安装和配置 VMware vRealize Orchestrator

2022 年 2 月

vRealize Orchestrator 8.7

vmware®

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

安装和配置 VMware vRealize Orchestrator

《安装和配置 VMware vRealize Orchestrator》提供有关安装和配置 VMware® vRealize Orchestrator 的信息和说明。

目标读者

本文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。

VMware vRealize Orchestrator 简介

1

VMware 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，可让您创建并运行可配置的自动化流程，用于管理 VMware 产品以及其他第三方技术。

vRealize Orchestrator 自动执行 VMware 及第三方应用程序的管理和运行任务，例如服务台、变更管理系统和 IT 资产管理系统。

本章讨论了以下主题：

Orchestrator 平台的主要功能

vRealize Orchestrator 由三个不同层组成：一个编排平台，用来提供编排工具所需的常用功能；一个插件基础架构，用来集成对子系统的控制，以及一个工作流库。vRealize Orchestrator 是一个开放式平台，可使用新插件和内容进行扩展，并可通过 REST API 集成到规模更大的架构中。

vRealize Orchestrator 包含若干有助于运行和管理工作流的重要功能。

持久性

生产级 PostgreSQL 数据库用于存储相关信息，例如进程、工作流状态和 vRealize Orchestrator 配置。

集中管理

vRealize Orchestrator 提供了集中管理各种进程的工具。基于应用程序服务器的平台拥有完整的版本历史记录，可在同一存储位置存储脚本和与进程相关的原语。这样，您就可以避免服务器上出现没有版本控制和适当更改控制的脚本。

检查点

工作流的每一步骤都会保存在数据库中，从而防止在服务器必须重启时丢失数据。此功能对于长时间运行的进程特别有用。

控制中心

控制中心是基于 Web 的门户，可通过一个界面对运行时操作、工作流监控以及工作流运行和系统资源之间的相关性进行集中管理，从而提高 vRealize Orchestrator 实例的管理效率。

版本控制

vRealize Orchestrator 平台的所有对象都有相关的版本历史记录。版本历史记录对于在向项目阶段或位置分发各种进程时的基本变更管理非常有用。

Git 集成

通过 vRealize Orchestrator Client，可以集成 Git 存储库，以进一步改进 vRealize Orchestrator 内容的版本和源控制。通过 Git，可以管理跨多个 vRealize Orchestrator 实例的工作流开发。请参见《使用 VMware vRealize Orchestrator 客户端》指南中的“结合使用 Git 和 vRealize Orchestrator 客户端”。

脚本引擎

Mozilla Rhino JavaScript 引擎提供了一种为 vRealize Orchestrator Client 平台创建构建块的方法。增强后的脚本引擎包含基本版本控制、变量类型检查、名称空间管理和异常处理。该引擎可用于以下构建块：

- 操作
- 工作流
- 策略

工作流引擎

工作流引擎可让您自动处理各种业务进程。它使用以下对象在工作流中创建分步式进程自动化处理：

- vRealize Orchestrator Client 提供的工作流和操作。
- 客户创建的自定义构建块。
- 插件向 vRealize Orchestrator Client 添加的对象。

用户、其他工作流、调度或策略可以启动工作流。

策略引擎

可以使用策略引擎监控并生成事件，以应对 vRealize Orchestrator Client 服务器或插件技术中不断变化的状况。策略可以汇总来自平台或插件的事件，帮助您处理任何集成技术中不断变化的条件。

vRealize Orchestrator Client

使用 vRealize Orchestrator Client 创建、运行、编辑和监控工作流。还可以使用 vRealize Orchestrator Client 来管理操作以及配置元素、策略元素和资源元素。请参见《使用 vRealize Orchestrator 客户端》。

开发和资源

通过 vRealize Orchestrator 登录页，可以快速访问资源，帮助您开发自己的插件，以便在 vRealize Orchestrator 中使用。您还会找到有关使用 vRealize Orchestrator REST API 将请求发送到 vRealize Orchestrator 服务器的信息。

安全

vRealize Orchestrator 提供以下高级安全功能：

- 公钥基础架构 (PKI)，用于对服务器之间导入和导出的内容签名并加密。
- 数字版权管理 (DRM)，用于控制对所导出内容进行查看、编辑和重新分发的方式。
- 传输层安全性 (TLS)，用于在 vRealize Orchestrator Client、vRealize Orchestrator 服务器和通过 HTTPS 访问 Web 前端之间提供加密通信。
- 高级访问权限管理，可对进程以及这些进程所操作的对象进行访问控制。

加密

vRealize Orchestrator 使用符合 FIPS 要求的高级加密标准 (AES) 和 256 位加密密钥对字符串进行加密。加密密钥随机生成，对群集以外的各种设备来说是唯一的。群集中的所有节点共享一个加密密钥。

vRealize Orchestrator 用户角色

vRealize Orchestrator 根据全局用户角色的具体职责提供不同的工具和界面。在 vRealize Orchestrator 中，可以有属于管理员组的具有全部权限的用户（**管理员**）、开发人员（**工作流设计人员**）、故障排除用户（**查看者**），以及具有有限访问权限的用户。

vRealize Orchestrator 用户角色在 vRealize Orchestrator Client 的**角色管理**菜单中进行管理。有关在 vRealize Orchestrator Client 中配置用户角色的详细信息，请参见《使用 VMware vRealize Orchestrator 客户端》指南中的“在 vRealize Orchestrator 客户端中分配角色”。

注 对于使用 vRealize Automation 进行身份验证或使用 vRealize Automation 许可证的 vRealize Orchestrator 部署，将通过 vRealize Automation 平台的身份和访问管理服务分配用户角色。请参见《使用 VMware vRealize Orchestrator 客户端》中的“在 vRealize Automation 中配置 vRealize Orchestrator 客户端角色”。

用户角色	说明
管理员	此用户对所有 vRealize Orchestrator 平台功能和内容（包括由特定组创建的内容）具有完全访问权限。管理员用户的主要职责包括： ■ 安装和配置 vRealize Orchestrator。 ■ 将用户添加到 vRealize Orchestrator Client，分配角色以及创建和删除组。请参见《使用 VMware vRealize Orchestrator 客户端》中的“在 vRealize Orchestrator 客户端中创建组”。 ■ 在开发人员的 vRealize Orchestrator 环境中为其创建与 Git 存储库的集成。请参见《使用 VMware vRealize Orchestrator 客户端》中的“配置与 Git 存储库的连接”。 ■ 通过工作流验证和调试工作流脚本等功能对 vRealize Orchestrator 环境进行故障排除。
查看者	此用户对所有 vRealize Orchestrator Client（包括组和组内容）具有只读访问权限。此用户可以查看但不能创建、编辑或运行内容，也不能导出工作流运行、工作流运行日志或软件包。查看者不受组权限限制。 注 只有通过 vRealize Automation 进行身份验证的 vRealize Orchestrator 实例才支持查看者角色。默认情况下，此角色不会映射到 vRealize Automation 角色，因此必须明确为用户分配此角色。

用户角色	说明
工作流设计人员	此用户可通过创建和编辑对象来扩展 vRealize Orchestrator 平台功能。工作流设计人员无权访问 vRealize Orchestrator Client 的管理功能和故障排除功能。工作流设计人员的主要职责包括： ■ 创建、编辑、运行和删除 vRealize Orchestrator 对象，如工作流、操作、策略和配置元素。 ■ 调度工作流运行。请参见《使用 VMware vRealize Orchestrator 客户端》中的“在 vRealize Orchestrator 客户端中调度工作流”。 ■ 将工作流开发人员创建的内容添加到其分配到的组。 ■ 将对 vRealize Orchestrator 内容清单的本地修改推送到连接的 Git 存储库。请参见《使用 VMware vRealize Orchestrator 客户端》中的“将修改推送到 Git 存储库”。
有限权限用户	无分配角色的用户仍可以登录到 vRealize Orchestrator Client，但对客户端功能和内容的访问受限。如果将这些用户分配到某个组，则此类用户可以查看和运行该组中包含的内容。

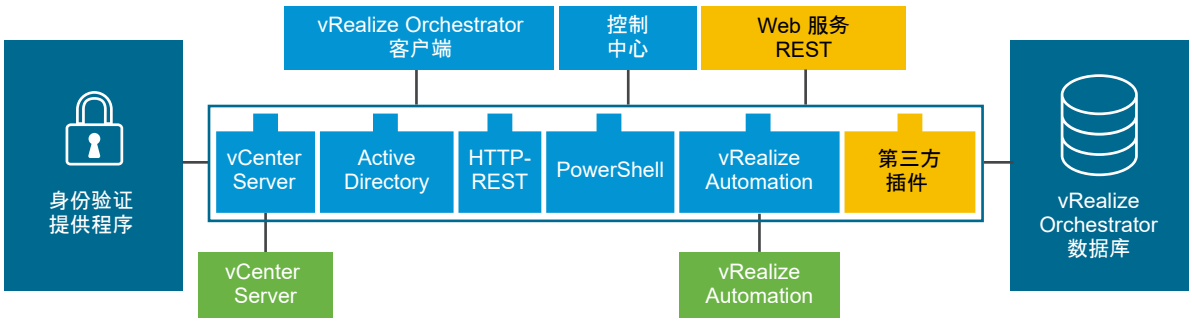
vRealize Orchestrator 架构

vRealize 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。vRealize 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。

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

vRealize 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。vRealize Orchestrator 连接到身份验证提供程序以管理用户帐户，并连接到预先配置的 PostgreSQL 数据库以存储来自其运行的工作流的信息。您可以通过 vRealize Orchestrator Client 或 Web 服务访问 vRealize Orchestrator 及其公开的对象，以及 vRealize Orchestrator 工作流。通过 vRealize Orchestrator Client 和控制中心来监控和配置 vRealize Orchestrator 工作流与服务。

图 1-1. VMware vRealize Orchestrator 架构



vRealize Orchestrator Plug-in

利用插件，您可以使用 vRealize Orchestrator 访问和控制外部技术与应用程序。通过在 vRealize Orchestrator 插件中公开外部技术，您可以将对象和功能并入到工作流中，从而可访问该外部技术的对象和功能。

通过插件可以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、目录服务和远程控制接口等。

vRealize Orchestrator 提供一组标准插件，可用于将 VMware vCenter Server API 和电子邮件功能等此类技术并入到工作流中。使用插件，可以自动交付新 IT 服务或调整现有基础架构和应用程序服务的功能。此外，还可以使用 vRealize Orchestrator 开放式插件架构开发用于访问其他应用程序的插件。

VMware 开发的 vRealize Orchestrator 插件采用 `.vmoapp` 文件形式分发。

有关 vRealize Orchestrator 插件的详细信息，请参见使用 VMware vRealize Orchestrator 插件。

有关第三方 vRealize Orchestrator 插件的详细信息，请访问 VMware Marketplace。

vRealize Orchestrator 系统要求 2

您的系统必须满足 vRealize Orchestrator 正常工作所需的技术要求。

有关受支持的 vCenter Server、vSphere Web Client、vRealize Automation 和其他 VMware 解决方案版本列表，请参见 VMware 产品互操作性列表。

本章讨论了以下主题：

- vRealize Orchestrator Appliance 组件
- vRealize Orchestrator Appliance 的硬件要求
- vRealize Orchestrator 可扩展性最大值
- vRealize Orchestrator 的网络要求
- vRealize Orchestrator 端口和端点
- vRealize Orchestrator 支持的浏览器
- 国际化和本地化支持级别

vRealize Orchestrator Appliance 组件

vRealize Orchestrator Appliance 是在容器中运行的基于 Photon 的虚拟设备。

vRealize Orchestrator Appliance 包含以下组件：

- 基础架构级别的 Kubernetes 层。
- 预配置的 PostgreSQL 数据库。
- 核心 vRealize Orchestrator 服务：服务器服务、控制中心服务和编排 UI 服务。

默认 vRealize Orchestrator Appliance 数据配置可用于生产环境。

注 要在生产环境中使用 vRealize Orchestrator Appliance，必须将 vRealize Orchestrator 服务器配置为通过 vRealize Automation 或 vSphere 进行身份验证。请参见配置独立 vRealize Orchestrator 服务器。

vRealize Orchestrator Appliance 的硬件要求

vRealize Orchestrator Appliance 是在容器中运行的基于 Photon 的预配置虚拟机。在部署设备前，验证系统是否满足最低硬件要求。

vRealize Orchestrator Appliance 具有以下硬件要求：

- 4 个 CPU
- 12 GB 内存
- 200 GB 硬盘

请勿降低默认内存大小，因为 vRealize Orchestrator 服务器至少需要 8 GB 可用内存。

vRealize Orchestrator 可扩展性最大值

可扩展性限制表概述了 vRealize Orchestrator 8.x 部署的建议最大值。

组件	扩展目标	更多信息
虚拟机	35,000	
vCenter Server 连接数	10	请参见 vCenter Server 设置
集群中的活动节点数	3	请参见配置 vRealize Orchestrator 集群
并行运行的工作流数	每个节点 300 个	请参见配置工作流运行属性
已排入队列的正在运行的工作流数	每个节点 10,000 个	
保留的工作流运行数	每个节点 100 个	
日志事件过期天数	15	

vRealize Orchestrator 的网络要求

每个 vRealize Orchestrator 节点都需要进行网络设置。

vRealize Orchestrator 的网络要求如下:

- 单个静态 IPv4 和网络地址
- 手动设置的可访问 DNS 服务器
- 手动设置的有效完全限定域名 (FQDN)，可通过 DNS 服务器进行正向和反向解析

注 不支持在安装后更改 IP 地址或主机名，这会导致设置损坏且无法恢复。

vRealize Orchestrator 端口和端点

vRealize Orchestrator Kubernetes 服务包括两个端点和多个主网络端口。

vRealize Orchestrator 网络端口

可以通过端口 443 访问 vRealize Orchestrator。443 端口由安装期间生成的自签名证书进行保护。使用外部负载均衡器时，必须将其设置为在端口 443 上进行均衡。

