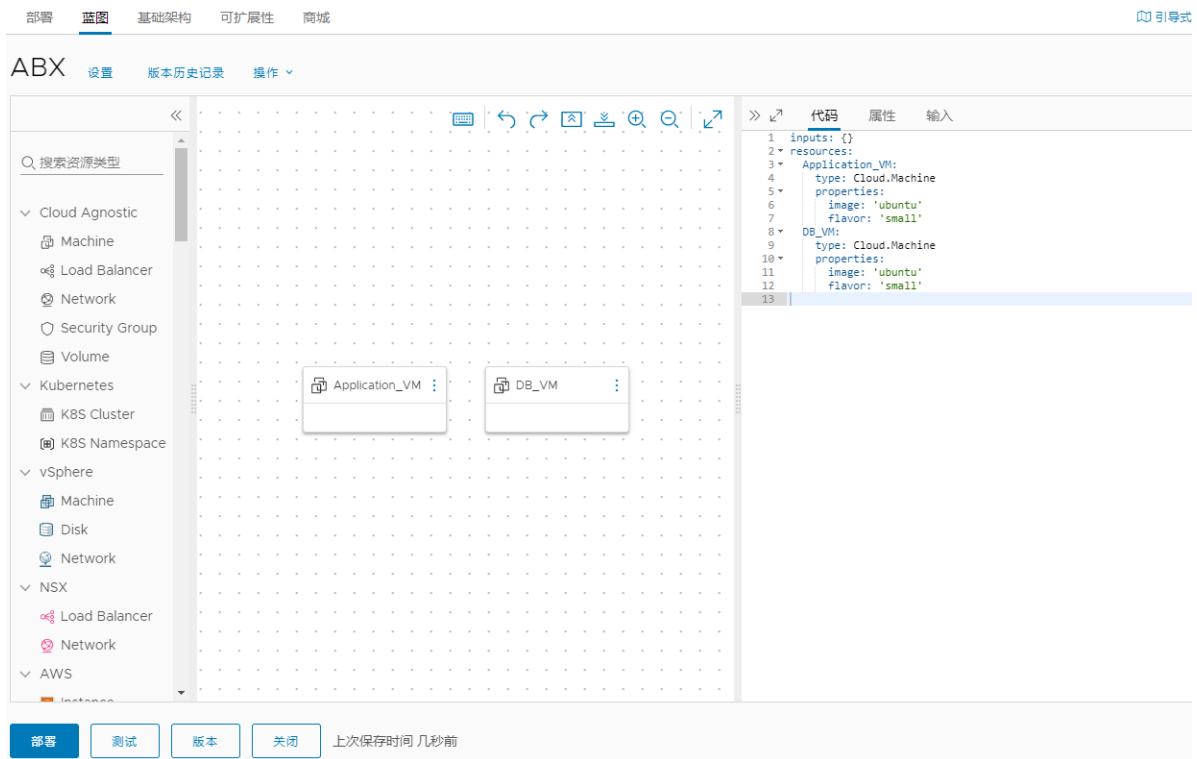
8 单击**新建订阅**。

9 输入以下订阅详细信息。

| 详细信息 | 设置 |
|--------|----------------------------------------------------------|
| 事件主题 | 选择与虚拟机的标记阶段相关的事件主题。例如，计算资源分配。 注 标记必须是订阅模式的一部分。 |
| 阻止 | 将订阅的超时设置为 1 分钟。 |
| 可运行项类型 | 选择可扩展性操作可运行类型。 |
| 可运行 ID | 选择自定义可扩展性操作。 |

10 要保存自定义可扩展性操作订阅，单击**创建**。

11 创建包含以下两个虚拟机的新蓝图：应用程序虚拟机和数据库虚拟机。



12 要部署虚拟机，单击部署。

13 在部署过程中，验证事件是否已启动，并验证可扩展性操作是否已运行。

14 要验证标记是否正确应用，导航到基础架构 > 资源 > 计算机。

了解有关可扩展性操作的更多信息

基于操作的可扩展性在 vRealize Automation Cloud Assembly 中使用简化的代码脚本自动执行可扩展性操作。

基于操作的可扩展性提供轻型灵活的运行时引擎接口，您可以在其中定义可编写脚本的小型操作并将其配置为在发生事件代理服务 (EBS) 提供的特定事件时启动。

您可以在 vRealize Automation Cloud Assembly 中创建这些可扩展性操作代码脚本，并将其分配给订阅。与工作流类似，当可扩展性订阅中包括的事件发生时，可扩展性操作脚本将触发。可使用可扩展性操作脚本实现更轻型、更简单的任务和步骤自动化。这些脚本还托管在云上，与通过使用 vRealize Orchestrator 服务器托管在内部部署的工作流截然相反。有关将 vRealize Automation Cloud Assembly 与 vRealize Orchestrator 服务器集成的详细信息，请参见在 [Cloud Assembly 中配置 vRealize Orchestrator 集成](#)。

基于操作的可扩展性具有以下优势：

- 可取代 vRealize Orchestrator 工作流，使用小型且可重用的可编写脚本的操作实现轻型集成和自定义。
- 提供一种方法，可用于重用包含可重用的参数化操作的操作模板。

可以通过编写用户定义的操作脚本代码或将预定义的脚本代码导入为 ZIP 包来创建可扩展性操作。基于操作的可扩展性支持 Node.js 和 Python 运行时环境，并依赖于 Amazon Web Services Lambda。因此，您必须使用一个具有 Amazon Web Services 身份与访问管理 (IAM) 的有效订阅，并将 Amazon Web Services 配置为 vRealize Automation Cloud Assembly 中的端点。有关 Amazon Web Services Lambda 入门的信息，请参见 [ABX: Cloud Assembly 服务的无服务器可扩展性](#)。

注 可扩展性操作是特定于项目的。

如何创建可扩展性操作

使用 vRealize Automation Cloud Assembly，您可以创建可扩展性操作，以便在可扩展性订阅中使用。

可扩展性操作是高度可自定义、轻型且灵活的方法，用于通过用户定义的脚本代码和操作模板扩展应用程序生命周期。操作模板包含预定义的参数，这些参数帮助构成了可扩展性操作的基础。

以下两种方法可用于创建可扩展性操作：

- 为可扩展性操作脚本编写用户定义的代码。

注 在可扩展性操作编辑器中编写用户定义的代码时，可能需要有效的 Internet 连接。

- 为可扩展性操作导入 ZIP 软件包格式的部署软件包。有关为可扩展性操作创建 ZIP 软件包的信息，请参见为 [Python 运行时可扩展性操作创建 ZIP 软件包](#)或为 [Node.js 运行时可扩展性操作创建 ZIP 软件包](#)。

以下步骤介绍了创建使用 Amazon Web Services 作为 FaaS 提供程序的可扩展性操作的过程。

前提条件

- 具备活动且有效项目中的成员资格。
- 已为 Lambda 函数配置 Amazon Web Services 角色。例如，AWSLambdaBasicExecutionRole。
- 已启用云管理员角色或 iam:PassRole 权限。

步骤

- 1 选择**可扩展性 > 库 > 操作**。
- 2 单击**新建操作**。
- 3 输入操作的名称，然后选择一个项目。
- 4 单击**下一步**。
- 5 搜索并选择操作模板。

注 要在不使用操作模板的情况下创建自定义操作，请选择**自定义脚本**。

此时将显示新的可配置参数。

- 6 选择**编写脚本**或**导入软件包**。
- 7 选择操作**运行时**。

- 8 为操作的入口点输入 **主函数** 名称。

注 对于从 ZIP 软件包导入的操作，主函数还必须包括含入口点的脚本文件的名称。例如，如果主脚本文件的标题为 `main.py`，并且输入点为 `handler (context, inputs)`，则主函数的名称必须为 *main.handler*。

- 9 定义操作的**输入**和**输出**参数。
- 10 （可选）将应用程序依赖关系添加到操作。

注 对于从 ZIP 软件包导入的操作，将自动添加应用程序依赖关系。

- 11 要定义超时和内存限制，请启用**设置自定义超时和限制**选项。
- 12 要测试操作，请单击**保存**，然后单击**测试**。

后续步骤

创建并验证可扩展性操作后，便可将其分配给订阅。

注 可扩展性订阅使用可扩展性操作的最新发布版本。创建新版本的操作后，请单击编辑器窗口右上角的**版本**。要发布您打算在订阅中使用的操作的版本，单击**发布**。

导出和导入可扩展性操作

使用 vRealize Automation Cloud Assembly，您可以导出和导入可扩展性操作以在不同的项目中使用。

前提条件

现有的可扩展性操作。

步骤

- 1 导出可扩展性操作。
 - a 导航到**可扩展性 > 库 > 操作**。
 - b 选择一个可扩展性操作，然后单击**导出**。
操作脚本及其依赖项以 ZIP 文件的形式保存在本地环境中。
- 2 导入可扩展性操作。
 - a 导航到**可扩展性 > 库 > 操作**。
 - b 单击**导入**。
 - c 选择导出的可扩展性操作并将其分配给项目。
 - d 单击**导入**。

