Red Hat Ansible Automation Platform 2.4

自动化控制器用户指南

Automation Controller 的用户指南

Last Updated: 2024-06-29

Red Hat Ansible Automation Platform 2.4 自动化控制器用户指南

Automation Controller 的用户指南

法律通告

摘要

本指南介绍了如何使用自动化控制器来定义、操作、调用和委派自动化。

目录

前言

感谢您对 Red Hat Ansible Automation Platform 自动化控制器的关注。自动化控制器通过增加控制、知识、协调基于 Ansible 的环境，帮助团队管理复杂的多层部署。

Automation controller 用户指南描述了自动化控制器中所有可用功能。它假定读者对 Ansible 有一定的了解，包括 playbook、变量和标签等概念。有关这些和其他 Ansible 概念的更多信息，请参阅 Ansible 文档。

对红帽文档提供反馈

如果您对本文档有任何改进建议，或发现了任何错误，请通过 https://access.redhat.com 联系技术支持，以使用 **docs-product** 组件在 Ansible Automation Platform JIRA 项目中创建一个问题。

第 1 章 自动化控制器概述

通过一个机构中的 Ansible Automation Platform 用户，可以通过简单、强大和无代理的技术实现共享、检查和管理自动化内容。IT 经理可以为各个团队提供如何应用自动化的指导。自动化开发人员可以编写使用现有知识的任务，而无需执行符合复杂工具和框架的操作开销。它是部署端到端自动化解决方案（从混合云到边缘）的更安全、稳定的基础。

Ansible Automation Platform 包括自动化控制器，允许用户在整个企业中定义、操作、扩展和委派自动化。

1.1. 实时 PLAYBOOK 输出和探索

自动化控制器允许您实时监控 playbook 运行，并在每个主机签入时查看它们。您可以返回并探索特定任务和主机的结果非常详细，搜索特定 play 或主机，仅查看这些结果，或者查找需要更正的错误。

1.2. "PUSH BUTTON"自动化

自动化控制器可让您访问您首选的项目，并从 Web 界面重新触发执行。自动化控制器要求输入变量，提示输入您的凭证、启动和监控作业，并显示结果和主机历史记录。

1.3. 简化的基于角色的访问控制和审计

自动化控制器可让您：

- 通过 *基于角色的访问控制* (RBAC)为不同的团队或显式用户授予权限。示例任务包括查看、创建或修改文件。
- 将一些项目保持私有，同时允许某些用户编辑清单，而其他用户可以在检查(dry run)或实时模式中针对特定系统运行 playbook。
- 启用特定用户使用凭证，而不向他们公开凭证。

自动化控制器记录了操作历史记录以及谁启动，包括编辑的对象和作业启动。

如果要授予任何用户或团队使用作业模板的权限，您可以直接在作业模板上分配权限。凭证是自动化控制器 RBAC 系统中的完整对象，并可分配给多个用户或团队供使用。

自动化控制器包括 *审核员* 类型。系统级审核员可以查看系统自动化的所有方面，但没有权限运行或更改自动化。审核员对于从 REST API 中提取自动化信息的服务帐户很有用。

其他资源

- 有关用户角色的更多信息，请参阅基于角色的访问控制。

1.4. 云和自动扩展灵活性

自动化控制器包含一个强大的可选置备回调功能，它允许节点按需请求配置。这是云自动扩展方案的理想解决方案，包括以下功能：

- 它与 Cobbler 等配置服务器集成，并处理受管系统无法预计的运行时间。
- 它要求不会在远程节点上安装管理软件。

- 回调解决方案可以通过调用 **curl** 或 **wget** 触发，并可嵌入到 初始化脚本、Kickstart 或 preseeds 中。
- 您可以控制访问，以便只有清单中列出的机器才能请求配置。

1.5. 理想的 RESTFUL API

自动化控制器 REST API 是系统管理应用程序的理想 RESTful API，所有资源都完全可发现、分页、可搜索并良好建模。风格的 API 浏览器启用了 API 根的 API 探索，地址为 **http://<server name>/api/**，显示每个资源和关系。用户界面中可在 API 中完成的所有操作。

1.6. 备份和恢复

Ansible Automation Platform 可以备份和恢复系统或系统，您可以根据需要轻松备份和复制实例。

1.7. ANSIBLE GALAXY 集成

通过在项目目录中包含 Ansible Galaxy **requirements.yml** 文件，自动化控制器会自动从 Galaxy、GitHub 或您的本地源控制获取 playbook 所需的角色。有关更多信息，请参阅 Ansible Galaxy 支持 。

1.8. OPENSTACK 的清单支持

OpenStack 支持动态清单。这可让您以 OpenStack 云中运行的任何虚拟机或镜像为目标。

如需更多信息，请参阅 Openstack。

1.9. 远程命令执行

使用远程命令执行来执行简单的任务，如添加单个用户、更新单个安全漏洞或重启失败的服务。您可以描述为单个 Ansible play 的任何任务都可以在清单中的主机或一组主机上运行，以便您快速轻松地管理系统。由于 RBAC 引擎和详细的审计日志记录，您知道哪个用户已完成特定的任务。

1.10. 系统跟踪

您可以使用事实缓存功能收集事实。如需更多信息，请参阅 事实缓存。

1.11. 集成通知

跟踪自动化的状态。

您可以配置以下通知：

- 作业模板、项目或整个机构的可堆栈通知
- 作业启动、作业成功、作业失败和作业批准的不同通知（用于工作流节点）

支持以下通知源：

- 电子邮件
- Grafana
- IRC

- Mattermost
- PagerDuty
- Rocket.Chat
- Slack
- Twilio
- Webhook （发布到任意 Webhook，用于集成到其他工具）

您还可以为每个上述通知类型自定义通知消息。

1.12. 集成

自动化控制器支持以下集成：

- Red Hat Satellite 6 的动态清单源。

如需更多信息，请参阅 Red Hat Satellite 6 。

- Red Hat Insights 集成，启用 Insights playbook 以用作 Ansible Automation Platform 项目。

如需更多信息，请参阅设置 Insights 修复。

- Automation hub 充当自动化控制器的内容供应商，需要自动化控制器部署和自动化中心部署一起运行。

1.13. 自定义虚拟环境

自定义 Ansible 环境支持可让您有不同的 Ansible 环境，并为不同的团队和作业指定自定义路径。

1.14. 身份验证增强

自动化控制器支持：

- LDAP
- SAML
- 基于令牌的身份验证

LDAP 和 SAML 支持可让您以更灵活的方式集成企业帐户信息。

基于令牌的身份验证允许通过集成的 OAuth 2 令牌支持通过自动化控制器验证第三方工具和服务。

1.15. 集群管理

集群组的运行时管理支持可配置的扩展。

1.16. 工作流增强

要对复杂置备、部署和编配工作流建模，您可以使用自动化控制器以多种方式扩展工作流：

- 工作流的清单覆盖您可以在工作流定义时间或启动时覆盖工作流中的清单。自动化控制器允许您定义应用程序部署工作流，然后在多个环境中重复使用它们。
- 在对复杂进程建模时，工作流的聚合节点 有时会等待多个步骤完成，然后才能继续。自动化控制器工作流可以复制它；工作流步骤可以等待任意数量的之前的工作流步骤正确完成，然后再继续。
- 工作流嵌套，您可以重新使用单独的工作流作为更大工作流的组件。示例包括将置备和应用程序部署工作流合并到单个工作流中。
- 工作流暂停和批准 您可以构建包含需要用户干预的批准节点的工作流。这样便可暂停 playbook 之间的工作流，以便用户可以提供批准（或拒绝）以继续工作流中的下一步。

如需更多信息，请参阅自动化控制器中的工作流

1.17. 作业分发

获取在数千台机器中运行的事实收集或配置作业，并将其划分为可在自动化控制器集群中分发的分片，以提高可靠性、作业完成速度并改进了集群使用。

例如，您可以大规模更改 15,000 个交换机的参数，或者在多节点 RHEL estate 中收集信息。

如需更多信息，请参阅 作业分片。

1.18. 支持在启用了 FIPS 的环境中部署

自动化控制器以受限模式（如 FIPS）部署和运行。

1.19. 限制每个机构中的主机数量

许多大型机构在很多机构间共享实例。为确保一个机构无法使用所有许可的主机，此功能使超级用户能够为每个机构分配多少许可主机设置指定的上限。自动化控制器算法因素更改机构的限制以及所有机构中的主机总数。如果清单同步使机构不符合该策略，清单更新会失败。此外，超级用户还可以通过警告来过度分配其许可证。

1.20. 清单插件

以下清单插件从上游集合使用：

- **amazon.aws.aws_ec2**
- **community.vmware.vmware_vm_inventory**
- **azure.azcollection.azure_rm**
- **google.cloud.gcp_compute**
- **theforeman.foreman.foreman**
- **openstack.cloud.openstack**
- **ovirt.ovirt.ovirt**
- **awx.awx.tower**

1.21. SECRET 管理系统

使用 secret 管理系统，外部凭证会存储，并可用于自动化控制器，因此您不需要直接提供这些凭证。

第 2 章 自动化控制器许可、更新和支持

自动化控制器作为年度 Red Hat Ansible Automation Platform 订阅的一部分提供。

Ansible 是一个开源软件项目，它使用 GNU General Public License version 3 的许可证，如 Ansible源代码所述。

在安装 Ansible Automation Platform 前，您必须附加有效的订阅。

如需更多信息，请参阅 附加订阅。

2.1. 试用和评估

您需要许可证来运行自动化控制器。您可以使用免费试用许可证开始。

- Ansible Automation Platform 的试用许可证位于 ：http://ansible.com/license
- 试用许可证或在试用自动化控制器软件期间不提供支持。

2.2. 组件许可证

要查看自动化控制器中包含的组件的许可证信息，请参阅 **/usr/share/doc/automation-controller-<version>/README**。

其中 **<**version> 是指已安装的自动化控制器版本。

要查看特定的许可证，请参阅 **/usr/share/doc/automation-controller-<version> done.txt**。

其中 * 是您引用的许可证文件名。

2.3. 许可证中的节点数

自动化控制器许可证定义了可作为 Red Hat Ansible Automation Platform 订阅进行管理的受管节点的数量。

一个典型的许可证显示"License Count: 500"，它把受管节点的最大数量设置为 500。

有关许可证的受管节点要求的更多信息，请参阅 https://access.redhat.com/articles/3331481。

注意

Ansible 不会回收节点数或重置自动化主机。

第 3 章 安装后登录到自动化控制器

安装自动化控制器后，您必须登录。

流程

1. 安装完成后提供的登录信息，打开 Web 浏览器，并通过导航到其服务器 URL 登录到自动化控制器，地址为 https://<CONTROLLER_SERVER_NAME>/
2. 使用安装过程中指定的凭证登录：
   - 默认用户名是 **admin**。
   - **admin** 的密码是指定的值。
3. 点所需用户旁的 **More Actions** 图标。
4. 点 **Edit**。
5. 编辑所需详情并点 **Save**。

第 4 章 管理 ANSIBLE 自动化控制器订阅

在使用自动化控制器前，您必须有一个有效的订阅，该订阅会授权其使用。

4.1. 订阅类型

Red Hat Ansible Automation Platform 以年度订阅的形式提供不同级别的支持，以及机器数量。

- Standard:
  - 管理任意规模的环境
  - 企业级 8x5 支持和 SLA
  - 包括维护和升级
  - 请参阅 产品支持条款中的SLA
  - 参阅红帽支持严重性级别定义
- 高级：
  - 管理任意规模的环境，包括关键任务环境
  - 高级 24x7 支持和 SLA
  - 包括维护和升级
  - 请参阅 产品支持条款中的SLA
  - 参阅红帽支持严重性级别定义

所有订阅级别包括常规的自动化控制器更新和发布，Ansible 以及平台的其他组件。

如需更多信息，请通过红帽客户门户网站或 http://www.ansible.com/contact-us/ 联络 Ansible。

4.2. 获取授权的 ANSIBLE 自动化控制器订阅

如果您已经订阅了红帽产品，则可通过该订阅获取自动化控制器订阅。如果您没有 Red Hat Ansible Automation Platform 和 Red Hat Satellite 订阅，您可以请求一个试用订阅。

流程

- 如果您有 Red Hat Ansible Automation Platform 订阅，请在启动自动化控制器时使用您的红帽客户凭证来访问您的订阅信息。请参阅导入订阅。
- 如果您有非 Ansible Red Hat 或 Satellite 订阅，使用以下方法之一访问自动化控制器：
  - 在许可证页面中输入您的用户名和密码。
  - 从红帽客户门户 的订阅分配页面中获取订阅清单。如需更多信息，请参阅获取订阅清单。
  - 如果您没有 Red Hat Ansible Automation Platform 订阅，请访问 Try Red Hat Ansible Automation Platform 并请求试用订阅。

其他资源

- 要了解您的订阅支持的内容，请参阅 自动化控制器许可证、更新和支持。
- 如果您的订阅有问题，请联络您销售账户经理或红帽客户服务 ： https://access.redhat.com/support/contact/customerService/。

4.3. 获取一个订阅清单

要上传订阅清单，首先设置您的订阅分配：

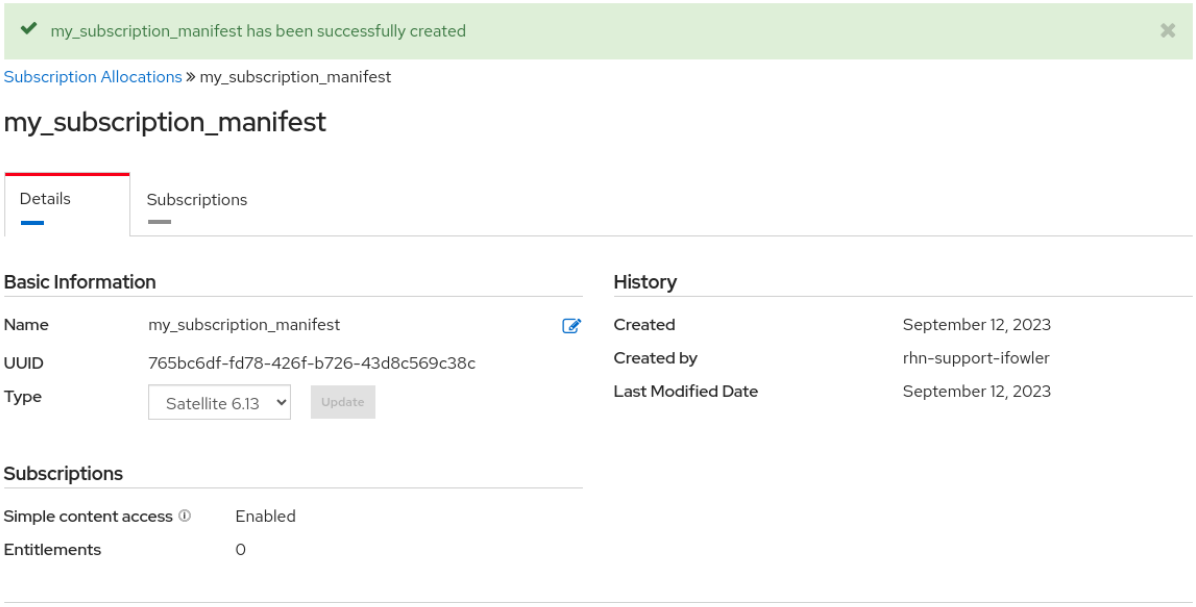
流程

1. 进入 https://access.redhat.com/management/subscription_allocations。在创建前，订阅分配 页面不包含任何订阅。
2. 点 **Create New subscription allocation**。

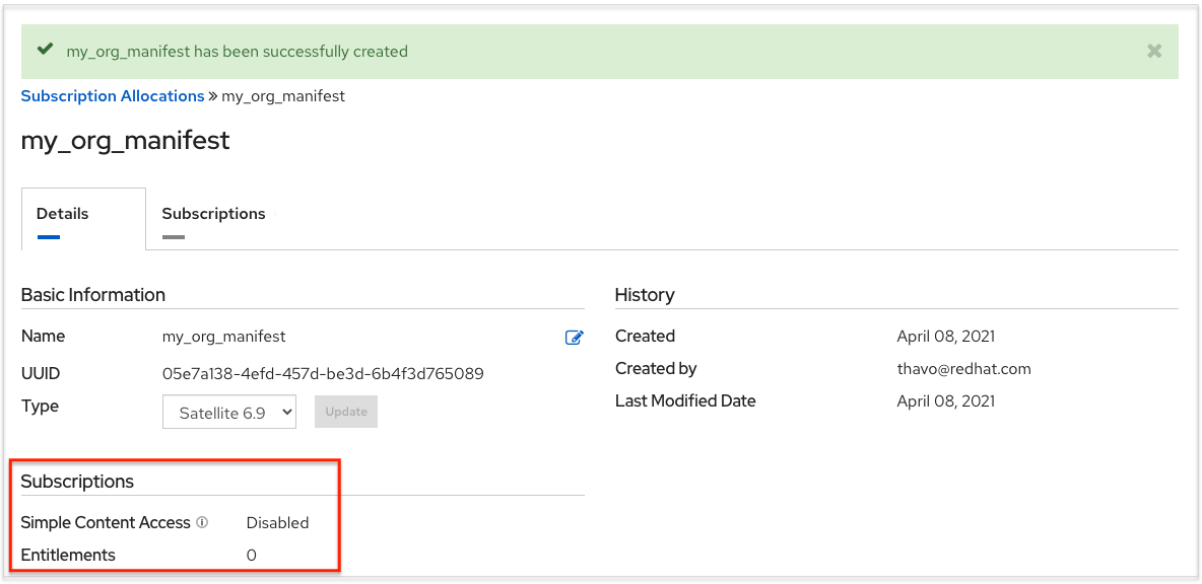
注意

如果没有显示 **Create New subscription allocation**，或者被禁用，则您没有创建订阅分配的适当权限。要创建订阅分配，您必须是客户门户网站中的管理员，或者具有 **Manage Your Subscriptions** 角色。请联络一个 **access.redhat.com** 管理员，或可以授予您管理订阅权限的机构管理员。

3. 输入您的订阅的名称，然后从 **Type** 下拉菜单中选择 **6.15**。



4. 点 **Create**。
当成功创建订阅清单时，权利 旁边显示的数字表示与您的订阅关联的权利数。



4.3.1. 设置订阅清单

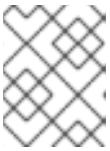
要获取订阅清单，您必须通过 Subscriptions 选项卡在您的订阅中添加权利。

流程

1. 点击 **Subscriptions** 选项卡。
2. 如果没有可以显示的订阅，请单击 **Add Subscriptions**。
3. 以下屏幕允许您选择并添加权利来放入清单文件中。

   您可以在订阅分配中选择多个 Ansible Automation Platform 订阅。有效的 Ansible Automation Platform 订阅通常遵循名称 *"Red Hat Ansible Automation..."*。

4. 指定要放入清单文件中的授权或受管节点数量。这可让您分割订阅，例如：开发集群中的 400 个节点，以及生产环境集群的 600 个节点，超过 1000 个节点订阅。



注意

您可以通过将同一类型的多个订阅添加到清单文件并上传它们，在单一安装中应用多个订阅。同样，您只能在创建清单时分配一部分订阅来应用订阅的子集。

5. 点 **Submit**。
   成功添加后，您指定的分配会在 Subscriptions 选项卡中显示。
6. 单击 Details 选项卡，以访问订阅清单文件。
7. 点 **Export Manifest** 以导出此订阅的清单文件。预先带有 **manifest_** 的文件夹会下载到您的本地驱动器中。聚合具有相同 SKU 的多个订阅。
8. 当您有订阅清单时，进入 Subscription 屏幕。

9. 点 **Browse** 上传整个清单文件。
10. 导航到保存文件的位置。不要打开它，或者上传它的各个部分。

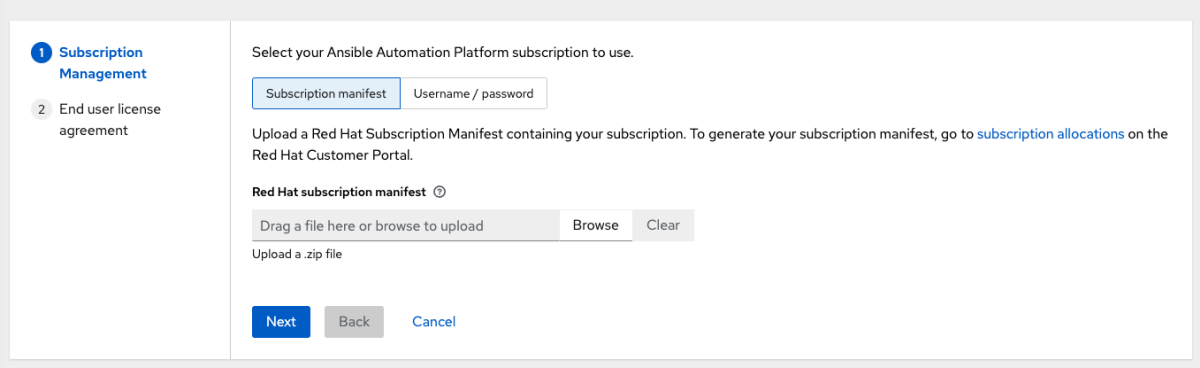
4.4. 导入订阅

获得授权 Ansible Automation Platform 订阅后，您必须在使用自动化控制器前将其导入到自动化控制器系统中。

- 您已获取了一个订阅清单。如需更多信息，请参阅获取订阅清单。

流程

1. 第一次启动自动化控制器。显示 订阅管理 屏幕。



2. 通过完成以下步骤之一检索并导入您的订阅：

   a. 如果您获取了订阅清单，请进入保存该文件的位置来上传该订阅清单。订阅清单是完整的 **.zip** 文件，不仅是其组件部分。

   注意

   如果 Subscription manifest选项中的Browse 选项被禁用，请清除 用户名和密码字段 以启用它。

   然后，订阅元数据从 RHSM/Satellite API 或提供的清单中检索。如果在单个安装中应用了多个订阅计数，自动化控制器会合并计数，但使用最早的过期日期作为过期日期（此时您必须刷新您的订阅）。

   b. 如果您使用您的红帽客户凭证，请在许可证页面中输入您的用户名和密码。如果您的自动化控制器集群节点使用 Subscription Manager 注册到 Satellite，请使用您的 Satellite 用户名或密码。输入凭证后，点 **Get Subscriptions**。
   自动化控制器检索您配置的订阅服务。然后，它会提示您选择要运行的订阅，并将该元数据应用到自动化控制器。您可以随着时间的推移登录，并在续订后检索新的订阅。

3. 点 **Next** 进入 Tracking 和 Insights 页面。
   跟踪和分析收集数据，以帮助红帽改进产品并提供更好的用户体验。有关数据收集的更多信息，请参阅 *自动化控制器管理指南中的*Usability Analytics 和 Data Collection。

   默认启用这个选项，但您可以选择不使用以下任一选项：

   - 用户分析.从控制器 UI 收集数据。

- 智能分析工具分析.通过自动化控制器提供对自动化的高级别分析。它可帮助您识别控制器的趋势和异常使用。为了选择使用自动化分析，您的自动化控制器实例必须在 Red Hat Enterprise Linux 上运行。如需更多信息，请参阅 Automation Analytics 部分。

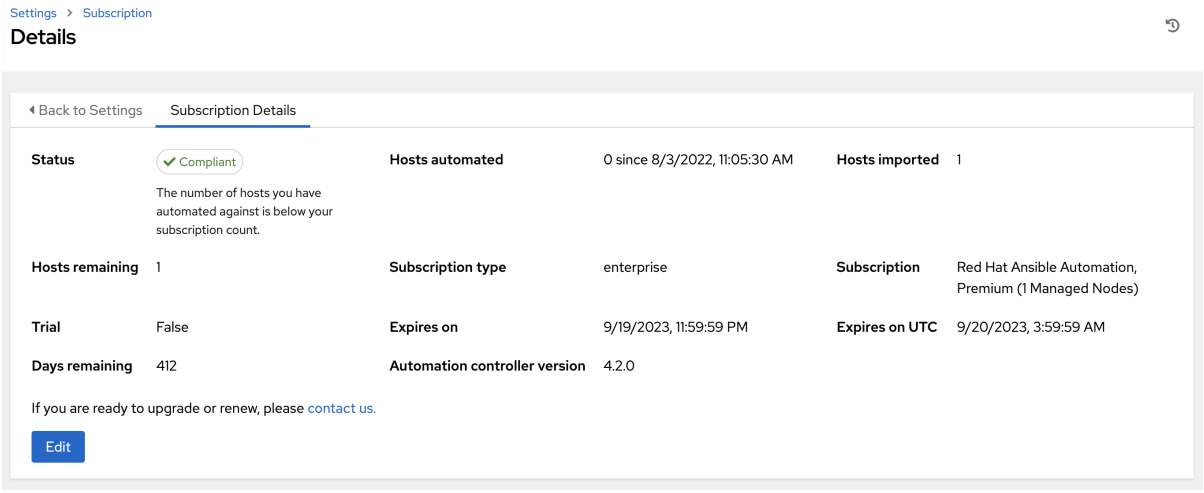
注意

您可以随时更改您的分析数据收集设置。

4. 指定跟踪和 Insights 首选项后，点 **Next** 进入 End User Agreement。

5. 检查并选中 I agree to the End User License Agreement复选框，然后单击 **Submit**。接受订阅后，自动化控制器会显示订阅详情并打开 Dashboard。要从 Dashboard 返回 Subscription 设置屏幕，请从导航面板中的 选项中选择+Settings → Subscription settings。

6. 可选： 要从 Dashboard 返回 Subscription 设置屏幕，请在导航面板中选择 Settings → Subscription settings 选项。



您的订阅故障排除

当您的订阅过期时（您可以在订阅设置窗口的订阅详情中查看它），您必须在自动化控制器中更新它。您可以通过导入新订阅或设置新订阅来完成此操作。

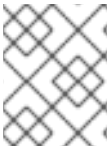
如果您满足"错误获取许可证"消息，请检查您是否具有 Satellite 用户所需的正确权限。自动化控制器管理员需要应用订阅。

Satellite 用户名和密码用于查询 Satellite API 现有订阅。在 Satellite API 中，自动化控制器会接收有关这些订阅的元数据，然后通过 过滤以查找您可应用的有效订阅。这些随后在 UI 中显示为有效的订阅选项。

以下 Satellite 角色授予适当的访问权限：

- 使用 **view_subscriptions** 和 **view_organizations** 过滤器进行自定义
- Viewer
- Administrator
- Organization Administrator（机构管理员）
- Manager（管理者）

将 Custom 角色用于自动化控制器集成，因为它是最严格的。如需更多信息，请参阅有关管理用户和角色的 Satellite 文档。



注意

System Administrator 角色不等于 Administrator 用户复选框，不提供足够的权限来访问订阅 API 页面。

4.5. 手动添加订阅

如果您无法使用自动化控制器用户界面应用或更新订阅信息，您可以在 Ansible playbook 中手动上传订阅清单。

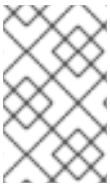
使用 **ansible.controller** 集合中的 license 模块：

```
- name: Set the license using a file
  license:
  manifest: "/tmp/my_manifest.zip"
```

如需更多信息，请参阅 Automation controller license 模块。

4.6. 附加订阅

在安装 Ansible Automation Platform 前，您必须 附加有效的 Ansible Automation Platform 订阅。



注意

如果您的红帽帐户启用了 简单内容访问模式，则不需要附加订阅。但是，在安装 Ansible Automation Platform 前，您必须注册到 *Red Hat Subscription Management*(RHSM)或 Red Hat Satellite。

流程

1. 要查找订阅的 **pool_id**，请输入以下命令：

```
# subscription-manager list --available --all | grep "Ansible Automation Platform" -B 3 -A 6
```

该命令返回以下内容：

```
Subscription Name: Red Hat Ansible Automation Platform, Premium (5000 Managed Nodes)
Provides: Red Hat Ansible Engine
Red Hat Single Sign-On
Red Hat Ansible Automation Platform
SKU: MCT3695
Contract: ********
Pool ID: ********************
Provides Management: No
Available: 4999
Suggested: 1
```

2. 要附加这个订阅，请输入以下命令：

```
# subscription-manager attach --pool=<pool_id>
```

如果所有节点都附加了，则找到软件仓库。

3. 要检查订阅是否已成功附加，请输入以下命令：

```
# subscription-manager list --consumed
```

4. 要删除此订阅，请输入以下命令：

```
#subscription-manager remove --pool=<pool_id>
```

4.7. 故障排除：保持您的订阅合规

您的订阅有两个可能的状态：

- 合规 ：指示您的订阅适用于您在订阅数中自动的主机数量。
- 不合规 ： 表示您已超过订阅中的主机数量。

合规计算如下：

```
managed > manifest_limit    =>  non-compliant
managed =< manifest_limit   =>  compliant
```

其中： **managed** 是没有删除删除的唯一受管主机数量，**inventory_limit** 是订阅清单中的受管主机数量。

显示的其他重要信息有：

- 主机自动: 作业自动化的主机计数，消耗许可证计数。
- 导入的主机 ：在所有清单源中考虑唯一主机名的主机数.这个数字不会影响剩余的主机。
- 主机剩余 ：主机总数减去主机自动化。
- 主机删除: 删除的主机，释放许可证容量。
- 活跃的 主机之前删除: 现在活跃的主机以前已被删除。

例如，如果您的订阅容量为 10 个主机：

- 从 9 个主机开始，添加了 2 个主机，并删除 3 个主机，您现在可以有 8 个主机（合规）。
- 3 个主机再次自动化，您现在有 11 个主机，可使您超过 10 个订阅限制（不合规）。
- 如果您删除主机，请刷新订阅详情以查看计数和状态的更改。

4.8. 查看主机活动

流程

1. 在导航面板中，选择 Host Metrics 来查看与主机关联的活动，如已自动和删除的那些活动。
每个唯一的主机名都会被列出，并根据用户的首选项进行排序。

Host Metrics

Hostname | Delete | 1 - 1 of 1

Hostname	First automated	Last automated	Automation	Deleted
host-1	4/18/2023, 8:08:41 AM	4/18/2023, 8:08:41 AM	1	0
host-2	4/18/2023, 8:08:41 AM	4/18/2023, 8:08:41 AM	1	0
host-3	4/18/2023, 8:08:41 AM	4/18/2023, 8:08:41 AM	1	0
host-4	4/18/2023, 8:08:41 AM	4/18/2023, 8:08:41 AM	1	0
host-5	4/18/2023, 8:08:41 AM	4/18/2023, 8:08:41 AM	1	0

注意

调度的任务会每周自动更新这些值，并使用上一次自动化超过一年的主机删除作业。

2. 通过选择所需的主机并单击 **Delete**，直接从 Host Metrics 视图中删除不必要的主机。这些是软删除的，这意味着它们的记录不会被删除，但不使用，因此不会计算您的订阅。

4.9. 主机指标工具

自动化控制器提供了一种通过命令行界面(CLI)生成主机指标数据和主机指标概述的 CSV 输出的方法。您还可以通过 API 批量删除主机。

4.9.1. awx-manage 工具

awx-manage 工具支持以下选项：

```
awx-manage host_metric --csv
```

此命令生成主机指标数据、主机指标摘要文件和集群信息文件。要将所有文件打包在一个 tar 包中，以进行分发和共享，请使用：

```
awx-manage host_metric --tarball
```

指定要输出到每个文件的行数(<n>)：

```
awx-manage host_metric --tarball --rows_per_file <n>
```

以下是配置文件的示例：

/tmp/_dc~~~/1894481089/config.json

File Edit View Encoding About

{"platform": {"system": "Linux", "dist": ["CentOS Stream", "9", ""], "release": "6.2.7-200.fc37.x86_64", "type": "traditional"}, "install_uuid": "576168fc-ec61-4333-a985-a66
"license_expiry": 119126884, "pendo_tracking": "off", "authentication_backends": ["awx.sso.backends.TACACSPlusBackend", "awx.main.backends.AWXModelBackend"], "logging_aggreg

1/1 1.3 K (100 %) Encoding: UTF-8 /tmp/_dc~~~/1894481089/config.json

自动化分析接收并使用 JSON 文件。

4.9.2. API 端点功能

API 仅列出非删除记录，可通过 **last_automation** 和 **used_in_inventories** 列进行排序。

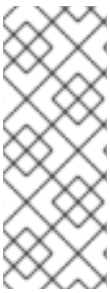
您可以使用主机指标 API 端点**api/v2/host_metric** 到 soft delete 主机：

```
api/v2/host_metric <n> DELETE
```

一个每月的调度任务会自动删除使用 Host Metric 表中主机（在一年前一次自动自动）的作业。

第 5 章 用户界面

自动化控制器 *用户界面* (UI)为您的 IT 编配要求提供图形框架。导航面板提供对自动化控制器资源的快速访问，如项目、清单、作业模板 和 Jobs。



注意

自动化控制器 UI 也作为技术预览提供，可能在以后的版本中有所变化。要预览新的 UI，请点击 Settings 菜单的 Miscellaneous System 选项中的 Enable Preview of New User Interface switch to On。

保存后，退出并重新登录，以从预览横幅访问新 UI。要返回当前的 UI，请点击指示器上的链接。

使用页面标头中的图标访问您的用户配置文件、About 页面、查看相关文档或注销。

您可以点击 活动流 图标查看该用户的活动流。

5.1. 视图

自动化控制器 UI 提供多个选项来查看信息。

- Dashboard 视图
- Jobs 视图
- Schedules 视图
- 活动流
- 工作流批准
- 主机指标

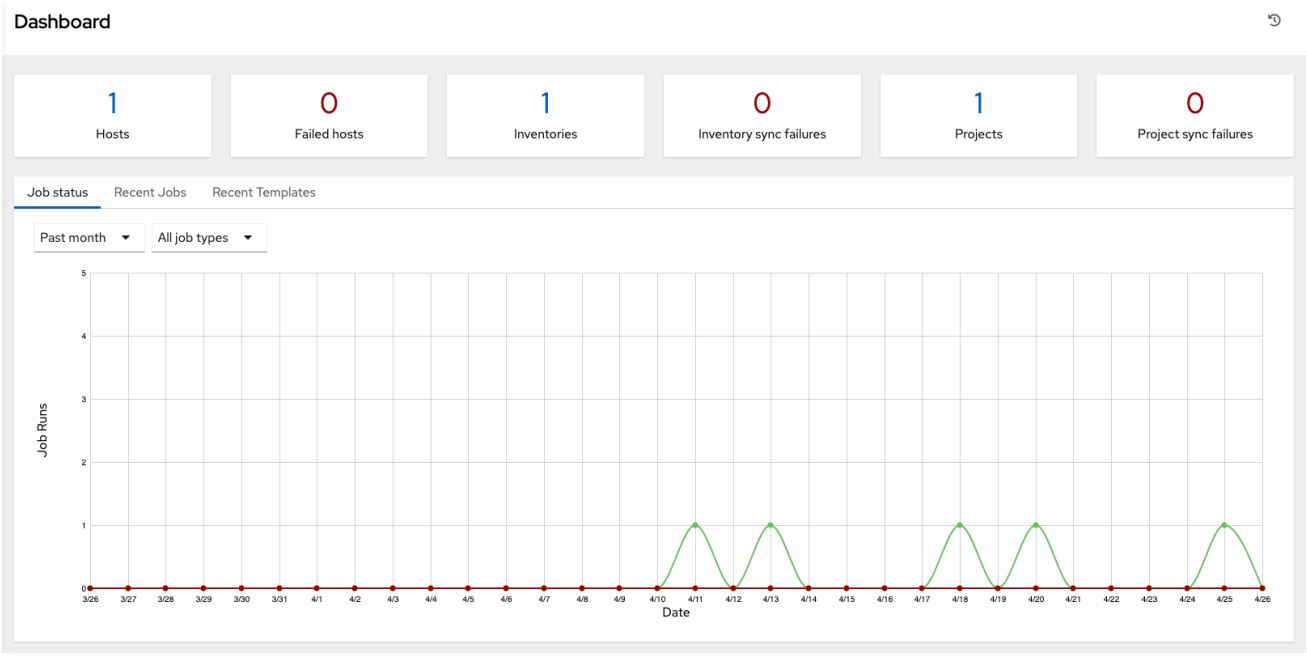
5.1.1. 仪表板视图

使用导航菜单完成以下任务：

- 显示不同的视图
- 进入您的资源
- 授予用户访问权限
- 在 UI 中管理自动化控制器功能

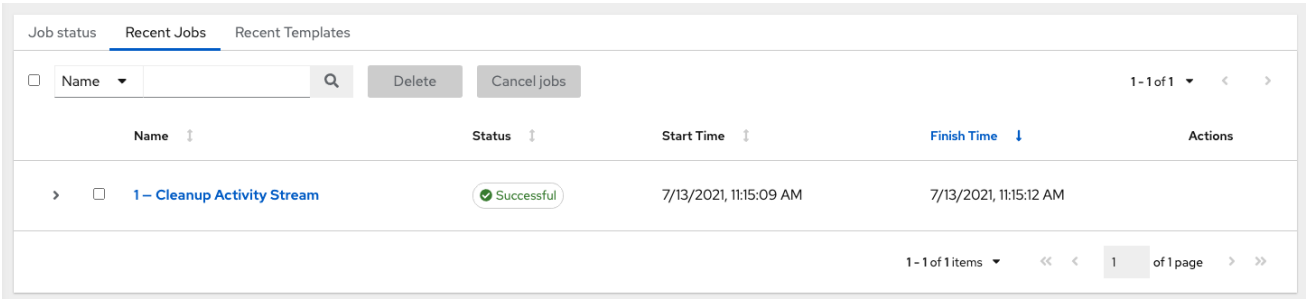
流程

- 在导航面板中，选择 Views 来隐藏或显示 Views 选项。
- 控制面板显示当前 作业状态 的摘要。
  - 您可以在一段时间内或按作业类型过滤作业状态。

