

# 生命周期可扩展性

2020 年 12 月 28 日

vRealize Automation 7.6

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](http://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](http://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](http://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](http://www.vmware.com/cn)

# 目录

<b>1 生命周期可扩展性</b>	<b>4</b>
更新可扩展性	4
计算机可扩展性概览	4
计算机生命周期可扩展性	5
选择生命周期可扩展性方案	6
使用 vRealize Orchestrator 扩展计算机生命周期	7
使用 vRealize Orchestrator 检查表扩展计算机生命周期	7
配置用于实现计算机可扩展性的 vRealize Automation 插件	7
使用 vRealize Orchestrator 自定义 IaaS 工作流	11
配置工作流订阅以扩展 vRealize Automation	12
随 vRealize Automation 提供的事件主题	12
工作流订阅和事件代理术语	13
可阻止和可回复事件主题	14
为工作流订阅创建 vRealize Orchestrator 工作流的最佳做法	15
工作流订阅设置	15
处理置备和生命周期工作流订阅	19
处理批准工作流订阅	38
工作流订阅故障排除	43
使用 vRealize Automation Designer 扩展计算机生命周期	45
使用 vRealize Automation Designer 检查表扩展计算机生命周期	45
安装和配置 vRealize Automation Designer	46
使用 vRealize Automation Designer 自定义 IaaS 工作流	50
工作流和分布式管理	64
通过使用技能来关联工作流与 DEM Worker	64
移除技能与 DEM Worker 之间的关联	65
移除技能与工作流之间的关联	65
移除技能	66
CloudUtil 命令参考	66
DEM 命令	66
File 命令	68
Operation 命令	71
Skill 命令	72
Workflow 命令	73
导入命令	75
vRealize Automation 工作流活动参考	77
DynamicOps.Repository.Activities	77
DynamicOps.Cdk.Activities	80

# 生命周期可扩展性

# 1

将 vRealize Orchestrator 与 vRealize Automation 配合使用，您可以扩展管理 IaaS 计算机生命周期的方式。

扩展 vRealize Automation 需要您使用提供的 vRealize Orchestrator 工作流并创建自定义工作流。

本章讨论了以下主题：

- [更新可扩展性](#)
- [计算机可扩展性概览](#)
- [使用 vRealize Orchestrator 扩展计算机生命周期](#)
- [配置工作流订阅以扩展 vRealize Automation](#)
- [使用 vRealize Automation Designer 扩展计算机生命周期](#)
- [工作流和分布式管理](#)
- [CloudUtil 命令参考](#)
- [vRealize Automation 工作流活动参考](#)

## 更新可扩展性

本《生命周期可扩展性》随每一版本的产品更新或在必要时进行更新。

下表提供了《《生命周期可扩展性》》的更新历史记录。

修订版本	描述
2018 年 10 月 4 日	次要更新。
2018 年 9 月 20 日	初始版本。

## 计算机可扩展性概览

置备或取消配置新计算机（尤其是针对任务关键型系统）通常需要与许多不同管理系统进行交互，包括 DNS 服务器、负载均衡器、CMDB、IP 地址管理和其他系统。

## 计算机生命周期可扩展性

利用 IaaS 状态更改工作流（称为工作流存根），可以在各个预定的 IaaS 生命周期阶段注入自定义逻辑。工作流存根可用来调用 vRealize Orchestrator，以便与外部管理系统进行双向集成。

通过创建状态更改工作流，您可以在 IaaS 主工作流进入特定状态之前触发某工作流的执行。例如，您可以在计算机生命周期的不同阶段创建自定义工作流，与外部数据库进行集成，并记录信息。

- 创建在主工作流进入 **MachineProvisioned** 状态之前运行的自定义工作流，以记录诸如计算机所有者和审批者等信息。
- 创建在计算机进入 **MachineDisposing** 状态之前运行的自定义工作流，以记录该计算机被销毁的时间和相关数据（例如，该计算机在上次数据收集时的资源利用率和上次登录时间等）。

主工作流示意图显示了主工作流的主要状态，并以黄色突出显示了可以使用 IaaS 工作流存根进行自定义的状态。**可自定义状态更改工作流**表列出了可用的工作流存根、这些工作流存根在主工作流状态中的相应位置以及可在每个状态下用来延长计算机生命周期的自定义逻辑的示例。

图 1-1. 置备计算机的主工作流状态

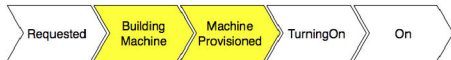


图 1-2. 有关计算机导入的主工作流状态

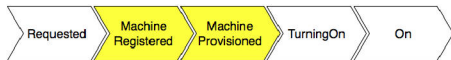


图 1-3. 有关计算机租约过期的主工作流状态



图 1-4. 有关计算机处置的主工作流状态

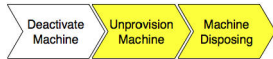


表 1-1. 可自定义状态更改工作流

主工作流状态	可自定义工作流名称	可扩展性示例
BuildingMachine	WFStubBuildingMachine	准备要在 Hypervisor 上创建的计算机。创建配置管理数据库 (CMDB) 记录，调用外部系统以便为计算机分配 IP 地址，然后在计算机处置期间使用 WFStubMachineDisposing 将该 IP 地址返回到池中。
RegisterMachine	WFStubMachineRegistered	将已导入的计算机添加到应用程序置备工具，以接收更新并进行合规性检查。
MachineProvisioned	WFStubMachineProvisioned	该计算机已存在于 Hypervisor 上，并且其他自定义（例如客户机代理自定义）在此时已完成。通过使用此工作流存根，可以使用 DHCP IP 地址和存储信息来更新配置管理数据库 (CMDB) 记录。通常可以使用 WFStubUnprovisionMachine 来撤消通过 WFStubMachineProvisioned 进行的自定义设置。

表 1-1. 可自定义状态更改工作流（续）

主工作流状态	可自定义工作流名称	可扩展性示例
Expired	WFStubMachineExpired	将已过期的计算机移动到低成本存储，以便降低存档成本并更新 CMDB 记录和计费系统以反映存储更改和成本更改。
UnprovisionMachine	WFStubUnprovisionMachine	将计算机从 Active Directory 帐户中移除。通常可以使用 WFStubUnprovisionMachine 来撤消通过 WFStubMachineProvisioned 进行的自定义设置。
Disposing	WFStubMachineDisposing	将 IP 地址返回到池中。

## 选择生命周期可扩展性方案

可以使用 vRealize Orchestrator 或 vRealize Automation Designer 来扩展计算机生命周期。

扩展计算机生命周期有两种方法，一种是使用 vRealize Automation Designer 调用 vRealize Orchestrator，另一种是直接使用 vRealize Orchestrator。这两种方法都可以将自定义逻辑添加到 IaaS 计算机生命周期的预定阶段，具体方法是先创建自定义 vRealize Orchestrator 工作流，然后将自定义工作流插入到状态更改工作流存根。但是，如果直接使用 vRealize Orchestrator，则只能将自定义状态更改逻辑限定到特定蓝图。通过使用 vRealize Automation Designer，您可以限定仅特定 Distributed Execution Manager (DEM) 可以运行工作流。

**注** 工作流存根被事件代理工作流订阅所取代。目前仍会提供工作流存根，它们仍受支持且可以使用，但预计会在将来的 vRealize Automation 版本中将其移除。为确保将来的产品兼容性，您应该使用工作流订阅，以基于状态更改运行自定义工作流。请参见[配置工作流订阅以扩展 vRealize Automation](#)。

表 1-2. 选择生命周期可扩展性方案

方案	步骤
将自定义逻辑添加到 IaaS 计算机生命周期的预定阶段，并将该自定义逻辑应用到特定蓝图。	使用 <a href="#">vRealize Orchestrator 检查表</a> 扩展计算机生命周期
将自定义逻辑添加到 IaaS 计算机生命周期的预定阶段，并以全局方式将该自定义逻辑应用到所有蓝图。	使用 <a href="#">vRealize Automation Designer 检查表</a> 扩展计算机生命周期
通过在 vRealize Automation Designer 中使用技能，限定仅特定 Distributed Execution Manager 可以运行工作流。技能类似于标记，可以同时应用到工作流和 DEM Worker 实例。 例如，您可能想要将云置备工作流限定给在可通过所需网络访问 Amazon URL 的主机上运行的特定 DEM。	通过使用技能来关联工作流与 <a href="#">DEM Worker</a>

## 使用 vRealize Orchestrator 扩展计算机生命周期

可以将自定义逻辑注入到 IaaS 计算机生命周期的预定阶段，方法是创建自定义 vRealize Orchestrator workflows，然后使用 vRealize Orchestrator 将自定义 workflow 插入到根据特定蓝图构建的计算机生命周期。

## 使用 vRealize Orchestrator 检查表扩展计算机生命周期

“使用 vRealize Orchestrator 检查表扩展计算机生命周期”从较高层面概述了安装和配置 vRealize Orchestrator 以自定义 IaaS 计算机生命周期所需的步骤。

表 1-3. 使用 vRealize Orchestrator 检查表扩展计算机生命周期

任务	详细信息
<input type="checkbox"/> 为 vRealize Orchestrator 配置 vRealize Automation 主机。	<a href="#">添加 vRealize Automation 主机</a>
<input type="checkbox"/> 为 vRealize Orchestrator 配置 IaaS 主机。	<a href="#">添加 IaaS 主机</a>
<input type="checkbox"/> 安装 vRealize Orchestrator 自定义以用于扩展 IaaS 计算机生命周期。	<a href="#">安装 vRealize Orchestrator 自定义</a>
<input type="checkbox"/> 为 vRealize Orchestrator 实例创建 vRealize Automation 端点。	<a href="#">创建 vRealize Orchestrator 端点</a>
<input type="checkbox"/> 使用 vRealize Automation 插件库的“可扩展性”子目录中提供的工作流模板创建要在计算机生命周期期间运行的自定义 vRealize Orchestrator  workflow。您可以运行同一个蓝图中具有相同状态的多个 workflow，只要将它们嵌套在一个包装程序 workflow 之下即可。	有关使用 vRealize Orchestrator 来开发 workflow 的信息，请参见 vRealize Orchestrator 文档。有关针对 vRealize Automation 集成的 vRealize Orchestrator 开发培训，请参见 VMware Education 提供的培训课程和 VMware Learning 提供的教材。
<input type="checkbox"/> 运行所提供的工作流，以将自定义 workflow 插入到 IaaS  workflow 存根中，以及将蓝图配置为调用该 IaaS  workflow 存根。	<a href="#">为蓝图及其虚拟机分配状态更改 workflow</a>
<b>注</b> workflow 存根被事件代理 workflow 订阅所取代。目前仍会提供 workflow 存根，它们仍受支持且可以使用，但预计会在将来的 vRealize Automation 版本中将其移除。为确保将来的产品兼容性，您应该使用 workflow 订阅，以基于状态更改运行自定义 workflow。请参见 <a href="#">配置 workflow 订阅以扩展 vRealize Automation</a> 。	

## 配置用于实现计算机可扩展性的 vRealize Automation 插件

应配置 vRealize Automation 主机和 IaaS 主机，安装用于实现计算机可扩展性的自定义，并为 vRealize Orchestrator 实例创建 vRealize Automation 端点。

### 添加 vRealize Automation 主机

可以运行 workflow 以添加 vRealize Automation 主机并配置主机连接参数。

#### 步骤

- 1 从 Orchestrator 客户端的下拉菜单中，选择 **运行或设计**。

- 2 单击**工作流**视图。
- 3 展开**库 > vRealize Automation > 配置**。
- 4 右键单击**添加 vRA 主机**工作流，然后选择**启动工作流**。
- 5 在**主机名称**文本框中输入主机的唯一名称。
- 6 在**主机 URL**文本框中输入主机的 URL 地址。  
例如：*https://hostname*。
- 7 （必选）在**租户**文本框中输入租户的名称。  
要使租户使用该插件的完整功能，请为每个租户创建专用的 vRealize Automation 主机。
- 8 选择是否自动安装 SSL 证书而无需用户确认。
- 9 （可选）要配置 vRealize Orchestrator 等待来自 vRealize Automation 的连接或响应所用的时间长度，请在**连接超时 (秒)**和**操作超时 (秒)**文本框中输入超时间隔。
- 10 从**会话模式**下拉菜单中选择与主机的连接类型。

选项	操作
共享会话	在 <b>身份验证用户名</b> 文本框和 <b>身份验证密码</b> 文本框中输入 vRealize Automation 用户的凭据。
每用户会话	<p>使用当前已登录的用户的凭据进行连接。您必须使用 vRealize Automation 系统管理员的凭据登录到 Orchestrator 客户端。</p> <p>要将此选项与外部 vRealize Orchestrator 服务器结合使用，您必须在 vRealize Automation 组件注册表中注册 Orchestrator 服务器。</p> <p><b>注</b> 要在组件注册表注册外部 vRealize Orchestrator 服务器，必须将 Orchestrator 配置为使用 vRealize Automation 作为身份验证提供程序。有关详细信息，请参见《安装和配置 VMware vRealize Orchestrator》。</p>

- 11 单击**提交**。

#### 后续步骤

添加 vRealize Automation 基础架构管理主机。

### 添加 IaaS 主机

可以运行工作流以添加 vRealize Automation 主机的 IaaS 主机并配置连接参数。

#### 步骤

- 1 从 Orchestrator 客户端的下拉菜单中，选择**运行或设计**。
- 2 单击**工作流**视图。
- 3 展开**库 > vRealize Automation > 基础架构管理 > 配置**。
- 4 右键单击**添加 IaaS 主机**，然后选择**启动工作流**。
- 5 从**VCAC 主机**下拉菜单中选择要为其配置 IaaS 主机的 vRealize Automation 主机。



- 6 在**主机名称**文本框中输入主机的唯一名称。
- 7 输入安装了 Model Manager 的计算机的 URL。  
例如: `https://model_manager_machine.com`。
- 8 要安装 SSL 证书, 请选择**是**。
- 9 要使用代理访问 Model Manager 计算机, 请选择**是**。  
如果选择此选项, 则必须在下一页上提供代理主机和代理端口。
- 10 单击**下一步**。
- 11 如果要配置显式代理, 请提供代理主机 URL 和端口。
- 12 单击**下一步**。
- 13 要配置您自己的超时值, 请单击**否**。
- 14 (可选) 要配置 vRealize Orchestrator 等待来自 vRealize Automation 的连接或响应所用的时间长度, 请在**连接超时 (秒)**和**操作超时 (秒)**文本框中输入超时间隔。
- 15 单击**下一步**。
- 16 选择主机的身份验证类型。

选项	描述
<b>SSO</b>	选择此选项可使用 vCenter Single Sign-On。
<b>NTLM</b>	选择此选项可仅在您的 Active Directory 基础架构依赖于 NT LAN 管理器 (NTLM) 身份验证时启用基于 NTLM 协议的身份验证。 如果选择此选项, 则必须指定其他 NTLM 凭据和身份验证选项。

- 17 如果已选择 NTLM, 请单击**下一步**, 然后输入 Workstation 计算机的名称和 NetBIOS 域名。
- 18 单击**提交**。

## 安装 vRealize Orchestrator 自定义

可以通过运行工作流来安装自定义的状态更改工作流存根和 Orchestrator 菜单操作工作流。

**注** 工作流存根被事件代理工作流订阅所取代。目前仍会提供工作流存根, 它们仍受支持且可以使用, 但预计会在将来的 vRealize Automation 版本中将其移除。为确保将来的产品兼容性, 您应该使用工作流订阅, 以基于状态更改运行自定义工作流。请参见[配置工作流订阅以扩展 vRealize Automation](#)。

### 步骤

- 1 从 Orchestrator 客户端的下拉菜单中, 选择**运行或设计**。
- 2 单击**工作流**视图。
- 3 选择**库 > vCloud Automation Center > 基础架构管理 > 可扩展性 > 安装**。
- 4 右键单击**安装 vCO 自定义工作流**, 然后选择**启动工作流**。

- 5 选择 IaaS 主机。
- 6 单击下一步。
- 7 选择要安装的一个或多个状态更改工作流存根，以选择要将自定义逻辑添加到的生命周期阶段。
- 8 单击提交。

## 创建 vRealize Orchestrator 端点

您可以创建 vRealize Orchestrator 端点以连接到 vRealize Orchestrator 服务器。

您可以配置多个端点并让它们连接到不同的 vRealize Orchestrator 服务器，但必须配置每个端点的优先级。

执行 vRealize Orchestrator 工作流时，vRealize Automation 将先尝试使用优先级最高的 vRealize Orchestrator 端点。如果该端点不可访问，则继续尝试下一个优先级最高的端点，直到 vRealize Orchestrator 服务器可用于运行该工作流为止。

### 前提条件

- 以 IaaS 管理员身份登录到 vRealize Automation。

### 步骤

- 1 选择基础架构 > 端点 > 端点。
- 2 选择新建 > Orchestration > vRealize Orchestrator。
- 3 输入名称和可选描述。
- 4 输入包含 vRealize Orchestrator 服务器的完全限定名称或 IP 地址以及 vRealize Orchestrator 端口号的 URL。

传输协议必须是 HTTPS。如果未指定端口，则使用默认端口 443。

要使用 vRealize Automation 设备中嵌入的默认 vRealize Orchestrator 实例，请键入 **https://vrealize-automation-appliance-hostname:443/vco**。

- 5 在用户名和密码文本框中提供 vRealize Orchestrator 凭据以连接到 vRealize Orchestrator 端点。

使用的凭据应该对要从 IaaS 调用的所有 vRealize Orchestrator 工作流具有“执行”权限。

要使用 vRealize Automation 设备中嵌入的默认 vRealize Orchestrator 实例，用户名应使用 **administrator@vsphere.local**，且密码应使用配置 SSO 时指定的管理员密码。

- 6 在优先级文本框中输入大于或等于 1 的整数。

值越低表示优先级越高。

- 7 （可选）单击属性，并为端点添加提供的自定义属性、属性组或您自己的属性定义。
- 8 单击确定。

## 使用 vRealize Orchestrator 自定义 IaaS 工作流

可以在 vRealize Orchestrator 中使用单个工作流来将自定义逻辑注入到 IaaS 工作流存根并将自定义生命周期分配给计算机蓝图。

**注** 工作流存根被事件代理工作流订阅所取代。目前仍会提供工作流存根，它们仍受支持且可以使用，但预计会在将来的 vRealize Automation 版本中将其移除。为确保将来的产品兼容性，您应该使用工作流订阅，以基于状态更改运行自定义工作流。请参见[配置工作流订阅以扩展 vRealize Automation](#)。

您必须对自定义 vRealize Orchestrator 工作流进行设计，使其接受字符串输入。如果自定义工作流需要复杂数据类型，请创建可查找此复杂值并将其转换为字符串的包装程序工作流。有关示例包装工作流，请参见[库 > vRealize Automation > 基础架构 > 可扩展性](#)中提供的示例工作流模板。

### 为蓝图及其虚拟机分配状态更改工作流

要将自定义 vRealize Orchestrator 工作流配置为在主计算机工作流中的特定阶段运行，请将自定义工作流与状态更改工作流存根相关联并将这些工作流分配给蓝图。

**注** 工作流存根被事件代理工作流订阅所取代。目前仍会提供工作流存根，它们仍受支持且可以使用，但预计会在将来的 vRealize Automation 版本中将其移除。为确保将来的产品兼容性，您应该使用工作流订阅，以基于状态更改运行自定义工作流。请参见[配置工作流订阅以扩展 vRealize Automation](#)。

#### 前提条件

使用 vRealize Automation 插件库的“可扩展性”子目录中提供的工作流模板创建要在计算机生命周期内运行的自定义工作流。

#### 步骤

- 1 从 Orchestrator 客户端的下拉菜单中，选择**运行或设计**。
- 2 单击**工作流**视图。
- 3 选择[库 > vRealize Automation > 基础架构 > 可扩展性](#)。
- 4 右键单击**为蓝图及其虚拟机分配状态更改工作流**，然后选择**启动工作流**。
- 5 从要启用的 **vCAC 工作流存根**下拉菜单中选择存根，以选择要运行该工作流的生命周期阶段。
- 6 选择 IaaS 主机。
- 7 单击**下一步**。
- 8 选择要向其分配工作流的蓝图。
- 9 选择是否将这些工作流应用到通过此蓝图置备的现有计算机。
- 10 选择要在计算机生命周期内运行的工作流。
- 11 配置哪些工作流输入值作为自定义属性添加到计算机中。
  - a 添加 vCO 工作流输入作为蓝图属性。
  - b 添加上次 vCO 工作流运行的输入值作为蓝图属性。