注 如果导入的可扩展性操作已分配给指定的项目，系统会提示您选择冲突解决策略。

备选方法 您还可以直接从操作编辑器中选择**导入软件包**选项来导入操作脚本。

为 Python 运行时可扩展性操作创建 ZIP 软件包

可以创建一个包含 vRealize Automation Cloud Assembly 可扩展性操作所用 Python 脚本和依赖关系的 ZIP 软件包。

为可扩展性操作构建脚本的方法有两种：

- 在 vRealize Automation Cloud Assembly 的可扩展性操作编辑器中直接编写脚本。
- 在本地环境中创建脚本，然后将其与任何相关依赖关系一起添加到 ZIP 软件包。

通过使用 ZIP 软件包，可以为操作脚本和依赖关系创建自定义的预配置模板，然后可以将其导入到 vRealize Automation Cloud Assembly，以便在可扩展性操作中使用。

另外，如果与操作脚本中的依赖关系相关联的模块无法由 vRealize Automation Cloud Assembly 服务解析，例如，环境中缺少 Internet 访问权限，也可以使用 ZIP 软件包。

此外，还可以使用 ZIP 软件包创建包含多个 Python 脚本文件的可扩展性操作。使用多个脚本文件有助于组织可扩展性操作代码的结构。

前提条件

如果使用的是 Python 3.3 或更低版本，请下载并配置 PIP 软件包安装程序。请参见 [Python 软件包索引](#)。

步骤

- 1 在本地计算机上，为操作脚本和依赖关系创建一个文件夹。

例如，/home/user1/zip-action。

- 2 将一个或多个主 Python 操作脚本添加到该文件夹。

例如，/home/user1/zip-action/main.py。

- 3 （可选）将 Python 脚本的任何依赖关系添加到该文件夹。

- a 创建包含依赖关系的 requirements.txt 文件。请参见[要求文件](#)。
- b 打开 Linux shell。

注 vRealize Automation Cloud Assembly 中基于操作的可扩展性的运行时基于 Linux。因此，在 Windows 环境中编译的任何 Python 依赖关系可能会导致生成的 ZIP 软件包无法用于创建可扩展性操作。因此，必须使用 Linux shell。

- c 运行以下命令，在脚本文件夹中安装 requirements.txt 文件：

```
pip install -r requirements.txt --target=home/user1/zip-action
```

- 4 在分配的文件夹中，选择您的脚本元素以及（如果适用）requirements.txt 文件并将其压缩为 ZIP 软件包。

注 脚本元素和依赖关系元素必须存储在 ZIP 软件包的根级别。在 Linux 环境中创建 ZIP 软件包时，可能会遇到软件包内容未存储在根级别的问题。如果遇到此问题，请通过在命令行 shell 中运行 `zip -r` 命令来创建软件包。

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

后续步骤

使用 ZIP 软件包创建可扩展性操作脚本。请参见[如何创建可扩展性操作](#)。

为 Node.js 运行时可扩展性操作创建 ZIP 软件包

可以创建一个包含 vRealize Automation Cloud Assembly 可扩展性操作所用 Node.js 脚本和依赖关系的 ZIP 软件包。

为可扩展性操作构建脚本的方法有两种：

- 在 vRealize Automation Cloud Assembly 的可扩展性操作编辑器中直接编写脚本。
- 在本地环境中创建脚本，然后将其与任何相关依赖关系一起添加到 ZIP 软件包。

通过使用 ZIP 软件包，可以为操作脚本和依赖关系创建自定义的预配置模板，然后可以将其导入到 vRealize Automation Cloud Assembly，以便在可扩展性操作中使用。

另外，如果与操作脚本中的依赖关系相关联的模块无法由 vRealize Automation Cloud Assembly 服务解析，例如，环境中缺少 Internet 访问权限，也可以使用 ZIP 软件包。

此外，还可以使用软件包创建包含多个 Node.js 脚本文件的可扩展性操作。使用多个脚本文件有助于组织可扩展性操作代码的结构。

步骤

- 1 在本地计算机上，为操作脚本和依赖关系创建一个文件夹。
例如，/home/user1/zip-action。
- 2 将一个或多个主 Node.js 操作脚本添加到该文件夹。
例如，/home/user1/zip-action/main.js。
- 3 （可选）将 Node.js 脚本的任何依赖关系添加到该文件夹。
 - a 在脚本文件夹中创建包含依赖关系的 package.json 文件。请参见[创建 package.json 文件和在 package.json 文件中指定 dependencies 和 devDependencies](#)。
 - b 打开命令行 shell。

- c 导航到为操作脚本和依赖关系创建的文件夹。

```
cd /home/user1/zip-action
```

- d 运行以下命令，在脚本文件夹中安装 package.json 文件：

```
npm install --production
```

注 此命令会在文件夹中创建 node_modules 目录。

- 4 在分配的文件夹中，选择您的脚本元素以及（如果适用）node_modules 目录并将其压缩为 ZIP 软件包。

注 脚本元素和依赖关系元素必须存储在 ZIP 软件包的根级别。在 Linux 环境中创建 ZIP 软件包时，可能会遇到软件包内容未存储在根级别的问题。如果遇到此问题，请通过在命令行 shell 中运行 zip -r 命令来创建软件包。

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

后续步骤

使用 ZIP 软件包创建可扩展性操作脚本。请参见[如何创建可扩展性操作](#)。

配置特定于云的可扩展性操作

可以将可扩展性操作配置为与您的云帐户配合使用。

当创建可扩展性操作时，可以配置该操作并将其链接到各个云端帐户：

- Microsoft Azure
- Amazon Web Services

前提条件

需要有效的云帐户。

步骤

- 1 选择**可扩展性 > 库 > 操作**。
- 2 单击**新建操作**。
- 3 根据需输入操作参数。
- 4 在 **FaaS 提供程序**下拉菜单中，选择您的云帐户提供程序或选择**自动**。

注 如果选择**自动**，操作会自动定义 FaaS 提供商。

- 5 单击**保存**。

结果

可扩展性操作已链接，可与配置的云帐户配合使用。

配置内部部署可扩展性操作

您可以将可扩展性操作配置为使用内部部署 FaaS 提供程序，而不是 Amazon Web Services 或 Microsoft Azure 云帐户。

通过使用内部部署 FaaS 提供程序来执行可扩展性操作，您可以在 vRealize Automation Cloud Assembly 可扩展性订阅中使用内部部署服务，例如 LDAP、CMDB 或 vCenter 数据中心。

步骤

- 1 选择**可扩展性 > 库 > 操作**。
- 2 单击**新建操作**。
- 3 输入可扩展性操作的名称和项目。
- 4 （可选）输入可扩展性操作的描述。
- 5 单击**下一步**。
- 6 创建或导入您的可扩展性操作脚本。
- 7 单击 **FaaS 提供程序** 下拉菜单，然后选择**内部部署**。
- 8 要保存新的可扩展性操作，单击**保存**。

后续步骤

在 vRealize Automation Cloud Assembly 可扩展性订阅中使用创建的可扩展性操作。

操作流是什么

操作流是一组可扩展性操作脚本，用于进一步扩展生命周期并自动化。

所有操作流均以 flow_start 开头，并以 flow_end 结尾。可以通过使用以下操作流元素将多个可扩展性操作脚本链接在一起：

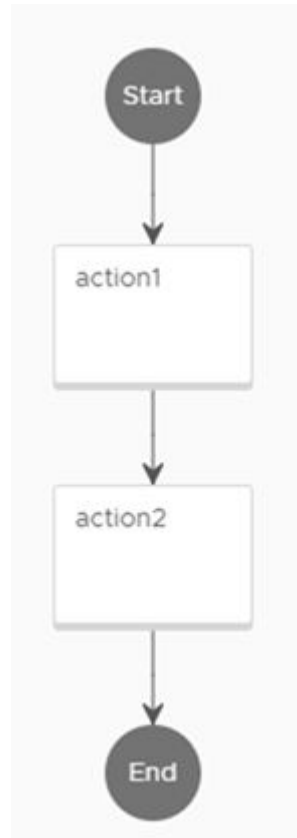
- **顺序操作流** - 多个可扩展性操作脚本按顺序运行。
- **分叉操作流** - 多个可扩展性操作脚本或操作流拆分为多个路径并构成相同的输出。
- **联接操作流** - 多个可扩展性操作脚本或操作流联接到一起并构成相同的输出。
- **条件操作流** - 多个可扩展性操作脚本或操作流在满足某个条件后运行。

顺序操作流

多个可扩展性操作脚本按顺序运行。

```
version: "1"
flow:
  flow_start:
    next: action1
  action1:
    action: <action_name>
    next: action2
  action2:
    action: <action_name>
    next: flow_end
```

注 通过将前一个操作分配为 `next`: 操作, 可以循环回到前一个操作。例如, 在此示例中, 输入 `next: action1` 可以重新运行 **action1** 并重新开始操作序列, 而不要输入 `next: flow_end`。