要查看所有 vRealize Orchestrator 端口，请访问端口和协议工具。

vRealize Orchestrator 端点

您可以在以下端点访问 vRealize Orchestrator Client 和控制中心服务。

服务	端点
vRealize Orchestrator 客户端	`https://your_orchestrator_FQDN/orchestration-ui`
控制中心	`https://your_orchestrator_FQDN/vco-controlcenter`

vRealize Orchestrator 支持的浏览器

确认您的浏览器支持 vRealize Orchestrator。

要访问 vRealize Orchestrator Client 和控制中心，必须使用以下浏览器之一：

- Microsoft Edge
- Mozilla Firefox
- Google Chrome

国际化和本地化支持级别

vRealize Orchestrator 控制中心和 vRealize Orchestrator Client 包括对非英语操作系统、非英语数据格式的支持，还包括对控制中心和客户端用户界面的多语言支持。

vRealize Orchestrator 控制中心和 vRealize Orchestrator Client 支持使用非英语操作系统、非英语输入和输出，还支持非英语格式的数据，如日期、时间和数字。

vRealize Orchestrator 和 vRealize Orchestrator Client 的用户界面已本地化为以下语言：

- 西班牙语
- 法语
- 德语
- 繁体中文
- 简体中文

- 韩语
- 日语
- 意大利语
- 荷兰语
- 巴西葡萄牙语
- 俄语

设置 vRealize Orchestrator 组件 3

下载并部署 vRealize Orchestrator Appliance 时，vRealize Orchestrator 服务器已经过预配置。在部署后，服务会自动启动。

要增加 vRealize Orchestrator 设置的可用性和可扩展性，请遵循以下准则：

- 安装并配置身份验证提供程序，然后将 vRealize Orchestrator 配置为与该提供程序配合使用。请参见配置独立 vRealize Orchestrator 服务器。
- 对于 vRealize Orchestrator 群集环境，安装和配置负载平衡服务器并将其配置为在 vRealize Orchestrator 服务器之间分发工作负载。

本章讨论了以下主题：

vCenter Server 设置

增加 vRealize Orchestrator 设置中的 vCenter Server 实例数会导致 vRealize Orchestrator 管理更多会话。活动会话过多可能会导致 vRealize Orchestrator 在出现 10 个以上 vCenter Server 连接时发生超时问题。

有关受支持的 vCenter Server 版本列表，请参见 VMware 产品互操作性列表。

注 如果您的网络具有足够的带宽和延迟，则可以在 vRealize Orchestrator 设置中的不同虚拟机上运行多个 vCenter Server 实例。如果要使用 LAN 改善 vRealize Orchestrator 和 vCenter Server 之间的通信，必须使用 100 MB 网线。

身份验证方法

要对用户权限进行身份验证和管理，vRealize Orchestrator 需要连接到 vRealize Automation 或 vSphere 服务器实例。

下载并部署 vRealize Orchestrator Appliance 时，必须使用 vRealize Automation 或 vSphere 身份验证配置服务器。请参见配置独立 vRealize Orchestrator 服务器。

注 仅 vRealize Automation 8.x 支持使用 vRealize Automation 执行 vRealize Orchestrator 8.x 身份验证。

安装 vRealize Orchestrator

4

vRealize Orchestrator 由一个服务器组件和一个客户端组件组成。

要使用 vRealize Orchestrator，必须部署 vRealize Orchestrator Appliance 并配置 vRealize Orchestrator 服务器。

可以使用 vRealize Orchestrator 控制中心更改默认的 vRealize Orchestrator 配置设置。

本章讨论了以下主题：

下载并部署 vRealize Orchestrator Appliance

必须先下载并部署 vRealize Orchestrator Appliance，然后才能访问 vRealize Orchestrator 内容和服务。

前提条件

- 确认您拥有正在运行的 vCenter Server 实例。vCenter Server 版本必须为 6.0 或更高版本。
- 确认要部署 vRealize Orchestrator Appliance 的主机满足最低硬件要求。请参见 vRealize Orchestrator Appliance 的硬件要求。
- 如果系统被隔离，无法访问 Internet，则您必须从 VMware 网站下载设备的 `.ova` 文件。

步骤

1. 以**管理员**身份登录到 vSphere Web Client。
2. 选择属于虚拟机的有效父对象的清单对象，例如数据中心、文件夹、集群、资源池或主机。
3. 选择**操作 > 部署 OVF 模板**。
4. 输入 `.ova` 文件的文件路径或 URL，然后单击**下一步**。
5. 输入 vRealize Orchestrator Appliance 的名称和位置，然后单击**下一步**。
6. 选择主机、集群、资源池或 vApp 作为要在其中运行设备的目标，然后单击**下一步**。
7. 查看部署详细信息，然后单击**下一步**。
8. 接受许可证协议中的条款，然后单击**下一步**。

9 选择要用于 vRealize Orchestrator Appliance 的存储格式。

格式	说明
厚置备延迟置零	以默认的厚格式创建虚拟磁盘。创建虚拟磁盘时为其分配所需的空间。创建时不会擦除物理设备上保留的任何数据（如有），但是以后从虚拟机首次执行写操作时会按需要将其置零。
厚置备快速置零	支持集群功能，例如 Fault Tolerance。创建虚拟磁盘时为其分配所需的空间。如果物理设备上保留了任何数据，则在创建虚拟磁盘时会将其置零。创建这种格式的磁盘所需的时间可能会比创建其他格式的磁盘长。
精简置备格式	节省硬盘空间。对于精简磁盘，可以根据输入的磁盘大小值置备磁盘所需的数据存储空间。精简磁盘开始时很小，只使用与初始操作所需大小完全相同的存储空间。

10 单击**下一步**。

11 配置网络设置，然后输入 **root** 密码。

在配置 vRealize Orchestrator Appliance 的网络设置时，必须使用 IPv4 协议。对于 DHCP 和静态网络配置，必须添加 vRealize Orchestrator Appliance 的完全限定域名 (FQDN)。

如果在已部署 vRealize Orchestrator Appliance 的 shell 中显示的主机名为 *photon-machine*，则不满足上述网络配置要求。

12 （可选） 为 vRealize Orchestrator Appliance 配置其他网络设置，例如启用 SSH 访问。

注 配置 Kubernetes 网络时，内部集群 CIDR 和内部服务 CIDR 的值必须至少允许 1024 个主机。由于此要求，网络掩码值必须等于或小于 22。大于 22 的网络掩码值无效。Kubernetes 网络属性必须具有以下默认值：

Kubernetes network property	Default value	Property description
Kubernetes 内部集群 CIDR	10.244.0.0/22	用于 Kubernetes 集群内运行的容器的 CIDR。
Kubernetes 内部服务 CIDR	10.244.4.0/22	用于 Kubernetes 集群内 Kubernetes 服务的 CIDR。

注 您也可以在部署后更改 Kubernetes CIDR 网络属性。请参见配置 vRealize Orchestrator Kubernetes CIDR。

13 （可选） 要为 vRealize Orchestrator Appliance 启用 FIPS 模式，请将 **FIPS 模式**设置为**严格**。

注 只有新的 vRealize Orchestrator 环境才支持启用 FIPS 140-2。如果要在环境中启用 FIPS 模式，必须在安装期间执行此操作。

14 单击**下一步**。

15 查看**即将完成**页面，然后单击**完成**。

结果

vRealize Orchestrator Appliance 即部署成功。

后续步骤

以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行，并确认您可以执行正向或反向 DNS 查找。

- 要执行正向 DNS 查找，请运行 `nslookup` *your_orchestrator_FQDN* 命令。该命令必须返回 vRealize Orchestrator Appliance 的 IP 地址。
- 要执行反向 DNS 查找，请运行 `nslookup` *your_orchestrator_IP* 命令。该命令必须返回 vRealize Orchestrator Appliance 的 FQDN。

注 如果未在部署过程中启用 SSH，也可以从 vSphere Web Client 中的虚拟机控制台执行 DNS 查找。

打开 vRealize Orchestrator Appliance 电源并打开主页

要使用独立 vRealize Orchestrator Appliance，必须先打开其电源。

步骤

1 以**管理员**身份登录到 vSphere Web Client。

2 右键单击 vRealize Orchestrator Appliance 并选择**电源 > 打开电源**。

3 在 Web 浏览器中，导航到在 OVA 部署期间配置的 vRealize Orchestrator Appliance 虚拟机的主机地址。

https://*your_orchestrator_FQDN*/vco。

启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问

您可以启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。

前提条件

- 下载并部署 vRealize Orchestrator Appliance。
- 确认 vRealize Orchestrator Appliance 已启动且正在运行。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 要启用 SSH 访问，请运行 `/usr/bin/toggle-ssh enable` 命令。

3 要禁用 SSH 访问，请运行 `/usr/bin/toggle-ssh disable` 命令。

初始配置

5

在开始使用 vRealize Orchestrator 自动执行任务并管理系统和应用程序之前，必须使用 vRealize Orchestrator 控制中心配置外部身份验证提供程序。还可以使用 vRealize Orchestrator 控制中心执行其他配置任务，例如管理许可证和证书信息、安装插件以及监控 vRealize Orchestrator 群集的状态。

本章讨论了以下主题：

配置独立 vRealize Orchestrator 服务器

尽管 vRealize Orchestrator Appliance 是基于 Photon 的预配置虚拟机，但必须先配置身份验证提供程序，然后才能访问 vRealize Orchestrator 控制中心和 vRealize Orchestrator Client 的完整功能。

使用 vRealize Automation 身份验证配置独立 vRealize Orchestrator 服务器

要准备 vRealize Orchestrator Appliance 以供使用，必须配置主机设置和身份验证提供程序。可以将 vRealize Orchestrator 配置为使用 vRealize Automation 进行身份验证。使用 vRealize Automation 8.x 执行 vRealize Automation 身份验证。。

前提条件

- 下载并部署最新版本的 vRealize Orchestrator Appliance。请参见下载并部署 vRealize Orchestrator Appliance。

- 安装并配置 vRealize Automation 8.x，并确认 vRealize Automation 服务器正在运行。请参见 vRealize Automation 文档。

  重要事项 vRealize Automation 身份验证提供程序的产品版本必须与 vRealize Orchestrator 部署的产品版本相匹配。例如，要对 vRealize Orchestrator 8.7 部署进行身份验证，必须使用 vRealize Automation 8.7 部署。

如果打算创建集群：

- 将负载均衡器设置为在多个 vRealize Orchestrator 实例中分发流量。请参见《VMware vRealize Orchestrator 8.x 负载均衡指南》。

步骤

1 访问控制中心以启动配置向导。

   a 导航到 https://*your_orchestrator_FQDN*/vco-controlcenter。

   b 以 **root** 用户身份使用您在 OVA 部署期间输入的密码登录。

2 配置身份验证提供程序。

   a 在**配置身份验证提供程序**页面中，从**身份验证模式**下拉菜单中选择 **vRealize Automation**。

   b 在**主机地址**文本框中，输入 vRealize Automation 主机地址并单击**连接**。

   vRealize Automation 主机地址的格式必须为 https://*your_vra_hostname*。

   c 单击**接受证书**。

   d 输入将在其下配置 vRealize Orchestrator 的 vRealize Automation 组织所有者的凭据。单击**注册**。

   e 单击**保存更改**。

   此时会显示一条消息，指明配置已成功保存。

结果

您已成功完成 vRealize Orchestrator 服务器配置。

后续步骤

- 在**许可**页面上验证 **CSP** 是否是已配置的许可证提供程序。
- 确认节点已在**验证配置**页面上正确配置。

  注 完成身份验证提供程序的配置之后，vRealize Orchestrator 服务器会在 2 分钟后自动重新启动。进行身份验证后立即验证配置可能会返回无效的配置状态。

使用 vSphere 身份验证配置独立 vRealize Orchestrator 服务器

您可以使用 vSphere 身份验证模式向 vCenter Single Sign-On 服务器注册 vRealize Orchestrator 服务器。vCenter Single Sign-On 身份验证仅适用于 6.0 及更高版本的 vCenter Server。

前提条件

- 下载并部署最新版本的 vRealize Orchestrator Appliance。请参见下载并部署 vRealize Orchestrator Appliance。
- 安装和配置 vCenter Server，并运行 vCenter Single Sign-On。请参见 vSphere 文档。

如果打算创建集群：

- 将负载均衡器设置为在多个 vRealize Orchestrator 实例中分发流量。请参见《VMware vRealize Orchestrator 8.x 负载均衡指南》。

步骤

1 访问控制中心以启动配置向导。

   a 导航到 https://*your_orchestrator_FQDN*/vco-controlcenter。

   b 以 **root** 用户身份使用您在 OVA 部署期间输入的密码登录。

2 配置身份验证提供程序。

   a 在**配置身份验证提供程序**页面中，从**身份验证模式**下拉菜单中选择 **vSphere**。

   b 在**主机地址**文本框中，输入包含 vCenter Single Sign-On 的 Platform Services Controller 实例的完全限定域名或 IP 地址，然后单击**连接**。

   注 如果您在负载均衡器后面使用外部 Platform Services Controller 或多个 Platform Services Controller 实例，则必须手动导入共享一个 vCenter Single Sign-On 域的所有 Platform Services Controller 的证书。

   注 要将其他 vSphere Client 与已配置的 vRealize Orchestrator 环境集成，您必须将 vSphere 配置为使用注册到 vRealize Orchestrator 的同一个 Platform Services Controller。对于高可用性 vRealize Orchestrator 环境，必须复制 vRealize Orchestrator 负载均衡器服务器后面的 PCS 实例。

   c 检查身份验证提供程序的证书信息，然后单击**接受证书**。

   d 输入 vCenter Single Sign-On 域的本地管理员帐户的凭据。单击**注册**。

   默认情况下，此帐户为 **`administrator@vsphere.local`**，默认租户的名称为 **`vsphere.local`**。

   e 在**管理员组**文本框中，输入管理员组的名称并单击**搜索**。

   例如 **`vsphere.local\vcoadmins`**

   f 选择要使用的管理组。

   g 单击**保存更改**。

   此时会显示一条消息，指明配置已成功保存。

结果

您已成功完成 vRealize Orchestrator 服务器配置。

后续步骤

- 确认 **CIS** 是**许可**页面上的已配置许可证提供程序。
- 确认节点已在**验证配置**页面上正确配置。

> 注 完成身份验证提供程序的配置之后，vRealize Orchestrator 服务器会在 2 分钟后自动重新启动。进行身份验证后立即验证配置可能会返回无效的配置状态。