**12** 单击提交。

## 配置 workflow 订阅以扩展 vRealize Automation

您可以创建执行如下操作的工作流订阅：使用事件代理服务监控 vRealize Automation 中已注册的服务的事件消息，然后在满足该订阅中的条件时运行指定的 vRealize Orchestrator 工作流。要配置订阅，请指定事件主题、触发条件和在触发时运行的工作流。

租户管理员可以创建和管理特定于其租户的工作流订阅。

系统管理员可以创建和管理系统工作流订阅。已创建的系统工作流订阅对任何租户中的事件和系统事件有效。

### 随 vRealize Automation 提供的事件主题

事件主题描述了其他服务发送到事件代理服务的事件消息的类型。可以选择事件主题并基于该主题配置工作流订阅。

**表 1-4. 事件主题**

事件主题名称	描述	服务
蓝图组件已完成	复合蓝图中的蓝图组件完成置备。该组件是复合蓝图中的任意蓝图。	composition-service
请求的蓝图组件	请求复合蓝图中的蓝图组件。该组件是复合蓝图中的任意蓝图。	composition-service
蓝图配置	创建、更新或删除蓝图。	composition-service
已完成的蓝图请求	复合蓝图完成置备。此事件主题包括所有蓝图组件。它不包括独立的 XaaS 蓝图。	composition-service
已请求的蓝图	复合蓝图已请求。此事件主题不包括 XaaS 蓝图。	composition-service
业务组配置	创建、更新或删除业务组。	identity
完成的组件操作	请求部署操作时，在已部署蓝图组件上运行的操作。	composition-service
请求的组件操作	请求部署操作时，请求要在已部署蓝图组件上运行的操作。	composition-service
完成的部署操作	已部署蓝图上的操作完成运行，包括运行所有组件操作。	composition-service
请求的部署操作	请求已部署蓝图上的操作。	composition-service
EventLog 默认事件	向事件日志添加一个标准条目。该日志条目不分配给订户。	eventlog-service
IPAM IP 生命周期事件完成	IP 分配或取消分配请求已完成。	ipam-service
计算机生命周期	在已置备的计算机上运行提供的 IaaS 操作。	iaas-service
计算机置备	正在置备 IaaS 计算机。	iaas-service

表 1-4. 事件主题 （续）

事件主题名称	描述	服务
Orchestration 服务器配置	创建、更新、删除或修改 vRealize Orchestrator 服务器配置，以使用其他默认实例。	o11n-gateway-service
Orchestration 服务器配置 (XaaS) - 已弃用	创建、更新、删除或修改 vRealize Orchestrator 服务器配置，以使用其他默认实例。	advanced-designer-service
批准后	批准后策略级别配置为使用事件订阅选项。	approval-service
批准前	批准前策略级别配置为使用事件订阅选项。	approval-service
资源回收完成事件	资源租约已过期，并且资源已回收。	management-service

## workflow 订阅和事件代理术语

在使用 workflow 订阅和事件代理服务时，您可能会遇到一些特定于订阅和事件代理服务的术语。

表 1-5. workflow 订阅和事件代理术语

术语	描述
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。
事件	指示生成者或由其管理的任意实体中的状态更改。事件是记录有关事件出现的信息的实体。
消息	在各个服务和组件之间传输有关事件的信息。例如，从生成者传输到事件代理服务，或者从事件代理服务传输到订户。
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。
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。
订户	基于订阅定义使用发布到事件代理服务的事件。订户还可以称为使用者。
提供者	在事件代理服务中注册事件主题。
生成者	将事件发布到事件代理服务。
系统管理员	拥有使用 API 或 vRealize Automation 插件来创建、读取、更新和删除租户 workflow 订阅与系统 workflow 订阅的特权的用户。vRealize Automation 不包含供系统管理员使用的用户界面。
租户管理员	拥有创建、读取、更新和删除其租户的租户 workflow 订阅的特权的用户。
workflow 订阅	指定事件主题和 vRealize Orchestrator workflow 的触发条件。

表 1-5. workflows 订阅和事件代理术语（续）

术语	描述
系统工作流订阅	对系统事件和所有租户中的事件做出响应的专用工作流订阅。
租户工作流订阅	指定了在哪些条件下对同一个租户中的事件触发 vRealize Orchestrator 工作流的专用工作流订阅。

## 可阻止和可回复事件主题

事件主题可能支持可阻止事件和可回复事件。工作流订阅的行为取决于主题是否支持这些事件类型以及您对工作流订阅的配置方式。

### 不可阻止事件主题

不可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。但是，触发事件肯定会发生，并将运行与该订阅关联的 vRealize Orchestrator 工作流。如果主题是可回复的，则非阻止订阅只会返回响应。

### 可阻止事件主题

有些事件主题支持阻止。如果工作流订阅被标记为阻止，则在第一个工作流完成之前，具有匹配条件的任何其他工作流订阅无法收到满足已配置条件的所有消息。如果同一个事件主题有多个阻止工作流订阅，您可以设置订阅的优先级。

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，这些订阅将基于名称按字母顺序运行。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。由于阻止工作流订阅以同步方式运行，因此当后续工作流订阅收到通知时，已更改的事件负载包含已更新的事件。

您可以对一个或多个工作流订阅应用阻止，具体取决于所选工作流和目标。

例如，您有两个置备工作流订阅，其中第二个工作流取决于第一个工作流的结果。如果第一个工作流在置备期间更改某个属性，则第二个工作流会在文件系统中记录该新属性（例如虚拟机名称）。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，因为后者使用 ChangeProperty 订阅的结果。置备虚拟机时，ChangeProperty 订阅会开始运行。由于 RecordProperty 订阅条件基于置备后条件，因此消息会触发 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消息。当名称已更改并且第一个工作流已完成，第二个工作流会开始运行并在文件系统中记录该名称。

即使事件主题支持阻止，您也可以创建非阻止工作流订阅，前提是该工作流订阅没有任何从属的后续工作流。该工作流订阅会被触发并运行 vRealize Orchestrator 工作流，而不会进一步与 vRealize Automation 或外部系统交互。

## 可回复事件主题

有些事件主题支持来自已订阅服务的回复。注册可回复事件主题的服务可以接受提供 workflow 输出的回复事件，该 workflow 输出通常在与系统或用户交互后生成。回复输出参数必须满足在回复架构中定义的标准，以便发布原始可回复事件的 **vRealize Automation** 服务可以处理它。例如，批准前 workflow 订阅和批准后 workflow 订阅是可回复的。如果您创建一个 workflow 以向外部系统发送批准请求，则 **vRealize Automation** 将处理回复（已批准或已拒绝）并置备目录项，否则会通知用户该请求。

回复可以是来自 **vRealize Orchestrator** workflow 的输出，但如果 workflow 超时或失败，则回复可能会失败。如果回复来自 workflow 输出参数，则该回复必须采用正确的回复架构格式。

## 为 workflow 订阅创建 vRealize Orchestrator workflow 的最佳做法

workflow 订阅基于特定主题架构。要确保订阅可以启动 **vRealize Orchestrator** workflow，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。

### workflow 输入参数

所创建的自定义 workflow 的负载中可以包含所有参数或单个参数（使用所有数据）。

- 要包含各个参数，请配置一个或多个参数。请确保名称和类型与架构中指定的名称和类型匹配。架构中的复杂类型应在 workflow 中定义为“属性”。
- 要使用单个参数，请配置一个类型为 **Properties** 的参数。可以提供任何有用名称。例如，可以使用 **payload** 作为参数名称。

### workflow 输出参数

所创建的自定义 workflow 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 workflow 输出参数必须与回复架构匹配。

## workflow 订阅设置

订阅选项基于 **vRealize Automation** 中的事件消息来确定 workflow 在何时运行。可以使用这些选项来管理订阅。

订阅表示用户有意订阅给定事件主题的事件并在收到与已定义的条件匹配的主题事件时运行 workflow。

您必须是租户管理员才能创建工作流订阅。所有 workflow 订阅均特定于租户。

要管理工作流订阅，请选择 **管理 > 事件 > 订阅**。

表 1-6. workflow 订阅选项

选项	描述
新建	创建新订阅。
编辑	修改所选订阅。 如果订阅已发布，已保存的更改将立即生效。 无论订阅已发布还是未发布，您都无法编辑事件主题或修改阻止选项。

表 1-6. 工作流订阅选项（续）

选项	描述
发布	使订阅处于活动状态。 处理来自事件代理服务的事件并评估订阅条件。如果符合已配置的条件，将触发工作流。
取消发布	使订阅恢复到草稿状态。 订阅在环境中不再处于活动状态，并且不再接收事件。 重新发布订阅后，该订阅将开始接收新事件，但它不会接收过去的事件。
删除	删除所选订阅。

## 将事件主题分配到订阅

事件主题是 vRealize Automation 中提供的事件类别。您可以选择事件主题，以基于该事件主题定义订阅。

事件主题是相似事件分组在一起形成的类。分配给一个订阅时，事件主题将定义哪些事件会触发订阅。

### 步骤

- 1 选择**管理 > 事件 > 订阅**
- 2 单击**新建**，然后选择**事件主题**。

表 1-7. 事件主题详细信息

事件主题详细信息	描述
主题 ID	事件主题标识符。
名称	事件主题的名称。
描述	事件主题的描述。
发布者	为其注册了此事件主题的服务的名称。
可阻止	指示是否可以为该事件主题创建阻止订阅。 阻止订阅用于更改事件的负载，或者在同一事件的第二个工作流的结果取决于第一个工作流的结果时用于运行自定义逻辑。
可回复	指示事件主题订阅是否可以将回复事件发布到最初生成了事件的服务。如果值为 <b>yes</b> ，则在工作流完成后，系统会向发布了原始事件的服务发送回复。回复包含 vRealize Orchestrator 工作流的输出和任何错误详细信息。
架构	描述了事件负载的结构。 您可使用架构创建工作流，并利用该工作流使用负载信息。

## 将工作流条件分配到订阅

为订阅配置的条件决定了是否基于事件数据触发并运行工作流。

您可以定义工作流条件来控制如何启动工作流。如果选择**根据条件运行**，则可用类型包括：

- 数据



此类型包括特定于所选事件主题的事件消息中的信息。例如，如果为虚拟机生命周期事件主题创建条件，则数据字段与蓝图和虚拟机相关。如果选择批准前事件主题，则数据字段与批准策略相关。

您还可以为不包含在架构中的字段添加条件，方法是在树上方的文本框中输入路径。请使用 **\${PATH}** 格式。PATH 表示架构中的路径。请用 ~ 分隔节点。例如，  
**\${data~machine~properties~SomeCustomProperty}**。

#### ■ 核心事件消息值

这包括有关事件消息的一般信息。例如，事件类型、时间戳或用户名。

### 前提条件

#### 步骤

- 1 选择**管理 > 事件 > 订阅**
- 2 单击**新建**，然后选择**事件主题**。
- 3 单击**下一步**，并定义 **workflow 条件**。

**表 1-8. 条件类型**

条件	描述
为所有事件运行	选定的 workflow 将在收到此事件主题的消息时运行。
根据条件运行	<p>选定的 workflow 将在检测到事件消息并且事件满足已配置的条件时运行。</p> <p>如果选择此选项，则您必须基于要触发选定的 workflow 的事件数据定义此订阅的条件。</p> <ul style="list-style-type: none"> <li>■ <b>单个条件</b>。workflow 将在配置的子句成立时触发。</li> <li>■ <b>以下所有项</b>。workflow 将在所有子句均成立并且您提供了至少两个条件时触发。</li> <li>■ <b>以下任一项</b>。workflow 将在至少一个子句成立并且您提供了至少两个条件时触发。</li> <li>■ <b>非以下项</b>。workflow 将在所有子句均不成立时触发。</li> </ul> <p>如果基于常数值创建条件，条件在处理值时不区分大小写。例如，如果您的条件是蓝图名称包含 UNIX，但您的蓝图名称使用 Unix，则条件仍会正确处理。</p> <p>要更改条件名称以匹配蓝图名称，必须先将值更改为不包含相同的字符串。例如，要编辑条件 UNIX，请将值更改为 xxxx 并保存，然后将 xxxx 更改为 Unix 并保存。</p>

### 将 workflow 分配到订阅

所选 vRealize Orchestrator workflow 将在经过评估确定订阅条件成立时运行。

workflow 可以将多种 ABX 操作、决策和结果整合在一起，按照特定顺序执行，从而在虚拟环境中完成特定的任务或进程。workflow 可执行的任务包括虚拟机置备、备份、定期维护、发送邮件、执行 SSH 操作、管理物理基础架构和其他常规实用操作。workflow 可按其功能接受输入值。workflow 还可以调用其他 workflow。例如，您可以在多个不同 workflow 中重复使用一个用于启动虚拟机的工作流。

您可以在订阅中链接工作流来自动化一个响应触发事件的过程。这样，工作流可以执行并生成结果，而无需用户干预。更具体地说使工作流可以在虚拟机置备生命周期事件中运行。您还可以在状态相同的工作流之间重用订阅输出来共享数据。注册到同一个生命周期状态的工作流可以合并输出负载。

### 前提条件

该工作流必须存在于 vRealize Orchestrator 中，就像在 **管理 > vRO 配置 > 服务器配置** 中所列的一样。

### 步骤

- 1 选择**管理 > 事件 > 订阅**
- 2 单击**新建**，然后选择**事件主题**。
- 3 单击**下一步**，并定义**工作流条件**。
- 4 单击**下一步**，然后选择要应用到订阅的**工作流**。

表 1-9. “工作流”选项卡

“工作流”选项卡	描述
选择工作流	导航到工作流。
选定的工作流	显示有关工作流的信息（包括输入参数和输出参数），以便您可以验证是否要运行该工作流。

## 定义工作流订阅详细信息

订阅详细信息决定了订阅的处理方式。

可以通过定义其他订阅详细信息，进一步配置和自定义订阅。

### 步骤

- 1 选择**管理 > 事件 > 订阅**
- 2 单击**新建**，然后选择**事件主题**。
- 3 单击**下一步**，然后分配**工作流条件**。
- 4 单击**下一步**，然后选择要分配给订阅的**工作流**。
- 5 单击**下一步**，然后定义**工作流订阅详细信息**。

表 1-10. 工作流订阅详细信息

详细信息	描述
名称	默认情况下，所显示的名称是所选工作流的名称。 此名称显示在订阅列表中。名称在租户中必须是唯一的。
优先级	阻止订阅的运行顺序。 零是最高优先级。如果某个事件主题有多个优先级相同的阻止工作流订阅，这些订阅将基于订阅名称按字母顺序进行处理。 此选项仅适用于阻止工作流订阅。

表 1-10. 工作流详细信息（续）

详细信息	描述
超时 (分钟)	<p>输入工作流必须在多少分钟内完成，否则视为失败。</p> <p>如果工作流无法在允许的时间内完成，系统会取消该工作流并向优先级列表中的下一个订阅发送消息。</p> <p>如果未提供任何值，则超时为无限制。</p> <p>需要回复阻止事件或可回复事件的服务可能具有各自的默认超时值。例如，IaaS 置备和生命周期事件主题的超时为 30 分钟。此值在 IaaS 服务器上配置。批准主题的默认值为 24 小时。此值在系统上配置。</p>
描述	默认情况下，所显示的描述是工作流描述。
阻止	<p>决定工作流是否可以在等待回复时阻止同一个事件主题的后续工作流接收事件消息。</p> <p>根据优先级顺序，同一个事件主题中已启用阻止的订阅比未配置为阻止的订阅优先接收消息。工作流完成后，系统会向下一个优先级的阻止订阅发送消息。处理完所有阻止订阅后，系统会同时向所有非阻止订阅发送消息。</p> <p>仅当事件主题可阻止时，阻止选项才可用。“事件主题”选项卡上提供了此信息。</p> <p>“事件主题”选项卡上会显示是否具备阻止条件。</p> <ul style="list-style-type: none"> <li>■ 如果未选中该复选框，则事件代理不会阻止后续工作流。</li> <li>■ 如果选中该复选框，则事件代理会基于已配置的条件计算哪些工作流订阅符合此事件的条件并按优先级顺序运行工作流。事件代理会等待来自每个工作流的响应，然后再运行下一个工作流。自运行当前工作流以来发生更改的所有参数会传递到队列中的下一个工作流。</li> </ul> <p>等待响应时，所有其他工作流都不会收到有关事件的通知，直至占用系统作出响应为止。</p> <p>创建工作流订阅后，无法修改此选项。</p>
工作流失败时停止处理	如果阻止工作流订阅失败，则只有解决故障错误后，所有后续工作流才会运行。失败消息将添加到事件日志中并向发出请求的用户发送一封电子邮件。

## 6 单击完成。

## 处理置备和生命周期工作流订阅

您可以创建置备和生命周期工作流订阅，以便使用 vRealize Orchestrator 来扩展 IaaS 计算机的管理。置备订阅用于扩展您可在置备过程中执行的操作。生命周期订阅用于扩展您可在用户正在管理已置备项目时执行的操作。

### IaaS 服务集成

您可以创建一个基于置备或生命周期事件主题的工作流订阅，并使得该工作流订阅可根据 IaaS 服务生成的消息运行自定义 vRealize Orchestrator 工作流。vRealize Automation 包含两个可用于 IaaS 集成的事件主题。

- 计算机置备。创建在置备和处置 IaaS 计算机期间运行工作流的工作流订阅。

- 计算机生命周期。创建在所有者用户对已置备计算机执行相关管理操作时运行工作流的工作流订阅。

## 为置备和生命周期工作流配置 vRealize Orchestrator 工作流

必须对 vRealize Orchestrator 工作流进行配置，以使其支持 IaaS 服务消息。

### 置备和生命周期事件主题架构

计算机置备和计算机生命周期事件主题使用相同的生命周期架构。两者的区别在于触发状态不同。计算机置备基于置备状态和事件接收消息，而计算机生命周期基于活动状态和事件接收消息。部分置备状态包括 BuildingMachine 和 Disposing。部分生命周期状态包括 InstallTools 和 Off。

事件消息是事件数据负载。以下是事件数据负载的结构。

```
{
  machine : {
    id          : STRING,      /* IaaS machine ID */
    name        : STRING,      /* machine name */
    externalReference : STRING, /* machine ID on the hypervisor */
    owner       : STRING,      /* machine owner */
    type        : INTEGER,      /* machine type: 0 - virtual machine; 1 - physical machine; 2
- cloud machine */
    properties   : Properties /* machine properties, see notes below how to expose virtual
machine properties */
  },
  blueprintName : STRING,      /* blueprint name */
  componentId   : STRING,      /* component id */
  componentTypeId : STRING,    /* component type id */
  endpointId    : STRING,      /* endpoint id */
  requestId     : STRING,      /* request id */
  lifecycleState : {                               /* see Life Cycle State
Definitions*/
    state : STRING,
    phase : STRING,
    event : STRING
  },
  virtualMachineEvent : STRING, /* fire an event on that machine - only processed
by Manager Service as consumer */
  workflowNextState   : STRING, /* force the workflow to a specific state - only
processed by Manager Service as consumer */
  virtualMachineAddOrUpdateProperties : Properties, /* properties on the machine to add/update - only
processed by Manager Service as consumer */
  virtualMachineDeleteProperties      : Properties /* properties to remove from the machine - only
processed by Manager Service as consumer */
}
```

vRealize Orchestrator 参数按名称和类型映射到事件的负载。

当您使用 `virtualMachineEvent` 和 `workflowNextState` 作为输出参数时，您提供的值必须表示来自触发事件和开始当前 vRealize Orchestrator 工作流的工作流的状态或事件。要查看可能的生命周期状态和事件，请参见 [VMPS 主工作流生命周期状态](#)和[置备按计算机类型的生命周期状态](#)。

## 使用可扩展性自定义属性

虚拟机自定义属性不包含在事件负载中，除非将其指定为生命周期状态的可扩展性自定义属性。您可以将这些属性添加到 IaaS 端点、预留、蓝图、请求和支持自定义属性的其他对象。

添加到对象的自定义属性的格式为 `Extensibility.Lifecycle.Properties.{workflowName}.{stateName}`。

例如，如果要在虚拟机状态为 `BuildingMachine` 时包含隐藏的属性以及所有以“Virtual”开头的属性，请将自定义属性添加到蓝图中的计算机。此示例中的自定义属性名称为 `Extensibility.Lifecycle.Properties.VMPMasterWorkflow32.BuildingMachine`，值为 `__*` 和 `Virtual*`（用逗号分隔）。