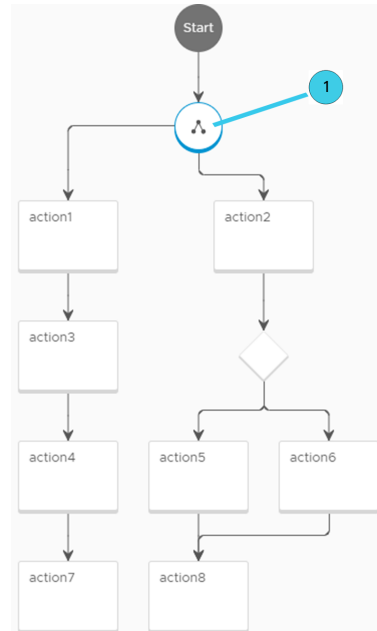


分叉操作流

多个可扩展性操作脚本或操作流拆分路径以构成相同的输出。

```
version: "1"
flow:
  flow_start:
    next: forkAction
  forkAction:
    fork:
      next: [action1, action2]
  action1:
    action: <action_name>
    next: action3
  action3:
    action: <action_name>
    next: action4
  action4:
    action: <action_name>
    next: action7
  action7:
    action: <action_name>
  action2:
    action: <action_name>
```

注 通过将前一个操作分配为 `next: 操作`，可以循环回到前一个操作。例如，输入 `next: action1` 可以重新运行 **action1** 并重新开始操作序列，而不要输入 `next: flow_end` 来结束操作流。



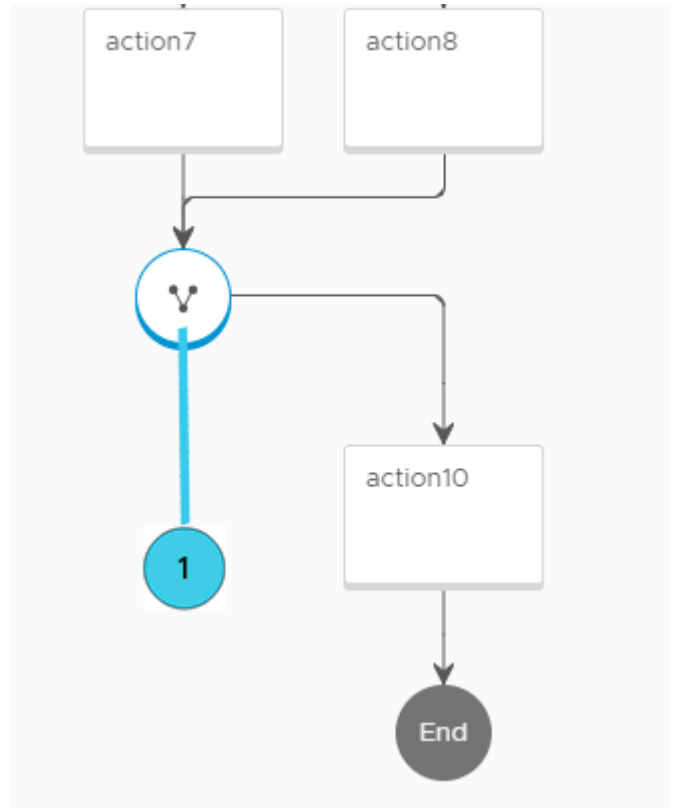
① 分叉元素

联接操作流

多个可扩展性操作脚本或操作流将路径联接到一起并构成相同的输出。

```
version: "1"
action7:
  action: <action_name>
  next: joinElement
action8:
  action: <action_name>
  next: joinElement
joinElement:
  join:
    type: all
    next: action10
action10:
  action: <action_name>
  next: flow_end
```

注 通过将前一个操作分配为 `next`: 操作，可以循环回到前一个操作。例如，在此示例中，输入 `next: action1` 可以重新运行 **action1** 并重新开始操作序列，而不要输入 `next: flow_end`。



① 联接元素

条件操作流

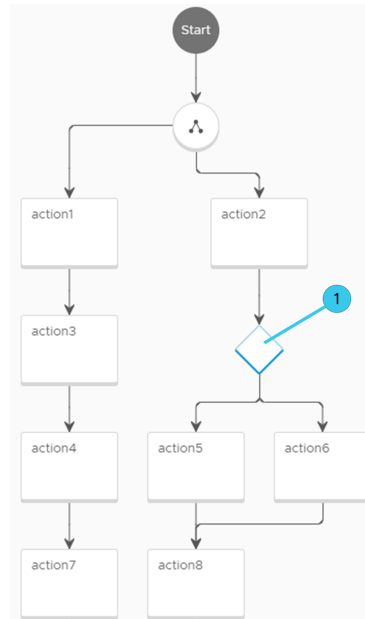
在使用开关元素满足某个条件时运行的多个可扩展性操作脚本或操作流。

在某些情况下，条件必须为 `true`，操作才能运行。而在其他情况下，如此示例中所示，必须满足参数值，操作才能运行。如果不满足任何条件，操作流将失败。

```

version: 1
id: 1234
name: Test
inputs: ...
outputs: ...
flow:
  flow_start:
    next: forkAction
  forkAction:
    fork:
      next: [action1, action2]
  action1:
    action: <action_name>
    next: action3
  action3:
    action: <action_name>
    next: action4
  action4:
    action: <action_name>
    next: action7
  action7:
    action: <action_name>
    next: joinElement
  action2:
    action: <action_name>
    next: switchAction
  switchAction:
    switch:
      "${1 == 1}": action5
      "${1 != 1}": action6
  action5:
    action: <action_name>
    next: action8
  action6:
    action: <action_name>
    next: action8
  action8:
    action: <action_name>

```



① 开关元素

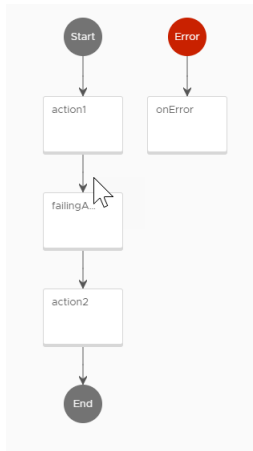
注 通过将前一个操作分配为 `next: 操作`，可以循环回到前一个操作。例如，输入 `next: action1` 可以重新运行 **action1** 并重新开始操作序列，而不要输入 `next: flow_end` 来结束操作流。

如何将错误处理程序与操作流配合使用

您可以配置操作流，以在流程的指定阶段使用错误处理程序元素发出错误。

错误处理程序元素需要以下两个输入：

- 为失败操作指定的错误消息。
- 操作流输入。



如果流程中的某个操作失败并且操作流包含错误处理程序元素，则会发出错误消息以提醒您操作失败。错误处理程序自己执行操作。以下脚本是可在操作流中使用的错误处理程序的示例。

```
def handler(context, inputs):

    errorMsg = inputs["errorMsg"]
    flowInputs = inputs["flowInputs"]

    print("Flow execution failed with error {0}".format(errorMsg))
    print("Flow inputs were: {0}".format(flowInputs))

    outputs = {
        "errorMsg": errorMsg,
        "flowInputs": flowInputs
    }

    return outputs
```

可以在“操作运行”窗口中查看成功和失败的运行。

| 状态 | 操作 | 运行 ID |
|-----|---------|----------------------------------|
| 已完成 | AWS-ABX | 8a7682b66df80a17016e115a3ef50c06 |
| 失败 | AWS-ABX | 8a7682b66df80a17016e1159db240c02 |
| 已完成 | AWS-ABX | 8a7682b66df80a17016e115612a40bfc |
| 已完成 | AWS-ABX | 8a769ecc6df809c7016e1154c6f10c07 |

在此示例中，**flow-with-handler** 操作流包含一个错误处理程序元素，并且已成功运行。但是，该流程的其中一个操作已失败，从而启动了错误处理程序以发出错误。

如何跟踪操作运行

“操作运行”选项卡显示订阅触发的可扩展性操作及其状态的日志。

可以使用**可扩展性 > 活动 > 操作运行**来查看操作运行的日志。此外，还可以一次按一个或多个属性筛选操作运行的列表。要查看各次操作运行的其他详细信息，请单击运行 ID。

对失败的可扩展性操作运行进行故障排除

如果您的可扩展性操作运行失败，则可以执行故障排除步骤以更正该问题。

当操作运行失败时，您可能会收到错误消息、失败状态和失败日志。如果操作运行失败，则可能是由于部署或代码故障造成的。

| 问题 | 解决方案 |
|------|-------------------------------------------------------------------------------------------------------------------|
| 部署故障 | 这些失败是与云帐户配置、操作部署或其他可能阻止该操作部署的依赖关系相关的问题造成的。确保您使用的项目是在配置的云帐户内定义的，并且已授予运行功能的权限。在重新启动该操作之前，您可以针对该操作的详细信息页面中的特定项目测试操作。 |
| 代码故障 | 这些故障是由于脚本或代码无效所致。使用“操作”运行日志对无效脚本进行故障排除和更正。 |

可扩展性工作流订阅

可以将 vRealize Orchestrator 托管的工作流与 vRealize Automation Cloud Assembly 一起使用来扩展应用程序生命周期。