通过许可证启用 vRealize Orchestrator 功能

对某些 vRealize Orchestrator 功能的访问基于您的 vRealize Orchestrator 部署所应用的许可证。

进行身份验证后，将根据身份验证提供程序为 vRealize Orchestrator 实例分配许可证。许可证可控制对以下 vRealize Orchestrator 功能的访问：

- Git 集成
- 角色管理
- 多语言支持（Python、Node.js 和 PowerShell）

可以从控制中心的**许可证**页面手动更改 vRealize Orchestrator 服务器的许可证。

注 无论许可证类型如何，可将该许可证应用到的 vRealize Orchestrator 部署数量没有任何限制。对于 vRealize Automation 许可证，不需要部署和配置 vRealize Automation 环境。

身份验证	许可证	Git 集成	角色管理	多语言支持
vSphere	vSphere vCloud Suite Standard	否	否	否
vSphere	vRealize Automation vRealize Suite Advanced 或 Enterprise vCloud Suite Advanced 或 Enterprise	是	是	是
vRealize Automation	vRealize Automation vRealize Suite Advanced 或 Enterprise vCloud Suite Advanced 或 Enterprise	是	从用于对 vRealize Orchestrator 进行身份验证的 vRealize Automation 实例管理角色。	是

注 vRealize Suite Standard 许可证不包括 vRealize Automation，因此不支持访问 vRealize Orchestrator 功能。

vRealize Orchestrator 数据库连接

vRealize Orchestrator 服务器需要一个数据库用于存储数据。

部署的 vRealize Orchestrator Appliance 包含一个预配置 PostgreSQL 数据库，由 vRealize Orchestrator 服务器用于存储数据。

用户无法访问 postgreSQL 数据库。

管理证书

证书针对特定服务器颁发，其中包含有关服务器公钥的信息，您可以使用证书对 vRealize Orchestrator 中创建的所有元素进行签名，保证其真实可靠。客户端收到来自您服务器的元素（通常为软件包）时，会验证您的身份并决定是否信任您的签名。

- 管理 vRealize Orchestrator 证书

  您可以从 vRealize Orchestrator 控制中心内的**证书**页面，或通过 vRealize Orchestrator Client 使用带 *ssl_trust_manager* 标记的工作流来管理 vRealize Orchestrator 证书。

管理 vRealize Orchestrator 证书

您可以从 vRealize Orchestrator 控制中心内的**证书**页面，或通过 vRealize Orchestrator Client 使用带 *ssl_trust_manager* 标记的工作流来管理 vRealize Orchestrator 证书。

将证书导入 Orchestrator 信任存储

vRealize Orchestrator 控制中心使用安全连接与 vCenter Server、关系型数据库管理系统 (RDBMS)、LDAP、Single Sign-On 和其他服务器进行通信。您可以从 URL 或 PEM 编码的文件导入所需 TLS 证书。每次要使用 TLS 连接到服务器实例时，必须从**证书**页面上的**受信任证书**选项卡导入相应的证书，并导入相应的 TLS 证书。

您可以从 URL 地址或 PEM 编码的文件将 TLS 证书加载到 vRealize Orchestrator 中。

选项	说明
从 URL 或代理 URL 导入	远程服务器的 URL： `https://your_server_IP_address` 或 `your_server_IP_address:port`
从文件导入	PEM 编码的证书文件的路径。 **注** 还可以通过在 vRealize Orchestrator Client 中运行**从文件中导入受信任证书**工作流来导入受信任证书。通过此工作流导入的文件必须使用 DER 编码。

有关导入证书的详细信息，请参见通过控制中心导入受信任证书。

软件包签名证书

从 vRealize Orchestrator 服务器导出的软件包已经过数字签名。导入、导出或生成用于软件包签名的新证书。软件包签名证书是一种数字身份标识形式，用来保证加密通信和 Orchestrator 软件包的签名。

vRealize Orchestrator Appliance 包含一个可根据设备的网络设置自动生成的软件包签名证书。如果设备的网络设置变更，则必须手动生成新的软件包签名证书。在生成新软件包签名证书后，未来导出的所有软件包都会使用新证书签名。

为 vRealize Orchestrator 生成自定义 TLS 证书

可以使用 vRealize Orchestrator Appliance 为环境生成新的 TLS 证书或设置现有的自定义证书。

vRealize Orchestrator Appliance 包含一个基于设备的网络设置自动生成的受信任层安全性 (TLS) 证书。如果设备的网络设置发生更改，则必须手动生成新证书。您可以创建证书链以确保通信加密，并为软件包提供签名。但是，收件人无法确定该自签名软件包是由您的服务器颁发还是由假冒您的第三方所颁发。要证明服务器的身份信息，请使用证书颁发机构 (CA) 签名的证书。

vRealize Orchestrator 生成在您的环境中唯一的服务器证书。私钥则存储在 vRealize Orchestrator 数据库的 `vmo_keystore` 表中。

注 要将 vRealize Orchestrator Appliance 配置为使用现有的自定义 TLS 证书，请参见为 vRealize Orchestrator 设置自定义 TLS 证书。

前提条件

确认已为 vRealize Orchestrator Appliance 启用 SSH 访问。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。

步骤

1 通过 SSH 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 运行 `vracli certificate ingress --generate auto --set stdin` 命令。

3 要将自定义证书应用于 vRealize Orchestrator Appliance，请运行部署脚本。

a 导航到 `/opt/scripts/` 目录。

```
cd /opt/scripts/
```

b 运行 `./deploy.sh` 脚本。

重要事项 请勿中断部署脚本。脚本运行完成后，您会收到以下消息：

```
Prelude has been deployed successfully. To access, go to your_orchestrator_address
```

后续步骤

要确认应用了新证书链，请运行 `vracli certificate ingress --list` 命令。

为 vRealize Orchestrator 设置自定义 TLS 证书

可以为 vRealize Orchestrator Appliance 设置自定义 TLS 证书。

vRealize Orchestrator Appliance 包含一个基于设备的网络设置自动生成的受信任层安全性 (TLS) 证书。

可以将 vRealize Orchestrator Appliance 配置为使用现有的自定义 TLS 证书。可以通过将相关的 PEM 文件从本地计算机导入到 vRealize Orchestrator Appliance 来设置证书。也可以通过将证书链直接复制到 vRealize Orchestrator Appliance 来设置自定义 TLS 证书。这两个过程都要求运行 `./deploy.sh` 脚本，然后才能在 vRealize Orchestrator 部署中使用新的 TLS 证书。

有关生成新的自定义 TLS 证书的信息，请参见为 vRealize Orchestrator 生成自定义 TLS 证书。

前提条件

- 确认已为 vRealize Orchestrator Appliance 启用 SSH 访问。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。
- 确认包含 TLS 证书的 PEM 文件按设定的顺序包含以下组件：
  a 证书的私钥。
  b 主证书。
  c 一个或多个证书颁发机构 (CA) 中间证书（如果适用）。
  d 根 CA 证书。

  例如，TLS 证书可以具有以下结构：

```
 -----BEGIN RSA PRIVATE KEY-----
<Private Key>
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
<Primary TLS certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Intermediate certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA certificate>
-----END CERTIFICATE-----
```

步骤

1 通过将 PEM 文件导入到 vRealize Orchestrator Appliance 来设置证书。

a 通过从 SSH shell 运行安全复制 (SCP) 命令，从本地计算机导入证书 PEM。

对于 Linux，可以使用终端 SCP 命令：

```
scp ~/PEM_local_filepath/your_cert_file.PEM root@orchestrator_FQDN_or_IP:/
PEM_orchestrator_filepath/your_cert_file.PEM
```

对于 Windows，可以使用 PuTTY 客户端 PSCP 命令：

```
pscp C:\PEM_local_filepath\your_cert_file.PEM root@<orchestrator_FQDN_or_IP>:/
PEM_orchestrator_filepath/your_cert_file.PEM
```

b 通过 SSH 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

c 运行 `vracli certificate ingress --set` *your_cert_file.PEM* 命令。

2 （可选） 通过将证书链直接复制到设备来设置证书。

a 通过 SSH 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

b 运行 `vracli certificate ingress --set stdin` 命令。

c 复制并粘贴证书链，然后按 Ctrl+D。

3 要应用新的 TLS 证书，请运行部署脚本。

a 导航到 `/opt/scripts/` 目录。

```
cd /opt/scripts/
```

b 运行 `./deploy.sh` 脚本。

重要事项 请勿中断部署脚本。脚本运行完成后，您会收到以下消息：

```
Prelude has been deployed successfully.To access, go to https://your_orchestrator_FQDN
```

结果

您已为 vRealize Orchestrator Appliance 设置自定义 TLS 证书。

后续步骤

要确认应用了新证书链，请运行 `vracli certificate ingress --list` 命令。

通过控制中心导入受信任证书

为了能与其他服务器安全地进行通信，vRealize Orchestrator 服务器必须能够验证其身份。为此，您可能需要将远程实体的 TLS 证书导入到 vRealize Orchestrator 信任存储区。要信任某个证书，您可以通过建立到特定 URL 的连接或直接将其作为 PEM 编码文件将该证书导入到信任存储区。

步骤

1 以 **root** 用户身份登录控制中心。

2 转到**证书**页面。

3 选择**受信任证书**，然后单击**导入**。

4 要从文件导入证书，请选择**从 PEM 编码文件导入**。

5 定位到证书文件，然后单击**导入**。

6 要从 URL 地址导入证书，请选择**从 URL 导入**。

7 输入存储证书的 URL 地址，然后单击**导入**。

结果

您已成功将远程服务器证书导入到 vRealize Orchestrator 信任存储区。

配置 vRealize Orchestrator Plug-In

可在 vRealize Orchestrator Appliance 中访问预安装的默认插件库。默认 vRealize Orchestrator Plug-In 在 vRealize Orchestrator 客户端中配置了插件特定的工作流运行。

默认 vRealize Orchestrator 插件随配置工作流一起提供。可从 vRealize Orchestrator 客户端运行这些工作流以注册端点进行管理。

配置工作流具有 *configuration* 标记。例如，要访问用于管理 AMQP 代理和订阅的工作流，请在工作流库的搜索文本框中输入 *AMQP* 和 *Configuration* 标记。

管理 vRealize Orchestrator 插件

在 vRealize Orchestrator 控制中心的**管理插件**页面上，可以查看在 vRealize Orchestrator 中安装的所有插件列表，并可执行基本管理操作。

安装或升级插件

使用 vRealize Orchestrator 插件，vRealize Orchestrator 服务器可以与其他软件产品进行集成。vRealize Orchestrator 附带一组预安装的默认插件。您可以通过安装自定义插件进一步扩展 vRealize Orchestrator 平台的功能。

可以从 vRealize Orchestrator 的**管理插件**页面安装或升级插件。可以使用的文件扩展名是 `.vmoapp`。

有关安装或升级 vRealize Orchestrator 插件的详细信息，请参见安装或更新 vRealize Orchestrator 插件。

更改插件日志记录级别

您只能更改特定插件的日志记录级别，而不能更改 vRealize Orchestrator 的日志记录级别。

禁用插件

可以通过取消选中插件名称旁边的**启用插件**选项来禁用插件。

此操作不会移除插件文件。有关在 vRealize Orchestrator 中卸载插件的详细信息，请参见删除插件。

安装或更新 vRealize Orchestrator 插件

可以在 vRealize Orchestrator 控制中心内安装或更新第三方插件。

前提条件

下载插件的 *.dar* 或 *.vmoapp* 文件。

注 vRealize Orchestrator 插件的首选文件格式为 *.vmoapp*。

步骤

1 以 **root** 用户身份登录到控制中心。

2 选择**管理插件**页面。

3 单击**浏览**，然后选择要安装或更新的插件的 *.dar* 或 *.vmoapp* 文件。

4 单击**上载**。

5 查看插件信息，（如果适用）接受最终用户许可协议，并单击**安装**。

将安装或更新插件，并重新启动 vRealize Orchestrator 服务器服务。

后续步骤

在**管理插件**页面上验证是否列出正确的插件信息。

删除插件

可以通过控制中心从 vRealize Orchestrator Appliance 中删除第三方插件。

注 从 vRealize Orchestrator 8.0 开始，不再需要手动从 vRealize Orchestrator Client 中删除插件软件包。

步骤

1 以 **root** 用户身份登录控制中心。

2 选择**管理插件**。

3 找到要删除的插件并单击删除图标 ( )。

4 确认要删除插件，然后单击**删除**。

结果

您即从 vRealize Orchestrator Appliance 中删除了该插件。

vRealize Orchestrator 高可用性

要提高 vRealize Orchestrator 服务的可用性，请在共享一个数据库的集群中启动多个 vRealize Orchestrator 服务器实例。在配置为作为集群的一部分运行前，vRealize Orchestrator 始终作为单个实例运行。

具有相同服务器配置和插件配置的多个 vRealize Orchestrator 服务器实例在一个集群中运行，并且共享一个数据库。

所有 vRealize Orchestrator 服务器实例可通过交换检测信号互相通信。每个检测信号都是一个时间戳，节点会按一定的时间间隔将这些时间戳写入到集群的共享数据库中。网络问题、数据库服务器未响应或过载都可能导致 vRealize Orchestrator 集群节点停止响应。如果活动 vRealize Orchestrator 服务器实例未能在故障切换超时时间段内发送检测信号，则会被认为未响应。故障切换超时时间等于检测信号间隔值乘以故障切换检测信号数量。可以据此来判定不可靠的节点，并可根据可用的资源和生产负载自定义该值。

vRealize Orchestrator 节点在断开与数据库的连接时会进入待机模式，并将此模式一直保持到数据库连接恢复为止。集群中的其他节点会接管活动的作业，恢复所有中断的工作流，完成之前未完成的项目，例如可编辑脚本任务或工作流调用等。

您可以从 vRealize Orchestrator Client 仪表板的**系统**选项卡监控 vRealize Orchestrator 集群的状态。要配置集群检测信号、故障切换检测信号数量以及活动节点数，请导航到 vRealize Orchestrator 控制中心的 **Orchestrator 集群管理**页面。

vRealize Orchestrator 可扩展性最大值

可扩展性限制表概述了 vRealize Orchestrator 8.x 部署的建议最大值。

组件	扩展目标	更多信息
虚拟机	35,000	
vCenter Server 连接数	10	请参见 vCenter Server 设置
集群中的活动节点数	3	请参见配置 vRealize Orchestrator 集群
并行运行的工作流数	每个节点 300 个	请参见配置工作流运行属性
已排入队列的正在运行的工作流数	每个节点 10,000 个	
保留的工作流运行数	每个节点 100 个	
日志事件过期天数	15	