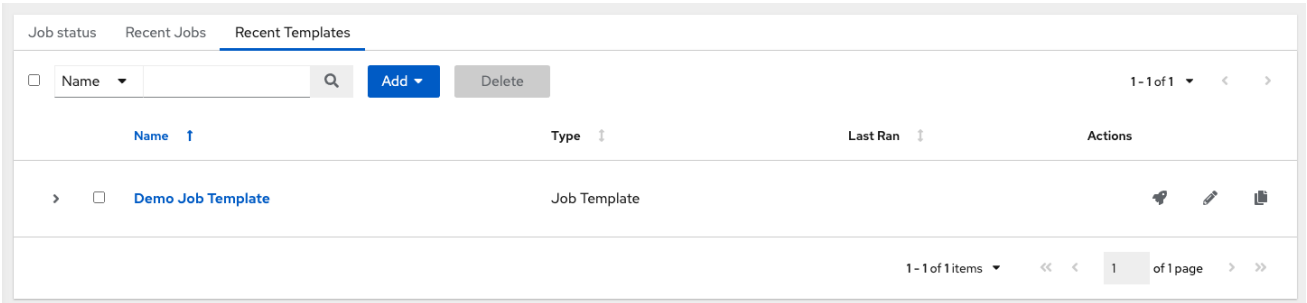


- 您还可以显示 最近作业和 最近 模板 摘要。

Recent Jobs 选项卡显示哪些作业是最近运行的、它们的状态以及运行的时间。



Recent Templates 选项卡显示最近使用的模板摘要。您还可以通过从导航面板中选择 Resources → Templates 来访问此摘要。

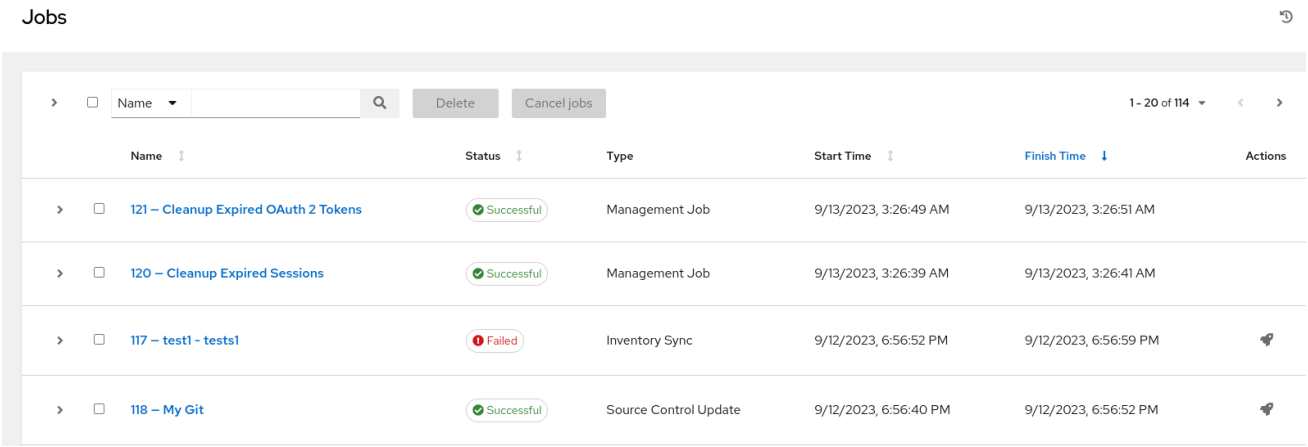


注意

点导航面板中的 Views → Dashboard，或者任何时候 Ansible Automation Platform 徽标返回仪表板。

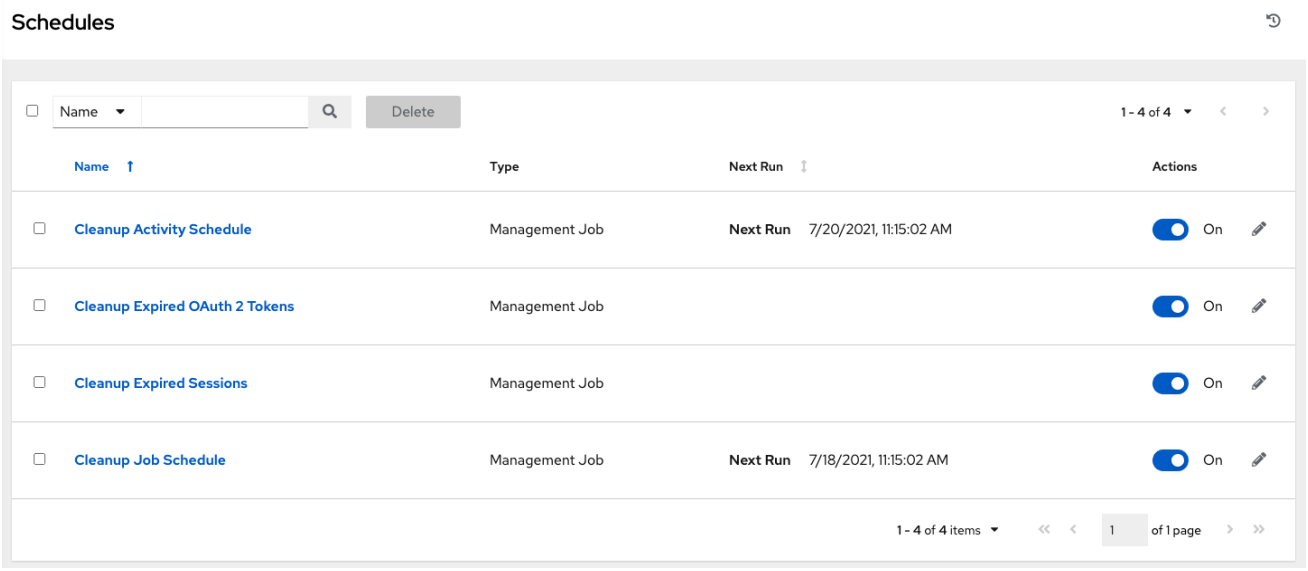
5.1.2. Jobs 视图

- 在导航面板中，选择 Views → Jobs。此视图显示已运行的作业，包括项目、模板、管理作业、SCM 更新和 playbook 运行。



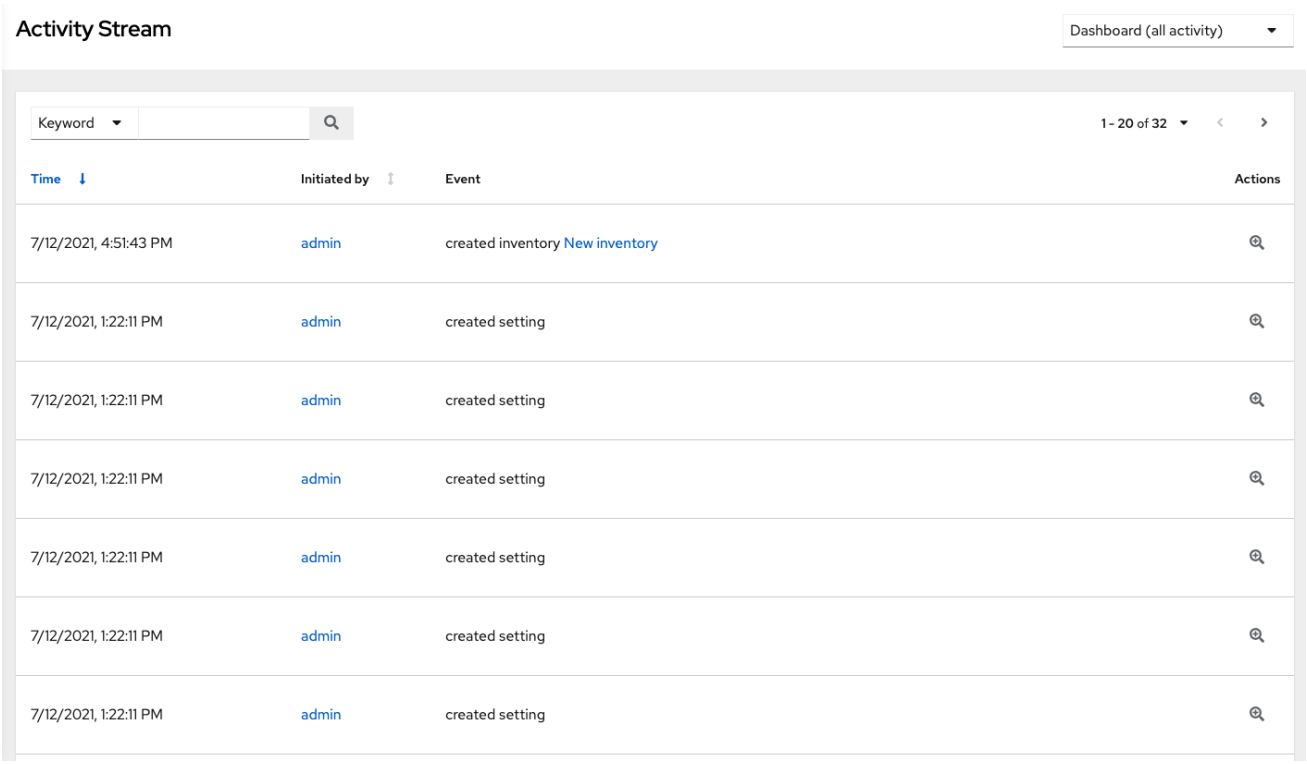
5.1.3. Schedules 视图

在导航面板中，选择 **Views → Schedules**。此视图显示所有配置的调度作业。

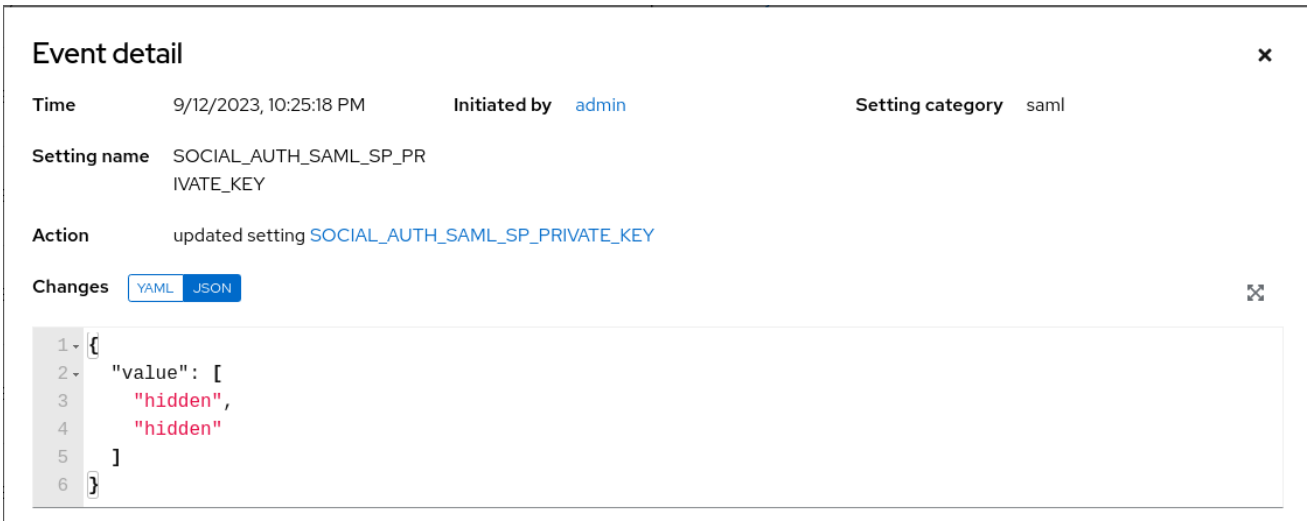


5.1.4. 活动流

- 在导航面板中，选择 **Views → Activity Stream** 以显示 **Activity Streams**。大多数屏幕都有一个活动流 图标。



活动流显示特定对象的所有更改。对于每个更改，活动流会显示事件的时间、启动事件的用户以及操作。显示的信息因事件类型而异。点 检查 图标显示更改的事件日志。



您可以通过启动用户、系统启动（如果是系统启动）或任何相关的对象（如凭证、作业模板或调度）过滤活动流。

主仪表板上的活动流显示整个实例的活动流。大多数页面都允许查看针对该特定对象过滤的活动流。

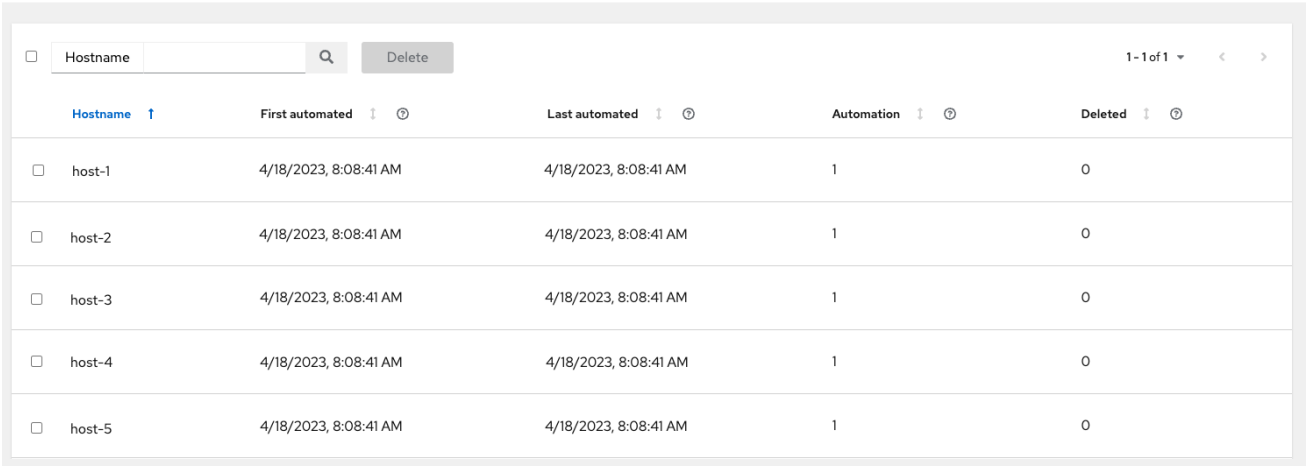
5.1.5. 工作流批准

- 在导航面板中，选择 **Views → Workflow Approvals** 来查看您的工作流批准队列。列表中包含需要您批准或拒绝在作业继续操作前的操作。

5.1.6. 主机指标

- 在导航面板中，选择 **Host Metrics** 来查看与主机关联的活动，其中包括已自动、清单中使用的计数以及删除的主机。

Host Metrics



如需更多信息，请参阅 故障排除：在合规中保持您的订阅。

5.2. 资源菜单

Resources 菜单提供对自动化控制器的以下组件的访问：

- 模板
- 凭证
- 项目
- 清单
- 主机

5.3. 访问菜单

Access 菜单允许您配置谁具有自动化控制器资源的权限：

- 机构
- 用户
- 团队

5.4. 管理

管理 菜单提供对自动化控制器的管理选项的访问。在这里，您可以创建、查看和编辑：

- 凭证类型
- 通知
- Management_jobs
- 实例组
- 实例

- 应用程序
- 执行环境
- 拓扑视图

5.5. SETTINGS 菜单

使用 Settings 菜单配置全局和系统级设置。Settings 菜单提供对自动化控制器配置设置的访问。

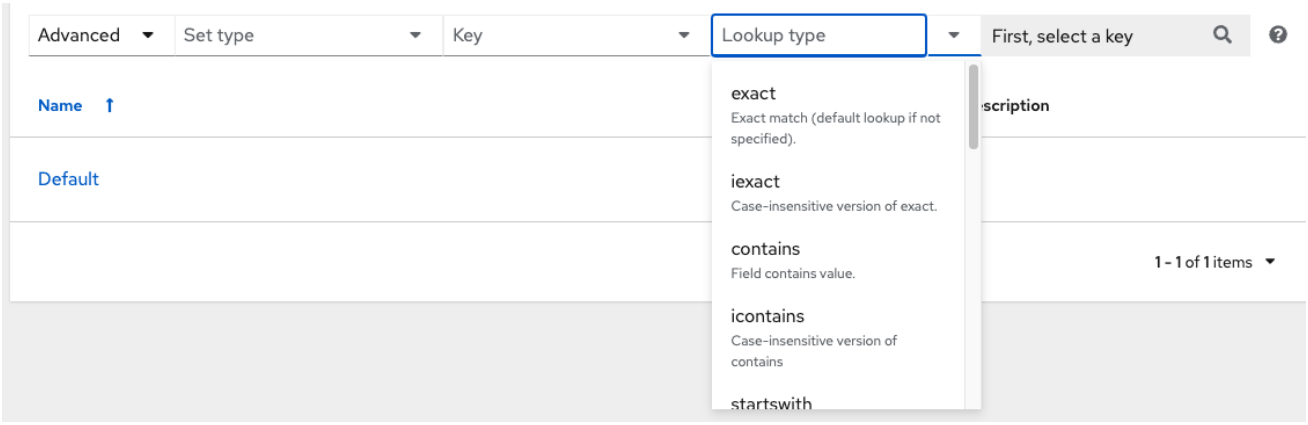
Settings 页面可让管理员配置以下内容：

- 身份验证
- Jobs
- 系统级属性
- 自定义 UI
- 产品许可证信息

第 6 章 搜索

使用自动化控制器的搜索工具在多个功能间搜索和过滤功能。在搜索字段中的 Name 菜单的 Advanced 选项中提供了可缩放的搜索条件列表。

从那里，使用 Set Type、key 和 Lookup 类型 的组合来过滤。



6.1. 用于搜索的规则

这些搜索提示假设您没有搜索主机。本节中的大多数仍然适用于主机，但有一些细微差别。

- 搜索的典型语法由一个字段（左边）和一个值（右边）组成。
- 冒号用于将您要搜索的字段与值分开。
- 如果搜索没有冒号（请参阅示例 3），它将被视为一个简单的字符串搜索，其中 **?search=foobar** 被发送。

以下是用于搜索的语法示例：

1. **name:localhost** 在此例中，用户在 name 属性中搜索字符串 'localhost'。如果该字符串与来自字段或 相关字段 的内容不匹配，则整个搜索将被视为字符串。
2. **organization.name:** 默认 示例显示了一个相关字段搜索。**organization.name** 中的句点将模型与字段分开。根据搜索的深度或复杂程度，您可以在查询的该部分中有多个句点。
3. **foobar** 这是一个简单字符串（关键字），搜索搜索词，它使用 **icontains** 搜索名称和描述字段查找所有实例。如果您在术语间使用一个空格，如 **foo bar**，则返回包含这两个术语的结果。例如，如果术语用引号括起，例如 ：**"foo bar"**，自动化控制器会搜索带有术语的字符串。

特定名称针对 API 名称搜索。例如，用户界面中的 管理作业是 API 中的 **system_job**。**Organization:Default** this example 显示了一个相关字段搜索，但没有指定与机构一起的字段。API 支持它，这与简单的字符串搜索类似，但对组织执行了一个 图标 （针对名称和描述进行搜索）。

6.1.1. 搜索字段的值

要查找某些字段的值，请参阅 API 端点以了解大量选项及其有效值。例如，如果要针对 **/api/v2/jobs** > **type** 字段搜索，您可以通过对 **/api/v2/jobs** 执行 OPTIONS 请求并在 API 中查找 **"type"** 的条目来查找这些值。另外，您可以通过滚动到每个屏幕的底部来查看相关的搜索。在 **/api/v2/jobs** 的示例中，相关的搜索显示：

```
"related_search_fields": [
    "modified_by__search",
    "project__search",
    "project_update__search",
    "credentials__search",
    "unified_job_template__search",
    "created_by__search",
    "inventory__search",
    "labels__search",
    "schedule__search",
    "webhook_credential__search",
    "job_template__search",
    "job_events__search",
    "dependent_jobs__search",
    "launch_config__search",
    "unifiedjob_ptr__search",
    "notifications__search",
    "unified_job_node__search",
    "instance_group__search",
    "hosts__search",
    "job_host_summaries__search"
```

字段的值来自 GET 请求中的键。未使用 URL、相关的 和 **summary_fields**。相关字段的值也来自 **OPTIONS** 响应，但来自不同的属性。相关字段通过获取 **related_search_fields** 中的所有值并从末尾去除 **__search** 来填充。

任何不是以来自字段的值或相关字段中的值开头的搜索都将被视为通用字符串搜索。例如，搜索 **localhost** 会导致 **UI** 将 **?search=localhost** 作为查询参数发送到 **API** 端点。这是有关名称和描述字段的 **icontains** 搜索的快捷方式。

6.1.2. 使用相关字段中的值搜索

搜索相关字段要求您使用相关字段启动搜索字符串。以下示例描述了如何使用来自相关字段 *机构中的* 值进行搜索。

搜索字符串的左侧必须以机构开始，例如 **organization:Default**。根据相关字段，您可以通过提供二级和第三字段来为搜索提供更具体的方向。例如，指定您要搜索使用与特定名称匹配的项目的所有作业模板。其语法类似于： **job_template.project.name:"A Project "**。

注意

此查询针对 **unified_job_templates** 端点执行，这是它以 **job_template** 开始的原因。如果您针对 **job_templates** 端点进行搜索，则不需要查询的 **job_template** 部分。

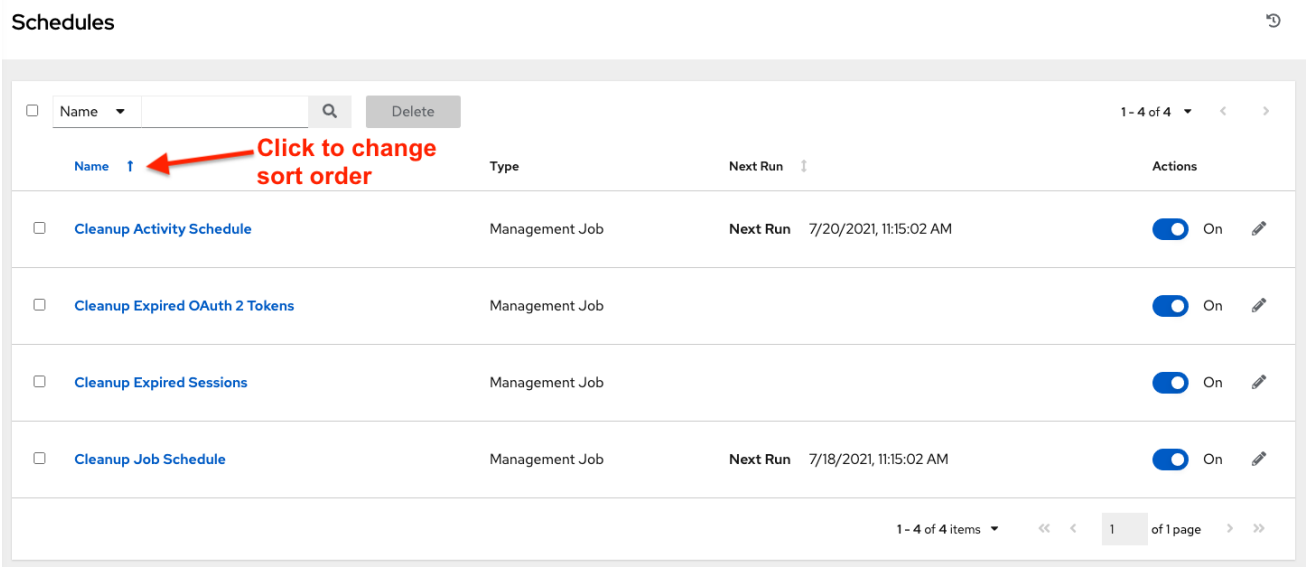
6.1.3. 其他与搜索相关的事项

在自动化控制器中搜索时请注意以下问题：

- 目前不支持 **OR** 查询的语法。所有搜索条件都是在查询参数中的 **AND**。
- 搜索参数的左首部分可以被引号包括来支持使用带有空格的搜索字符串。如需更多信息，请参阅 搜索的提示。
- 目前，字段中的值是 **GET** 请求中预期返回的直接属性。每当针对其中一个值进行搜索时，自动化控制器都会进行_ **icontains** 搜索。例如，**name:localhost** 发回 **?name__icontains=localhost**。自动化控制器目前对每个字段值执行此搜索，甚至 **id**。

6.2. 排序

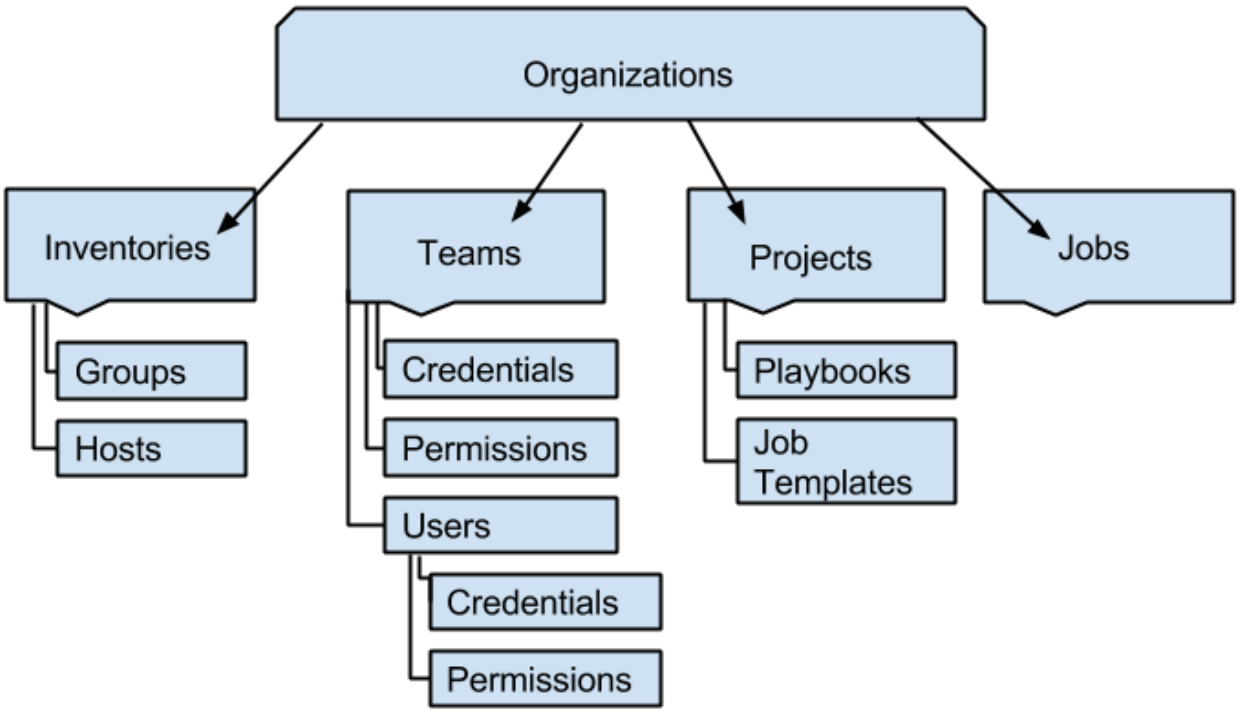
在适用的情况下，使用每个列中的箭头按升序排序。以下是调度列表中的示例：



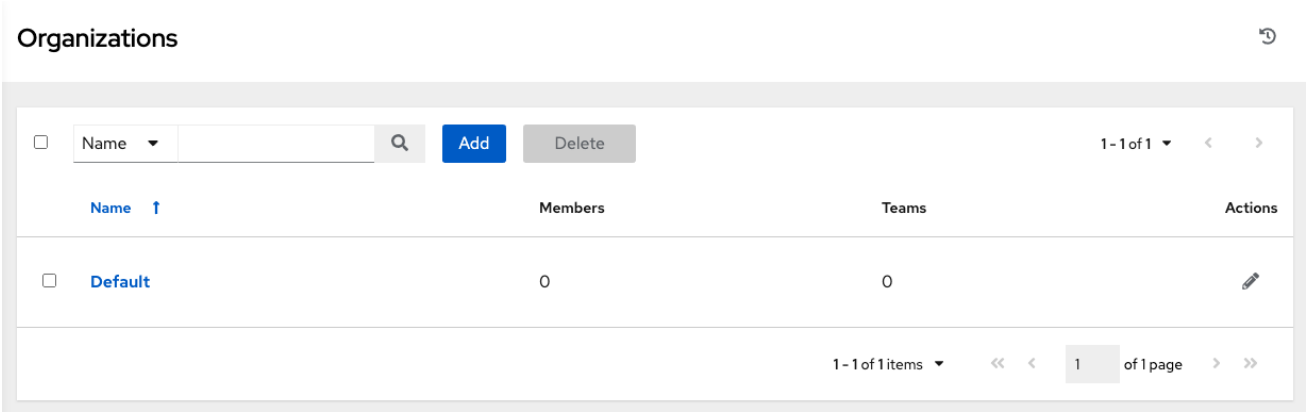
箭头的方向代表字段的排序顺序。

第 7 章 机构

机构是用户、团队、项目和清单的逻辑集合。它是控制器对象层次结构中的最高级别对象。



在导航菜单中，选择 **Organizations** 来显示您的安装的现有机构。

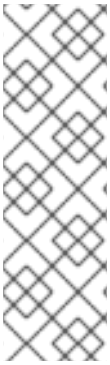


可以按照名称或描述搜索 机构。

使用

图标修改机构。点 **Delete** 删除所选机构。

7.1. 创建机构

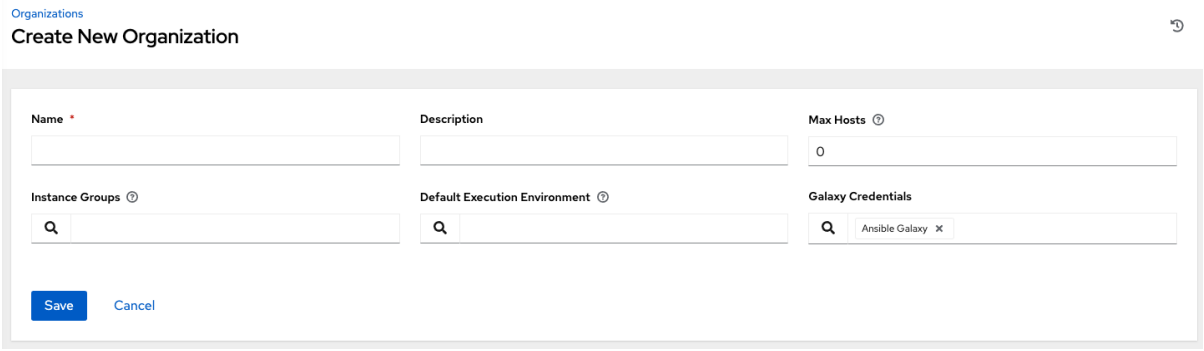


注意

自动化控制器会自动创建一个默认机构。如果您有自助支持级别许可证，则只有默认机构可用，且不得将其删除。

您可以使用默认机构，因为它最初设置并在以后对其进行编辑。

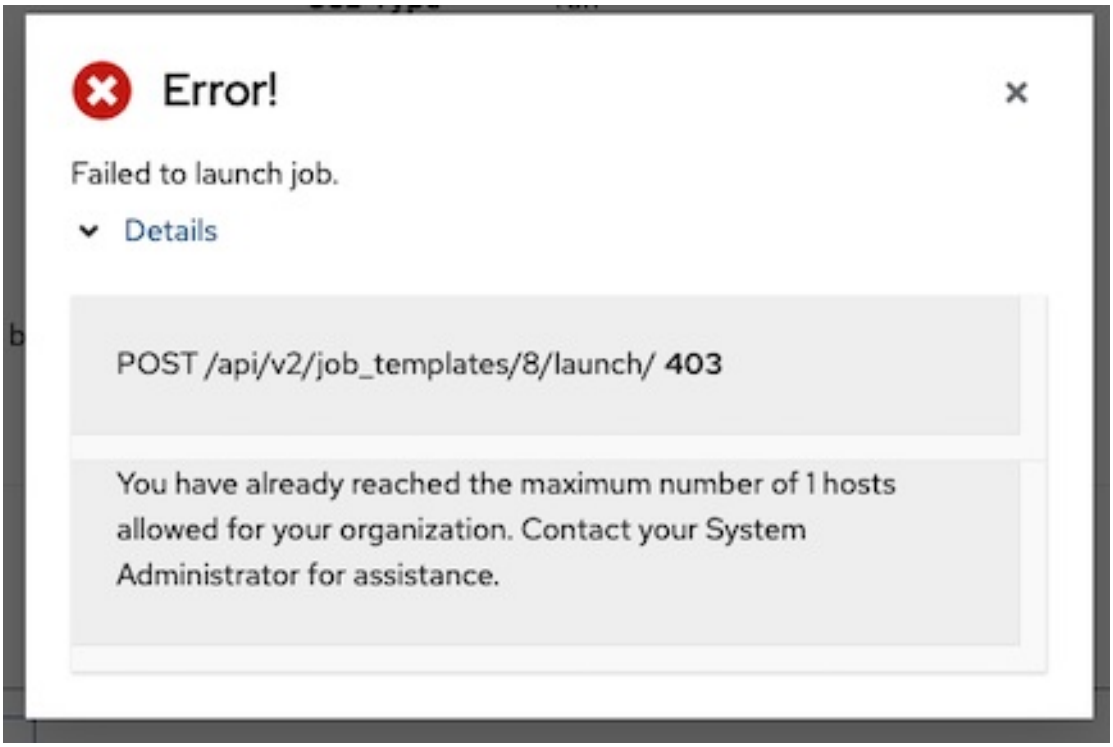
1. 单击 **Add** 以创建新组织。



2. 您可以配置机构的几个属性：

- 输入您的机构 **Name** （必需）。
- 输入组织的 **Description**。
- 最大 主机只能由超级用户编辑，以对机构可以具有的许可证主机数量设置上限。将此值设置为 **0** 表示没有限制。如果您试图将主机添加到已达到或超过其在主机上上限的机构中，则会显示错误消息：

清单同步输出视图还显示主机限制错误。

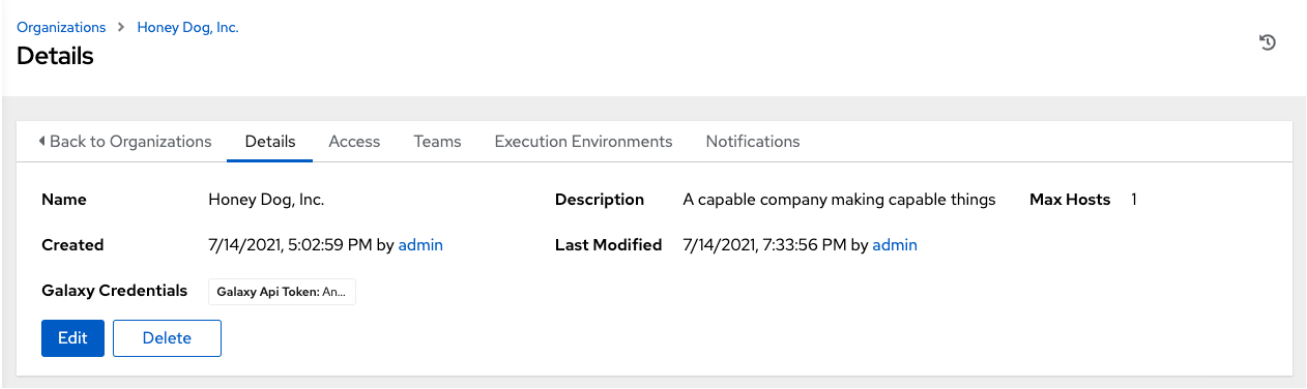


点 **Details** 来获取有关错误的额外信息。

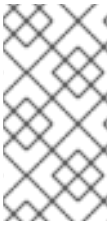
- 输入要 在其上运行此 机构的实例组的名称。
- 输入执行环境的名称，或搜索在其中运行此机构中存在的环境。如需更多信息，请参阅升级到执行环境。
- 可选：输入 **Galaxy** 凭证 或从现有凭证列表中搜索。

3. 单击 **Save** 以完成组织创建。

创建机构时，自动化控制器会显示机构详情，并允许您管理机构的访问和执行环境。



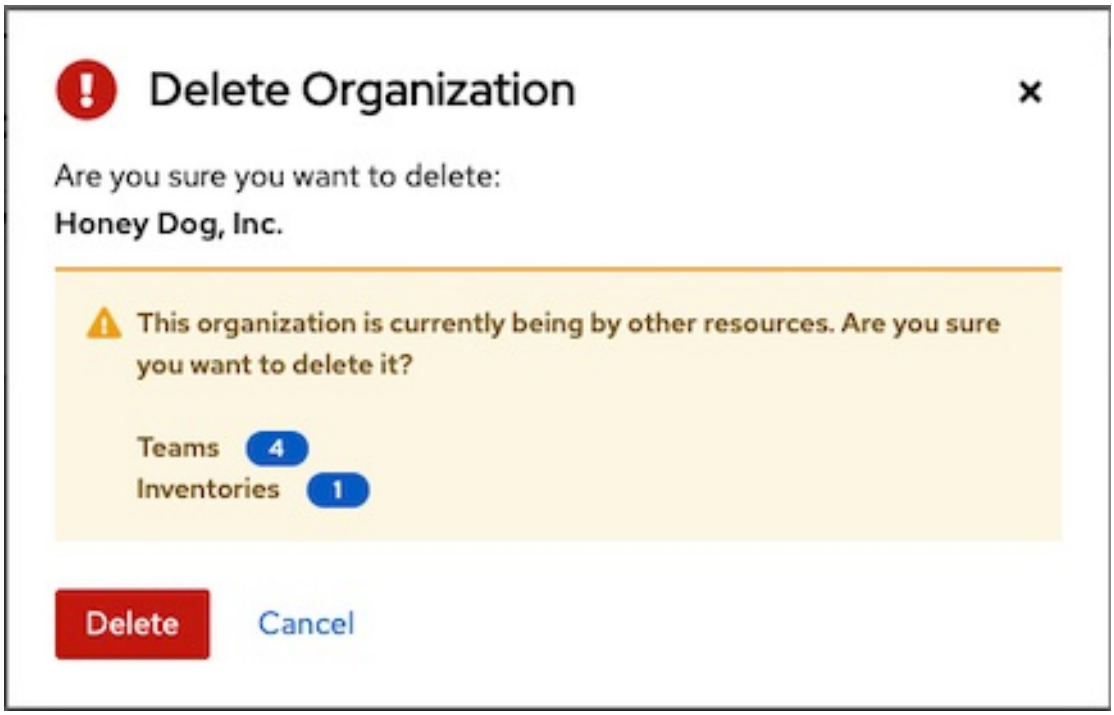
在 **Details** 选项卡中，您可以编辑或删除机构。



注意

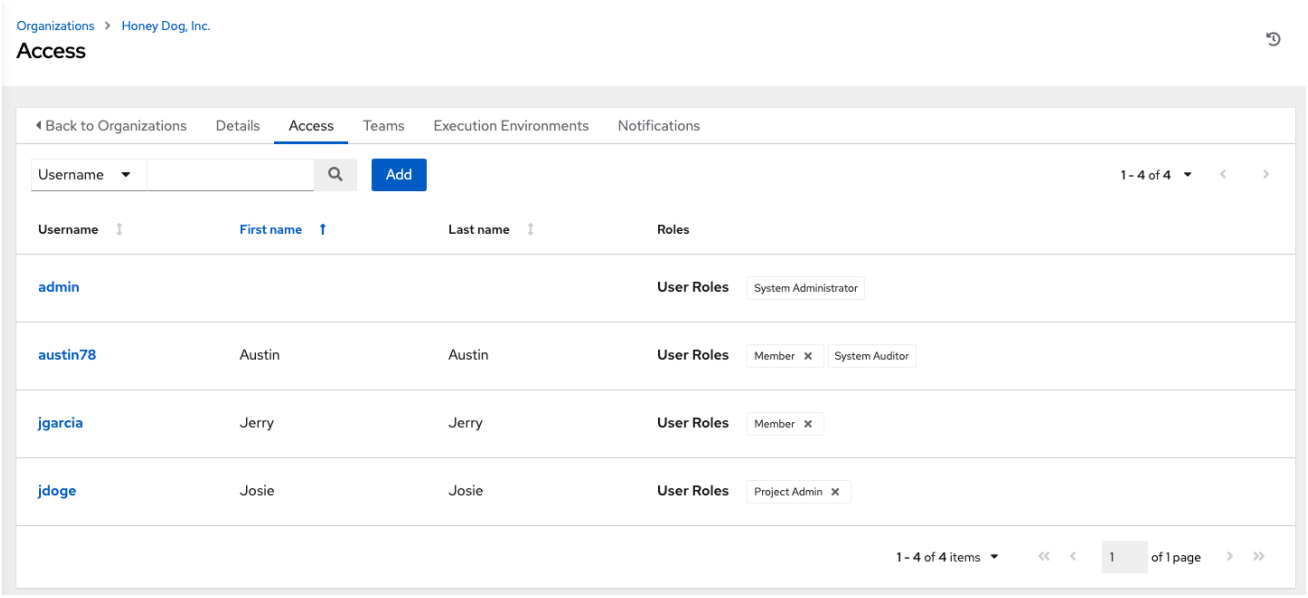
如果您尝试删除由其他工作项目使用的项目，则消息将列出受删除影响的项目，并提示您确认删除。有些屏幕包含无效的或之前已被删除的项目，且无法运行。

以下是此类消息的示例：



7.2. 访问机构

- 在查看您的机构时选择 **Access** 以显示与此机构关联的用户及其角色。



使用此页面完成以下任务：

- 管理此机构的用户成员资格。点导航面板中的 **Users**，从 **Users** 页面基于每个用户管理用户成员资格。
- 为特定用户分配机构中某些级别的权限。
- 使其充当特定资源的管理员。如需更多信息，请参阅基于角色的访问控制。

点用户显示该用户的详细信息。您可以查看、授予、编辑和删除该用户的相关权限。如需更多信息，请参阅 用户。

7.2.1. 添加用户或团队

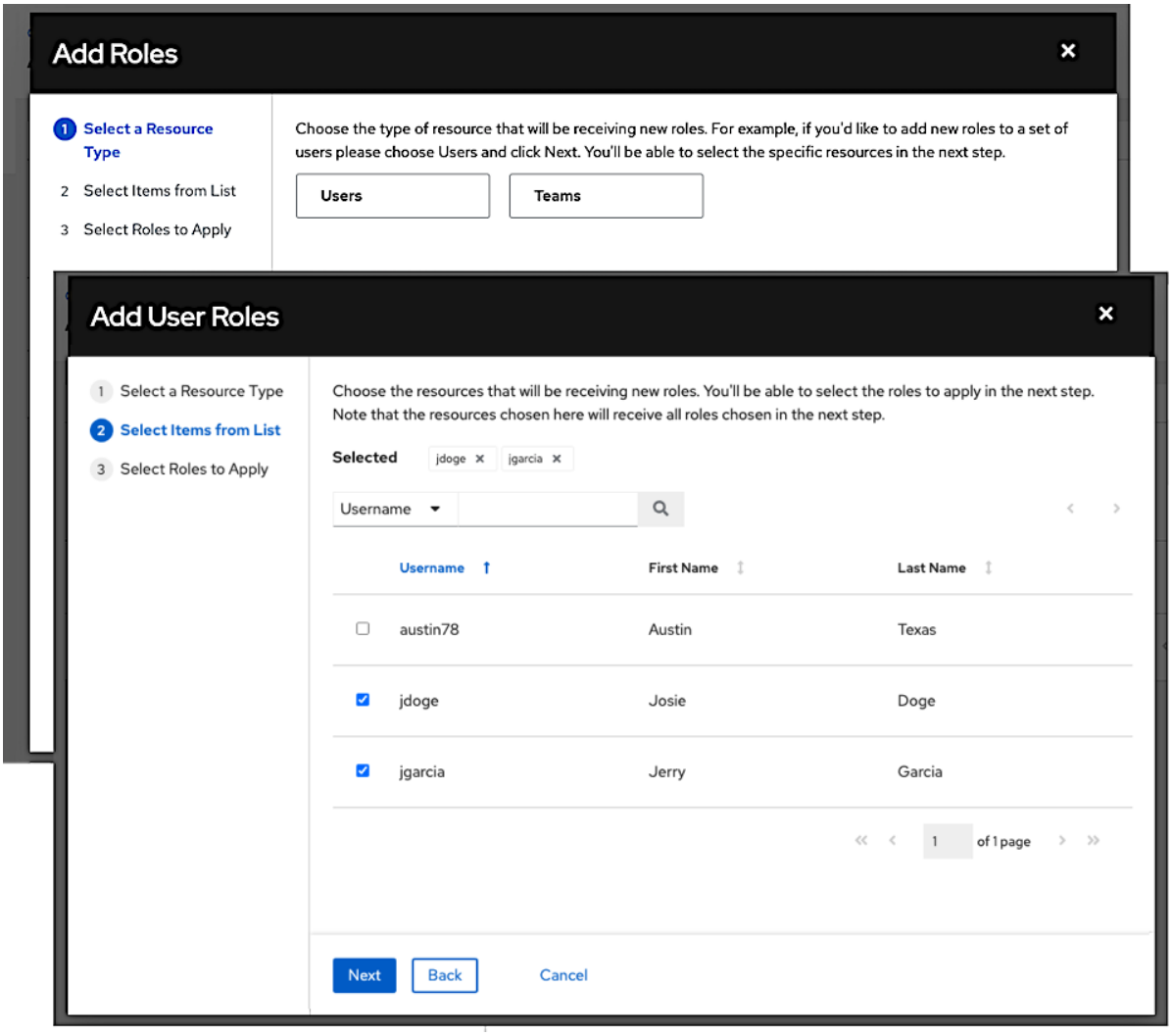
要为机构添加用户或团队，用户或团队必须已存在。

如需更多信息，请参阅 创建用户 和创建团队。

将现有用户或团队添加到机构中：

流程

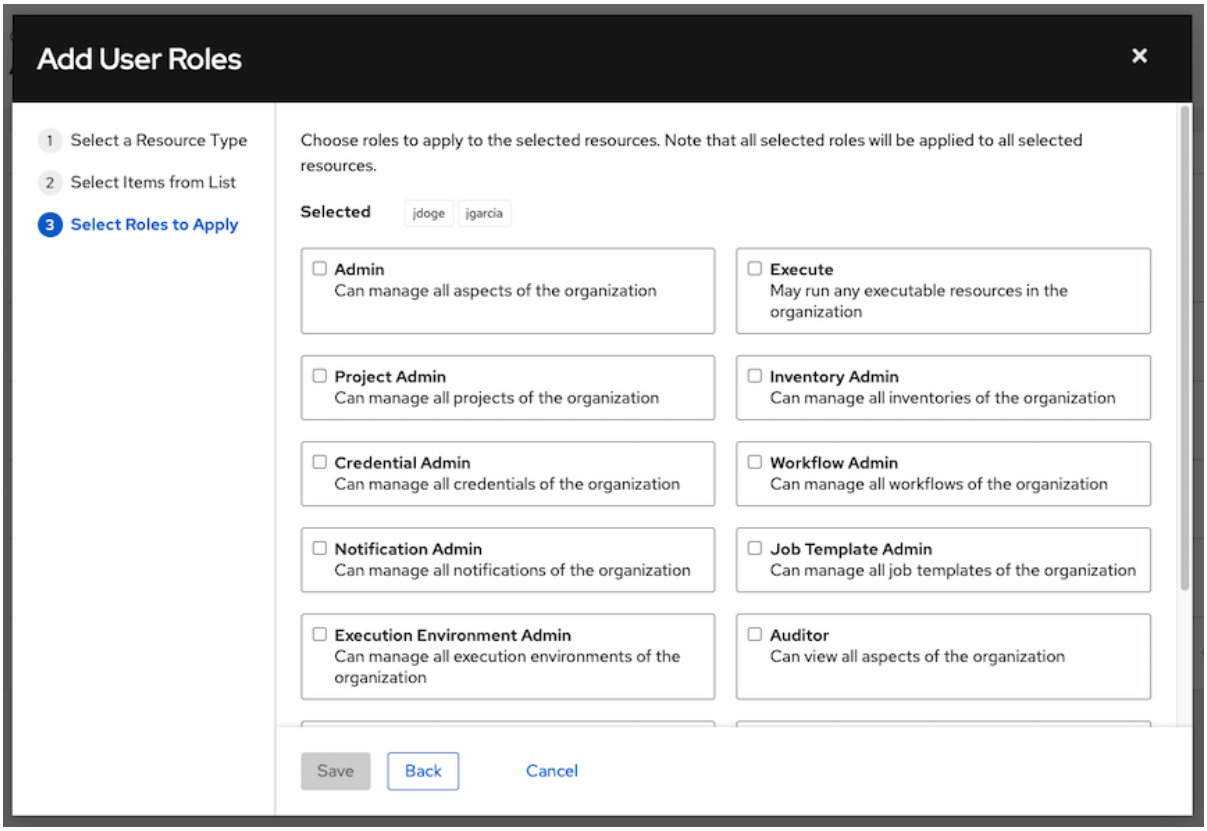
1. 在 **Organization** 页面的 **Access** 选项卡中，单击 **Add**。
2. 选择要添加的用户或团队。
3. 点击 **Next**。
4. 点名称旁边的复选框从列表中选择一个或多个用户或团队，将它们添加为成员。
5. 点击 **Next**。



在这个示例中，选择了两个用户。

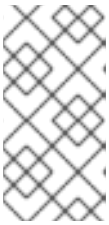
6.