如何使用 vRealize Orchestrator 工作流订阅修改虚拟机属性

可以使用现有的 vRealize Orchestrator 工作流修改虚拟机属性，以及将虚拟机添加到 Active Directory。

订阅模式定义事件代理服务 (EBS) 消息负载的格式。为了在工作流中接收和使用 EBS 消息负载，您必须定义“inputProperties”工作流输入参数。

前提条件

- 云管理员用户角色
- 现有 vRealize Orchestrator 内部部署工作流。
- 与 vRealize Orchestrator 客户端服务器的成功集成和连接。

步骤

- 1 选择**可扩展性 > 订阅**。
- 2 单击**新建订阅**。

3 使用以下参数创建订阅：

| 参数 | 值 |
|--------|--------------------------------------------------|
| 名称 | 重命名虚拟机 |
| 事件主题 | 选择适用于所需 vRealize Orchestrator 集成的事件主题。例如，计算资源分配。 |
| 阻止/非阻止 | 非阻止 |
| 可运行项 | 选择 vRealize Orchestrator 可运行类型。 |
| 可运行 ID | 选择所需的工作流。例如，“设置虚拟机名称”。 |

4 要保存您的订阅，请单击**创建**。

5 通过创建蓝图或部署现有蓝图，分配并激活订阅。

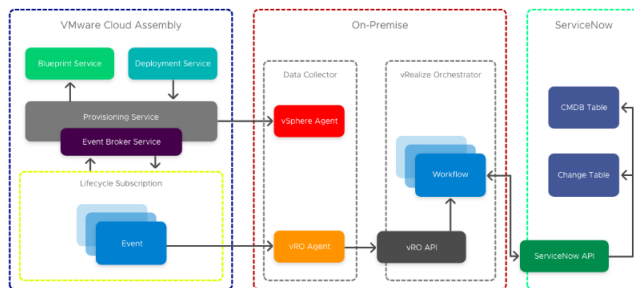
后续步骤

使用以下方法之一验证工作流是否已成功启动：

- 通过导航到**可扩展性 > 活动 > 工作流运行**，验证工作流运行日志。
- 打开 vRealize Orchestrator 客户端，然后通过导航到工作流并验证状态或打开特定日志选项卡来检查工作流状态。

如何使用 vRealize Orchestrator 工作流将适用于 ITSM 的 Cloud Assembly 与 ServiceNow 集成

使用 vRealize Orchestrator 托管的工作流，可以将 vRealize Automation Cloud Assembly 与 ServiceNow 集成以实现 ITSM 合规性



企业用户通常将其云计算管理平台与 IT 服务管理 (IT Service Management, ITSM) 和配置管理数据库 (Configuration Management Database, CMDB) 平台集成以实现合规性。按照此示例，可以使用 vRealize Orchestrator 托管的工作流将 vRealize Automation Cloud Assembly 与适用于 CMDB 和 ITSM 的 ServiceNow 集成。使用 vRealize Orchestrator 集成和工作流时，如果您有多个用于不同环境的实例，则功能标记特别有用。有关功能标记的更多信息，请参见在 [vRealize Automation Cloud Assembly 中使用功能标记](#)。

注 还可以使用可扩展性操作脚本将 ServiceNow 与 vRealize Automation Cloud Assembly 集成。有关使用可扩展性操作脚本集成 ServiceNow 的信息，请参见[如何使用可扩展性操作将 Cloud Assembly 与 ServiceNow 集成](#)。

在此示例中，ServiceNow 集成由三个顶级工作流组成。每个工作流都有自己的订阅，以便您可以单独更新和迭代每个组件。

- 事件订阅入口点 - 基本日志记录，标识请求用户和 vCenter VM（如果适用）。
- 集成工作流 - 分离对象并将输入传递到技术工作流，处理日志记录、属性和输出更新。
- 技术工作流 - 下游系统集成，供 ServiceNow API 用于创建具有负载和其他虚拟机属性的 CMDB CI、CR 和 CAS IaaS API。

前提条件

- 独立或群集 vRealize Orchestrator 环境。
- vRealize Automation Cloud Assembly 中的 vRealize Orchestrator 集成。有关将独立 vRealize Orchestrator 与 vRealize Automation Cloud Assembly 集成的信息，请参见在 [Cloud Assembly 中配置 vRealize Orchestrator 集成](#)。

步骤

- 1 在 vRealize Orchestrator 中创建并保存包含多个工作流中常用配置的配置文件。
- 2 将 CAS API 令牌保存到与步骤 1 中的配置文件相同的位置。

注 CAS API 令牌会过期。

- 3 使用提供的脚本元素在 vRealize Orchestrator 中创建工作流。此脚本将引用并查找 REST 主机。此脚本还将使用令牌可选参数的 REST 操作标准化，该参数添加为额外的授权标头。

```
var configPath = "CS"
var configName = "environmentConfig"
var attributeName = "CASRestHost"

//get REST Host from configuration element
var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath, configName, attributeName)

var ConfigurationElement =
System.getModule("au.com.cs.example").getConfigurationElementByName(configName, configPath);
System.debug("ConfigurationElement:" + ConfigurationElement);
```

```

var casToken = ConfigurationElement.getAttributeWithKey("CASToken")["value"]
if(!casToken){
    throw "no CAS Token";
}
//REST Template
var opName = "casLogin";
var opTemplate = "/iaas/login";
var opMethod = "POST";

// create the REST operation:
var opLogin =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

//cas API Token
var contentObject = {"refreshToken":casToken}
postContent = JSON.stringify(contentObject);

var loginResponse =
System.getModule("au.com.cs.example").executeOp(opLogin,null,postContent,null) ;

try{
    var tokenResponse = JSON.parse(loginResponse)['token']
    System.debug("token: " + tokenResponse);
} catch (ex) {
    throw ex + " No valid token";
}

//REST Template Machine Details
var opName = "machineDetails";
var opTemplate = "/iaas/machines/" + resourceId;
var opMethod = "GET";

var bearer = "Bearer " + tokenResponse;

var opMachine =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

// (Rest Operation, Params, Content, Auth Token)
var vmResponse =
System.getModule("au.com.cs.example").executeOp(opMachine,null,"",bearer) ;

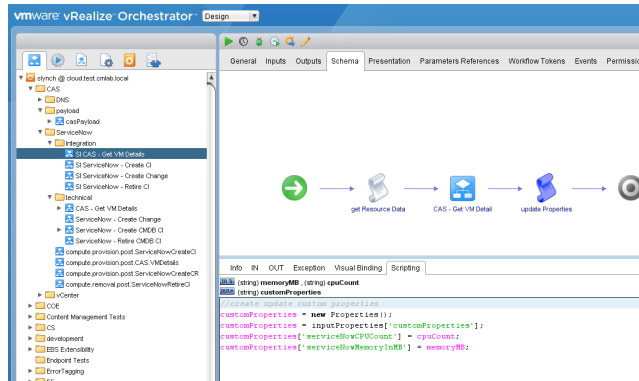
try{
    var vm = JSON.parse(vmResponse);
} catch (ex) {
    throw ex + " failed to parse vm details"
}

System.log("cpuCount: " + vm["customProperties"]["cpuCount"]);
System.log("memoryInMB: " + vm["customProperties"]["memoryInMB"]);

cpuCount = vm["customProperties"]["cpuCount"];
memoryMB = vm["customProperties"]["memoryInMB"];

```

此脚本将输出 `cpuCount` 和 `memoryMB` 发送到父工作流，并更新现有 `customProperties` 属性。创建 CMDB 时，可以在后续工作流中使用这些值。



- 将“创建 ServiceNow CMDB CI”脚本元素添加到工作流中。此元素使用配置项查找 ServiceNow REST 主机，为 `cmdb_ci_vmware_instance` 表创建 REST 操作，基于 `post` 数据的工作流输入创建一系列内容对象，并输出返回的 `sys_id`。

```
var configPath = "CS"
var configName = "environmentConfig"
var attributeName = "serviceNowRestHost"
var tableName = "cmdb_ci_vmware_instance"

//get REST Host from configuration element
var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath,configName,attributeName)

//REST Template
var opName = "serviceNowCreatCI";
var opTemplate = "/api/now/table/" + tableName;
var opMethod = "POST";

// create the REST operation:
var opCI =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

//cmdb_ci_vm_vmware table content to post;
var contentObject = {};
contentObject["name"] = hostname;
contentObject["cpus"] = cpuTotalCount;
contentObject["memory"] = MemoryInMB;
contentObject["correlation_id"] = deploymentId
contentObject["disks_size"] = diskProvisionGB
contentObject["location"] = "Sydney";
contentObject["vcenter_uuid"] = vcUuid;
contentObject["state"] = "On";
contentObject["owned_by"] = owner;

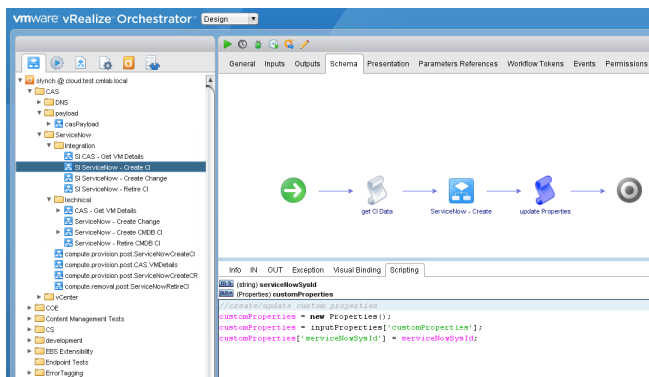
postContent = JSON.stringify(contentObject);
System.log("JSON: " + postContent);
```

```
// (Rest Operation, Params, Content, Auth Token)
var ciResponse =
System.getModule("au.com.cs.example").executeOp(opCI,null,postContent,null) ;

try{
    var cmdbCI = JSON.parse(ciResponse);
} catch (ex) {
    throw ex + " failed to parse ServiceNow CMDB response";
}

serviceNowSysId = cmdbCI['result']['sys_id'];
```