配置 vRealize Orchestrator 集群

通过部署三个节点并将其连接为一个集群，可以将新的 vRealize Orchestrator 部署配置为在高可用性下运行。

vRealize Orchestrator 集群由三个共享一个公用 PostgreSQL 数据库的 vRealize Orchestrator 实例组成。已配置 vRealize Orchestrator 集群的数据库只能在异步模式下运行。

要创建 vRealize Orchestrator 集群，必须选择一个 vRealize Orchestrator 实例作为集群的主节点。配置主节点后，将辅助节点连接到该节点。

创建的 vRealize Orchestrator 集群已预配置了自动故障切换。

注 如果自动故障切换失败，可能会导致数据库数据丢失。

前提条件

- 下载并部署三个独立的 vRealize Orchestrator 实例。请参见下载并部署 vRealize Orchestrator Appliance。

  注 可用于创建集群 vRealize Orchestrator 环境的节点数建议为三个。

- 确认已为所有 vRealize Orchestrator 节点启用 SSH 访问。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。
- 配置负载均衡器服务器。请参见《VMware vRealize Orchestrator 8.x 负载均衡指南》。

步骤

1 配置主节点。

   a 通过 SSH 以 **root** 用户身份登录到主节点的 vRealize Orchestrator Appliance 命令行。

   b 要配置集群负载均衡器服务器，请运行 `vracli load-balancer set` *load_balancer_FQDN* 命令。

   c 登录到主节点的控制中心，然后选择**主机设置**。

   d 单击**更改**，然后设置已连接负载均衡器服务器的主机地址。

   e 配置身份验证提供程序。请参见配置独立 vRealize Orchestrator 服务器。

2 将辅助节点连接到主节点。

   a 通过 SSH 以 **root** 用户身份登录到辅助节点的 vRealize Orchestrator Appliance 命令行。

   b 要将辅助节点连接到主节点，请运行 `vracli cluster join` *primary_node_hostname_or_IP* 命令。

   c 输入主节点的 root 密码。

   d 对其他辅助节点重复此过程。

3 （可选） 如果主节点使用自定义证书，则必须在设备中设置该证书或生成新证书。请参见为 vRealize Orchestrator 生成自定义 TLS 证书。

   注 包含证书链的文件必须采用 PEM 编码。

4 完成集群部署。

a 通过 SSH 以 **root** 用户身份登录到主节点的 vRealize Orchestrator Appliance 命令行。

b 要确认所有节点都处于就绪状态，请运行 `kubectl -n prelude get nodes` 命令。

c 运行 `/opt/scripts/deploy.sh` 脚本并等待部署完成。

结果

您已经创建了 vRealize Orchestrator 集群。创建集群后，只能从负载均衡器服务器的 FQDN 地址访问 vRealize Orchestrator 环境。

注 由于只能使用负载均衡器的 root 密码访问集群的控制中心，因此，如果集群节点具有不同的 root 密码，则无法编辑该集群节点的配置。要编辑此节点的配置，请将其从负载均衡器中移除，在控制中心中编辑配置，然后将该节点重新添加到负载均衡器。

后续步骤

要监控 vRealize Orchestrator 集群的状态，请登录到 vRealize Orchestrator Client 并导航到仪表板的**系统**选项卡。请参见监控 vRealize Orchestrator 集群。

移除 vRealize Orchestrator 群集节点

可以删除 vRealize Orchestrator，以减少群集容量。

从 vRealize Orchestrator 群集中移除节点后，该节点将无法再正常工作。如果要再次使用该节点，必须将其 vRealize Orchestrator Appliance 从 vCenter Server 中删除，然后重新部署。请参见下载并部署 vRealize Orchestrator Appliance。

前提条件

创建 vRealize Orchestrator 群集。请参见配置 vRealize Orchestrator 集群。

步骤

1 以 **root** 用户身份登录到要移除的节点的 vRealize Orchestrator Appliance 命令行。

2 要从 vRealize Orchestrator 中移除节点，请运行 `vracli cluster leave` 命令。

3 以 **root** 用户身份登录到其余节点之一的 vRealize Orchestrator Appliance 命令行。

4 运行 `kubectl -n prelude get nodes` 命令并确认移除的节点不再属于群集。

扩大独立 vRealize Orchestrator 部署

可以通过扩大配置的 vRealize Orchestrator 部署提高其可用性和可扩展性。

前提条件

- 下载、部署并配置 vRealize Orchestrator 实例。请参见下载并部署 vRealize Orchestrator Appliance 和配置独立 vRealize Orchestrator 服务器。

- 下载并部署另外两个 vRealize Orchestrator 实例。请参见下载并部署 vRealize Orchestrator Appliance。
- 配置负载均衡器服务器。请参见《VMware vRealize Orchestrator 8.x 负载均衡指南》。

步骤

1 配置主节点。

a 以 **root** 用户身份登录到已配置 vRealize Orchestrator 部署的控制中心。

b 选择**配置身份验证提供程序**，然后取消注册身份验证提供程序。

c 选择**主机设置**，然后输入负载均衡器服务器的主机名。

d 选择**配置身份验证提供程序**，然后重新注册身份验证提供程序。

e 以 **root** 用户身份登录到已配置实例的 vRealize Orchestrator Appliance 命令行。

f 要停止 vRealize Orchestrator 实例的所有服务，请运行 `/opt/scripts/deploy.sh --onlyClean` 命令。

g 要设置负载均衡器，请运行 `vracli load-balancer set` *load_balancer_FQDN*。

h （可选） 如果您的 vRealize Orchestrator 实例使用自定义证书，请运行 `vracli certificate ingress --set` *your_cert_file.pem* 命令。

注 包含证书链的文件必须采用 PEM 编码。

2 将辅助节点连接到配置的实例。

a 以 **root** 用户身份登录到辅助节点的 vRealize Orchestrator Appliance 命令行。

b 要将辅助节点连接到配置的实例，请运行 `vracli cluster join` *primary_node_hostname_or_IP* 命令。

c 对另一个辅助节点重复此操作。

3 完成扩大过程。

a 以 **root** 用户身份登录到已配置实例的 vRealize Orchestrator Appliance 命令行。

b 运行 `/opt/scripts/deploy.sh` 并等待脚本完成。

结果

您已扩大 vRealize Orchestrator 部署。

监控 vRealize Orchestrator 集群

您可以通过 vRealize Orchestrator Client 仪表板的**系统**选项卡监控现有的 vRealize Orchestrator 集群。

建议的方法是通过 vRealize Orchestrator Client 仪表板的**系统**选项卡来监控 vRealize Orchestrator 实例的配置同步状态。

注 如果无法访问 vRealize Orchestrator Client 仪表板，则还可以通过从 vRealize Orchestrator Appliance 命令行运行 `kubectl get pods -n prelude` 命令来监控 vRealize Orchestrator 实例的状态。

配置同步状态	说明
正在运行	vRealize Orchestrator 服务可用并可以接受请求。
待机	vRealize Orchestrator 服务无法处理请求，因为： ■ 此节点是高可用性 (HA) 集群的一部分，在主节点未发生故障的情况下将一直处于待机模式。 ■ 服务无法验证配置必备条件，如与数据库的有效连接、身份验证提供程序和 vRealize Orchestrator 实例许可证。
无法检索服务的运行状况	无法与 vRealize Orchestrator 服务器服务通信，因为该服务已停止或存在网络问题。
等待重启	控制中心检测到配置更改，然后 vRealize Orchestrator 服务器将自动重启。

配置客户体验改善计划

如果选择参加客户体验改善计划 (CEIP)，VMware 会匿名收集某些信息，帮助提高 VMware 产品和服务的质量、可靠性和功能。

VMware 接收的信息类别

客户体验提升计划 (CEIP) 将向 Vmware 提供可帮助 Vmware 改善其产品和服务以及修复问题的信息。

有关通过 CEIP 收集的数据以及 VMware 使用该数据的用途的详情，请参见“信任与保证中心”的规定：http://www.vmware.com/trustvmware/ceip.html。要加入或退出此产品的 CEIP，请参见加入或退出客户体验提升计划。

加入或退出客户体验提升计划

从 vRealize Orchestrator Appliance 命令行加入客户体验提升计划。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 要加入客户体验提升计划，请运行 `vracli ceip on` 命令。

3 查看客户体验提升计划信息，然后运行 `vracli ceip on --acknowledge-ceip` 命令。

4 重新启动 vRealize Orchestrator 服务。

   a 要重新启动服务器服务，请运行 `kubectl -n prelude exec -it your_vro_pod -c vco-server-app /bin/bash` 命令。

   b 要停止服务，请运行 `kill 1` 命令。

   c 要重新启动控制中心服务，请运行 `kubectl -n prelude exec -it your_vro_pod -c vco-controlcenter-app /bin/bash` 命令。

   d 要停止服务，请运行 `kill 1` 命令。

5 要退出客户体验提升计划，请运行 `vracli ceip off` 命令。

6 重复用于重新启动服务的步骤。

使用 vRealize Orchestrator API 服务

6

除了使用控制中心配置 vRealize Orchestrator 外，还可以使用存储在设备中的 vRealize Orchestrator REST API、控制中心 REST API 或命令行实用程序来修改 vRealize Orchestrator 服务器配置设置。

默认情况下，vRealize Orchestrator 软件包中包含配置插件。可以通过 vRealize Orchestrator 工作流库或 vRealize Orchestrator REST API 访问配置插件工作流。使用这些工作流，可以更改 vRealize Orchestrator 服务器的受信任证书以及密钥库设置。有关所有可用 vRealize Orchestrator REST API 服务调用的信息，请参见 vRealize Orchestrator 服务器 API 文档，位于 https://*your_orchestrator_FQDN*/vco/api/docs。

使用 REST API 管理 TLS 证书和密钥库

除了使用控制中心管理 TLS 证书外，您还可以通过运行配置插件中的工作流或使用 REST API 来管理受信任证书和密钥库。

配置插件包含用于导入和删除 TLS 证书及密钥库的工作流。可以通过在 vRealize Orchestrator Client 中导航到**库 > 工作流 > SSL 信任管理器**和**库 > 工作流 > 密钥库**来访问这些工作流。此外，还可以使用 vRealize Orchestrator REST API 运行这些工作流。

可以通过控制中心 REST API 访问用于配置 vRealize Orchestrator 服务器的资源。您可以使用控制中心 REST API 与第三方系统自动设置 vRealize Orchestrator 配置。控制中心 REST API 的 root 端点为 https://*your_orchestrator_FQDN*/vco/api。有关控制中心 REST API 的所有可用服务调用的信息，请参见《vRealize Orchestrator 控制中心 API》文档，位于 https://*your_orchestrator_FQDN*/vco-controlcenter/docs。

使用 REST API 删除 TLS 证书

您可以运行配置插件的“删除受信任证书”工作流或使用 REST API 删除 TLS 证书。

步骤

1 在“删除受信任证书”工作流的工作流服务的 URL 发起 GET 请求。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Delete trusted
certificate
```

2 在定义的 URL 发起 GET 请求以检索“删除受信任证书”工作流定义。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/8a70a326-
ffd7-4fef-97e0-2002ac49f5bd
```

3 在持有“删除受信任证书”工作流执行对象的 URL 发起 POST 请求。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/8a70a326-
ffd7-4fef-97e0-2002ac49f5bd/executions/
```

4 在请求正文中，将想要删除的证书的名称作为“删除受信任证书”工作流的输入参数应用于执行上下文的元素。

使用 REST API 导入 TLS 证书

您可以运行配置插件中的工作流或使用 REST API 导入 TLS 证书。

您可以从文件或 URL 导入受信任的证书。请参见通过控制中心导入受信任证书

步骤

1 在工作流服务的 URL 发起 GET 请求。

选项	描述
从文件导入受信任证书	从文件导入受信任证书。
从 URL 导入受信任证书	从 URL 地址导入受信任证书。
使用代理服务器从 URL 导入受信任证书	使用代理服务器从 URL 地址导入受信任证书。
从 URL 导入受信任证书及证书别名	从 URL 地址导入受信任证书及证书别名。

若要从文件导入受信任证书，请发起以下 GET 请求：

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Import
                trusted certificate from a file
```

2 在定义的 URL 发起 GET 请求以检索工作流定义。

若要检索“从文件导入受信任证书”工作流的定义，请发起以下 GET 请求：

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5
```

3 在工作流执行对象所在的 URL 发起 POST 请求。

对于“从文件导入受信任证书”工作流，请发起以下 POST 请求：

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5/executions
```

4 在请求正文中，提供工作流的输入参数值用于执行上下文元素。

参数	描述
cer	要从其中导入 TLS 证书的 CER 文件。 此参数适用于“从文件导入受信任证书”工作流。
url	要从其中导入 TLS 证书的 URL。对于非 HTTPS 服务，支持的格式为 *IP_address_or_DNS_name:port*。 此参数适用于“从 URL 导入受信任证书”工作流。

使用 REST API 创建密钥库

您可以运行配置插件的“创建密钥库”工作流或使用 REST API 创建密钥库。

步骤

1 在“创建密钥库”工作流的工作流服务的 URL 发起 GET 请求。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Create a keystore
```

2 在定义的 URL 发起 GET 请求以检索“创建密钥库”工作流的定义。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/
```

3 在持有“创建密钥库”工作流执行对象的 URL 发起 POST 请求。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/executions/
```

4 在请求正文中，将想要创建的密钥库的名称作为“创建密钥库”工作流的输入参数应用于执行上下文的元素。

使用 REST API 删除密钥库

您可以运行配置插件的“删除密钥库”工作流或使用 REST API 删除密钥库。

步骤

1 在“删除密钥库”工作流的工作流服务的 URL 发起 GET 请求。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Delete a keystore
```

2 在定义的 URL 发起 GET 请求以检索“删除密钥库”工作流的定义。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/
7a3389eb-1fab-4d77-860b-81b66bb45b86/
```

3 在“删除密钥库”工作流执行对象所在的 URL 发起 POST 请求。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/
7a3389eb-1fab-4d77-860b-81b66bb45b86/executions/
```