双下划线 (`__*`) 包括隐藏的属性。`Virtual*` 值包括所有以“Virtual”开头的属性。星号 (\*) 是通配符，可以用作唯一值，但这样使用通配符会导致传输大量数据。

如果您有多个后续触发的工作流订阅包括自定义属性，则必须在工作流中包括相应的条目，以确保负载检查保留自定义属性。

表 1-11. 保留自定义属性的任务条目

状态	任务条目
已添加或已更新的自定义属性	<pre>virtualMachineAddOrUpdateProperties = payload.virtualMachineAddOrUpdateProperties    new Properties();</pre>
已删除的自定义属性	<pre>virtualMachineDeleteProperties = payload.virtualMachineDeleteProperties    new Properties();</pre>

## 基于生命周期或置备架构创建 vRealize Orchestrator 工作流

所创建的自定义工作流必须具有输入参数 `payload`，且类型必须为 `Properties`。当工作流在 vRealize Orchestrator 中运行时，置备或生命周期事件数据负载会放在此参数中。您还可以在事件的负载中包含与字段名称和类型匹配的各个输入参数。

## 工作流订阅生命周期状态定义

如果基于生命周期状态配置工作流订阅条件，则以下定义可能有助于确定值。

每条消息都包含基于 IaaS 计算机状态更改的 `lifecycleState` 元素。

该元素在消息中具有以下结构。

```
lifecycleState : {
  state : STRING,
  phase : STRING,
  event : STRING
}
```

表 1-12. LifecycleState 元素

属性	描述	格式和值	示例
state	包含工作流名称和状态名称。	{workflowName}.{stateName}	<ul style="list-style-type: none"> <li>■ VMPSMasterWorkflow32.Requested</li> <li>■ VMPSMasterWorkflow32.MachineActivated</li> <li>■ BasicVmWorkflow.BuildComplete</li> </ul>
phase	包含触发了消息的阶段。	PRE、POST、EVENT	<ul style="list-style-type: none"> <li>■ PRE。在进入此状态时发布事件。</li> <li>■ POST。在退出此状态时发布事件。</li> <li>■ EVENT。在此状态下收到 IaaS 事件时发布事件。</li> </ul>
event	包含事件。此属性是可选的，并且仅在阶段为 EVENT 时存在。	{workflowName}. {stateName}.EVENT.{eventName}	<ul style="list-style-type: none"> <li>■ VMPSMasterWorkflow32.Requested.EVENT.OnProvisionMachine</li> <li>■ VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnBuildSuccess</li> <li>■ BasicVmWorkflow.CreatingMachine.EVENT.OnCreatingMachineComplete</li> </ul>

### VMPS 主工作流生命周期状态

VMPS 主工作流生命周期状态代表 IaaS 虚拟机从请求到销毁的生命周期。基于生命周期状态事件和生命周期状态名称创建触发条件时，可以使用 VMPS 主工作流状态和事件。

每个虚拟机都经历四个基本阶段。

- 请求。包括批准。
- 置备。包括不同的置备类型，如创建、克隆、kickstart 或 WIM。
- 管理。包括各种操作，如打开电源、关闭电源或生成快照。
- 销毁。包括对计算机进行取消激活、取消置备和处置。

这些基本阶段都包含在主工作流中。为以下事件主题创建条件时，可以使用 VMPSMasterWorkflow32 状态：

- 计算机生命周期
- 计算机置备

全局事件状态是 VMPS 主工作流发送到事件代理的消息。全局事件可以在任何时候触发。

可以订阅客户端以侦听事件，但只有表条目具有触发器字符串值时，才应引发事件。例如，事件 [触发字符串]（主题）。

表 1-13. 全局事件

状态（主题）	事件 [触发字符串]（主题）
Global	<ul style="list-style-type: none"> <li>■ onBuildFailure (Provision)</li> <li>■ OnBuildSuccess (Provision)</li> <li>■ OnFinalizeMachine [Destroy] (Provision)</li> <li>■ OnForceUnregisterEvent [ForceUnregister] (Provision)</li> <li>■ ReconfigureVM.Pending [ReconfigureVM.Pending] (Active)</li> <li>■ ReconfigureVM.ExecutionUpdated (Active)</li> <li>■ ReconfigureVM.RetryRequestMade (Active)</li> <li>■ ReconfigureVM.Failed (Active)</li> <li>■ ReconfigureVM.Successful (Active)</li> <li>■ ReconfigureVM.Complete (Active)</li> <li>■ ReconfigureVM.Canceled (Active)</li> </ul>

活动的全局状态是可在已置备的计算机上运行的操作。

表 1-14. 活动事件

状态	事件 [触发字符串]（主题）
Active	<ul style="list-style-type: none"> <li>■ OnExpireLease [Expire] (Active)</li> <li>■ OnForceExpire [ForceExpire] (Active)</li> <li>■ onReprovision [Reprovision] (Active)</li> <li>■ onResetBuildSuccess [ResetBuildSuccess] (Active)</li> </ul>

在主工作流中，置备事件在计算机置备生命周期内发生。活动事件是可在已置备的计算机上运行的操作。有关主工作流的示意图，请参见 [VMPS 主工作流的示例](#)。

每种计算机类型具有各自的置备工作流。有关各种计算机类型的信息，请参见[置备按计算机类型的生命周期状态](#)。

表 1-15. VMPSMasterWorkflow32 状态和事件

状态（主题）	事件 [触发字符串]（主题）
BuildingMachine	
■ Pre(Provision)	
■ Post(Provision)	
DeactivateMachine	
■ Pre(Provision)	
■ Post(Provision)	
Disposing	<ul style="list-style-type: none"> <li>■ OnDisposeComplete(Provision)</li> <li>■ OnDisposeTimeout(Provision)</li> <li>■ OnUnregisterMachine [Unregister] (Provision)</li> </ul>
Expired	<ul style="list-style-type: none"> <li>■ OnActiveExpiredMachine [ActivateExpiredMachine] (Active)</li> <li>■ TurnOffFromExpired [TurnOffExpiredMachine] (Active)</li> </ul>
■ Pre(Active)	
■ Post(Active)	

表 1-15. VMPSMasterWorkflow32 状态和事件（续）

状态（主题）	事件 [触发字符串]（主题）
InstallTools ■ Pre(Active) ■ Post(Active)	■ InstallToolsComplete(Active) ■ TimeoutInstallTools(Active)
Leased	■ OnChangeLease (Active) ■ OnUpdateDescription (Active) ■ OnUpdateOwner (Active)
MachineActivated ■ Pre(Provision) ■ Post(Provision)	■ OnCatalogRegistrationComplete (Provision)
MachineProvisioned ■ Pre(Provision) ■ Post(Provision)	
Off ■ Pre(Active) ■ Post(Active)	■ OnForceOn [ForceOn] (Active) ■ OnResetOff [Turn Off] (Active) ■ OnTurnOn [Turn On] (Active)
On ■ Pre(Active) ■ Post(Active)	■ OnForceOff [ForceOff] (Active) ■ onInstallTools [InstallTools] (Active) ■ OnReboot [Reboot] (Active) ■ OnReset [Reset] (Active) ■ OnResetOn [Turn On] (Active) ■ OnShutdown [Shutdown] (Active) ■ OnSuspend [Suspend] (Active) ■ OnTurnOff [Turn Off] (Active)
Rebooting ■ Pre(Active) ■ Post(Active)	■ OnRebootComplete(Active) ■ TimeoutFromReboot(Active)
RegisterMachine ■ Pre(Provision) ■ Post(Provision)	■ onRegisterComplete(Provision) ■ RegisterTimeout(Provision)
Requested ■ Pre(Provision) ■ Post(Provision)	■ OnProvisionMachine [Provision] (Provision)
Resetting ■ Pre(Active) ■ Post(Active)	■ OnResetComplete(Active) ■ TimeoutFromReset(Active)
ShuttingDown ■ Pre(Active) ■ Post(Active)	■ OnShutdownComplete(Active) ■ TimeoutFromShutdown(Active)
Suspending ■ Pre(Active) ■ Post(Active)	■ OnSuspendComplete(Active) ■ TimeoutFromSuspend(Active)

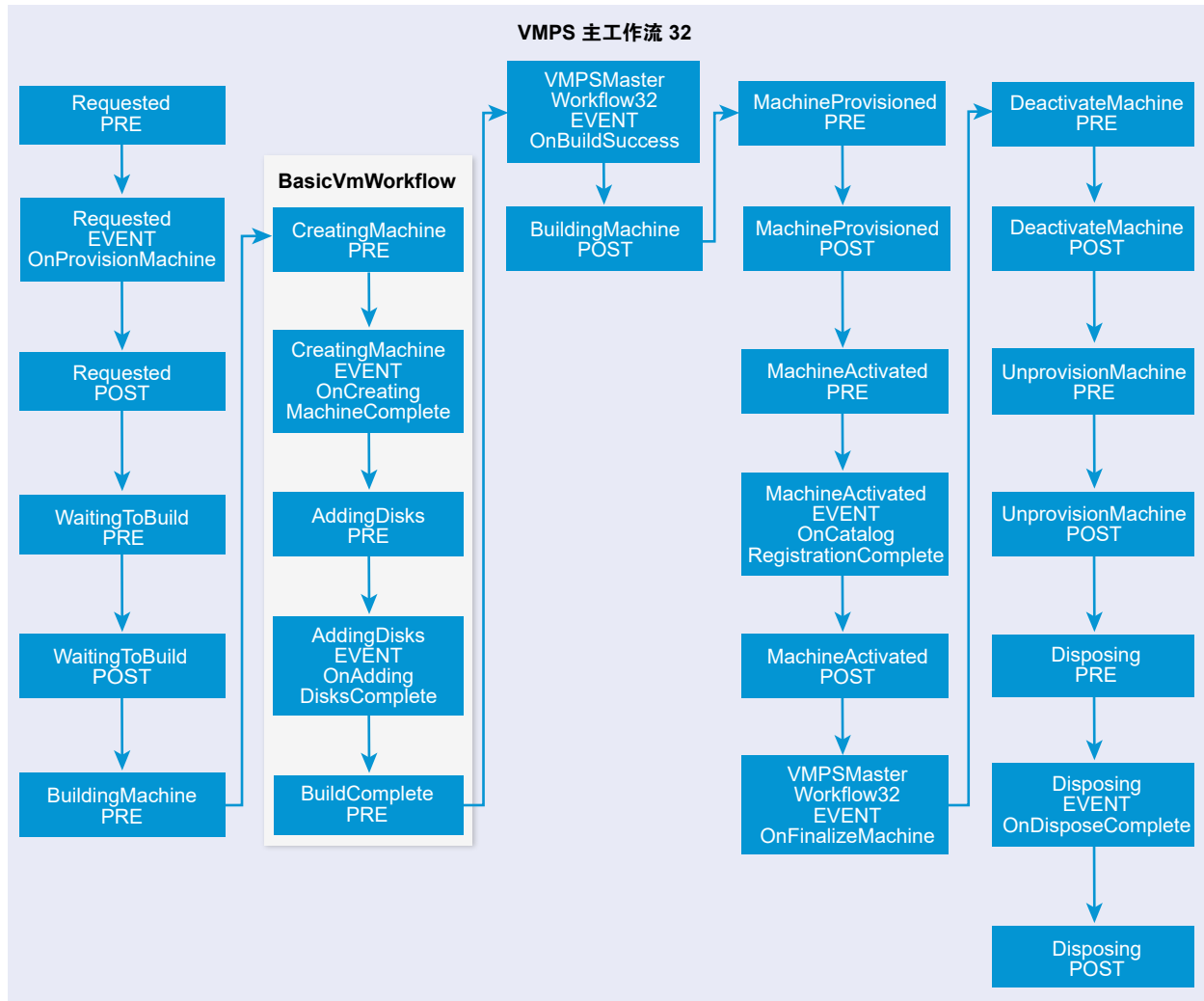


表 1-15. VMPSMasterWorkflow32 状态和事件（续）

状态（主题）	事件 [触发字符串]（主题）
TurningOff	■ OnTurningOffComplete(Active)
■ Pre(Active)	■ TimeoutFromPowerOff(Active)
■ Post(Active)	
TurningOn	■ OnTurningOnComplete(Active)
■ Pre(Active)	■ TimeoutPowerOn(Active)
■ Post(Active)	
UnprovisionMachine	
■ Pre(Provision)	
■ Post(Provision)	
WaitingToBuild	
■ Pre(Provision)	
■ Post(Provision)	

## VMPS 主工作流的示例

VMPS 工作流是主工作流，其他置备工作流嵌入在其中。此示例以基本虚拟机工作流为例来说明虚拟机的生命周期。该工作流不代表您的环境中的特定工作流。



## 置备按计算机类型的生命周期状态

按计算机类型的生命周期状态特定于某些虚拟机类型。除了主工作流，您还可以在创建工作流订阅的触发条件时使用置备工作流状态和事件。

可以订阅客户端以侦听事件，但只有表条目具有触发器字符串值时，才应引发事件。例如，事件[触发字符串]（主题）。

## Blade Logic（裸机）

### 状态（主题）

BuildFinished

■ Pre(Provision)

CreatingMachine

■ Pre(Provision)

### 事件（主题）

**Opware（裸机）**

状态（主题）	事件（主题）
BuildFinished	
■ Pre(Provision)	
OpwareRegister	■ OnOpwareRegister(Provision)
■ Pre(Provision)	

**云置备 workflow**

状态（主题）	事件（主题）
BuildComplete	
■ Pre(Provision)	
CloudProvisioning	■ OnCloudProvisioningTimeout(Provision)
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	

**应用程序服务置备 workflow**

状态（主题）	事件（主题）
AppServiceProvisioning	■ OnAppServiceProvisioningTimeout(Provision)
■ Pre(Provision)	
BuildComplete	
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	

**基本虚拟机 workflow**

状态（主题）	事件（主题）
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	

## Opware（虚拟机）

状态（主题）	事件（主题）
AddingDisks ■ Pre(Provision)	■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision)
BuildFinished ■ Pre(Provision)	
CreatingVM ■ Pre(Provision)	■ OnCreateVMComplete(Provision) ■ OnCreateVMTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
OpwareRegister ■ Pre(Provision)	■ OnOpwareRegister(Provision)

## 云 Linux Kickstart workflow

状态（主题）	事件（主题）
BuildComplete ■ Pre(Provision)	
CreatingMachine ■ Pre(Provision)	■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision)
CustomizeOS ■ Pre(Provision)	■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
InstallingOS ■ Pre(Provision)	■ OnInstallingOSComplete(Provision) ■ OnInstallingOSTimeout(Provision)

## 克隆 workflow

状态（主题）	事件（主题）
BuildComplete ■ Pre(Provision)	
CloneMachine ■ Pre(Provision)	■ OnCloneMachineComplete(Provision) ■ OnCloneMachineTimeout(Provision)
CustomizeMachine ■ Pre(Provision)	■ OnCustomizeMachineComplete(Provision) ■ OnCustomizeMachineTimeout(Provision)
CustomizeOS	■ OnCustomizeOS(Provision) ■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision)
EjectCD ■ Pre(Provision)	■ OnEjectCDComplete(Provision) ■ OnEjectCDTimeout(Provision)

状态（主题）	事件（主题）
FailedProvisioning	
■ Pre(Provision)	
FinalizeProvisioning	■ OnFinalizeComplete(Provision)
■ Pre(Provision)	■ OnFinalizeTimeout(Provision)
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallSoftware	■ OnInstallSoftwareComplete(Provision)
■ Pre(Provision)	■ OnInstallSoftwareTimeout(Provision)
MountCD	■ OnMountCDComplete(Provision)
■ Pre(Provision)	■ OnMountCDTimeout(Provision)
PostInstallSoftwareChecks	
■ Pre(Provision)	
PrepareInstallSoftware	
■ Pre(Provision)	

## 云 WIM 映像 workflow

状态（主题）	事件（主题）
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallOS	■ onInstallOSComplete(Provision)
■ Pre(Provision)	■ OnInstallOSTimeout(Provision)
Reboot	■ OnRebootComplete(Provision)
■ Pre(Provision)	■ OnRebootTimeout(Provision)
SetupOS	■ OnSetupOSComplete(Provision)
■ Pre(Provision)	■ OnSetupOSTimeout(Provision)

## 外部置备 workflow

状态（主题）	事件（主题）
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)

状态（主题）	事件（主题）
EpiRegister ■ Pre(Provision)	■ OnEpiRegisterComplete(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)

## Linux Kickstart workflow

状态（主题）	事件（主题）
AddingDisks ■ Pre(Provision)	■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision)
BuildComplete ■ Pre(Provision)	
CreatingMachine ■ Pre(Provision)	■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision)
CustomizeOS ■ Pre(Provision)	■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision)
EjectingCD ■ Pre(Provision)	■ OnEjectingCDComplete(Provision) ■ OnEjectingCDTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
InstallingOS ■ Pre(Provision)	■ OnInstallingOSComplete(Provision) ■ OnInstallingOSTimeout(Provision)

## 物理机置备 workflow

状态（主题）	事件（主题）
FailedProvisioning ■ Pre(Provision)	
FinalizeProvisioning ■ Pre(Provision)	■ OnFinalizeProvisioningTimeout(Provision)
InitializeProvisioning ■ Pre(Provision)	■ OnInitializeProvisioningTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallOS ■ Pre(Provision)	■ OnInstallOSComplete(Provision) ■ OnInstallOSTimeout(Provision)

状态（主题）	事件（主题）
Reboot	■ OnRebootComplete(Provision)
■ Pre(Provision)	■ OnRebootTimeout(Provision)
SetupOS	■ OnSetupOSComplete(Provision)
■ Pre(Provision)	■ OnSetupOSTimeout(Provision)

## 物理机 PXE 置备 workflow

状态（主题）	事件（主题）
CheckHardwareType	
■ Pre(Provision)	
CleanPxe	■ OnCleanPxeTimeout(Provision)
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	
FinalizeProvisioning	■ OnFinalizeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitializeProvisioning	■ OnInitializeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnTimeout(Provision)
■ Pre(Provision)	
InstallOS	■ OnInstallOSComplete(Provision)
■ Pre(Provision)	■ OnInstallOSTimeout(Provision)
Reboot	■ OnRebootComplete(Provision)
■ Pre(Provision)	■ OnRebootTimeout(Provision)
SetupOS	■ OnSetupOSComplete(Provision)
■ Pre(Provision)	■ OnSetupOSTimeout(Provision)
SetupPxe	■ OnSetupPxeTimeout(Provision)
■ Pre(Provision)	

## 物理机 SCCM 置备 workflow

状态（主题）	事件（主题）
CheckHardwareType	
■ Pre(Provision)	
Complete	■ OnCompleteProvisioningComplete(Provision)
■ Pre(Provision)	■ OnCompleteProvisioningTimeout(Provision)
FailedProvisioning	■ OnFailedProvisioningTimeout(Provision)
■ Pre(Provision)	
FinalizeProvisioning	■ OnFinalizeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitializeProvisioning	■ OnInitializeProvisioningTimeout(Provision)
■ Pre(Provision)	

状态（主题）	事件（主题）
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
SccmRegistration ■ Pre(Provision)	■ OnSccmRegistrationTimeout(Provision)

### 物理机 SCCM PXE 置备 workflow

状态（主题）	事件（主题）
CheckHardwareType ■ Pre(Provision)	
CleanPxe ■ Pre(Provision)	■ OnCleanPxeTimeout(Provision)
Complete ■ Pre(Provision)	■ OnCompleteProvisioningComplete(Provision) ■ OnCompleteProvisioningTimeout(Provision)
Disposing ■ Pre(Provision)	
FailedProvisioning ■ Pre(Provision)	■ OnFailedProvisioningTimeout(Provision)
FinalizeProvisioning ■ Pre(Provision)	■ OnFinalizeProvisioningTimeout(Provision)
InitializeProvisioning ■ Pre(Provision)	■ OnInitializeProvisioningTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
SccmRegistration ■ Pre(Provision)	■ OnSccmRegistrationTimeout(Provision)
SetupPxe ■ Pre(Provision)	■ OnSetupPxeTimeout(Provision)

### vApp 克隆 workflow

状态（主题）	事件 [触发字符串]（主题）
Global	■ OnFailProvisioning(Provision) ■ OnMasterProvisioned(Provision)
BuildComplete ■ Pre(Provision)	
CloneMachine ■ Pre(Provision)	■ OnCloneMachineComplete(Provision) ■ OnCloneMachineTimeout(Provision)
CustomizeMachine ■ Pre(Provision)	■ OnCustomizeMachineComplete(Provision) ■ OnCustomizeMachineTimeout(Provision)