- 5 使用来自子工作流的输出，使用现有 `customProperties` 创建一个属性对象并使用来自 ServiceNow 的值覆盖 `serviceNowSysId` 属性。此唯一 ID 在 CMDB 中用于将实例标记为销毁时注销。



结果

vRealize Automation Cloud Assembly 已成功与 ITSM ServiceNow 集成。有关如何使用工作流在 vRealize Automation Cloud Assembly 中集成 ServiceNow 的更多信息，请参见[使用 vRealize Orchestrator 为 ServiceNow 集成扩展 Cloud Assembly](#)。

了解有关工作流订阅的更多信息

借助 vRealize Orchestrator 与 vRealize Automation Cloud Assembly 的集成，您可以使用工作流来扩展应用程序的生命周期。

vRealize Automation Cloud Assembly 包含 vRealize Orchestrator 集成，您可以使用该集成来导入现有的内部部署工作流并将这些工作流链接到订阅。这些工作流在 vRealize Orchestrator 服务器上维护。您只能使用 vRealize Orchestrator 客户端来创建、修改和删除工作流。

创建 vRealize Orchestrator 工作流的最佳做法

工作流订阅基于特定主题结构定义。要确保订阅可以启动 vRealize Orchestrator 工作流，您必须为这些订阅配置正确的输入参数，以使它们能够使用事件数据。

工作流输入参数

自定义工作流的负载中可以包含所有参数或单个参数（使用所有数据）。

要使用单个参数，请配置一个类型为 `Properties` 且名称为 `inputProperties` 的参数。

工作流输出参数

自定义工作流可以包含回复事件主题类型所需的后续事件的相关输出参数。

如果事件主题需要回复，则工作流输出参数必须与回复结构定义匹配。

如何跟踪工作流运行

工作流运行 选项卡显示订阅触发的工作流及其状态的日志。

可以使用 **可扩展性 > 活动 > 工作流运行** 来查看工作流运行的日志。

对失败的工作流订阅进行故障排除

如果工作流订阅失败，您可以执行故障排除步骤来纠正它。

失败的工作流运行可能会导致工作流订阅无法成功启动或完成。工作流运行失败可能是由几个常见问题导致的。

| 问题 | 原因 | 解决方案 |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| vRealize Orchestrator 工作流订阅未启动，或者未成功完成。 | 您已将工作流订阅配置为在收到事件消息时运行自定义工作流，但该工作流未运行，或者未成功完成。 | <ol style="list-style-type: none"> 1 验证工作流订阅是否已正确保存。 2 验证工作流订阅的条件是否正确配置。 3 验证 vRealize Orchestrator 包含指定的工作流。 4 验证是否已在 vRealize Orchestrator 中正确配置了工作流。 |
| 您的批准请求 vRealize Orchestrator 工作流订阅未运行。 | 已将批准前或批准后工作流订阅配置为运行 vRealize Orchestrator 工作流，但在服务目录中请求满足已定义的条件时，该工作流未运行。 | <p>要成功运行批准工作流订阅，您必须验证所有组件已正确配置。</p> <ol style="list-style-type: none"> 1 验证批准策略是否处于活动状态并且正确应用。 2 验证工作流订阅是否正确配置并保存。 3 检查事件日志以查看与批准相关的消息。 |
| 您的批准请求 vRealize Orchestrator 工作流订阅被拒绝。 | <p>您配置了批准前或批准后工作流订阅，用于运行指定的 vRealize Orchestrator 工作流，但该请求在外部批准级别上被拒绝。</p> <p>一个可能的原因是 vRealize Orchestrator 中存在内部工作流运行错误。例如，该工作流缺失或 vRealize Orchestrator 服务器未在运行。</p> | <ol style="list-style-type: none"> 1 检查日志以查看与批准相关的消息。 2 验证 vRealize Orchestrator 服务器是否正在运行。 3 验证 vRealize Orchestrator 包含指定的工作流。 |

了解有关可扩展性订阅的更多信息

可以将可扩展性操作或 vRealize Orchestrator 托管的工作流与可扩展性订阅一起使用来扩展应用程序生命周期。

当环境中发生触发事件时，将启动订阅并运行指定的工作流或可扩展性操作。您可以在事件日志中查看系统事件，在工作流运行窗口中查看工作流运行，以及在操作运行窗口中查看操作运行。订阅特定于项目，这意味着订阅通过指定的项目链接到蓝图和部署。

可扩展性术语

在 vRealize Automation Cloud Assembly 中使用可扩展性和订阅时，您可能会遇到一些特定于订阅和事件代理服务的术语。

表 6-4. 可扩展性术语

| 术语 | 说明 |
|--------|-------------------------------------------------------------------------------------------------|
| 事件主题 | 描述一组具有相同逻辑意图和相同结构的事件。每个事件是事件主题的一个实例。 可以为某些事件主题分配阻止参数。有关更多信息，请参见 阻止事件主题 。 |
| 事件 | 指示生成者或由其管理的任意实体的状态更改。事件是记录有关事件出现的信息的实体。 |
| 事件代理服务 | 此服务负责将生成者发布的消息发送给已订阅使用者。 |
| 负载 | 事件数据，其中包含与该事件主题相关的所有属性。 |
| 订阅 | 指示订户希望通过订阅事件主题并定义通知的触发条件来接收有关事件的通知。订阅将可扩展性操作或工作流链接到用于使应用程序生命周期各个部分自动化的触发事件。 |
| 订户 | 根据订阅定义，这些用户将收到与发布到事件代理服务的事件相关的通知。订阅者也称为使用者。 |
| 系统管理员 | 拥有使用 vRealize Automation Cloud Assembly 来创建、读取、更新和删除租户工作流订阅与系统工作流订阅的特权的用户。 |
| 工作流订阅 | 指定事件主题和 vRealize Orchestrator 工作流的触发条件。 |
| 操作订阅 | 指定事件主题以及触发可扩展性操作运行的条件。 |
| 工作流 | 集成到 vRealize Automation Cloud Assembly 中的 vRealize Orchestrator 工作流。可以将这些工作流链接到订阅中的事件。 |
| 可扩展性操作 | 简化的代码脚本，可在发生订阅中的触发事件之后运行。可扩展性操作与工作流类似，但更加轻型。可扩展性操作可从 vRealize Automation Cloud Assembly 中进行自定义。 |
| 操作运行 | 可通过 操作运行 选项卡访问。操作运行是为响应触发事件而运行的可扩展性操作的详细日志。 |

阻止事件主题

有些事件主题支持阻止事件。可扩展性订阅的行为取决于主题是否支持这些事件类型以及订阅的配置方式。

vRealize Automation Cloud Assembly 可扩展性订阅可以使用两大类事件主题：非阻止事件主题和阻止事件主题。事件主题类型定义了可扩展性订阅的行为。

非阻止事件主题

非阻止事件主题仅允许创建非阻止订阅。系统以异步方式触发非阻止订阅，您不能依赖订阅的触发顺序。

阻止事件主题

有些事件主题支持阻止。如果订阅标记为阻止，则运行阻止订阅的可运行项之前，任何其他具有匹配条件的订阅都不会收到符合设置条件的所有消息。

阻止订阅按优先级顺序运行。最高优先级值是 0（零）。如果同一个事件主题有多个优先级相同的阻止订阅，这些订阅将基于订阅名称按反向字母顺序运行。处理完所有阻止订阅后，消息会同时发送到所有非阻止订阅。由于阻止订阅以同步方式运行，因此当后续订阅收到通知时，已更改的事件负载包含已更新的事件。

可以使用阻止事件主题管理相互依赖的多个订阅。

例如，您有两个置备工作流订阅，其中第二个订阅取决于第一个订阅的结果。如果第一个订阅在置备期间更改某个属性，则第二个订阅会在文件系统中记录该新属性（例如计算机名称）。如果 **ChangeProperty** 订阅的优先级值设置为 0，则 **RecordProperty** 的优先级值设置为 1，因为第二个订阅使用第一个订阅的结果。置备计算机时，**ChangeProperty** 订阅会开始运行。由于 **RecordProperty** 订阅条件基于置备后条件，因此事件会触发 **RecordProperty** 订阅。但是，由于 **ChangeProperty** 工作流是阻止工作流，因此在完成之前，此工作流不会收到该事件。当计算机名称已更改并且第一个工作流订阅已完成，第二个工作流订阅会开始运行并在文件系统中记录该计算机名称。