选择您希望所选用户或团队具有的角色。向下滚动以获得完整的角色列表。不同的资源有不同的可用选项。



7.

点 **Save** 将角色应用到所选用户或团队，并将它们添加为成员。**Add Users** 或 **Add Teams** 窗口显示为每个用户和团队分配的更新角色。



注意

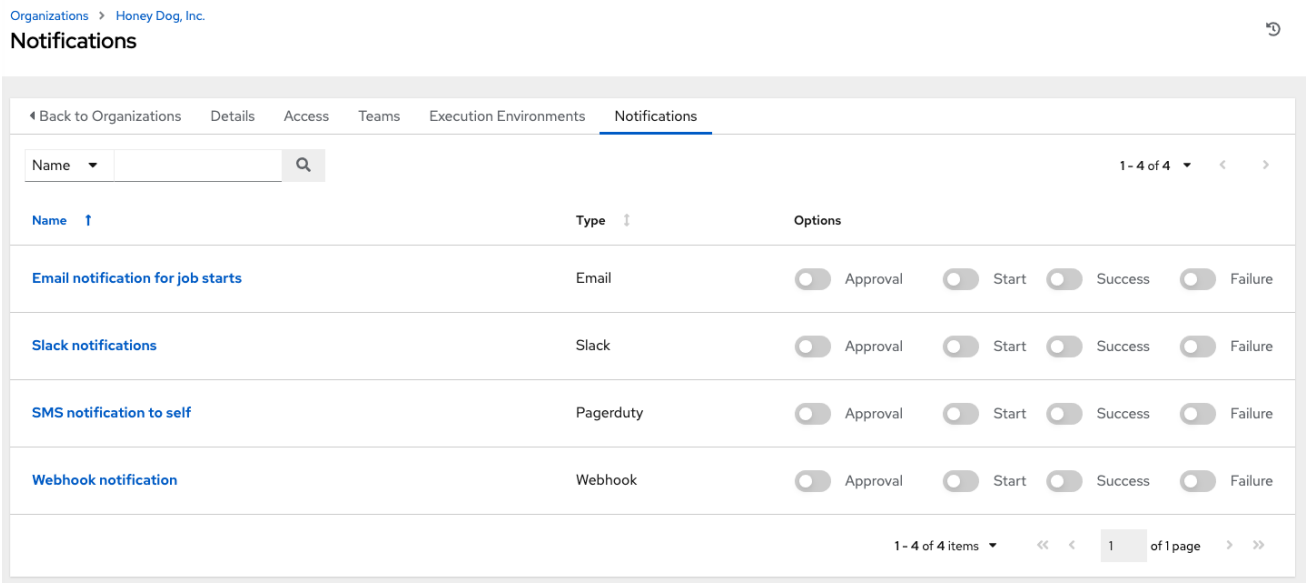
如果一个具有关联角色的用户或团队被重新分配给另一个机构，则会保留它们。

8.

要删除特定用户的角色，请点其资源旁边的解除关联 ✖ 图标。这会出现确认对话框，要求您确认解除关联。

7.2.2. 使用通知

选择机构详情页面中的 **Notifications** 选项卡可让您查看您设置的任何通知集成。



使用切换按钮启用或禁用要与特定机构搭配使用的通知。如需更多信息，请参阅 启用和禁用通知。

如果没有设置通知，请从导航面板中选择 **Administration → Notifications**。

有关配置通知类型的详情，请参考 通知类型。

第 8 章 在自动化控制器中管理用户

与某个机构关联的用户会显示在机构的 **Access** 选项卡中。

其他用户可以添加到机构中，包括 普通用户、 系统审核员 或系统管理员，但必须首先创建它们。

您可以根据 用户名、名字或 **Last Name** 进行排序或搜索 **User** 列表。点标头切换您的排序首选项。

您可以在 **Users** 页面中查看用户权限和用户类型。

8.1. 创建用户

要在自动化控制器中创建新用户，并为他们分配一个角色。

流程

1. 在 **Users** 页面中，单击 **Add**。

   此时会打开 **Create User** 对话框。

2. 输入新用户的详情。带星号 **rolebinding** 标记的字段是必需的。

注意

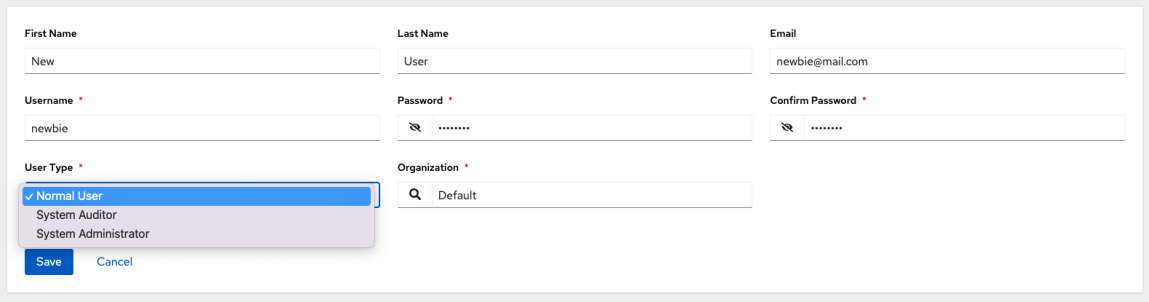
如果您要修改您自己的密码，请退出并重新登录以便其生效。

您可以分配三种类型的用户：

- 普通用户 ：普通用户具有读写访问权限，仅限于该用户获得了适当角色和权限的资源（如清单、项目和作业模板）。

- 系统审核员 ：审核员继承环境中所有对象的只读权限。

- 系统管理员 ：系统管理员（也称为超级用户）具有完整的系统管理权限 **swig-effort -** 对整个安装具有完全的读写权限。系统管理员通常负责管理所有方面，并将日常工作的职责委派给不同的用户。

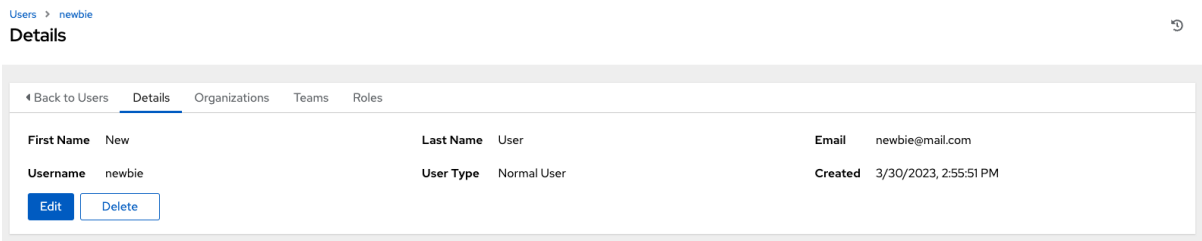


注意

在安装过程中自动创建 具有系统管理员角色的默认管理员，并可供自动化控制器的所有用户使用。一个系统管理员必须 始终存在。要删除 系统管理员 帐户，您必须首先创建另一个系统管理员帐户。

3. 点击 **Save**。

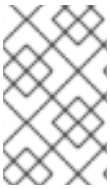
成功创建用户后，用户 对话框将打开。



4. 点 **Delete** 删除用户，也可以从当前用户列表删除用户。如需更多信息，请参阅 删除用户。

无论您点击用户名还是用户旁的 **Edit**

图标，都会打开同一窗口。您可以使用此窗口来检查和修改用户 的组织、团队、角色 和其他用户成员资格详细信息。



注意

如果没有新创建的用户，详细信息屏幕会显示该用户的最后登录活动。

如果您以自己的身份登录并查看用户配置集的详情，您可以从您的用户配置集管理令牌。

如需更多信息，请参阅 添加用户令牌。

8.2. 删除用户

在删除用户之前，您必须有用户权限。当您删除用户帐户时，用户的名称和电子邮件将从自动化控制器中永久删除。

流程

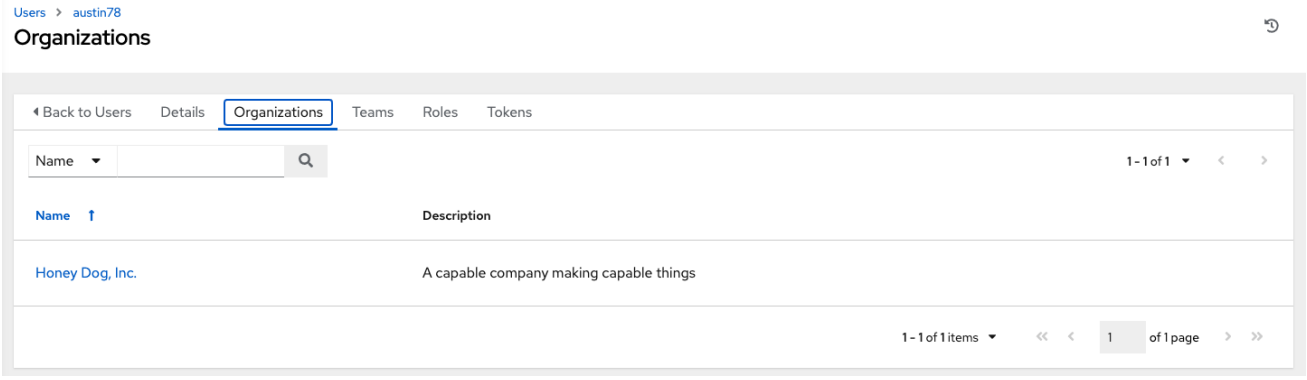
1. 在导航面板中，选择 **Access** → **Users**。
2. 单击 **Users** 以显示当前用户列表。
3. 选中您要删除的用户的复选框。
4. 单击 **Delete**。
5. 在确认警告消息中点 **Delete** 以永久删除该用户。

8.3. 显示用户机构

选择要显示 **Details** 页面的特定用户，选择 **Organizations** 选项卡来显示该用户所属的机构列表。

注意

无法从此显示面板中修改机构成员资格。



8.4. 显示用户的团队

在 **Users > Details** 页面中，选择 **Teams** 选项卡来显示该用户所属的团队列表。

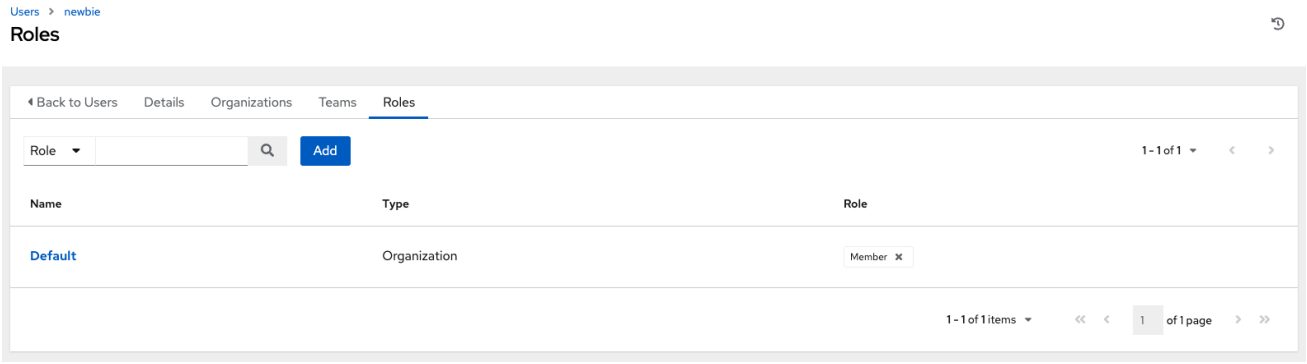
注意

您无法从此显示面板中修改团队成员资格。如需更多信息，请参阅 团队。

在创建了团队并为其分配用户之前，该用户分配的团队详情会显示为空。

8.5. 显示用户角色

在 **Users > Details** 页面中，选择 **Roles** 选项卡来显示分配给此用户的角色集合。它们提供了读取、更改和管理项目、清单、作业模板和其他元素的能力。



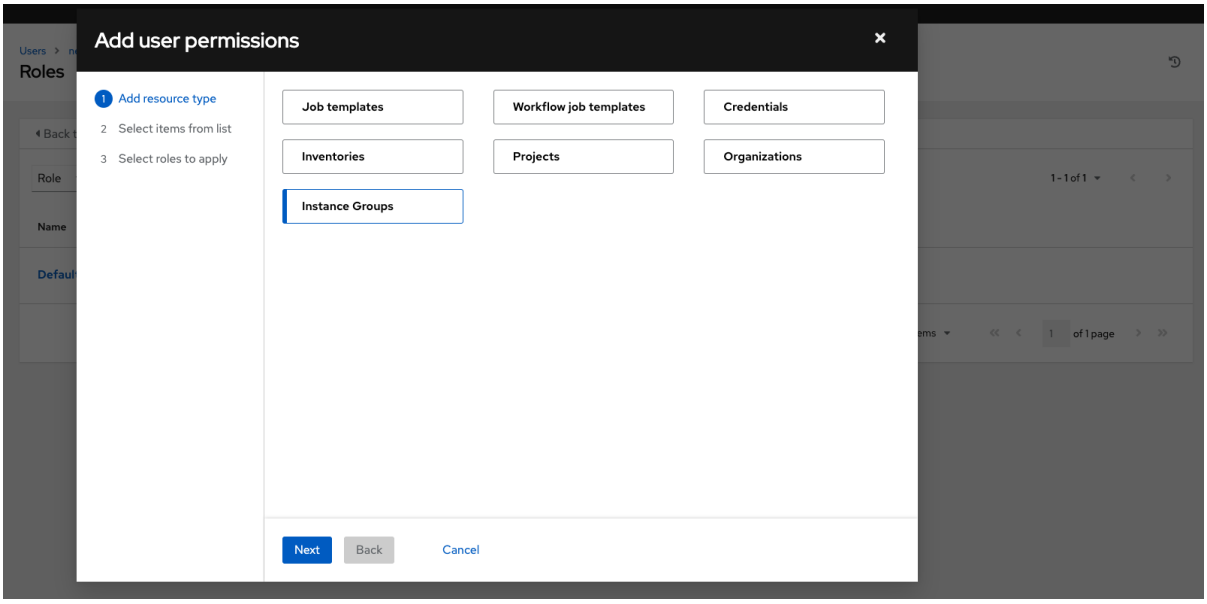
8.5.1. 添加和删除用户权限

要为特定用户添加权限，请执行以下操作：

流程

1. 从 **Users** 列表视图中，点用户的名称。

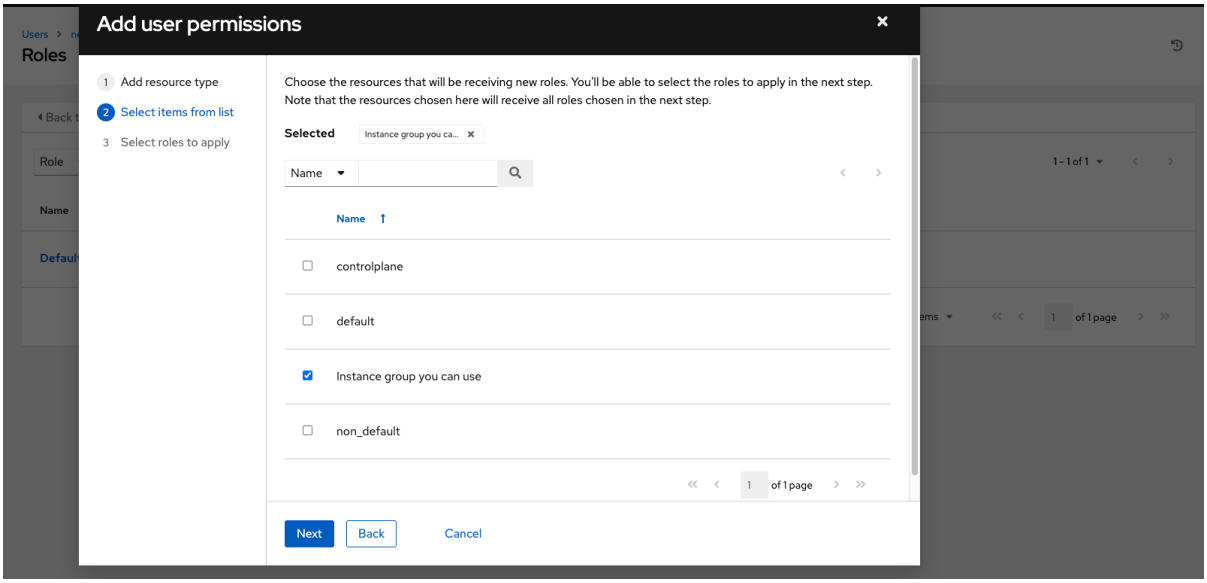
2. 在 **Details** 页面中，单击 **Add**。这会打开 **Add user permissions** 向导。



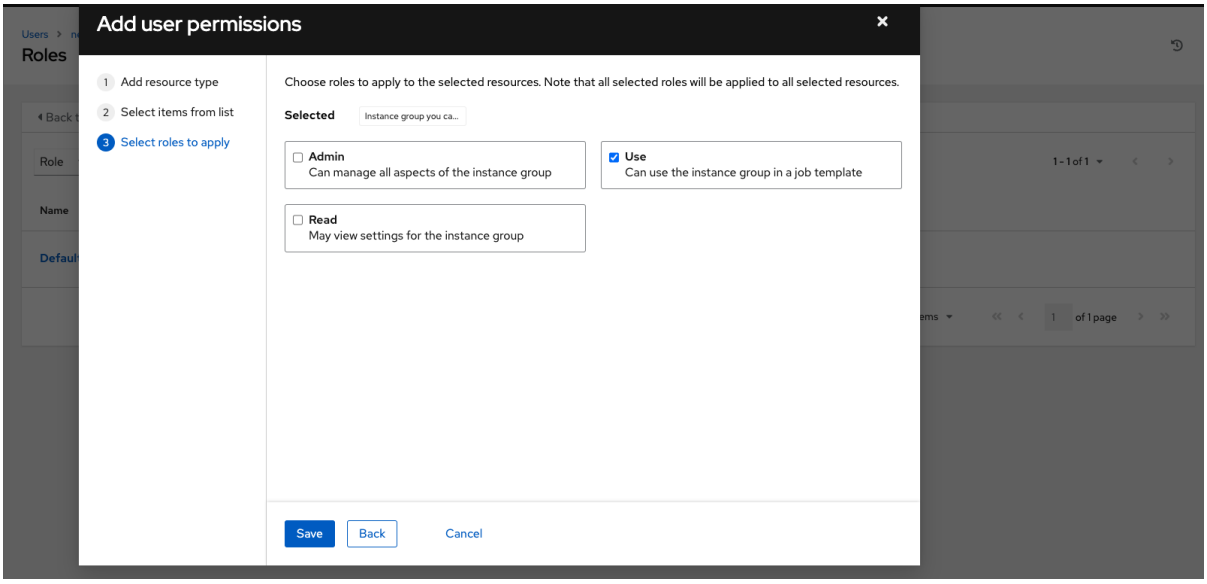
3. 选择对象到分配权限，供用户具有访问权限。

4. 点击 **Next**。

5. 选择要分配团队角色的资源，然后点 **Next**。

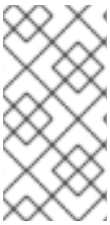


6. 选择您要为其分配权限的资源。不同的资源有不同的可用选项。



7. 点击 **Save**。

8. **Roles** 页面显示用户更新的配置文件，其具有为每个所选资源分配的权限。



注意

您还可以添加团队、个人或多个用户，并在对象级别为其分配权限。这包括模板、凭证、清单、项目、机构或实例组。这个功能减少了机构一次注册多个用户的时间。

删除权限：

- 点资源旁的 ✖ 图标。这会启动一个确认对话框，要求您确认解除关联。

8.6. 为用户创建令牌

Tokens 选项卡仅针对您为自己创建的用户显示。

在为您的用户添加令牌前，如果要将令牌与令牌关联，您可能需要 创建应用程序。

您还可以在不将其与任何应用程序关联的情况下 *创建个人访问令牌* **(PAT)**。

流程

1. 从 **Users** 列表视图中选择您的用户来配置 **OAuth 2** 令牌。
2. 从您的用户的配置文件中选择 **Tokens** 选项卡。
3. 点 **Add** 打开 **Create Token** 窗口。
4. 输入以下信息：
   - **Application:** 输入您要将令牌与令牌关联的应用程序名称。

另外，您还可以搜索点

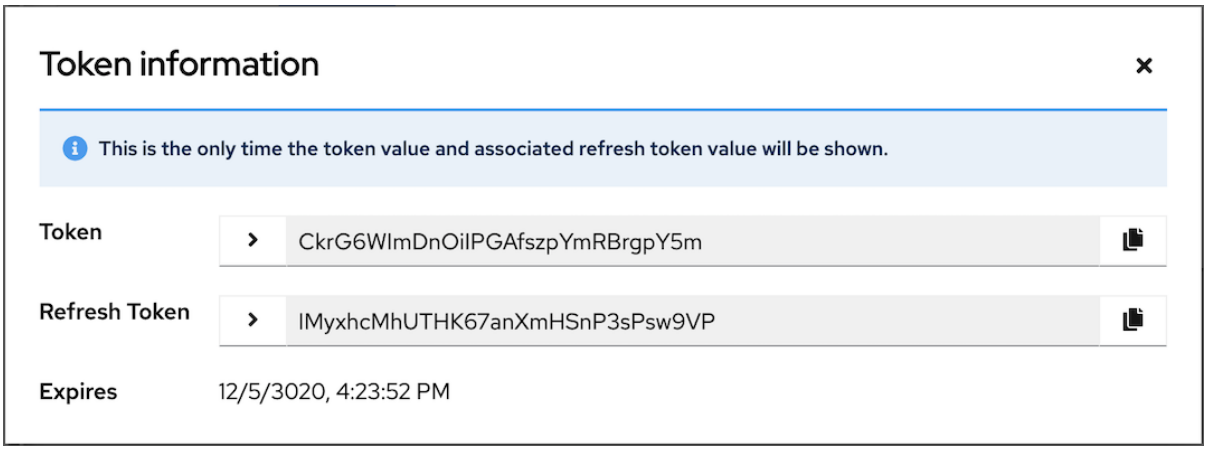
图标的应用程序名称。这会打开一个单独的窗口，供您从可用选项中选择。如果列表太长，请使用搜索栏按名称过滤。

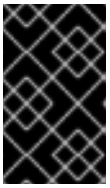
如果要创建一个未链接到任何应用程序的 **PAT**，请将此字段留空。

- 可选： 描述 ：为您的令牌提供简短描述。
- **Scope** ：指定您希望此令牌具有的访问级别。

5. 点 **Save** 或 **Cancel** 来取消您的更改。

6. 保存令牌后，会显示用户新创建的令牌。





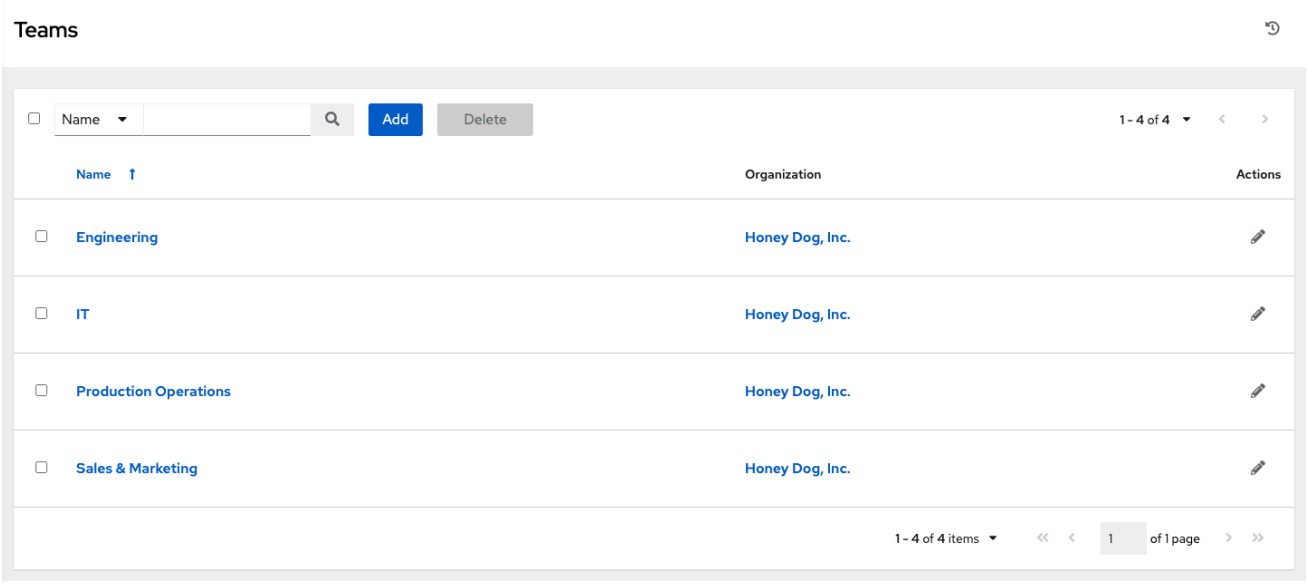
重要

这是唯一显示令牌值和关联刷新令牌值的时间。

第 9 章 管理团队

团队 是机构的一个子部门，它包括了关联的用户、项目、凭证和权限。团队提供了一种方式来实现基于角色的访问控制方案，并跨机构委派职责。例如，您可以为整个团队授予权限，而不是对团队中的每个用户授予权限。

在导航面板中，选择 **Access** → **Teams**。



您可以对团队列表进行排序和搜索，并根据 **Name** 或 **Organization** 搜索。

点击条目旁边的 **Edit**

图标编辑团队的信息。您还可以查看 与这个 团队 关联的用户和权限。

9.1. 创建团队

您可以根据您的机构的需求创建任意数量的用户团队。您可以为每个团队分配权限，就像用户一样。团队也可以为凭证分配所有权，从而尽量减少为同一用户分配相同凭据的步骤。

流程

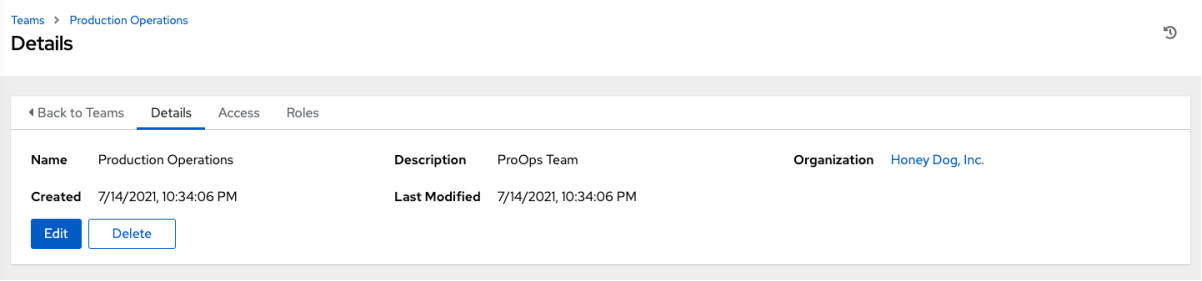
1. 在 **Teams** 页面上，单击 **Add**。

2. 在以下字段中输入相关信息：

   - **Name**
   - 可选： 描述
   - 机构 ：您必须选择一个现有机构

3. 点 **Save**。此时会打开 **Details** 对话框。

4. 查看并编辑您的团队信息。



9.1.1. 在团队中添加或删除用户

要将用户添加到团队中，用户必须已经创建。如需更多信息，请参阅 创建用户。向团队添加用户仅将他们添加为成员。使用 **Access** 选项卡为不同资源上的用户指定角色。

流程

1. 在 **Details** 页面的 **Access** 选项卡中，单击 **Add**。

2. 按照提示添加用户并将其分配到角色。

3. 点击 **Save**。

9.1.2. 删除用户的角色

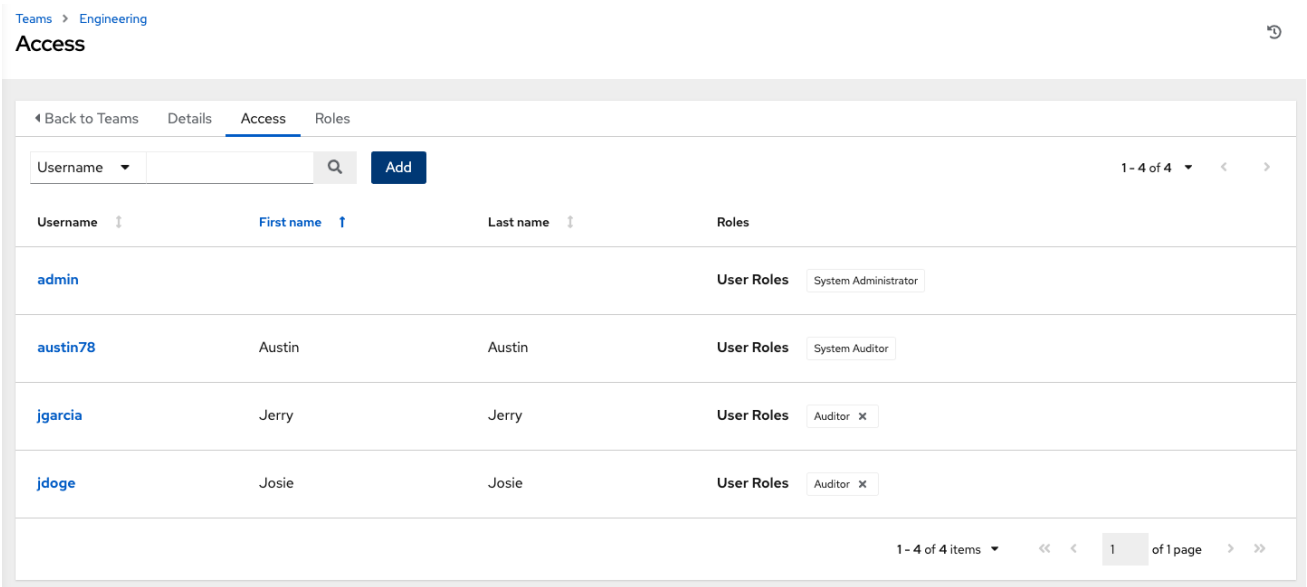
流程

- 要删除特定用户的角色，请点其资源旁的 ✖ 图标。

这会出现确认对话框，要求您确认解除关联。

9.1.3. 团队访问

Access 选项卡显示属于特定团队的成员的用户列表。



您可以根据 用户名、名字或 **Last Name** 搜索此列表。如需更多信息，请参阅 用户。

9.1.4. 团队角色和权限

选择 **Roles Details** 页面中的 **Roles** 选项卡，以显示此团队当前可用的权限列表。

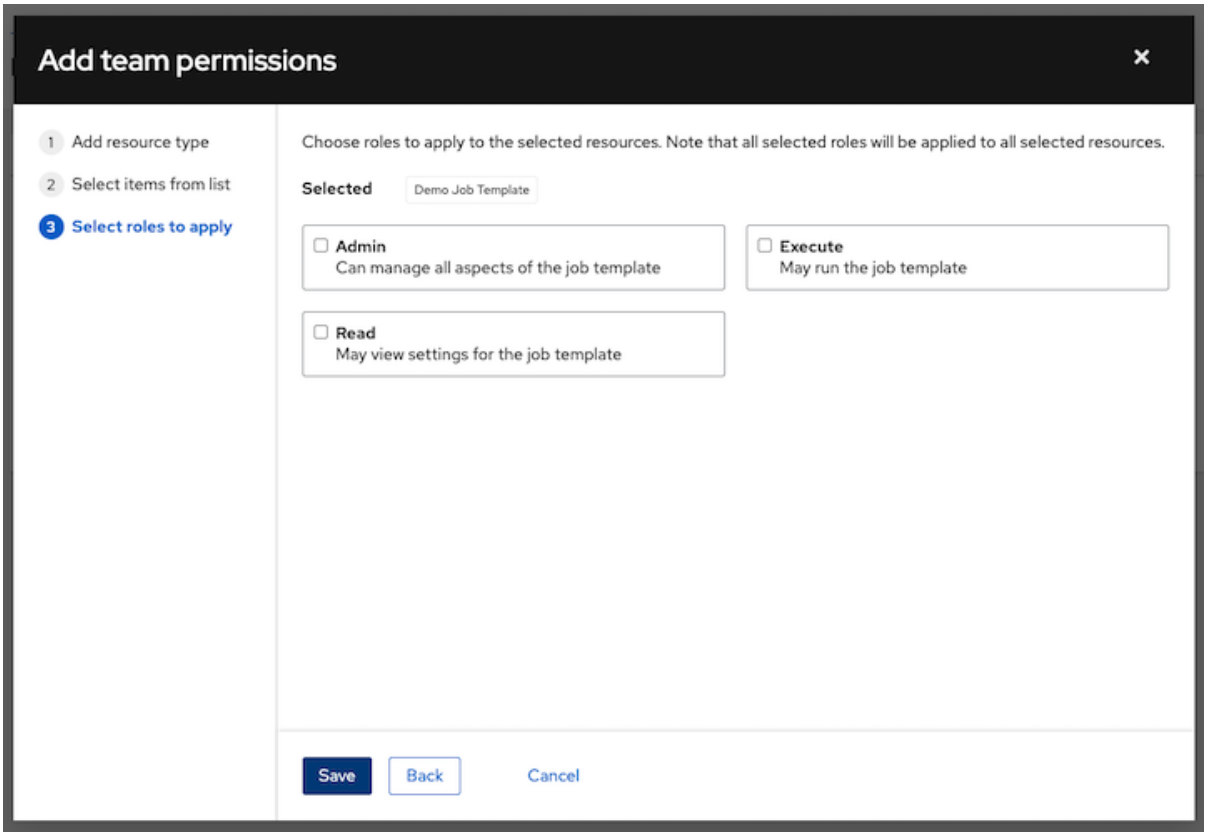
9.1.5. 添加和删除团队权限

默认情况下，您创建的所有团队都有读取权限。您可以分配其他权限，如编辑和管理项目、清单和其他元素。

您可以通过清单、项目、作业模板或 **Organizations** 视图来设置权限。

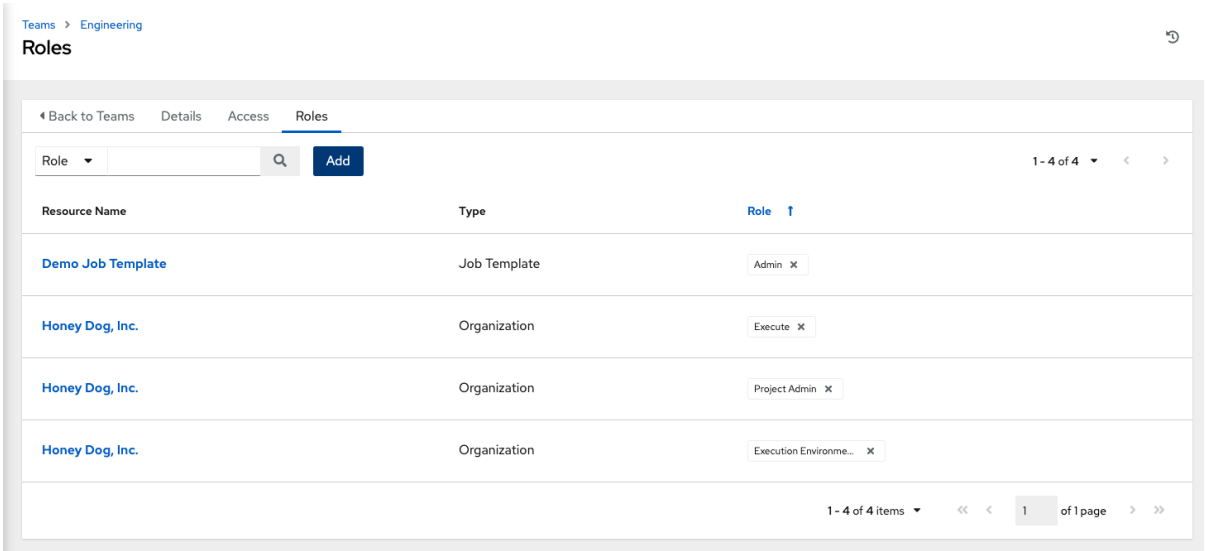
流程

1. 从 团队 列表视图中，单击所需用户。
2. 在 **Details** 页面中，单击 **Add**。这会打开 **Add team permissions** 向导。
3. 选择团队需要访问的对象。
4. 点击 **Next**。
5. 选择要分配团队角色的资源。
6. 点击 **Next**。
7. 点角色旁边的复选框，将该角色分配给您选择的资源类型。不同的资源有不同的可用选项。



8. 点击 **Save**。

9. 此时会显示用户更新的配置文件，其中包含为每个选定资源分配的角色。



9.1.5.1. 删除团队权限

- 要删除特定资源的权限，请点其资源旁的
✖
图标。这会出现确认对话框，要求您确认解除关联。

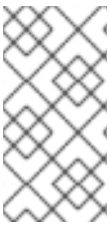
注意

您还可以添加团队、个人或多个用户，并在对象级别为其分配权限。这包括项目、清单、作业模板和工作流模板。这个功能减少了机构一次注册多个用户的时间。

第 10 章 管理用户凭证

在针对机器启动作业、与清单源同步以及从版本控制系统中导入项目内容时，凭证会验证自动化控制器用户。

您可以向用户和团队授予使用这些凭证的权限，而无需向用户公开凭证。如果用户移至不同的团队或离开机构，则不必再更新所有系统的密钥，因为这些凭证在自动化控制器中可用。



注意

自动化控制器会加密数据库中的密码和密钥信息，且永远不会通过 **API** 使 **secret** 信息可见。如需更多信息，*请参阅自动控制器管理指南*。

10.1. 凭证如何工作

自动化控制器使用 **SSH** 连接到远程主机。要将密钥从自动化控制器传递给 **SSH**，必须在将其写入命名管道之前解密密钥。自动化控制器使用该管道将密钥发送到 **SSH**，因此密钥永远不会写入磁盘。如果使用密码，自动化控制器会直接响应密码提示并解密密码，然后再将其写入提示符。

10.2. 创建新凭证

添加到团队的凭证可供团队的所有成员使用。您还可以为单个用户添加凭证。

作为初始设置的一部分，可以使用两个凭证： **Demo Credential** 和 **Ansible Galaxy**。使用 **Ansible Galaxy** 凭据作为模板。您可以复制此凭证，但不能编辑它。根据需要添加更多凭证。

流程

1. 在导航面板中，选择 **Resources** → **Credentials**。
2. 点**Add**。
3. 输入以下信息：

- 新凭证的名称。
- 可选：新凭证的描述。
- 可选：与凭证关联的机构名称。

注意

如果凭证被重新分配给另一个机构，则具有与一个机构关联的一组权限的凭证会保留。

4. 在 **Credential Type** 字段中，输入或选择您要创建的凭证类型。
5. 根据所选的凭证类型输入相关详情，如 凭证类型 所述。
6. 点击 **Save**。

10.3. 将新用户和作业模板添加到现有凭证

流程

1. 在导航面板中，选择 **Resources** → **Credentials**。
2. 选择您要分配给其他用户的凭证。
3. 点 **Access** 选项卡。您可以查看与此凭证及其角色关联的用户和团队。
4. 选择用户并单击 添加。如果没有用户，从 **Users** 菜单添加它们。如需更多信息，请参阅 用户。
5. 选择 **Job Templates** 以显示与此凭证关联的作业模板，以及最近使用这个凭证运行的作业。

6. 选择作业模板并点 **Add** 将凭证分配给额外的作业模板。有关创建新作业模板的更多信息，请参阅 作业模板 部分。

10.4. 凭证类型

自动化控制器支持以下凭证类型：

- **Amazon Web Services**
- **Ansible Galaxy/Automation Hub API** 令牌
- **Centrify Vault Credential Provider Lookup**
- 容器注册表
- **CyberArk Central Credential Provider Lookup**
- **CyberArk Conjur Secrets Manager Lookup**
- **GitHub** 个人访问令牌
- **GitLab** 个人访问令牌
- **Google Compute Engine**
- **GPG** 公钥
- **HashiCorp Vault Secret Lookup**

- **HashiCorp Vault Signed SSH**
- **Insights**
- 机器
- **Microsoft Azure Key Vault**
- **Microsoft Azure Resource Manager**
- 网络
- **OpenShift 或 Kubernetes API 持有者令牌**
- **OpenStack**
- **Red Hat Ansible Automation Platform**
- **Red Hat Satellite 6**
- **Red Hat Virtualization**
- 源控制
- **Thycotic DevOps Secrets Vault**
- **Thycotic Secret Server**