状态（主题）	事件 [触发字符串]（主题）
CustomizeOS	<ul style="list-style-type: none"> <li>■ OnCustomizeOS(Provision)</li> <li>■ OnCustomizeOSComplete(Provision)</li> <li>■ OnCustomizeOSTimeout(Provision)</li> </ul>
FailedProvisioning	<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>
FinalizeProvisioning	<ul style="list-style-type: none"> <li>■ OnFinalizeComplete(Provision)</li> <li>■ OnFinalizeTimeout(Provision)</li> </ul>
InitialPowerOn	<ul style="list-style-type: none"> <li>■ OnInitialPowerOnComplete(Provision)</li> <li>■ OnInitialPowerOnTimeout(Provision)</li> </ul>
WaitingForMaster	<ul style="list-style-type: none"> <li>■ OnWaitingForMasterTimeout(Provision)</li> </ul>
<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>	

### 虚拟机 SCCM 置备 workflow

状态（主题）	事件（主题）
AddingDisks	<ul style="list-style-type: none"> <li>■ OnAddingDisksComplete(Provision)</li> <li>■ OnAddingDisksTimeout(Provision)</li> </ul>
BuildComplete	<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>
CreatingMachine	<ul style="list-style-type: none"> <li>■ CreatingMachineComplete(Provision)</li> <li>■ OnCreatingMachineTimeout(Provision)</li> </ul>
Disposing	<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>
EjectingCD	<ul style="list-style-type: none"> <li>■ OnEjectingCDComplete(Provision)</li> <li>■ OnEjectingCDTimeout(Provision)</li> </ul>
FailedProvisioning	<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>
InitialPowerOn	<ul style="list-style-type: none"> <li>■ OnInitialPowerOnComplete(Provision)</li> <li>■ OnPowerOnTimeout(Provision)</li> </ul>
InstallingOS	<ul style="list-style-type: none"> <li>■ OnInstallingOSComplete(Provision)</li> <li>■ OnInstallingOSTimeout(Provision)</li> </ul>
SccmRegistration	<ul style="list-style-type: none"> <li>■ OnSccmRegistrationTimeout(Provision)</li> </ul>
<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>	

### WIM 映像 workflow

状态（主题）	事件（主题）
AddingDisks	<ul style="list-style-type: none"> <li>■ OnAddingDisksComplete(Provision)</li> <li>■ OnAddingDisksTimeout(Provision)</li> </ul>
BuildComplete	<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>

状态（主题）	事件（主题）
CreatingMachine ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnCreatingMachineComplete(Provision)</li> <li>■ OnCreatingMachineTimeout(Provision)</li> </ul>
EjectingCD ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnEjectingCDComplete(Provision)</li> <li>■ OnEjectingCDTimeout(Provision)</li> </ul>
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnInitialPowerOnComplete(Provision)</li> <li>■ OnInitialPowerOnTimeout(Provision)</li> </ul>
InstallOS ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ onInstallOSComplete(Provision)</li> <li>■ OnInstallOSTimeout(Provision)</li> </ul>
Reboot ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnRebootComplete(Provision)</li> <li>■ OnRebootTimeout(Provision)</li> </ul>
SetupOS ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnSetupOSComplete(Provision)</li> <li>■ OnSetupOSTimeout(Provision)</li> </ul>

## 配置状态和事件的超时值

所有状态和事件的默认超时值均为 30 分钟。该超时值在 vRealize Automation 全局设置中配置。有些工作流可能需要更长时间才能成功运行。要在环境中容纳不同工作流，您可以为各个工作流或状态添加超时替代值。

要修改默认超时值，请选择**基础架构 > 管理 > 全局设置**，然后编辑**可扩展性生命周期消息超时**的值。更改全局设置后，必须重新启动 Manager Service。

要配置个别超时值，请将工作流或事件属性添加到 IaaS 服务器上 ManagerService.exe.config 文件中的 appSetting 部分。该文件通常位于 %System-Drive%\Program Files x86\VMware\VCAC\Server 目录中。您应在编辑该文件之前，始终先复制该文件。更改每个设置后，必须重新启动 Manager Service。

这些项的基本格式类似于以下示例。

- 对于工作流：Extensibility.{workflow}.Timeout
- 对于事件：Extensibility.{workflow}.{state}.EVENT.{event}.Timeout
- 对于状态：Extensibility.{workflow}.{state}.(PRE/POST).Timeout

向 appSetting 部分添加项时，请以下文为例。超时值的格式为 D.HH:mm:ss.ms。其中 D 表示日，ms 表示毫秒。日和毫秒是可选的。小时、分钟和秒是必需的。

- 要将整个 BasicVmWorkflow 工作流的超时设置为 30 分钟，请添加 <add key="Extensibility.BasicVmWorkflow.Timeout" value="00:30:00"/>。
- 要将 VMPSMasterWorkflow32 中的 OnFinalizeMachine 全局事件的超时设置为两小时，请添加 <add key="Extensibility.VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnFinalizeMachine.Timeout" value="02:00:00"/>。

- 要将 VMPSMasterWorkflow32 的请求前状态的超时设置为 2 天，请添加 `<add key="Extensibility.VMPSMasterWorkflow32.Requested.PRE.Timeout" value="2.00:00:00"/>`。

## 配置状态和事件的错误行为

workflows 订阅超时和错误处理具有默认行为。您可以为您的环境中的计算机自定义该行为。

IaaS 可应对事件代理服务的事件超时和错误处理问题。

每次状态转换时，SendEBSMessage 会向事件代理服务发送事件并等待回复。默认情况下，如果事件代理服务报告可能发生超时或错误，系统会记录该超时或错误，并且工作流将继续运行。

如果在主工作流处于以下状态期间发生超时或错误，系统会强制该工作流进入错误状态，而不会继续运行该工作流。

**表 1-16. 出现以下异常时，工作流不会继续运行**

发生错误时所处的状态	错误状态
PRE MachineProvisioned	UnprovisionMachine
PRE BuildingMachine	Disposing
PRE RegisterMachine	Finalized

要自定义超时或错误行为，您可以针对任何事件或状态向计算机添加自定义属性，目的是在发生该事件或处于该状态时触发一个事件或强制进行状态转换。请以下文为例配置自定义属性。

- `Extensibility.Lifecycle.Error.Event.{Workflow}.{State}`。属性的值是出现超时或错误时要在工作流中触发的事件的名称。
- `Extensibility.Lifecycle.Error.State.{Workflow}.{State}`。属性的值是工作流在出现超时或错误时将强制转换到的状态的名称。

## 场景：生成虚拟机的置备后快照

作为租户管理员，您可能希望服务目录用户拥有其虚拟机的置备后快照，以便可以恢复为全新虚拟机而不必请求新的虚拟机。

### 步骤

#### 1 场景：创建用于置备后快照操作的 vRealize Orchestrator 工作流

您可以创建用于接收必需的输入参数的 vRealize Orchestrator 工作流。对工作流进行设计以实现置备后目标。

#### 2 场景：创建置备后快照工作流订阅

作为租户管理员，您可能希望在创建每个虚拟机后创建其快照。您可以基于计算机置备事件主题配置一个工作流订阅，然后发布该工作流订阅以使其处于活动状态。

### 场景：创建用于置备后快照操作的 vRealize Orchestrator 工作流

您可以创建用于接收必需的输入参数的 vRealize Orchestrator 工作流。对工作流进行设计以实现置备后目标。

有关创建 vRealize Orchestrator 文件夹和工作流的信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

#### 前提条件

使用允许您创建工作流的特权登录到为 vRealize Automation 配置的 vRealize Orchestrator 实例。

#### 步骤

1 为工作流库中的工作流订阅工作流创建文件夹。

2 创建新的工作流。

对于此场景，请将工作流命名为 **Automation Post-Provisioning Snapshot**。

3 添加以下输入参数。

名称	类型
payload	Properties

4 添加能够接收输入参数并创建虚拟机快照的脚本任务。

5 保存工作流。

#### 后续步骤

创建运行 Automation Post-Provisioning Snapshot 工作流的工作流订阅。[场景：创建置备后快照工作流订阅](#)。

### 场景：创建置备后快照工作流订阅

作为租户管理员，您可能希望在创建每个虚拟机后创建其快照。您可以基于计算机置备事件主题配置一个工作流订阅，然后发布该工作流订阅以使其处于活动状态。

您可以配置工作流订阅，使其在虚拟机已置备且检测到的事件消息处于已激活状态时运行创建快照工作流。

#### 前提条件

- 以**租户管理员**身份登录到 vRealize Automation。
- 将 vCenter Server 插件配置为 vRealize Orchestrator 端点。请参见《配置 vRealize Automation》。
- 验证您是否有 vSphere 虚拟机蓝图。
- 验证您是否具有可用于创建虚拟机快照的 vRealize Orchestrator 工作流。不能使用 vRealize Automation 插件提供的“创建快照”工作流。所提供的快照工作流特定于 XaaS 集成。请参见[为置备和生命周期工作流配置 vRealize Orchestrator 工作流](#)。

**步骤**

- 1 选择**管理 > 事件 > 订阅**
- 2 单击**新建**图标 (+)。
- 3 选择**计算机置备**。
- 4 单击**下一步**。
- 5 在“条件”选项卡中，配置触发条件。
  - a 选择**根据条件运行**。
  - b 从子句下拉菜单中，选择**以下所有项**。
  - c 配置以下条件：

属性	运算符	值
数据 > 计算机 > 计算机类型	等于	常数 > 虚拟机
数据 > 生命周期状态 > 生命周期状态名称	等于	常数 > VMPSMasterWorkflow32.MachineActivated
数据 > 生命周期状态 > 状态阶段	等于	常数 > POST

- d 单击**下一步**。
- 6 在“工作流”选项卡中，浏览 Orchestrator 树，然后选择 **Automation Post-Provisioning Snapshot** 工作流。
- 7 单击**下一步**。
- 8 在“详细信息”选项卡中，输入**名称**和**描述**。  
 在此场景中，请输入 **Post-Provisioning Virtual Machine Snapshot** 作为名称，并输入 **Create a snapshot when new virtual machine is provisioned and activated** 作为描述。
- 9 单击**完成**。
- 10 选择 Post-Provisioning Virtual Machine Snapshot 一行，然后单击**发布**。

**结果**

该工作流订阅现已处于活动状态，而且将在事件消息指示请求的虚拟机已置备并且已激活时触发快照工作流。

**后续步骤**

要测试该工作流订阅，可以请求服务目录中的虚拟机。请求指示置备成功后，请验证是否已创建快照。

## 处理批准 workflow 订阅

您可以创建批准前 workflow 订阅和批准后 workflow 订阅，以便将批准请求发送给外部系统进行处理。然后，响应（批准或拒绝）将由 vRealize Automation 进行处理。

### 批准服务集成

您可以创建运行自定义 vRealize Orchestrator 工作流的批准前 workflow 订阅或批准后 workflow 订阅，此 workflow 在 vRealize Automation 外部的系统中处理批准请求。

在批准策略批准级别，可以选择**使用事件订阅**作为审批者。此级别可以是批准策略中的级别之一。当服务目录用户请求批准策略应用到的项目而该批准策略包含**使用事件订阅**审批者时，批准服务将向事件代理服务发送消息并产生以下结果。

- 如果有已发布的 workflow 订阅且与条件匹配，vRealize Orchestrator 将运行您的批准 workflow 或拒绝 workflow。
- 如果有已发布的 workflow 订阅但与条件不匹配，您可以取消发布该 workflow 订阅，或者如果没有已发布的订阅，则批准级别是已批准，并且批准过程将进入下一个批准级别。

批准 workflow 订阅将收到来自批准服务的消息，并将这些消息与已配置的批准订阅条件相比较。如果找到与该条件匹配的消息，选定的 vRealize Orchestrator 工作流将开始运行。已发布的事件数据会作为输入传递到该工作流，并以该工作流中指定的方法进行处理。工作流的结果会返回到 vRealize Automation，并且请求会得到处理。如果处理结果是批准，则会评估下一个批准级别。如果处理结果是拒绝，则请求会被拒绝。如果批准服务未在 24 小时（这是批准服务的默认超时）内收到回复，则对请求的处理结果是拒绝。

### 配置适用于批准事件主题的 vRealize Orchestrator 工作流

必须对自定义 vRealize Orchestrator 工作流进行配置，以使其支持批准消息并使用 vRealize Automation 可以处理的正确格式信息进行回复。

#### 批准事件主题架构

批准前和批准后事件消息架构包含字段名称和值以及包含在请求中的信息和有关请求源的信息。

以下是事件数据负载的结构。

```
{
  fieldNames : Properties,           // Property names

  fieldValues : Properties,          // Property values

  // Information about the request
  requestInfo : {
    requestRef : STRING,              // Identifier for the source request
    itemName : STRING,               // Name of the requested item
    itemDescription : STRING,         // Description of the requested item
    reason : STRING,                 // Justification provided by the user specifying why the
request is required
    description : STRING,             // Description entered by the user specifying the purpose of
the request
    approvalLevel:ExternalReference, // Approval level ID. This is a searchable field
    approvalLevelName : STRING,      // Approval level name
  }
}
```

```

        createDate : DATE_TIME,           // Time the approval request is created
        requestedFor : STRING,           // Principal id of the user for whom the source request is
initiated
        subtenantId : STRING,           // Business group id
        requestedBy : STRING           // Principal id of the user who actually submits the request
    },

    // Information about the source of the request
    sourceInfo : {
        externalInstanceId : STRING,     // Identifier of the source object, as defined by the
initiator service
        serviceId : STRING,             // Identifier of the service which initiated the approval
        externalClassId : STRING        // Identifier of the class to which the source object belongs
    }
}

```

属性名称和属性值可以在批准策略级别配置的自定义属性或系统属性。在批准策略中配置这些属性可允许审批者在批准过程中更改值。例如，如果包含了 CPU，则审批者可以在批准请求表单中减少 CPU 数量。

回复事件数据负载是由工作流返回到 vRealize Automation 的信息。回复负载的内容决定了请求获得批准还是被拒绝。

```

{
    approved : BOOLEAN,

    // Property values
    fieldValues : Properties
}

```

回复事件负载中的 `approved` 参数为 `true`（表示请求获得批准）或 `false`（表示请求被拒绝）。属性值是已由 vRealize Orchestrator 工作流修改并返回到 vRealize Automation 的自定义属性或系统属性，并会包含在批准过程中。

最佳做法是为 vRealize Orchestrator 工作流配置 `businessJustification` 的输出参数。使用此参数可将审批者在外部系统中提供的备注传递到 vRealize Automation 批准过程。这些备注可供确定是批准还是拒绝。

### 基于批准架构创建 vRealize Orchestrator 工作流

所创建的自定义批准工作流必须具有输入参数（可以使用任何有用名称），且其类型必须配置为 `Properties`。触发并运行工作流订阅时，批准事件数据负载将放置在此参数中。

发送回 vRealize Automation 的工作流输出参数是 `approved : Boolean` 和 `fieldValues : Properties`。返回的 `approved : Boolean` 参数决定了批准级别是已批准还是已拒绝。`fieldValues : Properties` 参数包含已在外部系统中修改的值。

## 场景：将软件请求发送到外部系统进行批准

作为租户管理员，您可能希望 vRealize Automation 外部的用户能在服务目录用户请求包含某个软件的计算机时对该软件组件进行批准。您可以配置一个批准策略，要求所有软件置备和工作流订阅都要获得批准。该策略可配置为在收到与已定义的条件匹配的批准前消息时运行。

### 步骤

#### 1 场景：创建用于批准工作流订阅的 vRealize Orchestrator 工作流

创建执行如下操作的 vRealize Orchestrator 工作流：从 vRealize Automation 接收必需的批准输入参数，并返回完成批准过程所需的输出参数。

#### 2 场景：创建用于外部批准的批准策略

作为租户管理员，您可以创建一个批准策略，生成由批准服务发布的事件消息。如果已为工作流订阅配置了与该事件消息匹配的条件，vRealize Orchestrator 将运行选定的工作流。

#### 3 场景：创建批准前工作流订阅

作为租户管理员，您可以创建批准前工作流订阅，此工作流订阅将在服务目录请求生成与已配置的条件匹配的批准请求时运行 vRealize Orchestrator 工作流。

### 场景：创建用于批准工作流订阅的 vRealize Orchestrator 工作流

创建执行如下操作的 vRealize Orchestrator 工作流：从 vRealize Automation 接收必需的批准输入参数，并返回完成批准过程所需的输出参数。

必须对工作流进行设计以实现批准目标。有关创建 vRealize Orchestrator 文件夹和工作流的信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

### 前提条件

使用允许您创建工作流的特权登录到为 vRealize Automation 配置的 vRealize Orchestrator 实例。

### 步骤

#### 1 为工作流库中的工作流订阅工作流创建文件夹。

#### 2 创建新的工作流。

对于此场景，请将工作流命名为 **Automation Approval Request**。

##### a 添加以下输入参数。

名称	类型
input	Properties

##### b 添加以下输出参数。

名称	类型
approved	boolean
fieldValues	Properties

#### 3 创建能够处理输入参数和输出参数的脚本任务。



#### 4 保存工作流。

##### 后续步骤

创建使用工作流订阅作为审批者的批准策略。[场景：创建用于外部批准的批准策略](#)

##### 场景：创建用于外部批准的批准策略

作为租户管理员，您可以创建一个批准策略，生成由批准服务发布的事件消息。如果已为工作流订阅配置了与该事件消息匹配的条件，vRealize Orchestrator 将运行选定的工作流。

##### 前提条件

- 以**租户管理员**或**批准管理员**身份登录到 vRealize Automation。

##### 步骤

- 1 选择**管理 > 批准策略**。
- 2 创建用于软件组件的批准策略。
  - a 单击**新建**图标 (+)。
  - b 选择**选择批准策略类型**。
  - c 在列表中，选择**服务目录 - 目录项请求 - 软件组件**。
  - d 单击**确定**。
  - e 配置以下选项：

选项	配置
名称	输入 <b>Software external approval</b> 。
描述	输入 <b>Approval request sent to external approval system</b> 。
状态	选择 <b>Active</b> 。

- 3 在**预批准**选项卡上，单击**添加**图标 (+)。
- 4 使用触发条件和批准操作配置**级别信息**选项卡。
  - a 在**名称**文本框中，输入 **External level for software**。
  - b 在**描述**文本框中，输入 **Software approval request sent to external approval system**。
  - c 选择**始终需要**。
  - d 选择**使用事件订阅**。
- 5 单击**确定**。

##### 后续步骤

- 创建批准前工作流订阅，使其能根据已配置的批准级别接收事件消息。请参见[场景：创建批准前工作流订阅](#)。

- 将批准策略应用到授权中的软件组件。请参见《配置 vRealize Automation》。

### 场景：创建批准前工作流订阅

作为租户管理员，您可以创建批准前工作流订阅，此工作流订阅将在服务目录请求生成与已配置的条件匹配的批准请求时运行 vRealize Orchestrator 工作流。

#### 前提条件

- 以**租户管理员**身份登录到 vRealize Automation。
- 配置名为 **External level for software** 的批准策略级别。请参见[场景：创建用于外部批准的批准策略](#)。
- 创建用于向外部系统发送请求的自定义 vRealize Orchestrator 工作流。在此场景中，请使用 **Automation Approval Request** 工作流。

#### 步骤

1 选择**管理 > 事件 > 订阅**

2 单击**新建**图标 (+)。

3 单击**批准前**。

4 单击**下一步**。

5 在**条件**选项卡中，配置触发条件。

a 选择**根据条件运行**。

b 从子句下拉菜单中，配置以下条件：

属性	运算符	值
数据 > 有关请求的信息 > 批准级别名称	等于	External level for software

c 单击**下一步**。

6 在“工作流”选项卡中，浏览 Orchestrator 树，然后选择 **Automation Approval Request** 工作流。

7 单击**下一步**。

8 在“详细信息”选项卡中，输入名称和描述。

在此场景中，请输入 **Software External** 作为名称，并输入 **Sends approval request to external system** 作为描述。