恢复可运行项

对于阻止事件主题，可以将恢复可运行项添加到订阅。如果主可运行项失败，则订阅中的恢复可运行项将运行。例如，您可以创建一个工作流订阅，其中主可运行项是在 CMDB 系统（如 **ServiceNow**）中创建记录的工作流。即使该工作流订阅失败，也可能在 CMDB 系统中创建一些记录。在这种情况下，可以使用恢复可运行项清理失败的可运行项在 CMDB 系统中留下的记录。

对于包含相互依赖的多个订阅的用例，可以将 `ebs.recover.continuation` 属性添加到恢复可运行项。使用此属性，可以指示在当前订阅失败时可扩展性服务是否必须继续执行链中的下一个订阅。

随 vRealize Automation Cloud Assembly 提供的事件主题

vRealize Automation Cloud Assembly 包含预定义的事件主题。

事件主题

事件主题是相似事件分组到一起形成的类别。分配给订阅时，事件主题将定义哪个事件会触发订阅。默认情况下，以下事件主题随 vRealize Automation Cloud Assembly 一起提供。所有主题均可用于添加或更新资源的自定义属性或标记。如果 vRealize Orchestrator 工作流或可扩展性操作失败，则相应的任务也会失败。

表 6-5. Cloud Assembly 事件主题

| 事件主题 | 可阻止 | 说明 |
|--------------------------------------|-----|-----------------------------------------------------------|
| Blueprint.configuration | 否 | 当蓝图配置事件（如创建或删除蓝图）发生时发出，可用于向外部系统通知此类事件。 |
| Blueprint.version.configuration | 否 | 当新的蓝图版本控制事件（如创建、发布、取消发布或还原版本）发生时发出。此事件主题在集成第三方版本控制系统时很有用。 |
| Compute allocation | 是 | 在分配 resourcenames 和 hostselections 之前发出的事件。可以在此阶段修改这两个属性。 |
| Compute post provision | 是 | 在已成功置备资源之后发生的事件。 |
| Compute post removal | 是 | 在移除计算资源之后发生的事件。 |
| Compute provision | 是 | 在 Hypervisor 中置备资源之前发生的事件。 注 可以更改分配的 IP 地址。 |
| Compute removal | 是 | 在移除资源之前发生的事件。 |
| Compute reservation | 是 | 在预留时发生的事件。 注 可以更改布置顺序。 |
| Deployment action completed | 是 | 在部署操作完成之后发出。 |
| Deployment action requested | 是 | 在部署操作完成之前发出。 |
| Deployment completed | 是 | 在部署蓝图或目录请求之后发出。 |
| Deployment onboarded | 否 | 在载入新部署时发出。 |
| Deployment requested | 是 | 在部署蓝图或目录请求之前发出。 |
| Deployment resource action completed | 是 | 在部署资源操作之后发出。 |
| Deployment resource action requested | 是 | 在部署资源操作之前发出。 |
| Deployment resource completed | 是 | 在置备部署资源之后发出。 |
| Deployment resource requested | 是 | 在置备部署资源之前发出。 |
| Disk allocation | 是 | 针对磁盘资源预分配发出。 |
| Disk post removal | 是 | 在删除磁盘资源之后发出。 |
| Disk post resize | 是 | 在调整磁盘资源大小之后发出。 |
| EventLog | 是 | 记录相关的事件。 |
| Kubernetes cluster allocation | 是 | 针对 Kubernetes 集群资源预分配发出。 |

表 6-5. Cloud Assembly 事件主题（续）

| 事件主题 | 可阻止 | 说明 |
|-----------------------------------|-----|--------------------------------------------------------------|
| Kubernetes cluster post provision | 是 | 在置备 Kubernetes 集群之后发出。 |
| Kubernetes cluster post removal | 是 | 在删除 Kubernetes 集群之后发出。 |
| Kubernetes cluster provision | 是 | 在置备 Kubernetes 集群之前发出。 |
| Kubernetes cluster removal | 是 | 在启动删除 Kubernetes 集群过程之前发出。 |
| Load balancer post provision | 是 | 在置备负载均衡器之后发出。 |
| Network Configure | 是 | 在计算资源分配过程中配置网络时发生的事件。 注 “配置网络”主题支持多个 IP 地址/网卡。 |
| Project Lifecycle | 否 | 在创建、更新或删除项目时发生的事件。 |

事件结构定义

添加事件主题后，可以查看事件结构定义。此结构定义会定义事件负载或 `inputProperties` 的结构。

可扩展性事件日志

可扩展性事件选项卡显示环境中已发生的所有事件的列表。

可以通过导航至**可扩展性 > 事件**来查看可扩展性事件日志。此外，还可以一次按一个或多个属性筛选事件列表。要查看各个事件的其他详细信息，请单击事件 ID。

部署
蓝图
基础架构
可扩展性
商城

引导式设置

事件
3442 项

筛选...

| ID | 时间戳 | 事件主题 | 用户名 | 目标 ID | 描述 |
|--------------------------------------|-----------------|-------------------------------|---------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| d4b40f20-9792-4c48-aec0-0a16ce7eaabe | 20/01/14 下午6:14 | Blueprint configuration | pmartini@vmware.com | | |
| 1d86cdf8-9555-d312-1836-00d8b2a262e6 | 20/01/14 下午4:46 | Project Lifecycle Event Topic | project-service | 1d0b482d-8396-46a1-a493-54ed2d299ec8 | update |
| d4cf544c-d263-c30c-3b58-23b31be25f6a | 20/01/14 下午4:31 | EventLog | vro-gateway | 77ef8b8b-3180-44d0-a69a-503f27edc42c | Workflow run [77ef8b8b-3180-44d0-a69a-503f27edc42c] status changed to [FAILED]. workflowId:[f246b7b5-fe89-4da5-a640-361fc6874069] eventId:[33c01131-c9f9-3255-a686-a67671f44bcb] |
| 16e0c6ed-40e4-4cb8-9be6-f7543b9c6a7c | 20/01/14 下午3:55 | Deployment completed | pmartini@vmware.com | | |
| b754bc46-f255-4869-b789-252924b32c4d | 20/01/14 下午3:55 | Deployment requested | pmartini@vmware.com | | |

创建可扩展性订阅

通过将 vRealize Orchestrator 集成或可扩展性操作与 vRealize Automation Cloud Assembly 一起使用，可以创建订阅来扩展应用程序。

借助可扩展性订阅，您可以通过在发生特定生命周期事件时触发 workflow 或操作来扩展应用程序。还可以对订阅应用筛选器来为指定的事件设置布尔条件。例如，仅当布尔表达式为“true”时，事件和 workflow/操作才会触发。如果您要控制何时触发事件和操作，这很有用。

提示 在“按主题筛选事件”文本框中，按 **Alt+空格键**（对于 Windows）或 **Option+空格键**（对于 Mac）可显示筛选器选项。

前提条件

- 云管理员用户角色
- 如果使用 vRealize Orchestrator 工作流：
 - 嵌入式 vRealize Orchestrator 客户端的库或任何集成外部 vRealize Orchestrator 实例的库。
- 如果使用可扩展性操作：
 - 现有可扩展性操作脚本。有关详细信息，请参见[如何创建可扩展性操作](#)。

步骤

- 1 选择**可扩展性 > 订阅**。
- 2 单击**新建集成**。
- 3 输入订阅的详细信息。
- 4 选择**事件主题**。
- 5 （可选）设置事件主题的条件。
- 6 （可选）如果适用，请为事件主题配置阻止行为。
- 7 单击**可运行项**，然后从下拉菜单中选择 **vRO 工作流**或 **ABX 操作**。
- 8 选择要在订阅中运行的工作流或可扩展性操作。
- 9 （可选）要定义可扩展性订阅的项目范围，请禁用**任何项目**，然后单击**添加项目**。
- 10 单击**创建**以保存订阅。

结果

已创建订阅。发生按所选事件主题分类的事件时，将启动链接的 vRealize Orchestrator 工作流或可扩展性操作，并通知所有订阅者。

后续步骤

创建订阅之后，您可以创建或部署蓝图以链接并使用订阅。此外，您可以在 vRealize Automation Cloud Assembly 中的**可扩展性**选项卡中验证工作流运行的状态。对于包含 vRealize Orchestrator 工作流的订阅，还可以从 vRealize Orchestrator 客户端监控运行和工作流状态。

