

使用 VMware vRealize Orchestrator 插件

vRealize Orchestrator 7.5

您可以从 VMware 网站下载最新的技术文档:

<https://docs.vmware.com/cn/>。

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

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

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

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

目录

使用 VMware vRealize Orchestrator 插件 10

1 Orchestrator 插件简介 11

Orchestrator 架构 11

随 Orchestrator 服务器一起安装的插件 12

访问 Orchestrator API Explorer 14

2 配置 Orchestrator 插件 16

管理 Orchestrator 插件 16

卸载插件 17

3 使用 vCenter Server 插件 19

配置 vCenter Server 插件 19

配置 workflow 20

配置 vCenter Server 实例的连接 20

vCenter Server 插件脚本 API 21

使用 vCenter Server 插件清单 21

查询的性能注意事项 22

使用 XPath 表达式与 vCenter Server 插件 22

使用 XPath 表达式与 vCenter Server 插件 23

访问 vCenter Server 插件 workflow 23

vCenter Server 插件 workflow 24

批处理 workflow 26

群集和计算资源 workflow 26

配置 workflow 27

自定义属性 workflow 27

数据中心 workflow 28

数据存储和文件 workflow 28

数据中心文件夹管理工作流 28

主机文件夹管理工作流 29

虚拟机文件夹管理工作流 29

客户机操作文件 workflow 29

客户机操作进程 workflow 30

主机电源管理工作流 30

基本主机管理工作流 30

主机注册管理工作流 31

网络 workflow 31

分布式虚拟端口组 workflow	31
分布式虚拟交换机 workflow	32
标准虚拟交换机 workflow	32
网络 Virtual SAN workflow	32
资源池 workflow	33
存储 workflow	33
Storage DRS workflow	33
Storage VSAN workflow	34
基本虚拟机管理工作流	34
克隆 workflow	35
链接克隆 workflow	35
Linux 自定义克隆 workflow	36
工具克隆 workflow	36
Windows 自定义克隆 workflow	37
设备管理工作流	37
移动和迁移 workflow	37
其他 workflow	38
电源管理工作流	39
快照 workflow	39
VMware Tools workflow	39

4 使用 vRealize Automation 插件 41

VMware vRealize Orchestrator Plug-In for vRealize Automation 简介	41
包含 vRealize Automation 插件的 vRealize Orchestrator 的角色	42
配置 vRealize Automation 插件	42
配置 workflow	42
使用 vRealize Automation 插件 workflow	45
移除操作限制	45
使用 vRealize Automation 插件清单	47
使用 vRealize Automation 插件管理工作流	48
使用 vRealize Automation 插件基础架构管理工作流	53
使用 vRealize Automation 插件请求 workflow	56
使用 vRealize Automation 插件示例 workflow	57
访问 vRealize Automation 插件 API	57
示例 vRealize Automation 插件脚本	58
CRUD 基础架构管理任务示例脚本	58
查找 vRealize Automation 实体示例脚本	62
获取 vRealize Automation 示例脚本置备的资源	63
常规任务示例脚本	64

5 使用 Configuration 插件 68

[访问 Configuration 插件工作流库](#) 68

[Configuration 插件工作流库](#) 68

6 使用 Library 插件 70

[Library 插件工作流](#) 70

7 使用 SQL 插件 72

[配置 SQL 插件](#) 72

[SQL 插件配置工作流](#) 72

[添加数据库](#) 73

[将表格添加到数据库](#) 74

[更新数据库](#) 74

[运行 SQL 示例工作流](#) 75

[生成 JDBC URL](#) 75

[测试 JDBC 连接](#) 76

[使用 JDBC 创建表格](#) 76

[在 JDBC 表格中插入行](#) 77

[选择 JDBC 表格中的行](#) 77

[删除 JDBC 表格中的条目](#) 78

[删除 JDBC 表格中的所有条目](#) 79

[丢弃 JDBC 表格](#) 79

[运行完整的 JDBC 循环](#) 80

[使用 SQL 插件标准工作流](#) 80

[SQL 插件工作流库](#) 80

[为表格生成 CRUD 工作流](#) 81

8 使用 SSH 插件 83

[配置 SSH 插件](#) 83

[配置工作流](#) 84

[运行 SSH 插件示例工作流](#) 84

[生成密钥对](#) 85

[更改密钥对密码短语](#) 85

[在 SSH 主机上注册 Orchestrator 公共密钥](#) 86

[运行 SSH 命令](#) 86

[从 SSH 主机复制文件](#) 87

[将文件复制到 SSH 主机](#) 87

9 使用 XML 插件 89

[运行 XML 插件示例工作流](#) 89

[创建简单 XML 文档](#) 90

[查找 XML 文档中的元素](#) 90

- [修改 XML 文档 91](#)
 - [从 XML 创建示例通讯簿 92](#)
- 10 使用 Mail 插件 93**
 - [定义默认 SMTP 连接 93](#)
 - [使用 Mail 插件示例工作流 94](#)
 - [访问 Mail 插件示例工作流 94](#)
 - [Mail 插件示例工作流 94](#)
- 11 使用 Net 插件 96**
- 12 使用 Enumeration 插件 97**
 - [时区代码 97](#)
- 13 使用 Workflow Documentation 插件 100**
 - [Workflow Documentation 插件的工作流库 100](#)
 - [生成工作流文档 100](#)
- 14 使用 HTTP-REST 插件 102**
 - [配置 HTTP-REST 插件 102](#)
 - [配置工作流 102](#)
 - [配置 Kerberos 身份验证 103](#)
 - [添加 REST 主机 104](#)
 - [添加 REST 操作 106](#)
 - [将架构添加到 REST 主机 106](#)
 - [通过 REST 操作生成新工作流 107](#)
 - [调用 REST 操作 108](#)
 - [调用 REST 操作 108](#)
- 15 使用 SOAP 插件 109**
 - [配置 SOAP 插件 109](#)
 - [配置工作流 109](#)
 - [添加 SOAP 主机 110](#)
 - [配置 Kerberos 身份验证 111](#)
 - [通过 SOAP 操作生成新工作流 112](#)
 - [测试自定义生成的工作流 113](#)
 - [调用 SOAP 操作 114](#)
- 16 使用 AMQP 插件 115**
 - [配置 AMQP 插件 115](#)
 - [配置工作流 115](#)

添加代理	116
订阅队列	116
更新代理	117
使用 AMQP 插件标准工作流	117
声明绑定	118
声明队列	118
声明交换	119
发送文本消息	120
删除绑定	121
17 使用 SNMP 插件	122
管理 SNMP 设备	122
设备管理工作流	122
注册 SNMP 设备	123
管理 SNMP 查询	124
查询管理工作流	124
将查询添加到 SNMP 设备	124
管理 SNMP 陷阱主机	125
陷阱主机管理工作流	125
设置 SNMP 陷阱端口	125
接收 SNMP 陷阱	126
等待 SNMP 设备上的陷阱	126
设置 SNMP 陷阱策略	126
配置 SNMP 陷阱主机策略	127
编辑陷阱策略	128
通用 SNMP 请求工作流	128
18 使用 Active Directory 插件	130
配置 Active Directory 插件	130
Active Directory 配置工作流	130
使用 Active Directory 插件工作流库	131
使用 Active Directory 插件清单	131
访问 Active Directory 插件工作流库	131
Active Directory 插件工作流	131
19 使用 Dynamic Types 插件	134
Dynamic Types 配置工作流	134
20 使用 PowerShell 插件	136
VMware vRealize Orchestrator PowerShell 插件简介	136
PowerShell 插件组件	137

配置 WinRM	138
配置 Kerberos 身份验证	141
配置 PowerShell 插件	142
配置工作流	142
添加 PowerShell 主机	143
使用 PowerShell 插件清单	144
运行 PowerShell 脚本	144
调用 PowerShell 脚本	144
调用外部脚本	144
生成操作	145
从 PowerShell 脚本生成操作	145
为 PowerShellcmdlet 生成操作	146
在操作之间传递调用结果	147
PowerCLI 与 PowerShell 插件集成	147
转换器工作流	148
示例工作流	148
访问 PowerShell 插件 API	148
使用 PowerShell 结果	149
常规 PowerShell 任务脚本的示例	149
故障排除	151
启用 Kerberos 事件日志记录	151
Kerberos 数据库中未找到服务器	152
无法获取 Kerberos 票证	153
Kerberos 身份验证因时间设置不同导致失败	153
Kerberos 身份验证会话模式失败	153
无法访问领域的密钥分发中心	154
无法找到默认领域	154

21 使用 Multi-Node 插件 155

vRealize Orchestrator Multi-Node 插件简介	155
配置 Multi-Node 插件	156
服务器配置工作流	156
添加 Orchestrator 服务器	156
使用代理工作流	157
同步代理工作流	157
异步代理工作流	157
远程执行工作流	158
使用 Multi-Node 插件清单	159
远程管理工作流	159
访问 Multi-Node 插件 API	160
Multi-Node 插件用例	160

[创建多代理操作 160](#)

[维护远程和代理工作流 161](#)

[部署本地服务器中的软件包 162](#)

22 使用 vCloud Suite API (vAPI) 插件 163

[配置 vCloud Suite API 插件 163](#)

[导入 vCloud Suite API 元模型 163](#)

[添加 vCloud Suite API 端点 164](#)

[访问 vCloud Suite API 插件 API 165](#)

使用 VMware vRealize Orchestrator 插件

《使用 VMware vRealize Orchestrator 插件》提供了有关配置和使用随 VMware® vRealize Orchestrator 安装的标准插件组的信息和说明。

目标读者

本文档提供的信息主要面向熟悉虚拟机技术和数据中心操作且具有丰富经验的高级 vSphere 管理员以及系统管理员。

Orchestrator 插件简介

1

使用 Orchestrator 插件，可以访问并控制外部技术和应用程序。通过在 Orchestrator 插件中公开外部技术，您可以将对象和功能结合到工作流，并在该外部技术的对象上运行工作流。

可通过插件访问的外部技术包含虚拟化管理工具、电子邮件系统、数据库、目录服务、远程控制接口等。

Orchestrator 提供了一组标准预安装插件，公开了 VMware vCenter Server API、电子邮件和身份验证功能以及其他技术。此外，还可以使用 Orchestrator 开放式插件架构，开发用于访问其他应用程序的插件。Orchestrator 采用开放式标准，简化了与外部系统的集成。有关开发自定义内容的信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

标准插件组会自动随 Orchestrator 服务器一起安装。您可能需要配置部分插件（例如 vCenter Server 插件），然后才能使用这些插件。

这些插件采用新的对象类型和方法扩展了 Orchestrator 脚本引擎，外部系统触发了 Orchestrator 中和插件技术中的事件时，插件还会发布相应的通知事件。插件提供了 JavaScript 对象的清单，您可以通过 Orchestrator 客户端的清单选项卡对其访问。每个插件都包含由工作流和操作构成的软件包，您可用在清单的对象上运行，从而自动处理集成产品的典型用例。

本章讨论了以下主题：

- [Orchestrator 架构](#)
- [随 Orchestrator 服务器一起安装的插件](#)
- [访问 Orchestrator API Explorer](#)

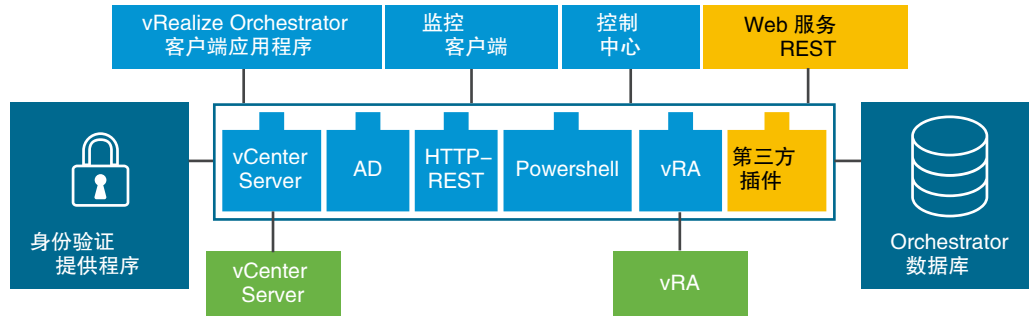
Orchestrator 架构

Orchestrator 包含一个工作流库和一个工作流引擎，可用于创建并运行相关工作流，实现编排流程自动化。Orchestrator 可通过一系列插件访问各种不同技术对象，您则可以对这些对象运行工作流。

Orchestrator 提供了一组标准插件，包括适用于 vCenter Server 和 vRealize Automation 的插件，您可在插件所公开的不同环境中编排各种任务。

Orchestrator 还提供了开放式架构，用于将外部第三方应用程序插入编排平台。您可以对自定义插件技术的对象运行工作流。Orchestrator 将连接到身份验证提供程序以管理用户帐户，并连接到数据库以存储来自其运行的工作流的信息。您可以通过 Orchestrator 客户端界面或 Web 服务访问 Orchestrator 及其公开的对象，以及 Orchestrator 工作流。通过监控客户端和控制中心来监控和配置 Orchestrator 工作流与服务。

图 1-1. VMware vRealize Orchestrator 架构



随 Orchestrator 服务器一起安装的插件

Orchestrator 随附一组标准插件，每个插件可向 Orchestrator 平台公开一个外部产品 API。插件用于向脚本引擎提供清单类、其他对象类型，并发布来自外部系统的通知事件。每个插件还可提供了工作流库，自动处理集成外部产品的典型用例。

在控制中心的**管理插件**页面中，可以查看已安装插件的列表。对于需要配置的插件，界面中提供单独的选项卡。

表 1-1. 随 Orchestrator 一起安装的插件

插件	用途	配置
vCenter Server	提供对 vCenter Server API 的访问权限，能够将所有 vCenter Server 对象和功能纳入可通过 Orchestrator 自动执行的管理流程中。	请参见 配置 vCenter Server 插件 。
配置	提供用于配置 Orchestrator 身份验证、数据库连接、SSL 证书等的工作流。	无
库	提供可充当客户端流程自定义和自动化基本构建块的工作流。工作流库中包括可用于生命周期管理、置备、灾难恢复、热备份以及其他标准系统管理流程的模板。您可以复制和编辑这些模板，根据需要对其进行修改。	无
SQL	提供 Java 数据库连接 (JDBC) API，一种专用于 Java 编程语言和各种数据库之间建立独立于数据库的连接的行业标准。这些数据库包括 SQL 数据库以及其他表格数据源，例如电子表格或平面文件。JDBC API 提供了调用级 API，以便从工作流访问基于 SQL 的数据库。	无
SSH	支持实现安全 Shell v2 (SSH-2) 协议。允许在工作流中采用基于密码和公钥的身份验证运行远程命令和文件传输会话。支持键盘交互式身份验证。（可选）SSH 插件支持直接在 Orchestrator 客户端清单中浏览远程文件系统。	请参见 配置 SSH 插件 。

表 1-1. 随 Orchestrator 一起安装的插件（续）

插件	用途	配置
XML	一种可在工作流中实施的完整文档对象模型 (DOM) XML 分析程序。或者，您也可以在 Orchestrator JavaScript API 中使用 ECMAScript for XML (E4X) 实施。	无
Mail	使用简单邮件传输协议 (SMTP) 从工作流发送电子邮件。	设置 <code>EmailMessage</code> 对象要使用的默认值。 请参见 定义默认 SMTP 连接 。
Net	使用 Jakarta Apache Commons Net 库。支持 Telnet、FTP、POP3 和 IMAP 协议实现。POP3 和 IMAP 协议用于读取电子邮件。Net 插件与 Mail 插件结合使用，可在工作流中提供完整的电子邮件收发功能。	无
工作流文档	提供相关工作流，可用于以 PDF 格式生成相关工作流或工作流类别的信息。	无
枚举	提供其他插件可在工作流中使用的常见枚举类型。	无
HTTP-REST	使您能够通过 vRealize Orchestrator 和 REST 主机之间的交互管理 REST Web 服务。	请参见 配置 HTTP-REST 插件 。
SOAP	通过提供 vRealize Orchestrator 与 SOAP 主机之间的交互，使您可以管理 SOAP Web 服务。	请参见 配置 SOAP 插件 。
AMQP	允许您与高级消息队列协议 (Advanced Message Queuing Protocol, AMQP) 服务器（也称为“代理”）进行交互。	请参见 配置 AMQP 插件 。
SNMP	使 vRealize Orchestrator 可以连接到启用了 SNMP 的系统和设备，并接收它们的信息。	无
Active Directory	提供 vRealize Orchestrator 与 Microsoft Active Directory 之间的交互。	请参见 配置 Active Directory 插件 。
动态类型	允许您定义动态类型并创建和使用这些动态类型的对象。	请参见第 19 章 使用 Dynamic Types 插件 。
多节点	包含用于分层管理、Orchestrator 实例管理和 Orchestrator 活动扩展的工作流。	请参见第 21 章 使用 Multi-Node 插件 。
PowerShell	用于管理 PowerShell 主机并运行自定义 PowerShell 操作。	请参见第 20 章 使用 PowerShell 插件 。

插件组件

每个插件是一个 DAR 文件包。DAR 文件存储在 Orchestrator Appliance 上的 `/var/lib/vco/app-server/plugins` 中。每个插件组件，例如工作流类别和 API 模块，均使用不同的命名约定。

表 1-2. 插件组件的名称

配置 UI 中的插件名称	DAR 文件	工作流类别	API 模块
vCenter Server	<code>o11nplugin-vsphere.dar</code>	vCenter	VC
vRO 配置	<code>o11nplugin-configurator.dar</code>	配置	Configurator

表 1-2. 插件组件的名称（续）

配置 UI 中的插件名称	DAR 文件	工作流类别	API 模块
库	o11nplugin-library.dar	锁定 Orchestrator 故障排除	不适用。
SQL	o11nplugin-database.dar	JDBC SQL	SQL
SSH	o11nplugin-ssh.dar	SSH	SSH
XML	o11nplugin-xml.dar	XML	XML
Mail	o11nplugin-mail.dar	Mail	Mail
Net	o11nplugin-jakartacommonsnet.dar	无	Net
工作流文档	o11nplugin-wfdocs.dar	工作流文档	工作流文档
常用枚举类型	o11nplugin-enums.dar	无	Enums
动态类型	o11n-plugin-dynamictypes.dar	配置	DynamicTypes
HTTP-REST	o11nplugin-rest.dar	配置	REST
SOAP	o11n-plugin-soap.dar	配置	SOAP
AMQP	o11n-plugin-amqp.dar	配置	AMQP
SNMP	o11n-plugin-snmp.dar	设备管理 查询管理 陷阱主机管理	SNMP
Active Directory		计算机 配置 组织单位 用户 用户组	AD
Orchestrator	o11nplugin-multi-node.dar	服务器配置 远程执行 远程管理 任务 工作流	VC0
PowerShell	o11nplugin-powershell.dar	配置 生成 模板	PowerShell

访问 Orchestrator API Explorer

Orchestrator 提供 API Explorer，可用于搜索 Orchestrator API 并查看有关 JavaScript 对象文档，了解可在脚本元素中使用的 JavaScript 对象。

您可以在 Orchestrator 文档主页查看 vCenter Server 插件的在线版脚本 API。

步骤

- 1 登录到 Orchestrator 客户端。
- 2 选择工具 > **API Explorer**。

结果

系统会显示 API Explorer。您可用其搜索 Orchestrator API 的所有对象和函数。

后续步骤

使用 API Explorer 对可编辑脚本元素编写脚本。

配置 Orchestrator 插件

2

默认 Orchestrator 插件仅通过 workflow 进行配置。

如果想要配置任一默认 Orchestrator 插件，需要使用 Orchestrator 客户端相应的工作流。

本章讨论了以下主题：

- [管理 Orchestrator 插件](#)
- [卸载插件](#)

管理 Orchestrator 插件

在控制中心的[管理插件](#)页面中，可以查看 Orchestrator 所安装全部插件的列表，并可执行基本的管理操作。

更改插件日志记录级别

您可以针对特定插件更改日志记录级别，而不用针对 Orchestrator。

安装新插件

使用 Orchestrator 插件，Orchestrator 服务器可以与其他软件产品进行集成。Orchestrator Appliance 包含一组预安装的插件，您也可以安装自定义插件。

所有 Orchestrator 插件都通过控制中心进行安装。可用的文件扩展名包括 `.vmoapp` 和 `.dar`。`.vmoapp` 文件可以包含多个不同的 `.dar` 文件，并可作为应用程序进行安装，而 `.dar` 文件则包含与某一个插件相关的所有资源。

禁用插件

您可取消选中插件名称旁的[启用](#)复选框来禁用插件。

此操作不会移除插件文件。有关在 Orchestrator 中卸载插件的更多信息，请参见[卸载插件](#)。

卸载插件

您可以使用控制中心来禁用插件，但此操作不会将插件文件从 Orchestrator Appliance 文件系统中移除。要移除插件文件，您必须登录 Orchestrator Appliance 并手动移除插件文件。

步骤

1 从 Orchestrator Appliance 删除插件。

- a 使用 SSH 以 **root** 用户身份登录 Orchestrator Appliance。
- b 使用文本编辑器打开 `/etc/vco/app-server/plugins/_VSOPuginInstallationVersion.xml` 文件。
- c 删除与想要移除的插件对应的代码行。
- d 导航到 `/var/lib/vco/app-server/plugins` 目录。
- e 删除其中包含想要移除的插件的 `.dar` 存档。

2 重新启动 vRealize Orchestrator 服务。

```
service vco-configurator restart && service vco-server restart
```

3 以 **root** 用户身份登录控制中心。

4 在**管理插件**页面，验证是否已移除插件。

5 通过 Orchestrator 客户端，删除与该插件相关的软件包和文件夹。

- a 登录到 Orchestrator 客户端。
- b 从位于左上角的下拉菜单中选择**设计**。
- c 单击**软件包**视图。
- d 右键单击要删除的软件包，然后选择**删除元素和内容**。

注 锁定为只读状态的 Orchestrator 元素（例如，标准库中的工作流）不会被删除。

e 从位于右上角的**工具**菜单中，选择**用户首选项**。

首选项上下文菜单将打开。

f 在**常规**页面，选中**允许删除非空文件夹**复选框。

您即可通过单击删除整个文件夹，包括其子文件夹和工作流。

g 单击**工作流**视图。

h 删除您想要移除的插件的文件夹。

i 单击**操作**视图。

j 删除您想要移除的插件的操作模块。

6 重新启动 vRealize Orchestrator 服务。

结果

您即移除了与插件相关的全部自定义工作流、操作、策略、配置、设置和资源。

使用 vCenter Server 插件

3

您可以使用 vCenter Server 插件管理多个 vCenter Server 实例。您可以创建其中使用 vCenter Server 插件 API 的工作流，用于自动处理 vCenter Server 环境中的任务。

vCenter Server 插件将 vCenter Server API 映射到 JavaScript，从而可在工作流中使用。此外，该插件还提供可执行单个 vCenter Server 任务（可包含到工作流中）的操作。

vCenter Server 插件提供了一个标准工作流库，可自动处理 vCenter Server 操作。例如：您可以运行用于创建、迁移或删除虚拟机的工作流。

vCenter Server 插件将基于策略的管理 (PBM) 和存储监控服务 (SMS) API 作为 Orchestrator 脚本 API 中的脚本对象包含在内。基于存储策略的管理策略和组件将显示在 Orchestrator 清单选项卡中。

本章讨论了以下主题：

- [配置 vCenter Server 插件](#)
- [vCenter Server 插件脚本 API](#)
- [使用 vCenter Server 插件清单](#)
- [查询的性能注意事项](#)
- [使用 XPath 表达式与 vCenter Server 插件](#)
- [访问 vCenter Server 插件工作流库](#)
- [vCenter Server 插件工作流库](#)

配置 vCenter Server 插件

在使用 Orchestrator 管理 vSphere 清单中的对象以及对这些对象运行工作流之前，您必须配置 vCenter Server 插件并定义 Orchestrator 与要编排的 vCenter Server 实例之间的连接参数。

您可以从 Orchestrator 客户端中运行 vCenter Server 配置工作流来配置 vCenter Server 插件。

若要使用 vSphere Web Client 管理 vSphere 清单中的对象，请确保将 Orchestrator 服务器配置为能与 vCenter Server 和 vSphere Web Client 指向的同一 vCenter Single Sign-On 实例结合使用。您还必须确保 Orchestrator 已注册为 vCenter Server 扩展。在指定对 vCenter Server 扩展具有管理权限的用户（通过提供用户名和密码）时，Orchestrator 会被注册为 vCenter Server 扩展。

配置工作流

vCenter Server 插件的“配置”工作流类别包含可用于管理 vCenter Server 实例连接的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > vCenter > 配置**访问以下工作流。

工作流名称	描述
添加 vCenter Server 实例	配置 Orchestrator 连接到新的 vCenter Server 实例，以便在 vSphere 基础架构中的对象上运行工作流。
列出 vCenter Server 的 vRealize Orchestrator 扩展	列出 vCenter Server 的所有 vRealize Orchestrator 扩展。
将 Orchestrator 注册为 vCenter Server 扩展	将 Orchestrator 实例注册为 vCenter Server 扩展。
移除 vCenter Server 实例	将 vCenter Server 实例从 Orchestrator 清单中移除。您将无法再编排此 vCenter Server 实例。
更新 vCenter Server 实例	更新 vCenter Server 实例的连接。例如：如果 vCenter Server 系统的 IP 地址发生更改，则必须更新 vCenter Server 实例的连接参数，以便使用 Orchestrator 管理 vSphere 清单。
取消注册 vCenter Server 扩展	取消注册 vSphere Web Client 扩展。

配置 vCenter Server 实例的连接

您可以从 Orchestrator 客户端中运行 vCenter Server 配置工作流来配置 vCenter Server 实例的连接。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > vCenter > 配置**并导航到**添加 vCenter Server 实例**工作流。
- 4 右键单击**添加 vCenter Server 实例**工作流，然后选择**启动工作流**。
- 5 输入安装了要添加的 vCenter Server 实例的计算机 IP 地址或 DNS 名称。

注 输入的主机名需区分大小写。

- 6 保留默认端口值 **443**。
- 7 保留要用来连接 vCenter Server 实例的 SDK 的默认位置。
- 8 选择是否要通过 Orchestrator 管理 vCenter Server 实例，然后单击**下一步**。
- 9 选择是否忽略要添加的 vCenter Server 实例弹出的证书警告。

如果选择忽略证书警告，则系统会在不提示的情况下接受 vCenter Server 实例证书并将其添加到受信任存储。

10 选择 vCenter Server 系统上的用户访问权限管理方法。

选项	描述
共享唯一会话	<p>允许 Orchestrator 仅创建 vCenter Server 的一个连接。</p> <p>在用户名和密码文本框中，输入要用来建立 vCenter Server 主机连接的 Orchestrator 凭据。</p> <p>所选的用户必须是有权管理 vCenter Server 扩展和一组自定义特权的有效用户。Orchestrator 会使用这些凭据监控 VirtualCenter Web 服务，这些凭据通常用于运行 Orchestrator 系统工作流。</p>
单用户会话	<p>创建 vCenter Server 的新会话。此操作可能会迅速占用 CPU、内存和带宽。</p> <p>仅在 vCenter Server 是 Active Directory 域或启用了 vCenter Server Single Sign-On 时选择该选项。</p> <p>所选的用户必须是有权管理 vCenter Server 扩展的有效用户。</p>

所选的用户帐户也会被策略引擎用来收集统计信息和其他数据。如果所选的用户权限不足，则策略引擎无法访问 vCenter Server 清单必要的部分，并且无法收集必要的信息。

11 （可选） 输入用户域。

只有选择使用共享会话时，才必须指定用户域名。

注 如果所选的是单用户会话，请填写此文本框。

12 （可选） 输入 vSphere 存储管理端点的 URL。

您可以配置基于策略的管理 (PBM) 端点、存储监控服务 (SMS) 端点或同时配置两者。

13 单击提交以运行工作流。**结果**

工作流成功运行后，属于该工作流的 vCenter Server 实例和所有 vSphere 对象会显示在清单视图中。

vCenter Server 插件脚本 API

vCenter Server 脚本 API 中包含各种各样的类及其各自属性、方法以及允许 vRealize Orchestrator 和 vCenter Server 之间进行交互的构造函数。您可以使用 API 来开发自定义工作流。

有关可用 API 对象的列表，请参见 <https://www.vmware.com/support/orchestrator/doc/vro-vsphere65-api/index.html>。

使用 vCenter Server 插件清单

vCenter Server 插件会在清单视图中显示所连接的 vCenter Server 实例中的所有对象。您可以使用清单选项卡添加授权元素或在 vCenter Server 对象上运行工作流。

如果在用户首选项工具的清单选项卡中启用在清单中使用上下文菜单选项，则系统会在弹出菜单中显示用户可在选定清单对象上运行的所有工作流。

查询的性能注意事项

通过适用于 vRealize Orchestrator 的 vCenter Server 插件，您可以查询特定对象的 vCenter Server 清单。

查询方法

您可以使用 `vcSearchIndex` 受管对象或包含在插件清单中的对象查找器方法（例如 `getAllDatastores()`、`getAllVirtualMachines()`、`findAllForType()` 等）进行查询。

性能

默认情况下，这两种方法均会返回查询的对象而不包括它们的任何属性，除非您指定一组属性作为搜索查询中方法参数的参数。

注 您必须始终将查询表达式与 `getAll...` 和 `findAll...` 查找器对象一起使用，以免 Orchestrator 客户端筛选大量返回对象，而这可能影响 Orchestrator 服务器的整体性能。

您可将两种类型的表达式用于查询 vCenter Server 清单。

表达式的类型	描述
表达式命名	您可以指定一个名称作为查询参数的参数。 注 根据 vCenter Server 插件清单中显示的插件对象名称按指定的名称参数进行对象筛选。
XPath 表达式	您可以使用基于 XPath 查询语言的表达式。有关详细信息，请参见 使用 XPath 表达式与 vCenter Server 插件 。

当您调用带有自定义属性的 vCenter Server 清单对象时，此对象在工作流或操作中的每个引用都将向 vCenter Server 发送查询，由此会产生明显的性能开销。要优化性能并避免在工作流运行中多次序列化和反序列化对象，最好使用共享资源存储该对象，而不是将其存储为工作流属性、输入参数或输出参数。此类共享资源可以是某个配置元素或资源元素。

使用 XPath 表达式与 vCenter Server 插件

您可以在 vCenter Server 插件中使用查找器方法来查询 vCenter Server 清单对象。您可以使用 XPath 表达式来定义搜索参数。

vCenter Server 插件包含一组对象查找器方法，例如：`getAllDatastores()`、`getAllResourcePools()`、`findAllForType()`。您可以使用这些方法来访问连接到 Orchestrator 服务器的 vCenter Server 实例的清单，并按 ID、名称或其他属性搜索对象。

出于性能考虑，查找器方法不返回查询对象的任何属性，除非您在搜索查询中指定一组属性。

您可以在 Orchestrator 文档主页参考在线版本的 vCenter Server 插件脚本 API。

重要事项 基于 XPath 表达式的查询可能会影响 Orchestrator 性能，因为查找器对象会返回 vCenter Server 端给定类型的所有对象，并将查询筛选器应用到 vCenter Server 插件端。

使用 XPath 表达式与 vCenter Server 插件

调用查找器方法时，可以使用基于 XPath 查询语言的表达式。搜索将返回与 XPath 表达式匹配的所有清单对象。如果想查询任何属性，可以将其以字符串数组形式包含到搜索脚本中。

以下 JavaScript 示例使用了 VcPlugin 脚本对象以及 XPath 表达式，用于返回 vCenter Server 所属受管对象的所有数据存储对象的名称，并且在其名称中包含字符串 **ds**。

```
var datastores = VcPlugin.getAllDatastores(null, "xpath:name[contains(.,'ds')]");
for each (datastore in datastores){
    System.log(datastore.name);
}
```

使用 Server 脚本对象和 findAllForType 查找器方法可以调用同一 XPath 表达式。

```
var datastores = Server.findAllForType("VC:Datastore", "xpath:name[contains(.,'ds')]");
for each (datastore in datastores){
    System.log(datastore.name);
}
```

以下脚本示例会返回其 ID 以数字 **1** 开头的所有主机系统对象的名称。

```
var hosts = VcPlugin.getAllHostSystems(null, "xpath:id[starts-with(.,'1')]");
for each (host in hosts){
    System.log(host.name);
}
```