9 在**超时 (分钟)** 文本框中，输入 120。

应根据目标系统来指定订阅工作流超时前的时间。如果 vRealize Automation 未在指定分钟数内处理来自目标系统的回复，系统会自动拒绝该请求。

如果未提供任何值，则默认超时为 24 小时。

10 单击**完成**。

**11** 选择 **Software External** 一行，然后单击**发布**。

## 结果

现在，**Software External** 批准前事件订阅将接收批准前事件消息。

## 后续步骤

- 如果已将批准策略应用到活动授权中的软件组件，则应请求服务目录中的项，并验证批准策略和工作流订阅是否按预期工作。

## 工作流订阅故障排除

工作流订阅故障排除包括一些常见问题。您可能还需要检查各个日志。

- **对未启动的 vRealize Orchestrator 工作流进行故障排除**  
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。
- **对用时太长的置备请求进行故障排除**  
置备 IaaS 计算机需要十小时或更长时间。
- **对没有针对批准请求运行的 vRealize Orchestrator 工作流进行故障排除**  
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计算机时，该工作流未运行。
- **对本应获得批准却被拒绝的批准请求进行故障排除**  
已将批准前或批准后工作流订阅配置为运行指定的 vRealize Orchestrator 工作流，但请求被拒绝，而您认为该请求已获得批准。
- **对被拒绝的批准请求进行故障排除**  
已将批准前或批准后工作流订阅配置为运行指定的 vRealize Orchestrator 工作流，但请求意外被拒绝。

### 对未启动的 vRealize Orchestrator 工作流进行故障排除

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。

#### 解决方案

- 1 验证已发布的工作流订阅。
- 2 验证工作流订阅的条件是否正确配置。
- 3 验证 vRealize Orchestrator 服务器是否具有指定的工作流。

### 对用时太长的置备请求进行故障排除

置备 IaaS 计算机需要十小时或更长时间。

#### 原因

如果将工作流订阅配置为在置备状态期间触发，则环境中可能会运行两个 IaaS Manager Service 实例。

## 解决方案

- ◆ 确认只有一个 IaaS Manager Service 实例处于活动状态。如果多个实例处于活动状态，还会在日志中看到类似以下内容的错误：

```
[EventBrokerService] Failed resuming workflow b6e9276a-f20f-40f1-99ad-6d9524560cc2 on queue
3679fa71-ac2a-42d5-8626-f98ea096f0d3. System.Workflow.Runtime.QueueException: Event Queue
operation failed with MessageQueueErrorCode QueueNotFound for queue '3679fa71-ac2a-42d5-8626-
f98ea096f0d3'. at System.Workflow.Runtime.WorkflowQueuingService.EnqueueEvent(IComparable
queueName, Object item) at System.Workflow.Runtime.WorkflowExecutor.EnqueueItem(IComparable
queueName, Object item, IPendingWork pendingWork, Object workItem) at
System.Workflow.Runtime.WorkflowInstance.EnqueueItem(IComparable queueName, Object item,
IPendingWork pendingWork, Object workItem) at
DynamicOps.VMPS.Service.Workflow.Services.EventBrokerService.OnMessage(EventObject obj)
[UTC:2015-11-14 07:14:25 Local:2015-11-13 23:14:25] [Error]: Thread-Id="15" - context="HKBsp6Tt"
token="JeuTG7ru" [EventBrokerClient] Invoking subscription callback failed: Event Queue operation
failed with MessageQueueErrorCode QueueNotFound for queue '3679fa71-ac2a-42d5-8626-f98ea096f0d3'.
```

## 对没有针对批准请求运行的 vRealize Orchestrator 工作流进行故障排除

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。

### 原因

要成功对批准运行工作流订阅，您必须确保所有组件均已正确配置。

### 解决方案

- 1 验证批准策略是否处于活动状态，并验证是否为该策略中的批准级别选择了使用事件订阅。
- 2 验证批准策略在您的授权中是否正确应用。
- 3 验证工作流订阅是否正确配置并正确发布。
- 4 检查事件日志以查看与批准相关的消息。

## 对本应获得批准却被拒绝的批准请求进行故障排除

已将批准前或批准后工作流订阅配置为运行指定的 vRealize Orchestrator 工作流，但请求被拒绝，而您认为该请求已获得批准。

### 解决方案

- 1 在 vRealize Orchestrator 中查看工作流。
  - a 使用管理员特权登录到 vRealize Orchestrator。
  - b 验证工作流是否正常运行而未发生错误。
  - c 验证是否为 approval 和 fieldValues 参数返回了预期值。

## 2 在 vRealize Automation 中查看请求。

- a 以请求了被拒绝的项的用户身份登录到 vRealize Automation。
- b 单击**部署**选项卡。
- c 单击被拒绝的请求部署名称，然后单击**历史记录**。
- d 单击状态，并查看标志以了解更多信息。

如果发生了错误，有关该错误的信息将显示为理由数据。

## 对被拒绝的批准请求进行故障排除

已将批准前或批准后工作流订阅配置为运行指定的 vRealize Orchestrator 工作流，但请求意外被拒绝。

### 问题

所有在此外部批准级别之前的批准级别都已获得批准，此级别本应获得批准，但处理结果却是拒绝。

### 原因

一个可能的原因是在 vRealize Orchestrator 尝试运行该工作流时发生内部错误。例如，该工作流缺失或 vRealize Orchestrator 服务器未在运行。

### 解决方案

- 1 选择**管理 > 事件 > 事件日志**。
- 2 检查日志以查看与批准相关的消息。

## 使用 vRealize Automation Designer 扩展计算机生命周期

可以将自定义逻辑注入到 IaaS 计算机生命周期的预定阶段，方法是使用 vRealize Automation Designer 直接编辑状态更改工作流存根并（可选）调用自定义 vRealize Orchestrator 工作流。

**注** 工作流存根被事件代理工作流订阅所取代。目前仍会提供工作流存根，它们仍受支持且可以使用，但预计会在将来的 vRealize Automation 版本中将其移除。为确保将来的产品兼容性，您应该使用工作流订阅，以基于状态更改运行自定义工作流。请参见[配置工作流订阅以扩展 vRealize Automation](#)。

## 使用 vRealize Automation Designer 检查表扩展计算机生命周期

“使用 vRealize Automation Designer 检查表扩展计算机生命周期”从较高层面概述了安装和配置 vRealize Automation Designer 以自定义 IaaS 计算机生命周期所需的步骤。

**表 1-17. 使用 vRealize Automation Designer 检查表扩展计算机生命周期**

任务	详细信息
<input type="checkbox"/> 下载和安装 vRealize Automation Designer。	<a href="#">安装 vRealize Automation Designer</a>
<input type="checkbox"/> 为 vRealize Orchestrator 实例创建 vRealize Automation 端点。	<a href="#">创建 vRealize Orchestrator 端点</a>

表 1-17. 使用 vRealize Automation Designer 检查表扩展计算机生命周期（续）

任务	详细信息
<input type="checkbox"/> 将 vRealize Orchestrator 端点与计算机蓝图相关联。	将 <a href="#">vRealize Orchestrator 端点与蓝图相关联</a>
<input type="checkbox"/> 使用 vRealize Automation Designer 活动来自定义 IaaS 工作流存根。	自定义 <a href="#">IaaS 工作流</a>
<p><b>注</b> 工作流存根被事件代理工作流订阅所取代。目前仍会提供工作流存根，它们仍受支持且可以使用，但预计会在将来的 vRealize Automation 版本中将其移除。为确保将来的产品兼容性，您应该使用工作流订阅，以基于状态更改运行自定义工作流。请参见<a href="#">配置工作流订阅以扩展 vRealize Automation</a>。</p> <p>（可选）可以使用 vRealize Orchestrator 工作流活动来调用自定义 vRealize Orchestrator 工作流。</p>	
<input type="checkbox"/> 创建自定义状态更改工作流后，租户管理员或业务组主管必须通过添加自定义属性来为特定蓝图启用该工作流。	<a href="#">配置蓝图以调用状态更改工作流</a>

## 安装和配置 vRealize Automation Designer

可以在 Windows 计算机上安装 vRealize Automation Designer，并将其配置为与远程 Model Manager 实例进行通信。如果使用 IaaS 工作流调用 vRealize Orchestrator 工作流，则还必须在 IaaS 中配置 vRealize Orchestrator 实例。

### 安装 vRealize Automation Designer

可以在 Windows 计算机上安装 vRealize Automation Designer，并将其配置为与远程 Model Manager 实例进行通信。

#### vRealize Automation Designer 必备条件

vRealize Automation Designer 通常安装在开发计算机上，而不是安装在服务器上。

#### 受支持的操作系统

VMware vRealize Automation 文档页上的《vRealize Automation 支持列表》列出了 vRealize Automation Designer 支持的操作系统。

#### 系统配置要求

有关此信息可能的更新，请参见适用于您的 vRealize Automation 版本的《vRealize Automation 支持列表》。

- 必须已安装 .NET Framework 4.5。
- vRealize Automation Designer 主机必须拥有 IaaS 网站组件（具体来说是 Model Manager Web 组件）的网络访问权限。
- 如果 Model Manager 安装在远程位置，则用于 Model Manager Web 组件的证书在 vRealize Automation Designer 主机上必须受信任。

## 下载 vRealize Automation Designer 安装程序

可以从 vRealize Automation 设备下载 vRealize Automation Designer 安装程序。

### 前提条件

- 以本地管理员身份登录到 Windows 计算机。
- 如果使用的是 Internet Explorer，请验证是否未启用增强的安全配置。请参见 <res://iesetup.dll/SoftAdmin.htm>。

### 步骤

- 1 打开浏览器。
- 2 使用 <https://vra-vd-hostname.domain.name:5480/installer/> 中的主机名导航到 Windows 安装程序下载页面。
- 3 单击 **vRealize Automation Designer**。
- 4 出现提示时，请保存安装程序。

### 后续步骤

[安装 vRealize Automation Designer](#)。

## 安装 vRealize Automation Designer

vRealize Automation Designer 安装程序打包为 Windows 安装向导。

### 前提条件

[下载 vRealize Automation Designer 安装程序](#)。

### 步骤

- 1 导航到安装程序的下载目录。
- 2 右键单击 **DesignCenter-Setup.exe**，然后选择**以管理员身份运行**。
- 3 在**欢迎使用**页面上，单击**下一步**。
- 4 阅读许可协议，选中**我接受许可协议中的条款**，然后单击**下一步**。
- 5 在**自定义安装**页面上，单击**下一步**。
- 6 以 *hostname:port* 格式指定 Model Manager Web 实例的完全限定域名和端口。  
默认端口为 443。
- 7 指定 Model Manager 服务用户凭据。
- 8 单击**下一步**。

安装程序会尝试访问 Model Manager，验证 Model Manager 主机和凭据组合。如果返回错误，则必须提供正确的 Model Manager 主机和凭据组合，然后再继续操作。

- 9 单击**安装**。

## 10 单击完成。

### 后续步骤

可以从 Windows “开始” 菜单导航到安装目录来启动 vRealize Automation Designer。

### 配置 vRealize Orchestrator 端点

如果使用 vRealize Automation 工作流调用 vRealize Orchestrator 工作流，则必须将 vRealize Orchestrator 实例或服务器配置为端点。

您可以将 vRealize Orchestrator 端点与计算机蓝图关联，确保通过该蓝图置备的计算机的所有 vRealize Orchestrator 工作流均在使用该端点运行。

默认情况下，vRealize Automation 包括嵌入式 vRealize Orchestrator 实例。在生产或测试环境中运行 vRealize Automation 工作流或创建概念证明时，建议您使用嵌入式实例作为 vRealize Orchestrator 端点。

您还可以在外部 vRealize Orchestrator 服务器上安装插件，尽管在生产环境中不建议使用此方法。

### vRealize Orchestrator 集成必备条件

如果要使用 vRealize Automation 工作流来运行具有 VC:VirtualMachine 类型输入参数和输出参数的 vRealize Orchestrator 工作流，请确认您具有可在 vRealize Orchestrator 和 IaaS 之间转换虚拟机类型的 vRealize Orchestrator 工作流。

在 vRealize Orchestrator 5.5 和更高版本中，默认情况下，所需的工作流作为 vCenter 插件的一部分提供。

如果使用的是 vRealize Orchestrator 5.1，请安装适用于 vRealize Orchestrator 的 vRealize Automation 集成包。从 vRealize Orchestrator 社区站点下载 com.vmware.library.vcenter.vcac-integration.package 软件包，网址为：<https://communities.vmware.com/t5/vRealize-Orchestrator-Documents/vCloud-Automation-Center-integration-package/ta-p/2777982>。将该软件包导入到设置为 IaaS 中的端点的每个 vRealize Orchestrator 服务器。

有关将软件包导入到 vRealize Orchestrator 的信息，请参阅 vRealize Orchestrator 文档。

### 创建 vRealize Orchestrator 端点

您可以创建 vRealize Orchestrator 端点以连接到 vRealize Orchestrator 服务器。

您可以配置多个端点并让它们连接到不同的 vRealize Orchestrator 服务器，但必须配置每个端点的优先级。

执行 vRealize Orchestrator 工作流时，vRealize Automation 将先尝试使用优先级最高的 vRealize Orchestrator 端点。如果该端点不可访问，则继续尝试下一个优先级最高的端点，直到 vRealize Orchestrator 服务器可用于运行该工作流为止。

### 前提条件

- 以 IaaS 管理员身份登录到 vRealize Automation。



**步骤**

- 1 选择**基础架构 > 端点 > 端点**。
- 2 选择**新建 > Orchestration > vRealize Orchestrator**。
- 3 输入名称和可选描述。
- 4 输入包含 vRealize Orchestrator 服务器的完全限定名称或 IP 地址以及 vRealize Orchestrator 端口号的 URL。

传输协议必须是 HTTPS。如果未指定端口，则使用默认端口 443。

要使用 vRealize Automation 设备中嵌入的默认 vRealize Orchestrator 实例，请键入 **https://vrealize-automation-appliance-hostname:443/vco**。

- 5 在**用户名**和**密码**文本框中提供 vRealize Orchestrator 凭据以连接到 vRealize Orchestrator 端点。

使用的凭据应该对要从 IaaS 调用的所有 vRealize Orchestrator 工作流具有“执行”权限。

要使用 vRealize Automation 设备中嵌入的默认 vRealize Orchestrator 实例，用户名应使用 **administrator@vsphere.local**，且密码应使用配置 SSO 时指定的管理员密码。

- 6 在**优先级**文本框中输入大于或等于 1 的整数。

值越低表示优先级越高。

- 7 （可选）单击**属性**，并为端点添加提供的自定义属性、属性组或您自己的属性定义。

- 8 单击**确定**。

**将 vRealize Orchestrator 端点与蓝图相关联**

可以指定将特定 vRealize Orchestrator 端点与蓝图配合使用。

当 IaaS 为通过此蓝图置备的任何计算机运行 vRealize Orchestrator 工作流时，它始终使用关联的端点。如果该端点不可访问，则工作流将失败。

**前提条件**

以**基础架构架构师**身份登录到 vRealize Automation。

**步骤**

- 1 选择**设计 > 蓝图**。
- 2 创建新蓝图或编辑现有蓝图

如果正在编辑现有蓝图，则您指定的 vRealize Orchestrator 端点仅应用到通过已更新的蓝图置备的新计算机。通过该蓝图置备的现有计算机将继续使用优先级最高的端点，除非您手动将以下属性添加到计算机。

- 3 单击**蓝图属性**图标 (  )。

**4** 单击**属性**选项卡。

a 单击**自定义属性 > 新建**。

b 在**名称**文本框中键入 **VMware.VCenterOrchestrator.EndpointName**。

属性名称区分大小写。

c 单击**确定**保存属性。

**5** 单击**确定**。

## 使用 vRealize Automation Designer 自定义 IaaS 工作流

VMware 提供了许多工作流，您可以使用 vRealize Automation Designer 来自定义这些工作流。这些工作流包括状态更改工作流和菜单操作工作流。

可以使用 Microsoft Windows Workflow Foundation 4（属于 .NET Framework 4 的一部分）来创建 IaaS 工作流。有关 Windows Workflow Foundation 和工作流创建的信息，请参阅 Microsoft 文档。vRealize Automation 还提供多个 vRealize Automation Designer 活动，用于运行和监控 vRealize Orchestrator 工作流。

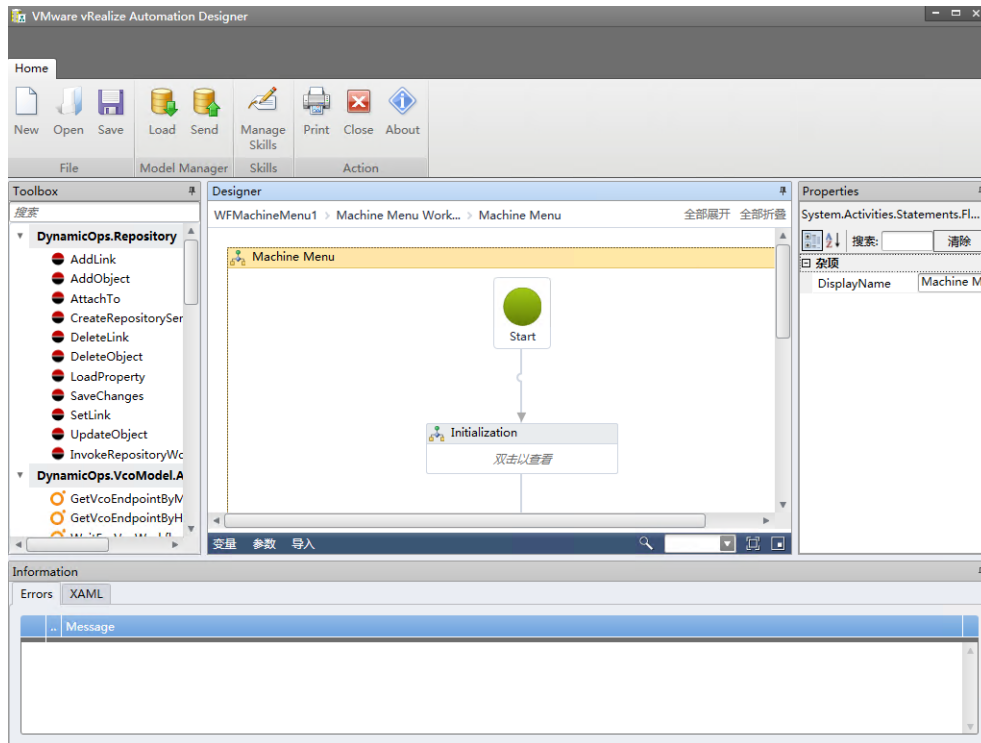
VMware 提供的可自定义工作流模板说明了对具有单独的初始化、自定义逻辑和终止化序列的工作流加以结构化的最佳做法。整个工作流包装在用于错误处理的 TryCatch 块中。执行工作流的 Distributed Execution Manager 会记录任何未捕获的异常或重新抛出的异常。

创建自定义 IaaS 工作流后，蓝图作者必须在特定蓝图上启用该工作流。

## vRealize Automation Designer 控制台

vRealize Automation Designer 控制台提供了用于自定义 IaaS 工作流的可视化工作流编辑器。

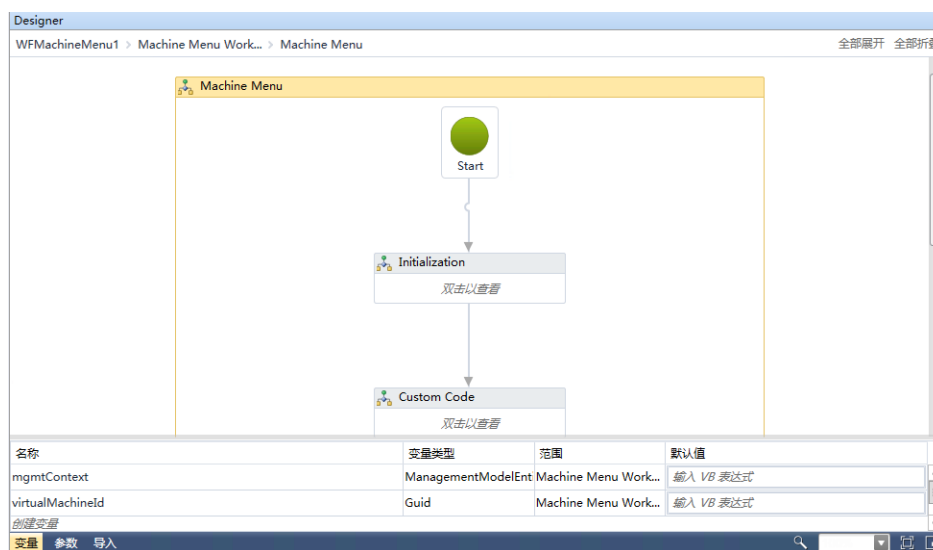
必须拥有 vRealize Automation Designer 主机（通常是开发计算机）上的本地管理员权限才能启动 vRealize Automation Designer 控制台。