对可扩展性订阅进行故障排除

对可扩展性订阅失败进行故障排除。

当您的订阅失败时，通常是因为您的工作流或可扩展性操作脚本出现错误。

查看主题参数和负载

可以使用转储订阅主题参数脚本在任何给定的事件阶段查看虚拟机的特定参数和负载。

此脚本主要用于调试和检查可用于您的 vRealize Orchestrator 工作流的输入。要查看虚拟机的所有参数，请对您的工作流使用以下脚本：

```
function dumpProperties(props, lvl) {
    var keys = props.keys;
    var prefix = ""
    for (var i=0; i<lvl; i++){
        prefix = prefix + " ";
    }
    for (k in keys){
        var key = keys[k];
        var value = props.get(keys[k])
        if ("Properties" == System.getObjectType(value)){
            System.log(Prefix + key + "[")
            dumpProperties(value, (lvl+2));
            System.log(prefix+ "]")
        } else{
            System.log( prefix + key + ":" + value)
        }
    }
}

dumpProperties(inputProperties, 0)

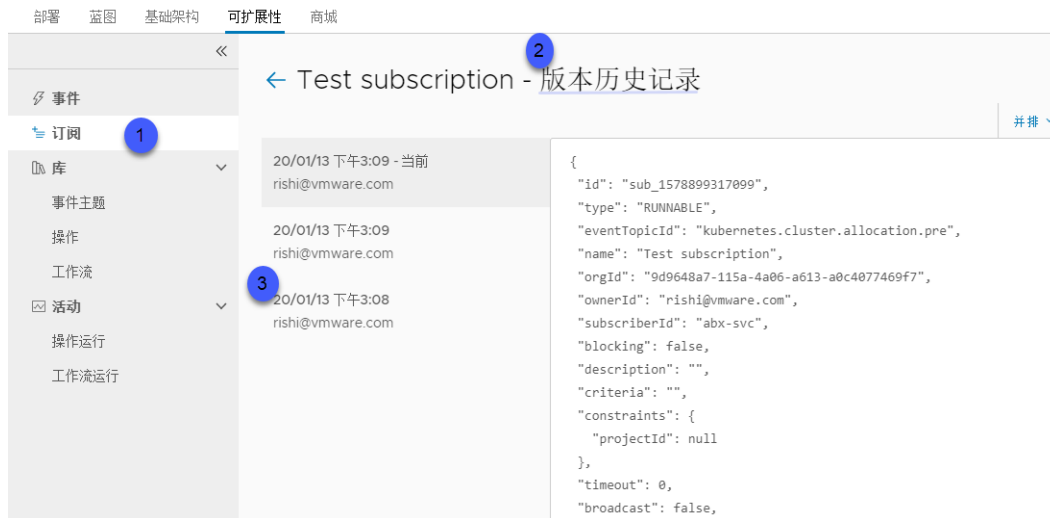
customProps = inputProperties.get("customProperties")
```

订阅版本历史记录

如果您的订阅失败，您可以查看版本历史记录。

查看订阅版本历史记录

“版本历史记录”选项卡可显示您的订阅的更改历史记录，以及更改的用户和日期。如果您的订阅失败或运行不正确，则版本历史记录有助于确定原因。



1

从订阅选项卡打开您的订阅。

2

要查看版本历史记录，请单击版本历史记录。

3

您可以单击每个更改条目以查看与更改关联的相应订阅代码。

管理 vRealize Automation Cloud Assembly 部署

7

作为 vRealize Automation Cloud Assembly 蓝图开发人员，您可以使用“部署”选项卡来管理您的部署。您可以对失败的置备过程进行故障排除，进行更改，并销毁未使用的部署。

部署是蓝图的已置备实例。“部署”选项卡显示成功的部署和失败的部署。您可以使用该页面管理成功的部署，或开始对任何失败的请求进行故障排除。

使用部署卡

可以使用卡列表来查找和管理部署。您可以筛选或搜索特定部署，然后对这些部署运行操作。

- 1 根据属性筛选请求。
- 2 根据关键字或请求者搜索部署。
- 3 对列表进行排序，以便按时间或名称排序。
- 4 对部署运行部署级别操作，包括删除未使用的部署以回收资源。

还可以查看部署成本、到期日期和状态。



本章讨论了以下主题：

- 如何监控 vRealize Automation Cloud Assembly 中的活动部署
- vRealize Automation Cloud Assembly 部署失败时可以执行哪些操作
- 如何管理已完成 vRealize Automation Cloud Assembly 部署的生命周期
- 可以对 vRealize Automation Cloud Assembly 部署运行哪些操作

如何监控 vRealize Automation Cloud Assembly 中的活动部署

在部署 vRealize Automation Cloud Assembly 蓝图后，您可以监控请求以确保资源已置备且正在运行。从部署卡开始，您可以验证资源的置备。接下来，您可以检查部署详细信息。

步骤

- 1 如果需要，单击**部署**，并使用筛选和搜索找到您的进行中的部署卡。
- 2 查看卡状态。

如果部署正在进行中，则进程栏会指示剩余任务数。如果部署成功完成，则卡会显示有关部署的基本详



- 3 要确定资源的部署位置，请单击部署名称，然后在“拓扑”页面上查看详细信息。

您可能需要主要组件的 IP 地址。单击每个组件时，请注意提供的特定于组件的信息。在此示例中，将突出显示 IP 地址。



外部链接的可用性取决于云提供商。如果外部链接可用，您必须具有该提供商的凭据才能访问相应的组件。

后续步骤

- 您可以对部署进行更改。请参见如何管理已完成 vRealize Automation Cloud Assembly 部署的生命周期。
- 如果部署失败，请参见 vRealize Automation Cloud Assembly 部署失败时可以执行哪些操作。

vRealize Automation Cloud Assembly 部署失败时可以执行哪些操作

您的部署请求可能由于多种原因而失败。失败可能由网络流量、目标云提供商缺少资源或部署规范存在缺陷导致。或者，部署已成功，但部署无法正常工作。可以使用 vRealize Automation Cloud Assembly 检查部署，查看任何错误消息，并确定问题由环境、请求的工作负载规范还是其他因素导致。

您可以使用此工作流程来开始调查。该过程可能表明失败是由于暂时的环境问题造成的。确认情况改善后重新部署请求可以解决此类问题。在其他情况下，调查可能要求您详细检查其他方面。

作为项目成员，您可以在 vRealize Automation Cloud Assembly 中查看请求详细信息。

步骤

- 要确定某个请求是否失败，单击**部署**选项卡，然后找到部署卡。



该卡指明了失败的部署。

- 查看错误消息。
- 有关详细信息，请单击部署名称以了解部署详细信息。

- 在部署详细信息页面中，单击**历史记录**选项卡。



- 查看事件树以确定置备过程失败的位置。在修改部署但更改失败时，该树很有用。该树还显示了运行部署操作的时间。您可以使用该树对失败的更改进行故障排除。
- 详细信息**提供错误消息的更详细版本。
- 如果请求的项是 vRealize Automation Cloud Assembly 蓝图，则消息右侧的链接将打开 vRealize Automation Cloud Assembly，以便您可以查看**请求详细信息**。