以下脚本会返回其名称中包含字符串 **DC**（无论大写或小写）的所有数据中心对象的名称和 ID。该脚本还可检索 **标记** 属性。

```
var datacenters = VcPlugin.getAllDatacenters(['tag'], "xpath:name[contains(translate(., 'DC', 'dc'), 'dc')]");
for each (datacenter in datacenters){
    System.log(datacenter.name + " " + datacenter.id);
}
```

访问 vCenter Server 插件工作流库

您必须使用 Orchestrator 客户端或 vSphere Web Client 访问 vCenter Server 插件工作流库中的元素。

前提条件

- 配置 vCenter Server 实例的连接。
- 验证您用来登录的用户帐户是否拥有运行 vCenter Server 工作流的必要权限。

步骤

- 1 在 Orchestrator 客户端中，从左上角的下拉菜单中选择 **设计** 或 **运行**。
- 2 单击 Orchestrator 客户端左窗格中的 **工作流** 视图。
- 3 在层次结构列表中，展开 **库 > vCenter**。

后续步骤

查看工作流库。

vCenter Server 插件工作流库

vCenter Server 插件工作流库中包含可用于运行 vCenter Server 管理相关的自动化流程的工作流。

- **批处理工作流**

批处理工作流会填充配置元素或在选定 vCenter Server 对象上运行工作流。

- **群集和计算资源工作流**

使用群集和计算资源工作流，可以创建、重命名或删除某个群集。您还可以在群集上启用或禁用高可用性、Distributed Resource Scheduler 和 vCloud Distributed Storage。

- **配置工作流**

vCenter Server 插件的“配置”工作流类别包含可用于管理 vCenter Server 实例连接的工作流。

- **自定义属性工作流**

使用自定义属性工作流，可以将自定义属性添加到虚拟机，或获取虚拟机的自定义属性。

- **数据中心工作流**

使用数据中心工作流，可以创建、删除、重新加载、重命名或重新扫描数据中心。

- **数据存储和文件工作流**

使用数据存储和文件工作流，可以删除一系列文件、查找数据存储中未使用的文件等。

- **数据中心文件夹管理工作流**

使用数据中心文件夹管理工作流，可以创建、删除或重命名数据中心文件夹。

- **主机文件夹管理工作流**

使用主机文件夹管理工作流，可以创建、删除或重命名主机文件夹。

- **虚拟机文件夹管理工作流**

使用虚拟机文件夹管理工作流，可以创建、删除或重命名虚拟机文件夹。

- **客户机操作文件工作流**

您可以借助客户机操作文件工作流，在客户机操作系统中管理文件。

- **客户机操作进程工作流**

借助客户机操作进程工作流，您可以在客户机操作系统中获取信息并控制运行的进程。

- **主机电源管理工作流**

使用主机电源管理工作流，可以重新引导或关闭主机。

- **基本主机管理工作流**

使用基本主机管理工作流，您可以将主机置于维护模式以及将主机退出维护模式。您还可以将一个主机移动到文件夹或群集，并从主机重新加载数据。

- [主机注册管理工作流](#)

使用主机注册管理工作流，可以将主机添加到群集、将主机与群集断开连接或重新连接等等。

- [网络工作流](#)

使用网络工作流，可以向分布式虚拟交换机添加端口组、使用端口组创建分布式虚拟机交换机等。

- [分布式虚拟端口组工作流](#)

使用分布式虚拟端口组工作流，可以更新或删除端口组，以及重新配置端口组。

- [分布式虚拟交换机工作流](#)

使用分布式虚拟交换机工作流，可以创建、更新或删除分布式虚拟交换机，以及创建、删除或更新专用 VLAN。

- [标准虚拟交换机工作流](#)

使用标准虚拟交换机工作流，可以创建、更新或删除标准虚拟交换机，以及创建、删除或更新标准虚拟交换机中的端口组。

- [网络 Virtual SAN 工作流](#)

通过 Virtual SAN 工作流，您可以配置 Virtual SAN 网络流量。

- [资源池工作流](#)

使用资源池工作流，可以创建、重命名、重新配置或删除资源池，以及获取资源池信息。

- [存储工作流](#)

使用存储工作流，可以执行与存储相关的操作。

- [Storage DRS 工作流](#)

使用 Storage DRS 工作流，可以执行与存储相关的操作，如创建和配置数据存储群集、从群集移除数据存储、向群集添加存储等。

- [Storage VSAN 工作流](#)

使用 Virtual SAN 工作流，您可以管理 Virtual SAN 群集中的非 SSD 磁盘和磁盘组。

- [基本虚拟机管理工作流](#)

使用基本虚拟机管理工作流，可以对虚拟机执行基本操作，例如创建、重命名或删除虚拟机，升级虚拟硬件等。

- [克隆工作流](#)

使用克隆工作流，可以选择克隆虚拟机时自定义或不自定义虚拟机属性。

- [链接克隆工作流](#)

使用链接克隆工作流，可以执行链接克隆操作（例如，从链接克隆还原虚拟机、创建链接克隆等）。

- [Linux 自定义克隆工作流](#)

使用 Linux 自定义工作流，可以克隆 Linux 虚拟机并自定义客户机操作系统。

- [工具克隆工作流](#)

使用工具克隆工作流，可以获取有关虚拟机操作系统的自定义信息、更新虚拟设备所需的信息等。

■ Windows 自定义克隆工作流

使用 Windows 自定义克隆工作流，可以克隆 Windows 虚拟机和自定义客户机操作系统。

■ 设备管理工作流

使用设备管理工作流，可以管理连接到虚拟机或主机数据存储的设备。

■ 移动和迁移工作流

通过移动和迁移工作流，可以迁移虚拟机。

■ 其他工作流

使用“其他”类别中的工作流，可以启用和禁用容错 (Fault Tolerance, FT)、提取虚拟机信息，以及查找孤立的虚拟机。

■ 电源管理工作流

使用电源管理工作流，可以打开和关闭虚拟机电源、重新引导虚拟机的客户机操作系统以及挂起虚拟机等。

■ 快照工作流

通过快照工作流，可以执行与快照相关的操作。

■ VMware Tools 工作流

使用 VMware Tools 工作流，可以在虚拟机上执行 VMware Tools 相关的任务。

批处理工作流

批处理工作流会填充配置元素或在选定 vCenter Server 对象上运行工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > vCenter > 批处理**访问批处理工作流。

工作流名称	描述
填写批处理配置元素	填充“在选择的对象上运行工作流”工作流所用的配置元素。执行以下任务： <ul style="list-style-type: none"> ■ 重置 BatchObject 和 BatchAction 配置元素。 ■ 在 BatchObject 配置元素中填入仅包含一个输入参数的所有工作流。 ■ 在 BatchAction 配置元素中填入不含或包含一个输入参数且存在一个 returnType 数组的所有操作。
在选择的对象上运行工作流	在选择的 vCenter Server 对象上运行工作流，期间一个操作会视为输入。此操作会检索在其上运行工作流的对象的列表。若要返回对象而不运行选定工作流，请在模拟模式下运行工作流。

群集和计算资源工作流

使用群集和计算资源工作流，可以创建、重命名或删除某个群集。您还可以在群集上启用或禁用高可用性、Distributed Resource Scheduler 和 vCloud Distributed Storage。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > vCenter > 群集和计算资源**访问群集和计算资源工作流。

工作流名称	描述
将 DRS 虚拟机组添加到群集	将 DRS 虚拟机组添加到群集。
将虚拟机添加到 DRS 组	将虚拟机列表添加到现有 DRS 虚拟机组。

工作流名称	描述
创建群集	在主机文件夹中创建群集。
删除群集	删除群集。
在群集上禁用 DRS	在群集上禁用 DRS。
在群集上禁用 HA	在群集上禁用高可用性。
在群集上禁用 vCloud Distributed Storage	在群集上禁用 vCloud Distributed Storage。
在群集上启用 DRS	在群集上启用 DRS。
在群集上启用 HA	在群集上启用高可用性。
在群集上启用 vCloud Distributed Storage	在群集上启用 vCloud Distributed Storage。
将虚拟机 DRS 组从群集中移除	将 DRS 虚拟机组从群集中移除。
将虚拟机从 DRS 组中移除	将虚拟机从群集 DRS 组中移除。
重命名群集	重命名群集。

配置工作流

vCenter Server 插件的“配置”工作流类别包含可用于管理 vCenter Server 实例连接的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > vCenter > 配置**访问以下工作流。

工作流名称	描述
添加 vCenter Server 实例	配置 Orchestrator 连接到新的 vCenter Server 实例，以便在 vSphere 基础架构中的对象上运行工作流。
列出 vCenter Server 的 Orchestrator 扩展	列出 vCenter Server 的所有 Orchestrator 扩展。
将 Orchestrator 注册为 vCenter Server 扩展	将 Orchestrator 实例注册为 vCenter Server 扩展。
移除 vCenter Server 实例	将 vCenter Server 实例从 Orchestrator 清单中移除。您将无法再编排此 vCenter Server 实例。
更新 vCenter Server 实例	更新 vCenter Server 实例的连接。例如：如果 vCenter Server 系统的 IP 地址发生更改，则必须更新 vCenter Server 实例的连接参数，以便使用 Orchestrator 管理 vSphere 清单。
取消注册 vCenter Server 扩展	取消注册 vCenter Server 扩展。

自定义属性工作流

使用自定义属性工作流，可以将自定义属性添加到虚拟机，或获取虚拟机的自定义属性。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > vCenter > 自定义属性**访问自定义属性工作流。

工作流名称	描述
将自定义属性添加到一台虚拟机	将自定义属性添加到一台虚拟机。
将自定义属性添加到多台虚拟机	将自定义属性添加到所选的多台虚拟机。
获取自定义属性	在 vCenter Server 中获取虚拟机的自定义属性。

数据中心工作流

使用数据中心工作流，可以创建、删除、重新加载、重命名或重新扫描数据中心。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 数据中心**访问数据中心工作流。

工作流名称	描述
创建数据中心	在数据中心文件夹中创建数据中心。
删除数据中心	删除数据中心。
重新加载数据中心	强制 vCenter Server 从数据中心重新加载数据。
重命名数据中心	重命名数据中心并等待任务完成。
重新扫描数据中心 HBA	扫描数据中心中的主机，然后对主机总线适配器启动重新扫描以发现新存储器。

数据存储和文件工作流

使用数据存储和文件工作流，可以删除一系列文件、查找数据存储中未使用的文件等。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 数据存储和文件**访问数据存储和文件工作流。

工作流名称	描述
删除所有文件	删除一系列文件。
删除所有未使用的数据存储文件	在 vCenter Server 环境中搜索所有数据存储并删除所有未使用的文件。
导出未使用的数据存储文件	搜索所有数据存储并创建列有所有未使用文件的 XML 描述符文件。
查找数据存储中未使用的文件	搜索 vCenter Server 环境中与 Orchestrator 中注册的任何 vCenter Server 实例都不关联的所有未使用磁盘 (*.vmdk)、虚拟机 (*.vmx) 和模板 (*.vmtx) 文件。
从虚拟机获取所有配置、模板和磁盘文件	为所有数据存储创建两张列表，一张列表包含所有的虚拟机描述符文件，另一张列表包含所有的虚拟机磁盘文件。
记录所有数据存储文件	为在所有数据存储中找到的每个虚拟机配置文件和虚拟机文件创建日志。
记录未使用的数据存储文件	搜索 vCenter Server 环境中虚拟机上注册的未使用文件，并以文本文件格式导出文件日志。
将文件上传至数据存储	将文件上传到特定数据存储上的现有文件夹。上传的文件会覆盖同一目标文件夹中的任何同名文件。

数据中心文件夹管理工作流

使用数据中心文件夹管理工作流，可以创建、删除或重命名数据中心文件夹。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 文件夹管理 > 数据中心文件夹**访问数据中心文件夹管理工作流。

工作流名称	描述
创建数据中心文件夹	创建数据中心文件夹。
删除数据中心文件夹	删除数据中心文件夹并等待任务完成。
重命名数据中心文件夹	重命名数据中心文件夹并等待任务完成。

主机文件夹管理工作流

使用主机文件夹管理工作流，可以创建、删除或重命名主机文件夹。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 文件夹管理 > 主机文件夹访问主机文件夹管理工作流**。

工作流名称	描述
创建主机文件夹	创建主机文件夹。
删除主机文件夹	删除主机文件夹并等待任务完成。
重命名主机文件夹	重命名主机文件夹并等待任务完成。

虚拟机文件夹管理工作流

使用虚拟机文件夹管理工作流，可以创建、删除或重命名虚拟机文件夹。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 文件夹管理 > 虚拟机文件夹访问虚拟机文件夹管理工作流**。

工作流名称	描述
创建虚拟机文件夹	创建虚拟机文件夹。
删除虚拟机文件夹	删除虚拟机文件夹并等待任务完成。
重命名虚拟机文件夹	重命名虚拟机文件夹并等待任务完成。

客户机操作文件工作流

您可以借助客户机操作文件工作流，在客户机操作系统中管理文件。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 客户机操作 > 文件访问客户机操作文件工作流**。

工作流名称	描述
检查客户机中的目录	验证客户机虚拟机中是否存在目录。
检查客户机中的文件	验证客户机虚拟机中是否存在文件。
将文件从客户机复制到 Orchestrator	将指定的文件从客户机文件系统复制到 Orchestrator 服务器上。
将文件从 Orchestrator 复制到客户机中	将指定文件从 Orchestrator 服务器复制到客户机文件系统。
在客户机中创建目录	在客户机虚拟机中创建目录。
在客户机中创建临时目录	在客户机虚拟机中创建临时目录。
在客户机中创建临时文件	在客户机虚拟机中创建临时文件。
删除客户机中的目录	将目录从客户机虚拟机中删除。
删除客户机中的文件	将文件从客户机虚拟机中删除。
在客户机中列出路径	在客户机虚拟机中显示路径。
在客户机中移动目录	在客户机虚拟机中移动目录。
在客户机中移动文件	在客户机虚拟机中移动文件。

客户机操作进程工作流

借助客户机操作进程工作流，您可以在客户机操作系统中获取信息并控制运行的进程。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 客户机操作 > 进程访问客户机操作文件工作流。

工作流名称	描述
从客户机获取环境变量	从客户机返回带有环境变量的列表。交互会话返回目前已登录的用户的变量。
从客户机获取进程	返回一份列有客户机操作系统中运行的进程以及最近由 API 启动的已完成进程的列表。
在客户机中运行程序	在客户机操作系统中启动程序。
在客户机中终止进程	在客户机操作系统中终止进程。

主机电源管理工作流

使用主机电源管理工作流，可以重新引导或关闭主机。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 主机管理 > 电源访问主机电源管理工作流。

重新引导主机

重新引导主机。如果 Orchestrator 客户端直接与主机相连，则它会断开与该主机的连接，并且不会在返回的任务中收到成功的指示。

关闭主机

关闭主机。如果 Orchestrator 客户端直接与主机相连，则它会断开与该主机的连接，并且不会在返回的任务中收到成功的指示。

基本主机管理工作流

使用基本主机管理工作流，您可以将主机置于维护模式以及将主机退出维护模式。您还可以将一个主机移动到文件夹或群集，并从主机重新加载数据。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 主机管理 > 基本访问基本主机管理工作流。

工作流名称	描述
进入维护模式	将主机置于维护模式。您可以取消该任务。
退出维护模式	退出维护模式。您可以取消该任务。
将主机移至群集	将现有主机移到群集中。该主机必须属于同一数据中心，如果该主机属于某个群集，则必须处于维护模式。
将主机移至文件夹	将主机作为独立主机移到文件夹中。该主机必须属于同一数据中心内的 ClusterComputeResource，且主机必须处于维护模式。
重新加载主机	强制 vCenter Server 从主机重新加载数据。

主机注册管理工作流

使用主机注册管理工作流，可以将主机添加到群集、将主机与群集断开连接或重新连接等等。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 主机管理 > 注册**访问主机管理注册工作流。

工作流名称	描述
将主机添加到群集	将主机添加到群集。如果无法对主机的 SSL 证书进行身份验证，则该工作流将失败。
添加独立主机	将主机作为独立主机进行注册。
断开主机	将主机与 vCenter Server 实例断开连接。
重新连接主机	通过仅提供主机信息重新连接已断开连接的主机。
通过所有信息重新连接主机	通过提供与主机有关的所有信息重新连接已断开连接的主机。
移除主机	从 vCenter Server 实例移除主机并取消注册。如果该主机为群集的一部分，则必须在尝试移除前使其处于维护模式。

网络工作流

使用网络工作流，可以向分布式虚拟交换机添加端口组、使用端口组创建分布式虚拟机交换机等。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 网络**访问网络工作流。

工作流名称	描述
将端口组添加到分布式虚拟交换机	将新的分布式虚拟端口组添加到指定的分布式虚拟交换机。
将主机系统附加到分布式虚拟交换机	将主机添加到分布式虚拟交换机。
创建含端口组的分布式虚拟交换机	创建含分布式虚拟端口组的新分布式虚拟交换机。

分布式虚拟端口组工作流

使用分布式虚拟端口组工作流，可以更新或删除端口组，以及重新配置端口组。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 网络 > 分布式虚拟端口组**访问分布式虚拟端口组工作流。

工作流名称	描述
将虚拟机网卡编号与分布式虚拟端口组相连接	重新配置与指定分布式虚拟端口组相连的指定虚拟机网卡编号的网络连接。如果未指定网卡编号，则将使用“零”。
删除分布式虚拟端口组	删除指定的分布式虚拟端口组。
设置成组选项	提供用于管理分布式虚拟端口组的成组选项的接口。
更新分布式虚拟端口组	更新指定的分布式虚拟端口组的配置。

分布式虚拟交换机工作流

使用分布式虚拟交换机工作流，可以创建、更新或删除分布式虚拟交换机，以及创建、删除或更新专用 VLAN。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 网络 > 分布式虚拟交换机**访问分布式虚拟交换机工作流。

工作流名称	描述
创建分布式虚拟交换机	通过指定的名称和上行链路端口名称在指定的网络文件夹中创建分布式虚拟交换机。必须至少指定一个上行链路端口名称。
创建专用 VLAN	在指定的分布式虚拟交换机上创建 VLAN。
删除分布式虚拟交换机	删除分布式虚拟交换机和所有关联的元素。
删除专用 VLAN	将 VLAN 从指定分布式虚拟交换机中删除。如果存在辅助 VLAN，必须首先删除辅助 VLAN。
更新分布式虚拟交换机	更新分布式虚拟交换机的属性。
更新专用 VLAN	在指定的分布式虚拟交换机上更新 VLAN。

标准虚拟交换机工作流

使用标准虚拟交换机工作流，可以创建、更新或删除标准虚拟交换机，以及创建、删除或更新标准虚拟交换机中的端口组。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 网络 > 标准虚拟交换机**访问标准虚拟交换机工作流。

工作流名称	描述
在标准虚拟交换机中添加端口组	在标准虚拟交换机中添加端口组。
创建标准虚拟交换机	创建标准虚拟交换机。
将端口组从标准虚拟交换机中删除	将端口组从标准虚拟交换机中删除。
删除标准虚拟交换机	将标准虚拟交换机从主机网络配置中删除。
检索所有标准虚拟交换机	从主机中检索所有标准虚拟交换机。
更新标准虚拟交换机中的端口组	更新标准虚拟交换机中端口组的属性。
更新标准虚拟交换机	更新标准虚拟交换机的属性。
更新标准虚拟交换机中端口组的虚拟网卡	更新标准虚拟交换机中端口组相关联的虚拟网卡。

网络 Virtual SAN 工作流

通过 Virtual SAN 工作流，您可以配置 Virtual SAN 网络流量。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 网络 > VSAN**访问网络工作流。

工作流名称	描述
设置集群的 VSAN 流量网络	设置集群的 Virtual SAN 流量网络。
设置主机的 VSAN 流量网络	设置主机的 Virtual SAN 流量网络。

资源池 workflow

使用资源池 workflow，可以创建、重命名、重新配置或删除资源池，以及获取资源池信息。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > vCenter > 资源池** 访问资源池 workflow。

workflow 名称	描述
创建资源池	通过默认的 CPU 和内存分配值创建资源池。要在群集中创建资源池，群集必须启用 VMware DRS。
通过指定的值创建资源池	通过指定的 CPU 和内存分配值创建资源池。要在群集中创建资源池，群集必须启用 VMware DRS。
删除资源池	删除资源池并等待任务完成。
获取资源池信息	返回给定资源池相关的 CPU 和内存信息。
重新配置资源池	重新配置给定资源池的 CPU 和内存分配配置。
重命名资源池	重命名资源池并等待任务完成

存储 workflow

使用存储 workflow，可以执行与存储相关的操作。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > vCenter > 存储** 访问存储 workflow。

workflow 名称	描述
在 iSCSI/FC/本地 SCSI 上添加数据存储	在光纤通道、iSCSI 或本地 SCSI 磁盘创建数据存储。只有当前未用于现有 VMFS 的磁盘才适用于新数据存储的创建。新的数据存储将分配指定磁盘的最大可用空间。
在 NFS 上添加数据存储	在 NFS 服务器上添加数据存储。
添加 iSCSI 目标	将 iSCSI 目标添加到 vCenter Server 主机。目标的类型可以是 Send 或 Static 。
为所有可用磁盘创建 VMFS	为指定主机的所有可用磁盘创建 VMFS 卷。
删除数据存储	将数据存储从 vCenter Server 主机中删除。
删除 iSCSI 目标	删除已配置的 iSCSI 目标。目标的类型可以是 Send 或 Static 。
禁用 iSCSI 适配器	禁用指定主机的软件 iSCSI 适配器。
显示所有数据存储和磁盘	显示指定主机上的现有数据存储和可用磁盘。
启用 iSCSI 适配器	启用 iSCSI 适配器。
列出所有存储适配器	列出指定主机的所有存储适配器。

Storage DRS workflow

使用 Storage DRS workflow，可以执行与存储相关的操作，如创建和配置数据存储群集、从群集移除数据存储、向群集添加存储等。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > vCenter > 存储 > 存储 DRS** 访问 Storage DRS workflow。

工作流名称	描述
将数据存储添加到群集	将数据存储添加到群集。如要添加到数据存储群集中，数据存储必须能连接所有主机。数据存储必须具有相同的连接类型，才能驻留在数据存储群集中。
更改每一虚拟机配置的 Storage DRS	为每台虚拟机设定 Storage DRS 设置。
配置数据存储群集	配置数据存储群集的自动化和运行时规则设置值。
创建简单的数据存储群集	通过默认配置创建简单的数据存储群集。新数据存储群集中不包含任何数据存储。
创建 Storage DRS 已调度任务	创建调度任务以重新配置数据存储群集。仅可以设置自动化和运行时规则。
创建虚拟机反关联性规则	创建反关联性规则，以指示某些虚拟机的所有虚拟磁盘必须保留在不同的数据存储上。
创建 VMDK 反关联性规则	为虚拟机创建 VMDK 反关联性规则，指示虚拟机的哪些虚拟磁盘必须保留在不同的数据存储上。该规则适用于选定虚拟机的虚拟磁盘。
移除数据存储群集	移除数据存储群集。移除数据存储群集将同时移除 vCenter Server 系统中的所有设置和群集警报。
将数据存储从群集中移除	将数据存储从数据存储群集中移除并将数据存储放入数据存储文件夹。
移除 Storage DRS 已调度任务	移除已调度的 Storage DRS 任务。
移除虚拟机反关联性规则	移除给定数据存储群集的虚拟机反关联性规则。
移除 VMDK 反关联性规则	移除给定数据存储群集的 VMDK 反关联性规则。

Storage VSAN 工作流

使用 Virtual SAN 工作流，您可以管理 Virtual SAN 群集中的非 SSD 磁盘和磁盘组。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 存储 > VSAN** 访问网络工作流。

工作流名称	描述
将磁盘添加到磁盘组	将非 SSD 磁盘添加到 Virtual SAN 磁盘组。
将磁盘声明至磁盘组	声明将由 Virtual SAN 系统使用的磁盘并自动创建磁盘组，然后将磁盘分发到现有磁盘组中。
创建磁盘组	创建 Virtual SAN 磁盘组。
列出主机、磁盘组和磁盘	列出群集中的所有主机及其磁盘组和磁盘（Virtual SAN 系统已用或可用）。
移除磁盘组	移除 Virtual SAN 磁盘组。
将磁盘从磁盘组中移除	将非 SSD 磁盘从 Virtual SAN 磁盘组中移除。

基本虚拟机管理工作流

使用基本虚拟机管理工作流，可以对虚拟机执行基本操作，例如创建、重命名或删除虚拟机，升级虚拟硬件等。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 虚拟机管理 > 基本**访问基本虚拟机管理工作流。

工作流名称	描述
创建自定义虚拟机	通过指定的配置选项和其他设备创建虚拟机。
创建简单的 dvPortGroup 虚拟机	创建简单的虚拟机。所使用的网络是分布式虚拟端口组。

工作流名称	描述
创建简单的虚拟机	通过最常用的设备和配置选项创建虚拟机。
删除虚拟机	将虚拟机从清单和数据存储中移除。
按名称获取虚拟机	返回一份来自与所提供表达式匹配的所有注册 vCenter Server 实例的虚拟机列表。
标记为模板	将现有虚拟机转换为模板并禁止其启动。可使用模板创建虚拟机。
标记为虚拟机	将现有模板转换为虚拟机并允许其启动。
将虚拟机移至文件夹	将虚拟机移至指定的虚拟机文件夹。
将虚拟机移至资源池	将虚拟机移至资源池。如果目标资源池不在同一群集中，则必须使用迁移或重定位工作流。
将虚拟机移至文件夹	将若干虚拟机移至一个指定的虚拟机文件夹。
将虚拟机移至资源池	将若干虚拟机移至一个资源池。
注册虚拟机	注册虚拟机。虚拟机文件必须置于现有数据存储中且不得处于已注册状态。
重新加载虚拟机	强制 vCenter Server 重新加载虚拟机。
重命名虚拟机	重命名 vCenter Server 系统或主机（而不是数据存储）上的现有虚拟机。
设置虚拟机性能	更改性能设置，例如虚拟机的共享、最小值和最大值、网络调整和磁盘访问。
取消注册虚拟机	将现有虚拟机从清单中移除。
升级虚拟机硬件（需要时可强制执行）	将虚拟机硬件升级至主机支持的最新版本。该工作流可在 VMware Tools 已过期的情况下强制继续升级。如果 VMware Tools 已过期，则强制继续升级会使客户机网络设置恢复为默认设置。为避免发生这种情况，应在运行工作流前升级 VMware Tools。
升级虚拟机	将虚拟硬件升级至主机支持的最新版本。输入参数可在 VMware Tools 已过期的情况下强制执行升级。
等待任务并应答虚拟机问题	等待 vCenter Server 任务完成或等待虚拟机提问。如果虚拟机要求应答，则接受用户输入并应答问题。

克隆工作流

使用克隆工作流，可以选择克隆虚拟机时自定义或不自定义虚拟机属性。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 虚拟机管理 > 克隆访问克隆工作流。

工作流名称	描述
通过属性克隆虚拟机	通过将属性用作输入参数来克隆虚拟机。
克隆虚拟机，但不进行自定义	克隆虚拟机，但不更改虚拟机 UUID 以外的任何内容。
通过属性自定义虚拟机	通过将属性用作输入参数来自定义虚拟机。

链接克隆工作流

使用链接克隆工作流，可以执行链接克隆操作（例如，从链接克隆还原虚拟机、创建链接克隆等）。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 虚拟机管理 > 克隆 > 链接克隆文件夹及其子文件夹访问链接克隆工作流。

工作流名称	描述
通过链接克隆还原虚拟机	将虚拟机从链接克隆设置中移除。
为链接克隆设置虚拟机	准备要进行链接克隆的虚拟机。
创建具有多个网卡的 Linux 虚拟机的链接克隆	创建 Linux 虚拟机的链接克隆，执行客户机操作系统自定义，并可最多配置四个虚拟网卡。
创建具有单个网卡的 Linux 虚拟机的链接克隆	创建 Linux 虚拟机的链接克隆，执行客户机操作系统自定义，并配置一个虚拟网卡。
创建具有多个网卡和单个凭据的 Windows 虚拟机的链接克隆	创建 Windows 虚拟机的链接克隆，并执行客户机操作系统自定义。配置最多四个虚拟网卡和一个本地管理员用户帐户。
创建具有单个网卡和单个凭据的 Windows 虚拟机的链接克隆	创建 Windows 虚拟机的链接克隆，并执行客户机操作系统自定义。配置一个虚拟网卡和一个本地管理员用户帐户。
创建链接克隆而不进行自定义	创建指定数目的虚拟机链接克隆。

Linux 自定义克隆工作流

使用 Linux 自定义工作流，可以克隆 Linux 虚拟机并自定义客户机操作系统。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 虚拟机管理 > 克隆 > Linux 自定义访问 Linux 自定义克隆工作流。

工作流名称	描述
克隆具有多个网卡的 Linux 虚拟机	克隆一台 Linux 虚拟机，执行客户机操作系统自定义，并可最多配置四个虚拟网卡。
克隆具有单个网卡的 Linux 虚拟机	克隆一台 Linux 虚拟机，执行客户机操作系统自定义，并配置一个虚拟网卡。

工具克隆工作流

使用工具克隆工作流，可以获取有关虚拟机操作系统的自定义信息、更新虚拟设备所需的信息等。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 虚拟机管理 > 克隆 > 工具访问工具克隆工作流。

工作流名称	描述
获取用于更改网络的虚拟以太网卡	返回用于更新虚拟设备的新以太网卡。仅包含给定虚拟设备的设备密钥和新网络。
获取 Linux 自定义	返回 Linux 自定义准备。
获取多个虚拟以太网卡设备变更	返回一系列需要在 VirtualEthernetCard 对象上进行添加和移除操作的 VirtualDeviceConfigSpec 对象。
获取网卡设置映射	使用 VimAdapterMapping 后系统会返回虚拟网卡的设置映射。
通过凭据获取 Syspre 的 Windows 自定义	通过凭据返回有关 Microsoft Sysprep 进程的自定义信息。用于克隆 Windows 虚拟机的工作流会使用该工作流。
通过 Unattended.txt 获取 Sysprep 的 Windows 自定义	使用 Unattended.txt 文件后，系统会返回有关 Microsoft Sysprep 进程的自定义信息。用于克隆 Windows 虚拟机的工作流会使用该工作流。
获取 Sysprep 的 Windows 自定义	返回有关 Microsoft Sysprep 进程的自定义信息。用于克隆 Windows 虚拟机的工作流会使用该工作流。

Windows 自定义克隆 workflow

使用 Windows 自定义克隆 workflow，可以克隆 Windows 虚拟机和自定义客户机操作系统。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > vCenter > 虚拟机管理 > 克隆 > Windows 自定义** 文件夹及其子文件夹访问 Windows 自定义克隆 workflow。

workflow 名称	描述
自定义具有单个网卡和凭据的 Windows 虚拟机	在 Windows 虚拟机上执行客户机操作系统自定义，并配置一个虚拟网卡和一个本地管理员用户帐户。
克隆具有单个网卡和凭据的精简置备的 Windows 虚拟机	克隆执行客户机操作系统自定义的 Windows 虚拟机。配置一个虚拟网卡和一个本地管理员用户帐户。vCenter Server 上必须装有 Sysprep 工具。
克隆具有单个网卡和凭据的 Windows 虚拟机 Sysprep	克隆执行客户机操作系统自定义的 Windows 虚拟机。配置一个虚拟网卡和一个本地管理员用户帐户。vCenter Server 上必须装有 Sysprep 工具。
克隆具有多个网卡和单个凭据的 Windows 虚拟机	克隆执行客户机操作系统自定义的 Windows 虚拟机。配置本地管理员用户帐户和最多四个虚拟网卡。vCenter Server 系统上必须装有 Sysprep 工具。
克隆具有单个网卡的 Windows 虚拟机	克隆执行客户机操作系统自定义的 Windows 虚拟机并配置一个虚拟网卡。vCenter Server 系统上必须装有 Sysprep 工具。
克隆具有单个网卡和凭据的 Windows 虚拟机	克隆执行客户机操作系统自定义的 Windows 虚拟机。配置一个虚拟网卡和一个本地管理员用户帐户。vCenter Server 系统上必须装有 Sysprep 工具。