左侧的“工具箱”窗格可用于访问 vRealize Automation 工作流活动库。将活动从工具箱拖动到“设计器”窗格即可将它们添加到工作流中。“属性”窗格显示了当前在“设计器”窗格上选择的活动的可配置属性。此界面与 Visual Studio 中的工作流设计器非常相似。

使用“设计器”窗格底部的详细信息选项卡，您可以显示和编辑所选活动范围内的变量或所选活动的参数。

**注** 变量和参数都指定为 Visual Basic 表达式。但是，变量名称不区分大小写，而参数名称区分大小写。有关 IaaS 工作流活动的有效参数的信息，请参见 [vRealize Automation 工作流活动参考](#)。



“导入”选项卡显示了已导入的命名空间，您可以从中选择要添加到工作流的实体类型。

控制台底部的可折叠“信息”窗格显示了配置活动时发生的任何错误，并可访问工作流的 XAML 表示。

## IaaS 工作流类型

使用 vRealize Automation Designer 可以自定义以下两种类型的工作流：状态更改工作流和菜单操作工作流。

- 当主工作流在两种状态之间转换（例如，处于新计算机置备过程中的特定阶段）时，将运行状态更改工作流。
- 当用户从服务目录中的“操作”菜单选择选项或从“基础架构”选项卡中的计算机菜单选择选项时，将运行菜单操作工作流。

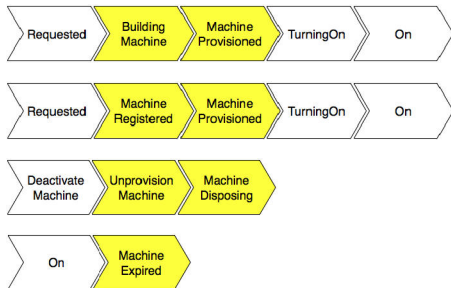
### 状态更改工作流

通过创建状态更改工作流，您可以在 IaaS 主工作流进入特定状态之前运行某工作流。

例如，您可以创建自定义工作流，以与处于计算机生命周期的不同阶段的外部数据库和记录信息集成：

- 创建在主工作流进入 **MachineProvisioned** 状态之前运行的自定义工作流，以记录诸如计算机所有者和审批者等信息。
- 创建在计算机进入 **MachineDisposing** 状态之前运行的自定义工作流，以记录该计算机被销毁的时间和相关数据（例如，该计算机在上次数据收集时的资源利用率和上次登录时间等）。

下图显示了主工作流的主要状态。



vRealize Automation Designer 为每种状态提供了可自定义工作流。

表 1-18. 可自定义状态更改工作流

主工作流状态	可自定义工作流名称
BuildingMachine	WFStubBuildingMachine
Disposing	WFStubMachineDisposing
Expired	WFStubMachineExpired
MachineProvisioned	WFStubMachineProvisioned
RegisterMachine	WFStubMachineRegistered
UnprovisionMachine	WFStubUnprovisionMachine

## 配置状态更改工作流概览

可以使用 vRealize Automation Designer 来自定义状态更改工作流。然后，蓝图作者可以为特定蓝图启用该工作流。

下文从较高层面概述了启用状态更改工作流所需的步骤：

- 1 workflows 开发人员使用 vRealize Automation Designer 自定义状态更改工作流模板之一。请参见 [自定义 IaaS 工作流](#)。

任何 IaaS 工作流都可以调用 vRealize Orchestrator 工作流。有关详细信息，请参见 [使用 vRealize Orchestrator 工作流活动](#)。

- 2 租户管理员或业务组主管配置蓝图，为通过该蓝图置备的计算机调用自定义工作流。请参见 [配置蓝图以调用状态更改工作流](#)。

## 菜单操作工作流

当用户在服务目录中从“操作”菜单中选择选项或在“基础架构”选项卡中从计算机菜单中选择选项时，将执行菜单操作工作流。

例如，您可以创建一个自定义工作流，以使用户能够通过从计算机菜单中选择“提出支持问题”来创建与计算机相关的支持票证。

vRealize Automation Designer 提供了用于自定义菜单操作工作流的模板。

除了工作流定义，菜单操作工作流还取决于操作配置文件。操作配置文件定义了自定义菜单选项的各个方面，例如，显示文本、哪些角色有权访问以及操作可供其使用的计算机状态。

---

**注** XaaS 架构师可以使用 XaaS 为任何目录项定义自定义操作。为并非通过 vSphere 或 vCloud Director 置备的 IaaS 计算机创建自定义操作时，需要使用 vRealize Automation 6.1 或更高版本。

---

## 配置菜单操作工作流概览

可以使用 vRealize Automation Designer 和 CloudUtil 命令行实用程序来自定义菜单操作工作流。然后，蓝图作者可以为特定蓝图启用该工作流。

下文从较高层面概述了启用菜单操作工作流所需的步骤：

- 1 workflows 开发人员使用 vRealize Automation Designer 自定义菜单操作工作流模板之一。请参见 [自定义 IaaS 工作流](#)。

任何 IaaS 工作流都可以调用 vRealize Orchestrator 工作流。有关详细信息，请参见 [使用 vRealize Orchestrator 工作流活动](#)。

- 2 workflows 开发人员在 Model Manager 中配置菜单操作。请参见 [配置菜单操作](#)。

- 3 workflows 开发人员向服务目录注册新的菜单操作。请参见 [向服务目录注册新的菜单操作](#)。

- 4 租户管理员或业务组主管配置蓝图，为通过该蓝图置备的计算机启用此菜单操作。请参见 [配置蓝图以启用菜单操作工作流](#)。

如果该菜单操作设计用于在服务目录中使用，则还必须将其授权给用户。有关详细信息，请参见《租户管理》。

## 自定义 IaaS 工作流

使用 vRealize Automation Designer，您可以在 Model Manager 中编辑可自定义工作流和更新工作流。

### 前提条件

启动 vRealize Automation Designer。

### 步骤

- 1 单击**加载**。
- 2 选择要自定义的工作流。

选项	描述
<b>WFMenuN</b>	可自定义菜单操作工作流
<b>WFStubBuildingMachine</b>	可自定义状态更改工作流，在计算机进入 BuildingMachine 状态之前执行
<b>WFStubMachineDisposing</b>	可自定义状态更改工作流，在计算机进入 Disposing 状态之前执行
<b>WFStubMachineExpired</b>	可自定义状态更改工作流，在计算机进入 Expired 状态之前执行
<b>WFStubMachineProvisioned</b>	可自定义状态更改工作流，在计算机进入 MachineProvisioned 状态之前执行
<b>WFStubMachineRegistered</b>	可自定义状态更改工作流，在计算机进入 RegisterMachine 状态之前执行
<b>WFStubUnprovisionMachine</b>	可自定义状态更改工作流，在计算机进入 UnprovisionMachine 状态之前执行

- 3 单击**确定**。

工作流将显示在“设计器”窗格中。

- 4 要自定义工作流，请将活动从工具箱拖动到“设计器”窗格中并配置其参数。
- 5 完成对工作流的编辑后，单击**发送**以在 Model Manager 中更新该工作流。

系统将保存该工作流。下次加载工作流时，该工作流将作为新修订版本显示在列表中。您可以随时访问工作流的早期版本。请参见[恢复为工作流的先前修订版本](#)。

## 使用 vRealize Orchestrator 工作流活动

可以使用 vRealize Automation Designer 活动来以同步或异步方式调用 vRealize Orchestrator 工作流。

可以通过以下方法之一指定 vRealize Orchestrator 端点：

- **VirtualMachineId** 是表示虚拟机 ID 的变量名称。将选择具有此 ID 的虚拟机，并使用从虚拟机的 **VMware.VCenterOrchestrator.EndpointName** 自定义属性中检索到的值作为 vRealize Orchestrator 端点名称。
- **GetVcoEndpointByManagementEndpoint** 将返回指定 **ManagementEndpoint** 对象上的自定义属性的值。如果未指定 **CustomPropertyName**，将使用 **VMware.VCenterOrchestrator.EndpointName** 属性的值。
- **GetVcoEndpointByHost** 将返回指定主机上的自定义属性的值。如果未指定 **CustomPropertyName**，将使用 **VMware.VCenterOrchestrator.EndpointName** 属性的值。

## 同步执行

`InvokeVcoWorkflow` 活动将调用 `vRealize Orchestrator` 工作流，并阻止父 `IaaS` 工作流继续运行，直至 `vRealize Orchestrator` 工作流完成。此活动将返回 `vRealize Orchestrator` 工作流的输出参数。

此外，同步运行支持以下属性：

- `WorkflowTimeout` 是以秒为单位的超时值。如果 `vRealize Orchestrator` 工作流未在指定时间内完成，则会生成异常，而不会阻止该工作流直至返回响应。如果未定义任何值或提供了零值，则不会激活超时。在该时间段内，系统每 10 秒会检查一次工作流状态，除非通过在 `VMware.VCenterOrchestrator.PollingInterval` 自定义属性中指定一个值来修改端点的轮询时间。

## 异步运行工作流

`InvokeVcoWorkflowAsync` 活动会调用 `vRealize Orchestrator` 工作流并继续运行 `IaaS` 工作流中的活动，而不等待 `vRealize Orchestrator` 工作流完成。

如果对 `vRealize Orchestrator` 服务器的 REST API 调用失败（例如，如果无法访问该服务器），此活动将返回可用于监控工作流的唯一工作流令牌，或者返回错误。

此外，可以将以下两个活动与此活动一起使用：

- `GetVcoWorkflowExecutionStatus` 可用于轮询 `vRealize Orchestrator` 工作流的状态。
- `WaitForVcoWorkflowCompletion` 可用于阻止继续运行 `IaaS` 工作流，直至 `vRealize Orchestrator` 工作流已完成或超时。使用此活动可以检索以异步方式运行的 `vRealize Orchestrator` 工作流的结果。

## 调用 vRealize Orchestrator 工作流

可以使用 `InvokeVcoWorkflow` 或 `InvokeVcoWorkflowAsync` 活动从 `IaaS` 工作流调用 `vRealize Orchestrator` 工作流。

有些 `vRealize Orchestrator` 工作流要求进行用户交互。对于这些工作流，用户提示将显示在 `vRealize Orchestrator` 客户端中，而不显示在 `vRealize Automation` 控制台中，从而导致 `vRealize Automation` 中的最终用户不知道工作流正在等待输入。

为避免阻止用户输入的工作流，请勿从 `IaaS` 工作流调用需要用户交互的 `vRealize Orchestrator` 工作流。

## 步骤

- 1 在 `vRealize Automation Designer` 中，打开工作流并导航到要从中调用 `vRealize Orchestrator` 工作流的上下文。
- 2 将 `InvokeVcoWorkflow` 或 `InvokeVcoWorkflowAsync` 活动拖动到“设计器”窗格中。
- 3 选择要运行的 `vCenter Orchestrator` 工作流。
  - a 在“常规”下，单击“工作流”旁边的省略号。
  - b 在“浏览 vCO 工作流”对话框中，选择工作流。
  - c 单击**确定**。

“输入”部分和“输出”部分分别显示所选工作流的输入参数和输出参数。

#### 4 在“属性”窗格中，指定以下目标参数之一。

- **VirtualMachineId** 是表示虚拟机 ID 的变量名称。将选择具有此 ID 的虚拟机，并使用从虚拟机的 **VMware.VCenterOrchestrator.EndpointName** 自定义属性中检索到的值作为 **vRealize Orchestrator** 端点名称。
- **VcoEndpointName** 是用于运行工作流的端点名称。如果已指定，则在选择 **vRealize Orchestrator** 端点时，此值会替代 **VirtualMachineId** 值。
- **WorkflowTimeout** 是以秒为单位的超时值。如果 **vRealize Orchestrator** 工作流未在指定时间内完成，则会生成异常，而不会阻止该工作流直至返回响应。如果未定义任何值或提供了零值，则不会激活超时。在该时间段内，系统每 10 秒会检查一次工作流状态，除非通过在 **VMware.VCenterOrchestrator.PollingInterval** 自定义属性中指定一个值来修改端点的轮询时间。

#### 5 指定 vRealize Orchestrator 工作流的参数。

- 在“设计器”窗格中输入活动中的值。
- 在“属性”窗格中，单击 **InputParameters** 或 **OutputParameters** 旁边的省略号以打开“参数”对话框。此对话框显示每个参数的 **IaaS** 类型。如果参数类型以粗体显示，则表示该参数是必需的。

指向任何参数的文本框可查看工具提示，其中指示了 **vRealize Orchestrator** 类型。

如果使用的是 **InvokeVcoWorkflowAsync** 活动，则会显示 **vRealize Orchestrator** 工作流的输出参数及其相应类型以提供信息，但您不能为此活动中的参数指定表达式。

#### 后续步骤

要检索以异步方式运行的工作流结果，请使用 **WaitForVcoWorkflowCompletion** 活动。

#### 获取 vRealize Orchestrator 工作流的状态

可以使用 **GetVcoWorkflowExecutionStatus** 活动来检查通过 **InvokeVcoWorkflowAsync** 活动调用的 **vRealize Orchestrator** 工作流的状态。

#### 前提条件

调用 **vRealize Orchestrator** 工作流使用 **InvokeVcoWorkflowAsync** 活动。

#### 步骤

- 1 在 **vRealize Automation Designer** 中，打开已在其中使用 **InvokeVcoWorkflowAsync** 活动的工作流。
- 2 导航到要从中检查 **vRealize Orchestrator** 工作流的状态的上下文。
- 3 将 **GetVcoWorkflowExecutionStatus** 活动拖动到“设计器”窗格中。
- 4 在“属性”窗格中，在 **VirtualMachineId** 中指定表示虚拟机 ID 的变量名称。  
可自定义工作流包含默认命名为 **virtualMachineId** 的变量，此变量在初始化期间设置。
- 5 创建类型为 **DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken** 的变量。
- 6 指定令牌变量的名称作为 **InvokeVcoWorkflowAsync** 活动的 **executionToken** 输出参数。



- 7 指定同一变量名称作为 `GetVcoWorkflowExecutionStatus` 活动的 `WorkflowExecutionToken` 属性。
- 8 创建类型为字符串的变量。
- 9 指定该字符串变量的名称作为 `GetVcoWorkflowExecutionStatus` 活动的 `VcoWorkflowExecutionStatus` 属性。

## 结果

当工作流运行时，`VcoWorkflowExecutionStatus` 变量的值设置为 `vRealize Orchestrator` 工作流的状态。

## 获取 vRealize Orchestrator 工作流的结果

如果要以异步方式调用 `vRealize Orchestrator` 工作流并在以后检索已完成的工作流的结果，您可以使用 `WaitForVcoWorkflowCompletion` 活动。

`WaitForVcoWorkflowCompletion` 活动会阻止 `IaaS` 工作流，直到 `vRealize Orchestrator` 工作流已完成或超时。如果 `vRealize Orchestrator` 工作流已成功完成，此活动将返回其结果；如果该工作流失败，则返回错误；如果该工作流超时，则返回空值。

## 前提条件

调用 [vRealize Orchestrator 工作流](#) 使用 `InvokeVcoWorkflowAsync` 活动。

## 步骤

- 1 在 `vRealize Automation Designer` 中，打开已在其中使用 `InvokeVcoWorkflowAsync` 活动的工作流。
- 2 导航到要从中检索 `vRealize Orchestrator` 工作流的结果的上下文。
- 3 将 `WaitForVcoWorkflowCompletion` 活动拖动到“设计器”窗格中。
- 4 在“属性”窗格中，在 `VirtualMachineId` 中指定表示虚拟机 ID 的变量名称。  
可自定义工作流包含默认命名为 `virtualMachineId` 的变量，此变量在初始化期间设置。
- 5 创建类型为 `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken` 的变量。
- 6 创建类型为 `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken` 的变量。
- 7 指定令牌变量的名称作为 `InvokeVcoWorkflowAsync` 活动的 `executionToken` 输出参数。
- 8 指定同一变量名称作为 `WaitForVcoWorkflowCompletion` 活动的 `WorkflowExecutionToken` 属性。
- 9 检索 `vRealize Orchestrator` 工作流的输出。
  - a 创建类型为 `DynamicOps.VcoModel.Common.VcoWorkflowExecutionResult` 的变量。
  - b 指定结果变量的名称作为 `WaitForVcoWorkflowCompletion` 活动的 `WorkflowOutput` 属性。

当工作流运行时，变量的值设置为 `vRealize Orchestrator` 工作流的结果（如果有）。

## vRealize Orchestrator 和 IaaS 对象类型

在 `vRealize Automation Designer` 中使用 `InvokeVcoWorkflow` 或 `InvokeVcoWorkflowAsync` 活动时，系统会基于所选 `vRealize Orchestrator` 工作流的参数自动填充该活动的输入属性和输出属性。

基本 `vRealize Orchestrator` 对象类型会转换为以下 `IaaS` 类型：

表 1-19. vRealize Orchestrator 和 IaaS 对象类型

vRealize Orchestrator 类型	IaaS 类型
string	string
boolean	bool
number	decimal
SecureString	string
Text	string
Array/T	Array<T>
Properties	Dictionary<string,object>
Date	DateTime
VC:VirtualMachine	VirtualMachine

**注** 如果使用的是 vRealize Orchestrator 5.1，则您必须已安装 vRealize Automation 集成包才能将 VC:VirtualMachine 对象类型转换为 VirtualMachine。

所有其他 vRealize Orchestrator 类型会转换为 IaaS 类型 VcoSdkObject。

## 配置蓝图以调用状态更改工作流

创建自定义状态更改工作流后，租户管理员或业务组主管必须通过添加自定义属性来为特定蓝图启用该工作流。


每个状态更改工作流都与特定的自定义属性关联。当计算机进入具有对应的状态更改工作流的状态时，IaaS 会检查该计算机是否具有对应的自定义属性；如果有，则会执行关联的工作流。例如，如果计算机具有自定义属性 ExternalWFStubs.MachineProvisioned，则会在主工作流进入 MachineProvisioned 状态之前执行 WFStubMachineProvisioned 工作流。

由于自定义属性可以应用到来自许多源的计算机，因此状态更改工作流的属性通常在蓝图中指定，以便为通过该蓝图置备的所有计算机启用此工作流。

### 前提条件

以租户管理员或业务组主管身份登录到 vRealize Automation。

### 步骤

- 1 选择设计 > 蓝图。
- 2 指向蓝图的名称，然后单击编辑。
- 3 选择蓝图属性图标 (  )。
- 4 单击属性选项卡。
- 5 单击自定义属性 > 新建。

- 6 在**名称**文本框中键入与要启用的工作流关联的自定义属性的名称。

可自定义工作流名称	关联的属性名称
<b>WFStubMachineProvisioned</b>	ExternalWFStubs.MachineProvisioned
<b>WFStubBuildingMachine</b>	ExternalWFStubs.BuildingMachine
<b>WFStubMachineDisposing</b>	ExternalWFStubs.MachineDisposing
<b>WFStubUnprovisionMachine</b>	ExternalWFStubs.UnprovisionMachine
<b>WFStubMachineRegistered</b>	ExternalWFStubs.MachineRegistered
<b>WFStubMachineExpired</b>	ExternalWFStubs.MachineExpired

- 7 将**值**文本框留空。

工作流取决于属性存在与否，而不取决于任何特定值。

- 8 单击**确定**保存属性。

- 9 单击**确定**。

## 结果

该工作流现已对通过此蓝图置备的新计算机启用。

## 配置菜单操作工作流

自定义菜单操作工作流后，需要先进行其他配置，然后该工作流在 vRealize Automation 控制台中才可供用户使用。

### 配置菜单操作

要配置菜单操作，请创建操作配置文件并将其安装到 Model Manager。

### 步骤

#### 1 创建操作配置文件

对于菜单操作工作流而言，操作配置文件是必需的。它指定了 vRealize Automation 控制台中自定义菜单选项的各个方面，例如显示文本、有权访问该选项的角色，以及该选项可用于的计算机状态。

#### 2 在 Model Manager 中安装操作

可以使用 CloudUtil 命令行实用程序在 Model Manager 中安装一个操作。

### 后续步骤

如果该菜单操作设计用于在服务目录中使用，则必须向服务目录注册该菜单操作，以便将其授权给用户。[向服务目录注册新的菜单操作。](#)

### 创建操作配置文件

对于菜单操作工作流而言，操作配置文件是必需的。它指定了 vRealize Automation 控制台中自定义菜单选项的各个方面，例如显示文本、有权访问该选项的角色，以及该选项可用于的计算机状态。