4 在请求正文中，将想要删除的密钥库作为“删除密钥库”工作流的输入参数应用于执行上下文的元素。

使用 REST API 添加密钥

您可以运行配置插件的“添加密钥”工作流或使用 REST API 添加密钥。

步骤

1 在“添加密钥”工作流的工作流服务的 URL 发起 GET 请求。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Add key
```

2 在定义的 URL 发起 GET 请求以检索“添加密钥”工作流的定义。

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/
```

3 在持有“添加密钥”工作流的执行对象的 URL 发起 POST 请求。

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/executions/
```

4 在请求正文中，将密钥库、密钥别名、PEM 加密的密钥、证书链和密钥密码作为“添加密钥”工作流的输入参数应用于执行上下文的元素。

其他配置选项

7

可以使用控制中心更改默认的 vRealize Orchestrator 行为。

本章讨论了以下主题：

- 重新配置身份验证
- 配置工作流运行属性
- vRealize Orchestrator 日志文件
- 启用 Opentracing 扩展和 Wavefront 扩展
- 为 vRealize Orchestrator 启用时间同步
- 为 vRealize Orchestrator 停用时间同步
- 配置 vRealize Orchestrator Kubernetes CIDR
- 更新 vRealize Orchestrator 的 DNS 设置

重新配置身份验证

在控制中心的初始配置期间设置身份验证方法后，您可以在任何时候更改身份验证提供程序或已配置的参数。

更改身份验证提供程序

要更改身份验证模式或身份验证提供程序连接设置，必须先取消注册现有的身份验证提供程序。

步骤

1 以 **root** 用户身份登录控制中心。

2 在**配置身份验证提供程序**页面中，单击主机地址文本框旁边的**注销**按钮以注销正在使用的身份验证提供程序。

结果

您即成功注销身份验证提供程序。

后续步骤

重新配置控制中心的身份验证。请参见配置独立 vRealize Orchestrator 服务器。

更改身份验证参数

在控制中心中将 vSphere 用作身份验证提供程序时，可以更改 vRealize Orchestrator 管理员组的默认租户。

前提条件

将 vSphere 配置为 vRealize Orchestrator 部署的身份验证提供程序。请参见使用 vSphere 身份验证配置独立 vRealize Orchestrator 服务器。

注 vRealize Automation 身份验证不包含这些参数。

步骤

1 以 **root** 用户身份登录控制中心。

2 选择**配置身份验证提供程序**。

3 单击**默认租户**文本框旁边的**更改**按钮。

4 替换租户的名称。

5 单击**管理员组**文本框旁边的**更改**按钮。

注 如果您未重新配置管理员组，它仍将保留为空且您将无法再访问控制中心。

6 输入管理员组的名称，然后单击**搜索**。

7 选择管理员组。

8 更改管理员组。

9 要完成身份验证参数的编辑，请单击**保存更改**。

配置工作流运行属性

默认情况下，每个节点最多可以运行 300 个工作流，在达到活动运行工作流的数量限制时最多可将 1 万个工作流排入队列。

如果 vRealize Orchestrator 节点需要运行的并发工作流数超过 300，挂起的工作流运行会排队等待。在活动工作流运行完成后，队列中的下一工作流开始运行。如果达到排队工作流的最大数量，则后续工作流运行会失败，直到某个挂起工作流开始运行为止。

您可以在控制中心的**高级选项**页面上配置工作流运行属性。

选项	说明
启用安全模式	如果启用安全模式，则所有正在运行的工作流都将被取消，而且在 vRealize Orchestrator 节点下次启动时不会恢复。
并行运行的工作流数	同时运行的工作流数。默认情况下，每个节点 300 个工作流。
队列中正在运行的工作流数上限	vRealize Orchestrator 服务器在变为不可用之前接受的工作流运行请求数。默认情况下，每个节点 10,000 个工作流。

选项	说明
每个工作流保存的运行数上限	每个工作流已完成且可保存为历史记录的工作流运行数上限。一旦超出此数，则最早的工作流运行将被删除。默认值为每个工作流 100 次运行。
日志事件保留天数	数据库中日志事件在被清除前的保留天数。默认为 15 天。

vRealize Orchestrator 日志文件

在您提交支持请求时，VMware 技术支持会例行要求您提供诊断信息。这一诊断信息包含了运行产品的主机上的产品特定日志和配置文件。

vRealize Orchestrator Appliance 日志存储在 `/data/vco/usr/lib/vco/app-server/logs/` 目录中。通过登录到设备命令行并运行 `vracli log-bundle` 命令，可以导出 vRealize Orchestrator Appliance 部署的日志。生成的日志包保存在 vRealize Orchestrator Appliance 的根文件夹中。

日志记录持久性

您能以任何形式的 vRealize Orchestrator 脚本（例如工作流、策略或操作）记录信息。此类信息都会具有类型和级别之分。类型可以是持久性和非持久性。级别可以是调试、信息、警告、错误、跟踪和严重。

表 7-1. 创建持久性和非持久性日志

日志级别	持久性类型	非持久性类型
调试	Server.debug("short text", "long text");	System.debug("text")
信息	Server.log("short text", "long text");	System.log("text");
警告	Server.warn("short text", "long text");	System.warn("text");
错误	Server.error("short text", "long text");	System.error("text");

持久性日志

持久性日志（服务器日志）会跟踪过往的工作流运行日志并存储在 vRealize Orchestrator 数据库中。

非持久性日志

使用非持久性日志（系统日志）创建脚本时，vRealize Orchestrator 服务器会就此日志通知所有正在运行的 vRealize Orchestrator 应用程序，但此信息不会存储在数据库中。在应用程序重启后，日志信息就会丢失。非持久性日志用于调试用途和实时信息。要查看系统日志，必须在 vRealize Orchestrator Client 中选择一个已完成的工作流运行，然后选择**日志**选项卡。

vRealize Orchestrator 日志配置

在控制中心的**配置日志**页面上，可以设置服务器日志以及所需脚本日志的日志级别。如果某个日志一天内生成多次，则会很难确定问题原因。

服务器日志以及脚本日志的默认日志级别为信息。更改日志级别会影响服务器输入到日志的所有新消息，以及数据库的活动连接数量。日志记录详细级别按降序递减。

小心 仅在调试问题时将日志级别设置为调试或所有。请勿在生产环境中使用这些设置，因为会严重影响性能。

生成 vRealize Orchestrator 日志

通过以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行并运行 `vracli log-bundle` 命令，可以导出部署的日志。生成的日志包存储在设备的根文件夹中。

。

注 当群集中有多个 vRealize Orchestrator 实例时，日志包会包含群集中所有 vRealize Orchestrator 实例的日志。

配置与 vRealize Log Insight 的日志记录集成

您可以将 vRealize Orchestrator 配置为将日志记录信息发送到 vRealize Log Insight 服务器。

您可以通过 vRealize Orchestrator Appliance 命令行配置到 vRealize Log Insight 服务器的日志记录集成。

注 有关配置与远程 syslog 服务器的日志记录集成的信息，请参见在 vRealize Orchestrator 中创建或覆盖 Syslog 集成。

前提条件

- 配置 vRealize Log Insight 服务器。请参见 vRealize Log Insight 文档。
- 确认您的 vRealize Log Insight 为 4.7.1 或更高版本。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 要配置与 vRealize Log Insight 的日志记录集成，请运行 `vracli vrli set` *vRLI_FQDN* 命令。

   注 如果您的 vRealize Orchestrator 实例使用自签名证书，则可以通过包括可选的 `-k` 或 `--insecure` 参数来禁用 SSL 身份验证。

后续步骤

有关 vRealize Log Insight 配置选项的详细信息，请运行 `vracli vrli -h` 命令。

在 vRealize Orchestrator 中创建或覆盖 Syslog 集成

您可以将 vRealize Orchestrator 配置为将日志记录信息发送到一个或多个远程 syslog 服务器。

`vracli remote-syslog set` 命令用于创建 syslog 集成或覆盖现有集成。

vRealize Orchestrator 远程 syslog 集成支持以下三种连接类型：

- 通过 UDP。
- 通过 TCP 且不使用 TLS。

  注 要在不使用 TLS 的情况下创建 syslog 集成，请将 `--disable-ssl` 标志添加到 `vracli remote-syslog set` 命令。

- 通过 TCP 且使用 TLS。

有关配置与 vRealize Log Insight 的日志记录集成的信息，请参见配置与 vRealize Log Insight 的日志记录集成。

前提条件

配置一个或多个远程 syslog 服务器。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 要创建与 syslog 服务器的集成，请运行 `vracli remote-syslog set` 命令。

```
vracli remote-syslog set -id name_of_integration protocol_type://
syslog_URL_or_FQDN:syslog_port
```

注 如果不在 `vracli remote-syslog set` 命令中输入端口，则端口值默认为 514。

注 您可以将证书添加到 syslog 配置中。要添加证书文件，请使用 `--ca-file` 标志。要以纯文本形式添加证书，请使用 `--ca-cert` 标志。

3 （可选） 要覆盖现有 syslog 集成，请运行 `vracli remote-syslog set`，并将 `-id` 标志值设置为要覆盖的集成的名称。

注 默认情况下，vRealize Orchestrator Appliance 会请求您确认希望覆盖 syslog 集成。要跳过确认请求，请将 `-f` 或 `--force` 标志添加到 `vracli remote-syslog set` 命令。

后续步骤

要查看设备中的当前 syslog 集成，请运行 `vracli remote-syslog` 命令。

在 vRealize Orchestrator 中删除 syslog 集成

您可以通过运行 `vracli remote-syslog unset` 命令从 vRealize Orchestrator Appliance 中删除 syslog 集成。

前提条件

在 vRealize Orchestrator Appliance 中创建一个或多个 syslog 集成。请参见在 vRealize Orchestrator 中创建或覆盖 Syslog 集成。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 从 vRealize Orchestrator Appliance 中删除 syslog 集成。

a 要删除特定 syslog 集成，请运行 `vracli remote-syslog unset -id` *Integration_name* 命令。

b 要删除 vRealize Orchestrator Appliance 上的所有 syslog 集成，请在不使用 `-id` 标志的情况下运行 `vracli remote-syslog unset` 命令。

> 注 默认情况下，vRealize Orchestrator Appliance 会请求您确认删除所有 syslog 集成。要跳过确认请求，请将 `-f` 或 `--force` 标志添加到 `vracli remote-syslog unset` 命令。

启用 Kerberos 调试日志记录

可以通过修改 vRealize Orchestrator 插件使用的 Kerberos 配置文件对插件问题进行故障排除。

Kerberos 配置文件位于 vRealize Orchestrator Appliance 的 `/data/vco/usr/lib/vco/app-server/conf/` 目录中。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 运行 `kubectl -n prelude edit deployment vco-app` 命令。

3 在部署文件中，找到并编辑 `-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf` 字符串。

```
-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf
-Dsun.security.krb5.debug=true'
```

4 保存更改并退出文件编辑器。

5 运行 `kubectl -n prelude get pods` 命令。

等待所有 pod 都运行。

6 验证 Kerberos 调试日志记录是否已启用。

```
kubectl -n prelude log {vco_app_name} -c vco-server-app | grep krb5
```

验证日志是否包含类似的消息。