设备管理工作流

使用设备管理工作流，可以管理连接到虚拟机或主机数据存储的设备。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > vCenter > 虚拟机管理 > 设备管理** 访问设备管理工作流。

workflow 名称	描述
添加 CD-ROM	将虚拟 CD-ROM 添加到虚拟机。如果虚拟机没有 IDE 控制器，该 workflow 会创建一个。
添加磁盘	将虚拟磁盘添加到一台虚拟机。
更改 RAM	更改虚拟机的 RAM 容量。
将磁盘转换成精简置备	将虚拟机的厚置备磁盘转换为精简置备磁盘。
转换独立磁盘	通过从磁盘移除独立标记将所有独立的虚拟机磁盘转换为普通磁盘。
断开所有可分离设备与正在运行的虚拟机的连接	断开软盘、CD-ROM 驱动器、并行端口和串行端口与正在运行的虚拟机的连接。
挂载 CD-ROM	挂载虚拟机的 CD-ROM。如果虚拟机没有 IDE 控制器或 CD-ROM 驱动器，则该 workflow 会创建两者。
挂载软盘驱动器	从 ESXi 数据存储挂载软盘驱动器 FLP 文件。

移动和迁移 workflow

通过移动和迁移 workflow，可以迁移虚拟机。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > vCenter > 虚拟机管理 > 移动和迁移** 访问移动和迁移 workflow。

工作流名称	描述
通过 Storage vMotion 大规模迁移虚拟机	使用 Storage vMotion 迁移单个虚拟机、选择的虚拟机或所有可用的虚拟机。
通过 vMotion 大规模迁移虚拟机	使用 vMotion、Storage vMotion 或同时使用两者来迁移单个虚拟机、选择的虚拟机或所有可用的虚拟机。
通过 vMotion 迁移虚拟机	通过 vSphere API 使用 MigrateVM_Task 操作，将虚拟机从一台主机迁移到另一台主机上。
将虚拟机移至其他 vCenter Server 系统中	将列表中的虚拟机移至其他 vCenter Server 系统中。
快速迁移多个虚拟机	如果这些虚拟机的电源已打开，则先挂起虚拟机，然后将它们迁移到使用同一存储器的其他主机上。
快速迁移虚拟机	如果该虚拟机的电源已打开，则先挂起虚拟机，然后将其迁移到使用同一存储器的其他主机上。
重定位虚拟机磁盘	在虚拟机电源关闭的情况下，通过 vSphere API 使用 RelocateVM_Task 操作，将虚拟机磁盘重定位到其他主机或数据存储上。

其他工作流

使用“其他”类别中的工作流，可以启用和禁用容错 (Fault Tolerance, FT)、提取虚拟机信息，以及查找孤立的虚拟机。

在 Orchestrator 客户端的工作流视图中，可以从库 > vCenter > 虚拟机管理 > 其他访问以下工作流。

工作流名称	描述
禁用 FT	禁用指定虚拟机的 Fault Tolerance。
启用 FT	启用指定虚拟机的 Fault Tolerance。
提取虚拟机信息	返回给定虚拟机的虚拟机文件夹、主机系统、资源池、计算资源、数据存储、硬盘驱动器大小、CPU 和内存、网络和 IP 地址。可能需要 VMware Tools。
查找孤立的虚拟机	列出 Orchestrator 清单中处于孤立状态的所有虚拟机。列出 Orchestrator 清单中所有数据存储内未与清单中任一虚拟机相关联的 VMDK 和 VMTX 文件。通过电子邮件发送列表（可选）。
按名称和 BIOS UUID 获取虚拟机	按名称搜索虚拟机，然后用特定的全局唯一标识符 (UUID) 筛选结果来识别唯一的虚拟机。 注 该工作流可用于以下情况：DynamicOps 调用包含 VC:VirtualMachine 类型的输入参数的 vRealize Orchestrator 工作流以在特定 DynamicOps 和 vRealize Orchestrator 虚拟机之间建立通信。
按名称和 UUID 获取虚拟机	按名称搜索虚拟机，然后用特定的全局唯一标识符 (UUID) 筛选结果来识别唯一的虚拟机。 注 该工作流可用于以下情况：DynamicOps 调用包含 VC:VirtualMachine 类型的输入参数的 vRealize Orchestrator 工作流以在特定 DynamicOps 和 vRealize Orchestrator 虚拟机之间建立通信。
获取虚拟机 UUID	按名称搜索虚拟机，然后用特定的全局唯一标识符 (UUID) 筛选结果来识别唯一的虚拟机。 注 该工作流可用于以下情况：DynamicOps 调用包含 VC:VirtualMachine 类型的输入参数的 vRealize Orchestrator 工作流以在特定 DynamicOps 和 vRealize Orchestrator 虚拟机之间建立通信。

电源管理工作流

使用电源管理工作流，可以打开和关闭虚拟机电源、重新引导虚拟机的客户机操作系统以及挂起虚拟机等。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 虚拟机管理 > 电源管理**访问电源管理工作流。

工作流名称	描述
关闭虚拟机电源并等待	关闭虚拟机电源并等待进程完成。
重新引导客户机操作系统	重新引导虚拟机上的客户机操作系统。不重置非持久虚拟机。VMware Tools 必须处于正在运行状态。
重置虚拟机并等待	重置虚拟机并等待进程完成。
恢复虚拟机并等待	恢复挂起的虚拟机并等待进程完成。
将客户机操作系统设为待机模式	将客户机操作系统设为待机模式。VMware Tools 必须处于正在运行状态。
关闭并删除虚拟机	关闭虚拟机并将其从清单和磁盘中删除。
关闭客户机操作系统并等待	关闭客户机操作系统并等待进程完成。
启动虚拟机并等待	启动虚拟机并等待 VMware Tools 启动。
挂起虚拟机并等待	挂起虚拟机并等待进程完成。

快照工作流

通过快照工作流，可以执行与快照相关的操作。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 虚拟机管理 > 快照**访问快照工作流。

工作流名称	描述
创建快照	创建快照。
在资源池中创建所有虚拟机的快照	在资源池中创建每一虚拟机的快照。
移除所有快照	在不恢复到之前快照的情况下移除所有现有快照。
移除多余的快照	找到所含快照超出指定数目的虚拟机，并有选择地删除最旧的快照。通过电子邮件发送结果。
移除旧快照	获取所有超过给定天数的旧快照，并提示用户选择要删除的快照。
移除给定尺寸的快照	获取所有大于给定尺寸的快照并提示用户确认删除。
恢复到当前快照	恢复到当前快照。
恢复到快照并等待	恢复到特定快照。不删除该快照。

VMware Tools 工作流

使用 VMware Tools 工作流，可以在虚拟机上执行 VMware Tools 相关的任务。

在 Orchestrator 客户端的工作流视图中，可以从**库 > vCenter > 虚拟机管理 > VMware Tools** 访问 VMware Tools 工作流。

workflow名称	描述
挂载 VMware Tools 安装程序	在虚拟 CD-ROM 上挂载 VMware Tools 安装程序。
设置控制台屏幕分辨率	设置控制台窗口的分辨率。必须打开虚拟机的电源。
打开时间同步	在 VMware Tools 中打开虚拟机与 ESX Server 之间的时间同步。
卸载 VMware Tools 安装程序	卸载 VMware ToolsCD-ROM。
在无需重新启动的情况下，在 Windows 虚拟机上更新工具	在无需重新引导的情况下，在 Windows 虚拟机上更新 VMware Tools。
升级 VMware Tools	在虚拟机上升级 VMware Tools。
在下次重新引导时升级 VMware Tools	在不执行自动重新引导的情况下，在虚拟机上升级 VMware Tools。

使用 vRealize Automation 插件

4

您需要使用 vRealize Automation 插件来运行 vRealize Automation 中的 vRealize Orchestrator 工作流。

随插件提供的工作流可帮助您部署和管理 vRealize Automation 的资源。除了提供的工作流外，您还可以创建并运行自定义工作流。

本章讨论了以下主题：

- [VMware vRealize Orchestrator Plug-In for vRealize Automation 简介](#)
- [配置 vRealize Automation 插件](#)
- [使用 vRealize Automation 插件工作流](#)
- [示例 vRealize Automation 插件脚本](#)

VMware vRealize Orchestrator Plug-In for vRealize Automation 简介

VMware vRealize Orchestrator Plug-in for vRealize Automation 支持在 vRealize Orchestrator 和 vRealize Automation 之间进行交互。

您可以使用 vRealize Automation 插件为以下 vRealize Automation 功能创建并运行工作流：

- XaaS 自定义资源和蓝图管理
- 目录项和资源的管理与申请
- 授权配置
- 批准策略配置
- 工作项交互
- vSphere 和 vCloud Director 虚拟机置备和置备后操作
- 在 vRealize Automation IaaS 模型上创建、读取、更新和删除 (CRUD) 操作

包含 vRealize Automation 插件的 vRealize Orchestrator 的角色

您可以使用 Orchestrator 客户端运行并创建工作流以及访问插件 API。您可以使用 vRealize Automation 系统中的嵌入式 vRealize Orchestrator 实例，或外部 vRealize Orchestrator 服务器。

vRealize Orchestrator 为 vRealize Automation 插件提供支持。vRealize Orchestrator 是一个开发与自动化处理平台，提供可扩展的工作流库，用于管理 VMware 云堆栈和第三方技术。

vRealize Orchestrator 可通过其开放插件架构与管理解决方案进行集成。

配置 vRealize Automation 插件

您需要添加 vRealize Automation 主机和 IaaS 主机来配置插件。

配置工作流

您可以使用配置工作流类别中的工作流来管理 vRealize Automation 主机。

vRealize Automation 主机

可以从 Orchestrator 客户端的工作流视图访问以下工作流，位于插件库的配置子目录中。

工作流名称	描述
添加 vRA 主机	将 vRealize Automation 主机添加到插件清单。对于租户管理和任务，您可以使用清单视图在每个租户上运行工作流。若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。
使用组件注册表添加 vRA 主机	<p>将 vRealize Automation 主机添加到使用“单用户会话”连接的插件清单。您必须使用 vRealize Automation 系统管理员的凭据登录到 Orchestrator 客户端。</p> <p>若要在此功能与外部 vRealize Orchestrator 服务器结合使用，必须在 vRealize Automation 组件注册表中注册 Orchestrator 服务器。</p> <p>注 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>
添加 vRA 主机的 IaaS 主机	将选定 vRealize Automation 主机的 IaaS 主机添加到插件清单。
移除 vRA 主机	移除插件清单中的 vRealize Automation 主机。
更新 vRA 主机	更新插件清单中的 vRealize Automation 主机。
验证 vRA 主机	验证 vRealize Automation 主机及其连接。

注 如果 vRealize Orchestrator 服务器已在 vRealize Automation 组件注册表中注册，则会自动添加名称为“默认”的 vRealize Automation 主机。“默认”主机使用“单用户会话”连接到默认租户。默认情况下，vRealize Automation 系统中的嵌入式 Orchestrator 服务器会在 vRealize Automation 组件注册表中注册。

vRealize Automation IaaS 主机

您可以从 Orchestrator 客户端的**工作流**视图访问以下工作流，位于插件库的**基础架构管理 > 配置**子目录中。

默认情况下，vRealize Automation 系统中的嵌入式 vRealize Orchestrator 服务器会在 vRealize Automation 组件注册表中注册。

工作流名称	描述
添加 IaaS 主机	将 vRealize Automation IaaS 主机添加到插件清单。此工作流的功能与“添加 vRA 主机的 IaaS 主机”相同，但不需要 vRealize Automation 主机。
移除 IaaS 主机	移除插件清单中的 vRealize Automation IaaS 主机。
更新 IaaS 主机	更新插件清单中的 vRealize Automation IaaS 主机。
验证 IaaS 主机	验证 vRealize Automation IaaS 主机及其连接。

添加 vRealize Automation 主机

可以运行工作流以添加 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 身份验证用户名 文本框和 身份验证密码 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>注 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
SSO	选择此选项可使用 vCenter Single Sign-On。
NTLM	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**。

使用 vRealize Automation 插件 workflow

vRealize Automation 插件 workflow 库中包含多种可用于常规任务（例如与目录交互、管理基础架构以及创建租户和服务）的 workflow。

您可以使用自定义 HTTP 标头（例如 vRealize Automation 特定标头“任务”和“身份”），并将其应用到 CRUD、置备和置备后等 workflow 中。

移除操作限制

从版本 7.0 开始，某些创建、读取、更新和删除操作会受到限制。如果使用了先前版本 workflow 的操作，则这些操作将无法用于 7.0 和更高版本。您可以将 workflow 更新为支持的操作，或可以重新启动所需的操作。

若要重新启用这些操作，必须将要启用的操作从 `operations.properties` 文件移除。有关文件中的操作列表，请参见[受限制的操作](#)。

步骤

- 1 在 vRealize Orchestrator 的下拉菜单中，选择**设计**。
- 2 单击**资源**视图。
- 3 在资源层次结构中，展开**库 > VCAC > Util**。
- 4 创建备份并修改 `operations.properties` 文件。
 - a 右键单击 `operations.properties`，然后选择**保存为文件**。
 - b 将副本另存为备份。
 - c 创建新副本并删除想要重新启动的操作。
 - d 保存新文件。

- 5 替换 vRealize Orchestrator 中的现有文件。
 - a 在 vRealize Orchestrator 中，右键单击 **Util** 文件夹并单击**导入资源**。
 - b 浏览到新版本的 **operations.properties** 文件，然后单击**打开**。
 - c 单击**替换一次**以保存修改后的版本。
- 6 重新启动 vRealize Orchestrator 服务器。
- 7 选择 **operations.properties** 文件，然后单击**查看器**选项卡。
- 8 验证要启用的操作是否已不在文件中。

结果

从文件中移除的操作现在可用于较旧的工作流。

后续步骤

创建新工作流时，避免使用受限制的操作。

受限制的操作

operations.properties 文件的内容包含受限制的操作。若要重新启用该操作，必须将其从文件中移除。

以下文本为 **operations.properties** 文件的默认版本。若要重新启用某个操作，请参见[移除操作限制](#)。

```
#Blueprints
operation.create=ManagementModelEntities.svc@VirtualMachineTemplates
operation.update=ManagementModelEntities.svc@VirtualMachineTemplates
operation.delete=ManagementModelEntities.svc@VirtualMachineTemplates
#Blueprint properties
operation.create=ManagementModelEntities.svc@VirtualMachineProperties
operation.read=ManagementModelEntities.svc@VirtualMachineProperties
operation.update=ManagementModelEntities.svc@VirtualMachineProperties
operation.delete=ManagementModelEntities.svc@VirtualMachineProperties
#Global profiles
operation.create=ManagementModelEntities.svc@GlobalProfiles
operation.read=ManagementModelEntities.svc@GlobalProfiles
operation.update=ManagementModelEntities.svc@GlobalProfiles
operation.delete=ManagementModelEntities.svc@GlobalProfiles
#Global profile properties
operation.create=ManagementModelEntities.svc@GlobalProfileProperties
operation.read=ManagementModelEntities.svc@GlobalProfileProperties
operation.update=ManagementModelEntities.svc@GlobalProfileProperties
operation.delete=ManagementModelEntities.svc@GlobalProfileProperties
#PropertySetXml
operation.create=ManagementModelEntities.svc@PropertySetXml
operation.read=ManagementModelEntities.svc@PropertySetXml
operation.update=ManagementModelEntities.svc@PropertySetXml
operation.delete=ManagementModelEntities.svc@PropertySetXml
#Property definitions
operation.create=ManagementModelEntities.svc@PropertyDefinitions
operation.read=ManagementModelEntities.svc@PropertyDefinitions
```

```

operation.update=ManagementModelEntities.svc@PropertyDefinitions
operation.delete=ManagementModelEntities.svc@PropertyDefinitions
#Property attributes
operation.create=ManagementModelEntities.svc@PropertyAttributes
operation.read=ManagementModelEntities.svc@PropertyAttributes
operation.update=ManagementModelEntities.svc@PropertyAttributes
operation.delete=ManagementModelEntities.svc@PropertyAttributes
#Property Attribute Types
operation.create=ManagementModelEntities.svc@PropertyAttributeTypes
operation.read=ManagementModelEntities.svc@PropertyAttributeTypes
operation.update=ManagementModelEntities.svc@PropertyAttributeTypes
operation.delete=ManagementModelEntities.svc@PropertyAttributeTypes
#Control layouts
operation.create=ManagementModelEntities.svc@ControlLayouts
operation.read=ManagementModelEntities.svc@ControlLayouts
operation.update=ManagementModelEntities.svc@ControlLayouts
operation.delete=ManagementModelEntities.svc@ControlLayouts
#Amazon Virtual Machine Templates
operation.create=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.read=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.update=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.delete=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
#Openstack Virtual Machine Templates
operation.create=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.read=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.update=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.delete=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
#Endpoint credentials
operation.create=ManagementModelEntities.svc@ConnectionCredentials
operation.update=ManagementModelEntities.svc@ConnectionCredentials
operation.delete=ManagementModelEntities.svc@ConnectionCredentials
#Management endpoints
operation.create=ManagementModelEntities.svc@ManagementEndpoints
operation.update=ManagementModelEntities.svc@ManagementEndpoints
operation.delete=ManagementModelEntities.svc@ManagementEndpoints
#Management endpoint properties
operation.create=ManagementModelEntities.svc@ManagementEndpointProperties
operation.read=ManagementModelEntities.svc@ManagementEndpointProperties
operation.update=ManagementModelEntities.svc@ManagementEndpointProperties
operation.delete=ManagementModelEntities.svc@ManagementEndpointProperties

```

使用 vRealize Automation 插件清单

您可以使用**清单**视图在 vRealize Automation 对象上运行工作流。

要显示可供清单对象使用的工作流，请导航到**工具 > 用户首选项 > 清单**，然后选中**在清单中使用上下文菜单复选框**。启用此选项后，当您右键单击 Orchestrator 清单中的对象时，将显示可供该对象使用的所有工作流。

使用 vRealize Automation 插件管理工作流

您可以使用管理工作流来管理 vRealize Automation 服务、租户、批准策略、授权、业务组、目录项和高级服务组件。

部分工作流包含 vRealize Automation 主机的输入参数 `VCACAFE:VCACHost`。配置 vRealize Automation 主机连接的方式决定了用户运行工作流时各种角色的应用方式。

- 如果将连接配置为共享会话，则共享会话的用户帐户必须具有运行工作流所需的角色。
- 如果将连接配置为单用户会话，运行工作流的每个用户必须具有所需的角色，与在 vRealize Automation 用户界面中一样。

您可以在 vRealize Orchestrator 客户端的**工作流**视图上找到以下工作流，位于**库 > vRealize Automation > 管理**子目录中。

您可以使用**批准策略**子目录中的工作流创建和管理批准策略。

表 4-1. 批准策略

工作流	描述
激活批准策略	激活批准策略。批准策略在激活后会变为只读。
添加批准级别	向批准添加始终需要的批准级别。您必须为审批者选择特定用户和组。
复制批准策略	复制批准策略。
创建批准策略	创建不含级别或审批者的批准策略草稿。若要为自己的策略创建批准级别并指定审批者，请运行“添加批准级别”工作流。
取消激活批准策略	取消激活批准策略。您还可以删除与批准策略相关的所有现有授权。
删除批准策略	删除草稿状态的批准策略。活动的批准策略为只读。

您可以使用**业务组**子目录中的工作流创建和管理业务组以及业务组自定义属性。

表 4-2. 业务组

工作流	描述
添加自定义属性	将自定义属性添加到业务组。
创建业务组	创建业务组。
删除业务组	删除业务组。
删除自定义属性	移除业务组中的自定义属性。
更新业务组	更新业务组的详细信息，例如默认计算机前缀、活动目录容器和用户角色。
更新自定义属性	更新业务组的自定义属性。

“管理”子目录包含**业务组（已弃用）**子目录，可与 vRealize Automation 7.0 之前的版本一起使用。请使用主文件夹中同名的工作流。

您可以使用**目录项**子目录中的工作流来管理目录项。

表 4-3. 目录项

工作流	描述
激活目录项	激活目录项。您必须激活一个目录项并将其分配给服务，然后用户才能请求该服务。
向服务分配目录项。	向服务分配目录项。您必须激活一个目录项并将其分配给服务，然后用户才能请求该服务。
取消激活目录项	取消激活目录项并将其从服务目录中移除，从而使用户无法再请求。

您可以使用**复合蓝图**子目录中的工作流来管理设计画布中的复合蓝图。

表 4-4. 复合蓝图

工作流	描述
删除复合蓝图	删除设计蓝图列表中未发布的蓝图。
导入复合蓝图	从 YAML 文件导入复合蓝图。
发布复合蓝图	发布处于草稿状态的复合蓝图。
取消发布复合蓝图	取消发布已发布的复合蓝图。

内容子目录工作流已弃用。使用 Cloud Client 执行导入和导出操作。Cloud Client 的下载及相关文档可从 <https://developercenter.vmware.com/tool/cloudclient> 获取。

表 4-5. 内容

工作流	描述
导出内容（已弃用）	使用 Cloud Client 执行导入和导出操作。Cloud Client 的下载及相关文档可从 https://developercenter.vmware.com/tool/cloudclient 获取。
导入内容（已弃用）	使用 Cloud Client 执行导入和导出操作。Cloud Client 的下载及相关文档可从 https://developercenter.vmware.com/tool/cloudclient 获取。
传输内容（已弃用）	使用 Cloud Client 执行导入和导出操作。Cloud Client 的下载及相关文档可从 https://developercenter.vmware.com/tool/cloudclient 获取。
验证内容（已弃用）	使用 Cloud Client 执行导入和导出操作。Cloud Client 的下载及相关文档可从 https://developercenter.vmware.com/tool/cloudclient 获取。

您可以使用**授权**子目录中的工作流创建和管理授权。

表 4-6. 授权

工作流	描述
激活授权	激活授权。
向授权分配目录项	向授权分配一个或多个目录项。您还可以使用该工作流分配批准策略。
向授权分配即时操作	向授权分配一个或多个即时操作。即时操作不会创建请求。

表 4-6. 授权（续）

工作流	描述
向授权分配资源操作	向授权分配一个或多个资源操作。您还可以使用该工作流分配批准策略。
向授权分配服务	向授权分配一个或多个服务。您还可以使用该工作流分配批准策略。
向授权分配用户和组	向授权分配一个或多个用户或组。
创建授权（已弃用）	创建授权。对子租户使用“创建授权”。
为子租户创建授权	创建授权。
取消激活授权	取消激活授权。
取消分配给授权的用户和组	移除某个授权的用户列表中的用户和组。

您可以使用**属性**子目录中的工作流来管理属性定义和属性组。为避免与 vRealize Automation 属性发生冲突，所有自定义属性名称都应使用前缀，例如公司名称或功能名称，后面加个点 (.)。

表 4-7. 属性定义

工作流	描述
创建属性定义	创建自定义属性。
删除属性定义	删除自定义属性。

属性组是属性定义的集合。

表 4-8. 属性组

工作流	描述
将属性添加到组	将定义的自定义属性添加到组。
创建属性组	创建可在其中添加定义的自定义属性的属性组。
删除属性组	删除属性组。
移除组中的属性	移除属性组中定义的自定义属性。
更新属性组	修改属性组的名称或描述。
更新组中的属性	修改属性组中属性的名称、值和行为。

您可以使用**服务**子目录中的工作流来管理服务。

表 4-9. 服务

工作流	描述
激活服务	激活服务。
向服务分配目录项	向服务分配一个或多个目录项。
复制服务	复制服务。
创建服务	创建服务。

表 4-9. 服务（续）

工作流	描述
取消激活服务	取消激活服务。
删除服务	删除服务。

您可以使用**租户**子目录中的工作流来创建和管理租户。

身份存储工作流已弃用。替换工作流能自如应对 Directories Management API 的 vRealize Automation 的更改。

表 4-10. 租户

工作流	描述
添加管理员	向租户添加一个或多个租户管理员和基础架构管理员。
将身份存储添加到租户	向 vRealize Automation 主机的租户添加身份存储。仅当作为系统管理员配置租户时，才能运行该工作流。
将身份存储添加到租户（已弃用）	使用“将身份存储添加到租户”工作流。
将身份存储添加到 vCAC 主机	向配置为 vRealize Automation 主机的租户添加身份存储。仅当作为系统管理员为租户配置身份存储时，才能运行该工作流。
将身份存储添加到 vCAC 主机（已弃用）	使用“将身份存储添加到 vCAC 主机”。
创建租户	创建租户。您必须选择使用系统管理员凭据添加的 vRealize Automation 主机。
删除租户的身份存储	删除 vRealize Automation 主机租户的身份存储。仅当作为系统管理员配置租户时，才能运行该工作流。
删除 vCAC 主机中的身份存储	删除配置为 vRealize Automation 主机的租户的身份存储。仅当作为系统管理员为租户配置身份存储时，才能运行该工作流。
删除租户	删除租户。
移除管理员	移除租户的一个或多个租户管理员和基础架构管理员。
更新租户的身份存储	更新 vRealize Automation 主机租户的现有身份存储。仅当作为系统管理员配置租户时，才能运行该工作流。
更新租户的身份存储（已弃用）	使用“更新租户的身份存储”工作流。
更新 vCAC 主机的身份存储	更新配置为 vRealize Automation 主机的租户的身份存储。仅当作为系统管理员为租户配置身份存储时，才能运行该工作流。
更新 vCAC 主机的身份存储（已弃用）	使用“更新 vCAC 主机的身份存储”工作流。
更新租户	更新现有租户的名称、描述和联系人电子邮件地址。

您可以使用**工作流订阅**子目录中的工作流来管理事件工作流订阅。

表 4-11. 工作流订阅

工作流	描述
删除工作流订阅	删除未发布的工作流订阅。此工作流适用于系统和租户工作流订阅。
导出系统工作流订阅	导出系统工作流订阅并将其以 JSON 格式另存为 vRealize Orchestrator 资源元素。 系统工作流订阅是一个专用的工作流订阅，用于对系统事件和所有租户中的事件做出响应。
导出租户工作流订阅	导出租户工作流订阅并将其以 JSON 格式另存为资源元素。 一个专用工作流订阅用于运行租户特定的工作流。
导入系统工作流订阅	从 JSON 文件导入系统工作流订阅。系统工作流订阅会在遇到系统事件时触发，并且可以跨租户触发。
导入租户工作流订阅	从 JSON 文件导入已导出的工作流订阅。这些工作流订阅特定于租户。
发布工作流订阅	发布处于草稿或未发布状态的工作流订阅。此工作流适用于系统和租户工作流订阅。
注册系统工作流订阅	创建系统工作流订阅，其中包含超时值和优先级值。
注册租户工作流订阅	创建租户特定的工作流订阅，其中包含超时值和优先级值。
取消发布工作流订阅	取消发布已发布的工作流订阅。此工作流适用于系统和租户工作流订阅。
更新工作流订阅	更改名称、描述、vRealize Orchestrator 工作流、订阅条件、超时值、状态值和优先级值。您无法更新事件主题或阻止状态。

您可以使用 **XaaS 自定义资源** 子目录中的工作流来创建和删除 XaaS 自定义资源。

表 4-12. XaaS 自定义资源

工作流	描述
创建自定义资源	创建自定义资源。
删除自定义资源	移除自定义资源。

您可以使用 **XaaS 资源操作** 子目录中的工作流来创建和管理 XaaS 资源操作。

表 4-13. XaaS 资源操作

工作流	描述
克隆资源操作	创建现有资源操作的副本。
创建资源操作	创建资源操作。
删除资源操作	删除资源操作。
发布资源操作	发布资源操作。
取消发布资源操作	取消发布资源操作。

您可以使用 **XaaS 资源映射** 子目录中的工作流来创建和管理非 XaaS 资源的 XaaS 映射。

表 4-14. XaaS 资源映射

工作流	描述
创建资源映射	将目录资源类型映射到 vRealize Orchestrator 类型。
删除资源映射	删除资源映射。
设置目标标准	指定确定资源映射可用性的条件。

您可以使用 **XaaS 服务器配置** 子目录中的工作流来管理目标 Orchestrator 实例。

表 4-15. XaaS 服务器配置

工作流	描述
更新 Orchestrator 服务器配置	修改服务器设置，包括端口、主机、用户名和密码。
验证 Orchestrator 服务器配置	验证 vRealize Orchestrator 设置是否有效。如果配置有效，则工作流会返回 TRUE 值，反之则返回 FALSE。

您可以使用 **XaaS 服务蓝图** 子目录中的工作流来创建和管理 XaaS 蓝图。

表 4-16. XaaS 蓝图

工作流	描述
克隆服务蓝图	创建服务蓝图的副本。
创建服务蓝图	创建服务蓝图。
删除服务蓝图	删除服务蓝图。
发布服务蓝图	发布服务蓝图。
取消发布服务蓝图	取消发布服务蓝图。

使用 vRealize Automation 插件基础架构管理工作流

您可以使用基础架构管理工作流来运行基本操作。您需要使用可扩展软件包对 vRealize Automation 进行自定义，使其能够在置备过程中调用 vRealize Orchestrator 工作流，或使用自定义操作菜单进行调用。

可以从 Orchestrator 客户端的**工作流**视图中查找基础架构管理工作流，位于插件库的**基础架构管理**子目录中。

您可以使用基础架构管理工作流来置备虚拟机并运行基本的创建、读取、更新或删除操作。

表 4-17. 基础架构管理

workflow名称	描述
等待虚拟机状态更改	<p>等待一组虚拟机的状态发生更改。如果所有虚拟机都处于成功状态，则会调用触发器并且成功结束工作流。如果指定的虚拟机中有任何一台处于失败状态或不存在，则工作流失败。您必须选择以下选项输入成功和失败状态：</p> <ul style="list-style-type: none"> ■ Requested ■ AwaitingApproval ■ RegisterMachine ■ BuildingMachine ■ AddingDisks ■ MachineProvisioned ■ MachineActivated ■ InstallTools (仅 VMware) ■ On ■ Off ■ TurningOn ■ TurningOff ■ ShuttingDown ■ Suspending ■ Resetting ■ Rebooting ■ Expired ■ DeactivateMachine ■ UnprovisionMachine ■ Disposing ■ Finalized
创建 IaaS 模型实体	为指定的 vRealize Automation 模型创建并保留实体。
删除 IaaS 模型实体	删除指定的 vRealize Automation 模型实体。
调用置备后操作（已弃用）	使用“请求资源操作”工作流。
从蓝图置备虚拟机（已在 vRealize Automation 7.0 中移除）	已替换为“请求目录项”或“通过置备请求来请求目录项”。
使用自定义筛选器读取 IaaS 实体	使用自定义筛选器读取 vRealize Automation 实体列表。如果未指定筛选器，则结果会返回所有实体。
使用系统查询读取 IaaS 实体	使用 OData 系统筛选器读取 vRealize Automation 实体列表。系统筛选器适用于 OData URI 约定。
读取 IaaS 模型实体	使用模型实体 ID 读取 vRealize Automation 模型实体。
更新 IaaS 模型实体	更新 vRealize Automation 模型实体（按 ID）。

您需要使用**可扩展性**子目录中的工作流对 vRealize Automation 进行自定义，使其能够在置备过程中调用 vRealize Orchestrator 工作流，或通过自定义操作菜单进行调用。

子目录中还包含可用于管理 IaaS 凭据、端点、企业组、计算机前缀以及其他实体的的工作流。

表 4-18. 可扩展性

工作流名称	描述
安装 vCO 自定义	安装 Orchestrator 自定义，包含自定义的状态更改工作流和菜单操作工作流。
卸载 vCO 自定义	卸载 Orchestrator 自定义，包含自定义的状态更改工作流和菜单操作工作流。
更改 IaaS 虚拟机的预留	更改受管虚拟机的属性（例如预留和业务组）。
导入 IaaS 虚拟机（已弃用）	使用 Cloud Client。Cloud Client 的下载及相关文档可从 https://developercenter.vmware.com/tool/cloudclient 获取。
导入 vCenter 虚拟机（已弃用）	使用 Cloud Client。Cloud Client 的下载及相关文档可从 https://developercenter.vmware.com/tool/cloudclient 获取。
取消注册虚拟机（已在 vRealize Automation 7.0 中移除）	不会提供替换工作流。
向蓝图及其虚拟机分配菜单操作（已弃用）	在虚拟机上添加或更新菜单操作。 备用的未弃用工作流包括“向授权分配资源操作”以及“导入复合蓝图”。
向虚拟机分配菜单操作（已弃用）	更新 vRealize Automation 模型实体（按 ID）。 备用的未弃用工作流包括“向授权分配资源操作”以及“导入复合蓝图”。
向蓝图及其虚拟机分配状态更改工作流（已弃用）	已替换为 vRealize Automation 中的事件代理订阅。
自定义菜单操作（已在 vRealize Automation 7.0 中移除）	不会提供替换工作流。
从蓝图及其虚拟机中移除菜单操作（已在 vRealize Automation 7.0 中移除）	不会提供替换工作流。
从蓝图及其虚拟机中移除状态更改工作流	从蓝图及其虚拟机中移除状态更改工作流。