## 步骤

- 1 创建新的 XML 文件。

```
<?xml version="1.0" encoding="utf-8"?>
```

- 2 创建根元素 `customOperations`。

```
<customOperations xmlns="http://www.dynamicops.com/schemas/2009/OperationConfig/">  
</customOperations>
```

此元素必须指定 XML 命名空间 `http://www.dynamicops.com/schemas/2009/OperationConfig/`。

- 3 对于要定义的每个操作，请在 `customOperations` 中添加 `operation` 元素。

```
<operation name="WFMachineMenu1" displayName="Execute Machine Menu task">  
</operation>
```

`operation` 元素采用以下属性：

属性	描述
<b>name</b>	此操作执行的工作流的名称。
<b>displayName</b>	该选项在计算机菜单中的描述性标签。

#### 4 指定要向其授予菜单操作的访问权限的角色。

##### a 添加 `authorizedTasks` 元素。

```
<operation name="WfMachineMenu1" displayName="Execute Machine Menu task">
  <authorizedTasks>
  </authorizedTasks>
</operation>
```

##### b 对于要向其授予操作的访问权限的每个角色，请添加 `task` 元素，例如：

```
<authorizedTasks>
  <task>VRM User Custom Event</task>
  <task>VRM Support Custom Event</task>
  <task>Group Administrator Custom Event</task>
  <task>Enterprise Administrator Custom Event</task>
  <task>VRM Administrator Custom Event</task>
</authorizedTasks>
```

`task` 元素的有效内容如下所示：

元素内容	描述
<b>VRM User Custom Event</b>	向所有用户授予操作的访问权限。
<b>VRM Support Custom Event</b>	向支持用户授予操作的访问权限。
<b>Group Administrator Custom Event</b>	向业务组主管授予操作的访问权限。
<b>Enterprise Administrator Custom Event</b>	向架构管理员授予操作的访问权限。
<b>VRM Administrator Custom Event</b>	仅向 IaaS 管理员授予操作的访问权限。

#### 5 （可选）指定操作在哪些计算机状态下可用。

##### a 添加 `machineStates` 元素。

```
<operation name="WfMachineMenu1" displayName="Execute Machine Menu task">
  <machineStates>
  </machineStates>
</operation>
```

##### b 对于应使操作可用的每个状态，请添加 `state` 元素。

```
<machineStates>
  <state>On</state>
  <state>Off</state>
</machineStates>
```

值可以是任意可能的计算机状态。有关计算机状态的完整列表，请参见《适用于虚拟平台的 IaaS 配置》、《适用于物理机的 IaaS 配置》或《适用于云平台的 IaaS 配置》。

如果省略此元素，则操作在所有计算机状态下均可用。

## 示例

以下是完整的操作配置文件的示例：

```
<?xml version="1.0" encoding="utf-8" ?>
<customOperations xmlns="http://www.dynamicops.com/schemas/2009/OperationConfig/">
  <operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
    <authorizedTasks>
      <task>VRM User Custom Event</task>
      <task>VRM Support Custom Event</task>
      <task>Group Administrator Custom Event</task>
      <task>Enterprise Administrator Custom Event</task>
      <task>VRM Administrator Custom Event</task>
    </authorizedTasks>
    <machineStates>
      <state>On</state>
      <state>Off</state>
    </machineStates>
  </operation>
</customOperations>
```

## 在 Model Manager 中安装操作

可以使用 CloudUtil 命令行实用程序在 Model Manager 中安装一个操作。

### 前提条件

[创建操作配置文件](#)。

### 步骤

- 1 打开提升的命令提示符。
- 2 运行带有以下参数的 CloudUtil.exe 命令。

- CloudUtil.exe Operation-Create -c <操作定义文件的路径>
- （可选）可以指定 Model Manager 主机并在发生错误时请求堆栈跟踪。

```
CloudUtil.exe Operation-Create -c <操作定义文件的路径>
--repository <Model Manager 根 URI> -v
```

### 后续步骤

如果该菜单操作设计用于在服务目录中使用，则必须向服务目录注册该菜单操作，以便将其授权给用户。

[向服务目录注册新的菜单操作](#)。

## 向服务目录注册新的菜单操作

安装新的菜单操作后， workflows 开发人员必须向服务目录注册这些操作，以便将这些操作授权给用户。

### 前提条件

- [配置菜单操作](#)。
- 在 IaaS Model Manager 主机上，以具有**管理员**特权的本地用户身份登录到 Windows。

## 步骤

1 打开提升的命令提示符。

2 导航到 IaaS 根安装目录。

在典型安装中，此目录为 C:\Program Files (x86)\VMware\VCAC。

3 导航到 Server\Model Manager Data\Cafe。

4 执行以下命令：

```
Vcac-Config.exe RegisterCatalogTypes -v
```

## 后续步骤

租户管理员或业务组主管授权新操作后，该操作在服务目录中才可供用户使用。有关详细信息，请参见《租户管理》。

## 配置蓝图以启用菜单操作 workflow

要为通过特定蓝图置备的计算机启用菜单操作 workflow，可更新蓝图的安全配置。

## 前提条件

- 以**租户管理员或业务组主管**身份登录到 vRealize Automation。
- 配置的菜单操作必须存在且已在服务目录中注册。

## 步骤

1 选择**设计 > 蓝图**。

2 指向蓝图的名称，然后单击**编辑**。

3 单击**操作选项卡**。

4 选中与要启用的操作对应的复选框。

5 单击**确定**。

## 结果

该菜单操作现已对通过此蓝图置备的计算机启用，并且可供操作配置文件中指定的所有用户角色使用。

## 后续步骤

如果该菜单操作设计用于在服务目录中使用，则还必须将其授权给用户。有关详细信息，请参见《租户管理》。

## 恢复为工作流的先前修订版本

**加载工作流**对话框显示了工作流在 Model Manager 中的所有修订版本，以便您能访问工作流的完整版本历史记录。

每次将工作流发送到 Model Manager 时，系统将更新修订版本和时间戳。

### 前提条件

启动 vRealize Automation Designer 控制台。

### 步骤

- 1 单击**加载**。
- 2 选择要恢复到的工作流修订版本。  
VMware 提供的原始工作流的修订版本为 0（零）。
- 3 单击**确定**。
- 4 单击**发送**以在 Model Manager 中更新该工作流。

### 结果

早期修订版本将在 Model Manager 中变为最新修订版本。例如，如果您已创建工作流的修订版本 1 和 2，然后加载并保存修订版本 0，则修订版本 0 和 3 现在是相同的，并且工作流已恢复为 VMware 提供的版本。

## 工作流和分布式管理

您可以使用技能将工作流限定为只使用特定 Distributed Execution Manager 运行。

技能类似于标记，可以同时应用到工作流和 DEM Worker 实例。如果工作流未与任何技能相关联，则任何 DEM Worker 都可以运行该工作流。如果工作流与一个或多个技能相关联，则只有关联了所有相同技能的 DEM Worker 才能运行该工作流。

当特定工作流要求 DEM 安装在满足特定必备条件的主机上时，技能非常有用。例如，您可能想要将云置备工作流限定给在可通过所需网络访问 Amazon URL 的主机上运行的特定 DEM。

技能还可用于将工作流与特定数据中心位置相关联。例如，您可以在波士顿数据中心安装一个 DEM 并在伦敦数据中心安装另一个 DEM，然后使用技能将某些操作定向到其中一个数据中心。

## 通过使用技能来关联工作流与 DEM Worker

要将工作流与特定 DEM Worker 或 Worker 实例集相关联，可以向 Model Manager 添加技能，然后将该技能与一个或多个工作流和 DEM Worker 相关联。

### 前提条件

启动 vRealize Automation Designer 控制台。

### 步骤


- 1 在功能区，单击**管理技能**。
- 2 在**管理技能**对话框左上方的文本字段中，键入新技能的名称，然后单击“添加”按钮。  
技能名称必须是唯一的。如果新技能的名称与现有技能的名称相匹配，“添加”按钮将不可用。
- 3 在左侧的列表中选择技能的名称。



4 将技能与一个或多个 DEM Worker 相关联。

- a 单击 Distributed Execution Manager 旁边的**添加**图标（）。
- b 在**选择 DEM** 对话框中，选择一个或多个 DEM Worker 实例。
- c 单击**确定**。

5 将技能与一个或多个工作流相关联。

- a 单击工作流旁边的**添加**图标（）。
- b 在**选择工作流**对话框中，选择一个或多个工作流。
- c 单击**确定**。

只有与技能关联的 DEM Worker 可以执行与此技能关联的工作流。

6 添加技能并将其与 DEM Worker 和工作流相关联后，单击**确定**关闭**管理技能**对话框并保存对 Model Manager 进行的更改。


## 移除技能与 DEM Worker 之间的关联

移除技能与 DEM Worker 之间的关联后，此 Worker 实例无法再执行与该技能关联的工作流。

### 前提条件

启动 vRealize Automation Designer 控制台。

### 步骤

- 1 在功能区，单击**管理技能**。
- 2 在**管理技能**对话框中，在左侧的列表中选择技能的名称。
- 3 从 Distributed Execution Manager 列表选择一个或多个 DEM Worker 实例的名称，然后单击**移除**图标（）。
- 4 单击**确定**关闭**管理技能**对话框并保存对 Model Manager 的更改。


## 移除技能与工作流之间的关联

移除技能与工作流之间的关联后，该工作流不再限定到关联了相同技能的 DEM Worker。

### 前提条件

启动 vRealize Automation Designer 控制台。

### 步骤

- 1 在功能区，单击**管理技能**。
- 2 在**管理技能**对话框中，在左侧的列表中选择技能的名称。
- 3 从“工作流”列表选择一个或多个工作流的名称，然后单击**移除**图标（）。
- 4 单击**确定**关闭**管理技能**对话框并保存对 Model Manager 的更改。


## 移除技能

移除技能将同时移除该技能与任何 DEM Worker 和工作流的关联。

### 前提条件

启动 vRealize Automation Designer 控制台。

### 步骤

- 1 在功能区，单击**管理技能**。
  - 2 在**管理技能**对话框中，在左侧的列表中选择技能的名称。
  - 3 单击技能列表顶部的**移除**图标 ( )。
- 确认要删除技能后，其名称将显示为灰色，以指示该技能已标记为删除。
- 4 单击**确定**以关闭**管理技能**对话框并保存对 Model Manager 进行的更改；或者如果不想删除技能及其与 DEM Worker 和工作流的关联，则单击**取消**。

## CloudUtil 命令参考

本节提供有关 CloudUtil 命令行界面中的命令的参考。

CloudUtil 是适用于 vRealize Automation Designer 的命令行界面。可以在运行设计器的 Windows 计算机上运行命令。Windows 计算机上的默认安装位置是 C:\Program Files (x86)\VMware\vCAC\Design Center。

---

**注** 在 CloudUtil 命令中，Model Manager 称为 repository，而 Distributed Execution Manager (DEM) 称为 agent。

---

## DEM 命令

使用 DEM 命令，您可以查看已向 Model Manager 注册的 Distributed Execution Manager 的列表，还可以添加或移除技能与 DEM 之间的关联。

### DEM-Add-Skills

将技能与已注册的 Distributed Execution Manager 相关联。

#### 摘要

```
CloudUtil.exe DEM-Add-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

## DEM-Add-Skills 参数

参数	描述
-n   - -name	已注册的 Distributed Execution Manager 的名称。
- -repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-s   - -skills	要与此 Distributed Execution Manager 关联的技能的逗号分隔列表。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

**注** 技能必须已存在于 Model Manager 中。请参见 [Skill-Install](#)。

## DEM-List

列出所有已向 Model Manager 注册的 Distributed Execution Manager 及其关联的技能。

### 摘要

```
CloudUtil.exe DEM-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

## DEM-List 参数

参数	描述
- -repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

## DEM-Remove-Skills

移除技能与 Distributed Execution Manager 之间的关联。

### 摘要

```
CloudUtil.exe DEM-Remove-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

## DEM-Remove-Skills 参数

参数	描述
-n   - -name	已注册的 Distributed Execution Manager 的名称。
- -repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-s   - -skills	要从此 Distributed Execution Manager 中移除的技能的逗号分隔列表。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

## File 命令

使用 File 命令，您可以在 Model Manager 中存储和管理文件（通常是脚本）。

### File-Export

从 Model Manager 导出文件。

#### 摘要

```
CloudUtil.exe File-Export -n|--name <Name> -o|--output <Output File> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### File-Export 参数

参数	描述
-i   -iteration	（可选）Model Manager 中文件的版本字符串。默认为 <b>0.0</b> 。
-n   -name	Model Manager 中文件的友好名称。
-o   -output	文件输出的路径。
-repository	（可选）Model Manager 的根 URI，例如， <a href="http://hostname/repository">http://hostname/repository</a> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

### File-Import

将文件导入到 Model Manager。

#### 摘要

```
CloudUtil.exe File-Import -n|--name <Name> -f|--filename <File Name> [-d|--description <Description>] [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### File-Import 参数

参数	描述
-d   -description	（可选）文件的描述。
-f   -filename	要导入到 Model Manager 的文件的文件的路径。
-i   -iteration	（可选）Model Manager 中文件的版本字符串。默认为 <b>0.0</b> 。
-n   -name	要分配给 Model Manager 中文件的友好名称。
-repository	（可选）Model Manager 的根 URI，例如， <a href="http://hostname/repository">http://hostname/repository</a> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

## File-List

列出所有已导入到 Model Manager 的文件。

### 摘要

```
CloudUtil.exe File-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-List 参数

参数	描述
- -repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

## File-Remove-All

从 Model Manager 中移除给定版本文件的所有修订版本。

### 摘要

```
CloudUtil.exe File-Remove-All -n|--name <Name> [-i|--iteration <Iteration>]
[--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Remove-All 参数

表 1-20.

参数	描述
-i   - -iteration	(可选) Model Manager 中文件的版本字符串。默认为 <b>0.0</b> 。
-n   - -name	Model Manager 中文件的友好名称。
- -repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

## File-Remove-Rev

从 Model Manager 中移除特定的文件修订版本。

### 摘要

```
CloudUtil.exe File-Remove-Rev -n|--name <Name> -r|--revision <Revision> [-i|--iteration <Iteration>]
[--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Export 参数

参数	描述
-i   - -iteration	(可选) Model Manager 中文件的版本字符串。默认为 <b>0.0</b> 。
-n   - -name	Model Manager 中文件的友好名称。

参数	描述
-r   -revision	要移除的文件修订版本。
-repository	（可选）Model Manager 的根 URI，例如， <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

## File-Rollback

将 Model Manager 中的文件恢复到指定修订版本。

### 摘要

```
CloudUtil.exe File-Rollback -n|--name <Name> -r|--revision <Revision> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Rollback 参数

表 1-21.

参数	描述
-i   -iteration	（可选）Model Manager 中文件的版本字符串。默认为 <b>0.0</b> 。
-n   -name	Model Manager 中文件的友好名称。
-r   -revision	要恢复到的文件修订版本。
-repository	（可选）Model Manager 的根 URI，例如， <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

## File-Update

使用新修订版本更新先前已导入到 Model Manager 的文件。

### 摘要

```
CloudUtil.exe File-Update -n|--name <Name> -f|--filename <File Name> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Update 参数

参数	描述
-f   -filename	已更新的文件的路径。
-i   -iteration	（可选）Model Manager 中文件的版本字符串。默认为 <b>0.0</b> 。
-n   -name	Model Manager 中文件的友好名称。
-repository	（可选）Model Manager 的根 URI，例如， <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

## Operation 命令

使用 Operation 命令，您可以管理 Model Manager 中的自定义操作。

### Operation-Create

基于操作定义文件，创建可对计算机执行的一个或一组自定义操作。

#### 摘要

```
CloudUtil.exe Operation-Create -c|--operationConfig <Operation Definition File> [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Operation-Create 参数

参数	描述
-c   - --operationConfig	操作定义文件 (XML) 的路径。
- --repository	(可选) Model Manager 的根 URI，例如，http://hostname/repository。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - --verbose	(可选) 如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

### Operation-Delete

从 Model Manager 中删除自定义操作。

#### 摘要

```
CloudUtil.exe Operation-Delete -n|--name <Name> [--force] [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Operation-Delete 参数

参数	描述
- --force	(可选) 强制删除操作。
-n   - --name	Model Manager 中的自定义操作的名称。
- --repository	(可选) Model Manager 的根 URI，例如，http://hostname/repository。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - --verbose	(可选) 如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

### Operation-List

列出 Model Manager 中的所有自定义操作。

#### 摘要

```
CloudUtil.exe Operation-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Operation-List 参数

参数	描述
-repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

## Skill 命令

使用 Skill 命令, 您可以管理与 Distributed Execution Manager 和工作流关联的技能。

### Skill-Install

在 Model Manager 中安装技能。

#### 摘要

```
CloudUtil.exe Skill-Install -n|--name <Name> [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Skill-Install 参数

参数	描述
-n   - -name	Model Manager 中的技能的名称。
-repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

### Skill-List

列出所有已安装在 Model Manager 中的技能。

#### 摘要

```
CloudUtil.exe Skill-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Skill-List 参数

参数	描述
-repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。



## Skill-Uninstall

从 Model Manager 中卸载技能。

### 摘要

```
CloudUtil.exe Skill-Uninstall -n|--name <Name> [--repository <Model Manager Root URI>]
[-v|--verbose]
```

### Skill-Uninstall 参数

参数	描述
-n   - --name	要从 Model Manager 中卸载的技能的名称。
- --repository	(可选) Model Manager 的根 URI，例如， <a href="http://hostname/repository">http://hostname/repository</a> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - --verbose	(可选) 如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

**注** 如果技能与 Distributed Execution Manager 或工作流关联，则无法卸载该技能。请参见 [DEM-Remove-Skills](#) 或 [Workflow-Remove-Skills](#)。

## Workflow 命令

使用 Workflow 命令，您可以管理 Model Manager 中的可自定义 IaaS 工作流以及与任何工作流关联的技能。

### Workflow-Add-Skills

将技能与 Model Manager 中的工作流相关联。

```
CloudUtil.exe Workflow-Add-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>]
[-v|--verbose]
```

表 1-22. Workflow-Add-Skills 参数

参数	描述
Name	Model Manager 中的工作流名称。
Skills	要与此工作流关联的技能的逗号分隔列表。
- --repository	(可选) Model Manager 的根 URI，例如， <a href="http://hostname/repository">http://hostname/repository</a> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - --verbose	(可选) 如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

**注** 技能必须已存在于 Model Manager 中。请参见 [Skill-Install](#)。

### Workflow-List

列出所有已安装在 Model Manager 中的工作流及其关联的技能。

```
CloudUtil.exe Workflow-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

表 1-23. Workflow-List 参数

参数	描述
- -repository	（可选）Model Manager 的根 URI，例如， <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

## Workflow-Remove-Skills

在 Model Manager 中移除技能和工作流之间的关联。

### 摘要

```
CloudUtil.exe Workflow-Remove-Skills -n|--name <Name> -s|--skills <Skills> [--repository
<Model Manager Root URI>] [-v|--verbose]
```

### Workflow-Remove-Skills 参数

参数	描述
-n   - -name	Model Manager 中的工作流名称。
- -repository	（可选）Model Manager 的根 URI，例如， <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-s   - -skills	要从此工作流中移除的技能的逗号分隔列表。
-v   - -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

## Workflow-Rollback

将工作流恢复到给定修订版本。

### 摘要

```
CloudUtil.exe Workflow-Rollback -n|--name <Name> -r|--revision <Revision> [--repository <Model
Manager Root URI>] [-v|--verbose]
```

### Workflow-Rollback 参数

参数	描述
-n   - -name	Model Manager 中的工作流名称。
- -repository	（可选）Model Manager 的根 URI，例如， <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-r   - -revision	要将工作流恢复到的修订版本。
-v   - -verbose	（可选）如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。

## Workflow-Update

使用新修订版本更新可自定义工作流。

```
CloudUtil.exe Workflow-Update -f|--filename <File Name> -n|--name <Name> [-d|--description <Description>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

表 1-24. Workflow-Update 参数

参数	描述
File Name	包含已更新工作流的文件 (XAML) 的路径。
Name	要更新的工作流的名称。
Description	(可选) 工作流的描述。
- -repository	(可选) Model Manager 的根 URI, 例如, <a href="http://hostname/repository">http://hostname/repository</a> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

## 导入命令

使用导入命令, 您可以将一个或多个虚拟机导入到 vRealize Automation 部署中。

## Machine-BulkRegisterExport

创建 CSV 数据文件以用来将虚拟机导入到 vRealize Automation 部署。

### 摘要