- Vault
- VMware vCenter

与 **Centrify**、**CyberArk**、**HashiCorp Vault**、**Microsoft Azure Key Vault** 和 **Thycotic** 关联的凭证类型是凭证插件功能的一部分，它允许外部系统查找您的 **secret** 信息。

如需更多信息，请参阅 **Secret** 管理系统。

10.4.1. Amazon Web Services 凭证类型

选择此凭证以启用与 **Amazon Web Services** 的云清单同步。

自动化控制器为 **AWS** 凭证使用以下环境变量：

```
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY
AWS_SECURITY_TOKEN
```

用户界面中会提示这些字段。

Amazon Web Services 凭据由 **AWS** 访问密钥和 **Secret Key** 组成。

自动化控制器提供对 **EC2 STS** 令牌的支持，也称为 **Identity and Access Management (IAM) STS** 凭证。*安全令牌服务* **(STS)**是一个 **Web** 服务，可让您为 **AWS IAM** 用户请求临时的、有有限权限的凭证。

注意

如果 **EC2** 中的标签值包含布尔值（是/**no**/**true**/**false**），则必须为它们加上引号。

警告

要使用隐式 **IAM** 角色凭证，在依赖 **IAM** 角色访问 **AWS API** 时请不要在自动化控制器中附加 **AWS** 云凭证。

将 **AWS** 云凭证附加到作业模板会强制使用 **AWS** 凭证，而不是您的 **IAM** 角色凭证。

其他资源

如需有关 **IAM/EC2 STS** 令牌的更多信息，请参阅 **IAM** 中的临时安全凭证。

10.4.1.1. 在 Ansible Playbook 中访问 Amazon EC2 凭证

您可以从作业运行时环境获取 **AWS** 凭证参数：

```
vars:
  aws:
    access_key: '{{ lookup("env", "AWS_ACCESS_KEY_ID") }}'
    secret_key: '{{ lookup("env", "AWS_SECRET_ACCESS_KEY") }}'
    security_token: '{{ lookup("env", "AWS_SECURITY_TOKEN") }}'
```

10.4.2. Ansible Galaxy/Automation Hub API 令牌凭证类型

选择此凭据以访问 **Ansible Galaxy**，或者使用在私有自动化中心实例上发布的集合。

在此屏幕上输入 **Galaxy** 服务器 **URL**。

使用 **Red Hat Hybrid Cloud Console** 上的 **Server URL** 字段的内容填充 **Galaxy Server URL** 字段。使用 **Red Hat Hybrid Cloud Console** 上的 **SSO URL** 字段的内容填充 **Auth Server URL** 字段。

其他资源

如需更多信息，请参阅 在自动化中心中使用集合。

10.4.3. Centrify Vault Credential Provider Lookup 凭证类型

这被视为 **secret** 管理功能的一部分。如需更多信息，请参阅 **Centrify Vault Credential Provider Lookup**。

10.4.4. Container Registry 凭证类型

选择此凭证可让自动化控制器访问容器镜像集合。如需更多信息，请参阅 容器 **registry** 是什么？

您必须指定一个名称。**Authentication URL** 字段预先填充一个默认值。您可以通过为不同的容器 **registry** 指定身份验证端点来更改值。

10.4.5. CyberArk Central Credential Provider Lookup 凭证类型

这被视为 **secret** 管理功能的一部分。

如需更多信息，请参阅 **CyberArk Central Credential Provider (CCP) Lookup**。

10.4.6. CyberArk Conjur Secrets Manager Lookup 凭证类型

这被视为 **secret** 管理功能的一部分。

如需更多信息，请参阅 **CyberArk Conjur Secrets Manager Lookup**。

10.4.7. GitHub 个人访问令牌凭证类型

选择这个凭证可让您使用 *个人访问令牌(PAT)访问* **GitHub**，您可以通过 **GitHub** 获取该凭证。

如需更多信息，请参阅使用 **Webhook**。

GitHub PAT 凭证需要在 **Token** 字段中提供一个值，它位于您的 **GitHub** 配置集设置中。

使用此凭证建立与 **GitHub** 的 **API** 连接，以用于 **Webhook** 侦听器作业，以发布状态更新。

10.4.8. GitLab 个人访问令牌凭证类型

选择这个凭证可让您使用 *个人访问令牌(PAT)访问* **GitLab**，您可以通过 **GitLab** 获取该凭证。

如需更多信息，请参阅使用 **Webhook**。

GitLab PAT 凭证需要在 **Token** 字段中提供一个值，它位于 **GitLab** 配置集设置中。

使用此凭证建立与 **GitLab** 的 **API** 连接，以用于 **Webhook** 侦听器作业，以发布状态更新。

10.4.9. Google Compute Engine 凭证类型

选择此凭证以启用与 **Google Compute Engine (GCE)**的云清单同步。

自动化控制器为 **GCE** 凭证使用以下环境变量：

```
GCE_EMAIL
GCE_PROJECT
GCE_CREDENTIALS_FILE_PATH
```

用户界面中会提示这些字段：

GCE 凭证需要以下信息：

- 服务帐户电子邮件地址 ：分配给 **Google Compute Engine** 服务帐户的电子邮件地址。
- 可选： 项目 ：提供 **GCE** 分配的标识或您在项目创建时提供的唯一项目 **ID**。
- 可选： 服务帐户 **JSON** 文件 ：上传 **GCE** 服务帐户文件。点 **Browse** 浏览具有特殊帐户信息的文件，这些文件可供 **GCE** 实例上运行的服务和应用程序用于与其他 **Google Cloud Platform**

API 交互。这为服务帐户和虚拟机实例授予权限。

- RSA 私钥 ：与服务帐户电子邮件关联的 PEM 文件。

10.4.9.1. 在 Ansible Playbook 中访问 Google Compute Engine 凭证

您可以从作业运行时环境获取 GCE 凭证参数：

```
vars:
  gce:
    email: '{{ lookup("env", "GCE_EMAIL") }}'
    project: '{{ lookup("env", "GCE_PROJECT") }}'
    pem_file_path: '{{ lookup("env", "GCE_PEM_FILE_PATH") }}'
```

10.4.10. GPG 公钥凭证类型

选择此凭证类型，以启用自动化控制器，在与源控制同步时验证项目的完整性。

有关如何生成有效密钥对的更多信息，请使用 CLI 工具签署内容，以及如何将公钥添加到控制器，请参阅 项目签名和验证。

10.4.11. HashiCorp Vault Secret Lookup 凭证类型

这被视为 secret 管理功能的一部分。

如需更多信息，请参阅 HashiCorp Vault Secret Lookup。

10.4.12. HashiCorp Vault Signed SSH 凭证类型

这被视为 secret 管理功能的一部分。

如需更多信息，请参阅 HashiCorp Vault Signed SSH。

10.4.13. Insights 凭证类型

选择此凭证类型，以启用与 **Red Hat Insights** 的云清单同步。

Insights 凭证是 **Insights Username** 和 **Password**，它们是用户的红帽客户门户网站帐户的用户名和密码。

Insights 的 **extra_vars** 和 **env injectors** 如下：

```
ManagedCredentialType(
   namespace='insights',
....
....
....

injectors={
      'extra_vars': {
         "scm_username": "{{username}}",
         "scm_password": "{{password}}",
      },
      'env': {
         'INSIGHTS_USER': '{{username}}',
         'INSIGHTS_PASSWORD': '{{password}}',
      },
```

10.4.14. 机器凭证类型

机器凭据可让自动化控制器在管理下的主机上调用 **Ansible**。您可以指定 **SSH** 用户名，可选地提供密码、**SSH** 密钥、密钥密码，或者让自动化控制器在部署时提示用户输入其密码。它们为 **playbook** 定义 **SSH** 和用户级特权升级访问权限，并在提交作业以在远程主机上运行 **playbook** 时使用。

以下网络连接使用 **Machine** 作为凭证类型： **httpapi**、**netconf** 和 **network_cli**

机器和 **SSH** 凭据不使用环境变量。它们通过 **ansible -u** 标志传递用户名，并在底层 **SSH** 客户端提示时以交互方式写入 **SSH** 密码。

机器凭证需要以下输入：

- 用户名 ：用于 **SSH** 身份验证的用户名。
- **Password:** 用于 **SSH** 验证的密码。如果输入，此密码将加密存储在数据库中。或者，您也

可以通过选择 **Prompt on launch** 来在启动时要求用户输入密码。在这些情况下，在作业启动时打开一个对话框，提升用户以输入密码和密码确认。

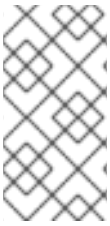
- **SSH Private Key** ：复制或拖放机器凭证的 **SSH** 私钥。
- **Private Key Passphrase:** 如果 **SSH** 私钥受密码保护，您可以为私钥配置密钥密码。如果输入，此密码将加密存储在数据库中。您还可以通过选择 **Prompt on launch**，将自动化控制器配置为在启动时要求用户输入密钥密码短语。在这些情况下，在作业启动时打开一个对话框，提示用户输入密钥密码短语和密钥密码短语确认。
- **Privilege Escalation Method** ：指定要分配给特定用户的升级权限类型。这与指定 **-become-method=BECOME_METHOD** 参数相同，其中 **BECOME_METHOD** 是任何现有方法或您编写的自定义方法。开始输入方法的名称，以及适当的名称 **auto-populates**。
- 空选择 ：如果某个任务或 **play** 已设为 **yes**，并且与空选择一起使用，则它将默认为 **sudo**。
- **sudo** ：执行具有超级用户**(root** 用户**)**特权的单个命令。
- **su** ：切换到超级用户**(root** 用户**)**帐户（或其他用户帐户）。
- **Pbrun** ：请求在受控帐户中运行应用程序或命令，并为高级 **root** 权限委托和键盘记录提供。
- **P fexec** ：执行带有预定义的进程属性的命令，如特定的用户或组 **ID**。
- **dzdo: sudo** 的增强版本，它使用 **Centrify** 的 **Active Directory** 服务中的 **RBAC** 信息。如需更多信息，请参阅 **DZDO** 上的 **Centrify** 站点。
- **pmrun** ：在控制帐户中运行应用程序的请求。请参阅 **Unix 6.0** 的特权管理器。
- **RunAs** ：使您能够以当前用户身份运行。
- 启用 ：切换到网络设备上的升级权限。

- **doas** ：使您的远程/登录用户能够以另一个用户的身份通过 ***doas*** **("Do as user")**实用程序运行命令。

- **ksu** ：使您的远程/登录用户能够通过 **Kerberos** 访问以另一个用户的身份运行命令。

- **machinectl** ：允许您通过 **systemd** 机器管理器管理容器。

- **sesu** ：使您的远程/登录用户能够通过 **CA** 特权访问管理器以另一个用户身份运行命令。

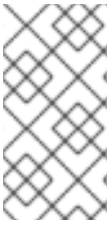


注意

Ansible 2.8+ 提供了自定义 **become** 插件。如需更多信息，请参阅了解 **Privilege Escalation** 和 **Become** 插件列表

- 权限升级用户名 ：只有在您为权限升级 选择了选项时，才会看到此字段。输入要与远程系统升级权限一起使用的用户名。

- 权限升级密码 ：只有在您为权限升级选择了选项时看到此字段。输入密码，以使用 通过远程系统中的所选权限升级类型来验证用户。此密码将加密存储在数据库中。您还可以通过选择 **Prompt on launch** 将自动化控制器配置为在启动时要求用户输入密码。在这些情况下，在作业启动时打开一个对话框，提升用户以输入密码和密码确认。



注意

您必须使用 **sudo** 密码或 **SSH** 私钥结合使用，因为自动化控制器必须首先与主机建立经过身份验证的 **SSH** 连接，然后才能调用 **sudo** 以更改为 **sudo** 用户。

警告

在调度作业中使用的凭证不能配置为 **Prompt on launch**。

10.4.14.1. 访问 ansible playbook 中的机器凭证

您可以从 **Ansible** 事实获取用户名和密码：

```
vars:
  machine:
    username: '{{ ansible_user }}'
    password: '{{ ansible_password }}'
```

10.4.15. Microsoft Azure Key Vault 凭证类型

这被视为 **secret** 管理功能的一部分。

如需更多信息，请参阅 **Microsoft Azure Key Vault**。

10.4.16. Microsoft Azure Resource Manager 凭证类型

选择此凭证类型，以启用与 **Microsoft Azure Resource Manager** 的云清单同步。

Microsoft Azure Resource Manager 凭证需要以下输入：

- 订阅 **ID** ：**Microsoft Azure** 帐户的订阅 **UUID**。
- 用户名 ：用于连接 **Microsoft Azure** 帐户的用户名。
- **Password:** 用于连接到 **Microsoft Azure** 帐户的密码。
- 客户端 **ID** ：**Microsoft Azure** 帐户的客户端 **ID**。
- **Client Secret** ：**Microsoft Azure** 帐户的客户端 **Secret**。
- 租户 **ID** ：**Microsoft Azure** 帐户的租户 **ID**。

- **Azure Cloud Environment** ：与 **Azure** 云或 **Azure** 堆栈环境关联的变量。

这些字段等于 **API** 中的变量。

要传递服务主体凭证，请定义以下变量：

```
AZURE_CLIENT_ID
AZURE_SECRET
AZURE_SUBSCRIPTION_ID
AZURE_TENANT
AZURE_CLOUD_ENVIRONMENT
```

要传递 **Active Directory** 用户名和密码对，请定义以下变量：

```
AZURE_AD_USER
AZURE_PASSWORD
AZURE_SUBSCRIPTION_ID
```

您还可以将凭证作为参数传递给 **playbook** 中的任务。优先级顺序是参数，然后是环境变量，最后是位于您主目录中的文件。

要将凭证作为参数传递给某个任务，请为服务主体凭证使用以下参数：

```
client_id
secret
subscription_id
tenant
azure_cloud_environment
```

或者，为 **Active Directory** 用户名/密码传递以下参数：

```
ad_user
password
subscription_id
```

10.4.16.1. 访问 ansible playbook 中的 Microsoft Azure 资源管理器凭证

您可以从作业运行时环境获取 **Microsoft Azure** 凭证参数：

```
vars:
  azure:
    client_id: '{{ lookup("env", "AZURE_CLIENT_ID") }}'
    secret: '{{ lookup("env", "AZURE_SECRET") }}'
    tenant: '{{ lookup("env", "AZURE_TENANT") }}'
    subscription_id: '{{ lookup("env", "AZURE_SUBSCRIPTION_ID") }}'
```

10.4.17. 网络凭证类型

注意

如果您使用与 *提供程序* 的本地连接，*请选择* **Network credential type**，以使用 **Ansible** 网络模块连接和管理网络设备。

当连接到网络设备时，凭证类型必须与连接类型匹配：

- 对于使用 提供程序的本地连接，凭证类型应该是 **Network**。
- 对于所有其他网络连接(**httpapi**、**netconf** 和 **network_cli**)，凭证类型应该是 **Machine**。

有关网络设备的可用连接类型的更多信息，请参阅 多通信协议。

自动化控制器为网络凭证使用以下环境变量：

```
ANSIBLE_NET_USERNAME
ANSIBLE_NET_PASSWORD
```

为网络凭证提供以下信息：

- 用户名 ：与网络设备结合使用的用户名。
- **Password** ：与网络设备结合使用的密码。
- **SSH Private Key** ：复制或拖放要用于通过 **SSH** 向网络验证用户的实际 **SSH** 私钥。

- **Private Key Passphrase:** 通过 **SSH** 向网络验证用户的私钥的密码短语。
- 授权 ：从 **Options** 字段中选择此项来控制是否进入特权模式。
- 如果选中 **Authorize**，请在 **Authorize Password** 字段中输入密码以访问特权模式。

有关更多信息，请参阅使用新连接插件移植 **Ansible Network Playbook**。

10.4.18. 访问 ansible playbook 中的网络凭证

您可以从作业运行时环境获取用户名和密码参数：

```
vars:
  network:
    username: '{{ lookup("env", "ANSIBLE_NET_USERNAME") }}'
    password: '{{ lookup("env", "ANSIBLE_NET_PASSWORD") }}'
```

10.4.19. OpenShift 或 Kubernetes API Bearer Token 凭证类型

选择此凭证类型来创建实例组以指向 **Kubernetes** 或 **OpenShift** 容器。

如需更多信息，请参阅 *自动化控制器管理指南中的* 容器和实例组。

为容器凭证提供以下信息：

- **OpenShift** 或 **Kubernetes API** 端点 （必需）：用于连接到 **OpenShift** 或 **Kubernetes** 容器的端点。
- **API Authentication Bearer Token** （必需）：用于验证连接的令牌。
- 可选： 验证 **SSL** ：您可以检查这个选项以验证服务器的 **SSL/TLS** 证书是否有效且可信。使用内部或私有 *证书颁发机构* **(CA)**的环境必须保留此选项来禁用验证。

- 证书颁发机构数据 ：如果提供，请在粘贴证书时包括 **BEGIN CERTIFICATE** 和 **END CERTIFICATE** 行。

容器组是具有关联凭证的实例组类型，它允许连接到 **OpenShift** 集群。要设置容器组，您必须具有以下项目：

- 您可以开始进入的命名空间。虽然每个集群都有一个 **default** 命名空间，但您可以使用特定的命名空间。
- 具有角色的服务帐户，使其能够启动和管理此命名空间中的 **pod**。
- 如果您在私有 **registry** 中使用执行环境，并在自动化控制器中关联有容器 **registry** 凭证，则服务帐户还需要角色在命名空间中获取、创建和删除 **secret**。

  如果您不想为服务帐户授予这些角色，您可以预先创建 **ImagePullSecrets**，并在容器组的 **pod** 规格中指定它们。在这种情况下，执行环境不能关联 **Container Registry** 凭证，或者自动化控制器会尝试为您在命名空间中创建 **secret**。
- 与该服务帐户关联的令牌（**OpenShift** 或 **Kubernetes Bearer Token**）
- 与集群关联的 **CA** 证书

10.4.19.1. 在 Openshift 集群中创建服务帐户

在 **Openshift** 或 **Kubernetes** 集群中创建服务帐户，用于通过自动化控制器在容器组中运行作业。创建服务帐户后，其凭证以 **Openshift** 或 **Kubernetes API bearer** 令牌凭证的形式提供给自动化控制器。

创建服务帐户后，使用新服务帐户中的信息来配置自动化控制器。

流程

1. 要创建服务帐户，请下载并使用 示例服务帐户 并根据需要进行修改，以获取之前的凭证。

2. 应用 示例服务帐户 中的配置 ：

```
oc apply -f containergroup-sa.yml
```

3. 获取与服务帐户关联的 **secret** 名称：

```
export SA_SECRET=$(oc get sa containergroup-service-account -o json | jq
'.secrets[0].name' | tr -d '"')
```

4. 从 **secret** 获取令牌：

```
oc get secret $(echo ${SA_SECRET}) -o json | jq '.data.token' | xargs | base64 --decode
> containergroup-sa.token
```

5. 获取 **CA** 证书：

```
oc get secret $SA_SECRET -o json | jq '.data["ca.crt"]' | xargs | base64 --decode >
containergroup-ca.crt
```

6. 使用 **containergroup-sa.token** 和 **containergroup-ca.crt** 的内容，为容器组所需的 **OpenShift** 或 **Kubernetes API Bearer Token** 提供信息。

10.4.20. OpenStack 凭证类型

选择此凭证类型，以启用与 **OpenStack** 的云清单同步。

为 **OpenStack** 凭证提供以下信息：

- 用户名 ：用于连接 **OpenStack** 的用户名。
- **Password (API Key)** ：用于连接 **OpenStack** 的密码或 **API** 密钥。
- 主机（身份验证 **URL**） ：用于身份验证的主机。

- **Project (Tenant Name):** 用于 **OpenStack** 的租户名称或租户 **ID**。这个值通常与用户名相同。
- 可选： **Project (Domain Name)** ：提供与您的域关联的项目名称。
- 可选： 域名 ：提供用于连接 **OpenStack** 的 **FQDN**。

如果您有兴趣使用 **OpenStack** 云凭证，请参阅使用带有 云清单的 **Cloud Credentials**，其中包含一个示例 **playbook**。

10.4.21. Red Hat Ansible Automation Platform 凭证类型

选择这个凭证来访问另一个自动化控制器实例。

Ansible Automation Platform 凭证需要以下输入：

- **Red Hat Ansible Automation Platform:** 要连接的其他实例的基本 **URL** 或 **IP** 地址。
- 用户名 ：用于连接的用户名。
- **Password** ：用于连接的密码。
- **OAuth** 令牌 ：如果没有使用用户名和密码，请提供一个 **OAuth** 令牌来进行验证。

Ansible Automation Platform 的 **env injectors** 如下：

```
ManagedCredentialType(
   namespace='controller',

....
....
....

injectors={
```

```
    'env': {
        'TOWER_HOST': '{{host}}',
        'TOWER_USERNAME': '{{username}}',
        'TOWER_PASSWORD': '{{password}}',
        'TOWER_VERIFY_SSL': '{{verify_ssl}}',
        'TOWER_OAUTH_TOKEN': '{{oauth_token}}',
        'CONTROLLER_HOST': '{{host}}',
        'CONTROLLER_USERNAME': '{{username}}',
        'CONTROLLER_PASSWORD': '{{password}}',
        'CONTROLLER_VERIFY_SSL': '{{verify_ssl}}',
        'CONTROLLER_OAUTH_TOKEN': '{{oauth_token}}',
    }
```

10.4.21.1. 访问 Ansible Playbook 中的自动化控制器凭证

您可以从作业运行时环境获取主机、用户名和密码参数：

```
vars:
 controller:
   host: '{{ lookup("env", "CONTROLLER_HOST") }}'
   username: '{{ lookup("env", "CONTROLLER_USERNAME") }}'
   password: '{{ lookup("env", "CONTROLLER_PASSWORD") }}'
```

10.4.22. Red Hat Satellite 6 凭证类型

选择此凭证类型，以启用与 **Red Hat Satellite 6** 的云清单同步。

自动化控制器会根据用户界面中提示的字段写入 **Satellite** 配置文件。文件的绝对路径在以下环境变量中设置：

```
FOREMAN_INI_PATH
```

Satellite 凭证有以下所需的输入：

- **Satellite 6 URL** ：要连接的 **Satellite 6 URL** 或 **IP** 地址。
- 用户名 ：用于连接 **Satellite 6** 的用户名。
- **Password:** 用于连接到 **Satellite 6** 的密码。

10.4.23. Red Hat Virtualization 凭证类型

选择此凭据以启用自动化控制器来访问 **Ansible** 的 **oVirt4.py** 动态清单插件，该插件由 ***Red Hat Virtualization*** 管理。

自动化控制器为 **Red Hat Virtualization** 凭证使用以下环境变量。以下是用户界面中的字段：

```
OVIRT_URL
OVIRT_USERNAME
OVIRT_PASSWORD
```

为 **Red Hat Virtualization** 凭证提供以下信息：

- 主机（身份验证 **URL**）：要连接的主机 **URL** 或 **IP** 地址。要与清单同步，凭证 **URL** 需要包含 **ovirt-engine/api** 路径。
- 用户名：用于连接 **oVirt4** 的用户名。这必须包含域配置集才能成功，例如 **username@ovirt.host.com**。
- **Password**：用于连接的密码。
- 可选： **CA** 文件：提供 **oVirt** 证书文件的绝对路径（它可能以 **.pem**、**.cer** 和 **.crt** 扩展结尾，但最好是 **.pem** 以实现一致性）

10.4.23.1. 访问 Ansible Playbook 中的虚拟化凭证

您可以从作业运行时环境获取 **Red Hat Virtualization** 凭证参数：

```
vars:
  ovirt:
    ovirt_url: '{{ lookup("env", "OVIRT_URL") }}'
    ovirt_username: '{{ lookup("env", "OVIRT_USERNAME") }}'
    ovirt_password: '{{ lookup("env", "OVIRT_PASSWORD") }}'
```

Red Hat Virtualization 的文件 和 **env injectors** 如下：

```
ManagedCredentialType(
```

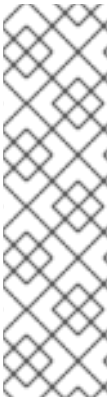
```
    namespace='rhv',

....
....
....

injectors={
        # The duplication here is intentional; the ovirt4 inventory plugin
        # writes a .ini file for authentication, while the ansible modules for
        # ovirt4 use a separate authentication process that support
        # environment variables; by injecting both, we support both
        'file': {
            'template': '\n'.join(
                [
                    '[ovirt]',
                    'ovirt_url={{host}}',
                    'ovirt_username={{username}}',
                    'ovirt_password={{password}}',
                    '{% if ca_file %}ovirt_ca_file={{ca_file}}{% endif %}',
                ]
            )
        },
        'env': {'OVIRT_INI_PATH': '{{tower.filename}}', 'OVIRT_URL': '{{host}}',
'OVIRT_USERNAME': '{{username}}', 'OVIRT_PASSWORD': '{{password}}'},
    },
)
```

10.4.24. 源控制凭证类型

源控制凭证 与项目一起使用，从远程修订控制系统（如 **Git** 或 **Subversion**）克隆和更新本地源代码存储库。

源控制凭证需要以下输入：

- 用户名 ：与源控制系统结合使用的用户名。
- **Password** ：与源控制系统结合使用的密码。
- **SCM** 私钥 ：复制或拖放要用于通过 **SSH** 向源控制系统验证用户的实际 **SSH** 私钥。
- **Private Key Passphrase** ：如果使用的 **SSH** 私钥受密码保护，您可以为私钥配置密钥密码。



注意

您不能将 **Source Control** 凭证配置为 启动时提示。

如果您使用 **GitHub** 帐户作为 **Source Control** 凭证，且您的帐户上启用了 *两个 **Factor Authentication*** **(2FA)**，您必须在 **password** 字段中使用您的个人访问令牌而不是您的帐户密码。

10.4.25. Thycotic DevOps Secrets Vault 凭证类型

这被视为 **secret** 管理功能的一部分。

如需更多信息，请参阅 Thycotic DevOps Secrets Vault。

10.4.26. Thycotic secret 服务器凭证类型

这被视为 **secret** 管理功能的一部分。

如需更多信息，请参阅 Thycotic Secret Server。

10.4.27. Ansible Vault 凭据类型

选择此凭证类型，以启用与 **Ansible Vault** 的清单同步。

如果应用多 **Vault** 凭证，则 **Vault** 凭证需要 **Vault** 密码 和可选的 **Vault** 标识符。

如需有关多 **Vault** 支持的更多信息，请参阅 *自动化控制器管理指南中的* 多 **Vault** 凭据 部分。

您可以通过选择 **Prompt on launch** 将自动化控制器配置为在启动时要求用户输入密码。

当您选择 **Prompt on Launch** 时，作业启动时会打开一个对话框，提示用户输入密码。

警告

在调度作业中使用的凭证不能配置为 **Prompt on launch**。

如需有关 **Ansible Vault** 的更多信息，请参阅使用 **Ansible vault** 保护敏感数据。

10.4.28. VMware vCenter 凭证类型

选择此凭证类型以启用与 **VMware vCenter** 的清单同步。

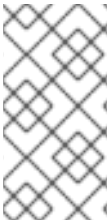
自动化控制器为 **VMware vCenter** 凭证使用以下环境变量：

```
VMWARE_HOST
VMWARE_USER
VMWARE_PASSWORD
VMWARE_VALIDATE_CERTS
```

用户界面中会提示这些字段。

VMware 凭证需要以下输入：

- **vCenter Host** ：要连接的 **vCenter** 主机名或 **IP** 地址。
- 用户名 ：用于连接到 **vCenter** 的用户名。
- **Password:** 用于连接到 **vCenter** 的密码。



注意

如果 **VMware** 客户机工具没有在实例上运行，**VMware** 清单同步不会返回该实例的 **IP** 地址。

10.4.28.1. 在 ansible playbook 中访问 VMware vCenter 凭证

您可以从作业运行时环境获取 **VMware vCenter** 凭证参数：

```
vars:
  vmware:
    host: '{{ lookup("env", "VMWARE_HOST") }}'
    username: '{{ lookup("env", "VMWARE_USER") }}'
    password: '{{ lookup("env", "VMWARE_PASSWORD") }}'
```

10.5. 在 PLAYBOOK 中使用自动化控制器凭证

以下 **playbook** 是在 **playbook** 中使用自动化控制器凭证的示例。

```
- hosts: all

  vars:
    machine:
      username: '{{ ansible_user }}'
      password: '{{ ansible_password }}'
    controller:
      host: '{{ lookup("env", "CONTROLLER_HOST") }}'
      username: '{{ lookup("env", "CONTROLLER_USERNAME") }}'
      password: '{{ lookup("env", "CONTROLLER_PASSWORD") }}'
    network:
      username: '{{ lookup("env", "ANSIBLE_NET_USERNAME") }}'
      password: '{{ lookup("env", "ANSIBLE_NET_PASSWORD") }}'
    aws:
      access_key: '{{ lookup("env", "AWS_ACCESS_KEY_ID") }}'
      secret_key: '{{ lookup("env", "AWS_SECRET_ACCESS_KEY") }}'
      security_token: '{{ lookup("env", "AWS_SECURITY_TOKEN") }}'
    vmware:
      host: '{{ lookup("env", "VMWARE_HOST") }}'
      username: '{{ lookup("env", "VMWARE_USER") }}'
      password: '{{ lookup("env", "VMWARE_PASSWORD") }}'
    gce:
      email: '{{ lookup("env", "GCE_EMAIL") }}'
      project: '{{ lookup("env", "GCE_PROJECT") }}'
    azure:
      client_id: '{{ lookup("env", "AZURE_CLIENT_ID") }}'
      secret: '{{ lookup("env", "AZURE_SECRET") }}'
      tenant: '{{ lookup("env", "AZURE_TENANT") }}'
      subscription_id: '{{ lookup("env", "AZURE_SUBSCRIPTION_ID") }}'
```

```
tasks:
  - debug:
      var: machine

  - debug:
      var: controller

  - debug:
      var: network

  - debug:
      var: aws

  - debug:
      var: vmware

  - debug:
      var: gce

  - shell: 'cat {{ gce.pem_file_path }}'
    delegate_to: localhost

  - debug:
      var: azure
```

使用 **'delegate_to'** 和任何查找变量

```
- command: somecommand
  environment:
    USERNAME: '{{ lookup("env", "USERNAME") }}'
    PASSWORD: '{{ lookup("env", "PASSWORD") }}'
  delegate_to: somehost
```

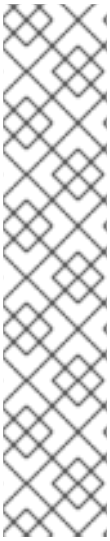
第 11 章 自定义凭证类型

作为系统管理员，您可以使用 **YAML** 或类 **JSON** 的定义以标准格式定义自定义凭证类型。您可以定义一个与现有凭证类型类似的自定义凭证类型。例如，自定义凭证类型可将第三方 **Web** 服务的 **API** 令牌注入到 **playbook** 或自定义清单脚本的环境变量中。

自定义凭证支持以下注入身份验证信息的方法：

- 环境变量
- **Ansible** 额外变量
- 基于文件的模板，这意味着生成包含凭证值的 **.ini** 或 **.conf** 文件

您可以将一个 **SSH** 和多个云凭证附加到作业模板。每个云凭证都必须是不同的类型。只允许每种凭证类型中的一个凭证。**Vault** 凭证和机器凭证是单独的实体。



注意

- 在创建新凭证类型时，您必须避免在 **extra_vars**、**env** 和 **file** 命名空间中发生冲突。
- 环境变量或额外变量名称不能以 **ANSIBLE_** 开头，因为它们是保留的。
- 您必须具有系统管理员（超级用户）权限才能创建和编辑凭证类型 **(CredentialType)**，并能够查看 **CredentialType.injection** 字段。

11.1. 从集合中获取内容

"受管**"**凭证类型 **kind=galaxy** 代表在运行项目更新时，在 **requirements.yml** 中定义的集合的内容源。内容源示例包括 **galaxy.ansible.com**、**console.redhat.com** 或内部自动化中心。这个新凭证类型代表了在项目更新运行 **ansible-galaxy** 集合安装 时构建环境变量所需的 **URL** 和（可选）身份验证详情，如 **Ansible** 文档 配置 **ansible-galaxy** 客户端 中所述。它有字段直接映射到公开给 **Ansible Galaxy CLI** 的配置选项，如每个服务器。

API 中的端点反映了在机构级别这些凭证的排序列表：

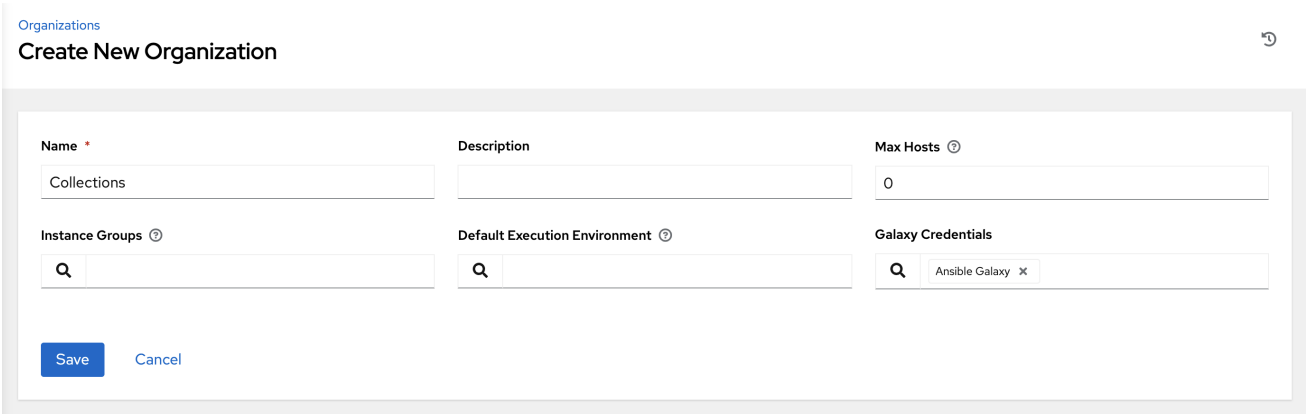
```
/api/v2/organizations/N/galaxy_credentials/
```

当自动化控制器安装迁移现有的面向 **Galaxy** 的设置值时，会创建升级后正确的凭证并附加到每个机构。升级到最新版本后，在升级前存在的每个机构现在都有与其关联的一个或多个 **"Galaxy"** 凭证的列表。

另外，这些设置在升级后无法从 **/api/v2/settings/jobs/** 端点可见（或编辑）。

即使 **galaxy.ansible.com** 不是机构列表中的第一个凭证，自动化控制器也会继续直接从公共 **Galaxy** 获取角色。全局 **Galaxy** 设置不再在作业级别配置，而是在用户界面中的机构级别配置。

机构的 **Add** 和 **Edit** 窗口具有一个可选的 **Credential lookup** 字段，用于 **kind=galaxy** 的凭证。



务必要将这些凭证的顺序指定为同步和查找内容的优先级。如需更多信息，请参阅创建机构。

有关如何使用集合设置项目的更多信息，请参阅使用带有自动化中心 的集合。

11.2. 后向兼容 API 注意事项

支持 **API** 版本 **2 (api/v2/)**意味着作业模板与凭证有一对多的关系（包括多组支持）。

您可以过滤 **v2 API** 的凭证：

```
curl "https://controller.example.org/api/v2/credentials/?credential_type__namespace=aws"
```

在 **V2 Credential Type** 模型中，关系定义如下：

机器	SSH
Vault	Vault
网络	设置环境变量，如 **ANSIBLE_NET_AUTHORIZE**
SCM	源控制
云	EC2、AWS
云	很多其他服务
Insights	Insights
Galaxy	galaxy.ansible.com, console.redhat.com
Galaxy	内部自动化中心

11.3. 内容验证

自动化控制器使用 **GNU Privacy Guard (GPG)**验证内容。

如需更多信息，请参阅 **GNU Privacy Handbook**。

11.4. 凭证类型入门

在导航面板中，选择 **Administration → Credential Types**。如果没有创建自定义凭证类型，则 凭证类型 会提示您添加一个。

如果创建了凭证类型，本页会显示现有和可用的凭证类型的列表。

要查看有关凭证类型的更多信息，请点凭证的名称或编辑

图标。

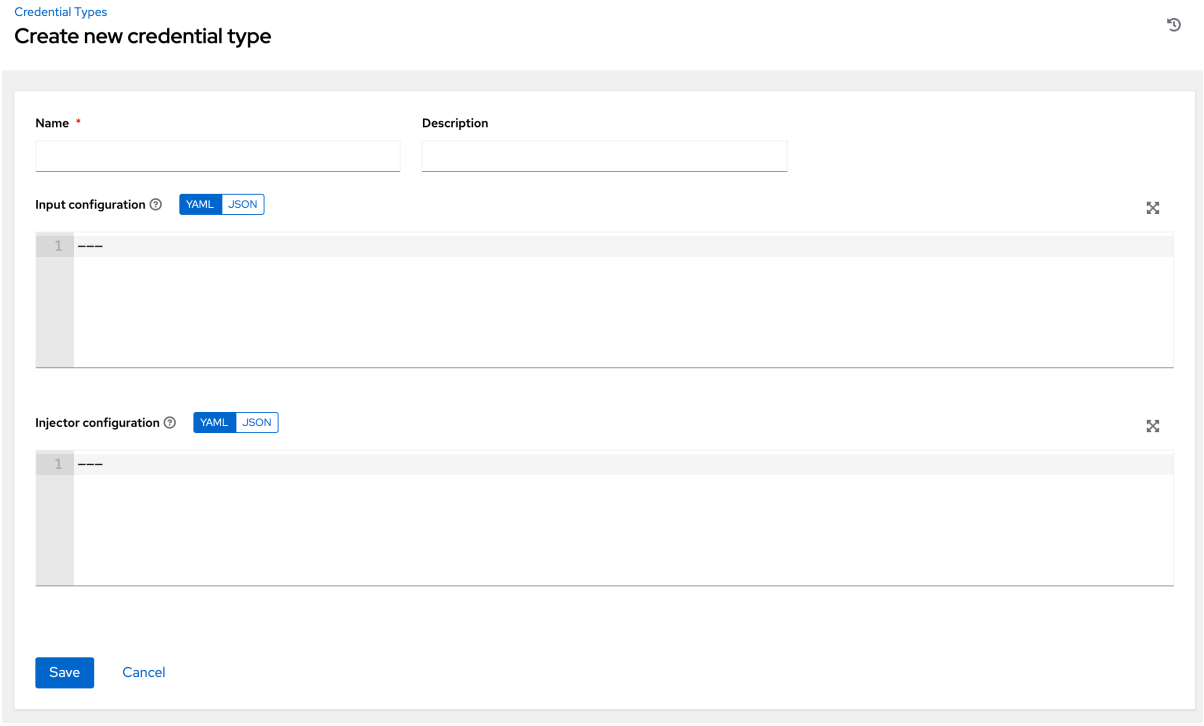
每个凭证类型在 **Input Configuration** 字段和 **Injector Configuration** 字段中显示自己的唯一配置（如果适用）。配置字段支持 **YAML** 和 **JSON** 格式。

11.5. 创建新凭证类型

要创建新凭证类型，请执行以下操作：

流程

1. 在 **Credential Types** 视图中，点 **Add**。



2. 在 **Name** 和 **Description** 字段中输入相关详情。

注意

在创建新凭证类型时，请不要为 **INPUT** 和 **INJECTOR** 名称和 **ID** 使用以 **ANSIBLE_** 开头的 保留变量名称，因为它们对于自定义凭证类型无效。

3. 在 **Input Configuration** 字段中，指定一个输入模式，为该类型定义一组排序字段。格式可以是 **YAML** 或 **JSON**：

YAML

```
fields:
  - type: string
    id: username
    label: Username
  - type: string
    id: password
    label: Password
    secret: true
required:
  - username
  - password
```

在 **YAML** 页面 查看更多 **YAML** 示例。

JSON

```
{
"fields": [
  {
  "type": "string",
  "id": "username",
  "label": "Username"
  },
  {
  "secret": true,
  "type": "string",
  "id": "password",
  "label": "Password"
   }
  ],
 "required": ["username", "password"]
}
```

请参阅 **JSON** 网站 查看更多 **JSON** 示例。

以下 **JSON** 格式的配置显示了每个字段以及它们的使用方式：

```
{
  "fields": [{
    "id": "api_token",    # required - a unique name used to reference the field value

    "label": "API Token", # required - a unique label for the field

    "help_text": "User-facing short text describing the field.",

    "type": ("string" | "boolean")   # defaults to 'string'

    "choices": ["A", "B", "C"]   # (only applicable to `type=string`)

    "format": "ssh_private_key"  # optional, can be used to enforce data format validity
                          for SSH private key data (only applicable to `type=string`)

    "secret": true,       # if true, the field value will be encrypted

    "multiline": false    # if true, the field should be rendered as multi-line for input entry
                    # (only applicable to `type=string`)
  },{
    # field 2...
  },{
    # field 3...
  }],

  "required": ["api_token"]   # optional; one or more fields can be marked as required
  },
```

当 **type=string** 时，字段可以选择性地指定多个选择选项：

```
{
  "fields": [{
      "id": "api_token",    # required - a unique name used to reference the field value
      "label": "API Token", # required - a unique label for the field
      "type": "string",
      "choices": ["A", "B", "C"]
   }]
  },
```

4.