创建 vRealize Automation IaaS 模型实体

您可以运行工作流来创建简单或复杂 vRealize Automation IaaS 实体，例如用户的虚拟机参考。

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**。
- 4 右键单击**创建 IaaS 模型实体**工作流，然后选择**启动工作流**。
- 5 选择一个 vRealize Automation 主机。
- 6 在**模型名称**文本框中输入模型的名称。
- 7 在**实体集名称**文本框中输入实体集的名称。

您需要使用脚本或 REST API 来设置简单属性、复杂属性的链接以及 HTTP 标头属性。

- 8 单击**提交**以运行工作流。

读取 vRealize AutomationIaaS 模型实体

您可以运行工作流来读取 vRealize AutomationIaaS 模型实体。

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**。
- 4 右键单击**读取 IaaS 模型实体**，然后选择**启动工作流**。
- 5 选择一个 vRealize Automation 主机。
- 6 在**模型名称**文本框中输入模型的名称。
- 7 在**实体集名称**文本框中输入实体集的名称。
您需要使用脚本或 REST API 来设置 HTTP 标头属性。
- 8 单击**提交**以运行工作流。

使用 vRealize Automation 插件请求工作流

您可以使用请求工作流来请求目录项和资源操作，以及完成或取消工作项。

工作项需要用户的输入或操作。例如工作流交互、批准操作或响应回收请求。

您可以从 vRealize Orchestrator 客户端的**工作流**视图访问以下工作流，位于插件库的**请求**子目录中。

工作流	描述
取消工作项	取消活动的工作项。仅当作为系统管理员时，您才能使用该工作流。
完成工作项	根据所提供的用户输入完成工作项。
请求目录项	为运行此工作流的用户请求一个目录项。 如果需要个工作流来请求复合蓝图，请使用“通过置备请求来请求目录”工作流。
代表用户请求目录项	代表用户发送一个目录项请求。该工作流仅可用于您和您代表其发送请求的用户同时有权访问的目录项。
通过置备请求来请求目录	为运行此工作流的用户请求一个复合蓝图作为目录项。如果为请求提供自定义输入，则必须自定义工作流。将此工作流用于复合蓝图。
请求资源操作	请求一项资源操作，用于运行该工作流的用户所拥有的目录项。
代表用户请求资源操作	代表用户发送一个资源操作请求。该工作流仅可用于您和您代表其发送请求的用户同时有权访问的资源操作。

工作流	描述
通过请求模板请求资源操作	<p>请求一项包含复杂参数的资源操作。最佳做法是复制工作流，然后根据操作对其进行自定义。您可以使用工作流来传送复杂参数或不想在请求表单上显示的隐藏参数。此工作流的主要应用之一就是自定义 IaaS 重新配置虚拟机操作。</p> <p>要在虚拟机上创建重新配置操作，必须创建工作流的副本，然后修改该脚本。配置显示在 vRealize Orchestrator 中的参数并设置 <code>Cafe.Shim.VirtualMachine.Reconfigure.Requestor</code> 参数。此参数用于日志记录且不能为空。请参见下面的示例。</p> <pre> var requestTemplate = vCACCAFERequestsHelper.getRequestForResourceAction(operation) var jsonData = vCACCAFERequestsHelper.getResourceActionRequestData(requestTemplate); var json = JSON.parse(jsonData); //Change cpu example json.cpu = 2; //This is a property needed for the Reconfigure IaaS operation: json["Cafe.Shim.VirtualMachine.Reconfigure.Requestor"] = 1; vCACCAFERequestsHelper.setResourceActionRequestData(requestTemplate, JSON.stringify(json)); request = System.getModule("com.vmware.library.vcaccafe.request").requestResourceActionWithRequestTemplate(operation, requestTemplate); </pre>
等待目录项请求	等待目录项请求完成。
等待资源操作请求	等待资源操作请求完成。
等待工作项	等待工作项完成。

使用 vRealize Automation 插件示例工作流

您可以将示例工作流用作示例，或用作创建自定义工作流的起点。

您可以在 vRealize Orchestrator 客户端的**工作流**视图上找到以下工作流，位于插件库的**示例**子目录中。

工作流名称	描述
创建权限	提供可与授权客户端交互的示例脚本以及可在 vRealize Automation 中创建权限的权限服务。
创建租户	创建与默认租户具有相同 vRealize Automation 主机和 Active Directory 配置的租户。若要运行此工作流，请选择使用您的系统管理员凭据添加的 vRealize Automation 主机。您可以先更改 Active Directory 设置，然后再运行工作流。
列出目录项	返回选定租户的目录项列表。
打印目录项置备请求（JSON 格式）	检索目录项的默认请求表单，并将其以 JSON 格式添加到控制台日志中。您可以使用这些数据来自定义置备请求。您可以使用该信息来修改 通过置备请求来请求目录项 工作流。

访问 vRealize Automation 插件 API

Orchestrator 提供 API Explorer，您可用其搜索 vRealize Automation 插件 API 并查看相关文档，了解可用于脚本元素的 JavaScript 对象。

有关更新的 vRealize Automation API 文档，请参见 <https://www.vmware.com/support/pubs/vcac-pubs.html>。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 选择工具 > **API Explorer**。
- 3 双击左侧窗格中的 **VCAC** 和 **VCACCFE** 模块以展开 vRealize Automation 插件 API 对象的层次结构列表。

后续步骤

可以从 API 元素复制代码并将代码粘贴到脚本编写框中。有关 API 脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

有关开发最佳做法的其他信息，请参见 [vRealize Orchestrator 文档](#)。

示例 vRealize Automation 插件脚本

您可以剪切、粘贴和编辑所提供的 JavaScript 示例以开发自己的自定义脚本，用于自动处理 vRealize Automation 任务。

CRUD 基础架构管理任务示例脚本

您可以剪切、粘贴和编辑 JavaScript 示例来编写脚本，用于执行 CRUD vRealize Automation 任务。

有关 vRealize Orchestrator 中脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

示例：创建 vRealize Automation 模型实体

此示例脚本会执行以下操作：

- 1 定义模型名称和实体集名称。
- 2 定义主机前缀的属性。
- 3 保存主机前缀实体。
- 4 定义置备组的属性。
- 5 将置备组定义为链接。
- 6 将置备组实体与主机名前缀链接，保存置备组实体。

表 4-19. 输入变量

变量	类型
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'HostNamePrefixes';
var links = null;
var headers = null;
//Create properties for prefix entity
var prefixInputProperties = {
```

```

    MachinePrefix:'test-prefix',
    NextMachineNo:1,
    MachineNumberLength:3
};
//Save the prefix
var prefixEntity = vCACEntityManager
    .createModelEntity(host.id, modelName, entitySetName, prefixInputProperties, links, headers);
entitySetName = 'ProvisioningGroups';
//Create properties for the provisioning group entity
inputProperties = {
    GroupName:'TestGroupName',
    GroupDescription:'This group was generated with a vCO workflow',
    AdministratorEmail:'test@test.com',
    AdContainer:'AD',
    IsTestGroup:false,
    Flags:2,
    GroupType:1};
//Add a reference to the newly created prefix entity
links = {
    HostNamePrefix:prefixEntity
};
//Save the provisioning group
var entity = vCACEntityManager.createModelEntity(host.id, modelName, entitySetName, inputProperties,
links, headers);

```

示例：更新 vRealize Automation 模型实体

此示例脚本会执行以下操作：

- 1 获取所提供实体的主机 ID。
- 2 获取所提供实体的模型名称。
- 3 获取所提供实体的实体集名称。
- 4 获取所提供实体的实体 ID。
- 5 定义将要更新的一组属性。
- 6 启动用于更新实体的操作。

表 4-20. 输入变量

变量	类型
entity	vCAC:Entity
updatedDescription	字符串

```

var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
var entityIdString = entity.keyString;
var links = null;
var headers = null;
var updateProperties = new Properties();
updateProperties.put("UserNameDescription", updatedDescription);

```

```
//Update the user description
System.getModule("com.vmware.library.vcac")
    .updateVCACEntity(hostId, modelName, entitySetName, entityIdString, updateProperties, links,
headers);
```

示例：读取 vRealize Automation 模型实体

此示例脚本会执行以下操作：

- 1 定义模型名称和实体集名称。
- 2 定义带有属性对象的蓝图 ID。
- 3 读取实体。

表 4-21. 输入变量

变量	类型
host	vCAC:VcacHost
blueprintID	字符串

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var links = null;
var headers = null;
//Create properties for the prefix entity
var blueprintId = {
    VirtualMachineTemplateID:blueprintId,
};
//Read the blueprint
var entity = vCACEntityManager
    .readModelEntity(host.id, modelName, entitySetName, blueprintId, headers);
```

示例：删除 vRealize Automation 模型实体

此示例脚本会执行以下操作：

- 1 获取所提供实体的主机 ID。
- 2 获取所提供实体的模型名称。
- 3 获取所提供实体的实体集名称。
- 4 获取所提供实体的实体 ID。
- 5 启动用于删除实体的操作。

表 4-22. 输入变量

变量	类型
entity	vCAC:Entity

```
var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
```

```
var entityKeyString = entity.keyString;
var headers = null;
//Delete the entity
System.getModule("com.vmware.library.vcac")
    .deleteVCACEntity(hostId, modelName, entitySetName, entityKeyString, headers);
```

示例：使用自定义筛选器读取 vRealize Automation 实体

此示例脚本会执行以下操作：

- 1 定义模型名称和实体集名称。
- 2 定义用来筛选实体的属性。
- 3 读取实体列表。

表 4-23. 输入变量

变量	类型
host	vCAC:VcacHost
templateName	字符串

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var headers = null;
//Create properties for prefix entity
var properties = {
    VirtualMachineTemplateName:templateName,
};
//Read a list of entities
var entities = vCACEntityManager
    .readModelEntitiesByCustomFilter(host.id, modelName, entitySetName, properties, headers);
```

示例：使用系统查询读取 vRealize Automation 实体

此示例脚本会执行以下操作：

- 1 定义模型名称和实体集名称。
- 2 定义用来筛选实体的系统查询，并选择所有虚拟机的前十个结果（按虚拟机状态和组件标记筛选）。
- 3 读取实体列表。

表 4-24. 输入变量

变量	类型
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachines';
var filter = "VirtualMachineState eq 'Off' and IsComponent eq true";
var orderBy = 'VirtualMachineName asc';
var top = 10; {
var skip = 0; ,
```

```
var headers = null;
var select = null;
var entities = vCACEntityManager
    readModelEntitiesBySystemQuery(host.id, modelName, entitySetName, filter, orderBy, select, top,
    skip, headers);
```

查找 vRealize Automation 实体示例脚本

您可以剪切、粘贴和编辑 JavaScript 示例以编写脚本，用于通过 vCACCAFEEntitiesFinder 脚本实用程序对象来查找 vRealize Automation 实体。

有关 vRealize Orchestrator 中脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

示例：查找按名称筛选的目录资源

表 4-25. 输入变量

变量	类型
host	vCACCAFE:VcacHost

您可以使用以下任一示例：

- 此示例脚本会获取目标主机上与 *name_of_the_resource* 查询匹配（按名称和描述）的所有目录资源。

```
var items = vCACCAFEEntitiesFinder.findCatalogResources(host, "name_of_the_resource");
```

- 此示例脚本会执行以下操作：
 - 获取“用户资源”服务并调用 `get` 方法，将 `Pageable` 对象的实例作为 `vCACCAFEPageOdataRequest` 参数进行传递。
 - 创建 `vCACCAFEPageOdataRequest` 对象时使用 `OData` 查询作为单个筛选器用于筛选与 *name_of_the_resource* 字符串匹配的 `name` 属性。

```
var service = host.createCatalogClient().getCatalogConsumerResourceService();

var filter = new Array();
filter[0] = vCACCAFEFilterParam.equal("name", vCACCAFEFilterParam.string("name_of_the_resource"));
var query = vCACCAFEQuery.query().addFilter(filter);

var items = service.getResourcesList(new vCACCAFEPageOdataRequest(query));
```

示例：查找按所有者筛选的目录资源

此示例脚本会执行以下操作：

- 1 获取“用户资源”服务并调用 `get` 方法，将 `vCACCAFEPageOdataRequest` 对象的实例作为 `Pageable` 参数进行传递。
- 2 创建 `vCACCAFEPageOdataRequest` 对象时使用 `OData` 查询作为单个筛选器用于筛选与 *user@domain.com* 字符串匹配的 `owner/ref` 属性。

`owners/ref` 属性是基于内部结构和目录资源字段的组合。`VCACCAFECatalogResource` 实体具有 `owners` 属性，即 `VCACCAFECatalogPrincipal` 实体的集合。`VCACCAFECatalogPrincipal` 实体具有 `ref` 属性，即用户主体 ID 的字符串表现形式。

```
var filter = new Array();
filter[0] = vCACCAFEFilterParam.substringOf("owners/ref",
vCACCAFEFilterParam.string("user@domain.com"));
var query = vCACCAFEQuery.query().addFilter(filter);

var items = service.getResourcesList(new VCACCAFEPage0dataRequest(query));
```

示例：查找按名称和所有者筛选的目录资源

此示例脚本使用 `vCACCAFEFilterParam.and(array of conditions)` 逻辑运算符将之前两个示例的 OData 查询合并为一个条件。

```
var conditions = new Array();
conditions[0] = vCACCAFEFilterParam.equal("name",
vCACCAFEFilterParam.string("name_of_the_resource_here"));
conditions[1] = vCACCAFEFilterParam.substringOf("owners/ref",
vCACCAFEFilterParam.string("user@domain.com"));

var filter = new Array();
filter[0] = vCACCAFEFilterParam.and(conditions);
var query = vCACCAFEQuery.query().addFilter(filter);

var items = service.getResourcesList(new VCACCAFEPage0dataRequest(query));
```

您可以使用其他逻辑运算符（例如 `vCACCAFEFilterParam.group(array of parameters)`、`vCACCAFEFilterParam.not(parameter)`、`vCACCAFEFilterParam.startsWith(id, string)`、`vCACCAFEFilterParam.endsWith(id, string)`、`vCACCAFEFilterParam.greaterThan(id, number)`、`vCACCAFEFilterParam.lessThan(id, number)` 等）定义其他条件。

获取 vRealize Automation 示例脚本置备的资源

您可以剪切、粘贴和编辑 JavaScript 示例来编写脚本，用于检索 vRealize Automation 置备的资源的实际实体。

`CatalogResource` 类型表示 vRealize Automation 中已置备的资源。此类型具有 `ProviderBinding` 类型的属性，代表具有以下属性的目录资源及其提供程序之间的关系：

- `bindingId` - 表示提供程序独具的实体的标识符
- `providerRef` - 用于识别与 vRealize Automation 组件注册表中所注册的服务直接对应的目录提供程序

有关 vRealize Orchestrator 中脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

示例：获得置备为 vRealize Automation 目录资源的虚拟机

此示例将 vRealize Automation 主机及其 IaaS 主机用作输入参数，并且针对提供的资源 ID，返回相应的 IaaS 虚拟机。脚本代码仅采用 `Virtual Machine` 类型的目录资源（由 `iaas-service` 提供程序置备）。

表 4-26. 输入变量

变量	类型
vcacHost	VCACAFE:VCACHost
iaasHost	VCAC:VCACHost

```
// Id of the catalog resource (or vCACAFECatalogResource_instance.getId())
var resourceId = "c222629c-6f90-4458-8c92-8ece0ba06173";

var resource = vCACAFEEntitiesFinder.getCatalogResource(vcacHost, resourceId);

var resourceType = resource.getResourceTypeRef().getLabel();
System.log("resource type: " + resourceType);

var providerBinding = resource.getProviderBinding();

var bindingId = providerBinding.getBindingId();
System.log("provider binding id: " + bindingId);

var provider = providerBinding.getProviderRef();
System.log("provider id: " + provider.getId());
System.log("provider name: " + provider.getLabel());

if ((resourceType == "Virtual Machine") && (provider.getLabel() == "iaas-service")) {
    System.log("It is an IaaS VM!");

    // IaaS virtual machine
    var vm = Server.findForType("VCAC:VirtualMachine", bindingId);
    System.log("IaaS VM id: " + vm.virtualMachineID);
    System.log("IaaS VM name: " + vm.displayName);

    // IaaS Entity
    var entity = System.getModule("com.vmware.library.vcac").getVirtualMachineEntityFromId(iaasHost,
bindingId);
    System.log("IaaS entity id: " + entity.keyString);
}
}
```

常规任务示例脚本

您可以剪切、粘贴和编辑 JavaScript 示例，或作为示例来帮助开发常规 vRealize Automation 任务的脚本。

有关 vRealize Orchestrator 中脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

示例：创建 vRealize Automation 高级服务蓝图

此示例脚本会执行以下操作：

- 1 设置 vRealize Orchestrator 工作流用来构建服务蓝图。
- 2 基于工作流生成服务蓝图的内容。

- 3 创建服务蓝图实体。
- 4 发布服务蓝图。

表 4-27. 输入变量

变量	类型
host	vCACCAFE:VCACHost

```
//ID of the workflow used to create the service blueprint
var workflowId = "44e42047-2fa0-4e4a-ba0c-12086540b28b";

var name = "MyBlueprint"
var description = "Blueprint description";
var workflowClient = host.createAdvancedDesignerClient().getAdvancedDesignerWorkflowService();

//Generate a service blueprint based on the workflow ID
var blueprint = workflowClient.generateServiceBlueprintByWorkflowId(workflowId);
blueprint.setTenant(host.tenant);
blueprint.setName(name);
blueprint.setDescription(description);

//Create the service blueprint
var blueprintService =
host.createAdvancedDesignerClient().getAdvancedDesignerServiceBlueprintService();
var uri = blueprintService.createServiceBlueprint(host.tenant , blueprint);

//Publish the service blueprint
var createdBlueprint = blueprintService.getServiceBlueprintByUri(uri);
blueprintService.updateServiceBlueprintStatus(host.tenant, createdBlueprint.getId(),
vCACCAFEDesignerPublishStatus.PUBLISHED);
```

示例：创建 vRealize Automation 批准策略

此示例脚本会执行以下操作：

- 1 获取批准策略类型。
- 2 设置需要批准的用户和组。
- 3 设置批准级别。
- 4 定义置备前批准阶段。
- 5 定义置备后批准阶段。
- 6 定义批准策略规范，例如名称、描述和类型。
- 7 创建批准策略。
- 8 发布批准策略。发布批准策略后，策略将变为只读。

表 4-28. 输入变量

变量	类型
host	vCACCAFE:VCACHost

```
// Get the type of approval policy by ID
var typeService = host.createApprovalClient().getApprovalApprovalPolicyTypeService();
var type = typeService.getApprovalPolicyType("com.vmware.cafe.catalog.request");

// Set the user and group required to complete the approval
var user = new vCACCAFEApprovalPrincipal();
user.setValue("user@domain.com");
user.setType(vCACCAFEApprovalPrincipalType.USER);

var group = new vCACCAFEApprovalPrincipal();
group.setValue("group@domain.com");
group.setType(vCACCAFEApprovalPrincipalType.GROUP);

// Set the level of the approval
var level = new vCACCAFEApprovalLevel();
level.setName("IT Approval Level");
level.setDescription("IT Approval Level description");
level.setApprovalMode(vCACCAFEApprovalMode.ALL);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers", user);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers", group);
level.setLevelNumber(1);

// Set pre-provisioning phase type and the phase of the approval
var phase1Type = new vCACCAFEApprovalPhaseType();
phase1Type.setId("com.vmware.cafe.catalog.request.pre");
phase1Type.setName("Pre-Provisioning type");
phase1Type.setDescription("Pre-Provisioning type description");
phase1Type.setPhaseOrder(1);

var phase1 = new vCACCAFEPhase();
phase1.setName("Pre-Provisioning");
phase1.setDescription("Pre provisioning phase");
phase1.setPhasetype(phase1Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase1, "getLevels", level);

// Set post-provisioning phase type and the phase of the approval
var phase2Type = new vCACCAFEApprovalPhaseType();
phase2Type.setId("com.vmware.cafe.catalog.request.post");
phase2Type.setName("Post-Provisioning type");
phase2Type.setDescription("Post-Provisioning type description");
phase2Type.setPhaseOrder(1);

var phase2 = new vCACCAFEPhase();
phase2.setName("Post-Provisioning");
phase2.setDescription("Post provisioning phase");
phase2.setPhasetype(phase2Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase2, "getLevels", level);

// Create the approval policy specifications
```

```
var spec = new vCACCAFEApprovalPolicy();
spec.setName("New Policy");
spec.setDescription("New Policy description");
spec.setPolicyType(type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase1);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase2);

// Create the approval policy
var approvalPolicyService = host.createApprovalClient().getApprovalApprovalPolicyService();
var approvalPolicy = approvalPolicyService.createPolicy(spec);

// Publish the approval policy
approvalPolicy.setState(vCACCAFEApprovalPolicyState.PUBLISHED);
approvalPolicy = approvalPolicyService.update(approvalPolicy);
System.log("New approval policy id: " + approvalPolicy.getId());
```

使用 Configuration 插件

5

除了使用控制中心配置 Orchestrator 外，您还可以运行 Configuration 插件中的 workflow 来修改 Orchestrator 服务器配置设置。

使用 Configuration 插件，可以配置和管理 Orchestrator 服务器密钥库和受信任证书。

本章讨论了以下主题：

- 访问 Configuration 插件 workflow 库
- Configuration 插件 workflow 库

访问 Configuration 插件 workflow 库

您必须使用 Orchestrator 客户端访问 Configuration 插件 workflow 库中的元素。

步骤

- 1 从 Orchestrator 客户端的下拉菜单中，选择 **运行或设计**。
- 2 单击 **workflow** 视图。
- 3 在层次结构列表中，展开 **库 > 配置**。

后续步骤

查看 workflow 库。

Configuration 插件 workflow 库

Configuration 插件 workflow 库中包含可用于运行与 vRealize Orchestrator 配置相关的自动处理操作的 workflow。

SSL Trust Manager 工作流

SSL Trust Manager 类别包含可用于删除和导入 SSL 证书的工作流。

在 Orchestrator 客户端的 **workflow** 视图中，可以从 **库 > 配置 > SSL Trust Manager 工作流** 访问以下工作流。

工作流名称	描述
删除受信任证书	删除服务器信任存储中的 SSL 证书。
从 URL 导入证书	将 SSL 证书从 URL 导入到服务器信任存储中。
使用已经过身份验证的代理服务器从 URL 导入证书	从可通过已经过身份验证的代理服务器访问的 URL 导入 SSL 证书。
使用代理服务器从 URL 导入证书	从可通过代理服务器访问的 URL 导入 SSL 证书。
从 URL 导入证书及证书别名	将 SSL 证书从 URL 导入到服务器信任存储中。
从文件导入受信任证书	将 SSL 证书从文件导入到服务器信任存储中。

密钥库工作流

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > 配置 > 密钥库**访问密钥库配置工作流。

工作流名称	描述
添加证书	将证书添加到密钥库。
添加密钥	添加密钥。
创建密钥库	创建新密钥库。
删除密钥库	删除密钥库。
删除证书	删除密钥库中的证书。
删除条目	删除条目。
删除密钥	删除密钥。

使用 Library 插件

6

您可以将 Library 插件工作流作为模板，用于客户端流程的自定义和自动处理，并进行 Orchestrator 疑难解答。

本章讨论了以下主题：

- [Library 插件工作流](#)

Library 插件工作流

Library 插件提供锁定、Orchestrator 和故障排除等工作流类别的工作流。

锁定工作流

在 Orchestrator 客户端的工作流视图中，可以从库 > 锁定访问以下工作流。

工作流名称	描述
显示所有锁	显示所有锁。
锁定测试	用于创建锁的测试工作流。
锁定测试 (x5)	用于创建五个锁的测试工作流。
释放所有锁	释放所有锁。

Orchestrator 任务工作流

在 Orchestrator 客户端的工作流视图中，可以从库 > Orchestrator > 任务访问以下工作流。

工作流名称	描述
创建重复任务	创建重复任务并返回最新创建的任务。
创建任务	调度工作流作为任务在以后的时间和日期运行。

Orchestrator 工作流

在 Orchestrator 客户端的工作流视图中，可以从库 > Orchestrator > 工作流访问以下工作流。

工作流名称	描述
刷新处于等待状态中的失效工作流运行	处理等待用于指定远程服务器的所有工作流运行，并根据远程工作流运行更新工作流状态。如果工作流运行之间存在数据丢失（例如 Orchestrator 服务器之间的连接丢失），则可以使用该工作流。
串行启动工作流	以串行形式（即依次运行实例）多次运行某个工作流。您需要以数组形式提供工作流参数。您还需要为每个启动的工作流实例提供一个属性列表（每个工作流输入对应一个属性）。数组中的属性数量定义了工作流运行数量。
并行启动工作流	采用不同参数多次运行某个工作流。您需要以数组形式提供工作流参数。您还需要为每个启动的工作流实例提供一个属性列表（每个工作流输入对应一个属性）。数组中的属性数量定义了工作流运行数量。

标记工作流

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > 标记**访问以下工作流。

工作流名称	描述
按标记查找对象	按对象所分配到的标记查找对象。您提供标记的名称和值后，工作流会返回应用了这些标记的对象的列表。
列出工作流标记	列出分配给工作流（指定为输入参数）的标记。
标记示例	演示工作流标记。
标记工作流	向工作流分配标记。您必须指定要标记的工作流以及标记的名称和值。
取消标记工作流	移除工作流的标记。您必须指定要取消标记的工作流以及要从指定工作流移除的标记。

使用 SQL 插件

7

您可以使用 SQL 插件提供的 API 来实现与 SQL 数据库以及其他表格数据源（例如电子表格或平面文件）的连接。

SQL 插件 API 基于 JDBC，并且提供可用于访问基于 SQL 的数据库的调用级 API。SQL 插件还提供可用于展示如何在工作流中使用 API 的示例工作流。

本章讨论了以下主题：

- [配置 SQL 插件](#)
- [运行 SQL 示例工作流](#)
- [使用 SQL 插件标准工作流](#)

配置 SQL 插件

您可以使用 SQL 插件中包含的工作流并通过 Orchestrator 客户端运行这些工作流，用于配置 SQL 插件并添加、更新或删除数据库。

SQL 插件配置工作流

SQL 插件的“配置”工作流类别包含可用于管理数据库和数据库表格的工作流。

在 Orchestrator 客户端的工作流视图中，可以从**库 > SQL > 配置**访问以下工作流。

工作流名称	描述
添加数据库	将数据库对象添加到数据库插件清单。
将表格添加到数据库	将数据库表格添加到数据库插件清单中的数据库。
移除数据库	移除数据库插件清单中的数据库对象。
移除数据库中的表格	从数据库插件清单的数据库中移除数据库表格。
更新数据库	更新数据库插件清单中数据库对象的配置。
验证数据库	验证数据库插件清单中的数据库。

添加数据库

您可以运行工作流将数据库添加到 Orchestrator 服务器并配置主机连接参数。

在添加需要安全连接的数据库时，必须导入数据库 SSL 证书。您可以在控制中心的**受信任证书**选项卡中导入 SSL 证书。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > SQL > 配置**并导航到**添加数据库**工作流。
- 4 右键单击**添加数据库**工作流，然后选择**启动工作流**。
- 5 在**名称**文本框中，输入数据库的名称。
- 6 选择数据库类型。
- 7 在**连接 URL**文本框中，输入数据库的地址。

数据库类型	语法
Oracle	<code>jdbc:oracle:thin:@database_url:port_number:SID</code>
Microsoft SQL (使用 SQL 身份验证)	<code>jdbc:jtds:sqlserver://database_url:port_number/database_name</code>
Microsoft SQL (使用 Windows 帐户身份验证)	<code>jdbc:jtds:sqlserver://database_url:port_number/database_name;useNTLMv2=true;domain=domain_name</code>
PostgreSQL	<code>jdbc:postgresql://database_url:port_number/database_name</code>
MySQL	<code>jdbc:mysql://database_url:port_number/database_name</code>

- 8 选择插件连接到数据库时所用的会话模式。

选项	描述
共享会话	该插件使用共享凭据连接到数据库。您必须提供共享会话的数据库凭据。
单用户会话	Orchestrator 客户端会检索已登录用户的凭据。 注 若要使用单用户会话模式，必须仅使用用户名进行身份验证。您不得使用 <code>domain\user</code> 或 <code>user@domain</code> 进行身份验证。

- 9 单击**提交**以运行工作流。

结果

工作流成功运行后，属于该工作流的数据库和所有表格会显示在**清单**视图中。

将表格添加到数据库

您可以运行工作流将表格添加到数据库插件清单中的数据库。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与数据库建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SQL > 配置**并导航到**将表格添加到数据库**工作流。
- 3 右键单击**将表格添加到数据库**工作流，然后选择**启动工作流**。
- 4 选择要向其添加表格的数据库。
- 5 选择要添加的表格。
- 6 单击**提交**以运行工作流。

结果

工作流运行成功后，添加的数据库表格会显示在 Orchestrator 客户端的**清单**视图中。

更新数据库

您可以运行工作流来更新插件清单中的数据库的配置。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > SQL > 配置**并导航到**更新数据库**工作流。
- 4 右键单击**更新数据库**工作流，然后选择**启动工作流**。
- 5 选择要更新的数据库。
- 6 在**名称**文本框中，输入数据库的新名称。
数据库会以您指定的名称显示在**清单**视图中。
- 7 选择数据库类型。
- 8 在**连接 URL**文本框中，输入数据库的新地址。

9 选择插件连接到数据库时所用的会话模式。

选项	描述
共享会话	该插件使用共享凭据连接到数据库。您必须提供共享会话的数据库凭据。
单用户会话	Orchestrator 客户端会检索已登录用户的凭据。 注 若要使用单用户会话模式，必须仅使用用户名进行身份验证。您不得使用 <code>domain\user</code> 或 <code>user@domain</code> 进行身份验证。

10 单击提交以运行工作流。

运行 SQL 示例工作流

您可以运行 SQL 插件工作流来执行 JDBC 操作，例如生成 JDBC URL、测试 JDBC 连接以及管理 JDBC 表格中的行。您还可以运行 SQL 插件工作流来管理数据库和数据库表格，并运行 SQL 操作。

生成 JDBC URL

您可以从 Orchestrator 客户端中运行相应的工作流来生成 JDBC 连接 URL。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > JDBC**并导航到“JDBC URL 生成器”工作流。
- 3 右键单击“JDBC URL 生成器”工作流并选择**启动工作流**。
- 4 选择要对其生成 URL 的数据库的类型。

注 如果使用 Microsoft 数据库，可能需要单击**下一步**并提供数据库实例名称和数据库用户域名。

- 5 提供所需信息以生成数据库 URL。
 - a 输入数据库服务器名称或 IP 地址。
 - b 输入数据库名称。
 - c （可选）输入数据库端口号。
如果未指定端口号，工作流会使用默认端口号。
 - d 键入用户名以访问数据库。
 - e 键入密码以访问数据库。
- 6 单击**提交**以运行工作流。

测试 JDBC 连接

您可以从 Orchestrator 客户端中运行相应的工作流来测试与数据库的连接。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“JDBC 连接”示例工作流。
- 3 右键单击“JDBC 连接”示例工作流，然后选择**启动工作流**。
- 4 提供所需信息以测试数据库连接。
 - a 键入用户名以访问数据库。
 - b 输入要测试的 URL。
 - c 键入密码以访问数据库。
- 5 单击**提交**以运行工作流。

使用 JDBC 创建表格

您可以从 Orchestrator 客户端中运行相应的工作流来创建数据库。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“JDBC 创建表格”示例工作流。
- 3 右键单击“JDBC 创建表格”示例工作流，然后选择**启动工作流**。
- 4 提供所需信息并单击**下一步**。
 - a 键入密码以访问数据库。
 - b 键入数据库连接 URL。
 - c 键入用户名以访问数据库。
- 5 输入 SQL 创建语句。