3 请求详细信息提供故障组件的置备 workflow，以便您可以研究问题。



- a 查看错误消息。
- b 打开**开发模式**可在简单置备 workflow 和更详细的流程图之间切换。
- c 单击该卡以查看部署脚本。

4 解决错误并重新部署蓝图。

这些错误可能存在于蓝图构造中，也可能与基础架构的配置方式有关。

后续步骤

解决错误并部署蓝图后，您可以在“请求详细信息”中看到类似以下示例的信息。要查看请求详细信息，请选择**基础架构 > 活动 > 请求**。



如何管理已完成 vRealize Automation Cloud Assembly 部署的生命周期

置备并运行部署后，您可以运行多个操作来管理部署。生命周期管理可以包括打开电源或关闭电源、调整部署大小以及删除部署。您还可以对各个组件运行各种操作以对其进行管理。

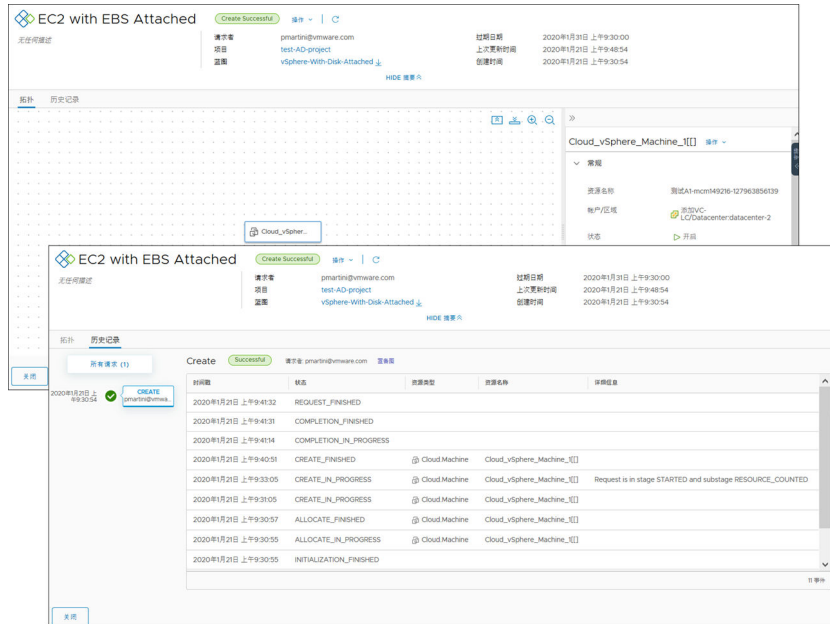
步骤

- 1 单击**部署**，并找到您的部署。
- 2 要访问部署详细信息，请单击部署名称。

可以使用“拓扑”选项卡来可视化部署结构和资源。

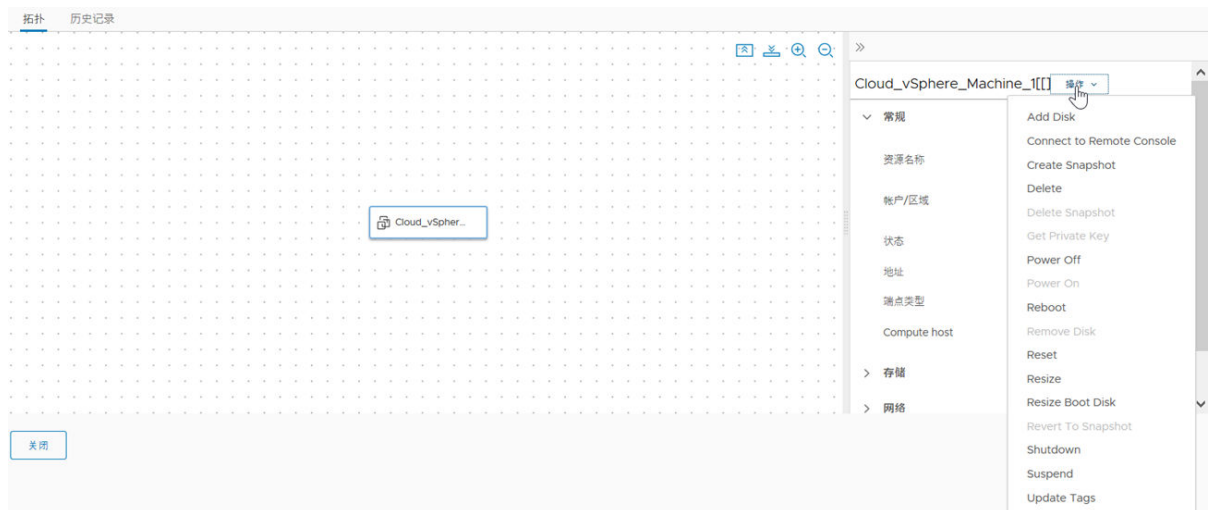
“历史记录”选项卡包含所有置备事件，以及与您部署请求项后运行的操作相关的任何事件。如果置备过程存在任何问题，“历史记录”选项卡中的事件可帮助您对故障进行故障排除。

“成本”选项卡提供部分组件自部署以来的当前成本。



- 3 如果您确定部署的当前配置成本太高，并且希望调整组件大小，则可以在拓扑页面上选择该组件，然后在组件页面上选择**操作 > 调整大小**。

可用操作取决于组件、云帐户和您的权限。



- 4 作为开发生命周期的一部分，不再需要您的其中一个部署。要移除部署并回收资源，选择**操作 > 删除**。

可用操作取决于部署的状态。



后续步骤

要了解有关可能操作的详细信息，请参见[可以对 vRealize Automation Cloud Assembly 部署运行哪些操作](#)。

可以对 vRealize Automation Cloud Assembly 部署运行哪些操作

部署蓝图之后，可以在 vRealize Automation Cloud Assembly 中运行操作来管理资源。可用操作取决于资源类型，以及这些操作在特定云帐户或集成平台上是否受支持。

可用操作还取决于管理员授权您运行的操作。

作为管理员或项目管理员，您可以在 vRealize Automation Service Broker 中设置“实施后操作”策略。请参见[如何授权使用者使用 Service Broker 实施后操作策略](#)

表 7-1. 可能操作的列表

| 操作 | 应用到以下资源类型 | 适用于以下云帐户或集成 | 说明 |
|------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| 添加磁盘 | 计算机 | <ul style="list-style-type: none">Amazon Web ServicesGoogle Cloud PlatformMicrosoft AzureVMware vSphere | 将其他磁盘添加到现有虚拟机。 |
| 更改租约 | 部署 | <ul style="list-style-type: none">Amazon Web ServicesMicrosoft AzureVMware vSphere | <p>更改租约过期日期和时间。</p> <p>当租约过期时，将销毁部署并回收资源。</p> <p>租约策略在 vRealize Automation Service Broker 中设置。</p> |

表 7-1. 可能操作的列表（续）

| 操作 | 应用到以下资源类型 | 适用于以下云帐户或集成 | 说明 |
|----------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 连接到远程控制台 | 计算机 | <ul style="list-style-type: none"> VMware vSphere | <p>在所选计算机上打开远程会话。</p> <p>检查以下成功连接的要求。</p> <ul style="list-style-type: none"> 作为部署使用者，确认置备的计算机已打开电源。 |
| 创建快照 | 计算机 | <ul style="list-style-type: none"> Google Cloud Platform VMware vSphere | <p>创建虚拟机的快照。</p> <p>如果 vSphere 中仅允许您拥有两个快照且您已拥有它们，则该命令只有在删除一个快照之后才可用。</p> |
| 删除 | 部署 | <ul style="list-style-type: none"> Amazon Web Services Google Cloud Platform Microsoft Azure VMware vSphere | <p>销毁部署。</p> <p>将删除并回收所有资源。</p> <p>如果删除失败，则可以再次对部署运行删除操作。在第二次尝试期间，您可以选择忽略删除失败。如果选择此选项，则会删除部署，但可能无法回收资源。您应检查置备了部署的系统，以确保移除所有资源。如果没有，则必须手动删除这些系统上的剩余资源。</p> |
| | 计算机和负载均衡器 | <ul style="list-style-type: none"> Amazon Web Services Microsoft Azure VMware vSphere | 从部署中删除计算机或负载均衡器。此操作可能会导致部署不可用。 |
| 删除快照 | 计算机 | <ul style="list-style-type: none"> VMware vSphere Google Cloud Platform | 删除虚拟机的快照。 |
| 编辑标记 | 部署 | <ul style="list-style-type: none"> Amazon Web Services Microsoft Azure VMware vSphere | 添加或修改应用于单个部署资源的资源标记。 |
| 关闭电源 | 部署 | <ul style="list-style-type: none"> Amazon Web Services Microsoft Azure VMware vSphere | 关闭部署，而不关闭客户机操作系统。 |
| | 计算机 | <ul style="list-style-type: none"> Amazon Web Services Google Cloud Platform Microsoft Azure VMware vSphere | 关闭计算机电源，而不关闭客户机操作系统。 |
| 打开电源 | 部署 | <ul style="list-style-type: none"> Amazon Web Services Microsoft Azure VMware vSphere | 打开部署。如果资源已挂起，则从资源挂起的时间点恢复正常操作。 |
| | 计算机 | <ul style="list-style-type: none"> Amazon Web Services Google Cloud Platform Microsoft Azure VMware vSphere | 打开计算机电源。如果计算机已挂起，则从计算机挂起的时间点恢复正常操作。 |
| 重新引导 | 计算机 | <ul style="list-style-type: none"> Amazon Web Services VMware vSphere | <p>在虚拟机上重新引导客户机操作系统。</p> <p>对于 vSphere 计算机，要使用此操作，必须在计算机上安装 VMware Tools。</p> |

表 7-1. 可能操作的列表（续）

| 操作 | 应用到以下资源类型 | 适用于以下云帐户或集成 | 说明 |
|--------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| 重新配置 | 负载均衡器 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ Google Cloud Platform ■ VMware vSphere | 更改负载均衡器协议、端口、运行状况配置和成员池设置。 |
| 移除磁盘 | 计算机 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere | 从现有虚拟机中移除磁盘。 |
| 重置 | 计算机 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ VMware vSphere | 强制虚拟机重新启动，而不关闭客户机操作系统。 |
| 调整大小 | 计算机 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ Google Cloud Platform ■ VMware vSphere | 增加或减少虚拟机的 CPU 和内存。 |
| 调整引导磁盘大小 | 计算机 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere | 增加或减少引导磁盘介质的大小。 |
| 调整磁盘大小 | 存储磁盘 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform | 增加存储磁盘的容量。 |
| 重新启动 | 计算机 | <ul style="list-style-type: none"> ■ Microsoft Azure | 先关闭再重新启动正在运行的计算机。 |
| 恢复到快照 | 计算机 | <ul style="list-style-type: none"> ■ Google Cloud Platform ■ VMware vSphere | 恢复到该计算机的上一个快照。 要使用此操作，必须存在现有快照。 |
| 运行 Puppet 任务 | 受管资源 | <ul style="list-style-type: none"> ■ Puppet Enterprise | 在部署中的计算机上运行所选任务。 任务在 Puppet 实例中定义。您必须能够确定任务并提供输入参数。 |
| 关机 | 计算机 | <ul style="list-style-type: none"> ■ VMware vSphere | 关闭客户机操作系统并关闭计算机电源。要使用此操作，必须在计算机上安装 VMware Tools。 |
| 挂起 | 计算机 | <ul style="list-style-type: none"> ■ Microsoft Azure ■ VMware vSphere | 暂停计算机，使其无法使用，并且不使用除所用存储之外的任何其他系统资源。 |
| 更新 | 部署 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere | 根据输入参数更改部署。 |
| 更新标记 | 计算机和磁盘 | <ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere | 添加、修改或删除应用于单个资源的标记。 |