```
kubectl -n prelude log vco-app-5c965f9b9d-v8srd -c vco-server-app | grep krb5
12:23:05,417 INFO O11N:75 - Sysprop: java.security.krb5.conf = /usr/lib/vco/app-server/
conf/krb5.conf
12:23:05,421 INFO O11N:75 - Sysprop: sun.security.krb5.debug = true
2019-10-22 12:23:38.521+0000 [Thread-19] INFO {} [O11N] Sysprop: java.security.krb5.conf
= /usr/lib/vco/app-server/conf/krb5.conf
2019-10-22 12:23:38.525+0000 [Thread-19] INFO {} [O11N] Sysprop: sun.security.krb5.debug =
true
Java config name: /usr/lib/vco/app-server/conf/krb5.conf
EType: sun.security.krb5.internal.crypto.Aes256CtsHmacSha1EType
```

启用 Opentracing 扩展和 Wavefront 扩展

vRealize Orchestrator 的 Opentracing 和 Wavefront 扩展提供了用于收集有关 vRealize Orchestrator 环境数据的工具。您可以使用这些数据对 vRealize Orchestrator 系统和工作流进行故障排除。

配置 vRealize Orchestrator 以使用 Opentracing 扩展和 Wavefront 扩展之前，必须先在 vRealize Orchestrator Appliance 中启用这些扩展。

前提条件

- 确认已启用 vRealize Orchestrator Appliance SSH 服务。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。
- 如果已启用先前版本的 Opentracing 或 Wavefront 扩展，则必须先移除这些扩展，然后再启用当前版本。例如，如果您先前启用了版本 8.1.0 的 Wavefront 扩展，则必须运行 `rm /data/vco/usr/lib/vco/app-server/extensions/wavefront-8.1.0.jar` 命令。

步骤

1. 使用 SSH 以 **root** 用户身份登录 vRealize Orchestrator Appliance。
2. 要列出所有可用的扩展，请运行 `ls /data/vco/usr/lib/vco/app-server/extensions/` 命令。
3. 运行以下命令以启用 Opentracing 扩展。

   ```
   mv /data/vco/usr/lib/vco/app-server/extensions/opentracing-8.7.0.jar.inactive /
   data/vco/usr/lib/vco/app-server/extensions/opentracing-8.7.0.jar
   ```

4. 运行以下命令以启用 Wavefront 扩展。

   ```
   mv /data/vco/usr/lib/vco/app-server/extensions/wavefront-8.7.0.jar.inactive /
   data/vco/usr/lib/vco/app-server/extensions/wavefront-8.7.0.jar
   ```

5. 登录到控制中心，并确认这些扩展显示在**扩展属性**页面中。

后续步骤

在**扩展属性**页面中配置 Opentracing 和 Wavefront 与 vRealize Orchestrator 的集成。请参见配置 Opentracing 扩展和配置 Wavefront 扩展。

配置 Opentracing 扩展

Opentracing 扩展将有关工作流运行的数据发送到 Jaeger 服务器。这些数据包括工作流状态、输入和输出参数、启动工作流运行的用户以及工作流 ID 数据。

前提条件

- 确认已在 vRealize Orchestrator Appliance 中启用 Opentracing。请参见启用 Opentracing 扩展和 Wavefront 扩展。
- 部署 Jaeger 服务器以供在 Opentracing 扩展中使用。有关详细信息，请参见 Jaeger 入门文档。

步骤

1. 以 **root** 用户身份登录控制中心。
2. 选择**扩展属性**页面。
3. 选择 Opentracing 扩展。
4. 输入 Jaeger 服务器主机地址和端口。

   注 插入两个正斜杠 ("//")，然后输入服务器地址。
5. 单击**保存**。

结果

您已为 vRealize Orchestrator 配置了 Opentracing 扩展。

后续步骤

- 要访问包含 Opentracing 扩展所收集数据的 Jaeger UI，请访问在配置期间输入的主机地址。
- 在**服务**选项下，选择**工作流**。
- 要指定要查看的数据，请使用**标记**选项。例如，要查看有关失败工作流的数据，请输入 **`status=failed`**。

配置 Wavefront 扩展

使用 Wavefront 扩展可以收集有关 vRealize Orchestrator 系统和工作流的衡量指标数据。

前提条件

1. 确认已在 vRealize Orchestrator Appliance 中启用 Wavefront。请参见启用 Opentracing 扩展和 Wavefront 扩展。

2 导入 Wavefront 证书。

a 以 **root** 用户身份登录到 vRealize Orchestrator 控制中心。

b 选择**证书**页面。

c 单击**导入**下拉菜单，然后选择**从 URL 导入**。

d 输入 Wavefront URL，然后单击**导入**。

3 配置 Wavefront 代理。有关详细信息，请参见安装和配置 Wavefront 代理。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator 控制中心。

2 选择**扩展属性**页面。

3 选择 Wavefront 扩展。

4 配置 Wavefront 属性。

选项	描述
代理	Wavefront 代理地址。
主机	可选。Wavefront 主机地址。
令牌	可选。Wavefront API 令牌。有关生成 Wavefront API 令牌的详细信息，请参见生成 API 令牌。
前缀	为发送到 Wavefront 的每个衡量指标添加前缀标签。前缀标签使用点号分隔。

5 （可选） 选择**在下次启动时发送默认仪表板**。

6 单击**保存**。

结果

您已为 vRealize Orchestrator 配置了 Wavefront 扩展。

后续步骤

- 要访问 Wavefront 收集的衡量指标，请通过在配置期间输入的地址访问仪表板。
- 要获取有关 vRealize Orchestrator 环境中特定事件的通知，可以使用 Wavefront 警示。有关详细信息，请参见 Wavefront 警示文档。

为 vRealize Orchestrator 启用时间同步

可以使用 vRealize Orchestrator Appliance 命令行在 vRealize Orchestrator 部署上启用时间同步。

可以使用网络时间协议 (Network Time Protocol, NTP) 通信协议为独立或集群 vRealize Orchestrator 部署配置时间同步。vRealize Orchestrator 支持两个互斥的 NTP 配置：

NTP 配置	说明
ESXi	当托管 vRealize Orchestrator Appliance 的 ESXi 服务器与 NTP 服务器同步时，可以使用此配置。如果使用的是集群部署，则所有 ESXi 主机都必须与 NTP 服务器同步。有关为 ESXi 配置 NTP 的详细信息，请参见使用 vSphere Web Client 在 ESXi 主机上配置网络时间协议 (NTP)。 **注** 如果 vRealize Orchestrator 部署迁移到与 NTP 服务器不同步的 ESXi 主机，可能会遇到时钟偏移问题。
systemd	此配置使用 systemd-timesyncd 守护进程同步 vRealize Orchestrator 部署的时钟。 **注** 默认情况下，systemd-timesyncd 守护进程处于启用状态，但未配置 NTP 服务器。如果 vRealize Orchestrator Appliance 使用动态 IP 配置，则该设备可以使用通过 DHCP 协议接收的任何 NTP 服务器。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 对 ESXi 启用 NTP。

a 运行 `vracli ntp esxi` 命令。

b （可选）要确认 NTP 配置的状态，请运行 `vracli ntp status` 命令。

3 对 systemd 启用 NTP。

a 运行 `vracli ntp systemd --set` *FQDN_or_IP_of_systemd_server* 命令。

注 可以添加多个 systemd NTP 服务器，用逗号分隔其网络地址即可。必须将每个网络地址置于单引号括内。例如，`vracli ntp systemd --set` *'ntp_address_1'*、*'ntp_address_2'*

b （可选）要确认 NTP 配置的状态，请运行 `vracli ntp status` 命令。

结果

您已为 vRealize Orchestrator 部署启用时间同步。

后续步骤

如果 NTP 服务器和 vRealize Orchestrator 部署之间的时间差超过 10 分钟，则 NTP 配置可能会失败。要解决此问题，请重新引导 vRealize Orchestrator Appliance。

为 vRealize Orchestrator 停用时间同步

可以使用 vRealize Orchestrator Appliance 命令行在 vRealize Orchestrator 部署上停用网络时间协议 (NTP) 时间同步。

您也可以通过运行 `vracli ntp reset` 命令，将 vRealize Orchestrator Appliance 的 NTP 配置重置为默认状态。

前提条件

确认您配置了与 ESXi 或 systemd 保持时间同步。请参见为 vRealize Orchestrator 启用时间同步。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 要停用与 ESXi 或 systemd 保持时间同步，请运行 `vracli ntp disable` 命令。

3 （可选） 要确认 NTP 配置的状态，请运行 `vracli ntp status` 命令。

配置 vRealize Orchestrator Kubernetes CIDR

您可以在部署后更改 Kubernetes 无类域间路由 (CIDR) 子网掩码。

vRealize Orchestrator Appliance 配置并运行 Kubernetes 集群。此集群中的容器和服务将部署在单独的 IPv4 子网中，分别由内部集群 CIDR 和内部服务 CIDR 表示。在 OVF 部署过程中设置的子网掩码的默认值如下：

Kubernetes network property	Default value	Property description
cluster-cidr	10.244.0.0/22	用于 Kubernetes 集群内运行的容器的 CIDR。
service-cidr	10.244.4.0/22	用于 Kubernetes 集群内 Kubernetes 服务的 CIDR。

默认 CIDR 网络地址可能会与您正在使用的外部专用网络发生冲突。在这种情况下，您可以在部署 vRealize Orchestrator Appliance 期间或之后更改这些 CIDR 值的配置。

注 有关在设备部署期间更改 CIDR 配置的信息，请参见 下载并部署 vRealize Orchestrator Appliance。

前提条件

- 确认 CIDR 地址值至少支持 1024 个主机。
- 内部集群 CIDR 和内部服务 CIDR 不得共享同一子网值。
- 其中一个子网的 CIDR 值不能包含要添加到另一个子网的值。

  注 例如，`cluster-cidr` 值不能为 **`10.244.4.0/22 10.244.4.0/24`**，因为这还包含 `service-cidr` 属性的子网值。必须单独添加每个子网值。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance。

2 运行 `vracli upgrade exec -y --prepare --profile k8s-subnets` 命令。

3 通过生成虚拟机 (VM) 快照备份 vRealize Orchestrator 部署。请参见生成虚拟机快照。

  小心 vRealize Orchestrator 8.x 当前不支持内存快照。创建 vRealize Orchestrator 部署的快照之前，请验证是否已停用**创建虚拟机内存的快照**选项。

4 通过运行 `vracli network k8s-subnets` 命令来更改集群 CIDR 和服务 CIDR 子网的值。

```
vracli network k8s-subnets --cluster-cidr <CIDR_value> --service-cidr <CIDR_value>
```

5 要完成 CIDR 配置过程，请运行 `vracli upgrade exec` 命令。

更新 vRealize Orchestrator 的 DNS 设置

管理员可以使用 `vracli network dns` 命令更新 vRealize Orchestrator 部署的 DNS 设置。

前提条件

确认已启用 vRealize Orchestrator Appliance SSH 服务。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。

步骤

1 通过 SSH 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

注 对于已加入集群的部署，请登录到集群中任何节点的设备。

2 要为您的 vRealize Orchestrator 部署设置新的 DNS 服务器，请运行 `vracli network dns set` 命令。

```
vracli network dns set --servers DNS1,DNS2
```

3 通过运行 `vracli network dns status` 命令，确认已将新 DNS 服务器正确应用于所有 vRealize Orchestrator 节点。

4 要停止部署中的 vRealize Orchestrator 服务，请运行以下命令集：

```
/opt/scripts/svc-stop.sh
sleep 120
/opt/scripts/deploy.sh --onlyClean
```

5 重新启动 vRealize Orchestrator 节点并等待它们完全启动。

6 通过 SSH 登录到每个 vRealize Orchestrator 节点的命令行，并确认 `/etc/resolve.conf` 文件中列出了新的 DNS 服务器。

7 要启动 vRealize Orchestrator 服务，请在部署中的其中一个节点上运行 `/opt/scripts/deploy.sh` 脚本。

结果

已按指定方式更改 vRealize Orchestrator DNS 设置。

配置用例及故障排除

8

配置用例提供了各种任务流，您可以执行这些任务流来满足 vRealize Orchestrator 服务器的具体配置要求，还提供了故障排除主题，以便您了解并解决问题。

本章讨论了以下主题：

验证 vRealize Orchestrator 服务器内部版本号

在某些情况下，您可能需要验证 vRealize Orchestrator 部署的服务器内部版本号。

您可以通过导航到 https://*your_orchestrator_FQDN*/vco/api/about，验证 vRealize Orchestrator 服务器内部版本号。服务器内部版本号显示在 `<ns2:build-number>` 标记中。

验证服务器内部版本号在各种用例中都非常有用，例如，可向通过 VMware 技术支持团队记录的支持请求 (Support Request, SR) 提供额外的信息。

注 vRealize Orchestrator 服务器内部版本号与 vRealize Orchestrator Appliance 内部版本号不同。要验证该设备内部版本号，请登录到 vRealize Orchestrator Appliance 命令行并运行 `vracli version` 命令。验证该设备内部版本号可帮助您确认已成功升级到最新版本的 vRealize Orchestrator。

为 vSphere Web Client 配置 vRealize Orchestrator 插件

要使用 vSphere Web Client 的 vRealize Orchestrator 插件，必须将 vRealize Orchestrator 注册为 vCenter Server 的扩展。

在将 vRealize Orchestrator 服务器注册到 vCenter Single Sign-On 并将其配置为与 vCenter Server 结合使用后，必须将 vRealize Orchestrator 注册为 vCenter Server 的扩展。

前提条件

- 确认已为 vRealize Orchestrator Appliance 启用 SSH 访问。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。
- 您必须使用 vSphere 身份验证将 vRealize Orchestrator 注册到受管 vCenter Server 进行身份验证时所用的同一 Platform Services Controller。
- 将 `vco-plugin.zip` 复制到 vRealize Orchestrator Appliance：
  a 从 VMware 技术网下载 `vco-plugin.zip` 文件。
  b 打开 SSH 客户端。

  注 对于 Linux 或 MacOS 环境，可以使用终端命令行界面。对于 Windows 环境，可以使用 PuTTY 客户端。

  c 要复制 `vco-plugin` 文件，请运行安全复制命令。

```
For Linux/MacOS: scp ~/<zip_download_dir>/vco-plugin.zip
root@<orchestrator_FQDN_or_IP>:/data/vco/usr/lib/vco/downloads/vco-plugin.zip
```

```
For Windows: pscp C:\<zip_download_dir>\vco-plugin.zip root@<orchestrator_FQDN_or_IP>:/
data/vco/usr/lib/vco/downloads/vco-plugin.zip
```

步骤

1 登录到 vRealize Orchestrator Client。
2 导航到**库 > 工作流**。
3 搜索**将 vCenter Orchestrator 注册为 vCenter Server 扩展**工作流并单击**运行**。
4 选择要向其注册 vRealize Orchestrator 的 vCenter Server 实例。
5 输入 https://*your_orchestrator_FQDN* 或负载平衡器的服务 URL（用于将请求重定向到 vRealize Orchestrator 服务器节点）。
6 单击**运行**。

取消正在运行的工作流

您可以使用 vRealize Orchestrator 控制中心取消未正常完成的工作流。

步骤

1 以 **root** 用户身份登录控制中心。
2 单击**故障排除**。

3 取消正在运行的工作流。

选项	描述
取消所有工作流运行	输入工作流 ID，取消该工作流的所有令牌。
按 ID 取消工作流运行	输入想要取消的所有令牌 ID，使用逗号进行分隔。
取消所有正在运行的工作流	取消服务器上正在运行的所有工作流。

注 按 ID 取消工作流的操作可能不会成功，因为没有可靠的方式能立即取消运行线程。

结果

在下一次服务器启动时，工作流会设置为已取消状态。

启用 vRealize Orchestrator 服务器调试

开发插件时，可以在调试模式下启动 vRealize Orchestrator 服务器，以调试遇到的问题。

前提条件

在本地计算机上安装并配置 Kubernetes 命令行工具。请参见安装并设置 kubectl。

步骤

1 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

2 运行 `kubectl -n prelude edit deployment vco-app` 命令。

3 编辑部署 YAML 文件，向 `vco-server-app` 容器中添加调试环境变量。该变量必须添加到 `vco-server-app` 容器的 `env` 部分下。

```
containers:
      - command:
      ...
        env:
        - name: DEBUG_PORT
          value: "your_desired_debug_port"
      ...
    name: vco-server-app
      ...
```

注 将调试环境变量添加到 `env` 部分时，必须采用 YAML 缩进格式，如上述示例所示。

4 保存对部署文件所做的更改。

如果编辑部署文件成功，您将收到 `deployment.extensions/vco-app edited` 消息。

5 通过运行 `vracli dev kubeconfig` 命令生成 Kubernetes 配置文件。

由于 kubeconfig 是开发人员环境，因此系统会提示您确认是否继续。输入 **yes** 继续；输入 **no** 停止。

6 复制所生成配置文件的内容，从 `apiVersion: v1` 直到 `client-key-data` 内容（并包括该内容）。

7 将生成的 Kubernetes 配置文件保存在本地计算机上。

8 注销 vRealize Orchestrator Appliance。

9 在本地计算机上完成调试模式的配置。

a 打开命令行 shell。

b 将 *KUBECONFIG* 环境变量绑定到保存的配置文件。

> **注** 此示例基于 Linux 环境。