示例语法为：

```
CREATE TABLE "table_name"  
("column1" "data_type_for_column1",  
"column2" "data_type_for_column2")
```

- 单击**提交**以运行工作流。

在 JDBC 表格中插入行

您可以从 Orchestrator 客户端中运行相应的工作流来测试在 JDBC 表格中插入行。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 单击 Orchestrator 客户端中的**工作流**视图。
- 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“JDBC 插入表格”示例工作流。
- 右键单击“JDBC 插入表格”示例工作流，然后选择**启动工作流**。
- 提供所需信息并单击**下一步**。
 - 键入数据库连接 URL。
 - 键入用户名以访问数据库。
 - 键入密码以访问数据库。
- 输入 SQL 插入语句，然后单击**下一步**。

示例语法为：

```
INSERT INTO "table_name" ("column1", "column2")  
VALUES ("value1", "value2")
```

- 输入要插入到行的值。
- 单击**提交**以运行工作流。

选择 JDBC 表格中的行

您可以从 Orchestrator 客户端中运行相应的工作流来选择 JDBC 表格中的行。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 单击 Orchestrator 客户端中的**工作流**视图。
- 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“JDBC 从表格选择”示例工作流。
- 右键单击“JDBC 从表格选择”示例工作流，然后选择**启动工作流**。

- 4 提供所需信息并单击**下一步**。
 - a 键入数据库连接 URL。
 - b 键入用户名以访问数据库。
 - c 键入密码以访问数据库。

- 5 输入 SQL 选择语句。

示例语法为：

```
SELECT * FROM "table_name"
```

- 6 单击**提交**以运行工作流。

删除 JDBC 表格中的条目

您可以从 Orchestrator 客户端中运行相应的工作流来测试删除 JDBC 表格中的条目。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“JDBC 删除表格条目”示例工作流。
- 3 右键单击“JDBC 删除表格条目”示例工作流，然后选择**启动工作流**。
- 4 提供所需信息并单击**下一步**。
 - a 输入要删除的用户条目的名字。
 - b 键入用户名以访问数据库。
 - c 输入 JDBC 连接 URL。
 - d 输入要删除的用户条目的姓氏。
 - e 键入密码以访问数据库。

- 5 输入 SQL 删除语句。

示例语法为：

```
DELETE FROM "table_name" where ("column1" = ?, "column2" = ?)
```

- 6 单击**提交**以运行工作流。

删除 JDBC 表格中的所有条目

您可以从 Orchestrator 客户端中运行相应的工作流来删除 JDBC 表格中的所有条目。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“JDBC 删除表格所有条目”示例工作流。
- 3 右键单击“JDBC 删除表格所有条目”示例工作流，然后选择**启动工作流**。
- 4 提供所需信息并单击**下一步**。
 - a 键入数据库连接 URL。
 - b 键入用户名以访问数据库。
 - c 键入密码以访问数据库。

- 5 输入 SQL 删除语句。

示例语法为：

```
DELETE FROM "table_name"
```

- 6 单击**提交**以运行工作流。

丢弃 JDBC 表格

您可以从 Orchestrator 客户端中运行相应的工作流来测试丢弃 JDBC 表格。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“JDBC 丢弃表格”示例工作流。
- 3 右键单击“JDBC 丢弃表格”示例工作流，然后选择**启动工作流**。
- 4 提供所需信息并单击**下一步**。
 - a 键入密码以访问数据库。
 - b 键入数据库连接 URL。
 - c 键入用户名以访问数据库。

5 输入 SQL 丢弃语句。

示例语法为：

```
DROP TABLE "table_name"
```

6 单击**提交**以运行工作流。

运行完整的 JDBC 循环

您可以从 Orchestrator 客户端中运行相应的工作流来测试所有的 JDBC 示例工作流（只需一个完整循环）。

前提条件

验证您用来登录的用户帐户是否拥有运行 JDBC 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > JDBC > JDBC 示例**并导航到“完整 JDBC 循环”示例工作流。
- 3 右键单击“完整 JDBC 循环”示例工作流，然后选择**启动工作流**。
- 4 提供所需信息并单击**下一步**。
 - a 键入数据库连接 URL。
 - b 键入用户名以访问数据库。
 - c 键入密码以访问数据库。
- 5 输入要用作数据库的条目的值。
- 6 单击**提交**以运行工作流。

使用 SQL 插件标准工作流

您可以使用 SQL 工作流来运行 SQL 操作。

SQL 插件工作流库

您可以运行 SQL 插件工作流来管理数据库和数据库表格，并运行 SQL 操作。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > SQL > 配置**访问数据库配置工作流。

工作流名称	描述
添加数据库	将数据库对象添加到插件清单。
将表格添加到数据库	将数据库表格添加到插件清单中的数据库。
移除数据库	移除插件清单中的数据库对象。
移除数据库中的表格	从插件清单的数据库中移除数据库表格。

工作流名称	描述
更新数据库	更新插件清单中数据库对象的配置。
验证数据库	验证插件清单中的数据库。

在 Orchestrator 客户端的工作流视图中，可以从**库 > SQL** 访问 SQL 操作工作流。

工作流名称	描述
在数据库上执行自定义查询	在指定数据库上运行自定义查询并返回受影响的行数。您可以运行工作流来更新、删除、插入和编写查询。
为表格生成 CRUD 工作流	为指定表格生成“创建”、“读取”、“更新”和“删除”工作流。
读取数据库中的自定义查询	在指定数据库上运行自定义查询并以属性数组形式返回结果。您可以运行工作流来选择和读取查询。

为表格生成 CRUD 工作流

您可以运行工作流为特定表格生成“创建”、“读取”、“更新”和“删除”工作流。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与数据库建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SQL** 并导航到**为表格生成 CRUD 工作流**工作流。
- 3 右键单击**为表格生成 CRUD 工作流**工作流，然后选择**启动工作流**。
- 4 选择要为其生成工作流的表格。
- 5 选择要在其中生成工作流的工作流文件夹。
- 6 选择是否要覆盖任何现有工作流。

选项	描述
是	生成的工作流会覆盖同名的现有工作流。
否	如果文件夹中存在同名的工作流，则不会生成新工作流。

- 7 （可选）选择不应填充内容的列。
您无法使用生成的 CRUD 工作流编辑选定的列。
- 8 单击**提交**以运行工作流。

结果

工作流成功运行后，CRUD 工作流会显示在选定工作流文件夹中。

后续步骤

您可以在选定的数据库表格上运行生成的工作流。

使用 SSH 插件

8

您可以使用 SSH 插件工作流在支持 SSH 的远程主机上运行 SSH 命令，并通过安全连接在 Orchestrator 服务器和远程主机之间传输文件。

本章讨论了以下主题：

- [配置 SSH 插件](#)
- [运行 SSH 插件示例工作流](#)

配置 SSH 插件

您可以设置 SSH 插件以确保使用加密连接。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > SSH** 并导航到“添加 SSH 主机”工作流。
- 4 右键单击“配置邮件”工作流，然后选择**启动工作流**。
- 5 在**主机名称**对话框中，输入要通过 Orchestrator 使用 SSH 访问的主机的名称。
- 6 输入目标端口。默认 SSH 端口为 22。
主机即添加到 SSH 连接列表中。
- 7 （可选）配置服务器上的入口路径。
 - a 单击**新建根文件夹**。
 - b 输入新路径，然后单击**插入值**。
- 8 为有权运行 SSH 命令的用户输入用户名。

9 选择身份验证类型。

选项	操作
是	输入密码以使用密码身份验证。
否	输入专用密钥的路径以及专用密钥密码短语以使用密钥身份验证。

10 单击提交以运行工作流。

结果

SSH 主机即会显示在 Orchestrator 客户端的**清单**视图中。

配置工作流

SSH 插件的“配置”类别中包含的工作流可用来管理 Orchestrator 和 SSH 主机之间的连接。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > SSH > 配置**访问以下工作流。

工作流名称	描述
将根文件夹添加到 SSH 主机	将根文件夹添加到 SSH 主机的现有连接。
添加 SSH 主机	将 SSH 主机的新连接添加到现有配置。
将根文件夹从 SSH 主机中移除	将根文件夹从 SSH 主机的现有连接中移除。
移除 SSH 主机	将 SSH 主机的现有连接从现有配置中移除。
更新 SSH 主机	更新 SSH 主机的现有连接。

运行 SSH 插件示例工作流

您可以从 Orchestrator 客户端中运行 SSH 插件示例工作流来测试 Orchestrator 服务器和 SSH 主机之间的连接。

■ 生成密钥对

您可以从 Orchestrator 客户端中运行相应的工作流来生成密钥对。您可以使用密钥对连接到 SSH 主机而无需密码。

■ 更改密钥对密码短语

您可以从 Orchestrator 客户端中运行相应的工作流来更改最近生成的密钥对密码短语。

■ 在 SSH 主机上注册 Orchestrator 公共密钥

您可以使用公共密钥来代替密码。若要在 SSH 主机上注册 Orchestrator 公共密钥，您可以从 Orchestrator 客户端中运行相应的工作流。

■ 运行 SSH 命令

您可以从 Orchestrator 客户端中运行相应的工作流以在远程 SSH 服务器运行 SSH 命令。

■ 从 SSH 主机复制文件

您可以在 Orchestrator 客户端上运行工作流将文件从 SSH 主机复制到 Orchestrator 服务器。

■ 将文件复制到 SSH 主机

您可以运行 Orchestrator 客户端上的工作流将文件从 Orchestrator 服务器复制到 SSH 主机。

生成密钥对

您可以从 Orchestrator 客户端中运行相应的工作流来生成密钥对。您可以使用密钥对连接到 SSH 主机而无需密码。

密钥对由一个公共密钥和一个专用密钥组成。Orchestrator 可以使用专用密钥连接到 SSH 主机上的公共密钥。您可以使用密码短语来增强安全性。

小心 所有拥有相应权限的 Orchestrator 用户都可以读取、使用和覆盖专用密钥。

前提条件

验证您用来登录的用户帐户是否拥有运行 SSH 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SSH** 并导航到“生成密钥对”工作流。
- 3 右键单击“生成密钥对”工作流，然后选择**启动工作流**。
- 4 提供所需信息。
 - a 选择密钥类型。
 - b 选择密钥大小。
 - c （可选）输入密码短语。

注 您可以稍后更改密码短语。

- d （可选）输入备注。
- 5 单击**提交**以运行工作流。

如果已存在密钥对，则新密钥对会将其覆盖。

更改密钥对密码短语

您可以从 Orchestrator 客户端中运行相应的工作流来更改最近生成的密钥对密码短语。

前提条件

验证您用来登录的用户帐户是否拥有运行 SSH 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，照片**库 > SSH** 并导航到“更改密钥对密码短语”工作流。
- 3 右键单击“更改密钥对密码短语”工作流，然后选择**启动工作流**。

- 4 重置密钥对密码短语。
 - a 输入当前密码短语。
 - b 输入新密码短语。
- 5 单击**提交**以运行工作流。

在 SSH 主机上注册 Orchestrator 公共密钥

您可以使用公共密钥来代替密码。若要在 SSH 主机上注册 Orchestrator 公共密钥，您可以从 Orchestrator 客户端中运行相应的工作流。

前提条件

验证您用来登录的用户帐户是否拥有运行 SSH 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SSH** 并导航到“在主机上注册 vCO 公共密钥”工作流。
- 3 右键单击“在主机上注册 vCO 公共密钥”工作流，然后选择**启动工作流**。
- 4 提供 SSH 主机的名称，以及用于登录该主机的用户名和密码。

注 您必须提供已在 SSH 主机上注册的凭据。

- 5 单击**提交**以运行工作流。

结果

在作为注册用户连接到 SSH 主机时，您可以使用公共密钥代替密码来进行身份验证。

运行 SSH 命令

您可以从 Orchestrator 客户端中运行相应的工作流以在远程 SSH 服务器运行 SSH 命令。

前提条件

验证您用来登录的用户帐户是否拥有运行 SSH 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SSH** 并导航到“运行 SSH 命令”工作流。
- 3 右键单击“运行 SSH 命令”工作流，然后选择**启动工作流**。
- 4 输入 SSH 主机名称或 IP 地址，然后单击**下一步**。
- 5 输入要运行的 SSH 命令，然后单击**下一步**。

注 默认 SSH 命令为 **uptime**。它会显示服务器已激活的时长以及该时段内的用户负载。

- 6 选择**是**以使用密码进行身份验证，然后单击**下一步**。

注 默认选项是使用密钥文件身份验证。

- 7 输入用户名，然后单击**下一步**。
- 8 如果身份验证方法需要密码，则请输入密码。否则，输入专用密钥的路径，然后输入该专用密钥的密码短语。
- 9 单击**提交**以运行工作流。

从 SSH 主机复制文件

您可以在 Orchestrator 客户端上运行工作流将文件从 SSH 主机复制到 Orchestrator 服务器。

SSH 插件使用 Java JCraft 库，后者实现了 SFTP。SCP get 命令工作流使用 SFTP 传输文件。

前提条件

验证您用来登录的用户帐户是否拥有运行 SSH 工作流的必要权限。

注 Orchestrator 必须拥有显式写入权限才能在文件夹中进行写入操作。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SSH** 并导航到“SCP get 命令”工作流。
- 3 右键单击 SCP get 命令，然后选择**启动工作流**。
- 4 提供所需信息并单击**下一步**。
 - a 输入 SSH 主机名或 IP 地址。
 - b 输入 SSH 身份验证信息。
- 5 输入文件信息。
 - a 输入指向 Orchestrator 服务器上要向其中复制文件的目录的路径。
 - b 输入指向要从远程 SSH 主机获取的文件的路径。
- 6 单击**提交**以运行工作流。

将文件复制到 SSH 主机

您可以运行 Orchestrator 客户端上的工作流将文件从 Orchestrator 服务器复制到 SSH 主机。

SSH 插件使用 Java JCraft 库，后者实现了 SFTP。SCP put 命令工作流使用 SFTP 传输文件。

前提条件

验证您用来登录的用户帐户是否拥有运行 SSH 工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SSH** 并导航到 “SCP put 命令” 工作流。
- 3 右键单击 SCP put 命令，然后选择**启动工作流**。
- 4 提供所需信息并单击**下一步**。
 - a 输入 SSH 主机名或 IP 地址。
 - b 输入 SSH 身份验证信息。
- 5 输入文件信息。
 - a 输入要从本地 Orchestrator 服务器复制到远程 SSH 主机的文件的路径。
 - b 输入远程 SSH 主机上要向其中复制文件的目录的路径。
- 6 单击**提交**以运行工作流。

使用 XML 插件

9

您可以使用 XML 插件来运行用于创建和修改 XML 文档的工作流。

XML 插件将文档对象模型 (DOM) XML 解析程序的实现添加到 Orchestrator JavaScript API。XML 插件还提供了部分示例工作流，用于展示如何通过工作流创建和修改 XML 文档。

或者，您也可以在 Orchestrator JavaScript API 中使用 ECMAScript for XML (E4X) 以在 JavaScript 中直接处理 XML 文档。有关 E4X 脚本示例，请参见《使用 VMware vRealize Orchestrator 进行开发》。

有关 E4X 的信息，请访问维护 ECMA-357 标准的组织的网站。

本章讨论了以下主题：

- [运行 XML 插件示例工作流](#)

运行 XML 插件示例工作流

您可以从 Orchestrator 客户端中运行 XML 插件示例工作流来创建和修改用于测试的 XML 文档。

由于工作流可以创建、读取或修改文件，您必须拥有足够的权限才能访问工作目录。

Orchestrator 对服务器系统 root 目录中名为 **orchestrator** 的文件夹具有读取、写入和执行权限。尽管工作流在此文件夹中具有读取、写入和执行权限，但您必须在服务器系统中创建该文件夹。如果使用 Orchestrator Appliance，则该文件夹将命名为 **vco**，且位于 **/var/run/vco**。

您可以通过工作流和 JavaScript 更改服务器文件系统访问权限的设置，从而允许访问其他文件夹。请参见《安装和配置 VMware vRealize Orchestrator》中的“设置工作流和操作对服务器文件系统的访问权限”。

- [创建简单 XML 文档](#)

您可以从 Orchestrator 客户端中运行相应的工作流来创建用于测试的简单 XML 文档。

- [查找 XML 文档中的元素](#)

您可以从 Orchestrator 客户端中运行相应的工作流来查找由“创建简单 XML 文档”工作流创建的 XML 中的元素。

■ 修改 XML 文档

您可以从 Orchestrator 客户端中运行相应的工作流来修改由“创建简单 XML 文档”工作流创建的 XML。

■ 从 XML 创建示例通讯簿

您可以从 Orchestrator 客户端中运行相应的工作流来创建用于测试的通讯簿。

创建简单 XML 文档

您可以从 Orchestrator 客户端中运行相应的工作流来创建用于测试的简单 XML 文档。

前提条件

- 验证您用来登录的用户帐户是否拥有运行 XML 工作流的必要权限。
- 验证您已在 Orchestrator 服务器系统根目录中创建 `c:/orchestrator` 文件夹，或已设置另一文件夹的访问权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，打开**库 > XML > 示例 XML (简单)**并导航到“创建简单 XML 文档”工作流。
- 3 右键单击“创建简单 XML 文档”工作流，然后选择**启动工作流**。
- 4 输入指向要创建的 XML 文档的文件路径。
例如：`c:/orchestrator/filename.xml`。
- 5 单击**提交**以运行工作流。

结果

该工作流会创建一个包含用户列表的 XML 文档。每个条目的属性为 `user ID` 和 `name`。

查找 XML 文档中的元素

您可以从 Orchestrator 客户端中运行相应的工作流来查找由“创建简单 XML 文档”工作流创建的 XML 中的元素。

前提条件

- 验证您用来登录的用户帐户是否拥有运行 XML 工作流的必要权限。
- 验证您已在 Orchestrator 服务器系统根目录中创建 `c:/orchestrator` 文件夹，或已设置另一文件夹的访问权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，打开**库 > XML > 示例 XML (简单)**并导航到“在文档中查找元素”工作流。

- 3 右键单击“在文档中查找元素”工作流，然后选择**启动工作流**。
- 4 输入 XML 文档的文件路径。

例如：c:/orchestrator/*filename.xml*。

- 5 单击**提交**以运行工作流。

该工作流会搜索元素并在系统日志中显示结果。

后续步骤

若要查看结果，请在 Orchestrator 客户端中选择已完成的工作流运行，然后在**架构**选项卡上单击**日志**。

修改 XML 文档

您可以从 Orchestrator 客户端中运行相应的工作流来修改由“创建简单 XML 文档”工作流创建的 XML。

前提条件

- 验证您用来登录的用户帐户是否拥有运行 XML 工作流的必要权限。
- 验证您已在 Orchestrator 服务器系统根目录中创建 c:/orchestrator 文件夹，或已设置另一文件夹的访问权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，打开**库 > XML > 示例 XML (简单)**并导航到“修改 XML 文档”工作流。
- 3 右键单击“修改 XML 文档”工作流，然后选择**启动工作流**。
- 4 提供输入和输出文件路径。
 - a 输入指向要修改的 XML 文档的文件路径。
例如：c:/orchestrator/*filename.xml*。
 - b 输入指向已修改的 XML 文档的文件路径。
例如：c:/orchestrator/*filename.xml*。

注 如果在两个字段中输入同一文件路径，则工作流会用修改后的文件覆盖原始文件。如果输入一个不存在的文件的输出路径，则工作流会创建修改后的文件。

- 5 单击**提交**以运行工作流。

结果

该工作流会搜索元素并修改元素所在的条目。

从 XML 创建示例通讯簿

您可以从 Orchestrator 客户端中运行相应的工作流来创建用于测试的通讯簿。

前提条件

- 验证您用来登录的用户帐户是否拥有运行 XML 工作流的必要权限。
- 验证您已在 Orchestrator 服务器系统根目录中创建 `c:/orchestrator` 文件夹，或已设置另一文件夹的访问权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，打开**库 > XML > 示例 XML (通讯簿)**并导航到“完整通讯簿测试”工作流。
- 3 右键单击“完整通讯簿测试”工作流，然后选择**启动工作流**。
- 4 输入通讯簿文件夹的路径。

例如： `c:/orchestrator/foldername`。

如果该文件夹不存在，则工作流会自动创建文件夹。

- 5 单击**提交**以运行工作流。

结果

该工作流会创建 DTD、XML 和 CSS 文件、附加样式表并将文件存储在指定的文件夹中。

使用 Mail 插件

10

您可以使用 Mail 插件（使用简单邮件传输协议，即 SMTP）通过工作流发送电子邮件消息。例如：您可以创建工作流用于在某个工作流需要用户交互或者完成运行后向给定地址发送一封电子邮件。

本章讨论了以下主题：

- 定义默认 SMTP 连接
- 使用 Mail 插件示例工作流

定义默认 SMTP 连接

邮件插件随 Orchestrator 服务器一起安装，用于发送和接收电子邮件通知。您可以设置进行 SMTP 服务器身份验证时所用的默认电子邮件帐户，来发送和接收电子邮件通知。

注 在 Orchestrator 中配置邮件时避免使用负载均衡器，否则，可能会收到 SMTP_HOST_UNREACHABLE 错误。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的工作流视图。
- 3 在工作流层次结构列表中，展开库 > 邮件并导航到“配置邮件”工作流。
- 4 右键单击“配置邮件”工作流，然后选择启动工作流。
- 5 输入所需信息。

文本框	说明
SMTP 主机	输入 SMTP 服务器的 IP 地址或域名。
SMTP 端口	输入与 SMTP 配置匹配的端口号。 默认 SMTP 端口为 25。
用户名	输入有效的电子邮件地址。 这是 Orchestrator 用于发送电子邮件的电子邮件帐户。

文本框	说明
密码	输入与用户名关联的密码。
发件人姓名和地址	输入发件人信息，该信息会显示在 Orchestrator 发送的所有电子邮件中。

6 单击 **提交** 以运行工作流。

使用 Mail 插件示例工作流

您可以从自定义工作流中调用 Mail 插件的示例工作流，以在自定义工作流中实现电子邮件收发功能。您可以通过运行示例工作流测试 Orchestrator 与 SMTP 服务器之间的交互。

■ 访问 Mail 插件示例工作流

您可以通过 Orchestrator 客户端访问 Mail 插件示例工作流。

■ Mail 插件示例工作流

您可以集成示例 Mail 插件工作流来增强自定义工作流。

访问 Mail 插件示例工作流

您可以通过 Orchestrator 客户端访问 Mail 插件示例工作流。

前提条件

验证您用来登录的用户帐户是否拥有运行邮件工作流的必要权限。

步骤

- 1 单击 Orchestrator 客户端中的 **工作流** 视图。
- 2 在层次结构列表中，展开 **库 > Mail**。

后续步骤

查看并运行示例工作流。

Mail 插件示例工作流

您可以集成示例 Mail 插件工作流来增强自定义工作流。

在 Orchestrator 客户端的 **工作流** 视图中，可以从 **库 > Mail** 访问邮件工作流。

工作流名称	描述
配置邮件	定义 SMTP 服务器的连接、SMTP 身份验证帐户以及发件人的地址和显示名称。
检索消息	使用 POP3 协议检索给定电子邮件帐户的消息。
检索消息（通过 MailClient）	使用 MailClient 类提供的新脚本 API 检索特定电子邮件帐户的消息，而不将其删除。

工作流名称	描述
发送通知	将包含指定内容的电子邮件发送到给定的电子邮件地址。如果未指定可选参数，工作流会使用“配置邮件”工作流中设置的默认值。
向邮件列表发送通知	将包含指定内容的电子邮件发送到给定的电子邮件地址列表、抄送列表和密抄列表。如果未指定可选参数，工作流会使用“配置邮件”工作流中设置的默认值。

使用 Net 插件

11

您可以使用 Net 插件在工作流中实现 Telnet、FTP、POP3 和 IMAP 协议。采用 POP3 和 IMAP 设置可以下载并读取电子邮件。将 Net 插件与 Mail 插件结合使用，即可在工作流中提供完整的电子邮件收发功能。

使用 Enumeration 插件

12

您可以使用 Enumeration 插件在工作流中实现常用枚举类型。

本章讨论了以下主题：

■ 时区代码

时区代码

您可以使用时区代码作为 Enums:MSTimeZone 枚举的可能值。

时区代码	时区名称	描述
000	国际日期变更线标准时间	(GMT-12:00) 国际日期变更线西侧
001	萨摩亚群岛标准时间	(GMT-11:00) 中途岛、萨摩亚
002	夏威夷标准时间	(GMT-10:00) 夏威夷
003	阿拉斯加标准时间	(GMT-09:00) 阿拉斯加
004	太平洋标准时间	(GMT-08:00) 太平洋时间（美国和加拿大）；提华纳
010	山地标准时间	(GMT-07:00) 山地时间（美国和加拿大）
013	墨西哥标准时间 2	(GMT-07:00) 奇瓦瓦、拉巴斯、马萨特兰
015	美国山地标准时间	(GMT-07:00) 亚利桑那
020	中部标准时间	(GMT-06:00) 中部时间（美国和加拿大）
025	加拿大中部标准时间	(GMT-06:00) 萨斯喀彻温
030	墨西哥标准时间	(GMT-06:00) 瓜达拉哈拉、墨西哥城、蒙特雷
033	中美洲标准时间	(GMT-06:00) 中美洲
035	东部标准时间	(GMT-05:00) 东部时间（美国和加拿大）
040	美国东部标准时间	(GMT-05:00) 印地安纳（东部）
045	南美洲太平洋标准时间	(GMT-05:00) 波哥大、利马、基多
050	大西洋标准时间	(GMT-04:00) 大西洋时间（加拿大）
055	南美州西部标准时间	(GMT-04:00) 加拉加斯、拉巴斯

时区代码	时区名称	描述
056	南美州太平洋标准时间	(GMT-04:00) 圣地亚哥
060	纽芬兰和拉布拉多标准时间	(GMT-03:30) 纽芬兰和拉布拉多
065	东南美洲标准时间	(GMT-03:00) 巴西利亚
070	南美洲东部标准时间	(GMT-03:00) 布宜诺斯艾利斯、乔治敦
073	格陵兰标准时间	(GMT-03:00) 格陵兰
075	中大西洋标准时间	(GMT-02:00) 中大西洋
080	亚速尔群岛标准时间	(GMT-01:00) 亚速尔群岛
083	佛得角群岛标准时间	(GMT-01:00) 佛得角群岛
085	GMT 标准时间	(GMT) 格林威治时间：都柏林、爱丁堡、里斯本、伦敦
090	格林威治标准时间	(GMT) 卡萨布兰卡、蒙罗维亚
095	中欧标准时间	(GMT+01:00) 贝尔格莱德、布拉迪斯拉发、布达佩斯、卢布尔雅那、布拉格
100	中欧标准时间	(GMT+01:00) 萨拉热窝、斯科普里、华沙、萨格勒布
105	罗马标准时间	(GMT+01:00) 布鲁塞尔、哥本哈根、马德里、巴黎
110	西欧标准时间	(GMT+01:00) 阿姆斯特丹、柏林、伯尔尼、罗马、斯德哥尔摩、维也纳
113	中非西部标准时间	(GMT+01:00) 中非西部
115	东欧标准时间	(GMT+02:00) 布加勒斯特
120	埃及标准时间	(GMT+02:00) 开罗
125	FLE 标准时间	(GMT+02:00) 赫尔辛基、基辅、里加、索非亚、塔林、维尔纽斯
130	GTB 标准时间	(GMT+02:00) 雅典，伊斯坦布尔，明斯克
135	以色列标准时间	(GMT+02:00) 耶路撒冷
140	南非标准时间	(GMT+02:00) 哈拉雷、比勒陀利亚
145	俄罗斯标准时间	(GMT+03:00) 莫斯科、圣彼得堡、伏尔加格勒
150	阿拉伯标准时间	(GMT+03:00) 科威特、利雅得
155	东非标准时间	(GMT+03:00) 内罗毕
158	阿拉伯标准时间	(GMT+03:00) 巴格达
160	伊朗标准时间	(GMT+03:30) 德黑兰
165	阿拉伯半岛标准时间	(GMT+04:00) 阿布扎比、马斯喀特
170	高加索标准时间	(GMT+04:00) 巴库、第比利斯、埃里温
175	阿富汗标准时间	(GMT+04:30) 喀布尔
180	叶卡捷琳堡标准时间	(GMT+05:00) 叶卡特琳堡

时区代码	时区名称	描述
185	西亚标准时间	(GMT+05:00) 伊斯兰堡、卡拉奇、塔什干
190	印度标准时间	(GMT+05:30) 马德拉斯、加尔各答、孟买、新德里
193	尼泊尔标准时间	(GMT+05:45) 加德满都
195	中亚标准时间	(GMT+06:00) 阿斯塔纳、达卡
200	斯里兰卡标准时间	(GMT+06:00) 斯里贾亚瓦德纳普拉
201	中亚北部标准时间	(GMT+06:00) 阿拉木图、新西伯利亚
203	缅甸标准时间	(GMT+06:30) 仰光
205	东南亚标准时间	(GMT+07:00) 曼谷、河内、雅加达
207	北亚标准时间	(GMT+07:00) 克拉斯诺亚尔斯克
210	中国标准时间	(GMT+08:00) 北京、重庆、香港特别行政区、乌鲁木齐
215	新加坡标准时间	(GMT+08:00) 吉隆坡、新加坡
220	台北标准时间	(GMT+08:00) 台北
225	澳大利亚西部标准时间	(GMT+08:00) 珀斯
227	北亚东部标准时间	(GMT+08:00) 伊尔库次克、乌兰巴托
230	韩国标准时间	(GMT+09:00) 首尔
235	东京标准时间	(GMT+09:00) 大坂、札幌、东京
240	雅库茨克标准时间	(GMT+09:00) 雅库茨克
245	澳大利亚中部标准时间	(GMT+09:30) 达尔文
250	澳大利亚中部标准时间	(GMT+09:30) 阿德莱德
255	澳大利亚东部标准时间	(GMT+10:00) 堪培拉、墨尔本、悉尼
260	澳大利亚西部标准时间	(GMT+10:00) 布里斯班
265	塔斯马尼亚岛标准时间	(GMT+10:00) 霍巴特
270	符拉迪沃斯托克标准时间	(GMT+10:00) 符拉迪沃斯托克
275	太平洋西部标准时间	(GMT+10:00) 关岛、莫尔兹比港
280	太平洋中部标准时间	(GMT+11:00) 马加丹、所罗门群岛、新喀里多尼亚
285	斐济标准时间	(GMT+12:00) 斐济群岛、勘察加半岛、马绍尔群岛
290	新西兰标准时间	(GMT+12:00) 奥克兰、惠灵顿
300	汤加标准时间	(GMT+13:00) 努库阿洛法

使用 Workflow Documentation 插件

13

您可以使用 Workflow Documentation 插件来生成有关特定工作流或工作流类别的 PDF 文档。

本章讨论了以下主题：

- [Workflow Documentation 插件的工作流库](#)
- [生成工作流文档](#)

Workflow Documentation 插件的工作流库

使用 Workflow Documentation 插件工作流，可以生成有关特定工作流或工作流类别的 PDF 文档。

在 Orchestrator 客户端的工作流视图中，可以从库 > **Workflow documentation** 访问以下工作流。

工作流名称	描述
获取工作流文档	生成所选工作流的相关信息。
获取工作流类别的文档	生成所选工作流类别的相关信息。

生成工作流文档

您可以将在任意时间选择的工作流或工作流文件夹的相关文档以 PDF 格式导出。

导出的文档包含有关选定工作流或文件夹中工作流的详细信息。每个工作流相关的信息包括名称、工作流版本历史记录、属性、参数展示、工作流架构和工作流操作。此外，文档还提供了所用操作的源代码。

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 导航到要生成文档的工作流或工作流文件夹，然后右键单击。
- 4 选择**生成文档**。
- 5 浏览并找到要在其中保存 PDF 文件的文件夹，然后提供文件名并单击**保存**。

结果

包含选定工作流或文件夹中工作流相关信息的 PDF 文件会保存在您的系统中。

使用 HTTP-REST 插件

14

HTTP-REST 插件可提供 vRealize Orchestrator 与 REST 主机之间的交互，从而可以管理 REST Web 服务。您可以运行配置工作流将 REST 服务及其操作定义为清单对象，然后在定义的对象上执行 REST 操作。