```
CloudUtil.exe Machine-BulkRegisterExport [-b|--blueprint] [-m|--managed] [-e|--exportNames] [-p|--properties] -f|--filename <Value> [-g|--group <Value>] [-i|--ignore] [-o|--owner <Value>] [-t|--machinetype <Value>] [-n|--resourceName <Value>] [-r|--resourceType <Value>] [--repository <Value>] [-sn|--sourcename <Value>] [-st|--sourcetype <Value>] -u|--user <value> [-v|--verbose]
```

## Machine-BulkRegisterExport 参数

表 1-25.

参数	描述
-b   - -blueprint	(可选) 包括蓝图名称。
-e   - -exportNames	(可选) 导出名称而非 GUID。
-f   - -filename	指定包含计算机名称列表的 CSV 数据文件的名称, 例如, <b>filename.csv</b> 。默认情况下, 此文件保存在当前路径中。也可以指定指向首选目录的完整路径。
-g   - -group	(可选) 指定业务组名称, 例如, <b>Engineering</b> 。
-i   - -ignore	(可选) 忽略无效参数。
-m   - -managed	(可选) 导出受管虚拟机。默认设置是导出非受管虚拟机。
-n   - -resourceName	(可选) 要按资源名称筛选, 请指定计算资源或端点的名称。
-o   - -owner	(可选) 指定导入的虚拟机的所有者, 例如, <b>jsmith</b> 。

表 1-25. (续)

参数	描述
-p   - -properties	(可选) 导出受管虚拟机的属性。
-r   - -resourceType	(可选) 要按资源类型筛选, 请为计算资源指定 1 或为端点指定 2。
- -repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-sn   - -sourcename	(可选) 指定群集或端点的名称。
-st   - -sourcetype	(可选) 将源类型指定为群集或端点。
-t   - -machinetype	(可选) 指定要导出的计算机类型, 例如, 虚拟机、物理机、云、AppService 或 vApp。
-u   - -user	指定将执行批量注册的架构管理员。
-v   - -verbose	(可选) 如果发生错误, 将输出堆栈跟踪, 而不是仅输出异常消息。

## Machine-BulkRegisterImport

将一个或多个虚拟机导入到目标 vRealize Automation 部署。

### 摘要

```
CloudUtil.exe Machine-BulkRegisterImport [-b|--batch][--delay <value>] -f|--filename <value> [-i|--ignore] [-h|--humanreadable] -n|--name <value> [--repository <value>] [-s|--skipUser] -t|--time <value> -u|--user <value> [-v|--verbose] [-w|--whatIf]
```

### Machine-BulkRegisterImport 参数

表 1-26.

参数	描述
-b   - -batch	(可选) 批处理大小。
-d   - -delay	(可选) 以 hh:mm:ss 格式指定处理延迟时间, 例如, 02:20:10。
-f   - -filename	指定包含计算机名称列表的 CSV 数据文件的名称。例如, filename.csv。
-h   - -humanreadable	(可选) 输入文件包含虚拟机名称, 而不包含 GUID。
-i   - -ignore	(可选) 忽略已注册的虚拟机或受管虚拟机。
-n   - -name	指定向目标 vRealize Automation 执行导入的工作队列的名称。
- -repository	(可选) Model Manager 的根 URI, 例如, <code>http://hostname/repository</code> 。CloudUtil 配置文件的 <appSettings> 部分下的 repositoryAddress 项中指定了默认值。
-s   - -skipUser	(可选) 将计算机的所有者设置为 CSV 数据文件的“所有者”列中列出的值, 而不需要验证用户是否存在。选择此选项可缩短导入所需的时间。

表 1-26. (续)

参数	描述
-t   - -time	以 MM/DD/YYYY hh:mm GMT 格式指定 workflow 开始时间，例如，04/18/2014 10:01 GMT。指定的开始时间假定为服务器的本地时间，而不是用户工作站的本地时间。
-u   - -user	指定将执行批量注册的架构管理员。
-v   - -verbose	(可选) 如果发生错误，将输出堆栈跟踪，而不是仅输出异常消息。
whatif	(可选) 设置此参数可验证 CSV 文件而不导入任何虚拟机。

## vRealize Automation workflow 活动参考

VMware 随 vRealize Automation Designer 提供了 workflow 活动库，您可以在自定义 workflow 时使用该库。

**注** 从 vRealize Automation 7.0 开始，CDK 已弃用。您可以使用 vRealize Orchestrator workflow 处理之前通过 CDK 处理的用例。

vRealize Automation Designer 中还包含五种类别的 Windows Workflow Foundation 活动，包括控制流、流程图、基元、集合与错误处理。

本节提供 vRealize Automation Designer 附带的位于 `DynamicOps.Repository.Activities` 和 `DynamicOps.Cdk.Activities` 命名空间的 IaaS workflow 的参考信息。使用 [vRealize Orchestrator workflow 活动](#) 介绍了与调用 vRealize Orchestrator workflow 相关的活动。

**注** 在 IaaS 活动库中，Model Manager 称为 `repository`。

## DynamicOps.Repository.Activities

`DynamicOps.Repository.Activities` 命名空间包含 IaaS workflow 的基本 workflow 活动。

**注** 从 vRealize Automation 7.0 开始，CDK 已弃用。您可以使用 vRealize Orchestrator workflow 处理之前通过 CDK 处理的用例。

### AddLink

向 `DataServiceContext` 跟踪的对象集添加指定链接。

表 1-27. AddLink 活动输入参数

参数	类型	描述
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	要将链接添加到的 <code>DataServiceContext</code> 。
<code>Source</code>	<code>Object</code>	新链接的源对象。
<code>SourceProperty</code>	<code>String</code>	返回相关对象的源对象上的导航属性的名称。
<code>Target</code>	<code>Object</code>	通过新链接与源对象相关的对象。

## AddObject

向 `DataServiceContext` 跟踪的对象集添加指定对象。

表 1-28. AddObject 活动输入参数

参数	类型	描述
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	要将对象添加到的 <code>DataServiceContext</code> 。
<code>Instance</code>	<code>Object</code>	要由 <code>DataServiceContext</code> 跟踪的对象。

## AttachTo

通知 `DataServiceContext` 开始跟踪指定资源。

表 1-29. AttachTo 活动输入参数

参数	类型	描述
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	应跟踪资源的 <code>DataServiceContext</code> 。
<code>Instance</code>	<code>Object</code>	要由 <code>DataServiceContext</code> 跟踪的资源。该资源是在未更改状态下附加的。

## CreateRepositoryServiceContext<T>

为加载到 Model Manager 中的模型创建上下文。

在 vRealize Automation Designer 中向 workflow 添加此活动时，您必须选择从 `RepositoryServiceContext` 类继承的类。

表 1-30. CreateRepositoryServiceContext<T> 活动输入参数

参数	类型	描述
<code>Uri</code>	<code>URI</code>	(可选) 用于连接到模型的根 URI。
<code>Username</code>	<code>String</code>	(可选) 用于连接到上下文的用户名。

表 1-31. CreateRepositoryServiceContext<T> 活动输出参数

参数	类型	描述
<code>Result</code>	<code>RepositoryServiceContext</code>	返回的特定类型是向 workflow 添加此活动时选择的类的实例。

## DeleteLink

将正在由 `DataServiceContext` 跟踪的链接列表中的链接的状态更改为已删除。

表 1-32. DeleteLink 活动输入参数

参数	类型	描述
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	要从中删除链接的 <code>DataServiceContext</code> 。
<code>Source</code>	<code>Object</code>	要标记为删除的链接中的源对象。

表 1-32. DeleteLink 活动输入参数（续）

参数	类型	描述
SourceProperty	String	用于访问目标对象的源对象上的导航属性的名称。
Target	Object	包含在链接中且绑定到源对象的目标对象。目标对象必须属于 source 属性可识别的类型或子类型。

## DeleteObject

将 DataServiceContext 中的指定对象的状态更改为待删除。

表 1-33. DeleteObject 活动输入参数

参数	类型	描述
DataServiceContext	RepositoryServiceContext	要从中删除资源的 DataServiceContext。
Instance	Object	要更改为已删除状态的受跟踪实体。

## InvokeRepositoryWorkflow

执行已安装在 Model Manager 中的工作流。

表 1-34. InvokeRepositoryWorkflow 活动输入参数

参数	类型	描述
WorkflowType	WorkflowDefinition 实体	要执行的工作流。
WorkflowInputs	Dictionary<string, object>	（可选）工作流的输入。
CallingInstance	WorkflowInstance 实体	（可选）调用执行的工作流并向其返回消息的工作流。

## LoadProperty

从数据服务加载指定属性的延迟内容。

表 1-35. LoadProperty 活动输入参数

参数	类型	描述
DataServiceContext	RepositoryServiceContext	要从中加载属性的 DataServiceContext。
Instance	Object	包含要加载属性的实体。
InstanceProperty	String	要从指定实体加载的属性的名称。

## SaveChanges

将 DataServiceContext 正在跟踪的更改保存到存储。

表 1-36. SaveChanges 活动输入参数

参数	类型	描述
DataServiceContext	RepositoryServiceContext	正在跟踪要保存的更改的 DataServiceContext。

## SetLink

通知 `DataServiceContext` 指定的对象之间存在新链接，并且该链接由 `SourceProperty` 参数中指定的属性表示。

表 1-37. SetLink 活动输入参数

参数	类型	描述
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	要向其通知链接的 <code>DataServiceContext</code> 。
<code>Source</code>	<code>Object</code>	新链接的源对象。
<code>SourceProperty</code>	<code>String</code>	源对象上的属性，标识了新链接的目标对象。
<code>Target</code>	<code>Object</code>	包含在新链接中并可通过调用此方法进行初始化的子对象。目标对象必须属于 <code>SourceProperty</code> 标识的类型的子类型。如果 <code>Target</code> 设置为空值，则调用表示删除链接操作。

## UpdateObject

将 `DataServiceContext` 中的指定对象的状态更改为“已修改”。

表 1-38. UpdateObject 活动输入参数

参数	类型	描述
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	用来跟踪要更新的实体的 <code>DataServiceContext</code> 。
<code>Instance</code>	<code>Object</code>	将指定为“已修改”状态的受跟踪实体。

## DynamicOps.Cdk.Activities

`DynamicOps.Cdk.Activities` 命名空间包含 IaaS 工作流的高级活动。

**注** 从 vRealize Automation 7.0 开始，CDK 已弃用。您可以使用 vRealize Orchestrator 工作流处理之前通过 CDK 处理的用例。

## ExecutePowerShellScript

执行以指定名称存储在 Model Manager 中的 PowerShell 脚本。

使用 `ExecutePowerShellScript` 活动之前，必须先使用 `CloudUtil File-Import` 命令将要执行的脚本加载到 Model Manager 中。

表 1-39. ExecutePowerShellScript 活动输入参数

参数	类型	描述
<code>ScriptName</code>	<code>String</code>	要执行的脚本在 Model Manager 中的名称。
<code>ScriptVersion</code>	<code>Object</code>	（可选）要执行的脚本在 Model Manager 中的版本。默认为 0.0。
<code>Machinelid</code>	<code>Guid</code>	（可选）如果已指定，则会加载计算机，并且该计算机的所有属性会传递到脚本。



表 1-39. ExecutePowerShellScript 活动输入参数（续）

参数	类型	描述
Arguments	Dictionary<string,string>	要传递到脚本的其他参数。如果指定了 <b>MachineId</b> ，并且存在与参数具有相同名称（不区分大小写）的计算机属性，则该计算机属性的值会替代参数的值。
PSModules	IEnumerable<string>	（可选）在命令执行期间加载到 PowerShell 运行时中的模块。此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。

表 1-40. ExecutePowerShellScript 活动输出参数

参数	类型	描述
Output	Collection<PSObject>	脚本的输出（如果有）。在发生错误时抛出异常。

如果处理 ExecutePowerShellScript 的输出时在 vRealize Automation Designer 控制台中收到错误消息未定义类型 PSObject (Type PSObject is not defined)，请执行以下步骤：

- 1 单击“设计器”窗格左下角的**导入**。
- 2 选择 **System.Management.Automation** 程序集。

## ExecuteSshScript

执行以指定名称存储在 Model Manager 中的 SSH 脚本。

使用 ExecuteSshScript 活动之前，必须先使用 CloudUtil File-Import 命令将要执行的脚本加载到 Model Manager 中。

表 1-41. ExecuteSshScript 活动输入参数

参数	类型	描述
ScriptName	String	要执行的脚本在 Model Manager 中的名称。
Host	String	要对其执行脚本的服务器的名称。
Username	String	用于连接到主机的用户名。
Password	String	用于连接到主机的密码。
ScriptVersion	Object	（可选）要执行的脚本在 Model Manager 中的版本。默认为 0.0。
Timeout	TimeSpan	（可选）脚本执行超时前的时间段。默认值为 30 分钟。

表 1-42. ExecuteSshScript 活动输出参数

参数	类型	描述
EnvironmentVariables	Dictionary<string, string>	脚本执行结果（如果有）。

## GetMachineName

获取计算机的名称。

表 1-43. GetMachineName 活动输入参数

参数	类型	描述
MachineId	Guid	要检索其名称的计算机。

表 1-44. GetMachineName 活动输出参数

参数	类型	描述
MachineName	String	用 MachineId 标识的计算机名称。

## GetMachineOwner

获取计算机所有者的用户名。

表 1-45. GetMachineOwner 活动输入参数

参数	类型	描述
MachineId	Guid	要检索其所有者的计算机。

表 1-46. GetMachineOwner 活动输出参数

参数	类型	描述
Owner	String	用 MachineId 标识的计算机所有者，或者为空值（如果没有所有者）。

## GetMachineProperties

获取与计算机关联的自定义属性的列表。

表 1-47. GetMachineProperties 活动输入参数

参数	类型	描述
MachineId	Guid	要检索其属性的计算机。

表 1-48. GetMachineProperties 活动输出参数

参数	类型	描述
Properties	Dictionary<string, string>	计算机的属性列表。如果返回的值以加密方式存储，则会将其解密。

## GetMachineProperty

获取计算机的指定属性的值。

表 1-49. GetMachineProperty 活动输入参数

参数	类型	描述
MachineId	Guid	要从中检索属性的计算机。
PropertyName	String	要返回其值的属性的名称。
IsRequired	bool	如果属性是必需属性，但却找不到，该活动将抛出异常或返回空值。

表 1-50. GetMachineProperty 活动输出参数

参数	类型	描述
PropertyValue	String	由 <b>PropertyName</b> 指定的属性的值。如果返回的值以加密方式存储，则会将其解密。

## GetScriptFromName

获取以指定名称存储在 Model Manager 中的脚本的内容。

表 1-51. GetScriptFromName 活动输入参数

参数	类型	描述
ScriptName	String	要检索的脚本在 Model Manager 中的名称。
ScriptVersion	Object	（可选）要检索的脚本在 Model Manager 中的版本。默认为 0.0。

表 1-52. GetScriptFromName 活动输出参数

参数	类型	描述
ScriptContent	String	用 <b>ScriptName</b> 标识的脚本内容。

## InvokePowerShell

执行 PowerShell 命令。

表 1-53. InvokePowerShell 活动输入参数

参数	类型	描述
CommandText	String	要执行的命令。
Arguments	IEnumerable<string>	（可选）命令的参数。
Input	IEnumerable	（可选）输入管道。
IsScript	bool	（可选）指示 <b>CommandText</b> 是否为脚本。默认值为 <b>False</b> 。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
Parameters	Collection	（可选）作为参数传递到 PowerShell 脚本的名称-值对的集合。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。

表 1-53. InvokePowerShell 活动输入参数（续）

参数	类型	描述
PowerShellVariables	Collection	（可选）复制到 PowerShell 运行时的变量。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
PSModules	IEnumerable<string>	（可选）在命令执行期间加载到 PowerShell 运行时中的模块。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
Runspace	Runspace	（可选）通过创建 PowerShell 运行空间并将其提供给此参数，您可以在多次 PowerShell 调用中重复使用同一个运行空间，这样可能有助于提高性能。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。

表 1-54. InvokePowerShell 活动输出参数

参数	类型	描述
Output	Collection<PSObject>	命令的输出（如果有）。在发生错误时抛出异常。
Errors	Collection<ErrorRecord>	执行导致的错误（如果有）。

如果处理 ExecutePowerShellScript 的输出时在 vRealize Automation Designer 控制台中收到错误消息未定义类型 PSObject (Type PSObject is not defined)，请执行以下步骤：

- 1 单击“设计器”窗格左下角的**导入**。
- 2 选择 **System.Management.Automation** 程序集。

## InvokeSshCommand

执行 SSH 命令。

表 1-55. InvokeSshCommand 活动输入参数

参数	类型	描述
CommandText	String	要执行的命令。
Host	String	要对其执行命令的服务器的名称。
Username	String	用于连接到主机的用户名。
Password	String	用于连接到主机的密码。
Timeout	TimeSpan	（可选）命令执行超时前的时间段。默认值为 30 分钟。

表 1-56. InvokeSshCommand 活动输出参数

参数	类型	描述
EnvironmentVariables	Dictionary<string, string>	命令的输出（如果有）。在发生错误时抛出异常。

## LogMachineEvent

将计算机事件记录到对计算机所有者可见的用户日志。

表 1-57. LogMachineEvent 活动输入参数

参数	类型	描述
Machinelid	Guid	生成要记录的事件的计算机。
Message	String	要写入到用户日志的消息。
Type	String	从下拉列表中选择消息类型（信息、警告、错误）

## LogMessage

记录到 Distributed Execution Manager 日志。

表 1-58. LogMessage 活动输入参数

参数	类型	描述
Message	String	要写入到 DEM 日志的消息。
MessageCategory	String	从下拉菜单中选择类别（ <b>调试、错误、信息、跟踪</b> ），或者输入自定义类别。
MessageSeverity	String	从下拉菜单中选择严重性；绑定到 <code>System.Diagnostics.TraceEventType</code> 中提供的严重性列表。

## RunProcess

在执行此活动的 DEM 所在的同一台计算机上执行进程。

**注** vRealize Automation 无法通过由 RunProcess 活动启动的进程向用户提供 UI，因此这些进程必须是非交互式的。为了避免在 DEM 计算机上保留孤立进程，这些进程还必须能够自行终止。

表 1-59. RunProcess 活动输入参数

参数	类型	描述
Command	String	在 DEM 计算机上运行的可执行文件的路径。
WorkingDirectory	String	（可选）进程应在其中运行的工作目录。
Arguments	String	（可选）要传递到命令的命令行参数的列表。
WaitForExit	bool	（可选）如果值为 <b>true</b> ，工作流将等待进程完成，然后再继续执行工作流。默认值为 <b>false</b> 。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。

## SendEmail

向给定地址集发送电子邮件。

表 1-60. SendEmail 活动输入参数

参数	类型	描述
To	IEnumerable<string>	要向其发送电子邮件的地址列表。
From	String	用来填充电子邮件“发件人”字段的地址。
Subject	String	电子邮件的主题行。
Body	String	电子邮件的正文文本。
Host	String	出站 SMTP 服务器的主机名或 IP 地址。
Port	Integer	在 Host 中指定的服务器的 SMTP 端口。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
CC	IEnumerable<string>	（可选）要将电子邮件抄送到的地址或地址列表。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
Bcc	IEnumerable<string>	（可选）要将电子邮件密送到的地址或地址列表。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
EnableSsl	bool	（可选）指示是否使用 SSL。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
UserName	String	用来向 Host 中指定的 SMTP 服务器进行身份验证的用户名。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。
Password	String	在 UserName 中指定的用户的密码。 此选项仅在“属性”窗格中可用，在“设计器”窗格中不可用。

## SetMachineProperty

在计算机上创建或更新自定义属性。

表 1-61. SetMachineProperty 活动输入参数

参数	类型	描述
MachineId	Guid	要在上面创建或更新自定义属性的计算机。
PropertyName	String	要创建或更新的属性的名称。
PropertyValue	String	用来创建或更新属性的值。
IsEncrypted	bool	（可选）指示是否将属性值加密。
IsHidden	bool	（可选）指示属性是否为隐藏属性。
IsRuntime	bool	（可选）指示请求用户是否在请求时提供属性值（相当于在 vRealize Automation 控制台中标记为“提示用户”）。

## SetWorkflowResult

将外部工作流的状态设置为“完成”或“失败”，以供 ExternalWF.xml 设置遵守。

表 1-62. SetWorkflowResult 活动输入参数

参数	类型	描述
WorkflowId	Guid	要为其设置状态的工作流。
Next State	WorkflowState	从下拉菜单中选择 <b>完成</b> 或 <b>失败</b> 。