```
export KUBECONFIG=/file/path/fileName
```

c 要验证服务是否正在运行，请运行 `kubectl cluster-info` 命令。

d 要完成调试模式的配置，请执行以下 Kubernetes API 请求。

> **注** *localhost_debug_port* 变量的值是在集成开发环境 (IDE) 的远程调试配置中设置的端口。*vro_debug_port* 变量的值在此过程的步骤 3 中生成。

```
kubectl port-forward pod/vco_app_pod_ID localhost_debug_port:vro_debug_port
```

> **重要事项** 配置调试工具时，请提供在其中执行端口转发命令的本地计算机的 DNS 和 IP 设置。

结果

您已为 vRealize Orchestrator Appliance 配置服务器调试。

调整 vRealize Orchestrator Appliance 磁盘的大小

可以通过在 vSphere 中编辑 vRealize Orchestrator Appliance 虚拟机的磁盘大小设置来修改 vRealize Orchestrator Appliance 的磁盘大小。

前提条件

确认已启用 vRealize Orchestrator Appliance SSH 服务。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。

步骤

1 验证 vRealize Orchestrator Appliance 中的当前可用磁盘空间。

> **注** vRealize Orchestrator Appliance 磁盘至少需要 20% 的可用磁盘空间。

a 通过 SSH 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

b 运行 `vracli disk-mgr` 命令。

2 在 vSphere 中调整 vRealize Orchestrator Appliance 虚拟机的磁盘大小。

a 以**管理员**身份登录到 vSphere Client。

b 右键单击虚拟机，然后选择**编辑设置**。

c 在**虚拟硬件**选项卡上，展开**硬盘**以查看并更改磁盘设置，然后单击**确定**。

有关更改 vSphere 虚拟机磁盘大小的详细信息，请参见《vSphere 虚拟机管理》中的“更改虚拟磁盘配置”。

3 在 Photon OS 中触发自动调整大小。

a 通过 SSH 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

b 运行 `vracli disk-mgr resize` 命令。

注 您可以在以下位置跟踪调整磁盘大小过程的进度：`/var/log/vmware/prelude/disk_resize.log`。

您已调整 vRealize Orchestrator Appliance 磁盘的大小。

4 通过运行 `disk-mgr` 命令验证调整磁盘大小的进程是否成功。

```
vracli disk-mgr
```

后续步骤

要对调整磁盘大小过程中出现的问题进行故障排除，请参见知识库文章 79925。

如何缩放 vRealize Orchestrator 服务器的堆内存大小

您可以通过创建自定义配置文件并修改资源衡量指标文件来扩展 vRealize Orchestrator 服务器的堆内存大小。

可以调整 vRealize Orchestrator 服务器的堆内存大小，以便编排环境能够管理不断变化的工作负载。例如，如果计划管理多个 vCenter Server，则可以增加 vRealize Orchestrator 部署的堆内存。

前提条件

- 启用对 vRealize Orchestrator Appliance 的 SSH 访问。请参见启用或禁用对 vRealize Orchestrator Appliance 的 SSH 访问。
- 将部署了 vRealize Orchestrator 的虚拟机的 RAM 增加到下一个适当的增量。由于务必要为其余服务保留足够的可用内存，因此必须先扩展 vRealize Orchestrator Appliance 资源。例如，如果所需的堆内存为 `7G`，则 vRealize Orchestrator Appliance RAM 应分别递增 `4G`，因为默认堆内存值 `3G` 与所需堆内存之间相差 `4G`。有关在 vSphere 中增加虚拟机的 RAM 的信息，请参见《vSphere 虚拟机管理》中的更改内存配置。

步骤

1 通过 SSH 以 **root** 用户身份登录 vRealize Orchestrator Appliance 命令行。

2 要创建自定义配置文件目录和在配置文件处于活动状态时使用的所需目录树，请运行以下脚本：

```
vracli cluster exec -- bash -c 'base64 -d <<<
IyBDcmVhdGUgY3VzdG9tIHByb2ZpbGUgZGlyZWN0b3J5Cm1rZGlyIC1wIC9ldGMvdm13YXJlLXByZWx1ZGUvcHJvZml
sZXMvY3VzdG9tLXByb2ZpbGUvCgojIENyZWF0ZSB0aGUgcmVxdWlyZWQgZGlyZWN0b3J5IHRyZWUgdGhhdCB3aWxsIG
JlIHVzZWQgd2hlbiB0aGUgcHJvZmlsZSBpcyBhY3RpdmUKbWtkaXIgLXAgL2V0Yy92bXdhcmUtcHJlbHVkZS9wcm9ma
Wxlcy9jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLwoKIyBDcmVhdGUgImNoZWNrIiBmaWxlIHRoYXQgaXMg
YW4gZXhlY3V0YWJsZSBmaWxlIHJ1biBieSBkZXBsb3kgc2NyaXB0LgpjYXQgPDxFT0YgPiAvZXRjL3Ztd2FyZS1wcmV
sdWRlL3Byb2ZpbGVzL2N1c3RvbS1wcm9maWxlL2NoZWNrCiMhL2Jpbi9iYXNoCmV4aXQgMApFT0YKY2htb2QgNzU1IC
9ldGMvdm13YXJlLXByZWx1ZGUvcHJvZmlsZXMvY3VzdG9tLXByb2ZpbGUvY2hlY2sKCiMgQ29weSB2Uk8gcmVzb3VyY
2UgbWV0cmljcyBmaWxlIHRvIHlvdXIgY3VzdG9tIHByb2ZpbGUKY2F0IDw8RU9GID4gL2V0Yy92bXdhcmUtcHJlbHVk
ZS9wcm9maWxlcy9jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLzkwLXJlc291cmNlcy55YW1sCnBvbHlnbG9
0UnVubmVyTWVtb3J5TGltaXQ6IDYwMDBNCnBvbHlnbG90UnVubmVyTWVtb3J5UmVxdWVzdDogMTAwME0KcG9seWdsb3
RSdW5uZXJNZW1vcnlMaW1pdFZjbzogNTYwME0KCnNlcnZlck1lbW9yeUxpbWl0OiA2RwpzZXJ2ZXJNZW1vcnlSZXF1Z
XN0OiA1RwpzZXJ2ZXJKdm1IZWFwTWF4OiA0RwoKY29udHJvbENlbnRlck1lbW9yeUxpbWl0OiAxLjVHCmNvbnRyb2xD
ZW50ZXJNZW1vcnlSZXF1ZXN0OiA3MDBtCkVPRgpjaG1vZCA2NDQgL2V0Yy92bXdhcmUtcHJlbHVkZS9wcm9maWxlcy9
jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLzkwLXJlc291cmNlcy55YW1sCg== | bash'
```

3 使用所需的内存值编辑自定义配置文件中的资源衡量指标文件。

```
vi /etc/vmware-prelude/profiles/custom-profile/helm/prelude_vco/90-resources.yaml
```

4 保存对资源衡量指标文件所做的更改，然后运行 `deploy.sh` 脚本。

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

结果

您已更改 vRealize Orchestrator 服务器的堆内存大小。

使用 Site Recovery Manager 对 vRealize Orchestrator 进行灾难恢复

您必须配置 Site Recovery Manager 为 vRealize Orchestrator 提供保护。完成 Site Recovery Manager 的常规配置任务以完善该保护。

准备环境

在开始配置 Site Recovery Manager 前，必须确保满足以下必备条件。

- 确认已将 vSphere 6.0 或更高版本安装在受保护站点和恢复站点上。
- 确认使用的是 Site Recovery Manager 8.1 或更高版本。
- 验证已配置 vRealize Orchestrator。

为 vSphere Replication 配置虚拟机

您必须为 vSphere Replication 配置虚拟机或基于阵列的复制以便使用 Site Recovery Manager。

若要在所需虚拟机上启用 vSphere Replication，请执行以下步骤。

步骤

1 在 vSphere Web Client 中，选择要在其中启动 vSphere Replication 的虚拟机并单击**操作 > 所有 vSphere Replication 操作 > 配置复制**。

2 在**复制类型**窗口中，选择**复制到 vCenter Server**，然后单击**下一步**。

3 在**目标站点**窗口中，为恢复站点选择 vCenter 并单击**下一步**。

4 在**复制服务器**窗口中，选择 vSphere Replication 服务器并单击**下一步**。

5 在**目标位置**窗口中，单击**编辑**并选择目标数据存储（用于存储复制的文件），然后单击**下一步**。

6 在**复制选项**窗口中，保留默认设置并单击**下一步**。

7 在**恢复设置**窗口中，为**恢复点对象 (RPO)** 和**时间实例中的点**输入时间，然后单击**下一步**。

8 在**即将完成**窗口中，验证设置并单击**完成**。

9 在所有要启用 vSphere Replication 的虚拟机上重复这些步骤。

创建保护组

创建保护组可以使 Site Recovery Manager 保护虚拟机。

可以通过文件夹来组织保护组。**保护组**选项卡显示保护组的名称，但不显示所放入的文件夹。如果将两个同名的保护组放在不同的文件夹中，则可能很难将其区分开。因此，请确保保护组名称在所有文件夹中是唯一的。由于在某些环境中，并非所有用户都对所有文件夹拥有查看特权，因此，为确保保护组名称的唯一性，请不要将保护组放在文件夹中。

创建保护组时，请等待以确保操作按预期完成。请确保 Site Recovery Manager 创建了保护组并且成功保护了组中的虚拟机。

前提条件

确认已执行以下任一任务：

- 已将虚拟机放入配置了基于阵列的复制的数据存储中。
- 已满足《Site Recovery Manager 管理》指南中“存储策略保护组的必备条件”的要求，并且已查看“存储策略保护组的限制”。
- 已在虚拟机上配置了 vSphere Replication。
- 已执行上述部分或全部操作。

步骤

1 在 vSphere Client 或 vSphere Web Client 中，单击 **Site Recovery > 打开 Site Recovery**。

2 在 Site Recovery 主页选项卡上，选择一个站点对，然后单击**查看详细信息**。

3 选择**保护组**选项卡，然后单击**新建**以创建保护组。

4 在“名称和方向”页面上，输入保护组的名称和描述，选择方向，然后单击**下一步**。

5 在“保护组类型”页面上，选择保护组类型，然后单击**下一步**。

选项	操作
创建基于阵列的复制保护组	选择**数据存储组 (基于阵列的复制)**，然后选择一个阵列对。
创建 vSphere Replication 保护组	选择**单个虚拟机 (vSphere Replication)**。
创建存储策略保护组	选择**存储策略 (基于阵列的复制)**。

6 选择要添加到保护组的数据存储组、虚拟机或存储策略。

选项	操作
基于阵列的复制保护组	选择数据存储组并单击**下一步**。 选择数据存储组时，该组包含的虚拟机将显示在“虚拟机”表中。
vSphere Replication 保护组	选择列表中的虚拟机，然后单击**下一步**。 只有已配置 vSphere Replication 但尚未列入保护组中的虚拟机才会显示在列表中。
存储策略保护组	从列表中选择存储策略，然后单击**下一步**。

7 在“恢复计划”页面上，可以选择将保护组添加到恢复计划。

选项	操作
添加到现有恢复计划	将保护组添加到现有恢复计划。
添加到新恢复计划	将保护组添加到新的恢复计划。如果选择此选项，则必须输入恢复计划名称。
现在不添加到恢复计划。	如果不希望将保护组添加到恢复计划，请选择此选项。

8 查看设置，然后单击**完成**。

您可以在**保护组**选项卡上监控保护组的创建进度。

- 对于基于阵列的复制保护组和 vSphere Replication 保护组，如果 Site Recovery Manager 成功将清单映射应用到受保护的虚拟机，则保护组的保护状态为*良好*。
- 对于存储策略保护组，如果 Site Recovery Manager 成功保护与存储策略关联的所有虚拟机，则保护组的保护状态为*良好*。
- 对于基于阵列的复制保护组和 vSphere Replication 保护组，如果未配置清单映射，或者如果 Site Recovery Manager 无法应用清单映射，则保护组的保护状态为*未配置*。
- 对于存储策略保护组，如果 Site Recovery Manager 无法保护与存储策略关联的所有虚拟机，则保护组的保护状态为*未配置*。

后续步骤

对于基于阵列的复制保护组和 vSphere Replication 保护组，如果保护组的保护状态为*未配置*，请将清单映射应用于虚拟机：

- 要应用站点范围的清单映射，或者要检查已设置的清单映射是否有效，请参见《Site Recovery Manager 管理》指南中的“配置清单映射”。要将这些映射应用于所有虚拟机，请参见《Site Recovery Manager 管理》指南中的“将清单映射应用于保护组的所有成员”。
- 要单独将清单映射应用于保护组中的每个虚拟机，请参见《Site Recovery Manager 管理》指南中的“为保护组中的单个虚拟机配置清单映射”。

对于存储策略保护组，如果保护组的保护状态为*未配置*，请验证是否满足《Site Recovery Manager 管理》指南中“存储策略保护组的必备条件”的要求，并查看“存储策略保护组的限制”。

创建恢复计划

创建恢复计划，以建立 Site Recovery Manager 恢复虚拟机的方式。

步骤

1. 在 vSphere Client 或 vSphere Web Client 中，单击 **Site Recovery > 打开 Site Recovery**。
2. 在 Site Recovery 主页选项卡上，选择一个站点对，然后单击**查看详细信息**。
3. 选择**恢复计划**选项卡，然后单击**新建**以创建恢复计划。
4. 输入计划的名称、描述和方向，然后选择文件夹并单击**下一步**。
5. 从菜单中选择组类型。

选项	描述
针对各个虚拟机或数据存储组的保护组	选择此选项以创建包含基于阵列的复制和 vSphere Replication 保护组的恢复计划。
存储策略保护组	选择此选项以创建包含存储策略保护组的恢复计划。 如果使用的是延伸存储，请选择此选项。

6. 为要恢复的计划选择一个或多个保护组，然后单击**下一步**。
7. 从**测试网络**下拉菜单中，选择要在测试恢复期间使用的网络，然后单击**下一步**。

   如果没有站点级别的映射，则默认选项**使用站点级别映射**将创建一个隔离测试网络。
8. 查看摘要信息，然后单击**完成**创建恢复计划。

将恢复计划整理到文件夹中

要控制不同用户或组对恢复计划的访问权限，可以将恢复计划整理到文件夹中。

如果有许多恢复计划，将其整理到文件夹中会很有用。您可以将恢复计划放置在文件夹中并为不同用户或组指定不同的文件夹许可，从而限制恢复计划的访问权限。有关如何对文件夹分配权限的信息，请参见《Site Recovery Manager 管理》指南中的“分配 Site Recovery Manager 角色和权限”。

步骤

1 在 **Site Recovery** 主页选项卡上，选择一个站点对，然后单击**查看详细信息**。

2 单击**恢复计划**选项卡，在左窗格中右键单击**恢复计划**，然后单击**新建文件夹**。

3 输入要创建的文件夹的名称，然后单击**添加**。

4 将新的或现有恢复计划添加到文件夹。

选项	描述
创建新的恢复计划	右键单击文件夹并选择**新建恢复计划**。
添加现有恢复计划	右键单击清单树中的恢复计划，然后单击**移动**。选择目标文件夹，然后单击**移动**。

编辑恢复计划

可以编辑恢复计划以更改此恢复计划创建时指定的属性。可从受保护站点或恢复站点编辑恢复计划。

步骤

1 在 vSphere Client 或 vSphere Web Client 中，单击 **Site Recovery > 打开 Site Recovery**。

2 在 **Site Recovery** 主页选项卡上，选择一个站点对，然后单击**查看详细信息**。

3 单击**恢复计划**选项卡，右键单击一个恢复计划，然后单击**编辑**。

4 （可选） 更改计划的名称或描述，然后单击**下一步**。

无法更改恢复计划的方向和位置。

5 （可选） 选择或取消选择一个或多个保护组，从而将其添加到计划或从计划中移除，然后单击**下一步**。

6 （可选） 从下拉菜单中选择恢复站点上的其他测试网络，然后单击**下一步**。

7 查看摘要信息，然后单击**完成**将指定更改应用于恢复计划。

您可以在**近期任务**视图中监控计划的更新。

设置系统属性

9

您可以设置系统属性来更改默认的 Orchestrator 行为。

本章讨论了以下主题：

设置工作流和操作对服务器文件系统的访问权限

在 vRealize Orchestrator 中，工作流和操作对特定文件系统目录的访问受限。可以通过修改 `js-io-rights.conf` 配置文件，将访问权限扩展到服务器文件系统的其他部分。

js-io-rights.conf 文件中允许对 vRealize Orchestrator 系统进行写入访问的规则

`js-io-rights.conf` 文件包含的规则允许对服务器文件系统中已定义目录拥有写入权限。

js-io-rights.conf 文件的必需内容

`js-io-rights.conf` 文件的每一行都必须包含以下信息。

- 加号 (`+`) 或减号 (`-`)，表示允许或拒绝权限
- 读取 (`r`)、写入 (`w`) 和运行 (`x`) 权限级别
- 应用这些权限的路径。

> 注 `js-io-rights.conf` 文件的根文件夹始终为 `/var/run/vco`。在 vRealize Orchestrator Appliance 文件系统中，此文件夹位于 `/data/vco/var/run/vco` 下。必须在此根文件夹下映射有权访问 vRealize Orchestrator 文件系统的所有内容。

js-io-rights.conf 文件的默认内容

Orchestrator Appliance 中 `js-io-rights.conf` 配置文件的默认内容如下：