该插件包含一组与管理 REST 任务和调用 REST 操作相关的标准工作流。您还可以生成自定义工作流来自动处理 REST 环境中的任务。

本章讨论了以下主题：

- [配置 HTTP-REST 插件](#)
- [通过 REST 操作生成新工作流](#)
- [调用 REST 操作](#)

配置 HTTP-REST 插件

您必须使用 Orchestrator 客户端来配置 HTTP-REST 插件。

配置工作流

“配置”工作流类别包含可用于管理 REST 主机的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > HTTP-REST > 配置**访问以下工作流。

工作流名称	描述
添加 REST 主机	将 REST 主机添加到插件清单。
基于 Swagger 规格（以字符串形式）添加 REST 主机	基于 Swagger 规格网络资源（以字符串形式）添加 REST 主机。
基于 Swagger 规格（来自 URL）添加 REST 主机	基于 Swagger 规格（来自特定 URL）添加 REST 主机。
添加 REST 操作	将操作添加到 REST 主机。
将架构添加到 REST 主机	将 XSD 架构添加到 REST 主机。
克隆 REST 主机	创建 REST 主机的克隆。
克隆 REST 操作	创建 REST 操作的克隆。
重新加载插件配置	刷新插件清单中的 REST 主机列表。

工作流名称	描述
移除 REST 主机	移除插件清单中的 REST 主机。
移除 REST 操作	移除 REST 主机中的操作。
移除 REST 主机中的架构	移除 REST 主机中所有关联的 XSD 架构。
更新 REST 主机	更新插件清单中的 REST 主机。
更新 REST 操作	更新 REST 主机上的操作。

配置 Kerberos 身份验证

在添加和管理 PowerShell 主机时可以使用 Kerberos 身份验证。

借助 Kerberos 身份验证，域用户可以通过 WinRM 在启用 PowerShell 的远程计算机上运行命令。

步骤

- 1 在 WinRM 服务上启用 Kerberos 身份验证。

- a 运行以下命令以检查是否允许进行 Kerberos 身份验证。

```
c:\> winrm get winrm/config/service
```

- b 请运行以下命令以启用 Kerberos 身份验证。

```
c:\> winrm set winrm/config/service/auth @{Kerberos="true"}
```

- 2 在 WinRM 客户端上启用 Kerberos 身份验证。

- a 运行以下命令以检查是否允许进行 Kerberos 身份验证。

```
c:\> winrm get winrm/config/client
```

- b 请运行以下命令以启用 Kerberos 身份验证。

```
c:\> winrm set winrm/config/client/auth @{Kerberos="true"}
```

- 3 运行以下命令以测试与 WinRM 服务的连接。

```
c:\> winrm identify -r:http://winrm_server:5985 -auth:Kerberos -u:user_name -p:password -encoding:utf-8
```

- 4 创建 krb5.conf 文件并将其保存到以下位置。

操作系统	路径
Windows	C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\
Linux	用于外部 vRealize Orchestrator 的 /usr/java/jre-vmware/lib/security/。 用于 vRealize Automation 内置 vRealize Orchestrator 的 /etc/krb5.conf。

krb5.conf 文件具有以下结构：

```
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
```

```
[realms]
YOURDOMAIN.COM = {
  kdc = kdc.yourdomain.com
  default_domain = yourdomain.com
}
[domain_realm]
.yourdomain.com=YOURDOMAIN.COM
yourdomain.com=YOURDOMAIN.COM
```

krb5.conf 必须包含特定的配置参数及其值。

Kerberos 配置标记	详细信息
default_realm	客户端用于针对 Active Directory 服务器进行身份验证的默认 Kerberos 领域。 注 必须是大写字母。
kdc	可作为密钥分发中心 (KDC) 并签发 Kerberos 票证的域控制器。
default_domain	用于生成完全限定域名的默认域。 注 该标记面向 Kerberos 4 兼容性。

[注](#) Java Kerberos 配置默认使用 UDP 协议。要仅使用 TCP 协议，您必须指定值为 **1** 的 udp_preference_limit 参数。

[注](#) Kerberos 身份验证要求使用完全限定域名 (Fully Qualified Domain Name, FQDN) 主机地址。

重要事项 添加或修改 krb5.conf 文件时，必须重新启动 Orchestrator 服务器服务。

添加 REST 主机

您可以运行工作流来添加 REST 主机并配置主机连接参数。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的 **工作流** 视图。
- 3 在工作流层次结构列表中，展开 **库 > HTTP-REST > 配置** 并导航到“添加 REST 主机”工作流。
- 4 右键单击“添加 REST 主机”工作流，然后选择 **启动工作流**。
- 5 在 **名称** 文本框中，输入主机的名称。
- 6 在 **URL** 文本框中，输入主机的地址。

[注](#) Kerberos 身份验证要求使用完全限定域名 (Fully Qualified Domain Name, FQDN) 主机地址。

- 7 在 **连接超时** 文本框中，输入连接超时之前的秒数。
- 8 在 **操作超时** 文本框中，输入操作超时之前的秒数。

9 选择是否接受 REST 主机证书。

证书即添加到 Orchestrator 服务器信任存储。

10 选择身份验证类型。

选项	描述
无	无需进行身份验证。
OAuth 1.0	提供所需的身份验证参数。
OAuth 2.0	提供身份验证令牌。
基本	提供基本访问身份验证。 选择会话模式。 <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。
摘要	提供使用加密的摘要访问身份验证。 选择会话模式。 <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。
NTLM	在 Windows 安全支持提供程序 (SSPI) 框架内，提供 NT LAN Manager (NTLM) 访问身份验证。 选择会话模式。 <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。 提供 NTLM 设置。
Kerberos	提供 Kerberos 访问身份验证。 选择会话模式。 <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。

11 若要使用代理，请输入代理服务器的地址和端口。**a (可选) 选择代理身份验证类型。**

选项	描述
无	无需进行身份验证。
基本	提供基本访问身份验证。 选择会话模式。 <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。

12 选择是否要将目标主机名与服务器证书中存储的名称进行匹配。**13 (可选) 选择要用于服务器身份验证的密钥库条目。密钥库条目的类型必须为 PrivateKeyEntry。****14 单击提交以运行工作流。**

结果

工作流成功运行后，REST 主机会显示在**清单**视图中。

后续步骤

您可以将操作和 XSD 架构添加到 REST 主机，并通过**清单**视图运行工作流。

添加 REST 操作

您可以运行工作流将操作添加到插件清单中的 REST 主机。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 REST 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > HTTP-REST > 配置**并导航到“添加 REST 操作”工作流。
- 3 右键单击“添加 REST 操作”工作流，然后选择**启动工作流**。
- 4 选择要向其添加操作的主机。
- 5 在**名称**文本框中，输入操作的名称。
- 6 在**模板 URL** 文本框中，仅输入 URL 的操作部分。

您可以包含运行操作时相关参数所需的占位符。

以下是示例 URL 语法。

```
/customer/{id}/orders?date={date}
```

- 7 选择该项操作要使用的 HTTP 方法。
如果选择 **POST** 或 **PUT**，您可以为方法提供内容类型请求标头。
- 8 单击**提交**以运行工作流。

后续步骤

您可以在**清单**视图对操作运行工作流。

将架构添加到 REST 主机

您可以运行工作流将 XSD 架构添加到插件清单中的 REST 主机。

XSD 架构描述了可作为 Web 服务输入和输出内容使用的 XML 文档。通过将此类架构与主机关联，您可以指定在通过 REST 操作生成工作流时需作为输入参数的 XML 元素。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。

- 验证您是否从**清单**视图与 REST 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > HTTP-REST > 配置**并导航到“将架构添加到 REST 主机”工作流。
- 3 右键单击“将架构添加到 REST 主机”工作流，然后选择**启动工作流**。
- 4 选择要向其添加 XSD 架构的主机。
- 5 选择是否要从 URL 加载架构。

选项	操作
是	输入架构的 URL。
否	提供架构内容。

- 6 单击**提交**以运行工作流。

通过 REST 操作生成新工作流

您可以通过 REST 操作创建自定义工作流。

您可以将自定义生成的工作流集成到高级别工作流中。有关工作流开发的更多信息，请参见《vRealize Orchestrator 开发人员手册》。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 REST 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > HTTP-REST**并导航到“通过 REST 操作生成新工作流”工作流。
- 3 右键单击“通过 REST 操作生成新工作流”工作流，然后选择**启动工作流**。
- 4 在可用操作列表中选择 REST 操作。
如果操作采用输入并且已将 XSD 架构添加到其主机，您可以指定请求输入类型。
- 5 在**名称**文本框中，输入要生成的工作流的名称。
- 6 选择要在其中生成新工作流的工作流文件夹。
您可以选择工作流库中的任意现有文件夹。
- 7 单击**提交**以运行工作流。

调用 REST 操作

若要发起 REST 请求，您可以调用已配置的 REST 操作或调用动态的 REST 操作，方法是将配置后的 REST 操作作为模板并替换运行时的任何参数。

调用 REST 操作的方法有多种。

- 运行**添加 REST 主机**和**添加 REST 操作**工作流以配置 REST 主机并关联 REST 操作。注册的 REST 主机和 REST 操作具有持久性，且可在**清单**和**资源**视图找到。
- 运行**库 > HTTP-REST** 示例中的**调用动态 REST 操作**工作流来调用 REST 操作，而无需先配置 REST 主机并添加 REST 操作。使用该工作流，可以提供 REST 主机库 URL 和操作参数。此数据为非持久数据且不在**清单**和**资源**视图中提供。
- 运行**库 > HTTP-REST** 示例中的**使用动态参数调用 REST 主机**和**使用动态参数调用 REST 操作**工作流，从而配置 REST 主机并关联 REST 操作，同时将配置好的 REST 主机和 REST 操作作为未来项目的模板使用。运行工作流时，您可以替换已配置的 REST 主机和 REST 操作的部分参数。原始 REST 主机和 REST 操作不受影响。

调用 REST 操作

直接调用 REST 操作

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 REST 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > HTTP-REST** 并导航到**调用 REST 操作**工作流。
- 3 右键单击**调用 REST 操作**工作流并选择**启动工作流**。
- 4 在可用操作列表中选择 REST 操作。
- 5 提供操作所需的输入参数和内容。
- 6 单击**提交**以运行工作流。

使用 SOAP 插件

15

SOAP 插件可提供 vRealize Orchestrator 与 SOAP 主机之间的交互，从而可以管理 SOAP Web 服务。您可以运行配置工作流将 SOAP 服务定义为服务对象，然后在定义的对象上执行 SOAP 操作。

该插件包含一组与管理 SOAP 任务和调用 SOAP 操作相关的标准工作流。您还可以生成自定义工作流来自动处理 SOAP 环境中的任务。

本章讨论了以下主题：

- [配置 SOAP 插件](#)
- [通过 SOAP 操作生成新工作流](#)
- [调用 SOAP 操作](#)

配置 SOAP 插件

您必须使用 Orchestrator 客户端来配置 SOAP 插件。

配置工作流

“配置”工作流类别包含可用于管理 SOAP 主机的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > SOAP > 配置**访问以下工作流。

工作流名称	描述
添加 SOAP 主机	将 SOAP 主机添加到插件清单。
重新加载插件配置	刷新插件清单中的 SOAP 主机列表。
移除 SOAP 主机	移除插件清单中的 SOAP 主机。 小心 从清单移除主机时，该主机生成的所有工作流都会停止工作。
更新 SOAP 主机	更新插件清单中的 SOAP 主机。
更新 SOAP 主机的端点 URL	更新 SOAP 主机的首选端点地址。新端点地址将取代 WSDL 中定义的端点地址，用于发送和接收 SOAP 消息。

添加 SOAP 主机

您可以运行工作流来添加 SOAP 主机并配置主机连接参数。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > SOAP > 配置**并导航到**添加 SOAP 主机**工作流。
- 4 右键单击**添加 SOAP 主机**工作流，然后选择**启动工作流**。
- 5 在**名称**文本框中，输入主机的名称。
- 6 选择是否将 WSDL 内容以文本形式提供。

选项	操作
是	复制 WSDL 内容 文本框中的文本。
否	在 WSDL URL 文本框中输入正确的路径。

- 7 在**连接超时**文本框中输入秒数，Orchestrator 必须在此时间内连接到 SOAP 主机，否则连接超时。
- 8 在**请求超时**文本框中指定秒数，SOAP 请求必须在此时间内成功，否则视为超时。
- 9 选择是否使用代理。

选项	操作
是	提供代理地址和代理端口。
否	继续下一步操作。

- 10 选择身份验证类型。

选项	描述
无	无需进行身份验证。
基本	提供基本访问身份验证。 选择会话模式。 <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。
摘要	提供使用加密的摘要访问身份验证。 选择会话模式。 <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。

选项	描述
NTLM	<p>在 Windows 安全支持提供程序 (SSPI) 框架内，提供 NT LAN Manager (NTLM) 访问身份验证。</p> <p>选择会话模式。</p> <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。 <p>提供 NTLM 设置。</p>
Kerberos	<p>提供 Kerberos 访问身份验证。</p> <p>选择会话模式。</p> <ul style="list-style-type: none"> ■ 如果选择共享会话，请提供共享会话的凭据。 ■ 如果选择单用户会话，则 Orchestrator 客户端会检索已登录用户的凭据。

11 单击**提交**以运行工作流。

结果

工作流成功运行后，SOAP 主机会显示在**清单**视图中。

后续步骤

您可以浏览 SOAP 主机对象并在**清单**视图中对其运行工作流。

配置 Kerberos 身份验证

在添加和管理 PowerShell 主机时可以使用 Kerberos 身份验证。

借助 Kerberos 身份验证，域用户可以通过 WinRM 在启用 PowerShell 的远程计算机上运行命令。

步骤

1 在 WinRM 服务上启用 Kerberos 身份验证。

a 运行以下命令以检查是否允许进行 Kerberos 身份验证。

```
c:\> winrm get winrm/config/service
```

b 请运行以下命令以启用 Kerberos 身份验证。

```
c:\> winrm set winrm/config/service/auth @{Kerberos="true"}
```

2 在 WinRM 客户端上启用 Kerberos 身份验证。

a 运行以下命令以检查是否允许进行 Kerberos 身份验证。

```
c:\> winrm get winrm/config/client
```

b 请运行以下命令以启用 Kerberos 身份验证。

```
c:\> winrm set winrm/config/client/auth @{Kerberos="true"}
```

3 运行以下命令以测试与 WinRM 服务的连接。

```
c:\> winrm identify -r:http://winrm_server:5985 -auth:Kerberos -u:user_name -p:password -encoding:utf-8
```

4 创建 krb5.conf 文件并将其保存到以下位置。

操作系统	路径
Windows	C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\
Linux	用于外部 vRealize Orchestrator 的 /usr/java/jre-vmware/lib/security/。 用于 vRealize Automation 内置 vRealize Orchestrator 的 /etc/krb5.conf。

krb5.conf 文件具有以下结构：

```
[libdefaults]
default_realm = YOURDOMAIN.COM
udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
kdc = kdc.yourdomain.com
default_domain = yourdomain.com
}
[domain_realm]
.yourdomain.com=YOURDOMAIN.COM
yourdomain.com=YOURDOMAIN.COM
```

krb5.conf 必须包含特定的配置参数及其值。

Kerberos 配置标记	详细信息
default_realm	客户端用于针对 Active Directory 服务器进行身份验证的默认 Kerberos 领域。 注 必须是大写字母。
kdc	可作为密钥分发中心 (KDC) 并签发 Kerberos 票证的域控制器。
default_domain	用于生成完全限定域名的默认域。 注 该标记面向 Kerberos 4 兼容性。

注 Java Kerberos 配置默认使用 UDP 协议。要仅使用 TCP 协议，您必须指定值为 **1** 的 `udp_preference_limit` 参数。

注 Kerberos 身份验证要求使用完全限定域名 (Fully Qualified Domain Name, FQDN) 主机地址。

重要事项 添加或修改 `krb5.conf` 文件时，必须重新启动 Orchestrator 服务器服务。

通过 SOAP 操作生成新工作流

您可以通过 SOAP 操作创建自定义工作流。

您可以将自定义生成的工作流集成到高级别工作流中。有关工作流开发的更多信息，请参见《vRealize Orchestrator 开发人员手册》。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SOAP 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SOAP** 并导航到“通过 SOAP 操作生成新工作流”工作流。
- 3 右键单击“通过 SOAP 操作生成新工作流”工作流，然后选择**启动工作流**。
- 4 在可用操作列表中选择 SOAP 操作。
- 5 在**名称**文本框中，输入要生成的工作流的名称。
- 6 选择要在其中生成新工作流的工作流文件夹。
您可以选择工作流库中的任意现有文件夹。
- 7 单击**提交**以运行工作流。

后续步骤

您可以测试生成的工作流。

测试自定义生成的工作流

您可以运行 SOAP 操作生成的自定义工作流来获取操作的输出参数。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SOAP 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 导航到工作流位置。
- 3 右键单击自定义工作流，然后选择**启动工作流**。
- 4 提供 SOAP 操作需要的输入参数。
- 5 单击**提交**以运行工作流。
- 6 （可选）在**日志**选项卡中，查看可用输出参数的列表。

调用 SOAP 操作

您可以直接调用 SOAP 操作，无需生成新工作流。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SOAP 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SOAP** 并导航到“调用 SOAP 操作”工作流。
- 3 右键单击“调用 SOAP 操作”工作流并选择**启动工作流**。
- 4 在可用操作列表中选择 SOAP 操作。
- 5 提供 SOAP 操作需要的输入参数。
- 6 单击**提交**以运行工作流。
- 7 （可选）在**日志**选项卡中，查看可用输出参数的列表。

使用 AMQP 插件

16

AMQP 插件允许您与高级消息队列协议 (Advanced Message Queuing Protocol, AMQP) 服务器（也称为“代理”）进行交互。您可以运行配置工作流将 AMQP 代理和队列订阅定义为清单对象，然后在定义的对象上执行 AMQP 操作。

该插件包含一组与管理 AMQP 代理和调用 AMQP 操作相关的标准工作流。

本章讨论了以下主题：

- [配置 AMQP 插件](#)
- [使用 AMQP 插件标准工作流](#)

配置 AMQP 插件

您必须使用 Orchestrator 客户端来配置 AMQP 插件。

配置工作流

“配置”工作流类别包含可用于管理 AMQP 代理的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > AMQP > 配置**访问以下工作流。

工作流名称	描述
添加代理	添加 AMQP 代理。
移除代理	移除 AMQP 代理。
移除订阅	移除 AMQP 消息订阅。
订阅队列	创建新订阅元素。
更新代理	更新代理属性。
验证代理	通过尝试启动连接来验证代理。

添加代理

您可以运行工作流来添加 AMQP 代理。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > AMQP > 配置**并导航到“添加代理”工作流。
- 4 右键单击“添加代理”工作流，然后选择**启动工作流**。
- 5 提供“添加代理”工作流所需的信息。

选项	操作
名称	输入代理的名称。
主机	输入主机的地址。
端口	输入 AMQP 代理服务的端口。默认端口为 5672。
虚拟主机	输入虚拟主机的地址。提供的默认值为 /。
使用 SSL	选择是否使用 SSL 证书。
接受所有证书	选择是否接受所有 SSL 证书且不进行验证。
用户名	输入代理的用户名。
密码	输入代理的密码。

- 6 单击**提交**以运行工作流。

结果

工作流成功运行后，AMQP 代理会显示在**清单**视图中。

后续步骤

您可以运行“验证代理”工作流。如果发生错误，请先使用“更新代理”工作流更改代理的属性，然后再次验证。

订阅队列

您可以运行工作流来创建新的订阅元素。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 AMQP 代理建立了连接。
- 验证 AMQP 代理是否已在声明的订阅中包含了所有队列。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > AMQP > 配置**并导航到“订阅队列”工作流。
- 3 右键单击“订阅队列”工作流，然后选择**启动工作流**。
- 4 在**名称**文本框中，输入要显示的队列名称。
- 5 选择要在其中添加订阅的代理。
- 6 选择用于消息订阅的所有队列。
- 7 单击**提交**以运行工作流。

结果

工作流成功运行后，代理的子项会显示在**清单**视图中。

后续步骤

您可以创建策略。

更新代理

您可以运行工作流来更新代理属性。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 AMQP 代理建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > AMQP**并导航到“更新代理”工作流。
- 3 右键单击“更新代理”工作流，然后选择**启动工作流**。
- 4 选择要更新的代理。
此时会显示代理的当前属性。
- 5 编辑所需的属性。
- 6 单击**提交**以运行工作流。

使用 AMQP 插件标准工作流

AMQP 工作流类别包含可用于运行 AMQP 操作的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > AMQP**访问以下工作流。

工作流名称	描述
绑定	在指定代理中创建绑定。
声明队列	将队列添加到指定代理。
声明交换	将交换添加到指定代理。
删除队列	删除指定代理中的队列。
删除交换	删除指定代理中的交换。
接收文本消息	从指定代理接收文本消息。
发送测试消息	使用指定代理发送文本消息。
取消绑定	取消指定代理中的绑定。

声明绑定

您可以运行工作流在指定代理中创建绑定。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 AMQP 代理建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > AMQP** 并导航到“绑定”工作流。
- 3 右键单击“绑定”工作流，然后选择**启动工作流**。
- 4 选择要在其中创建绑定的代理。
- 5 提供绑定的相关信息。

选项	操作
队列名称	输入队列的名称。
交换名称	输入交换的名称。
路由密钥	输入路由密钥。

- 6 单击**提交**以运行工作流。

声明队列

您可以运行工作流将队列添加到指定代理。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 AMQP 代理建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > AMQP** 并导航到“声明队列”工作流。
- 3 右键单击“声明队列”工作流，然后选择**启动工作流**。
- 4 选择要向其添加队列的代理。
- 5 在**名称**文本框中，输入要显示的队列名称。
- 6 选择队列是否可持久使用。

选项	描述
是	该队列会在重启代理后被移除。
否	该队列会在重启代理后继续保留。

- 7 选择是否为特定队列设置专用客户端。

选项	描述
是	为该特定队列设置一个客户端。
否	为该特定队列设置多个客户端。

- 8 选择是否自动删除带有激活订阅的队列。

选项	描述
是	当无任何客户端与队列连接时自动将其删除。该交换只要有一个客户端订阅，就会继续保留。
否	不自动删除队列。

- 9 单击**提交**以运行工作流。

声明交换

您可以运行工作流在指定代理中添加交换。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 AMQP 代理建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > AMQP** 并导航到“声明交换”工作流。
- 3 右键单击“声明交换”工作流，然后选择**启动工作流**。
- 4 选择要向其添加交换的代理。

5 在**名称**文本框中，输入交换的名称。

6 选择交换类型。

选项	描述
直接	在消息所提供的路由密钥与队列绑定到此交换时所使用的路由条件之间建立直接配对。
扇出	将发送到此交换的所有消息转发到与此交换绑定的所有队列。与此交换绑定的队列中不包含任何参数。
标头	队列可与此交换绑定，其参数表中可包含标头和值。一个名为 x-match 的特殊参数用于确定匹配算法。
主题	在路由密钥与绑定中指定的路由模式之间执行通配符匹配。

7 选择交换是否可持久使用。

选项	描述
是	该交换会在重启代理后继续保留。
否	该交换会在重启代理后被移除。

8 选择是否自动删除带有激活订阅的交换。

选项	描述
是	当无任何队列与交换绑定时自动将其删除。该交换只要绑定有一个队列，就会继续保留。
否	不自动删除交换。

9 单击**提交**以运行工作流。

发送文本消息

您可以运行工作流使用指定代理发送文本消息。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 AMQP 代理建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > AMQP**并导航到“发送文本消息”工作流。
- 3 右键单击“发送文本消息”工作流，然后选择**启动工作流**。
- 4 选择要通过其来发送消息的代理。
- 5 在**交换名称**文本框中，指定交换的名称。

- 6 在**路由密钥**文本框中，指定路由密钥。
- 7 在**内容**文本框中，输入要发送的消息。
- 8 单击**提交**以运行工作流。

删除绑定

您可以运行工作流在指定代理中删除绑定。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 AMQP 代理建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > AMQP** 并导航到“取消绑定”工作流。
- 3 右键单击“取消绑定”工作流，然后选择**启动工作流**。
- 4 选择要从其移除绑定的代理。
- 5 提供绑定的相关信息。

选项	操作
队列名称	指定队列的名称。
交换名称	指定交换的名称。
路由密钥	指定路由密钥。

- 6 单击**提交**以运行工作流。

使用 SNMP 插件

17

SNMP 插件可将 vRealize Orchestrator 连接到启用了 SNMP 的系统和设备，并接收它们的信息。您可以运行工作流将 SNMP 设备定义为清单对象，然后在定义的对象上执行 SNMP 操作。

您可以使用插件连接到 SNMP 设备，例如路由器、交换机、网络打印机和 UPS 设备。该插件还可通过 SNMP 协议接收 vCenter Server 的事件。

SNMP 插件提供了两种与 SNMP 设备通信的方法。

- 查询特定 SNMP 变量的值。
- 侦听设备生成的事件（SNMP 陷阱）并将其推送到注册的 SNMP 管理器。

该插件包含一组与管理 SNMP 设备、查询、陷阱主机以及执行 SNMP 操作相关的标准工作流。您还可以创建自定义工作流来自动处理 SNMP 环境中的任务。

本章讨论了以下主题：

- [管理 SNMP 设备](#)
- [管理 SNMP 查询](#)
- [管理 SNMP 陷阱主机](#)
- [接收 SNMP 陷阱](#)
- [通用 SNMP 请求工作流](#)

管理 SNMP 设备

您可以运行工作流在 Orchestrator 上注册 SNMP 设备、编辑现有设备的设置并取消注册设备。

设备管理工作流

“设备管理”工作流类别包含可用于管理 SNMP 设备的工作流。

在 Orchestrator 客户端的工作流视图中，可以从 **库 > SNMP > 设备管理** 访问以下工作流。

工作流名称	描述
编辑 SNMP 设备	编辑已注册 SNMP 设备的配置。
注册 SNMP 设备	将启用了 SNMP 的设备注册到插件清单。
取消注册 SNMP 设备	从插件清单中取消 SNMP 设备注册。

注册 SNMP 设备

您可以运行工作流来注册 SNMP 设备并（可选）配置高级连接参数。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > SNMP > 设备管理**并导航到“注册 SNMP 设备”工作流。
- 4 右键单击“注册 SNMP 设备”工作流，然后选择**启动工作流**。
- 5 在**设备地址**文本框中，输入 SNMP 设备的 IP 地址或 DNS 名称。

注 若要建立更加可靠的连接，应使用 IP 地址。

- 6 （可选）在**名称**文本框中，输入想要在**清单**视图中显示的设备的名称。

如果将文本框留空，则会使用设备地址自动生成名称。

- 7 （可选）若要配置高级连接参数，请选择**是**。

- a 在**端口**文本框中，指定连接端口。

默认端口为 161。

- b 在**版本**下拉菜单中，选择要使用的 SNMP 版本并提供凭据。

对 SNMPv3 的支持限制为 AuthPriv 安全级别与 MD5 身份验证。DES 密码短语与 MD5 密码相同。

注 SNMPv3 的支持已弃用。

- 8 单击**提交**以运行工作流。

结果

工作流成功运行后，SNMP 设备会显示在**清单**视图中。

后续步骤

您可以将查询添加到 SNMP 设备，并通过**清单**视图运行工作流。

管理 SNMP 查询

您可以将查询添加到已注册的 SNMP 设备，运行、复制和编辑现有查询并从 SNMP 设备移除查询。您可以将 SNMP 查询用作更复杂的工作流的构建基块。

查询管理工作流

“查询管理”工作流类别包含可用于管理 SNMP 查询的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > SNMP > 查询管理**访问以下工作流。

工作流名称	描述
将查询添加到 SNMP 设备	将查询添加到 SNMP 设备。
复制 SNMP 查询	将 SNMP 查询从一台设备复制到另一台。
编辑 SNMP 查询	编辑现有 SNMP 查询。
移除 SNMP 设备中的查询	移除设备中的 SNMP 查询。
运行 SNMP 查询	对 SNMP 设备运行查询。

将查询添加到 SNMP 设备

您可以运行工作流将查询添加到插件清单中的 SNMP 设备。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SNMP 设备建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SNMP > 查看管理**并导航到“将查询添加到 SNMP 设备”工作流。
- 3 右键单击“将查询添加到 SNMP 设备”工作流，然后选择**启动工作流**。
- 4 选择要向其添加查询的设备。
- 5 在**类型**下拉菜单中，选择查询类型。
- 6 在 **OID** 文本框中，输入要查询的变量的对象标识符。

以下为示例 OID 值。

- 1.3.6.1.2.1.1.5.0
- .1.3.6.1.2.1.1.5.0
- iso.3.6.1.2.1.1.5.0

注 该插件仅支持纯数字或以 iso 开头且其余仍为数字的 OID 值。

7 （可选）在**名称**文本框中，输入查询的名称。

如果将文本框留空，则会使用类型和 OID 参数自动生成名称。

8 单击**提交**以运行工作流。

后续步骤

您可以在**清单**视图中对查询运行工作流。

管理 SNMP 陷阱主机

vRealize Orchestrator 可以充当 SNMP 侦听器。您可以启动和停止 SNMP 陷阱主机，并更改 Orchestrator 用来侦听 SNMP 陷阱的端口。

SNMP 插件支持 SNMPv1 和 SNMPv2c 陷阱。

注 SNMPv3 的支持已弃用。

陷阱主机管理工作流

“陷阱主机管理”工作流类别包含可用于管理 SNMP 陷阱主机的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > SNMP > 陷阱主机管理**访问以下工作流。

工作流名称	描述
设置 SNMP 陷阱端口	设置 Orchestrator 用来侦听 SNMP 陷阱的端口。
启动陷阱主机	Orchestrator 开始侦听 SNMP 陷阱。
停止陷阱主机	Orchestrator 停止侦听 SNMP 陷阱。

设置 SNMP 陷阱端口

您可以运行工作流来设置 Orchestrator 用来侦听 SNMP 陷阱的端口。

SNMP 陷阱的默认端口为 162。但是在 Linux 系统中，您可以打开 1024 下的端口，但只能由超级用户权限实现。

注 若要确保更好的兼容性，SNMP 插件中用于侦听 SNMP 陷阱的默认端口应设置为 4000。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SNMP 设备建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SNMP > 陷阱主机管理**并导航到“设置 SNMP 陷阱端口”工作流。

- 3 右键单击“设置 SNMP 陷阱端口”工作流，然后选择**启动工作流**。
- 4 在**端口**文本框中，输入 Orchestrator 用来侦听 SNMP 陷阱的端口号。
- 5 单击**提交**以运行工作流。

结果

工作流即停止陷阱主机、设置新端口并再次启动陷阱主机。

接收 SNMP 陷阱

SNMP 插件可以通过运行工作流（等待单条陷阱消息）或通过策略（可连续处理陷阱）来接收 SNMP 陷阱。该插件支持 SNMPv1 和 SNMPv2c 陷阱。

等待 SNMP 设备上的陷阱

您可以运行工作流来等待接收指定设备的 SNMP 陷阱。

此工作流包含一个触发器，会停止工作流运行并等待 SNMP 陷阱，然后再继续运行。收到陷阱后，工作流会恢复运行。您可以在更复杂的工作流中使用该工作流，或将其用作示例以根据特殊需求进行自定义或扩展。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SNMP 设备建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > SNMP** 并导航到“等待 SNMP 设备上的陷阱”工作流。
- 3 右键单击“等待 SNMP 设备上的陷阱”工作流，然后选择**启动工作流**。
- 4 选择要在其中等待 SNMP 陷阱的设备。
- 5 （可选）在 **OID** 文本框中，输入特定陷阱的对象标识符。

注 如果将文本框留空，则工作流会在收到指定 SNMP 设备的任何陷阱后恢复运行。

- 6 单击**提交**以运行工作流。

设置 SNMP 陷阱策略

您可以设置策略以持续侦听已在插件清单中注册的 SNMP 设备的陷阱。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SNMP 设备建立了连接。