在 **Injector Configuration** 字段中输入环境变量或额外变量，用于指定凭证类型可注入的值。格式可以是 **YAML** 或 **JSON** （请参阅上一步中的示例）。

以下 **JSON** 格式的配置显示了每个字段以及它们的使用方式：

```
{
  "file": {
```

```
    "template": "[mycloud]\ntoken={{ api_token }}"
  },
  "env": {
    "THIRD_PARTY_CLOUD_API_TOKEN": "{{ api_token }}"
  },
  "extra_vars": {
    "some_extra_var": "{{ username }}:{{ password }}"
  }
}
```

凭证类型也可以生成临时文件来支持 **.ini** 文件或证书或密钥数据：

```
{
  "file": {
    "template": "[mycloud]\ntoken={{ api_token }}"
  },
  "env": {
    "MY_CLOUD_INI_FILE": "{{ tower.filename }}"
  }
}
```

在本例中，自动化控制器会编写一个具有以下内容的临时文件：

```
[mycloud]\ntoken=SOME_TOKEN_VALUE
```

生成的文件的绝对路径存储在名为 **MY_CLOUD_INI_FILE** 的环境变量中。

以下是引用自定义凭证模板中的很多文件的示例：

输入

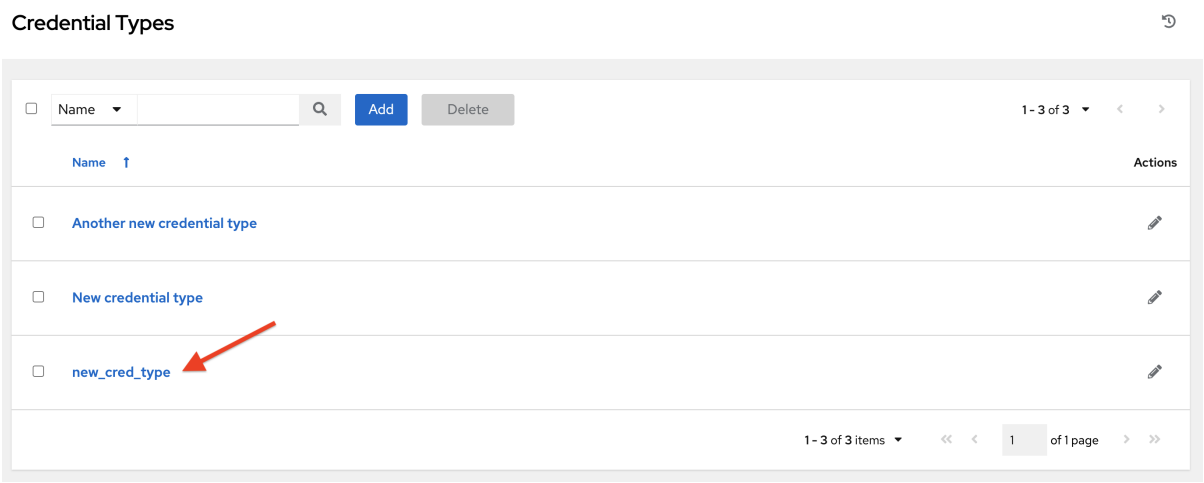
```
{
  "fields": [{
    "id": "cert",
    "label": "Certificate",
    "type": "string"
  },{
    "id": "key",
    "label": "Key",
    "type": "string"
  }]
}
```

injectors

```
{
  "file": {
    "template.cert_file": "[mycert]\n{{ cert }}",
    "template.key_file": "[mykey]\n{{ key }}"
},
"env": {
    "MY_CERT_INI_FILE": "{{ tower.filename.cert_file }}",
    "MY_KEY_INI_FILE": "{{ tower.filename.key_file }}"
}
}
```

5. 点击 **Save**。

您新创建的凭证类型显示在凭证类型列表中：



6. 点 **Edit**

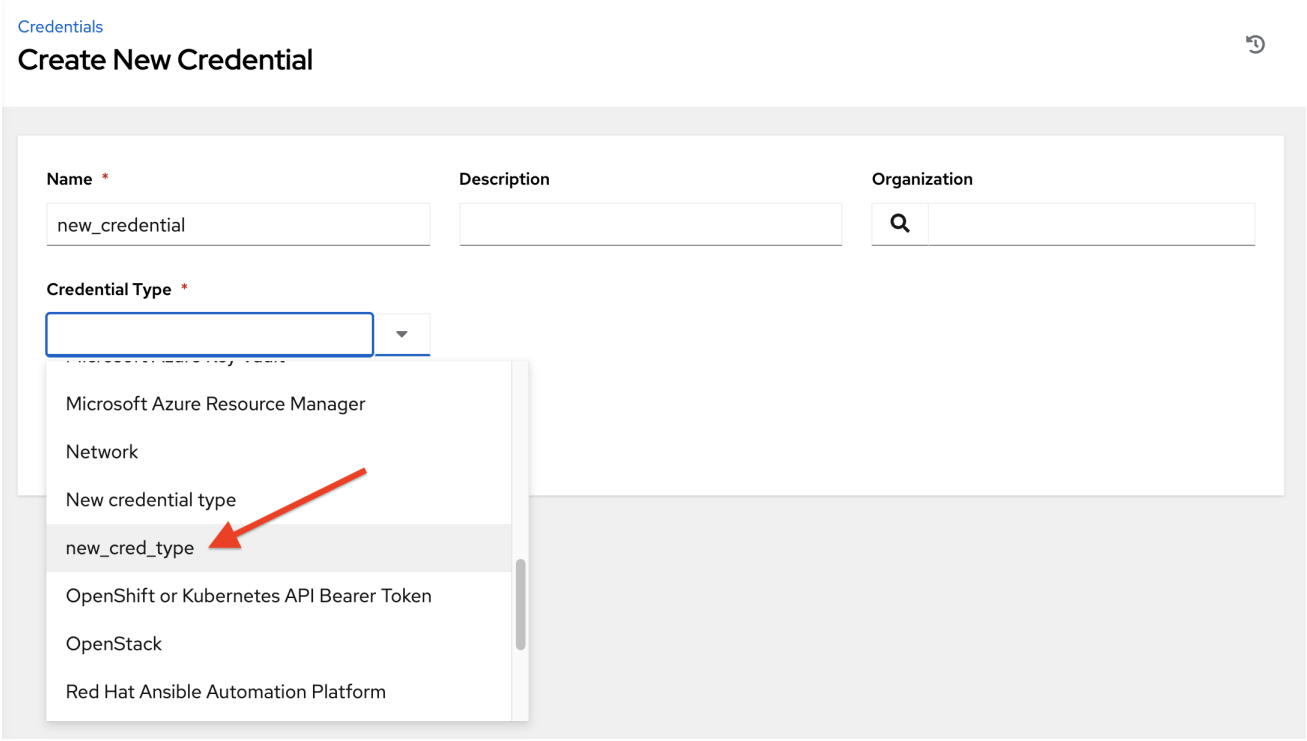
图标修改凭证类型选项。

注意

在 **Edit** 屏幕中，您可以修改详情或**删**除凭证。如果 **Delete** 选项被禁用，这意味着凭证正在使用凭证类型，且您必须在**删**除它前从所有使用的凭证中**删**除凭证类型。

验证

- 验证在创建新凭证时是否可从 **Credential Type** 选择窗口中选择新创建的凭证类型：



其他资源

有关如何创建新凭证的详情，请参考 创建凭证。

第 12 章 SECRET 管理系统

用户和系统管理员上传机器和云凭证，以便自动化可以代表他们访问机器和外部服务。默认情况下，敏感凭证值，如 **SSH** 密码、**SSH** 私钥和云服务的 **API** 令牌在加密后存储在数据库中。

使用由凭证插件支持的外部凭证，您可以将凭证字段（如密码或 **SSH** 私钥）映射到存储在 **secret** 管理系统中的值，而不是直接将它们提供给自动化控制器。

自动化控制器提供了一个 **secret** 管理系统，其中包含以下的集成：

- **AWS Secrets Manager Lookup**
- **Centrify Vault Credential Provider Lookup**
- ***CyberArk Central Credential Provider* Lookup (CCP)**
- **CyberArk Conjur Secrets Manager Lookup**
- **HashiCorp Vault *Key-Value* Store (KV)**
- **HashiCorp Vault SSH Secrets Engine**
- **Microsoft Azure *Key Management System* (KMS)**
- **Thycotic DevOps Secrets Vault**
- **Thycotic Secret Server**

在运行需要它们的 **playbook** 之前，会获取这些外部 **secret** 值。

其他资源

有关在用户界面中指定 **secret** 管理系统凭证的更多信息，请参阅 凭证。

12.1. 配置和链接 SECRET 查找

从第三方系统拉取 **secret** 时，您要将凭证字段链接到外部系统。要将凭证字段链接到存储在外部系统中的值，请选择与该系统对应的外部凭证，并提供 元数据 来查找所需的值。元数据输入字段是源凭证的外部凭证类型定义的一部分。

自动化控制器为开发人员、集成商、系统管理员和电源用户提供了一个凭证插件界面，能够添加新的外部凭证类型来扩展它以支持其他 **secret** 管理系统。

使用以下步骤使用自动化控制器来配置和使用每个支持的第三方 **secret** 管理系统。

流程

1. 创建一个外部凭证来使用 **secret** 管理系统进行身份验证。至少，为外部凭证指定一个名称，并为 **Credential Type** 字段选择以下之一：

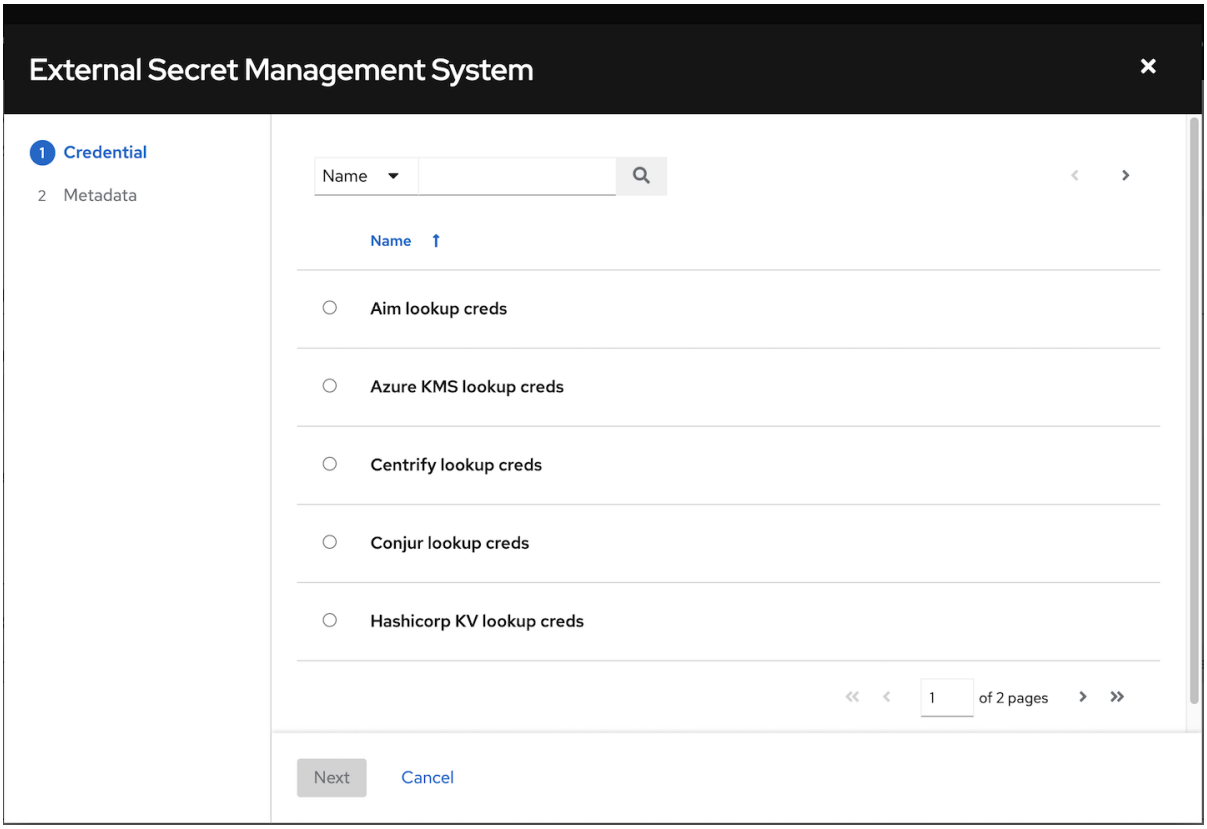
   - **AWS Secrets Manager Lookup**
   - **Centrify Vault Credential Provider Lookup**
   - **CyberArk Central Credential Provider (CCP) Lookup**
   - **CyberArk Conjur Secrets Manager Lookup**
   - **HashiCorp Vault Secret Lookup**
   - **HashiCorp Vault Signed SSH**

- **Microsoft Azure Key Vault**
- **Thycotic DevOps Secrets Vault**
- **Thycotic Secret Server**

  在本例中，***Demo Credential*** 是目标凭据。

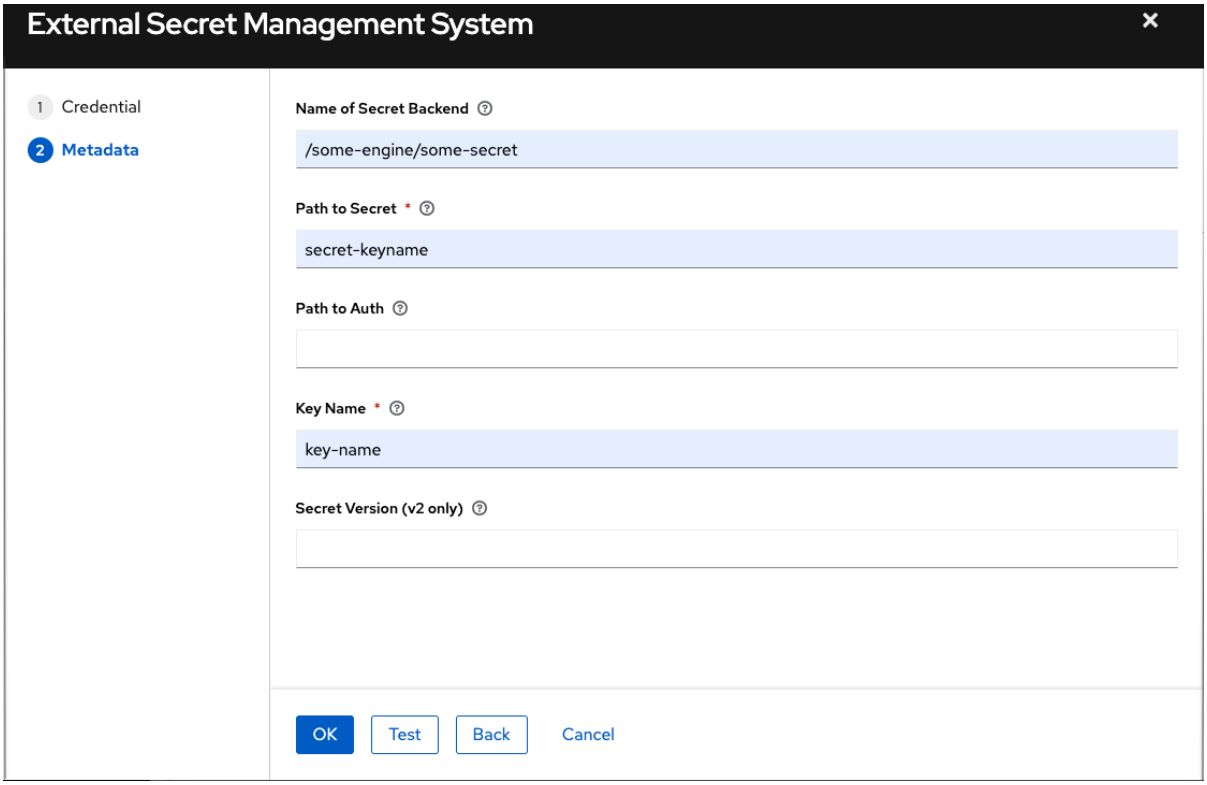
2. 对于遵循您要链接到外部凭证的 **Type Details** 区域的任何字段，点输入字段中的键
   
   图标将一个或多个输入字段链接到外部凭证，以及元数据，以便在外部系统中找到 **secret**。

3. 选择用于检索您的 **secret** 信息的输入源。



4. 选择您要链接到的凭据，然后单击下一步。这会进入输入源的 **Metadata** 选项卡。本例演示了 **HashiVault Secret Lookup** 的元数据提示。元数据特定于您选择的输入源。

如需更多信息，请参阅凭证输入源表的元数据。



5. 点 **Test** 来验证与 **secret** 管理系统的连接。如果查找失败，则会显示类似如下的错误消息：

6. 点 **OK**。返回目标凭证的 **Details** 屏幕。

7. 重复这些步骤，从第 **3** 步开始完成目标凭证的其余输入字段。通过以这种方式链接信息，自动化控制器会从第三方管理系统检索敏感信息，如用户名、密码、密钥、证书和令牌，并使用该数据填充目标凭证表单的其余字段。

8. 如有必要，请为不使用链接作为检索敏感信息方法的那些字段手动提供任何信息。有关每个字段的更多信息，请参阅相应的 凭证类型。

9. 点击 **Save**。

其他资源

如需更多信息，请参阅凭证插件的开发 文档。

12.1.1. 凭证输入源的原始

输入源 的元数据 选项卡所需的信息。

AWS Secrets Manager Lookup

元数据	描述
AWS Secrets Manager 区域（必需）	secret 管理器所在的区域。
AWS Secret Name （必需）	指定 AWS 访问密钥生成的 AWS secret 名称。

Centrify Vault Credential Provider Lookup

元数据	描述
帐户名称（必需）	与 Centrify Vault 关联的系统帐户或域名。
系统名称	指定 Centrify 门户使用的名称。

CyberArk Central Credential Provider Lookup

元数据	描述
对象查询（必需）	对象查找查询。
对象查询格式	选择 **Exact** 用于特定的 secret 名称，或使用 **Regexp** 用于动态生成名称的 secret。
对象属性	指定要返回的属性的名称。例如，**UserName** 或 默认内容以外的 地址。
原因	如果对象策略需要，请提供签出 secret 的原因，如 CyberArk 日志。

CyberArk Conjur Secrets Lookup

元数据	描述
secret 标识符	secret 的标识符。
secret 版本	如果需要，请指定 secret 的版本，否则保留为空，以使用最新版本。

HashiVault Secret Lookup

元数据	描述
secret 后端的名称	指定要使用的 KV 后端的名称。将它留空，以使用 **Path to Secret** 字段的第一个路径片段。
Path to Secret（必需）	指定存储 secret 信息的路径，例如 /**path/username**。
Key Name （必需）	用来查找 secret 的密钥名称。
Secret Version（只适用于 V2）	如果需要，请指定版本，否则为空，以使用最新版本。

HashiCorp Signed SSH

元数据	描述
Unsigned Public Key（必需）	指定您要签名的证书的公钥。它需要存在于目标主机的授权密钥文件中。
Path to Secret（必需）	指定存储 secret 信息的路径，例如 /**path/username**。
Role Name（必需）	角色是存储在 Hashi vault 中的 SSH 设置和参数的集合。通常，您可以指定一些不同的特权或超时，例如：因此，您可以有一个允许为 root 签名的证书的角色，例如：
Valid Principals	指定非默认用户（或用户），您要请求 vault 为存储的密钥授权证书。Hashi vault 有一个默认用户，用于为其签名，如 ec2-user。

Microsoft Azure KMS

元数据	描述
Secret Name（必需）	Microsoft Azure 的 Key vault 应用程序中引用的 secret 名称。
secret 版本	如果需要，请指定 secret 的版本，否则保留为空，以使用最新版本。

Thycotic DevOps Secrets Vault

元数据	描述
Secret 路径（必需）	指定存储 secret 信息的路径，如 /path/username。

Thycotic Secret Server

元数据	描述
Secret ID（必需）	secret 的标识符。
Secret 字段	指定要从 secret 中使用的字段。

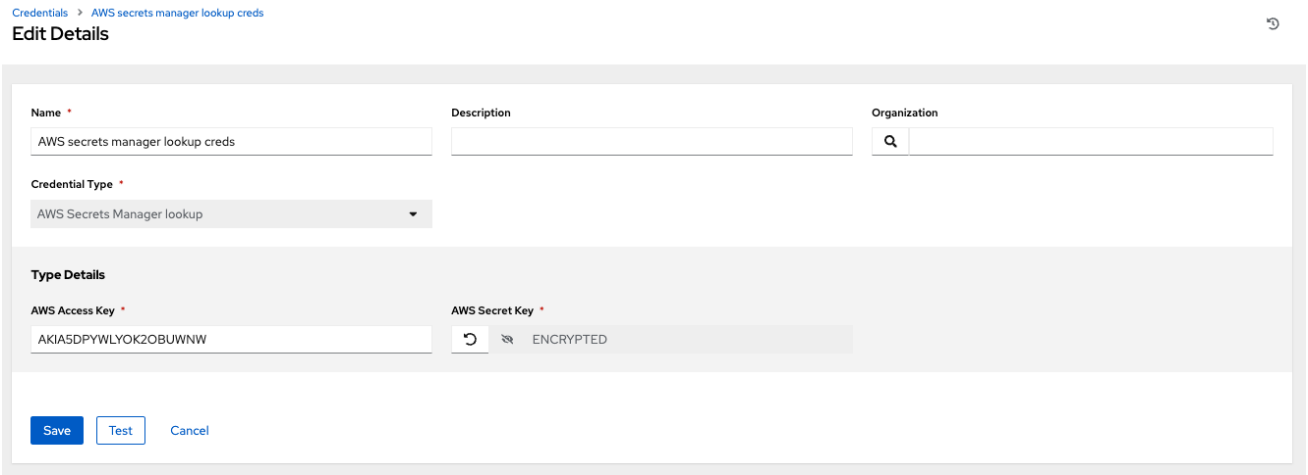
12.1.2. AWS Secrets Manager Lookup

此插件可让 **Amazon Web Services** 用作凭证输入源，以便从 **Amazon Web Services Secrets Manager** 中拉取 **secret**。**AWS Secrets Manager** 提供与 **Microsoft Azure Key Vault** 类似的服务，**AWS** 集合为它提供了一个查找插件。

当为凭证类型选择 **AWS Secret Manager** 查找时，请提供以下元数据来配置您的查找：

- **AWS Access Key** （必需）：提供用于与 **AWS** 密钥管理系统通信的访问密钥
- **AWS Secret Key** （必需）：提供 **AWS IAM** 控制台获取的 **secret**

以下是配置的 **AWS Secret Manager** 凭证的示例。



12.1.3. Centrify Vault Credential Provider Lookup

您需要运行 **Centrify Vault web** 服务来存储此集成的 **secret**。当您为凭证类型选择 **Centrify Vault Credential Provider Lookup** 时，请提供以下元数据来配置查找：

- **Centrify Tenant URL** （必需）：提供用于与 **Centrify** 的 **secret** 管理系统通信的 **URL**
- **Centrify API User** （必需）：指定用户名

- **Centrify API Password** （必需）：提供密码

- **OAuth2** 应用 **ID** ：指定与 **OAuth2** 客户端关联的标识符

- **OAuth2 Scope** ：指定 **OAuth2** 客户端的范围

12.1.4. CyberArk Central Credential Provider (CCP) Lookup

必须运行 **CyberArk Central Credential Provider Web** 服务，以存储 **secret** 才能使此集成正常工作。当您为凭证类型选择 **CyberArk Central Credential Provider Lookup** 时，请提供以下元数据来配置查找：

- **CyberArk CCP URL** （必需）：提供用于与 **CyberArk CCP** 的 **secret** 管理系统通信的 **URL**。它必须包含 **URL** 方案，如 **http** 或 **https**。

- 可选： **Web Service ID** ：指定 **Web** 服务的标识符。将此字段留空默认为 **AIMWebService**。

- 应用程序 **ID** （必需）：指定 **CyberArk CCP** 服务提供的标识符。

- 客户端密钥 ：如果由 **CyberArk** 提供，则粘贴客户端密钥。

- 客户端证书 ：如果由 **CyberArk** 提供，请在粘贴证书时包括 **BEGIN CERTIFICATE** 和 **END CERTIFICATE** 行。

- 验证 **SSL** 证书 ：此选项仅在 **URL** 使用 **HTTPS** 时才可用。检查这个选项以验证服务器的 **SSL/TLS** 证书是否有效且可信。对于使用内部或私有 **CA** 的环境，请保留此选项来禁用验证。

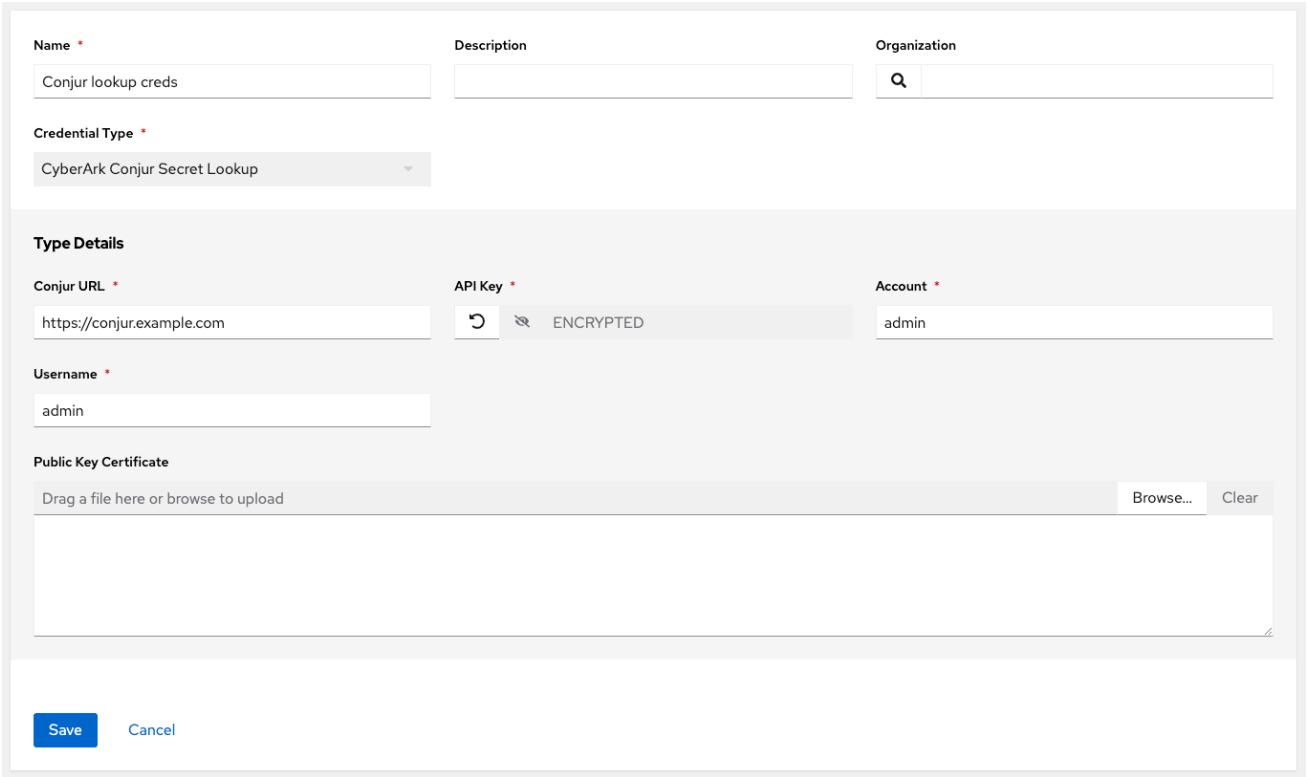
12.1.5. CyberArk Conjur Secrets Manager Lookup

使用针对目标的 **Conjur Cloud** 租户，配置 **CyberArk Conjur Secrets Lookup** 外部管理系统凭证插件。

当您为凭证类型选择 **CyberArk Conjur Secrets Manager Lookup** 时，请提供以下元数据来配置您的查找：

- **Conjur URL** （必需）：提供用于与 **CyberArk Conjur** 的 **secret** 管理系统通信的 **URL**。这必须包含 **URL** 方案，如 **http** 或 **https**。
- **API Key** （必需）：提供 **Conjur admin** 提供的密钥
- **account** （必需）：机构的帐户名称
- **username** （必需）：此服务的具体经过身份验证的用户
- 公钥证书 ：如果由 **CyberArk** 提供，请在粘贴公钥时包括 **BEGIN CERTIFICATE** 和 **END CERTIFICATE** 行

以下是配置的 **CyberArk Conjur** 凭证的示例。



12.1.6. HashiCorp Vault Secret Lookup

当您为凭证类型选择 **HashiCorp Vault Secret Lookup** 时，请提供以下元数据来配置查找：

- **Server URL** （必需）：提供用于与 **HashiCorp Vault** 的 **secret** 管理系统通信的 **URL**。
- **Token:** 指定用于验证 **HashiCorp** 的服务器的访问令牌。
- **CA** 证书 ：指定用于验证 **HashiCorp** 的服务器的 **CA** 证书。
- **Approle Role_ID** ：如果使用 **Appprole** 进行身份验证，请指定 **ID**。
- **Approle Secret_ID** ：为 **Approle** 身份验证指定对应的 **secret ID**。
- 客户端证书 ：在使用 **TLS** 身份验证方法时指定 **PEM** 编码的客户端证书，包括 **Hashicorp Vault** 所需的中间证书。
- 客户端证书 密钥 ：在使用 **TLS** 身份验证方法时，指定 **PEM** 编码的证书私钥。
- **TLS Authentication Role** ：在 **Hashicorp Vault** 中指定角色或证书名称，在使用 **TLS** 验证方法时对应于您的客户端证书。如果没有提供，**Hashicorp Vault** 会尝试自动匹配证书。
- 命名空间名称 ：指定命名空间名称（仅限 **Hashicorp Vault enterprise**）。
- **Kubernetes** 角色 ：使用 **Kubernetes** 身份验证时指定角色名称。
- 用户名 ：输入用于验证此服务的用户的用户名。
- **Password** ：输入与要验证此服务的用户关联的密码。

- **Auth** 的路径 ：如果不是默认路径 /**approle**，则指定路径。

- **API Version** （必需）：选择 **v1** 进行静态查找，选择 **v2** 进行版本化查找。

LDAP 身份验证需要在 **HashiCorp** 的 **Vault UI** 中配置 **LDAP**，以及添加到用户的策略。**Cubbyhole** 是默认 **secret** 挂载的名称。如果您有正确的权限，您可以创建其他挂载并将键值写入这些值。

要测试查找，请创建另一个使用 **Hashicorp Vault** 查找的凭证。

其他资源

有关 **LDAP** 验证方法及其字段的详情，请查看 **LDAP** 身份验证方法的 **Vault** 文档。

有关 **Approle** 验证方法及其字段的更多信息，请参阅 **AppRole auth** 方法的 **Vault** 文档。

有关 **userpass** 身份验证方法及其字段的更多信息，请参阅 **userpass auth** 方法的 **Vault** 文档。

有关 **Kubernetes auth** 方法及其字段的更多信息，请参阅 **Kubernetes auth** 方法的 **Vault** 文档。

有关 **TLS** 证书 **auth** 方法及其字段的更多信息，请参阅 **TLS** 证书 **auth** 方法的 **Vault** 文档。

12.1.7. HashiCorp Vault Signed SSH

当您为凭证类型选择 **HashiCorp Vault Signed SSH** 时，请提供以下元数据来配置查找：

- **Server URL** （必需）：提供用于与 **HashiCorp Signed SSH** 的 **secret** 管理系统通信的 **URL**。

- **Token:** 指定用于验证 **HashiCorp** 的服务器的访问令牌。

- **CA 证书** ：指定用于验证 **HashiCorp** 的服务器的 **CA** 证书。
- **Approle Role_ID** ：指定 **Approle** 身份验证的 **ID**。
- **Approle Secret_ID** ：为 **Approle** 身份验证指定对应的 **secret ID**。
- 客户端证书 ：在使用 **TLS** 身份验证方法时指定 **PEM** 编码的客户端证书，包括 **Hashicorp Vault** 所需的中间证书。
- 客户端证书 密钥 ：在使用 **TLS** 身份验证方法时，指定 **PEM** 编码的证书私钥。
- **TLS Authentication Role** ：在 **Hashicorp Vault** 中指定角色或证书名称，在使用 **TLS** 验证方法时对应于您的客户端证书。如果没有提供，**Hashicorp Vault** 会尝试自动匹配证书。
- 命名空间名称 ：指定命名空间名称（仅限 **Hashicorp Vault enterprise**）。
- **Kubernetes** 角色 ：使用 **Kubernetes** 身份验证时指定角色名称。
- 用户名 ：输入用于验证此服务的用户的用户名。
- **Password** ：输入与要验证此服务的用户关联的密码。
- **Auth** 的路径 ：如果不是默认路径 **/approle**，则指定路径。

其他资源

有关 **Approle** 验证方法及其字段的更多信息，请参阅 **Approle Auth** 方法的 **Vault** 文档。

有关 **Kubernetes** 身份验证方法及其字段的更多信息，请参阅 **Kubernetes auth** 方法的 **Vault** 文档。

有关 **TLS** 证书 **auth** 方法及其字段的更多信息，请参阅 **TLS** 证书 **auth** 方法的 **Vault** 文档。

12.1.8. Microsoft Azure Key Vault

当您为凭证类型选择 **Microsoft Azure Key Vault** 时，请提供以下元数据来配置查找：

- **Vault URL (DNS Name)** （必需）：提供用于与 **Microsoft Azure** 的密钥管理系统通信的 **URL**
- 客户端 **ID** （必需）：提供 **Microsoft Azure Active Directory** 获取的标识符
- **Client Secret** （必需）：提供 **Microsoft Azure Active Directory** 获取的 **secret**
- 租户 **ID** （必需）：提供与 **Azure** 订阅中的 **Microsoft Azure Active Directory** 实例关联的唯一标识符
- 云环境 ：选择要应用的云环境

12.1.9. Thycotic DevOps Secrets Vault

当您为凭证类型选择 **Thycotic DevOps Secrets Vault** 时，请提供以下元数据来配置查找：

- **tenant** （必需）：提供用于与 **Thycotic** 的 **secret** 管理系统通信的 **URL**
- 顶级域**(TLD)** ：提供顶级域设计，如 **.com**、**.edu** 或 **.org**，与您要集成的 **secret** 库相关联
- 客户端 **ID** （必需）：提供由 **Thycotic secret** 管理系统获取的标识符
- **Client Secret** （必需）：提供由 **Thycotic secret** 管理系统获取的 **secret**

12.1.10. Thycotic Secret Server

当您为凭证类型选择 **Thycotic Secrets Server** 时，请提供以下元数据来配置查找：

- **Secret Server URL** （必需）：提供用于与 **Thycotic Secrets Server** 管理系统通信的 **URL**
- **username** （必需）：指定此服务的经过身份验证的用户
- **Password** （必需）：提供与用户关联的密码

第 13 章 应用程序

为外部应用（如 **ServiceNow** 和 **Jenkins**）创建和配置基于令牌的身份验证。通过基于令牌的身份验证，外部应用程序可以轻松地与自动化控制器集成。

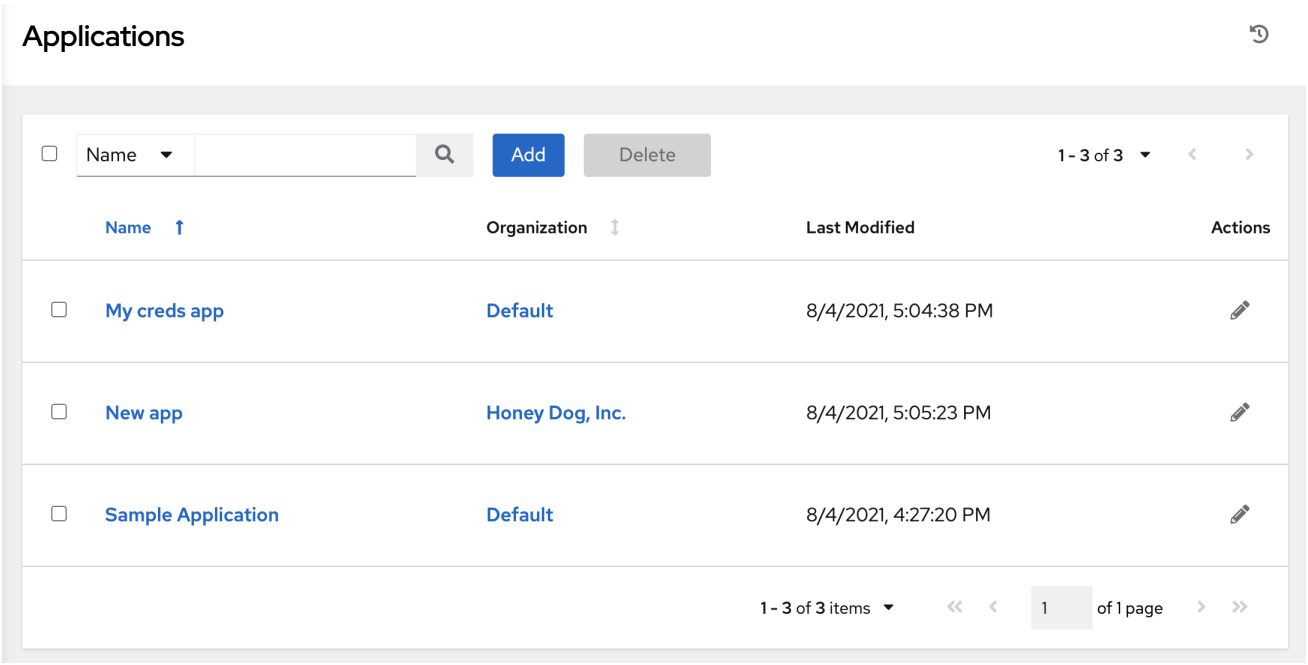
使用 **OAuth 2**，您可以使用令牌与应用程序共享数据，而无需公开登录信息。您可以将这些令牌配置为只读。

您可以创建一个代表您要与之集成的外部应用程序的应用程序，然后使用它为应用程序创建令牌，以代表其用户使用。

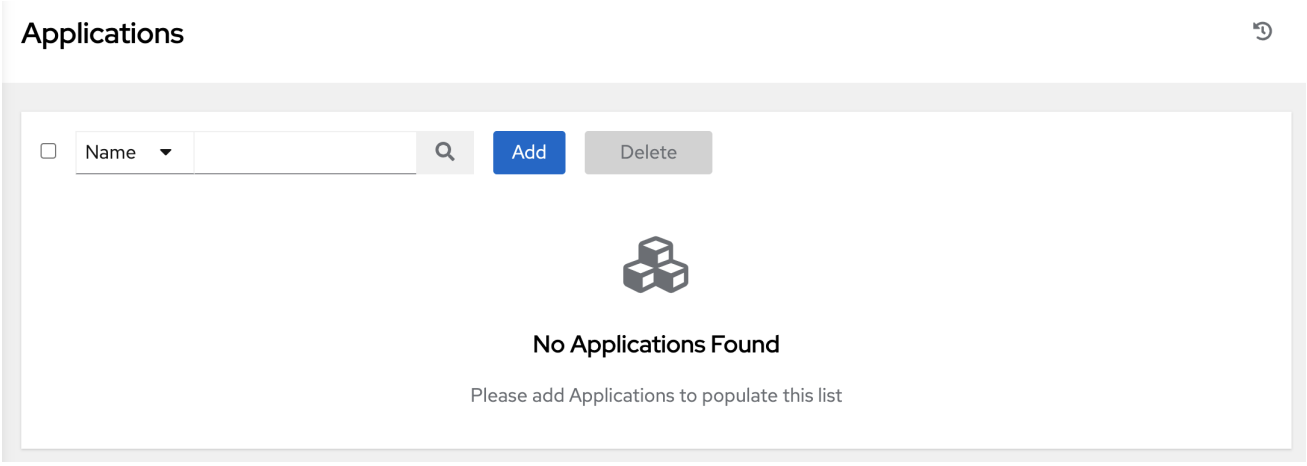
将这些令牌与应用程序资源关联，您可以管理为特定应用程序发布的所有令牌。通过在 **Applications** 下分离令牌问题，您可以根据应用程序撤销所有令牌，而无需撤销系统中的所有令牌。

13.1. 应用程序入门

在导航面板中，选择 **Administration** → **Applications**。**Applications** 页面显示目前由自动化控制器管理的所有可用应用程序的可搜索列表，并可以根据 **Name** 进行排序。



如果没有应用程序，则需要添加应用程序。



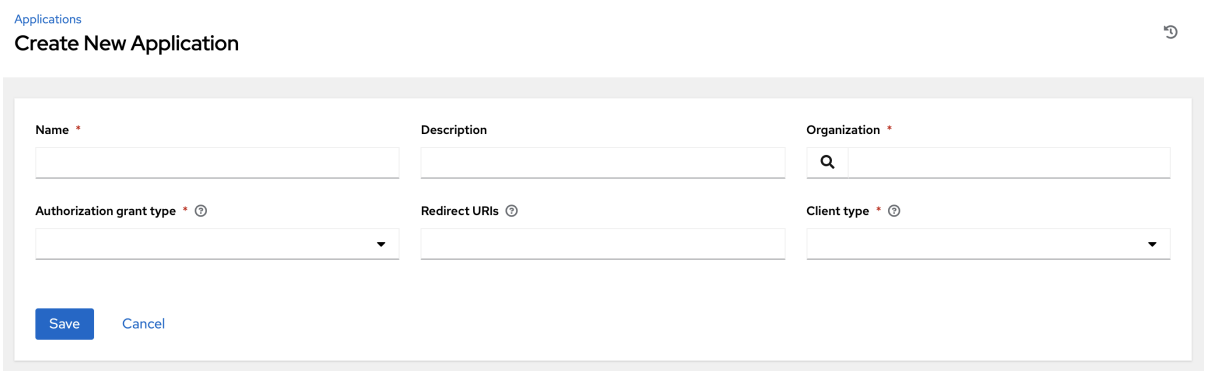
13.2. 创建新应用程序

当将外部 **Web** 应用程序与自动化控制器集成时，**Web** 应用可能需要代表 **web** 应用的用户创建 **OAuth2** 令牌。使用 **Authorization Code** 授权类型创建应用程序是这样做的首选方法：

- 外部应用程序可以使用用户的凭证获取令牌。
- 为特定应用程序发布的令牌部分化，可以轻松地管理这些令牌。例如，撤销与该应用程序关联的所有令牌。

流程

1. 在导航面板中，选择 **Administration** → **Applications**。
2. 点**Add**。**Create New Application** 页面将打开。



3. 输入以下详情：

- **Name** （必需）：为您要创建的应用程序提供名称
- 可选： 描述 ：为您的应用程序提供简短描述
- **Organization** （必需）：提供一个与这个应用程序关联的机构
- 授权授予类型 （必需）：选择一个授权类型，供用户用于获取此应用的令牌。如需更多信息，请参阅自动控制器 *管理指南中的应用程序功能部分*。
- 重定向 **URIS** ：提供允许的 **URI** 列表，用空格分开。如果您将授权类型指定为授权代码，则需要此项。
- 客户端类型 （必需）：选择客户端设备的安全性级别。