```
-rwx /
+rwx /var/run/vco
+rx /etc/vco
-rwx /etc/vco/app-server/security/
+rx /var/log/vco/
```

默认 `js-io-rights.conf` 配置文件中的前两行允许以下访问权限：

`-rwx /`

拒绝对文件系统的所有访问权限。

`+rwx /var/run/vco`

`/var/run/vco` 目录中允许读取、写入和运行访问权限。

js-io-rights.conf 文件中的规则

vRealize Orchestrator 会按各访问权限在 `js-io-rights.conf` 文件中的显示顺序对其进行解析。每一行都可以覆盖上一行。

重要事项 您可以在 `js-io-rights.conf` 文件中设置 `+rwx /` 来允许访问文件系统的所有部分。但是，这么做会面临较高的安全风险。

设置工作流和操作对服务器文件系统的访问权限

要更改工作流和 vRealize Orchestrator API 对服务器文件系统内具体区域的访问权限，请修改 `js-io-rights.conf` 配置文件。当工作流尝试访问 vRealize Orchestrator 服务器文件系统时，会创建 `js-io-rights.conf` 文件。

步骤

1. 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。
2. 导航到 `/data/vco/var/run/vco/` 目录。
3. 在文本编辑器中打开 `js-io-rights.conf` 配置文件。
4. 将必要的命令行添加到 `js-io-rights.conf` 文件以允许或拒绝访问文件系统的相关区域。

   例如，以下行会拒绝 `/data/vco/var/run/vco/noexec` 目录中的执行权限：

   ```
   -x /data/vco/var/run/vco/noexec
   ```

   `/data/vco/var/run/vco/noexec` 会保留执行权限，但 `/data/vco/var/run/vco/noexec/bar` 不会保留。两个目录都仍然可进行读写操作。

结果

您即修改了工作流和 vRealize Orchestrator API 对文件系统的访问权限。

设置工作流和操作对操作系统命令的访问权限

vRealize Orchestrator API 提供了脚本类 (Command)，可在 vRealize Orchestrator 服务器主机操作系统中运行命令。为防止对服务器主机进行未经授权的访问，默认情况下，vRealize Orchestrator 应用程序没有 Command 类的运行权限。如果 vRealize Orchestrator 应用程序需要在主机操作系统上运行命令的权限，可以激活 Command 脚本类。

可以通过设置 vRealize Orchestrator 配置系统属性，授予 Command 类的使用权限。

步骤

1 以 **root** 用户身份登录控制中心。

2 单击**系统属性**。

3 单击**新建**。

4 在**键**文本框中，输入 **com.vmware.js.allow-local-process**。

5 在**值**文本框中，输入 **true**。

6 在**说明**文本框中，输入系统属性的说明。

7 单击**添加**。

8 在弹出菜单中单击**保存更改**。

此时系统会显示一条消息，提示您已保存成功。

9 等待 vRealize Orchestrator 服务器重新启动。

结果

您即向 vRealize Orchestrator 应用程序授权权限可在 vRealize Orchestrator 服务器主机操作系统上运行本地命令。

注 将 com.vmware.js.allow-local-process 系统属性设置为 true，您可以允许 Command 脚本类在文件系统任意位置中进行写入。此属性会覆盖您在 js-io-rights.conf 文件中针对 Command 脚本类设置的任何文件系统访问权限。在 js-io-rights.conf 文件中设置的文件系统访问权限仍会适用于 Command 以外的所有脚本类。

设置 JavaScript 对 Java 类的访问权限

默认情况下，vRealize Orchestrator 会限制 JavaScript 访问有限的一组 Java 类。如果需要 JavaScript 访问范围更广的 Java 类，必须设置 vRealize Orchestrator 系统属性。

允许 JavaScript 引擎全权访问 Java 虚拟机 (JVM) 会带来潜在的安全问题。有缺陷或恶意的脚本可能有权访问运行 vRealize Orchestrator 服务器的用户有权访问的所有系统组件。因此，默认情况下，vRealize Orchestrator JavaScript 引擎仅能访问 `java.util.*` 软件包中的类。

如果需要 JavaScript 访问除 `java.util.*` 软件包以外的类，您可在配置文件中列出允许 JavaScript 访问的 Java 软件包。随后，将 `com.vmware.scripting.rhino-class-shutter-file` 系统属性设置为指向该文件。

步骤

1 创建一个文本配置文件以存储要允许 JavaScript 访问的 Java 软件包列表。

例如，若要允许 JavaScript 访问 `java.net` 软件包中的所有类和 `java.lang.Object` 类，您可在文件中添加以下内容。

```
java.net.*
java.lang.Object
```

2 输入配置文件的名称。

3 将配置文件保存在 `/data/vco/usr/lib/vco` 的子目录中。

注 配置文件不能保存在其他目录下。

4 以 **root** 用户身份登录控制中心。

5 单击**系统属性**。

6 单击**新建**。

7 在**键**文本框中，输入 **com.vmware.scripting.rhino-class-shutter-file**。

8 在**值**文本框中，输入 `vco/usr/lib/vco/`*your_configuration_file_subdirectory*。

9 在**说明**文本框中，输入系统属性的说明。

10 单击**添加**。

11 在弹出菜单中单击**保存更改**。

此时系统会显示一条消息，提示您已保存成功。

12 等待 vRealize Orchestrator 服务器重新启动。

结果

JavaScript 引擎即有权访问指定的 JavaScript 类。

设置自定义超时属性

vCenter Server 过载时，会花费更多时间（相比默认设置 20000 毫秒）将响应返回到 vRealize Orchestrator 服务器。为防止出现此类情况，必须修改 vRealize Orchestrator 配置文件以增加默认超时时间段。

如果默认超时时间段在完成特定操作前过期，则 vRealize Orchestrator 服务器日志会包含错误。

```
Operation 'getPropertyContent' total time : '5742228' for 1823 calls, mean time : '3149.0', min time : '0', max time : '32313' Timeout, unable to get property 'info' com.vmware.vmo.plugin.vi4.model.TimeoutException
```

步骤

1 以 **root** 用户身份登录控制中心。

2 单击**系统属性**。

3 单击**新建**。

4 在**键**文本框中，输入 **com.vmware.vmo.plugin.vi4.waitUpdatesTimeout**。

5 在**值**文本框中，输入新的超时时间段（单位：毫秒）。

6 （可选） 在**说明**文本框中，输入系统属性的说明。

7 单击**添加**。

8 在弹出菜单中单击**保存更改**。

此时系统会显示一条消息，提示您已保存成功。

9 重新启动 Orchestrator 服务器。

结果

您设置的值会替换现有的 20000 秒默认超时设置。

为 vRealize Orchestrator SQL 插件添加 JDBC 连接器

本示例介绍了如何为 vRealize Orchestrator SQL 插件添加 MySQL 连接器。

步骤

1 将 MySQL connector.jar 文件添加到 vRealize Orchestrator Appliance。

a 通过 SSH 以 **root** 用户身份登录到 vRealize Orchestrator Appliance 命令行。

b 导航到 `/data/vco/var/run/vco/` 目录。

```
cd /data/vco/var/run/vco
```

c 创建 `plugins/SQL/lib/` 目录。

```
mkdir -p plugins/SQL/lib/
```

d 通过运行安全复制 (SCP) 命令，将 MySQL connector.jar 文件从本地计算机复制到 `/data/vco/var/run/vco/plugins/SQL/lib/` 目录。

```
scp ~/local_machine_dir/your_mysql_connector.jar root@orchestrator_FQDN_or_IP:/
data/vco/var/run/vco/plugins/SQL/lib/
```

注 还可以使用替代方法将 connector.jar 文件复制到 vRealize Orchestrator Appliance，例如 PSCP。

2 将新的 MySQL 属性添加到控制中心。

a 以 **root** 用户身份登录控制中心。

b 选择**系统属性**。

c 单击**新建**。

d 在**键**下，输入 **o11n.plugin.SQL.classpath**。

e 在**值**下，输入 **/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar**。

注 值文本框可以包含多个 JDBC 连接器。每个 JDBC 连接器以分号（ “;” ）分隔。例如：

```
/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar;/var/run/vco/plugins/SQL/lib/
your_mssql_connector.jar;/var/run/vco/plugins/SQL/lib/your_other_connector.jar
```

f （可选） 输入 MySQL 系统属性的说明。

g 单击**添加**，然后等待 vRealize Orchestrator 服务器重新启动。

注 请勿将 JDBC connector.jar 文件保存在其他目录，并且请勿为 `o11n.plugin.SQL.classpath` 属性设置不同的值。否则，JDBC 连接器将不可用于 vRealize Orchestrator 部署。

设置已调度任务和策略身份验证令牌续订属性

通过设置相应系统属性，管理如何实现已调度任务或策略中所用身份验证令牌的续订。

如果非管理员用户在 vRealize Orchestrator Client 中配置了没有结束时间的已调度任务，则该已调度任务工作流的身份验证令牌将在指定开始时间八小时后过期。除了已调度任务外，此身份验证令牌还用于 vRealize Orchestrator 策略。要确保 vRealize Orchestrator 部署中的已调度任务工作流或策略持续运行，您可以在控制中心中设置相应系统属性。

注 身份验证令牌将在初始开始日期 90 天后无法续订。

前提条件

确认您的 vRealize Orchestrator 部署使用 vRealize Automation 身份验证提供程序或已集成到 vRealize Automation 中。`com.vmware.o11n.auth.csp.renewTokens` 系统属性不可用于使用 vSphere 进行身份验证的 vRealize Orchestrator 部署。

步骤

1 以 **root** 用户身份登录控制中心。

2 选择**系统属性**。

3 单击**新建**。

4 在**密钥**下，输入 **`com.vmware.o11n.auth.csp.renewTokens`**。

5 在**值**下，输入 **`true`**。

> 注 对于 vRealize Automation 和 vRealize Automation Cloud 中的 vRealize Orchestrator 部署，从 vRealize Automation 启动的长时间运行的工作流会在身份验证令牌过期后将其损坏。此令牌被设为在指定开始时间八小时后过期。

6 （可选） 输入对新系统属性的说明。

7 单击**添加**，然后等待 vRealize Orchestrator 服务器重新启动。

后续操作

10

安装并配置 vRealize Orchestrator 后，可以使用 vRealize Orchestrator 自动执行与虚拟环境管理相关的频繁性重复过程。

- 登录到 vRealize Orchestrator Client，在 vCenter Server 清单对象或 vRealize Orchestrator 通过其插件访问的其他对象上运行并调度工作流。请参见《使用 VMware vRealize Orchestrator 客户端》。
- 复制并修改标准 vRealize Orchestrator 工作流并自行编写操作和工作流以在 vCenter Server 中自动处理相关操作。
- 要扩展 vRealize Orchestrator 平台的功能，请开发插件。
- 通过集成远程 Git 存储库，跨多个 vRealize Orchestrator 实例管理 vRealize Orchestrator 清单。请参见《使用 VMware vRealize Orchestrator Client》。
- 使用 vSphere Web Client 在 vSphere 清单对象上运行工作流。