步骤

- 1 从 Orchestrator 客户端的下拉菜单中，选择**管理**。
- 2 单击**策略模板**视图。
- 3 在工作流层次结构列表中，展开**库 > SNMP** 并导航到 SNMP 陷阱策略模板。
- 4 右键单击 SNMP 陷阱策略模板，然后单击**应用策略**。
- 5 在**策略名称**文本框中，输入要创建的策略的名称。
- 6 （可选）在**策略描述**文本框中，输入策略的描述。
- 7 选择要在其中设置策略的 SNMP 设备。
- 8 单击**提交**以创建策略。

Orchestrator 客户端切换到**运行**视图。

- 9 在**策略**视图中，右键单击已创建的策略并选择**启动策略**。

结果

陷阱策略即开始侦听 SNMP 陷阱。

后续步骤

您可以编辑 SNMP 陷阱策略。

配置 SNMP 陷阱主机策略

SNMP 陷阱主机策略用于侦听可能未添加为已注册的 SNMP 设备的主机的 SNMP 陷阱。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SNMP 设备建立了连接。

步骤

- 1 从 Orchestrator 客户端的下拉菜单中，选择**管理**。
- 2 单击**策略模板**视图。
- 3 在工作流层次结构列表中，展开**库 > SNMP** 并导航到 SNMP 陷阱主机策略模板。
- 4 右键单击 SNMP 陷阱主机策略模板，然后单击**应用策略**。
- 5 在**策略名称**文本框中，输入要创建的策略的名称。
- 6 （可选）在**策略描述**文本框中，输入策略的描述。
- 7 从清单树中选择 Trap Host (Online)。
- 8 单击**提交**以创建策略。

Orchestrator 客户端切换到**运行**视图。

- 9 右键单击策略并选择**编辑**。
- 10 在**脚本**选项卡中，展开**主机 > OnTrapAll**。
 - a 选择工作流或脚本与该策略关联。
- 11 单击**保存并关闭**以应用编辑后的设置。
- 12 在**策略**视图中，右键单击已编辑的策略并选择**启动策略**。

后续步骤

您可以编辑 SNMP 陷阱主机策略。

编辑陷阱策略

您可以编辑并自定义陷阱策略以用于特定用例。编辑陷阱策略时，您可以更改其优先级和启动设置，并自定义设置与策略关联的脚本和权限。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 SNMP 设备建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**策略**视图。
- 2 如果想要编辑的策略正在运行，请右键单击该策略并选择**停止策略**。
- 3 右键单击策略并选择**编辑**。
- 4 在**常规**选项卡上，编辑策略的启动设置、优先级和描述。
- 5 （可选）在**脚本**选项卡上，您可以将特定工作流或脚本代码与策略关联，用于更复杂场景中的集成。
您可以设置策略以在收到陷阱时触发自定义工作流。
- 6 （可选）在**权限**选项卡上，您可以修改访问权限。
您可以向用户或组授予策略启动权限，而不授予策略编辑权限。
- 7 单击**保存并关闭**以应用编辑后的设置。
- 8 在**策略**视图中，右键单击已编辑的策略并选择**启动策略**。

通用 SNMP 请求工作流

SNMP 工作流类别包含可用于执行基本的 SNMP 请求（无需创建查询）的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > SNMP** 访问以下工作流。

工作流名称	描述
获取批量 SNMP 值	对 SNMP 设备运行 GET BULK 查询。
获取下一 SNMP 值	对 SNMP 设备运行 GET NEXT 查询。

工作流名称	描述
获取 SNMP 值	对 SNMP 设备运行 GET 查询。
发送 SNMP 陷阱	将 SNMP 陷阱发送到指定地址。
等待所有设备上的陷阱	等待从发送陷阱到 Orchestrator 的所有主机接收 SNMP 陷阱。
等待 SNMP 设备上的陷阱	等待接收指定设备的 SNMP 陷阱。

使用 Active Directory 插件

18

Active Directory 插件 (VMware vRealize Orchestrator plug-in for Microsoft Active Directory) 可在 vRealize Orchestrator 和 Microsoft Active Directory 之间实现交互。您可以使用插件运行可自动处理 Active Directory 操作的 Orchestrator 工作流。

该插件包含一组标准工作流。您还可以创建采用插件 API 的自定义工作流，自动处理 Active Directory 环境中的任务。

本章讨论了以下主题：

- [配置 Active Directory 插件](#)
- [使用 Active Directory 插件工作流库](#)

配置 Active Directory 插件

若要使用 Active Directory 插件连接到 Microsoft Active Directory 实例，必须为 Microsoft Active Directory 实例配置连接参数。

您可以运行插件中包含的配置工作流来配置 Active Directory。

Active Directory 配置工作流

Active Directory 插件的“配置”工作流类别包含可用于配置 Active Directory 服务器并管理 SSL 证书的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > Microsoft > Active Directory > 配置**访问以下工作流。

工作流名称	描述
添加 Active Directory 服务器	配置 Microsoft Active Directory 服务器。
配置 Active Directory 插件选项	配置 Active Directory 插件的搜索限制选项。
配置 Active Directory 服务器（已弃用）	创建或更新默认 Active Directory 服务器配置。使用“更新 Active Directory 服务器”。
移除 Active Directory 服务器	移除 Active Directory 服务器配置。
重置配置（已弃用）	删除默认 Active Directory 服务器配置。使用“移除 Active Directory 服务器”。

工作流名称	描述
更新 Active Directory 服务器	修改现有 Active Directory 服务器配置。

使用 Active Directory 插件工作流库

Active Directory 插件工作流库中包含多种可用于运行与管理 Microsoft Active Directory 对象相关自动处理操作的工作流。

工作流会根据对象类型分成不同类别。您可以在自定义工作流中集成工作流库中的标准工作流。

使用 Active Directory 插件清单

Active Directory 插件会在**清单**视图中公开所连接的 Microsoft Active Directory 示例中的所有对象。您可以使用**清单**视图添加授权元素或在 Microsoft Active Directory 对象上运行工作流。

要显示可供清单对象使用的工作流，请导航到**工具 > 用户首选项 > 清单**，然后选中在**清单中使用上下文菜单**复选框。启用此选项后，当您右键单击 Orchestrator 清单中的对象时，将显示可供该对象使用的所有工作流。

访问 Active Directory 插件工作流库

您必须使用 Orchestrator 客户端访问 Active Directory 插件工作流库中的元素。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击**工作流**视图。
- 3 在层次结构列表中，展开 **库 > Microsoft > Active Directory** 并展开选择内容。

Active Directory 插件工作流

Active Directory 插件包含一组标准工作流，涵盖最常用的 LDAP 功能。您可以将工作流作为构建块，用于创建复杂的自定义解决方案。您可通过将标准工作流合并，在 Active Directory 环境中自动处理多步骤的操作。

计算机工作流

“计算机”工作流类别包含与 Active Directory 计算机管理相关的工作流。

您可以通过**库 > Microsoft > Active Directory > 计算机**访问以下工作流。

工作流名称	描述
在组中创建计算机	在组中创建 Active Directory 计算机。
在组织单位中创建计算机	在组织单位中创建 Active Directory 计算机。
销毁计算机	删除 Active Directory 实例中的计算机。
销毁计算机并删除其子树	删除 Active Directory 实例中的计算机以及计算机子树中的所有对象。

工作流名称	描述
禁用计算机	在 Active Directory 实例中禁用计算机。
启用计算机	在 Active Directory 实例中启用计算机。

组织单位工作流

“组织单位”工作流类别包含与 Active Directory 组织单位管理相关的工作流。

您可以通过**库 > Microsoft > Active Directory > 组织单位**访问以下工作流。

工作流名称	描述
创建组织单位	在现有组织单位中创建组织单位。
销毁组织单位	删除 Active Directory 实例中的组织单位。
销毁组织单位并删除其子树	删除 Active Directory 实例中的组织单位以及组织单位子树中的所有对象。

用户工作流

“用户”工作流类别包含与 Active Directory 用户管理相关的工作流。

您可以通过**库 > Microsoft > Active Directory > 用户**访问以下工作流。

工作流名称	描述
将用户添加到用户组	将一个用户添加为用户组成员。
更改用户密码	更改用户的密码。需要 SSL 连接，并且密码必须满足 Active Directory 限制。
在组中创建用户	创建用户而不指定密码。必须在下次登录时更改密码。域策略必须允许用户拥有空密码。
在组织单位中创建用户	在组织单位中创建用户。如果禁用了 SSL 连接，则无法指定密码。域策略必须允许用户拥有空密码。
在组中创建用户和密码	创建用户并为其设置密码。可以在下次登录时更改密码。
在组织单位中创建用户和密码	在组织单位中创建一个用户并为其设置密码。可以在下次登录时更改密码。如果禁用了 SSL 连接，则无法指定密码。
销毁用户	删除 Active Directory 实例中的用户。
禁用用户	在 Active Directory 实例中禁用用户。
启用用户	在 Active Directory 实例中启用用户。
移除用户组中的用户	移除用户组中的用户。

用户组工作流

“用户组”工作流类别包含与 Active Directory 用户组管理相关的工作流。

您可以通过**库 > Microsoft > Active Directory > 用户组**访问以下工作流。

工作流名称	描述
将计算机添加到组成员	将一台或多台计算机添加为用户组成员。
将组添加到组成员	将一个或多个用户组添加为用户组成员。
将用户添加到组成员	将一个或多个用户添加为用户组成员。

workflow名称	描述
在组中创建用户组	在现有容器（组）中创建用户组。
在组中创建用户组并设置属性 “Group name (pre-Windows 2000)”	在现有容器（组织单位）中创建用户组并设置 Group name (pre-Windows 2000) 属性。
在组织单位中创建用户组	在现有容器（组织单位）中创建用户组。
销毁用户组	删除 Active Directory 实例中的用户组。
移除组成员中的计算机	移除用户组中的一台或多台计算机。
移除组成员中的组	移除用户组中的一个或多个用户组。
移除组成员中的用户	移除用户组中的一个或多个用户。

使用 Dynamic Types 插件

19

Orchestrator Dynamic Types 插件可用于定义动态类型、创建这些类型的对象和设置彼此间的关系。

动态类型的定义包含对其属性的描述，以及一组可用于查找该类型动态对象的查找器工作流程和操作。动态类型的运行时实例称为动态对象。您可以在自己创建的动态对象上运行工作流程并对其进行不同操作。

必须在命名空间中定义每个动态类型。命名空间是帮助程序动态对象，可用于对容器中的动态类型进行分组。

您可以结合使用 Dynamic Types 插件以及 HTTP-REST 插件将第三方 REST API 服务集成到 Orchestrator 并将第三方对象公开为 Orchestrator 类型。

- 1 运行 Dynamic Types 插件中的“定义命名空间”和“定义类型”工作流程来定义新的动态类型。这样，您就能获得一组查找器和清单工作流程，用来查找新动态类型的对象及其与其他对象之间的关系。
- 2 修改新查找器和清单工作流程，以便其接收第三方 REST API 的输入。
 - a 使用 HTTP-REST 插件中的“添加 REST 操作”工作流程创建 REST 操作，并将这些操作映射到相应的 REST API 方法。
 - b 修改查找器和清单工作流程来调用这些 REST 操作并使用其输出。

本章讨论了以下主题：

- [Dynamic Types 配置工作流程](#)

Dynamic Types 配置工作流程

Dynamic Types 插件的“配置”软件包中的工作流程可用于创建动态类型、从 XSD 文件导出和导入类型定义，以及定义所创建动态类型之间的关系。

在 Orchestrator 客户端的工作流视图中，可以从库 > **Dynamic Types** > **配置** 访问以下工作流程。

工作流程名称	描述
定义命名空间	定义新的命名空间。
定义关系	定义类型之间的新关系。
定义类型	定义给定命名空间内的新类型。
将配置导出为软件包	将 Dynamic Types 定义配置导出为基于文件的配置。导出的软件包可用于导入到其他服务器。

workflow名称	描述
从软件包导入配置	将基于文件的配置导入到插件配置。
从 XSD 导入类型定义	从 XSD 文件导入类型定义。
移除命名空间	移除命名空间。
移除关系	移除关系。
移除类型	移除类型。
更新命名空间	更新命名空间。
更新类型	更新类型。

使用 PowerShell 插件

20

PowerShell 插件工作流库中包含多种可用于管理 PowerShell 主机并运行自定义 PowerShell 操作的工作流。

您可以在 Orchestrator 客户端中使用清单视图来管理可用的 PowerShell 资源。您可以使用插件的脚本 API 来开发自定义工作流。

本章讨论了以下主题：

- [VMware vRealize Orchestrator PowerShell 插件简介](#)
- [配置 PowerShell 插件](#)
- [使用 PowerShell 插件清单](#)
- [运行 PowerShell 脚本](#)
- [生成操作](#)
- [在操作之间传递调用结果](#)
- [PowerCLI 与 PowerShell 插件集成](#)
- [示例工作流](#)
- [访问 PowerShell 插件 API](#)
- [使用 PowerShell 结果](#)
- [常规 PowerShell 任务脚本的示例](#)
- [故障排除](#)

VMware vRealize Orchestrator PowerShell 插件简介

PowerShell 插件允许 vRealize Orchestrator 和 Windows PowerShell 之间进行交互。

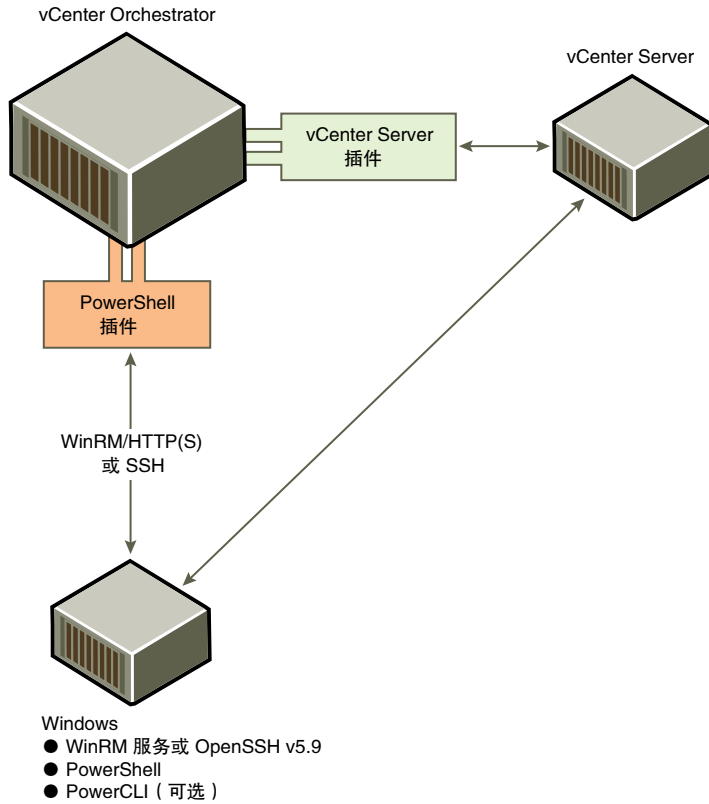
使用插件从 Orchestrator 操作和工作流中调用 PowerShell 脚本和 cmdlet 并处理结果。该插件包含一组标准工作流。您还可以创建实施插件 API 的自定义工作流。

PowerShell 插件组件

PowerShell 插件要依靠多个组件才能正常运行。

vRealize Orchestrator 和 WindowsPowerShell 为插件提供平台，而插件提供这些产品之间的交互。PowerShell 插件还可以与其他组件（例如 vCenter Server 和 vSphere PowerCLI）进行交互。

图 20-1. 组件关系



该插件可以通过 OpenSSH 和 WinRM 通信协议与 WindowsPowerShell 通信。请参见[配置 WinRM](#)。

您可以选择将 PowerShell 插件与 vSphere PowerCLI 和 vCenter Server 集成。请参见[PowerCLI 与 PowerShell 插件集成](#)。

注 您可以在本地主机上安装所有组件。如果在同一台计算机上安装了 vRealize Orchestrator 和 Windows PowerShell，PowerShell 插件的使用情况、功能和通信协议要求不会改变。

包含 PowerShell 插件的 vRealize Orchestrator 的角色

您必须使用 Orchestrator 配置界面来安装 PowerShell 插件。您可以使用 Orchestrator 客户端运行并创建工作流以及访问插件 API。

PowerShell 插件由 vRealize Orchestrator 提供支持。Orchestrator 是一个开发和过程自动化平台，提供可扩展工作流库来管理 VMware vCenter 基础架构和其他技术。

Orchestrator 可通过其开放插件架构与管理解决方案进行集成。PowerShell 是一个通过使用插件与 Orchestrator 集成的管理解决方案的示例。

使用插件与 Windows PowerShell 进行交互

您可以使用该插件运行 Orchestrator 工作流，从而与 Windows PowerShell 主机进行交互并执行任务（例如：调用 PowerShell 脚本）。

Windows PowerShell 是基于任务的命令行 shell 和脚本语言，专为系统管理而设计。

配置 WinRM

要在 PowerShell 插件和 Windows PowerShell 之间建立连接，您必须将 WinRM 配置为使用支持的通信协议之一。

PowerShell 插件支持 Windows Remote Management (WinRM) 2.0 作为管理协议。

下面是受支持的身份验证方法。

身份验证方法	详细信息
基本	需要用户名和密码的非安全身份验证机制。
Kerberos	使用票证验证客户端和服务端身份的安全身份验证协议。

注 PowerShell 插件不支持在 WinRM 委派用户凭据且不支持 CredSSP 身份验证方法。

基于 HTTP 的 WinRM

PowerShell 插件支持通过 HTTP 协议与 WinRM 主机进行通信。尽管 WinRM 对通信进行身份验证，但数据传输并未加密，并在网络上以纯文本形式发送。如果进行通信的计算机之间配置了 IPsec，您应使用 HTTP 协议。

要使用基本身份验证，必须在服务和客户端 WinRM 配置中将 `AllowUnencrypted` 属性设置为 `true`。有关 HTTP 配置的示例，请参见[将 WinRM 配置为使用 HTTP](#)。

基于 HTTPS 的 WinRM

PowerShell 插件支持通过 HTTPS 协议与 WinRM 主机进行通信。可以将 HTTPS 协议用作更安全的通信方法。

要使用 HTTPS 协议，必须生成一个用于服务器身份验证的证书，并在 WinRM 主机上安装该证书。有关 HTTPS 配置的示例，请参见[将 WinRM 配置为使用 HTTPS](#)。

将 WinRM 配置为使用 HTTP

您可以将 WinRM 主机配置为通过 HTTP 协议与 PowerShell 插件进行通信。

您必须通过在 WinRM 主机上运行命令来修改 WinRM 配置。您可以将同一台计算机用作 WinRM 服务和 WinRM 客户端。

重要事项 如果您在将 WinRM 配置为使用 HTTP 时跳过任何步骤，则可能不会添加主机，并且可能会在日志中显示错误消息，例如

```
Caused by: org.dom4j.DocumentException: Error on line -1 of document : Premature end of file.
Nested exception: Premature end of file.
at org.dom4j.io.SAXReader.read(SAXReader.java:482)
at org.dom4j.DocumentHelper.parseText(DocumentHelper.java:278)
at
com.xebialabs.overthere.cifs.winrm.connector.JdkHttpConnector.sendMessage(JdkHttpConnector.java:117)
```

步骤

- 1 运行以下命令以设置 WinRM 配置的默认值。

```
c:\> winrm quickconfig
```

- 2 （可选）运行以下命令以检查侦听器是否正在运行，并验证默认端口。

```
c:\> winrm e winrm/config/listener
```

HTTP 的默认端口为 5985，HTTPS 的默认端口为 5986。

- 3 在 WinRM 服务上启用基本身份验证。

- a 运行以下命令以检查是否允许基本身份验证。

```
c:\> winrm get winrm/config/service
```

- b 运行以下命令以启用基本身份验证。

```
c:\> winrm set winrm/config/service/auth @{Basic="true"}
```

- 4 运行以下命令以允许在 WinRM 服务上传输未加密数据。

```
c:\> winrm set winrm/config/service @{AllowUnencrypted="true"}
```

- 5 如果 WinRM 服务的通道绑定令牌强化级别设置为 **strict**，则将其值更改为 **relaxed**。

```
c:\> winrm set winrm/config/service/auth @{CbtHardeningLevel="relaxed"}
```

- 6 在 WinRM 客户端上启用基本身份验证。

- a 运行以下命令以检查是否允许基本身份验证。

```
c:\> winrm get winrm/config/client
```

- b 运行以下命令以启用基本身份验证。

```
c:\> winrm set winrm/config/client/auth @{Basic="true"}
```

- 7 运行以下命令以允许在 WinRM 客户端上传输未加密数据。

```
c:\> winrm set winrm/config/client @{AllowUnencrypted="true"}
```

- 8 如果 WinRM 主机是外部域，请运行以下命令以指定受信任的主机。

```
c:\> winrm set winrm/config/client @{TrustedHosts="host1, host2, host3"}
```

- 9 运行以下命令以测试与 WinRM 服务的连接。

```
c:\> winrm identify -r:http://winrm_server:5985 -auth:basic -u:user_name -
p:password -encoding:utf-8
```

将 WinRM 配置为使用 HTTPS

您可以将 WinRM 主机配置为通过 HTTPS 协议与 PowerShell 插件进行通信。

WinRM 主机需要证书才可以通过 HTTPS 协议进行通信。您可以获取证书或生成证书。例如，您可以使用属于 .NET Framework SDK 一部分的证书创建工具 (`makecert.exe`) 生成自签名证书。

前提条件

- 将 WinRM 配置为使用 HTTP 协议。有关详细信息，请参见[将 WinRM 配置为使用 HTTP](#)。
- 验证您可以在 WinRM 主机上访问 Microsoft 管理控制台 (`mmc.exe`)。

步骤

- 1 生成自签名证书。

以下命令行包含使用 `makecert.exe` 在 WinRM 主机上创建证书的示例语法。

```
makecert.exe -r -pe -n "CN=host_name-3,0=organization_name" -e mm/dd/yyyy -eku
1.3.6.1.5.5.7.3.1 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA
SChannel Cryptographic Provider" -sy 12 certificate_name.cer
```

- 2 通过使用 Microsoft 管理控制台添加生成的证书。

- a 运行 `mmc.exe`。
- b 选择文件 > 添加/删除管理单元。
- c 从可用管理单元列表中，选择证书，然后单击添加。
- d 选择计算机帐户并单击下一步。
- e 单击完成。
- f 验证在控制台根 > 证书(本地计算机) > 个人 > 证书和控制台根 > 证书(本地计算机) > 受信任的根证书颁发机构 > 证书中安装了证书。

如果受信任的根证书颁发机构和个人文件夹中未安装证书，则必须手动安装。

- 3 使用正确的指纹和主机名创建 HTTPS 侦听器。

以下命令行包含用于创建 HTTPS 侦听器的示例语法。

```
winrm create winrm/config/Listener?Address=*&Transport=HTTPS
@{Hostname="host_name";CertificateThumbprint="certificate_thumbprint"}
```

注 忽略证书指纹中的空格。

4 测试连接。

以下命令行包含用于测试连接的示例语法。

```
winrs -r:https://host_name:port_number -u:user_name -p:password hostname
```

配置 Kerberos 身份验证

在添加和管理 PowerShell 主机时可以使用 Kerberos 身份验证。

借助 Kerberos 身份验证，域用户可以通过 WinRM 在启用 PowerShell 的远程计算机上运行命令。

步骤

1 在 WinRM 服务上启用 Kerberos 身份验证。

- a 运行以下命令以检查是否允许进行 Kerberos 身份验证。

```
c:\> winrm get winrm/config/service
```

- b 请运行以下命令以启用 Kerberos 身份验证。

```
c:\> winrm set winrm/config/service/auth @{Kerberos="true"}
```

2 在 WinRM 客户端上启用 Kerberos 身份验证。

- a 运行以下命令以检查是否允许进行 Kerberos 身份验证。

```
c:\> winrm get winrm/config/client
```

- b 请运行以下命令以启用 Kerberos 身份验证。

```
c:\> winrm set winrm/config/client/auth @{Kerberos="true"}
```

3 运行以下命令以测试与 WinRM 服务的连接。

```
c:\> winrm identify -r:http://winrm_server:5985 -auth:Kerberos -u:user_name -
p:password -encoding:utf-8
```

4 创建 krb5.conf 文件并将其保存到以下位置。

操作系统	路径
Windows	C:\Program Files\Common Files\VMware\VMware vCenter Server - Java Components\lib\security\
Linux	用于外部 vRealize Orchestrator 的 /usr/java/jre-vmware/lib/security/。 用于 vRealize Automation 内置 vRealize Orchestrator 的 /etc/krb5.conf。

krb5.conf 文件具有以下结构：

```
[libdefaults]
default_realm = YOURDOMAIN.COM
```

```

udp_preference_limit = 1
[realms]
YOURDOMAIN.COM = {
  kdc = kdc.yourdomain.com
  default_domain = yourdomain.com
}
[domain_realm]
.yourdomain.com=YOURDOMAIN.COM
yourdomain.com=YOURDOMAIN.COM

```

krb5.conf 必须包含特定的配置参数及其值。

Kerberos 配置标记	详细信息
default_realm	客户端用于针对 Active Directory 服务器进行身份验证的默认 Kerberos 领域。 注 必须是大写字母。
kdc	可作为密钥分发中心 (KDC) 并签发 Kerberos 票证的域控制器。
default_domain	用于生成完全限定域名的默认域。 注 该标记面向 Kerberos 4 兼容性。

[注](#) Java Kerberos 配置默认使用 UDP 协议。要仅使用 TCP 协议，您必须指定值为 **1** 的 udp_preference_limit 参数。

[注](#) Kerberos 身份验证要求使用完全限定域名 (Fully Qualified Domain Name, FQDN) 主机地址。

重要事项 添加或修改 krb5.conf 文件时，必须重新启动 Orchestrator 服务器服务。

配置 PowerShell 插件

您必须使用 Orchestrator 客户端来配置 PowerShell 插件。

配置工作流

“配置”工作流类别包含可用于管理 PowerShell 主机的工作流。

在 Orchestrator 客户端的[工作流](#)视图中，可以从[库 > PowerShell > 配置](#)访问以下工作流。

工作流名称	描述
添加 PowerShell 主机	将 PowerShell 主机添加到插件清单。
移除 PowerShell 主机	移除插件清单中的 PowerShell 主机。
更新 PowerShell 主机	更新插件清单中指定的 PowerShell 主机。
验证 PowerShell 主机	验证指定 PowerShell 主机的配置。

添加 PowerShell 主机

添加 PowerShell 主机并通过运行工作流配置主机连接参数。您可以设置与远程或本地 PowerShell 主机的连接。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > PowerShell > 配置**并导航到“添加 PowerShell 主机”工作流。
- 4 右键单击添加 PowerShell 主机工作流，然后选择**启动工作流**。
- 5 在**名称**文本框中，输入主机的名称。
- 6 在**主机/IP** 文本框中，输入主机的地址。

注 Kerberos 身份验证要求使用完全限定域名 (Fully Qualified Domain Name, FQDN) 主机地址。

- 7 (可选) 在**端口**文本框中，输入主机的端口。
HTTP 协议使用端口 5985，HTTPS 协议使用端口 5986。
- 8 选择插件要连接的 PowerShell 主机类型。

- a 选择传输协议。

注 如果使用 HTTPS 传输协议，则远程 PowerShell 主机的证书会导入到 Orchestrator 密钥库。

- b 选择身份验证类型。

重要事项 如果要使用 Kerberos 身份验证，必须在 WinRM 服务上启用它。

- 9 选择插件连接到 PowerShell 主机时所用的会话模式类型。

选项	描述
共享会话	该插件使用共享凭据连接到远程主机。您必须提供共享会话的 PowerShell 主机凭据。
单用户会话	Orchestrator 客户端会检索已登录用户的凭据。您必须以 user@domain 的格式登录到 Orchestrator 以使用 单用户会话 模式。

- 10 从 **Shell 代码**页下拉菜单中，选择 PowerShell 使用的编码类型。
- 11 单击**提交**以运行工作流。

结果

工作流成功运行后，PowerShell 主机会显示在**清单**视图中。

使用 PowerShell 插件清单

PowerShell 插件会在**清单**视图中公开所连接的 PowerShell 主机中的所有对象。您可以使用**清单**视图添加授权元素或在 PowerShell 对象上运行工作流。

您可以在插件的清单内监控 PowerShell 主机及其管理单元和 cmdlet。每个远程主机都可以包含管理单元，并且每个管理单元中都可以包含 cmdlet。

要显示可供清单对象使用的工作流，请导航到**工具 > 用户首选项 > 清单**，然后选中在**清单中使用上下文菜单**复选框。启用此选项后，当您右键单击 Orchestrator 清单中的对象时，将显示可供该对象使用的所有工作流。

运行 PowerShell 脚本

您可以运行工作流以在 PowerShell 主机上调用外部或自定义脚本。

调用 PowerShell 脚本

您可以在插件清单中的一台主机上运行现有或自定义 PowerShell 脚本。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 PowerShell 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > PowerShell**并导航到“调用 PowerShell 脚本”工作流。
- 3 右键单击“调用 PowerShell 脚本”工作流，然后选择**启动工作流**。
- 4 选择要在其中运行脚本的 PowerShell 主机。
- 5 在**脚本**文本框中，输入或粘贴想要运行的 PowerShell 脚本。
- 6 单击**提交**以运行工作流。

调用外部脚本

您可以在插件清单中的一台主机上运行外部 PowerShell 脚本。

外部 PowerShell 脚本包含在 .ps1 文件中。您想要运行的 .ps1 文件必须存储于 PowerShell 主机中。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 PowerShell 主机建立了连接。
- 验证您是否拥有脚本可能引用的其他 .ps1 文件的访问权限。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > PowerShell** 并导航到“调用外部脚本”工作流。
- 3 右键单击“调用外部脚本”工作流，然后选择**启动工作流**。
- 4 选择要在其中运行脚本的 PowerShell 主机。
- 5 在**名称**文本框中，输入想要运行的外部 **.ps1** 脚本的文件名。

注 如果 **.ps1** 文件不在默认文件夹中，必须输入绝对文件路径。您可以使用系统环境变量来指定脚本路径。例如：**\$env:HOME\PATH\test1.ps1**。

- 6 在**参数**文本框中，输入脚本参数。

有效的语法与 PowerShell 控制台中所用语法相同。

- 7 单击**提交**以运行工作流。

生成操作

您可以运行工作流来生成基于 PowerShell 脚本或 PowerShell cmdlet 的操作。您可以将生成的操作作为构建块用于自定义工作流。

从 PowerShell 脚本生成操作

您可以运行工作流，通过所提供的 PowerShell 脚本生成操作。您还可以选择是否生成一个可以运行所生成操作的示例工作流。

您可以使用占位符对生成的操作的脚本进行自定义。对于每个占位符，工作流都会在生成的操作中创建 **string** 类型的对应操作参数。运行操作时，可以将实际值提供为操作参数以替代占位符。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 PowerShell 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > PowerShell > 生成**并导航到“通过 PowerShell 脚本生成操作”工作流。
- 3 右键单击“通过 PowerShell 脚本生成操作”工作流，然后选择**启动工作流**。
- 4 在**脚本**文本框中，输入或粘贴生成操作所依据的 PowerShell 脚本。

注 您可以使用 **{#ParamName#}** 作为用户输入的占位符。如果占位符为 **string** 类型，则您必须使用双引号将占位符的值传递到操作。

以下脚本示例显示了如何将生成的操作参数链接到脚本参数。

```
param($name={#ParamName#})
echo $name;
```

- 5 在**名称**文本框中，输入要生成的操作的名称。
- 6 选择要在其中生成操作的现有模块。
- 7 选择是否生成工作流。

选项	描述
是	<p>生成一个可以运行所生成操作的示例工作流。您应选择要在其中生成工作流的文件夹。</p> <p>注 所生成工作流的名称包含预定义的字符串 Invoke Script 以及所生成操作的名称。</p>
否	不会生成示例工作流。

- 8 单击**提交**以运行工作流。

后续步骤

您可以在自定义工作流中整合所生成的操作。

为 PowerShellcmdlet 生成操作

您可以运行工作流为 PowerShellcmdlet 和所提供的参数集生成操作。使用该操作，您可以在 Orchestrator 中使用 PowerShell 功能。您还可以选择是否生成一个运行所生成操作的示例工作流。