4. 点 **Save**，或者点 **Cancel** 来取消您的更改。

客户端 **ID** 显示在窗口中。

13.2.1. 添加令牌

您可以通过选择 **Tokens** 选项卡 **Application details** 页面来查看具有令牌访问应用程序的用户列表。

为您的用户配置身份验证令牌。您可以选择关联令牌的应用程序以及令牌具有的访问级别。

注意

您只能通过 **API** 或 **UI** 为用户创建 **OAuth 2** 令牌，这意味着您只能访问自己的用户配置集来配置或查看您的令牌。

流程

1. 在导航面板中，选择 **Access** → **Users**。

2. 选择您要为其配置 **OAuth 2** 令牌的用户。

3. 选择用户配置文件上的 **Tokens** 选项卡。

   当没有令牌时，**Tokens** 屏幕会提示您添加它们。

4. 点 **Add** 打开 **Create Token** 窗口。

5. 输入以下详情：

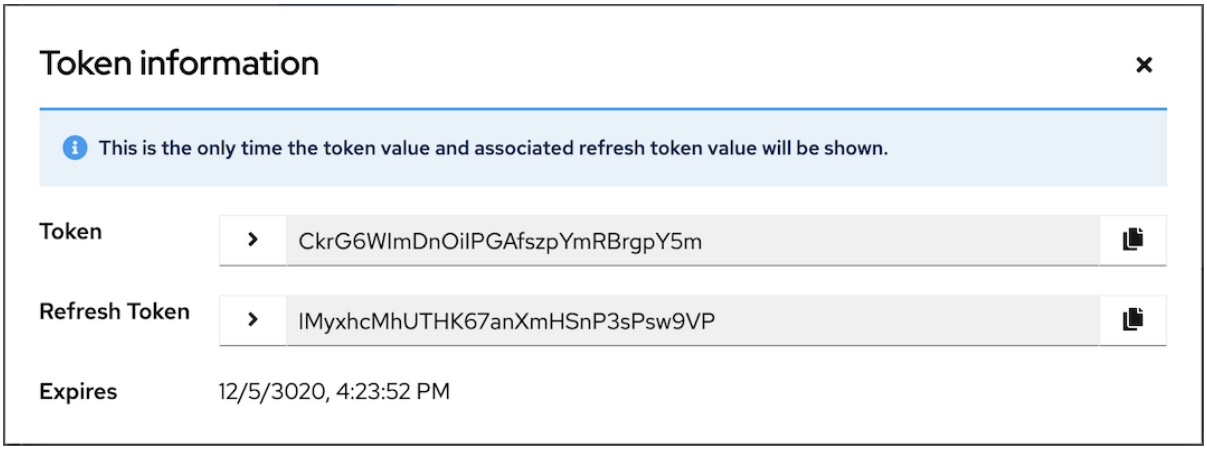
   - **Application:** 输入您要将令牌与之关联的应用程序的名称。另外，您可以通过点 图标进行搜索。这会打开一个单独的窗口，供您从可用选项中选择。如果列表太长，请使用搜索栏按名称过滤。如果要创建一个未链接到任何应用程序的个人访问令牌**(PAT)**，请将此字段留空。

   - 可选： 描述 ：为您的令牌提供简短描述。

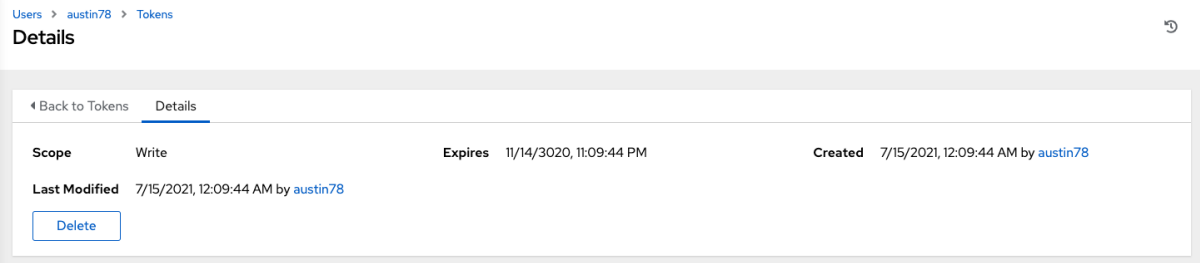
   - **Scope** （必需）：指定此令牌具有的访问级别。

6. 点 **Save**，或者点 **Cancel** 来取消您的更改。

   保存令牌后，用户新创建的令牌会显示令牌信息及其过期的时间。

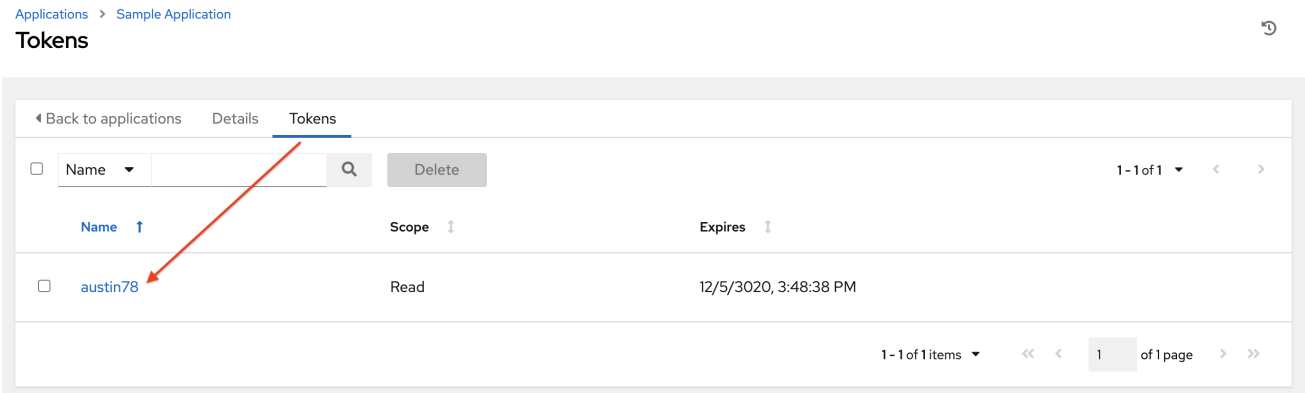


7. 要查看令牌关联的应用程序，以及令牌到期日期，请转至令牌列表视图。



验证

要验证应用程序现在显示具有适当令牌的用户，请打开 **Applications** 窗口的 **Tokens** 选项卡：



其他资源

如果您是系统管理员，并且必须为其他用户创建或删除令牌，请参阅 *自动化控制器管理指南中的* 令牌和 会话管理部分中的 **revoke** 和 **create** 命令。

第 14 章 执行环境

与旧的虚拟环境不同，执行环境是容器镜像，可以纳入系统级别的依赖项和基于集合的内容。每个执行环境都可让您有一个自定义镜像来运行作业，且仅在运行作业时需要什么。

14.1. 构建执行环境

如果您的 **Ansible** 内容依赖于自定义虚拟环境而不是默认环境，则必须执行额外的步骤。您必须在每个节点上安装软件包，并与主机系统上安装的其他软件进行正常交互，并使其保持同步。

要简化此过程，您可以构建作为 **Ansible** 控制节点 的容器镜像。**https://docs.ansible.com/ansible/latest/network/getting_started/basic_concepts.html#control-node**这些容器镜像称为自动化执行环境，您可以使用 **ansible-builder** 创建。然后，**ansible-runner** 可以使用这些镜像。

14.1.1. 安装 ansible-builder

要构建镜像，您必须安装 **Podman** 或 **Docker**，以及 **ansible-builder Python** 软件包。

--container-runtime 选项必须与您要使用的 **Podman** 或 **Docker** 可执行文件对应。

如需更多信息，请参阅 **Ansible Builder** 的 **Quickstart** 或 创建和使用执行环境。

14.1.2. 执行环境所需的内容

ansible-builder 用于创建执行环境。

执行环境必须包含：

- **Ansible**
- **Ansible Runner**

- **Ansible Collections**
- **Python** 和系统依赖项：
  - 集合中的模块或插件
  - **ansible-base** 中的内容
  - 自定义用户需要

构建新的执行环境涉及定义，该定义指定要包含在执行环境中的内容，如集合、**Python** 要求和系统级软件包。定义必须是 **.yml** 文件

从迁移到执行环境生成的输出中的内容有一些需要的数据，这些数据可以传送到文件或粘贴到此定义文件中。

其他资源

如需更多信息，请参阅 迁移旧的 **venvs** 到执行环境。如果您没有从虚拟环境迁移，您可以使用 执行环境设置参考 中描述的所需数据创建一个定义文件。

集合开发人员可以通过提供适当的元数据来声明其内容要求。

如需更多信息，请参阅 依赖项。

14.1.3. 构建镜像的 YAML 文件示例

ansible-builder build 命令将执行环境定义用作输入。它输出构建执行环境镜像所需的构建上下文，然后构建该镜像。镜像可以使用其他构建上下文重新构建，并生成相同的结果。默认情况下，构建器在当前目录中搜索名为 **execution-environment.yml** 的文件。

以下示例 **execution-environment.yml** 文件可用作起点：

```
-
```

```
---
version: 3
dependencies:
  galaxy: requirements.yml
```

requirements.yml 的内容：

```
---
collections:
  - name: awx.awx
```

要使用前面的文件构建执行环境，并运行以下命令：

```
ansible-builder build
...
STEP 7: COMMIT my-awx-ee
--> 09c930f5f6a
09c930f5f6ac329b7ddb321b144a029dbbfcc83bdfc77103968b7f6cdfc7bea2
Complete! The build context can be found at: context
```

除了生成随时可用的容器镜像外，构建上下文也会保留。这可以通过您选择的工具（如 **docker build** 或 **podman build** ）在不同时间或位置进行重建。

其他资源

如需有关 **ansible-builder build** 命令的更多信息，请参阅 **Ansible** 的 **CLI** 用法 文档。

14.1.4. 执行环境挂载选项

重建执行环境是添加证书的一种方式，但从主机继承证书提供了一种更方便的解决方案。对于基于虚拟机的安装，自动化控制器会在作业运行时会在执行环境中自动挂载系统信任存储。

您可以在 **Paths** 中自定义执行环境挂载选项和挂载路径，以公开给 **Job Settings** 页面的 **isolated jobs** 字段，其中支持 **Podman** 风格的卷挂载语法。

其他资源

如需更多信息，请参阅 **Podman** 文档。

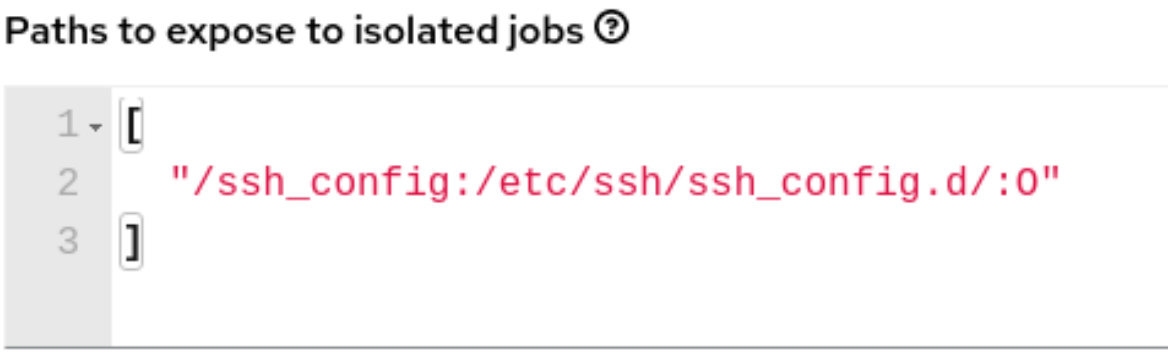
14.1.4.1. 执行环境挂载选项故障排除

在某些情况下，由于自定义执行环境，**/etc/ssh decisions** 文件被添加到执行环境镜像中，可能会出现 **SSH** 错误。例如，公开 **/etc/ssh/ssh_config.d:/etc/ssh/ssh_config.d:O** 路径可让容器被挂载，但所有权权限无法正确映射。

如果您满足这个错误，或者已从旧版本的自动化控制器升级，请使用以下步骤：

流程

1. 将挂载卷上的容器所有权更改为 **root**。
2. 在导航面板中，选择 **Settings**。
3. 从 **Jobs** 选项中选择 **Jobs** 设置。
4. 使用当前示例，在 **Paths** 中公开要公开给隔离的 **jobs** 字段的路径：



注意

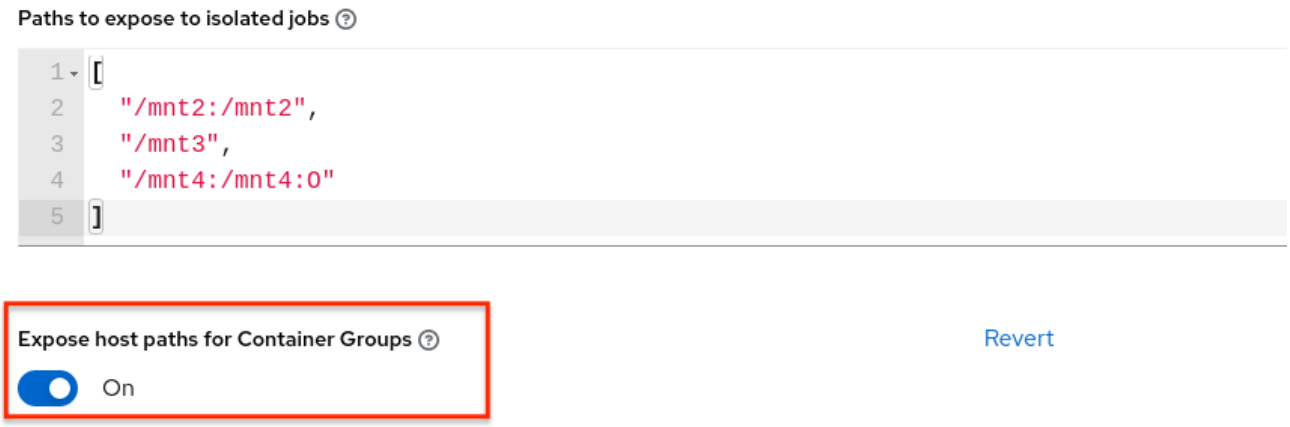
:O 选项只支持目录。尽可能详细，特别是在指定系统路径时。直接挂载 **/etc** 或 **/usr** 会影响到故障排除的难度。

这会通知 **Podman** 运行类似以下示例的命令，其中挂载了配置，**ssh** 命令可以正常工作：

```
podman run -v /ssh_config:/etc/ssh/ssh_config.d/:O ...
```

-

要在 **OpenShift** 或 **Kubernetes** 容器中公开隔离路径为 **HostPath**，请使用以下配置：



将容器组的主机路径设置为 **On** 以启用它。

当 **playbook** 运行时，生成的 **Pod** 规格类似以下示例。请注意 **volumeMounts** 和 **volumes** 部分的详情。

```
apiVersion: v1
kind: Pod
spec:
  containers:
  - image: registry.redhat.io/ansible-automation-platform-22/ee-minimal-rhel8
    args:
      - ansible-runner
      - worker
      - --private-data-dir=/runner
    volumeMounts:
      - mountPath: /mnt2
        name: volume-0
        readOnly: true
      - mountPath: /mnt3
        name: volume-1
        readOnly: true
      - mountPath: /mnt4
        name: volume-2
        readOnly: true
  volumes:
    - hostPath:
        path: /mnt2
        type: ""
      name: volume-0
    - hostPath:
        path: /mnt3
        type: ""
      name: volume-1
    - hostPath:
        path: /mnt4
        type: ""
      name: volume-2
```

14.1.4.2. 将执行节点中的目录挂载到执行环境容器中

在 **Ansible Automation Platform 2.1.2** 中，只有 **O** 和 **z** 选项可用。从 **Ansible Automation Platform 2.2** 开始，提供了其他选项，如 **rw**。这在使用 **NFS** 存储时很有用。

流程

1. 在导航面板中，选择 **Settings**。
2. 从 **Jobs** 选项中选择 **Jobs** 设置。
3. 编辑 要向隔离的 **jobs** 字段公开 的路径：

- 输入卷从执行节点或混合节点挂载到容器的路径列表。
- 每行输入一个路径。
- 支持的格式为 **HOST-DIR[:CONTAINER-DIR[:OPTIONS]**。允许的路径有 **z**、**O**、**ro** 和 **rw**。

  Example

```
[
  "/var/lib/awx/.ssh:/root/.ssh:O"
]
```

- 对于 **rw** 选项，请正确配置 **SELinux** 标签。例如，要挂载 **/foo** 目录，请完成以下命令：

```
sudo su
mkdir /foo
chmod 777 /foo
semanage fcontext -a -t container_file_t "/foo(/.*)?"
restorecon -vvFR /foo
```

awx 用户必须至少允许在这个目录中读取或写入。将权限设置为 **777**。

其他资源

有关挂载卷的更多信息，请参阅 **Podman** 文档中的 **podman-run (1)**部分中的**--volume** 选项。

14.2. 将执行环境添加到作业模板

先决条件

- 必须使用 **ansible-builder** 创建执行环境，如 构建执行环境 中所述。创建执行环境后，您可以使用它来运行作业。使用自动化控制器 **UI** 指定作业模板中使用的执行环境。

- 根据执行环境是否为全局用途或者与某个机构关联，您必须具有适当的管理员特权级别才能在作业中使用执行环境。与组织关联的执行环境需要机构管理员能够使用这些执行环境运行作业。

- 在运行使用为其分配了凭证的执行环境的作业或作业模板前，请确保凭据包含用户名、主机和密码。

流程

1. 在导航面板中，选择 **Administration → Execution Environments**。

2. 单击 **Add** 以添加执行环境。

3. 在以下字段中输入相关信息：

   - **Name** （必需）：输入执行环境的名称。

   - **Image** （必需）：输入镜像名称。镜像名称需要其完整位置（存储库）、**registry**、镜像名称和版本标签，格式为 **quay.io/ansible/awx-ee:latestrepo/project/image-name:tag**。

   - 可选： **Pull:** 在运行作业时选择拉取类型：

     - 在运行前始终拉取容器 ：为容器拉取最新的镜像文件。

     - 只有在运行前不存在时才拉取镜像 ：如果没有指定，则仅拉取最新的镜像。

     - 永不会拉取容器，然后再运行**:** 永不拉取容器镜像的最新版本。

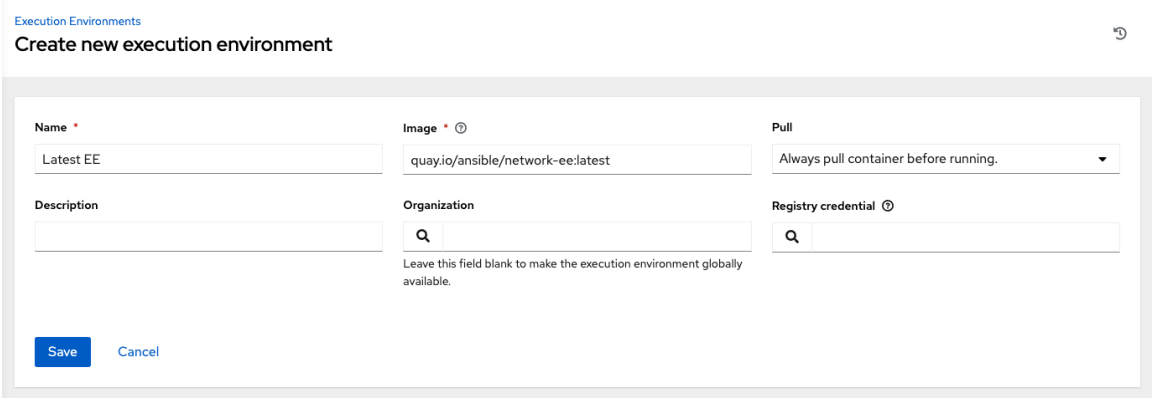
注意

如果您没有为 **pull** 设置拼写错误，则默认值为 仅在运行 之前不存在时才拉取镜像。

- 可选： 描述 ：

- 可选： 机构 分配机构以专门使用此执行环境。要使执行环境可用于多个机构，请将此字段留空。

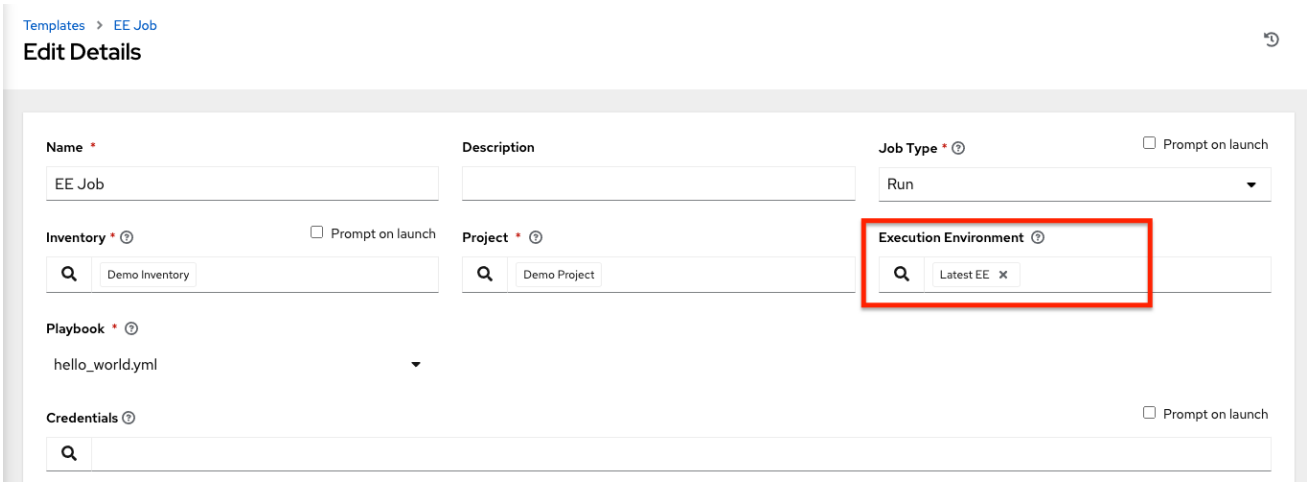
- **Registry** 凭证 ：如果镜像有一个受保护的容器 **registry**，提供访问它的凭证。



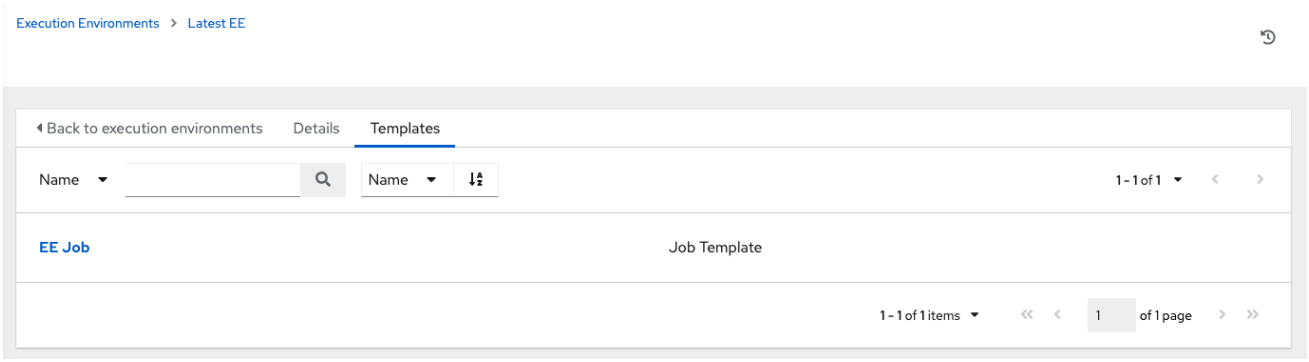
4. 点击 **Save**。

您新添加的执行环境已准备好在作业模板中使用。

5. 要将执行环境添加到作业模板中，在作业模板的 **Execution Environment** 字段中指定它，如下例所示：



当您将执行环境添加到作业模板时，这些模板会在执行环境的 **Templates** 选项卡中列出：



第 15 章 执行环境设置参考

本节包含与执行环境定义关联的参考信息。您可以在 **YAML** 文件中定义执行环境的内容。默认情况下，该文件名为 **execution_environment.yml**。此文件告知 **Ansible Builder** 如何创建构建指令文件（适用于 **Podman** 的**Containerfile**，**Docker** 的 **Dockerfile**）并为容器镜像构建上下文。

注意

Ansible Builder 3.x 的定义模式记录在此处。如果您正在运行旧版本的 **Ansible Builder**，则需要一个旧的 **schema** 版本。如需更多信息，请参阅本文档 的旧版本。我们建议使用版本 **3**，它提供了比之前的版本更可配置的选项和功能。

15.1. 执行环境定义示例

您必须创建一个定义文件来为执行环境构建镜像。该文件采用 **YAML** 格式。

您必须在定义文件中指定 **Ansible Builder** 的版本。默认版本为 **1**。

以下定义文件使用 **Ansible Builder** 版本 **3**：

```
version: 3
build_arg_defaults:
  ANSIBLE_GALAXY_CLI_COLLECTION_OPTS: '--pre'
dependencies:
  galaxy: requirements.yml
  python:
    - six
    - psutil
  system: bindep.txt
images:
  base_image:
    name: registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest
additional_build_files:
    - src: files/ansible.cfg
      dest: configs
additional_build_steps:
  prepend_galaxy:
    - ADD _build/configs/ansible.cfg /home/runner/.ansible.cfg
  prepend_final: |
    RUN whoami
    RUN cat /etc/os-release
  append_final:
    - RUN echo This is a post-install command!
    - RUN ls -la /etc
```

15.2. 配置选项

在定义文件中使用以下配置 **YAML** 键。

Ansible Builder 3.x 执行环境定义文件接受七个顶级部分：

- additional_build_files
- additional_build_steps
- build_arg_defaults
- dependencies
- images
  - 镜像验证
- options
- version

15.2.1. additional_build_files

构建文件指定要添加到构建上下文目录中的内容。然后，可以在任何构建阶段通过 **additional_build_steps** 引用或复制它们。

格式是字典值的列表，每个值都有一个 **src** 和 **dest** 键和值。

每个列表项都必须是包含以下所需键的字典：

src	指定要复制到构建上下文目录中的源文件。 这可以是绝对路径，例如 **/home/user/.ansible.cfg**，也可以是相对于该文件的路径。相对路径可以是与一个或多个文件匹配的 glob 表达式，例如 **files/*.cfg**。请注意，绝对路径不得包含正则表达式。如果 **src** 是目录，则该目录的整个内容将复制到 **dest**。
dest	指定构建上下文目录的 **_build** 子目录下的子目录路径，该路径包含源文件，如 **files/configs**。 这不能是绝对路径，也不能包含 **..**。如果该目录不存在，则会为您创建该目录。 **注意** 当使用 **ansible.cfg** 文件将私有帐户的令牌和其他设置传递给自动化中心服务器时，在此处列出配置文件路径作为字符串，使其作为构建初始阶段的构建参数包含。

15.2.2. additional_build_steps

构建步骤为任何构建阶段指定自定义构建命令。这些命令直接插入到容器运行时的 **build** 指令文件中，如 **Containerfile** 或 **Dockerfile**。命令必须符合容器化工具所需的任何规则。

您可以在镜像创建过程的任何阶段之前或之后添加构建步骤。例如，如果在安装依赖项前需要安装 **git**，您可以在基本构建阶段添加构建步骤。

以下是有效的密钥。分别支持多行字符串或字符串列表。

append_base	构建基础镜像后插入的命令。
append_builder	构建构建器镜像后要插入的命令。
append_final	构建最终镜像后要插入的命令。
append_galaxy	构建 galaxy 镜像后插入的命令。
prepend_base	在构建基础镜像前插入的命令。
prepend_builder	在构建构建器镜像前插入的命令。
prepend_final	在构建最终镜像前插入的命令。
prepend_galaxy	在构建 galaxy 镜像前插入的命令。

15.2.3. build_arg_defaults

这将构建参数的默认值指定为字典。

这是使用**- build-arg CLI** 标志的替代选择。

Ansible Builder 使用以下构建参数：

ANSIBLE_GALAXY_CLI_COLLECTION_OPTS	允许用户传递 **-pre** 标志和其他标志来启用预发布集合的安装。
ANSIBLE_GALAXY_CLI_ROLE_OPTS	这可让用户将任何标记（如 **--no-deps** ）传递给角色安装。
PKGMGR_PRESERVE_CACHE	这控制在镜像构建过程中清除软件包管理器缓存的频率。 如果没有设置这个值（这是默认值），则缓存会被频繁清除。如果值始终为，则缓存永远不会清除。在最终构建阶段安装系统依赖项后，任何其他值都会强制缓存被清除。

在 **build_arg_defaults** 中给出的 **Ansible Builder** 硬编码值，以便在手动运行容器构建时保留它们。

如果您在定义中指定相同的变量，且在命令行中使用 **CLI build-arg** 标志，**CLI** 值会覆盖定义中的值。

15.2.4. 依赖项

指定要安装到最终镜像的依赖项，包括 **ansible-core**、**ansible-runner**、**Python** 软件包、系统软件包和集合。**Ansible Builder** 会自动安装您安装的任何 **Ansible** 集合的依赖项。

通常，您可以使用标准语法来约束软件包版本。使用您传递给 **dnf**、**pip**、**ansible-galaxy** 或任何其他软件包管理工具的语法。您还可以在单独的文件中定义软件包或集合，并在定义文件的 **dependencies** 部分中引用这些文件。

以下键有效：

ansible_core	要安装 **ansible-core** Python 软件包的版本。 这个值是一个字典，其中包含一个键 **package_pip**。**package_pip** 值直接传递给 pip 进行安装，并可采用 pip 支持的任何格式。以下是一些示例值： ansible_core: package_pip: ansible-core ansible_core: package_pip: ansible-core==2.14.3 ansible_core: package_pip: https://github.com/example_user/ansible/archive/refs/heads/ansible.tar.gz
ansible_runner	要安装的 Ansible Runner Python 软件包的版本。 这个值是一个字典，其中包含一个键 **package_pip**。**package_pip** 值直接传递给 pip 进行安装，并可采用 pip 支持的任何格式。以下是一些示例值： ansible_runner: package_pip: ansible-runner ansible_runner: package_pip: ansible-runner==2.3.2 ansible_runner: package_pip: https://github.com/example_user/ansible-runner/archive/refs/heads/ansible-runner.tar.gz
galaxy	要从 Ansible Galaxy 安装的集合。 这可以是 Ansible Galaxy **requirements.yml** 文件的文件名、字典或多行字符串。有关要求文件格式的更多信息，请参阅 Galaxy 用户指南。

python	Python 安装要求。 这可以是文件名，也可以是要求列表。Ansible Builder 使用 **requirements-parser** 库将所有集合中的所有 Python 要求文件合并到一个文件中。 此库支持复杂的语法，包括对其他文件的引用。如果很多集合需要相同的*软件包名称*，Ansible Builder 会将它们合并到一个条目中，并组合了约束。 Ansible Builder 排除 Python 依赖项组合文件中的一些软件包，即使集合将它们列为依赖项。其中包括提供 Ansible 本身的测试软件包和软件包。完整列表可以在 **src/ansible_builder/_target_scripts/introspect.py** 中的 **EXCLUDE_REQUIREMENTS** 下提供。 如果您需要包含其中一个排除的软件包名称，请使用 introspection 命令的 **--user-pip** 选项，在用户要求文件中列出它。 以这种方式提供的软件包不会针对排除的 **Python** 软件包列表进行处理。
python_interpreter	定义 dnf 要安装的 Python 系统软件包名称的字典(**package_system**)或要使用的 Python 解释器的路径(**python_path**)。
system	要安装的系统软件包，采用 bindep 格式。这可以是文件名或要求列表。 有关 bindep 的更多信息，请参阅 OpenDev 文档。 对于系统软件包，使用 **bindep** 格式来指定跨平台要求，因此可以通过执行环境使用的软件包管理系统进行安装。集合必须为 **[platform:rpm]** 指定必要的要求。Ansible Builder 将来自多个集合的系统软件包条目合并到一个文件中。只有没有配置集（运行时要求）的要求才会安装到镜像中。许多集合中可以合并在组合文件中重复的条目。

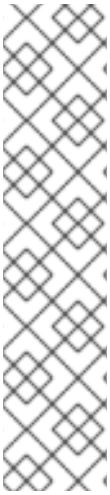
以下示例使用包含各种依赖项的文件名：

```
dependencies:
  python: requirements.txt
  system: bindep.txt
  galaxy: requirements.yml
  ansible_core:
     package_pip: ansible-core==2.14.2
  ansible_runner:
```

```
    package_pip: ansible-runner==2.3.1
  python_interpreter:
    package_system: "python310"
    python_path: "/usr/bin/python3.10"
```

这个示例使用内联值：

```
dependencies:
  python:
    - pywinrm
  system:
    - iputils [platform:rpm]
  galaxy:
    collections:
      - name: community.windows
      - name: ansible.utils
        version: 2.10.1
  ansible_core:
    package_pip: ansible-core==2.14.2
  ansible_runner:
    package_pip: ansible-runner==2.3.1
  python_interpreter:
    package_system: "python310"
    python_path: "/usr/bin/python3.10"
```



注意

如果这些依赖项文件**(requirements.txt**、**bindep.txt** 和 **requirements.yml)**位于集合的 **build_ignore** 中，则构建会失败。

集合维护者可以使用 **introspection** 命令验证 **ansible-builder** 是否识别他们期望的要求：

```
ansible-builder introspect --sanitize ~/.ansible/collections/
```

--sanitize 选项查看所有集合要求并**删**除重复项。它还会**删**除通常排除的任何 **Python** 要求（请参阅 **python** 依赖项）。

使用 **-v3** 选项 内省，以查看关于被排除的要求的消息。

15.2.5. images

指定要使用的基础镜像。至少，您必须为基础镜像指定源、镜像和标签。基础镜像提供操作系统，也

可以提供一些软件包。使用标准 **host/namespace/container:tag** 语法来指定镜像。您可以使用 **Podman** 或 **Docker** 快捷方式语法，但完整定义更为可靠和可移植。

本节的有效键包括：

base_image	定义执行环境的父镜像的字典。 必须为要使用的容器镜像提供 **name** 键。如果镜像在存储库中镜像，但使用原始镜像的签名密钥签名，请使用 **signature_original_name** 密钥。

15.2.6. 镜像验证

如果使用 **podman** 容器运行时，您可以验证签名的容器镜像。

设置 **container-policy CLI** 选项，以控制与容器镜像签名验证相关的 **Podman policy.json** 文件。

- **ignore_all** 策略：在构建上下文目录 **< context>** 中生成一个 **policy.json** 文件，但没有执行签名验证。
- 系统 策略：使用标准系统位置中的预先存在的 **policy.json** 文件来执行签名验证。**Ansible-builder** 不负责这些文件中的内容，并且用户对内容有完全的控制权。
- **signature_required** 策略： **ansible-builder** 使用容器镜像定义在构建期间使用的 构建上下文目录 **<context>** 中生成 **policy.json** 文件来验证镜像。

15.2.7. options

影响运行时功能的关键字或选项的字典。

本节的有效键包括：

- **container_init:** 带有键的字典，用于自定义容器 **ENTRYPOINT** 和 **CMD** 指令（及相关行为）。自定义这些行为是一个高级任务，可能会导致无法调试失败。由于提供的默认值控制多个中间的行为，覆盖任何值都跳过此字典中的所有剩余默认值。

有效键是：

- **cmd: CMD Containerfile** 指令的 **Literal** 值。默认值为 **["bash "]**。

- **ENTRYPOINT: ENTRYPOINT Containerfile** 指令的 **Literal** 值。默认入口点行为处理到子进程的信号传播，并在运行时确保容器用户具有具有有效写入主目录（在 **/etc/passwd** 中代表）的正确环境（在 **/etc/passwd** 中表示），并将 **HOME** 环境变量设置为匹配。当无法适当调整用户运行时环境时，默认入口点脚本会向 **stderr** 发送警告。此行为可以忽略或提升为致命错误；请查阅 **entrypoint** 目标脚本的来源以了解更多详细信息。

  默认值为 **["/opt/builder/bin/entrypoint", "dumb-init "]**。

- **package_pip** ：使用 **pip** 进行入口点支持安装的软件包。此软件包安装在最终构建镜像中。

  默认值为 **dumb-init==1.2.5**。

- **package_manager_path:** 字符串，带有要使用的软件包管理器**(dnf** 或 **microdnf)**的路径。默认为 **/usr/bin/dnf**。这个值用于安装 **Python** 解释器，如果在 依赖项 中指定，并在 **assemble** 脚本的构建阶段指定。

- **skip_ansible_check** ：此布尔值控制检查在最终镜像上执行 **Ansible** 和 **Ansible Runner** 的安装。

  将此值设置为 **True** 以不执行此检查。

  默认值为 **False**。

- **relax_passwd_permissions** ：此布尔值控制 **root** 组**(GID 0)**是否明确授予最终容器镜像中的 **/etc/passwd** 的写入权限。默认入口点脚本可以在某些容器运行时下尝试通过动态创建用户来更新 **/etc/passwd**，以确保全功能 **POSIX** 用户环境和主目录。禁用此功能可能会导致软件功能失败，要求用户在 **/etc/passwd** 中列出具有有效且可写入的主目录，例如 **ansible-core** 中的 **async** 和 **~username shell** 扩展。

  默认值为 **True**。

-

WORKDIR ：在最终容器镜像下启动新进程的默认当前工作目录。有些容器运行时也将此值用作 **root (GID 0)**组中动态创建用户的 **HOME**。当指定这个值时，如果目录不存在，则创建该目录，设置为 **root** 组所有权，并且 **rwx** 组权限会递归应用到其中。

默认值为 **/runner**。

- 用户 ：这会设置用户名或 **UID**，以用作最终容器镜像的默认用户。

  默认值为 **1000**。

示例选项：

```
options:
  container_init:
    package_pip: dumb-init>=1.2.5
    entrypoint: '["dumb-init"]'
    cmd: '["csh"]'
  package_manager_path: /usr/bin/microdnf
  relax_password_permissions: false
  skip_ansible_check: true
  workdir: /myworkdir
  user: bob
```

15.2.8. version

一个整数值，用于设置执行环境定义文件的 **schema** 版本。

默认为 **1**。

如果使用 **Ansible Builder 3.x**，则该值必须是 **3**。

15.3. AWX 的默认执行环境

test/data/pytz 中的示例需要定义中的 **awx.awx** 集合。**lookup** 插件 **awx.awx.tower_schedule_rrule** 需要 **PyPI pytz** 和另一个库正常工作。如果提供了 **test/data/pytz/execution - environment.yml** 文件，

它会在镜像中安装集合，读取集合中的 **requirements.txt** 文件，然后将 **pytz** 安装到镜像中。

生成的镜像可以通过将这些变量放在私有数据目录中来在 **ansible-runner** 项目内使用。

```
---
container_image: image-name
process_isolation_executable: podman # or docker
process_isolation: true
```

awx.awx 集合是默认 **AWX** 中包含的内容子集。

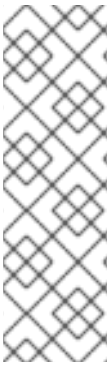
如需更多信息，请参阅 **awx-ee** 存储库。

第 16 章 项目

项目是在自动化控制器中表示的 **Ansible playbook** 的逻辑集合。您可以以不同的方式管理 **playbook** 和 **playbook** 目录：

- 将它们手动放置到自动化控制器服务器的 **Project Base** 路径下。
- 通过将 **playbook** 放置到自动化控制器支持的源代码管理(SCM)系统中。这包括 **Git**、**Subversion**、**Mercurial** 和 **Red Hat Insights**。

有关创建 **Red Hat Insights** 项目的更多信息，请参阅设置 **insights** 修复。



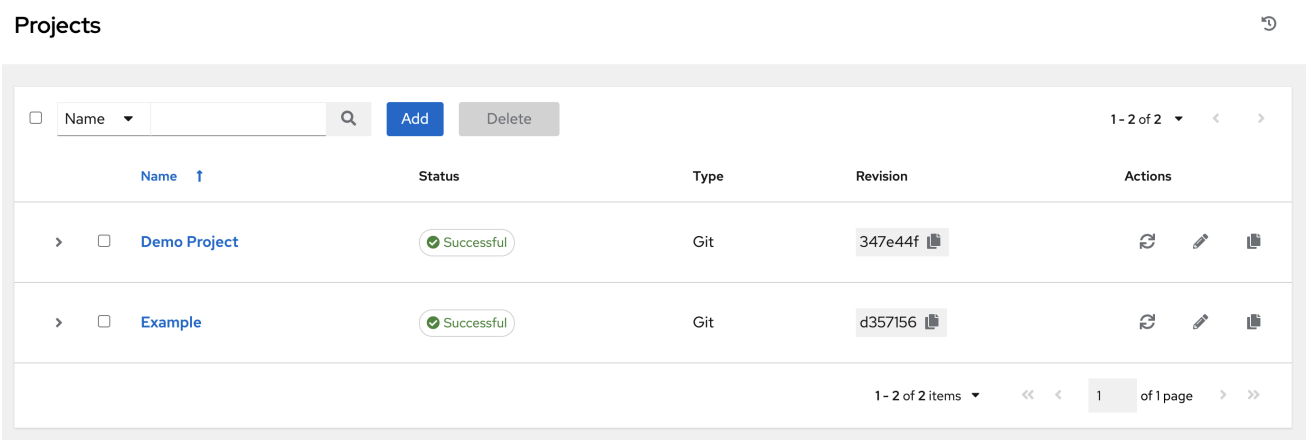
注意

项目基本路径为 **/var/lib/awx/projects**。但是，这可以被系统管理员进行修改。它在 **/etc/tower/conf.d/custom.py** 中配置。