您可以将大量数据类型与 PowerShell 脚本引擎配合使用。您可以使用的数据类型包含原始类型（例如 Integer、Boolean、Char）、.NET 程序集中的类型或用户定义的类型。基于 PowerShellcmdlet 定义生成操作时，输入和输出 cmdlet 参数会以 Orchestrator 平台支持的类型表示。PowerShell 插件定义了类型映射。通常情况下，原始类型会映射到相应的 Orchestrator 类型，复杂类型会以 PowerShellRemotePSObject 对象的形式表示。

前提条件

- 验证您是否已经以管理员身份登录到 Orchestrator 客户端。
- 验证您是否从**清单**视图与 PowerShell 主机建立了连接。

步骤

- 1 单击 Orchestrator 客户端中的**工作流**视图。
- 2 在工作流层次结构列表中，展开**库 > PowerShell > 生成**并导航到“为 PowerShell cmdlet 生成操作”工作流。
- 3 右键单击“为 PowerShellcmdlet 生成操作”工作流，然后选择**启动工作流**。
- 4 选择在使用生成的操作时要运行的 PowerShellcmdlet。

5 为 cmdlet 选择参数集。

参数集定义值会显示在**参数集定义**文本框中。

注 您无法通过编辑**参数集定义**文本框中的字符串来修改参数集定义值。您可以查看字符串，获取有关该参数集包含的参数的信息。

6 在**名称**文本框中，输入要生成的操作的名称。

7 选择要在其中生成操作的现有模块。

8 选择是否生成工作流。

选项	描述
是	<p>生成一个可以运行所生成操作的示例工作流。您应选择要在其中生成工作流的文件夹。</p> <p>注 生成的工作流的名称包含预定义的字符串 <code>Execute Cmdlet</code> 以及所生成操作的名称。</p>
否	不会生成示例工作流。

9 单击**提交**以运行工作流。

后续步骤

您可以在自定义工作流中整合所生成的操作。

在操作之间传递调用结果

PowerShell 插件支持将结果作为参数在 PowerShell 脚本调用之间传递。若要正确传递结果，所有调用都必须位于同一会话中。

PowerCLI 与 PowerShell 插件集成

您可以将第三方管理单元中的可用功能（例如 VMware vSphere PowerCLI）与 PowerShell 插件结合使用。

若要使用第三方管理单元功能，该管理单元必须可用于 PowerShell 主机。若要在当前会话中加载管理单元，必须同时调用 `AddPsSnapin` 操作。使用 PowerCLI 时，必须将管理单元的名称设置为 `VMware.VimAutomation.Core`。

PowerShell 插件不会对第三方管理单元提供预生成的操作。您可以运行“为 PowerShellcmdlet 生成操作”工作流来生成适用于第三方管理单元的操作。请参见[为 PowerShellcmdlet 生成操作](#)。

`com.vmware.library.powershell.converter` 软件包中包含可将 `VC:<SomeObjectType>` 对象转换成 PowerCLI 相应对象的基本构建块。此功能可让 vCenter Server 插件中的工作流与 PowerShell 插件中的工作流进行交互，并在两个插件之间传送参数。

转换器 workflow

您可以使用转换器 workflow 类别中的示例 workflow 来测试 PowerShell 插件与 PowerCLI 之间的集成。若要测试集成，PowerCLI 必须安装在 PowerShell 主机上。

“转换器”示例 workflow 展示了插件中的转换功能。

注 PowerShell 插件并不支持 PowerCLI 和 vCenter Server 插件中的所有可用类型。不支持的类型会返回异常。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > PowerShell > 示例 > 转换器** 访问以下 workflow。

workflow 名称	描述
将 PSObject 转换为 vCO 对象	将 PowerShellRemotePSObject 转换为 VC:<SomeObjectType>。
将 PSObject 转换为 vCO 对象再转换为 PSObject	将 PowerShellRemotePSObject 转换为 VC:<SomeObjectType> 并再反向转换。
将 vCO 对象转换为 PSObject	将 VC:<SomeObjectType> 转换为 PowerShellRemotePSObject。

示例 workflow

“示例” workflow 类别包含可用于测试基本用例的 workflow。

在 Orchestrator 客户端的 workflow 视图中，可以从 **库 > PowerShell > 示例** 访问以下 workflow。

workflow 名称	描述
通过 API 调用脚本	演示如何通过可用的脚本 API 调用 PowerShell 脚本。
列出目录内容	列出 PowerShell 主机文件系统上的目录的内容。
管线执行示例	演示如何运行多个已安排在管道中的 cmdlet。
切换虚拟机状态	切换虚拟机的电源状况。

访问 PowerShell 插件 API

借助 Orchestrator API Explorer，您可用其搜索 PowerShell 插件 API 并查看相关文档，了解可用于脚本元素的 JavaScript 对象。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 通过 Orchestrator 客户端或通过 workflow 编辑器、策略编辑器和操作编辑器的 **运行脚本** 选项卡访问 API Explorer。
 - 要通过 Orchestrator 客户端访问 API Explorer，请在 Orchestrator 客户端工具栏中单击 **工具 > API Explorer**。
 - 要通过 workflow 编辑器、策略编辑器和操作编辑器的 **运行脚本** 选项卡访问 API Explorer，请单击左侧的 **搜索 API**。
- 3 若要展开 PowerShell 插件 API 对象的层次结构列表，请双击左侧窗格中的 **PowerShell** 模块。

后续步骤

可以从 API 元素复制代码并将代码粘贴到脚本编写框中。有关 API 脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

使用 PowerShell 结果

您可以使用来自 PowerShell 插件 API 的对象以使用 Windows PowerShell 返回的结果。

您可以使用 `PowerShellInvocationResult` 类的方法来检索所运行脚本的相关信息。

方法	描述
<code>getErrors()</code>	返回脚本调用期间由 PowerShell 引擎报告的错误列表。
<code>getInvocationState()</code>	脚本的状态。可能的值为 <code>Completed</code> 或 <code>Failed</code> 。
<code>getHostOutput()</code>	PowerShell 控制台上显示的脚本输出。
<code>getResults()</code>	PowerShell 引擎返回的对象。返回的对象类型为 <code>PowershellRemotePSObject</code> 。

`PowershellRemotePSObject` 是 PowerShell 引擎所返回对象的远程表现形式。`PowershellRemotePSObject` 包含 XML 序列化结果，可通过调用 `getXml()` 方法进行访问。

PowerShell 插件还提供一个可用于对 XML 结果进行包装的对象模型，从而更轻松地访问特定对象属性。`getRootObject()` 方法提供了对象模型的访问权限。通常情况下，`getRootObject()` 方法会使用以下规则将 PowerShell 类型映射到 Orchestrator 中的可用类型。

- 如果返回的对象是原始 PowerShell 类型，则对象会映射到相应的 Orchestrator 原始类型。
- 如果返回的对象是 `collection` 类型，则对象会表示为 `ArrayList`。
- 如果返回的对象是 `dictionary` 类型，则对象会表示为 `Hashtable`。
- 如果返回的对象是 `complex` 类型，则对象会表示为 `PSObject`。

常规 PowerShell 任务脚本的示例

您可以剪切、粘贴和编辑 JavaScript 示例来编写脚本，用于执行常规 PowerShell 任务。

有关脚本的更多信息，请参见《vRealize Orchestrator 开发人员手册》。

示例：通过 API 运行 PowerShell 脚本

您可以使用 JavaScript 通过插件 API 运行 PowerShell 脚本。

此示例脚本会执行以下操作。

- 将会话打开到 PowerShell 主机。
- 提供要运行的脚本。
- 检查调用结果。

- 关闭会话。

```
var sess;
try {
    //Open session to PowerShell host
    var sess = host.openSession()
    //Set executed script
    var result = sess.invokeScript('dir')

    //Check for errors
    if (result.invocationState == 'Failed'){
        throw "PowerShellInvocationError: Errors found while executing script \n" +
result.getErrors();
    }
    //Show result
    System.log( result.getHostOutput() );
} catch (ex){
    System.error (ex)
} finally {
    if (sess) {
        //Close session
        host.closeSession( sess.getSessionId() );
    }
}
```

示例： 使用结果

您可以使用 JavaScript 处理 PowerShell 脚本运行的结果。

此示例脚本会执行以下操作。

- 检查调用状态。
- 提取结果的值。
- 检查 RemotePSObject 类型。

```
var sess = host.openSession()
sess.addCommandFromString("dir " + directory)
var invResult = sess.invokePipeline();
//Show result
System.log( invResult.getHostOutput() );

//Check for errors
if (invResult.invocationState == 'Failed'){
    System.error(invResult.getErrors());
} else {
    //Get PowerShellRemotePSObject
    var psObject = invResult.getResults();
    var directories = psObject.getRootObject();

    var isList = directories instanceof Array
    if ( isList ){
        for (idx in directories){
            var item = directories[idx];
```

```

        if ( item instanceof('System.IO.FileInfo') ){//Check type of object
            System.log( item.getProperty('FullName') );//Extract value from result
        }
    } else {
        System.log( directories.getProperty('FullName') );//Extract value from result
    }
}

host.closeSession( sess.getSessionId());

```

示例： 使用自定义凭据连接

您可以使用 JavaScript 通过自定义凭据连接到 PowerShell 主机。

```

var sess;
try {
    sess = host.openSessionAs(userName, password);

    var invResult = sess.invokeScript('$env:username');

    //Check for errors
    if (invResult.invocationState == 'Failed'){
        System.error(invResult.getErrors());
    } else {
        //Show result
        System.log( invResult.getHostOutput() );
    }
} catch (ex){
    System.error (ex)
} finally {
    if (sess) {
        host.closeSession( sess.getSessionId());
    }
}

```

故障排除

如果在使用 PowerShell 插件时遇到问题且没有找到解决办法，您可以参考故障排除主题以了解或解决问题。

启用 Kerberos 事件日志记录

为了进行故障排除，您可能要启用密钥分发中心 (Key Distribution Center, KDC) 计算机上的 Kerberos 事件日志记录功能。

前提条件

备份 Windows 注册表。

步骤

- 1 登录到可充当密钥分发中心 (Key Distribution Center, KDC) 的域控制器。
- 2 以**管理员**身份运行注册表编辑器。
- 3 在注册表窗口中，展开 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters。
- 4 如果 **LogLevel** 注册表项值不存在，右键单击即可创建。
 - a 右键单击**参数**，选择**新建 > DWORD (32 位) 值**，然后输入 **LogLevel**。
 - b 选择**参数**并在右侧窗格中，双击 **LogLevel**，然后在**数值数据:** 文本框中输入 **1**。

Windows Server 2003 及更高版本无需重新启动，新设置即可生效。

结果

Kerberos 错误事件条目记录在系统 Windows 事件日志中。

后续步骤

要禁用 Kerberos 事件日志记录功能，请删除 **LogLevel** 注册表项值或将其数值数据更改为 **0**。

Kerberos 数据库中未找到服务器

添加了使用 Kerberos 身份验证的服务器后，服务器可能会因添加不当而未被发现。

问题

尝试连接到服务器时，在 Kerberos 数据库中未找到该服务器。

```
未提供有效凭据(机制级别: 未提供有效凭据(机制级别: Kerberos 数据库中未发现服务器 (7))) (No valid credentials provided (Mechanism level: No valid credentials provided (Mechanism level: Server not found in Kerberos database (7))))
```

原因

此错误可能由多个配置错误所致。

- PowerShell 主机不是域的一部分。
- 主机到领域的映射不正确。
- 未正确构建 PowerShell 主机的服务主体名称。

注 目标为 IP 地址时，Kerberos 身份验证不起作用。

解决方案

使用 Kerberos 身份验证添加 PowerShell 主机时，输入 DNS 或 NetBIOS 目标。

无法获取 Kerberos 票证

如果提供的凭据有误，插件将无法获取 Kerberos 票证。

问题

您无法将主机添加到插件清单并会显示以下错误消息。

```
预身份验证信息无效 (Pre-authentication information was invalid) (24)
```

原因

您提供的凭据有误。

解决方案

提供正确的凭据。

Kerberos 身份验证因时间设置不同导致失败

使用 Kerberos 配置的环境中的时间设置不一致可能会导致身份验证失败。

问题

尝试使用 Kerberos 进行主机初始身份验证或进行资源访问时失败，并显示以下错误消息。

```
Clock Skew (时钟偏差)
```

原因

如果环境中计算机的系统时间与域控制器或其他控制器的系统时间相差 5 分钟以上，则 Kerberos 身份验证会失败。

解决方案

同步环境中的系统时间。

Kerberos 身份验证会话模式失败

使用“共享会话”或“单用户会话”模式进行 Kerberos 身份验证时，添加 PowerShell 主机可能会失败。

问题

尝试将 PowerShell 主机添加到使用“共享会话”或“单用户会话”的插件清单时，工作流会失败并显示以下错误。

```
空领域名称 (601) - 未指定默认领域(动态脚本模块名称: addPowerShellHost#16) (Null realm name (601) - default realm not specified (Dynamic Script Module name : addPowerShellHost#16))
```

原因

默认领域未在 Kerberos 配置文件 `krb5.conf` 中指定，也没有作为用户名的一部分提供。

解决方案

在 Kerberos 配置文件中提供默认领域，或在 Kerberos 身份验证时在用户名中包含领域。

无法访问领域的密钥分发中心

如果 `krb5.conf` 文件中存在任何拼写错误，可能会导致主机添加失败。

问题

添加主机时，Kerberos 身份验证无法访问 *yourrealm* 的密钥分发中心 (KDC)。

```
无法获取领域 YOURREALM.COM 的 kdc (Cannot get kdc for realm YOURREALM.COM)
```

原因

`krb5.conf` 文件中的 `libdefaults` 和 `realms` 部分可能存在拼写错误。

解决方案

验证 `krb5.conf` 文件中的 `libdefaults` 和 `realms` 是否拼写无误。

无法找到默认领域

如果 Kerberos 配置文件的格式或编码不正确，需要进行 Kerberos 身份验证的 Orchestrator 工作流可能会失败。

问题

Kerberos 身份验证无法识别默认领域。

```
无法找到默认领域
```

原因

您上传到 vRealize Orchestrator Appliance 的 Kerberos 配置文件 `krb5.conf` 已在非 UNIX 操作系统上进行编辑。因此，格式和编码可能不正确。

解决方案

为了使 Orchestrator 设备读取 `krb5.conf` 文件，文件的格式必须是 UNIX，字符编码则必须为 ANSI 和 UTF-8。

使用 Multi-Node 插件

21

Multi-Node 插件工作流库中包含多种可用于分层编排、Orchestrator 实例管理以及 Orchestrator 活动扩展等的工作流。

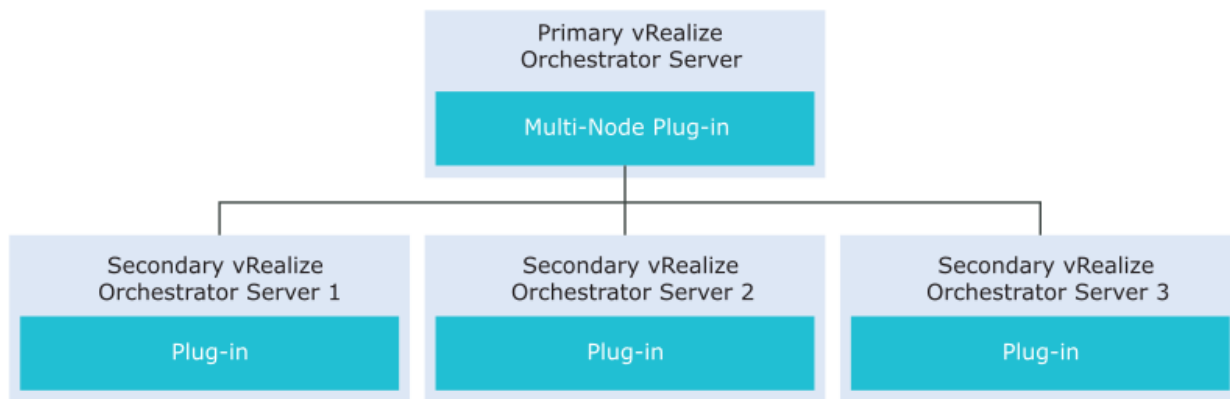
本章讨论了以下主题：

- [vRealize Orchestrator Multi-Node 插件简介](#)
- [配置 Multi-Node 插件](#)
- [使用代理工作流](#)
- [使用 Multi-Node 插件清单](#)
- [访问 Multi-Node 插件 API](#)
- [Multi-Node 插件用例](#)

vRealize Orchestrator Multi-Node 插件简介

Multi-Node 插件会在 vRealize Orchestrator 服务器之间创建主辅关系，从而扩展了软件包管理和工作流执行的范围。

图 21-1. Multi-Node 插件架构



该插件中包含一组可用于分层编排、vRealize Orchestrator 实例管理以及 vRealize Orchestrator 活动扩展等的标准工作流。

配置 Multi-Node 插件

您必须使用 Orchestrator 客户端来配置 Multi-Node 插件。

服务器配置工作流

“服务器配置”工作流类别包含可用于配置已连接 Orchestrator 服务器的工作流。

在 Orchestrator 客户端的工作流视图中，可以从**库 > Orchestrator > 服务器配置**访问以下工作流。

工作流名称	描述
添加 Orchestrator 服务器	将 Orchestrator 服务器添加到插件清单。
删除 Orchestrator 服务器	移除插件清单中的 Orchestrator 服务器并删除该服务器上所有已创建的代理。
更新 Orchestrator 服务器	通过更改详细信息对插件清单中的 Orchestrator 服务器进行更新。

添加 Orchestrator 服务器

您可以运行工作流建立与新 vRealize Orchestrator 服务器的连接。

前提条件

验证主要和辅助 Orchestrator 服务器是否为同一版本。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > Orchestrator > 服务器配置**并导航到“添加 Orchestrator 服务器”工作流。
- 4 右键单击“添加 Orchestrator 服务器”工作流，然后选择**启动工作流**。
- 5 提供新服务器的详细信息。
- 6 选择是否共享连接。

选项	说明
否	使用已登录用户的凭据连接到远程 Orchestrator 服务器。
是	所有用户可以使用相同的凭据访问远程 Orchestrator 服务器。提供共享连接的凭据。

- 7 单击**提交**以运行工作流。

使用代理工作流

您可以使用代理工作流管理本地 Orchestrator 服务器与远程 Orchestrator 服务器工作流之间的交互。

您可以使用 Multi-Node 插件生成可与远程工作流交互的本地工作流。这些本地工作流被称为代理工作流。代理工作流会采用 Multi-Node 插件清单中的输入参数。运行代理工作流时，会将参数转换为远程工作流所需的类型。远程工作流运行完成后，输出参数会转换回主 Orchestrator 服务器上的本地表示。

同步代理工作流

代理工作流的同步类型保留了 API 以及远程工作流的操作协定。

所有同步代理工作流的架构均相同，但包含的脚本不同。



同步代理工作流会在远程工作流完成运行并提供输出参数后再完成运行。

本地工作流在等待远程工作流的结果时不会消耗服务器资源。

在成功运行后，该代理工作流的输出参数会包含远程工作流令牌的本地表示。如果输出参数是简单类型（例如布尔值、数字、字符串及相似内容），则可以直接用于本地 Orchestrator 服务器上的其他工作流。

异步代理工作流

您可以使用异步代理工作流来优化远程工作流的运行。

所有异步代理工作流的架构均相同，但包含的脚本不同。



异步代理工作流会立即返回结果，即远程工作流令牌对象的本地包装程序。代理工作流使用该令牌检查运行的状态，并在远程工作流完成运行时检索输出参数。如果输出参数是简单类型（例如布尔值、数字、字符串及相似内容），则可以直接用于本地 Orchestrator 服务器上的其他工作流。

远程执行工作流

“远程执行”工作流类别包含可用于管理代理工作流的工作流。

远程执行标准工作流

在 Orchestrator 客户端的工作流视图中，可以从库 > Orchestrator > 远程执行访问用来创建代理工作流的工作流。

工作流名称	描述
创建多代理操作	创建多代理操作以在多台服务器上运行工作流。
创建代理工作流	创建您用来在远程 Orchestrator 服务器上启动工作流的代理工作流。
从文件夹创建代理工作流	为远程 Orchestrator 服务器上文件夹中的所有工作流创建代理工作流。

服务器代理

在 Orchestrator 客户端的工作流视图中，可以从库 > Orchestrator > 远程执行 > 服务器代理访问用来管理服务器代理的工作流。

工作流名称	描述
创建 Orchestrator 服务器的代理工作流	通过镜像远程服务器的结构，在本地 Orchestrator 服务器上创建代理工作流。
删除 Orchestrator 服务器的代理工作流	移除本地 Orchestrator 服务器的代理工作流，并删除所有生成的工作流。
刷新 Orchestrator 服务器的代理工作流	从远程服务器重新生成本地 Orchestrator 服务器的所有代理工作流。

使用 Multi-Node 插件清单

Multi-Node 插件会在**清单**视图中镜像所连接的 vRealize Orchestrator 服务器的所有清单。您可以使用**清单**视图添加授权元素或在远程 Orchestrator 服务器上运行工作流。

要显示可供清单对象使用的工作流，请导航到**工具 > 用户首选项 > 清单**，然后选中在**清单中使用上下文菜单**复选框。启用此选项后，当您右键单击 Orchestrator 清单中的对象时，将显示可供该对象使用的所有工作流。

单台远程服务器主要由系统对象和插件对象两部分组成。这两种对象都是可将远程对象变为本地可用类型的包装程序：

系统对象

系统对象位于称为**系统**的顶层组中，包含配置、软件包、工作流、操作和相关文件夹。远程系统对象都有各自的包装程序类型。

插件对象

插件对象会镜像远程 Orchestrator 服务器上附加的所有插件清单。所有远程插件对象都会包装成为一个本地类型 **VCO:RemotePluginObject**。

远程管理工作流

“远程管理”工作流类别包含可用于管理远程 Orchestrator 实例上软件包和工作流的工作流。

软件包

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > Orchestrator > 远程管理 > 软件包**访问用来管理远程软件包的工作流。

工作流名称	描述
删除软件包	删除远程 Orchestrator 服务器中的软件包及其内容。
按名称删除软件包	按名称删除远程 Orchestrator 服务器上的软件包及其内容。
部署本地服务器中的软件包	将本地 Orchestrator 服务器中的软件包部署到远程 Orchestrator 服务器上。
部署远程服务器中的软件包	将一个远程 Orchestrator 服务器中的软件包部署到一组远程 Orchestrator 服务器上。
部署本地服务器中的软件包	将本地 Orchestrator 服务器中的软件包部署到远程 Orchestrator 服务器上。

工作流

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > Orchestrator > 远程管理 > 工作流**访问用来管理远程工作流的工作流。

工作流名称	描述
删除远程工作流	删除远程 Orchestrator 服务器中的工作流。
删除所有已完成的工作流运行	删除远程工作流中所有已完成的工作流运行。
部署本地服务器中的工作流	将本地 Orchestrator 服务器中的工作流部署到一组远程 Orchestrator 服务器上。
部署远程服务器中的工作流	将远程 Orchestrator 服务器中的工作流部署到一组其他远程 Orchestrator 服务器上。

访问 Multi-Node 插件 API

Orchestrator 提供 API Explorer，您可用其搜索 Multi-Node 插件 API 并查看相关文档，了解可用于脚本元素的 JavaScript 对象。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 通过 Orchestrator 客户端或通过工作流编辑器、策略编辑器和操作编辑器的**运行脚本**选项卡访问 API Explorer。
 - 要通过 Orchestrator 客户端访问 API Explorer，请在 Orchestrator 客户端工具栏中单击**工具 > API Explorer**。
 - 要通过工作流编辑器、策略编辑器和操作编辑器的**运行脚本**选项卡访问 API Explorer，请单击左侧的**搜索 API**。
- 3 若要展开 Multi-Node 插件 API 对象的层次结构列表，请双击左侧窗格中的 **VCO** 模块。

后续步骤

可以从 API 元素复制代码并将代码粘贴到脚本编写框中。有关 API 脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。

Multi-Node 插件用例

Multi-Node 插件用例包含多种用户场景：将本地 Orchestrator 服务器的软件包导入到远程服务器中、使用多代理操作以及有关远程和代理工作流的维护信息。

创建多代理操作

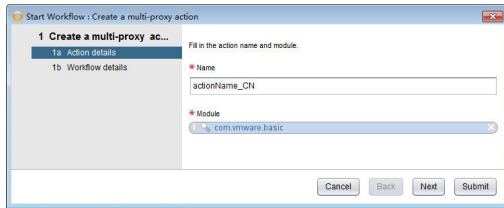
您可以运行“创建多代理操作”工作流在多台服务器上运行工作流。

您可以创建一个操作，以便稍后在远程 Orchestrator 服务器上运行工作流。

步骤

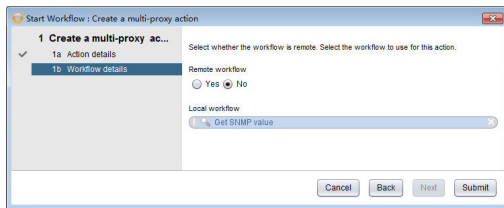
- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > Orchestrator > 远程执行**并导航到“创建多代理操作”工作流。
- 4 右键单击“创建多代理操作”工作流，然后选择**启动工作流**。
- 5 在**操作名称**文本框中，输入操作的名称。

操作名称必须仅包含字母数字字符，且不得包含空格。



创建新操作时，即使存在另一同名操作也会继续创建。

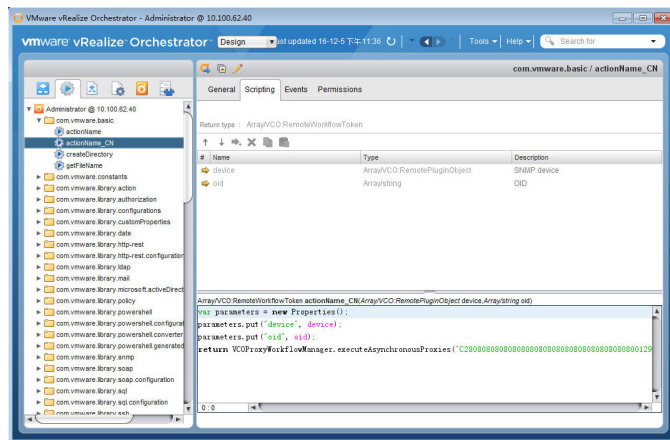
- 6 选择要在其中添加操作的模块。
- 7 选择工作流是本地还是远程。



- 8 选择要用于此操作的工作流。
- 9 单击提交以运行工作流。

结果

生成的操作接受与源工作流相同的参数，但如果有多项对象选择，则会将参数提升到数组。数组中的值会进行索引。



维护远程和代理工作流

如果远程和代理工作流发生更改，您可能需要更新代理或将其删除（如果不再需要）。出于维护目的，Multi-Node 插件提供了相关工作流，用于更新或删除代理和远程工作流信息。

在 Orchestrator 客户端的工作流视图中，可以从库 > **Orchestrator** > 远程执行 > 服务器代理访问用来管理代理工作流的工作流。

工作流名称	描述
刷新 Orchestrator 服务器的代理工作流	从远程服务器重新生成本地 Orchestrator 服务器的所有代理工作流。
删除 Orchestrator 服务器的代理工作流	移除本地 Orchestrator 服务器的代理工作流，并删除所有生成的工作流。

在 Orchestrator 客户端的**工作流**视图中，可以从**库 > Orchestrator > 远程管理 > 工作流**访问用来代理工作流进一步维护的工作流。

工作流名称	描述
删除所有已完成的工作流运行	删除远程工作流中所有已完成的工作流运行。
删除远程工作流	删除远程 Orchestrator 服务器中的工作流。
部署本地服务器中的工作流	将本地 Orchestrator 服务器中的工作流部署到一组远程 Orchestrator 服务器上。

部署本地服务器中的软件包

您可以运行工作流将本地 Orchestrator 服务器中的软件包部署到远程 Orchestrator 服务器上。

在此例中，您可以将本地服务器中的软件包部署到远程服务器阵列上。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > Orchestrator > 远程管理**并导航到“部署本地服务器中的软件包”工作流。
- 4 右键单击“部署本地服务器中的软件包”，然后选择**启动工作流**。
- 5 选择本地存储中要部署的软件包。
- 6 选择要在其中部署该软件包的远程服务器。
- 7 选择是否要覆盖远程服务器软件包。

选项	描述
是	远程服务器上的软件包会被替换，放弃已打包元素的版本。
否	对服务器和要部署软件包执行版本检查。在检查成功后即部署软件包。

- 8 单击**提交**以运行工作流。

结果

运行工作流后，状态信息会显示在日志视图和插件清单中。

使用 vCloud Suite API (vAPI) 插件

22

vCloud Suite API 插件可以使用任何 vCloud Suite API 提供程序公开的 API。vCloud Suite API 采用以服务为导向的架构，可通过 vCloud Suite 端点向 vCenter Server 发出请求，从而访问虚拟环境中的资源。

该插件包含一组标准工作流和示例工作流。您还可以创建采用插件 API 的自定义工作流，自动处理自己虚拟环境中的任务。有关 vCloud Suite API 的信息，请参见《VMware vCloud Suite SDKs 编程手册》。

本章讨论了以下主题：

- [配置 vCloud Suite API 插件](#)
- [访问 vCloud Suite API 插件 API](#)

配置 vCloud Suite API 插件

您可以运行插件中包含的配置工作流来配置 vCloud Suite API。

导入 vCloud Suite API 元模型

vCloud Suite API 插件会通过查询 vCloud Suite API 提供程序元数据服务来查找 vCloud Suite API 服务。未公开元数据服务的 vCloud Suite API 提供程序将不受支持。

您必须导入 vCloud Suite API 元模型并随后添加端点。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的工作流视图。
- 3 在工作流层次结构列表中，展开库 > VAPI 并导航到导入 vAPI 元模型工作流。
- 4 右键单击导入 vAPI 元模型工作流，然后选择启动工作流。
- 5 在 vAPI 端点 URL 文本框中，输入 vCloud Suite API 端点的 URL。

6 选择是否要使用安全协议连接：

选项	描述
否	导入 vCloud Suite API 元模型（不使用安全协议连接）。
是	若要使用安全协议连接导入 vCloud Suite API 元模型： <ul style="list-style-type: none"> a 选择是否要忽略证书警告并自动接受 vCloud Suite 端点。 b 提供向 vCloud Suite 端点进行身份验证时所需的用户凭据。

7 单击提交以运行工作流。**后续步骤**

[添加 vCloud Suite API 端点](#)

添加 vCloud Suite API 端点

添加 vCloud Suite API 端点。

前提条件

导入 vCloud Suite API 元模型。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 单击 Orchestrator 客户端中的**工作流**视图。
- 3 在工作流层次结构列表中，展开**库 > VAPI**并导航到**添加 vAPI 端点**工作流。
- 4 右键单击**添加 vAPI 端点**工作流，然后选择**启动工作流**。
- 5 在**vAPI 端点 URL**文本框中，输入 vCloud Suite API 端点的 URL。
- 6 选择是否要使用安全协议连接：

选项	描述
否	导入 vCloud Suite API 元模型（不使用安全协议连接）。
是	若要使用安全协议连接导入 vCloud Suite API 元模型： <ul style="list-style-type: none"> a 选择是否要忽略证书警告并自动接受 vCloud Suite 端点。 b 提供向 vCloud Suite 端点进行身份验证时所需的用户凭据。

7 单击提交以运行工作流。

访问 vCloud Suite API 插件 API

Orchestrator 提供 API Explorer，您可用其搜索 vCloud Suite API 插件 API 并查看相关文档，了解可用于脚本元素的 JavaScript 对象。

步骤

- 1 以管理员身份登录到 Orchestrator 客户端。
- 2 通过 Orchestrator 客户端或通过工作流编辑器、策略编辑器和操作编辑器的**运行脚本**选项卡访问 API Explorer。
 - 要通过 Orchestrator 客户端访问 API Explorer，请在 Orchestrator 客户端工具栏中单击**工具 > API Explorer**。
 - 要通过工作流编辑器、策略编辑器和操作编辑器的**运行脚本**选项卡访问 API Explorer，请单击左侧的**搜索 API**。
- 3 若要展开 vCloud Suite API 插件 API 对象的层次结构列表，请双击左侧窗格中的 **VAPI** 模块。

后续步骤

可以从 API 元素复制代码并将代码粘贴到脚本编写框中。有关 API 脚本编写的更多信息，请参见《使用 VMware vRealize Orchestrator 进行开发》。