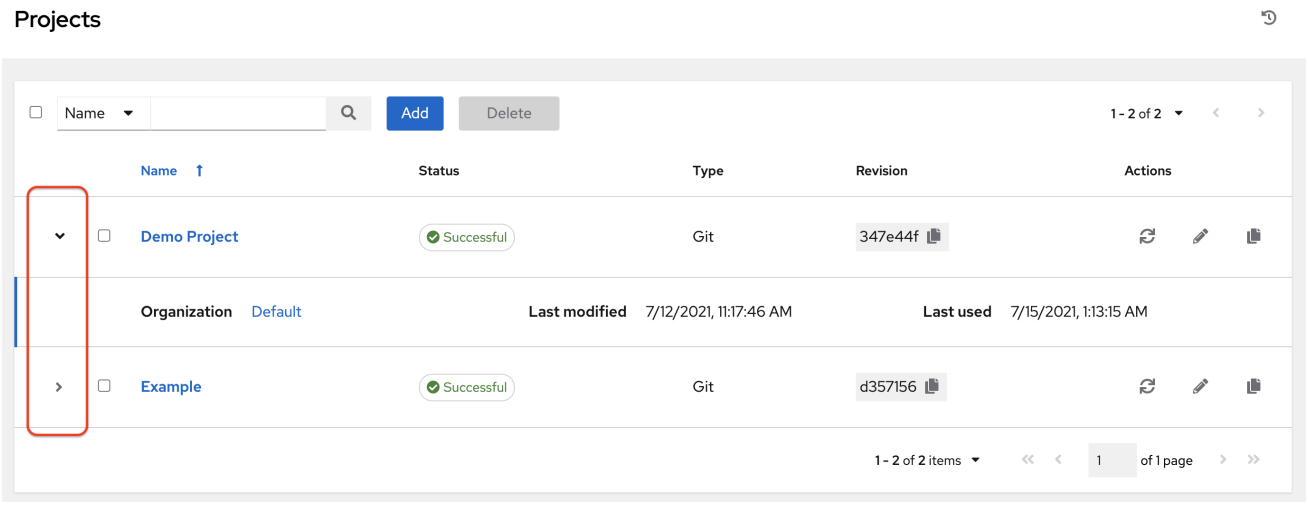
编辑此文件时要小心，因为不正确的设置可以禁用您的安装。

Projects 页面中显示当前可用的项目列表。

自动化控制器为您提供了可初始使用的 **Demo Project**。



默认视图为折叠状态(**Compact**)，带有项目名称及其状态，但您可以使用每个条目旁边的
>
来展开以了解更多信息。



对于列出的每个项目，您可以使用每个项目旁边的图标获取最新的 **SCM** 修订
，编辑
项目，或者复制
项目属性。

在相关作业运行时，可以更新项目。

如果您有大型项目（大约 **10 GB**），**/tmp** 上的磁盘空间可能会出现问题。

status 表示项目的状态，可以是以下之一（注意您也可以根据特定状态类型过滤您的视图）：

- **Pending -** 已创建源控制更新，但尚未排队或启动。任何作业（不仅仅是源控制更新）会一直处于待处理状态，直到系统准备好运行为止。可能的原因包括：
  - 它有依赖项当前正在运行，因此必须等到它们完成为止。

- 在其配置的位置没有足够的运行容量。

- **waiting -** 源控制更新处于等待执行的队列中。
- **Running -** 源控制更新当前正在进行中。
- 成功 **-** 此项目的最后源控制更新成功。
- **Failed -** 此项目的最后源控制更新失败。
- **Error -** 最后的源控制更新作业根本无法运行。
- 已取消 **-** 项目的最新源控制更新已被取消。
- 永不更新 **-** 项目是为源控制配置的，但从未更新。
- **OK -** 项目没有配置为源控制，且正确就位。
- 缺少 项目基本路径的 **/var/lib/awx/projects** 中没有。这适用于手动或源控制受管项目。

注意

凭证类型 **Manual** 的项目无法更新或调度基于源控制的操作，而无需重新配置为 **SCM** 类型凭证。

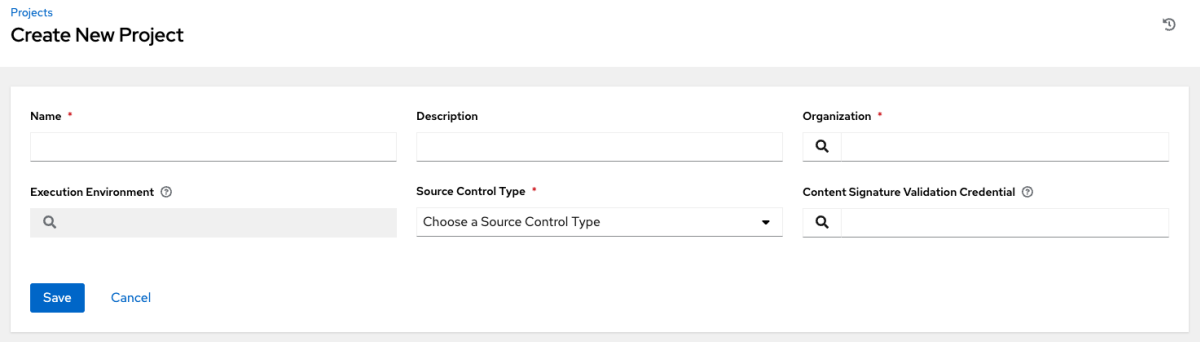
16.1. 添加新项目

您可以在自动化控制器中创建一个名为 **projects** 的 **playbook** 的逻辑集合。

流程

1. 在导航面板中，选择 **Resources** → **Projects**。

2. 在 **Projects** 页面上，单击 **Add** 以启动 **Create Project** 窗口。



3. 在以下必填字段中输入相关信息：

- **Name** （必需）
- 可选： 描述
- **Organization** （必需）：项目必须至少有一个机构。现在选择一个机构来创建项目。创建项目时，您可以添加额外的机构。
- 可选： 执行环境 ：输入执行环境的名称，或从现有环境列表中搜索以运行此项目。如需更多信息，请参阅 ***Red Hat Ansible Automation Platform*** *升级和迁移指南中的迁移到执行环境*。
- **Source Control Type** （必需）：从菜单中选择与此项目关联的 **SCM** 类型。以下部分中的选项会根据所选的类型提供。如需更多信息，请参阅 手动管理 **playbook** 或使用 源控制管理 **playbook**。
- 可选： **Content Signature Validation Credential:** 使用此字段启用内容验证。指定用于在项目同步期间验证内容签名的 **GPG** 密钥。如果内容已被篡改，则该作业将不会运行。如需更多信息，请参阅 项目签名和验证。

4. 点击 **Save**。

其他资源

以下描述了项目源的方式：

- 手动管理 **playbook**
- 使用源控制管理 **playbook**
  - **SCM 类型 - Git 和 Subversion**
  - **SCM 类型 - Red Hat Insights**
  - **SCM 类型 - 远程归档**

16.1.1. 手动管理 playbook

流程

- 在 **Project Base Path** 下创建一个或多个目录来存储 **playbook**，例如 **/var/lib/awx/projects/**。
- 创建 **playbook** 文件或将其复制到 **playbook** 目录中。
- 确保 **playbook** 目录和文件属于运行该服务的同一 **UNIX** 用户和组。
- 确保权限适合 **playbook** 目录和文件。

故障排除

- 如果您还没有将任何 **Ansible Playbook** 目录添加到基础项目路径中，则会显示错误消息。选择以下选项之一排除这个错误：

- 创建适当的 **playbook** 目录并从您的 **SCM** 中签出 **playbook** （拼写此*）。
- 将 **playbook** 复制到适当的 **playbook** 目录中。

16.1.2. 使用源控制管理 playbook

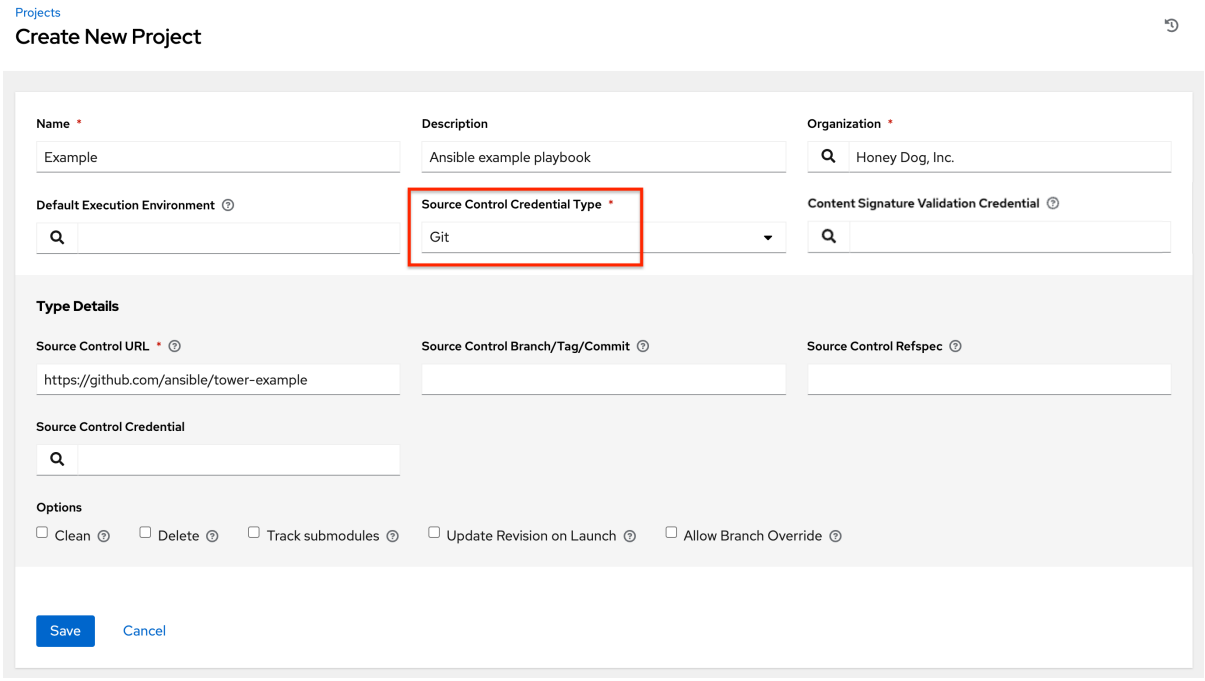
在使用源控制管理 **playbook** 时选择以下选项之一：

- **SCM 类型 - 配置 playbook 以使用 Git 和 Subversion**
- **SCM 类型 - 配置 playbook 以使用 Red Hat Insights**
- **SCM 类型 - 配置 playbook 以使用远程归档**

16.1.2.1. SCM 类型 - 配置 playbook 以使用 Git 和 Subversion

流程

1. 在 **Project Details** 选项卡中，从 **SCM Type** 菜单中选择适当的选项(**Git** 或 **Subversion**)。



2. 在以下字段中输入相关信息：

- **SCM URL -** 请参阅工具提示 中的示例。
- 可选： **SCM Branch/Tag/Commit** ：从源控制**(Git** 或 **Subversion)**输入 **SCM** 分支、标签、提交散列、任意 **refs** 或修订号（如果适用）。除非在下一字段中还提供了自定义 **refspec**，否则某些提交哈希和引用可能不可用。如果留空，则默认为 **HEAD**，这是此项目最后一次签出的 **Branch**、**Tag** 或 **Commit**。
- **SCM Refspec -**此字段特定于 **git** 源控制的选项，只有熟悉和熟悉的高级用户才应指定要从远程存储库下载哪些引用。如需更多信息，请参阅 作业分支覆盖。
- 源控制凭证 **-**如果需要身份验证，请选择适当的源控制凭证

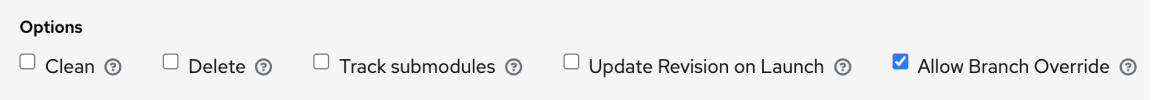
3. 可选： **SCM Update Options -** 选择启动行为（如果适用）：

- 在 进行更新前清除任何本地修改。
- 删除 **-** 在进行更新前删除整个本地存储库。根据存储库的大小，这可能会显著增加完成更新所需的时间。

- 跟踪子模块 - 跟踪最新的提交。工具提示 中提供了更多信息。

- 更新启动时的 **Revision**，将项目的修订更新至远程源控制中的当前修订版本，并缓存来自 **Galaxy** 或 **Collections** 的角色目录支持。自动化控制器可确保本地修订版本匹配，并且角色和集合与最近更新保持同步。另外，为了避免在生成作业的速度超过项目可以同步的速度，为了避免作业溢出，请选择此选项可让您将缓存超时配置为缓存之前的项目同步。

- **Allow Branch Override** - 启用作业模板或使用此项目的清单源，从项目以外的指定 **SCM** 分支或修订开始。如需更多信息，请参阅 作业分支覆盖。



4. 点 **Save** 保存您的项目。

提示

使用 **GitHub** 链接是使用 **playbook** 的一种简单方法。为了帮助您入门，请使用 此处 提供的 **helloworld.yml** 文件。

此链接提供了与 自动化控制器用户指南 中手动创建的 **playbook** 非常相似的 **playbook**。使用它不会以任何方式改变或损害您的系统。

16.1.2.2. SCM 类型 - 配置 playbook 以使用 Red Hat Insights

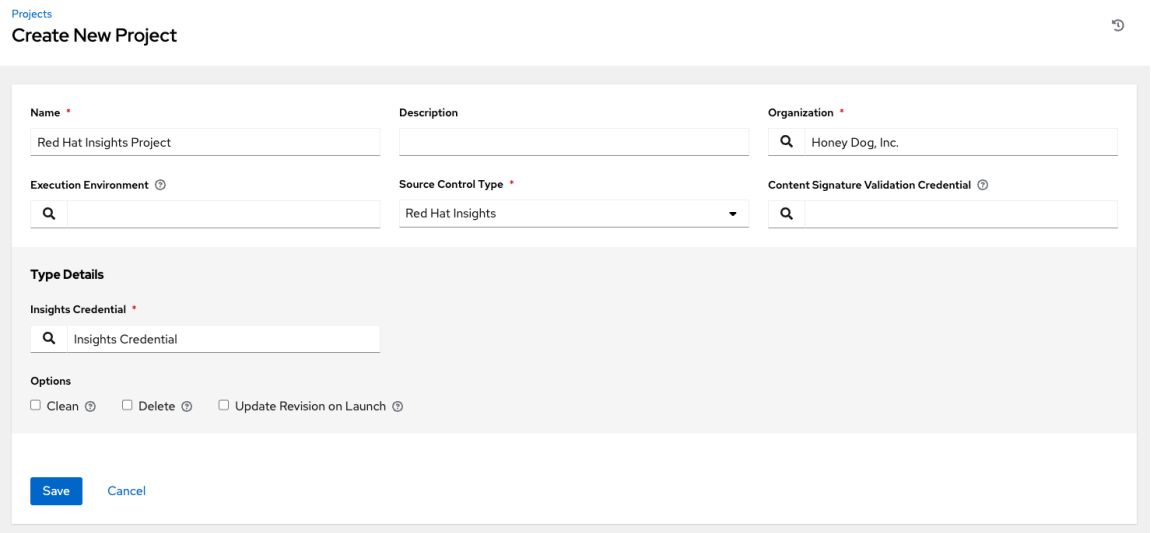
流程

1. 在 **Project Details** 页面中，从 **SCM Type** 菜单中选择 **Red Hat Insights**。

2. 在 **Credential** 字段中，选择用于 **Insights** 的适当凭证，因为 **Red Hat Insights** 需要凭证进行身份验证。

3. 可选：在 **SCM Update Options** 字段中，选择启动行为（如果适用）。

- 在 进行更新前清除任何本地修改。
- **删除 -** 在进行更新前**删**除整个本地存**储**库。根据存**储**库的大小，这可能会显著增加完成更新所需的**时**间。
- 更新启动时的 **Revision**，将项目的修订更新至远程源控制中的当前修订版本，并缓存 **Ansible Galaxy** 支持 或集合中的 **roles** 目录。**???**自动化控制器可确保本地修订版本匹配，并且角色和集合是最新的。如果生成作业的速度比项目可以同步的速度快，请选择此选项可让您将缓存超时配置为缓存以前的项目同步，以避免作业溢出。



4. 点击 **Save**。

16.1.2.3. SCM 类型 - 配置 playbook 以使用远程归档

使用远程存档的 **playbook** 使项目基于生成版本工件或发行版本的构建过程，其中包含该项目在单个存档中的所有要求。

流程

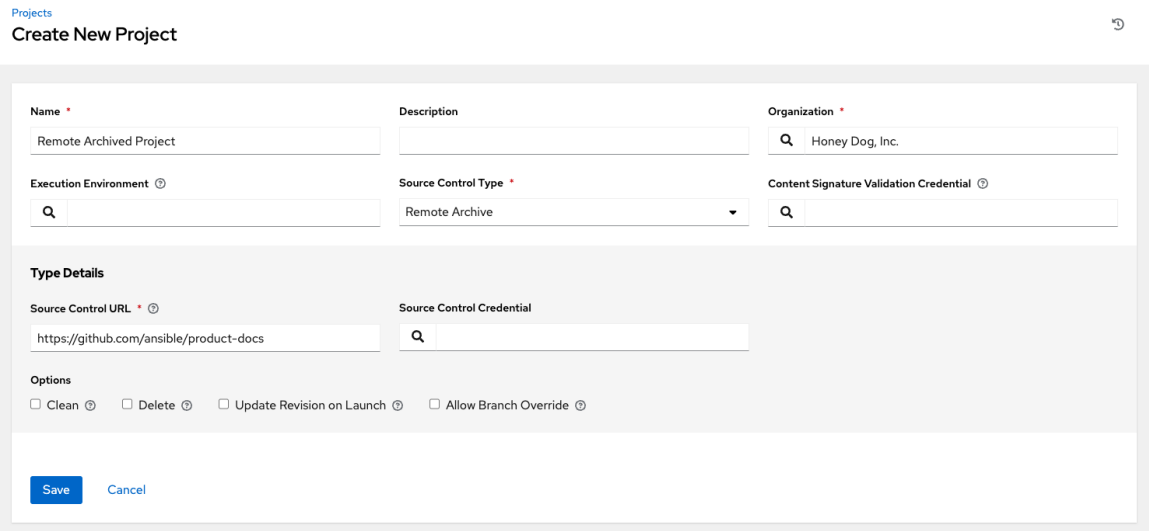
1. 在 **Project Details** 页面中，从 **SCM Type** 菜单中选择 **Remote Archive**。

2. 在以下字段中输入相关信息：

- **SCM URL -** 需要远程归档的 **URL**，如存储在 ***Artifactory*** 中的 ***GitHub*** 发行版本或构建工件，并将其解包到要使用的项目路径中。
- **SCM Credential -** 如果需要身份验证，请选择适当的 **SCM** 凭据。

3. 可选：在 **SCM Update Options** 字段中，选择启动行为（如果适用）：

- 在 进行更新前清除任何本地修改。
- **删除 -** 在进行更新前删除整个本地存储库。根据存储库的大小，这可能会显著增加完成更新所需的时间。
- 不建议 在启动时更新修订.此选项将项目的修订更新至远程源控制中的当前修订版本，并缓存 **Ansible Galaxy** 支持 或 集合 支持的 **roles** 目录。
- 不建议使用 **Branch Override -**。此选项可让使用此项目的作业模板通过项目以外的指定 **SCM** 分支或修订启动。



注意

由于此 **SCM** 类型旨在支持未更改工件的概念，因此建议禁用 **Galaxy** 集成（至少针对角色）。

4. 点击 **Save**。

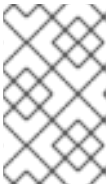
16.2. 从源控制更新项目

流程

1. 在导航面板中，选择 **Resources** → **Projects**。

2. 点您要更新的项目旁的同步

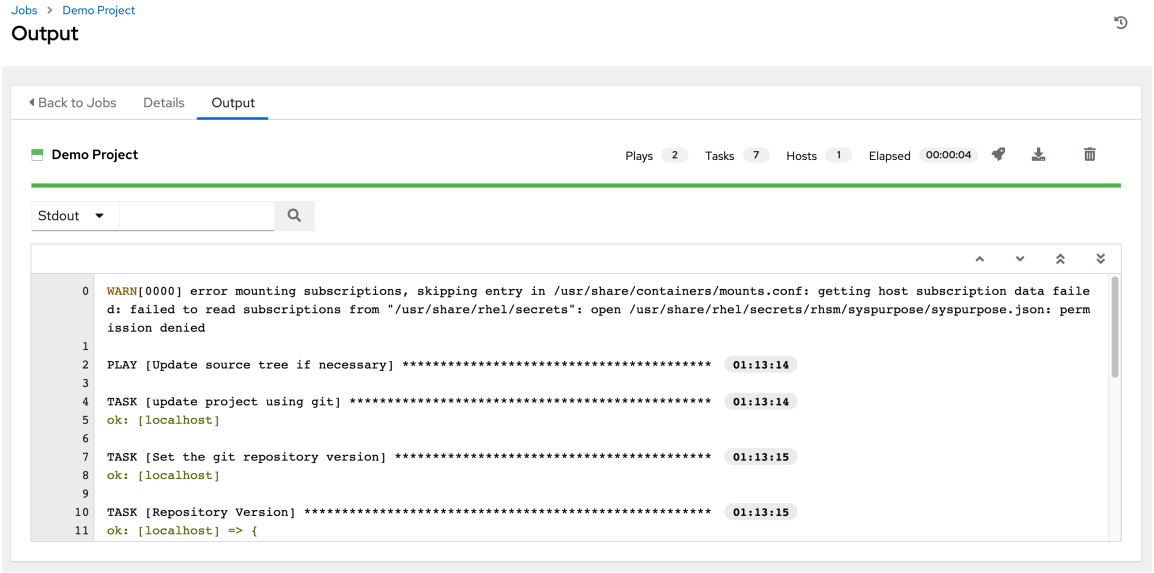
图标。



注意

在添加项目设置以使用源控制后，同步开始从配置的源控制获取项目详情。

- 单击 **Status** 列下项目的状态，以获取有关更新过程的更多信息。这会进入 **Jobs** 部分的 **Output** 选项卡。



16.3. 使用权限

分配给项目的权限集（基于角色的访问控制），可提供读取、更改和管理项目、清单、作业模板及其他元素的能力。

要访问项目权限，请选择 **Projects** 页面的 **Access** 选项卡。此屏幕显示目前具有此项目权限的用户列表。

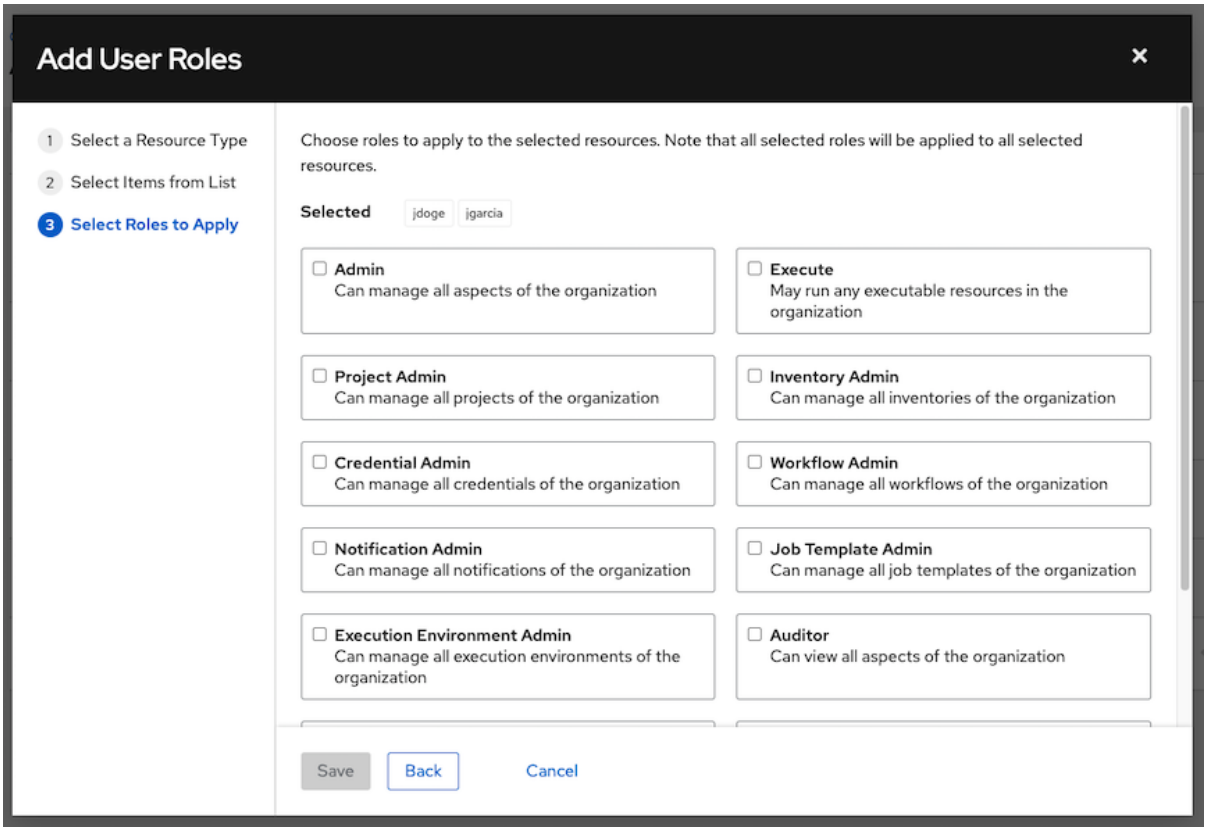
您可以根据 用户名、名字或 **Last Name** 进行排序和搜索此列表。

16.3.1. 添加项目权限

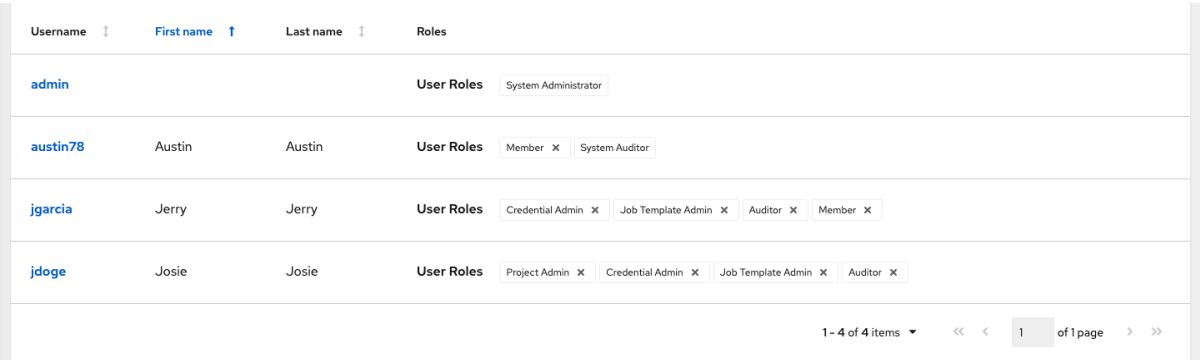
管理用户和团队需要访问项目的权限。

流程

1. 在导航面板中，选择 **Resources** → **Projects**。
2. 选择您要更新的项目，然后点 **Access** 选项卡。
3. 点**Add**。
4. 选择要添加的用户或团队，然后点 **Next**。
5. 点名称旁边的复选框从列表中选择一个或多个用户或团队，将它们添加为成员。
6. 点击 **Next**。
7. 选择您希望所选用户或团队具有的角色。务必向下滚动以获得完整的角色列表。不同的资源有不同的可用选项。



8. 点 **Save** 将角色应用到所选用户或团队，并将它们添加为成员。此时会显示为每个用户和团队分配的更新角色。



16.3.2. 从项目中删除权限

删除特定用户的角色。

流程

1. 在导航面板中，选择 **Resources** → **Projects**。

2. 选择您要更新的项目，然后点 **Access** 选项卡。

3. 点 **Roles** 列中用户角色旁边的

图标。

4. 在确认窗口中点击 **Delete** 确认解除关联。

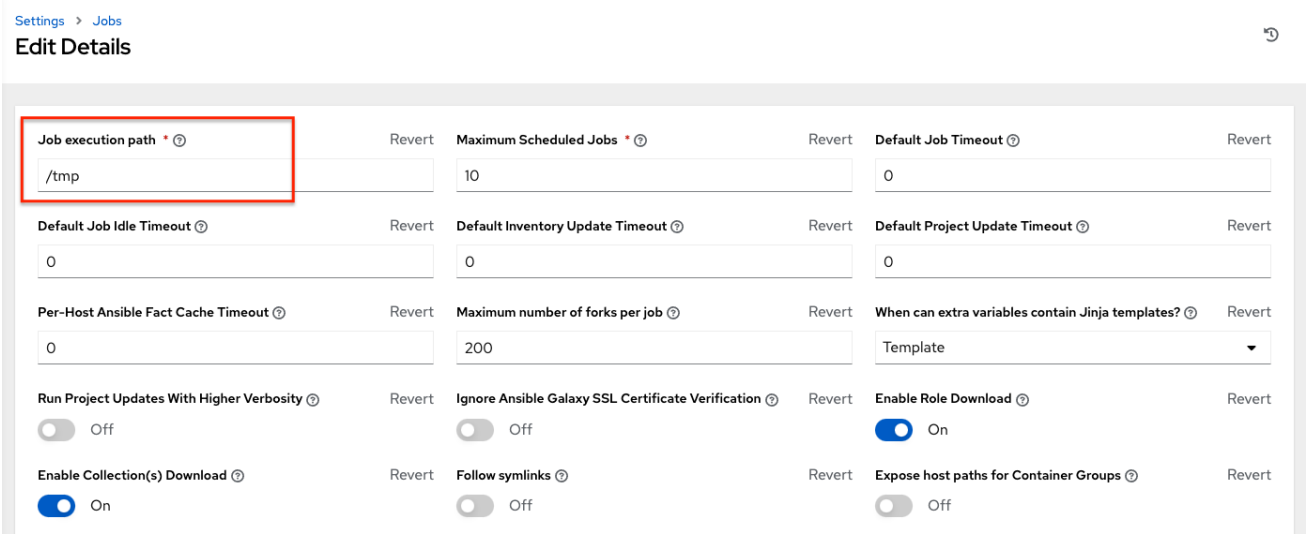
16.4. ANSIBLE GALAXY 支持

在项目更新结束时，自动化控制器会在 **roles** 目录中搜索 **requirements.yml** 文件，位于 **< project-top-level-directory>/roles/requirements.yml**。

如果找到这个文件，以下命令会自动运行：

```
ansible-galaxy role install -r roles/requirements.yml -p <project-specific cache location>/requirements_roles -vvv
```

此文件允许您引用 **Ansible Galaxy** 角色或其他可以与您自己的项目一起签出的其他存储库中的角色。增加 **Ansible Galaxy** 访问后，无需创建 **git** 子模块来实现此结果。假设 **SCM** 项目以及角色或集合是从私有作业环境中拉取并执行的，默认为创建特定于 **/tmp** 中的项目的 **<private job directory >**。但是，您可以根据 **Settings** 窗口的 **Jobs Settings** 选项卡中的环境指定另一个 作业执行路径 ：



缓存目录是全局项目文件夹中的一个子目录。内容可以从缓存位置复制到 **< job private directory>/requirements_roles**。

默认情况下，自动化控制器有一个系统范围的设置，可让您从 **SCM** 项目的 **roles/requirements.yml** 文件中动态下载角色。您可以通过将 **Enable Role Download toggle** 按钮切换到 **Off**，在 **Settings** 菜单的 **Jobs settings** 屏幕中关闭此设置。

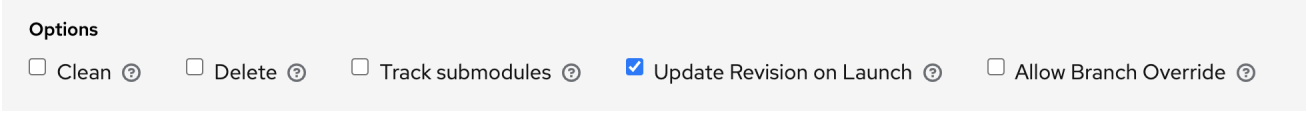
每当项目同步运行时，自动化控制器会决定项目源以及来自 **Galaxy** 或 **Collections** 的任何角色是否与项目不同步。项目更新将下载更新中的角色。

如果作业需要获取上游角色的更改，更新项目可确保发生这种情况。对角色的更改意味着新提交已推送到 ***provision-role*** 源控制。

要使此更改在作业中生效，您不必将新提交推送到 ***playbook*** 存储库。您必须更新项目，将角色下载到本地缓存中。

例如，假设您在源控制中有两个 **git** 存储库。第一个是 ***playbook***，自动化控制器中的项目指向此 **URL**。第二个是 ***provision-role***，它由 ***playbook*** **git** 存储库中的 **roles/requirements.yml** 文件引用。

每次作业运行前，作业都会下载最新的角色。由于性能的原因，角色和集合会在本地缓存。您必须在项目 **SCM** 更新选项中选择 **Update Revision on Launch**，以确保在每次作业运行前都会重新下载上游角色：



与同步相比，更新会在进程早期发生，因此这会更快地识别错误和详情。

如需有关 **requirements.yml** 文件语法的更多信息和示例，请参阅 **Ansible** 文档中的 角色要求部分。

如果有需要特别公开的目录，您可以在 **Settings** 屏幕的 **Paths to Expose to Isolated Jobs** 的 **Jobs** 部分中指定它们。您还可以更新设置文件中的以下条目：

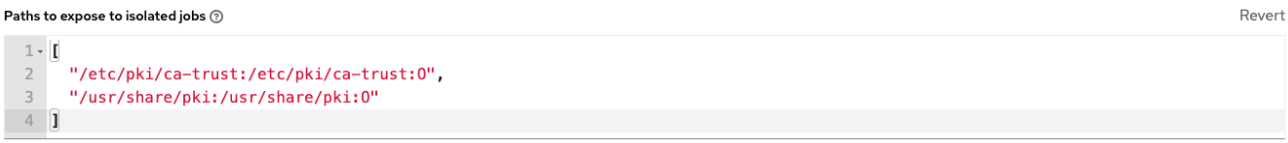
```
AWX_ISOLATION_SHOW_PATHS = ['/list/of/', '/paths']
```

注意

如果您的 **playbook** 需要使用 **AWX_ISOLATION_SHOW_PATHS** 中定义的密钥或设置，您必须将 **AWX_ISOLATION_SHOW_PATHS** 添加到 **/var/lib/awx/.ssh**。

如果您在设置文件中进行了更改，请确保在保存更改后使用 **automation-controller-service restart** 命令重启服务。

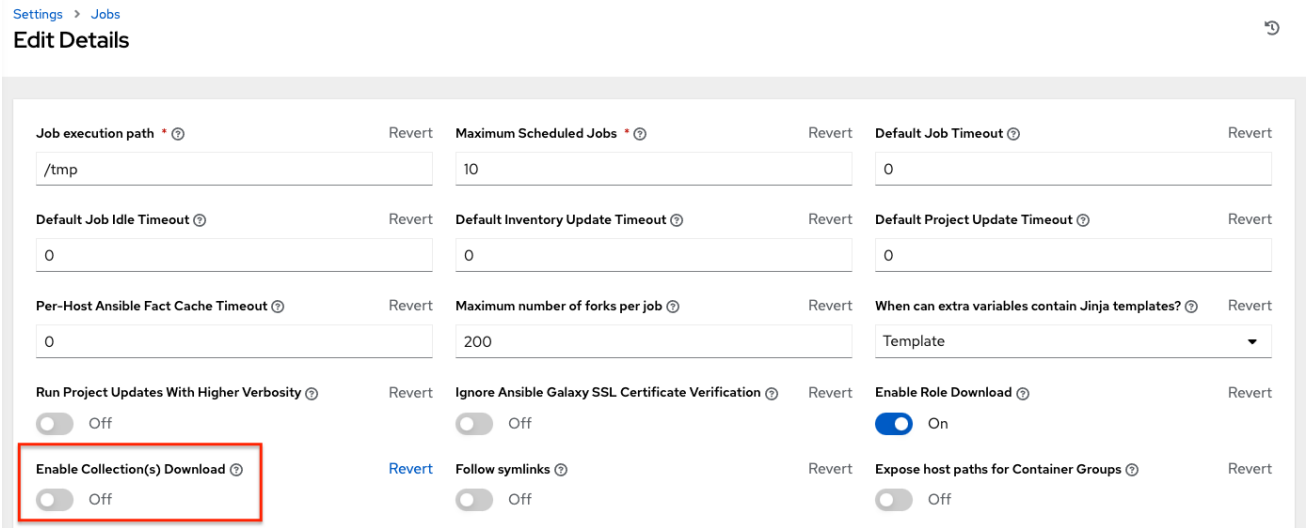
在 **UI** 中，您可以在 **Jobs settings** 窗口中配置这些设置。



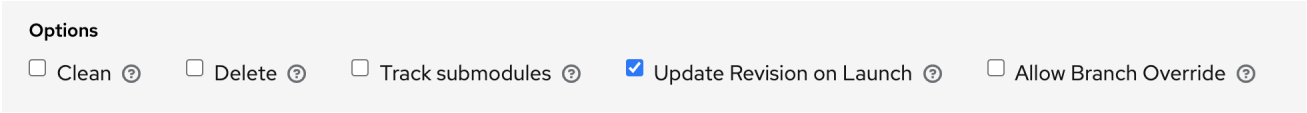
16.5. 集合支持

自动化控制器在作业运行中支持特定于项目的 **Ansible** 集合。如果您在 **SCM** 中指定集合要求文件在 **collections/requirements.yml** 处，则自动化控制器会在作业运行前在隐式项目同步中将集合安装到该文件中。

自动化控制器有一个系统范围的设置，它允许从 **SCM** 项目的 **collections/requirements.yml** 文件中动态下载集合。您可以通过将 **Enable Collections Download toggle** 按钮设置为 **Off**，在 **Settings** 菜单的 **Jobs settings** 选项卡中关闭此设置。



因为性能的原因，角色和集合会在本地缓存，您可以在项目 **SCM** 更新选项中选择 **Update Revision on Launch** 以确保：



注意

如果您在执行环境中安装了集合，则项目的 **requirements.yml** 文件中指定的集合将在运行作业时优先使用。无论集合版本是什么，这个优先级都适用。例如，如果 **requirements.yml** 中指定的集合比执行环境中的集合旧，则会使用 **requirements.yml** 中指定的集合。

16.5.1. 在自动化中心中使用集合

在自动化控制器可以使用自动化中心作为集合内容的默认源前，您必须在自动化中心 **UI** 中创建 **API** 令牌。然后，您可以在自动化控制器中指定此令牌。

使用以下步骤连接到私有自动化中心或自动化中心，唯一区别是您指定的 **URL**。

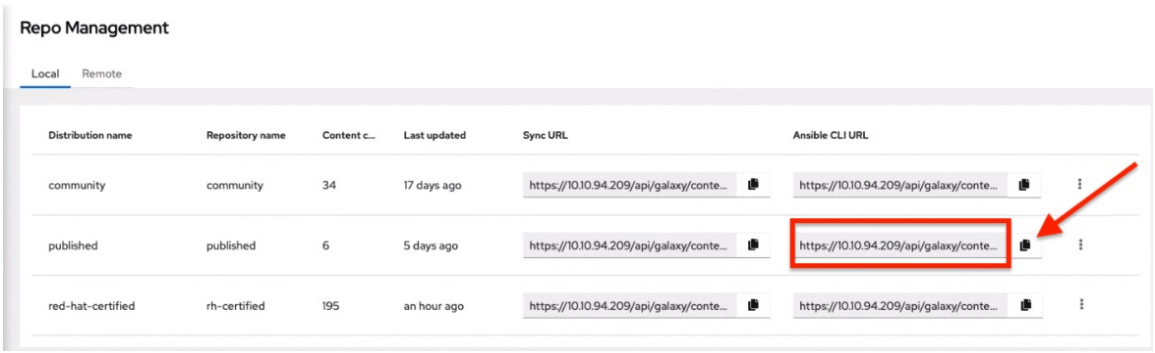
流程

1. 进入 **https://console.redhat.com/ansible/automation-hub/token**。
2. 单击 **Load token**。
3. 点复制 图标将 **API** 令牌复制到剪贴板。
4. 通过选择以下选项之一来创建凭证：

a. 要使用自动化中心，请使用复制的令牌创建一个自动化中心凭证，并指向令牌页面的 **Server URL** 和 **SSO URL** 字段中显示的 **URL**：

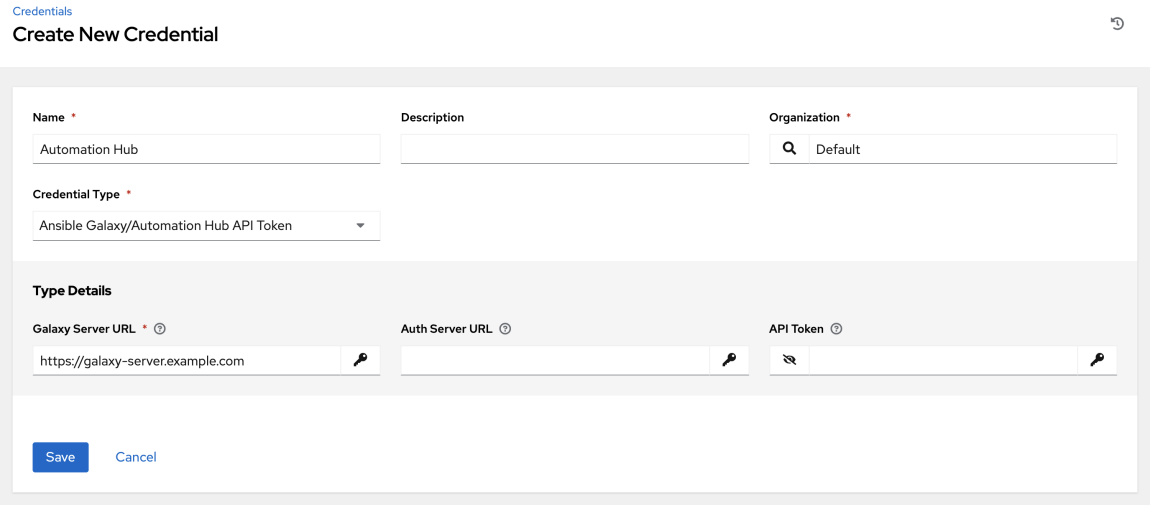
- **Galaxy Server URL = https://console.redhat.com/api/automation-hub/**
- **auth sever URL = https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/token**

b. 要使用私有自动化中心，请使用从私有自动化中心的 **Repo Management** 仪表板中获取的令牌创建一个自动化中心凭证，并指向公布的存储库 **URL**，如下所示：



您可以使用不同命名空间或集合创建不同的存储库。对于自动化中心中的每个存储库，您必须创建不同的凭证。

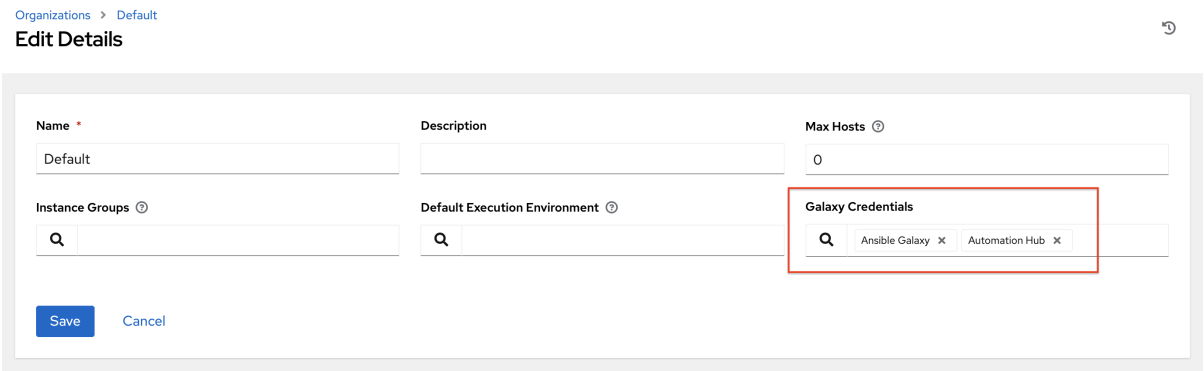
将 **UI** 中的 **Ansible CLI URL** （格式为 **/https://$<hub_url>/api/galaxy/content/<repo**）复制到 ***Create Credential*** 表单的 **Galaxy Server URL** 字段中：



有关 **UI** 具体步骤，请参阅 **Automation Hub** 中的红帽认证、验证和 **Ansible Galaxy** 内容。

5.

进入您要从中同步内容的组织，并将新凭据添加到机构中。这可让您将每个机构与要使用的内容的凭证或存储库关联。



Example

您有两个软件仓库：

- ***prod*: Namespace 1 和 Namespace 2，每个命名空间都带有集合 A 和 B，因此 namespace1.collectionA:v2.0.0 和 namespace2.collectionB:v2.0.0**

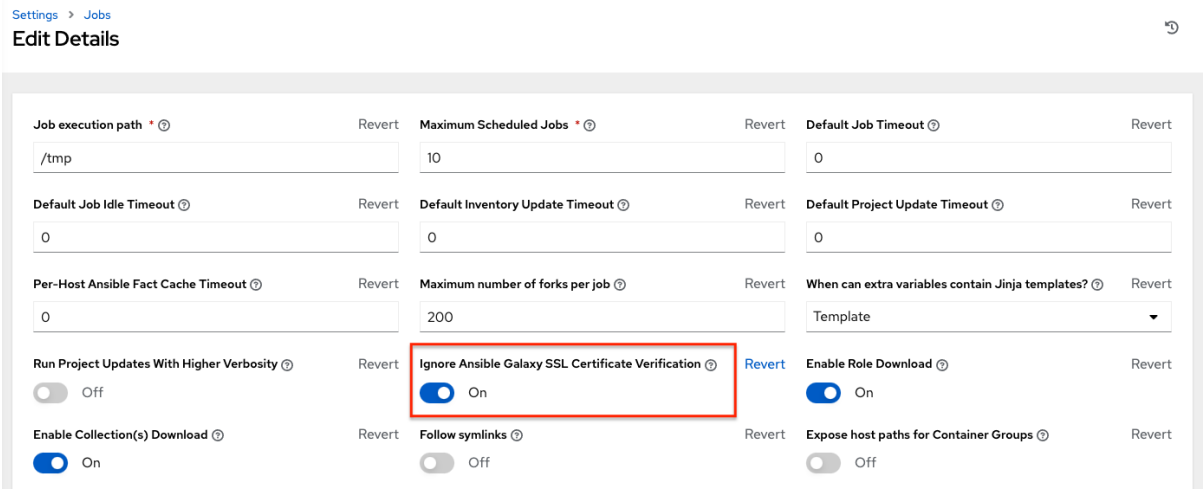
- ***stage*: Namespace 1 仅带有集合 A so: namespace1.collectionA:v1.5.0，您具有 *Prod* 和 *Stage* 的存储库 URL。**

您可以为每个凭证创建一个凭证。

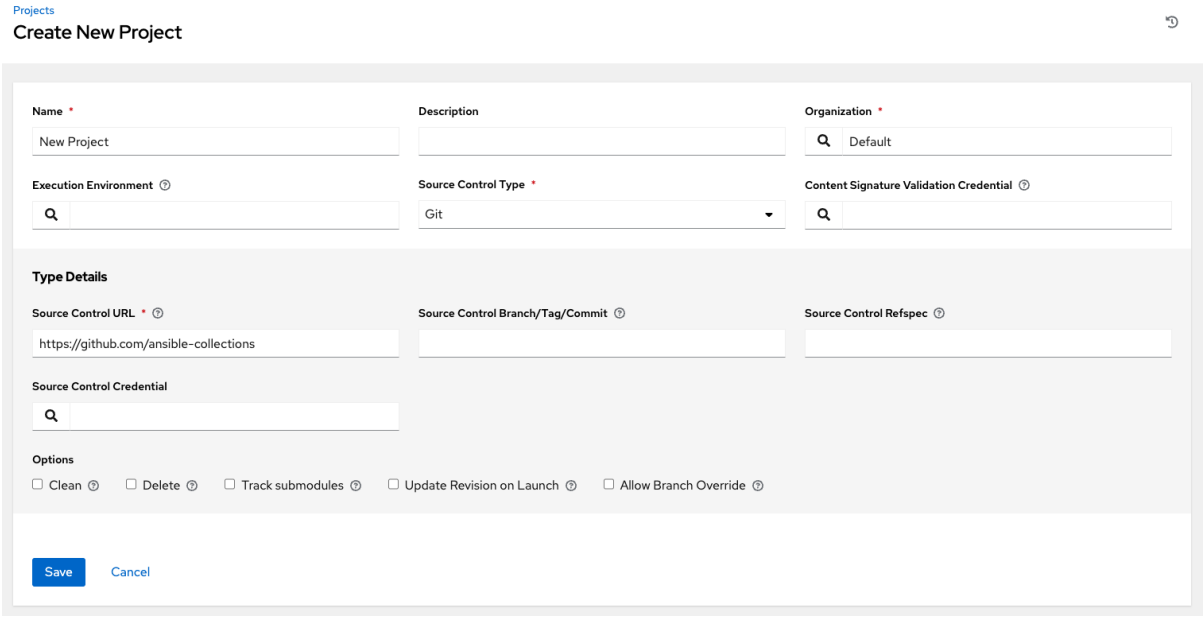
然后，您可以为不同的机构分配不同的访问权限级别。例如，您可以创建一个有权访问这两个存储库的 **Developers** 组织，而 **Operations** 组织仅有权访问 **Prod** 存储库。

有关 **UI** 具体步骤，请参阅在 私有自动化中心中为容器存储库配置用户访问权限。

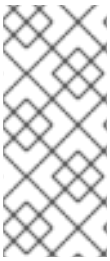
6. 如果自动化中心有自签名证书，请使用切换按钮启用设置 **Ignore Ansible Galaxy SSL Certificate Verification**。对于使用签名证书的自动化中心，请使用切换来禁用它。这是一个全局设置：



7. 创建一个项目，其中源仓库在 **collections/requirements.yml** 文件中的要求文件中指定必要的集合。有关使用的语法信息，请参阅 **Ansible** 文档中的使用 **Ansible** 集合。



8. 在 **Projects** 列表视图中，点同步 图标更新此项目。自动化控制器从 **collections/requirements.yml** 文件获取 **Galaxy** 集合，并根据更改进行报告。使用此项目为任何作业模板安装集合。



注意

如果需要从 **Galaxy** 或 **Collections** 进行更新，则会执行一个下载所需角色的同步，这会消耗您的 **/tmp** 文件中更多的空间。如果您有大型项目（大约 **10 GB**），**/tmp** 上的磁盘空间可能会出现问题。

其他资源

有关集合的更多信息，请参阅使用集合。

有关红帽如何发布其中一个官方集合（可用于直接自动化安装）的更多信息，请参阅 **AWX Ansible Collection** 文档。

第 17 章 项目签名和验证

项目签名和验证可让您在项目目录中签署文件，然后验证内容是否以任何方式更改，或者文件被意外地从项目中添加或删除。要做到这一点，您需要私钥来签名和匹配的公钥进行验证。

对于项目维护人员，支持签署内容的方式是使用 **ansible-sign** 实用程序，其 *使用提供的命令行界面* **(CLI)**。

CLI 旨在方便地使用加密技术，如 ***GNU Privacy Guard*** **(GPG)**等技术来验证项目中的文件是否未被篡改。目前，**GPG** 是唯一支持的签名和验证方法。

自动化控制器用于验证签名的内容。在与签名项目关联了匹配的公钥后，自动化控制器会验证包含的文件在签名过程中没有改变，并且文件已被意外添加或删除。如果签名无效或者文件已更改，则项目无法更新，并且使用项目的作业不会启动。项目的验证状态可确保只有安全、未被调整的内容才能在作业中运行。

如果已经为签名和验证配置了存储库，更改项目的常见工作流如下：

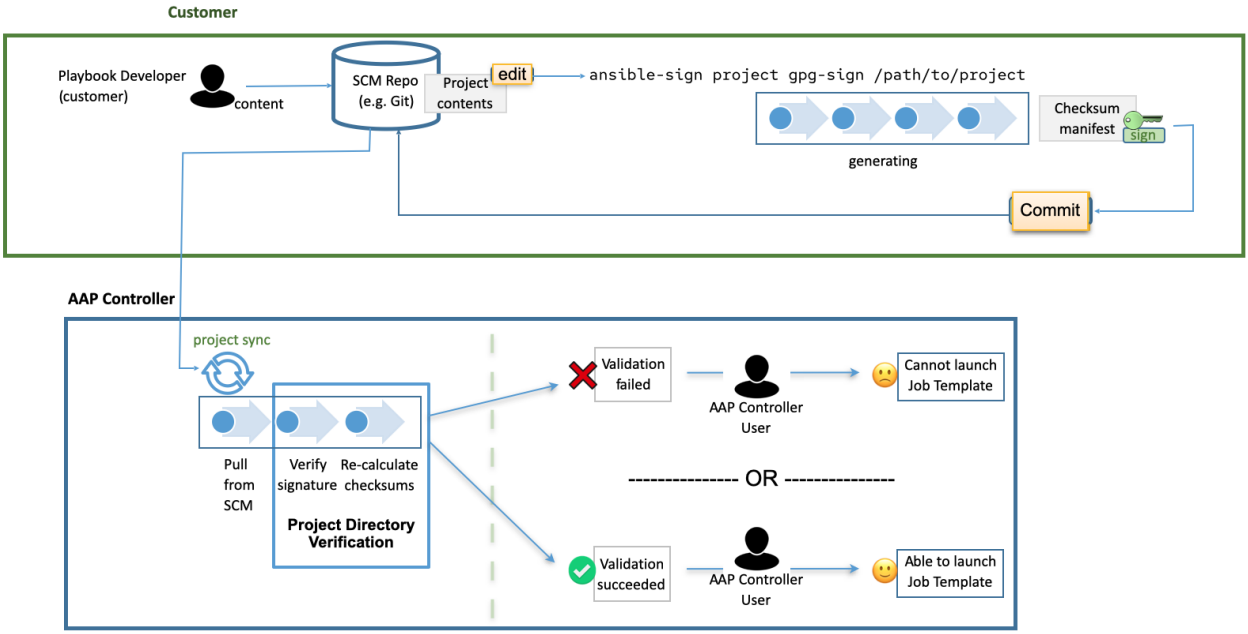
1. 您已设置了项目存储库，并希望更改文件。
2. 您可以进行更改，并运行以下命令：

```
ansible-sign project gpg-sign /path/to/project
```

此命令更新校验和清单并进行签名。

3. 您可以将更改、更新的校验和清单以及签名提交到存储库。
4. 当您同步项目时，自动化控制器会拉取新的更改，检查与自动化控制器中的项目关联的公钥是否与校验和清单签名的私钥匹配（这可以防止对校验和清单本身的篡改），然后重新计算清单中每个文件的校验和，以确保校验和匹配（从而没有更改文件）。它还确保考虑了所有文件：

文件必须包含在 **MANIFEST.in** 文件中，或排除在 **MANIFEST.in** 文件中。有关此文件的更多信息，请参阅 签名项目 **if** 文件被意外添加或删除，验证会失败。



17.1. 先决条件

- **RHEL** 节点必须正确订阅：
  - 必须启用带有 **baseos** 和 **appstream** 软件仓库的 **RHEL** 订阅。
  - 您的 **Red Hat Ansible Automation Platform** 订阅和正确的频道必须启用：

```
ansible-automation-platform-2.4-for-rhel-8-x86_64-rpms for RHEL 8
ansible-automation-platform-2.4-for-rhel-9-x86_64-rpms for RHEL 9
```

- 签名内容需要一个有效的 **GPG** 公共或私有密钥对。有关更多信息，请参阅 如何创建 **GPG** 密钥对。

  有关 **GPG** 密钥的更多信息，请参阅 **GnuPG** 文档。

  使用以下命令，验证您是否在默认的 **GnuPG** 密钥环中具有有效的 **GPG** 密钥对：

```
gpg --list-secret-keys
```

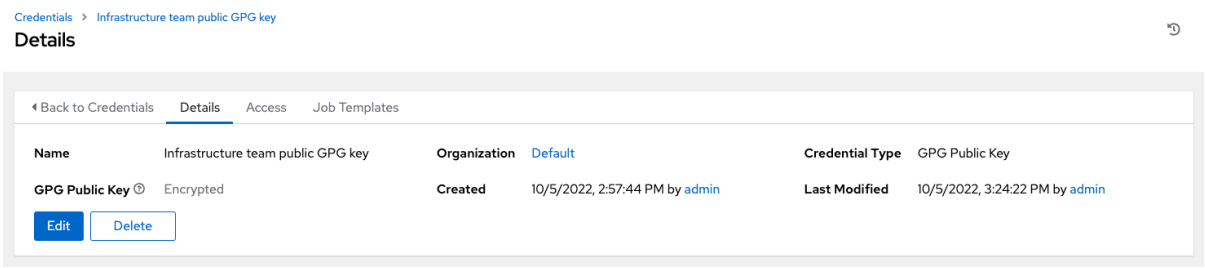
如果此命令没有生成输出，或者输出的一行内容，则 **trustdb** 已创建，那么您在默认密钥环中没有 **secret** 密钥。在这种情况下，请参阅 如何创建 **GPG** 密钥对 以了解如何创建新密钥对，然后再继续。如果生成任何其他输出，则代表您有有效的 **secret** 密钥，并可使用 **ansible-sign**。

17.2. 在自动化控制器中添加 GPG 密钥

要将 **GPG** 密钥用于自动化控制器中的内容签名和验证，请在 **CLI** 中运行以下命令来添加它：

```
$ gpg --list-keys
$ gpg --export --armour <key fingerprint> > my_public_key.asc
```

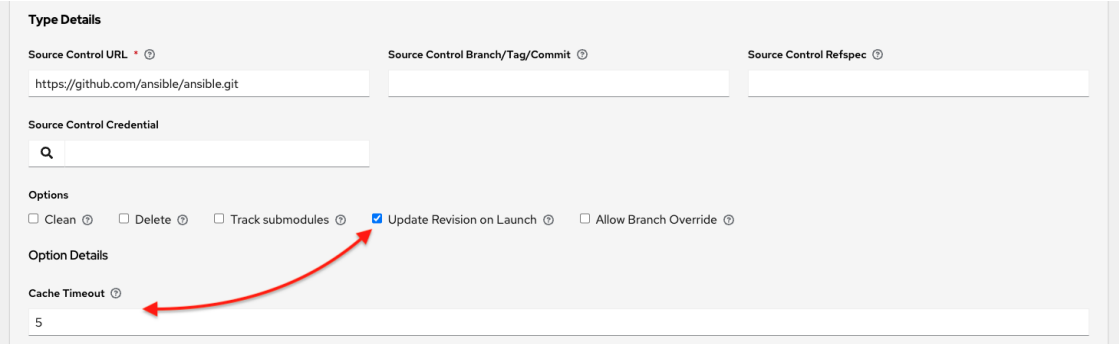
1. 在导航面板中，选择 **Resources** → **Credentials**。
2. 点**Add**。
3. 为新凭证提供一个有意义的名称，例如：**"Infrastructure team public GPG key"**。
4. 在 **Credential Type** 字段中，选择 **GPG Public Key**。
5. 单击 **Browse**，再选择公钥文件，如 **my_public_key.asc**。
6. 点击 **Save**。



现在，这个凭证可以在 项目 **<ug_projects_add>** 中被选择，内容验证会在将来的项目同步上自动进行。

注意

使用项目缓存 **SCM** 超时来控制自动化控制器重新验证签名内容的频率。当项目配置为在启动时更新（任何配置为使用该项目的作业模板），您可以启用缓存超时设置，这会在 **N** 秒后进行更新。如果验证运行太频繁，您可以通过在项目的 **Option Details** 窗格的 **Cache Timeout** 字段中指定时间来减慢项目更新的频率。



17.3. 安装 ANSIBLE-SIGN CLI 工具

使用 **ansible-sign** 实用程序为用户提供签名和验证项目是否已签名的选项。

流程

1. 运行以下命令来安装 **ansible-sign** ：

```
$ dnf install ansible-sign
```

2. 使用以下命令验证 **ansible-sign** 是否已成功安装：

```
$ ansible-sign --version
```

类似于以下内容的输出表示您已成功安装了 **ansible-sign** ：

```
ansible-sign 0.1
```

17.4. 为项目签名

对项目进行签名涉及 **Ansible** 项目目录。如需有关项目目录结构的更多信息，请参阅 **Ansible** 文档中的示例 **Ansible** 设置。

以下示例项目有一个非常简单的结构：一个清单文件，以及 **playbook** 目录下的两个小 **playbook**：

```
$ cd sample-project/
$ tree -a .
.
├── inventory
└── playbooks
    └── get_uptime.yml
    └── hello.yml

  1 directory, 3 files
```

注意

所用的命令假定您的工作目录是项目的根目录。**ansible-sign** 项目 命令将项目根目录用作其最后一个参数。

使用 **.** 表示当前工作目录。

Ansible-sign 通过对项目中所有安全文件进行校验和**(SHA256)**来保护内容，将它们编译到校验和清单文件中，然后对该清单文件进行签名。

要签署内容，请在项目根目录中创建 **MANIFEST.in** 文件，该文件告知 **ansible-sign which** 文件要保护。

在内部，**ansible-sign** 使用 **Python distlib** 库的 **distlib.manifest** 模块，因此 **MANIFEST.in** 必须遵循这个库指定的语法。有关**"MANIFEST.in"**文件指令的说明，请参阅 **Python** 打包用户指南。

在示例项目中，包括两个指令，生成以下 **MANIFEST.in** 文件：

```
include inventory
recursive-include playbooks *.yml
```

使用这个文件，生成您的校验和清单文件并对其进行签名。这两个步骤都在一个 **ansible-sign** 命令中实现：

```
$ ansible-sign project gpg-sign .
```

成功执行会显示类似如下的输出：

```
[OK   ] GPG signing successful!
[NOTE ] Checksum manifest: ./.ansible-sign/sha256sum.txt
[NOTE ] GPG summary: signature created
```

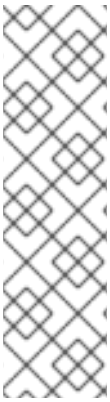
该项目现已签名。

请注意，**gpg-sign** 子命令位于 **project** 子命令下。

对于签名项目内容，每个命令都以 **ansible-sign** 项目 开头。

每个 **ansible-sign project** 命令都使用项目根目录 。 作为其最终参数。

Ansible-sign 使用默认密钥环，并查找可找到的第一个可用 **secret** 密钥来为您的项目签名。您可以使用**-- fingerprint** 选项指定要使用的特定 **secret** 密钥，甚至使用 **--gnupg-home** 选项指定完全独立的 **GPG** 主目录。



注意

如果您使用桌面环境，**GnuPG** 会自动提示您输入 **secret** 密钥的密码短语。

如果此功能无法正常工作，或者您在没有桌面环境的情况下工作，例如通过 **SSH**，您可以在 **gpg-sign** 后面使用 **the -p --prompt-passphrase** 标志，这会导致 **ansible-sign** 提示输入密码。

请注意，在项目目录中创建了一个 **.ansible-sign** 目录。此目录包含校验和清单以及分离的 **GPG** 签名。

```
$ tree -a .
.
├── .ansible-sign
│   ├── sha256sum.txt
│   └── sha256sum.txt.sig
```

```
├── inventory
├── MANIFEST.in
└── playbooks
    ├── get_uptime.yml
    └── hello.yml
```

17.5. 验证您的项目

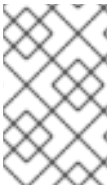
要验证已签名的 **Ansible** 项目没有被更改，您可以使用 **ansible-sign** 检查签名是否有效，以及文件的校验和是否与应该是什么的校验和匹配。**ansible-sign project gpg-verify** 命令可用于自动验证这两个条件。

```
$ ansible-sign project gpg-verify .
[OK   ] GPG signature verification succeeded.
[OK   ] Checksum validation succeeded.
```

注意

默认情况下，**ansible-sign** 使用您的默认 **GPG** 密钥环来查找匹配的公钥。您可以使用 **-keyring** 选项指定 密钥环文件，或使用**-- gnugpg-home** 选项指定不同的 **GPG** 主页。

如果出于某种原因验证失败，则会显示信息以帮助您调试原因。通过在命令中的 **ansible-sign** 后立即传递 **global-- debug** 标志来启用更详细的详细程度。



注意

在项目中使用 **GPG** 凭据时，内容验证会在将来的项目同步上自动发生。

17.6. 自动签名

在具有高度可信 *持续集成* **(CI)**环境（如 **OpenShift** 或 **Jenkins**）的环境中，可以自动执行签名过程。

例如，您可以将 **GPG** 私钥存储在一个选择的 **CI** 平台上作为一个 **secret**，并在 **CI** 环境中将其导入到 **GnuPG** 中。然后，您可以通过普通的 **CI** 环境中的签名工作流运行。

使用 **GPG** 对项目进行签名时，环境变量 **ANSIBLE_SIGN_GPG_PASSPHRASE** 可以设置为签名密钥的密码短语。这可以在 **CI** 管道中注入和屏蔽或保护。

根据场景，在签名和验证过程中，**ansible-sign** 会以不同的退出代码返回。这在 **CI** 和自动化上下文中也很有用，因为 **CI** 环境根据故障而不同。例如，它可以发送一些错误的警报，但对其他错误有静默失败。

这些是 **ansible-sign** 中使用的当前退出代码，可被视为稳定：

退出代码	大约含义	示例情境
0	成功	• 签名成功 • 验证成功
1	常规故障	• 验证和清单文件在验证过程中包含语法错误 • 验证过程中不存在签名文件 • 在签名过程中不存在 **MANIFEST.in**
2	验证和验证失败	• 在验证过程中计算的校验和哈希值与签名校验和清单中的内容不同，例如，项目文件已更改，但签名过程不会被重新完成。
3	签名验证失败	• 签名人的公钥不在用户的 GPG 密钥环中 • 指定了错误的 GnuPG 主目录或密钥环文件 • 签名的验证和清单文件被修改
4	签名进程失败	• 签名人的私钥没有在 GPG 密钥环中找到 • 指定了错误的 GnuPG 主目录或密钥环文件

第 18 章 清单

Red Hat Ansible Automation Platform 使用清单文件，根据您以逻辑方式组织的基础架构中的受管节点或主机列表进行工作。您可以使用 **Red Hat Ansible Automation Platform** 安装程序清单文件指定您的安装场景，并描述 **Ansible** 的主机部署。通过使用清单文件，**Ansible** 可以通过一个命令管理大量主机。清单还可以通过减少您指定的命令行选项数目来更有效地使用 **Ansible**。清单被分成不同的组，这些组包含主机。

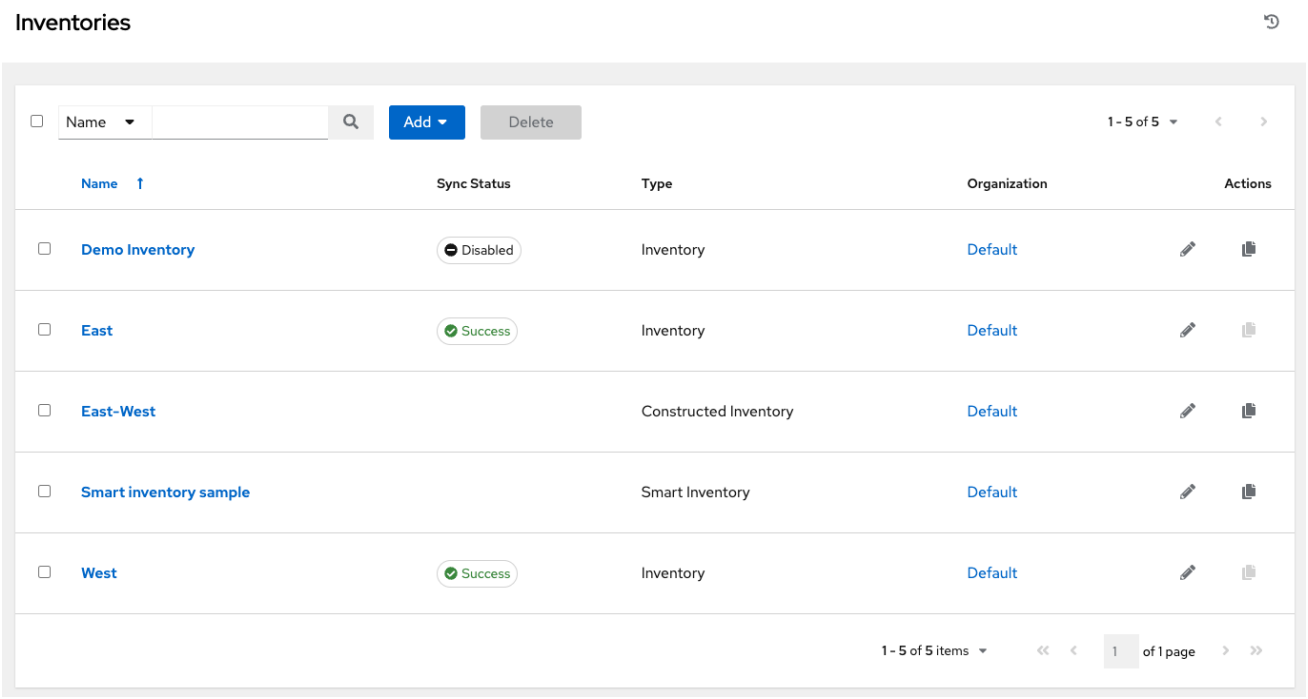
组可以通过将主机名输入到自动化控制器或其支持的云供应商来手动提供。

注意

如果您有自定义动态清单脚本，或者尚未在自动化控制器中受到原生支持的云供应商，您也可以将其导入到自动化控制器中。

如需更多信息，请参阅 *自动化控制器管理指南中的* 清单文件导入。

在导航面板中，选择 **Resources** → **Inventories**。**Inventories** 窗口显示当前可用的清单列表。您可以根据 名称、类型 或 **Organization** 排序清单列表。



清单详情页面 包括：

- **Name** ：清单名称。
- **Status**

状态为：

- 成功 ：清单源同步成功完成
- **disabled:** 没有添加到清单中的清单源
- **Error:** 当清单源同步完成并出错时
  - **Type:** 标识它是标准清单、智能清单还是构建的清单。
  - **Organization:** 清单所属的机构。
  - **Actions:** 以下操作可用于所选清单：
- 编辑 
  ：编辑所选清单的属性
- 复制 
  ：制作现有清单的副本作为创建新清单的模板

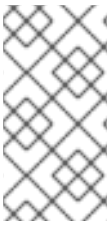
单击 **Inventory** 名称，以显示所选清单的 **Details** 页面，其中显示清单的组和主机。

18.1. 智能清单

智能清单是由存储的搜索定义的主机集合，可以像标准清单一样查看，可轻松用于作业运行。机构管理员对其机构中的清单具有 **admin** 权限，并可创建智能清单。

智能清单由 **KIND=smart** 标识。

您可以使用与搜索相同的方法定义智能清单。**InventorySource** 与清单直接关联。



注意

智能清单已弃用，并将在以后的发行版本中删除。考虑移至构建的清单以进行功能增强和替换。

清单 模型具有以下新字段，默认为空白，但会针对智能清单进行相应设置：

- 对于智能清单，**kind** 设置为 **smart**。
- **host_filter** 是针对智能清单设置的 **AND kind**，设置为 **smart**。

主机模型 具有一个相关的端点 **smart_inventories**，用于标识主机所关联的所有智能清单的集合。每次作业针对智能清单运行时，都会更新成员资格表。

注意

要更频繁地更新成员资格，您可以将 **AWX_REBUILD_SMART_MEMBERSHIP** 文件的设置改为 **True**。（默认为 **False**）。如果发生以下事件，这个更新成员资格：

- 添加了新主机
- 修改（更新或删除）现有主机
- 添加了新智能清单
- 修改（更新或删除）现有智能清单

您可以在不编辑的情况下查看清单：

- 清单源同步后创建的主机和组的名称。
- 组记录无法编辑或移动。

您不能像普通清单一样从智能清单主机端点(**/inventories/N/hosts/**)创建主机。智能清单的管理员具有编辑名称、描述、变量以及删除功能等字段的权限，但没有修改 **host_filter** 的权限，因为这会影响智能清单中包含哪些主机（在另一个清单中具有主要成员资格）。

host_filter 仅适用于智能清单机构中清单内的主机。

要修改 **host_filter**，您必须是清单机构的机构管理员。机构管理员具有对机构内所有清单的隐式"管理员"访问权限，因此，这不会使他们尚未拥有的任何权限。

智能清单的管理员可以向智能清单授予其他用户（不是您机构的管理员）权限。它们允许角色指示的操作，与其他标准清单一样。但是，这不会为主机（位于不同的清单中）授予任何特殊权限。它不允许主机直接读取权限，或者允许它们查看 **/#/hosts/** 下的其他主机，但它们仍然可以查看智能清单主机列表下的主机。

在有些情况下，您可以修改以下内容：

- 使用清单源在清单上手动创建的新主机。
- 清单源同步后创建的组。
- 主机和组上的变量不可更改，即使作为本地系统管理员也是如此。

与智能清单关联的主机会在查看时显示。如果智能清单的结果包含多个具有相同主机名的主机，则只会包含一个匹配的主机作为智能清单的一部分，按主机 **ID** 排序。

18.1.1. 智能主机过滤器

您可以使用搜索过滤器为清单填充主机。此功能使用事实搜索功能。

每当每个作业模板设置了 **use_fact_cache=True** 时，自动化控制器会在数据库中保存 **Ansible playbook** 生成的事实。新事实与现有事实合并，并按主机合并。这些存储的事实可用于使用 **GET** 查询参数 **host_filter** 使用 **/api/v2/hosts** 端点过滤主机。

例如：

```
/api/v2/hosts?host_filter=ansible_facts__ansible_processor_vcpus=8
```

host_filter 参数允许：

- 使用（）进行分组
- 使用布尔值和运算符：
  - __ 用于引用关系字段中的相关字段

- **ansible_facts** 中的 __ 用于分隔 **JSON** 密钥路径中的键
- **[]** 用于表示路径规格中的 **json** 数组
- 当值中需要空格时，"" 可以在值中使用

- **"Classic" Django** 查询可以嵌入到 **host_filter**中

示例：

```
/api/v2/hosts/?host_filter=name=localhost
/api/v2/hosts/?host_filter=ansible_facts__ansible_date_time__weekday_number="3"
/api/v2/hosts/?host_filter=ansible_facts__ansible_processor[]="GenuineIntel"
/api/v2/hosts/?host_filter=ansible_facts__ansible_lo__ipv6[]__scope="host"
/api/v2/hosts/?host_filter=ansible_facts__ansible_processor_vcpus=8
/api/v2/hosts/?host_filter=ansible_facts__ansible_env__PYTHONUNBUFFERED="true"
/api/v2/hosts/?host_filter=(name=localhost or name=database) and (groups__name=east or
groups__name="west coast") and ansible_facts__an
```

您可以根据主机名、 组名称 和 **Ansible** 事实 来搜索 **host _filter**。

组搜索具有以下格式：

```
groups.name:groupA
```

事实搜索具有以下格式：

```
ansible_facts.ansible_fips:false
```

您还可以执行由主机名和主机描述组成的智能搜索。

```
host_filter=name=my_host
```

注意

如果 **host_filter** 中的搜索词为字符串类型，要使值成为数字（如 **2.66**）或 **JSON** 关键字（如 **null**、**true** 或 **false**）有效，请在值之间添加双引号以防止控制器将其解析为非字符串：

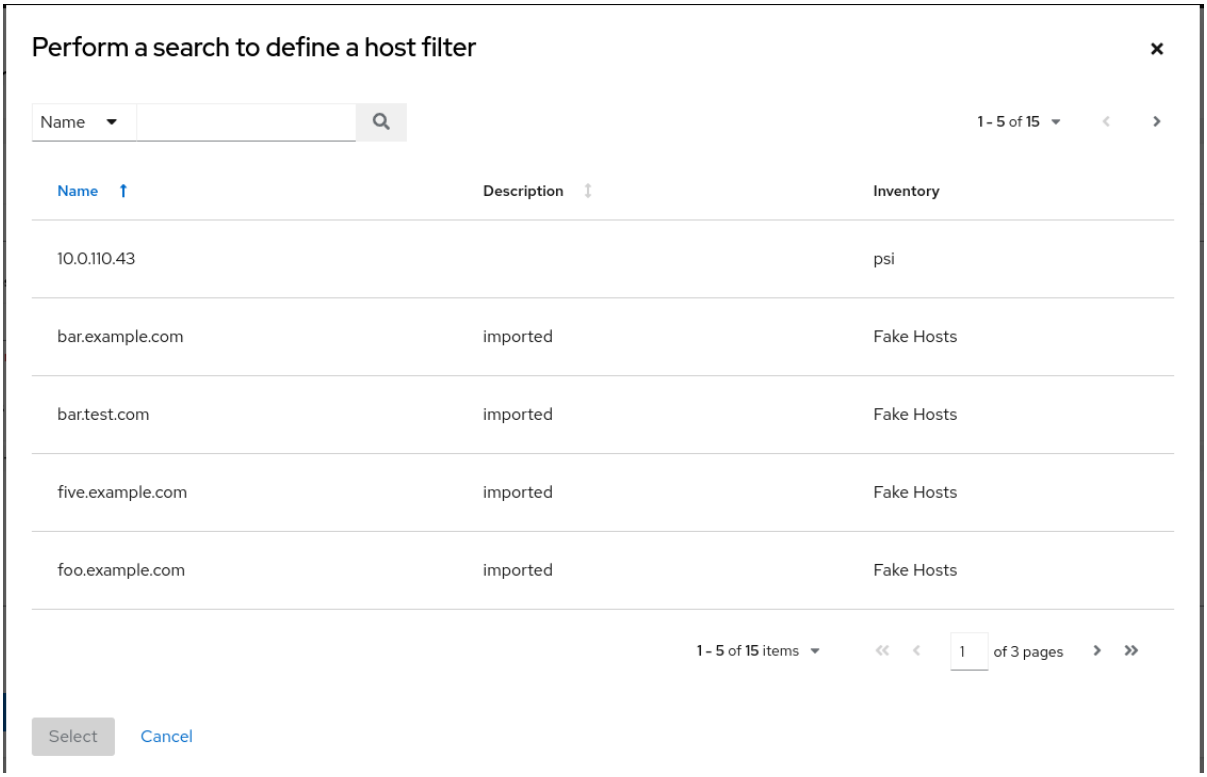
```
host_filter=ansible_facts__packages__dnsmasq[]__version="2.66"
```

18.1.2. 使用 ansible_facts 定义主机过滤器

使用以下步骤在创建智能清单时使用 **ansible_facts** 定义主机过滤器。

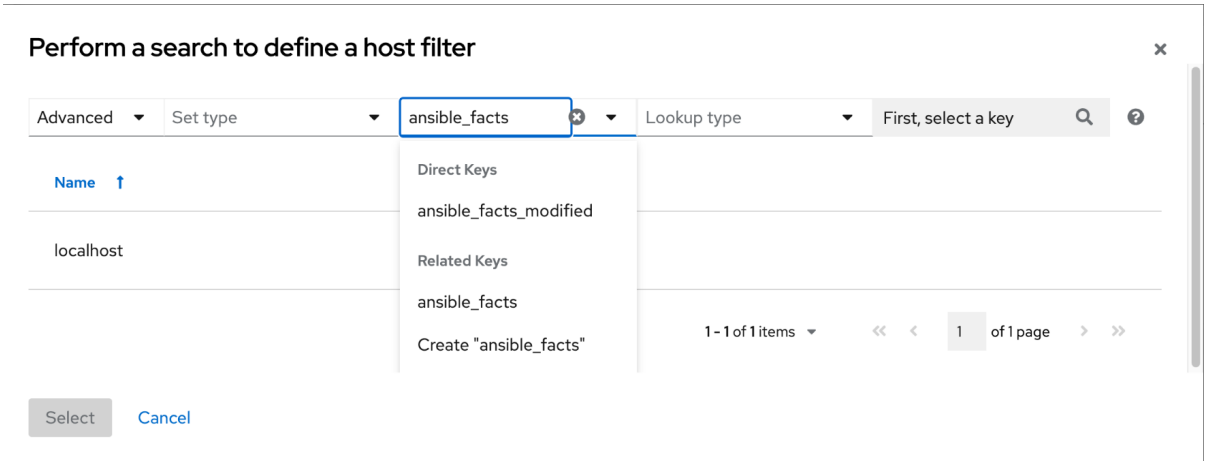
流程

1. 在导航面板中，选择 **Resources** → **Inventories**。
2. 从 **Add** 列表中选择 **Add Smart Inventory**。
3. 在 **Create new Smart inventory** 页面中，点 **Smart host filter** 字段中的 图标。这会打开一个窗口，用于过滤此清单的主机。



4.

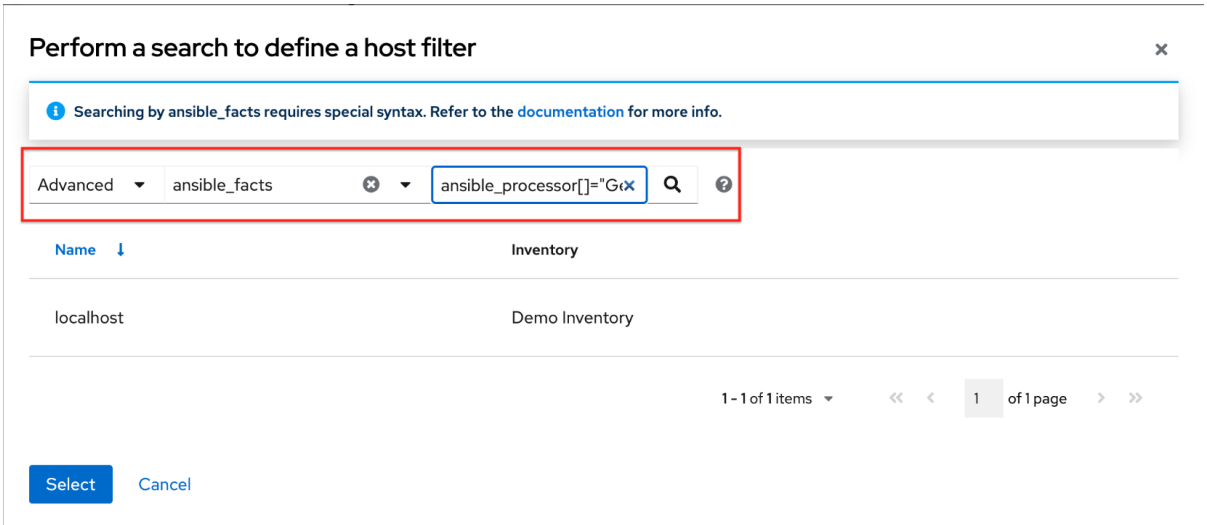
在搜索菜单中，将搜索条件从 **Name** 更改为 **Advanced**，然后从 **Key** 字段中选择 **ansible_facts**。



如果要添加以下 **ansible** 事实：

```
/api/v2/hosts/?host_filter=ansible_facts__ansible_processor[]="GenuineIntel"
```

在搜索字段中，输入 **ansible_processor[]="GenuineIntel"**（在值前面没有额外空格或 __），然后单击 **Enter**。



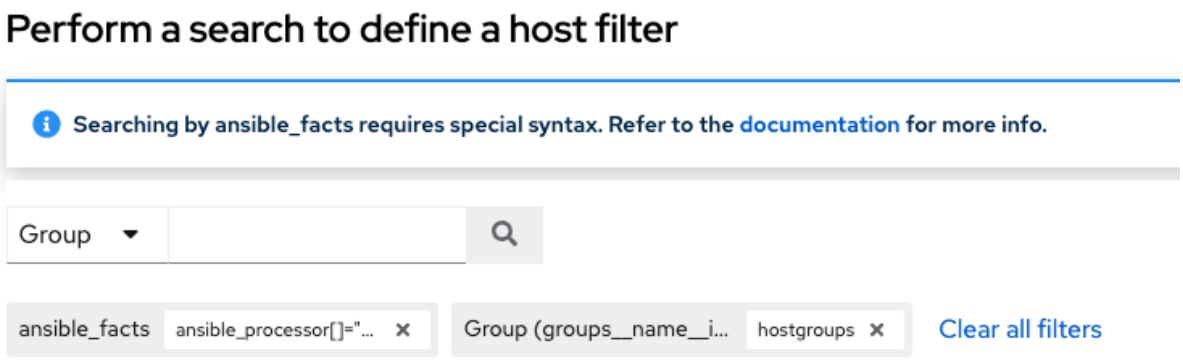
此时会显示指定 **ansible** 事实的搜索条件。

5. 点 **Select** 将其添加到 **Smart host** 过滤器 字段。

6. 点击 **Save**。

7. 新智能清单的 **Details** 选项卡会打开，并在 **Smart host filter** 字段中显示指定的 **ansible** 事实。

8. 在 **Details** 视图中，您可以通过点 **Edit and delete existing filters**、清除所有现有过滤器或添加新过滤器来编辑 **Smart host filter** 字段。



18.2. 构建的清单

您可以从输入清单列表中创建新清单（称为构建的清单）。

构建的清单包含其输入清单中的主机和组副本，允许作业在多个清单间针对服务器组。可将组和 **hostvars** 添加到清单内容中，并且可以过滤主机来限制构建的清单的大小。

构建的清单使用 **ansible.builtin.constructed** 清单 模型。

构建的清单的关键因素是：

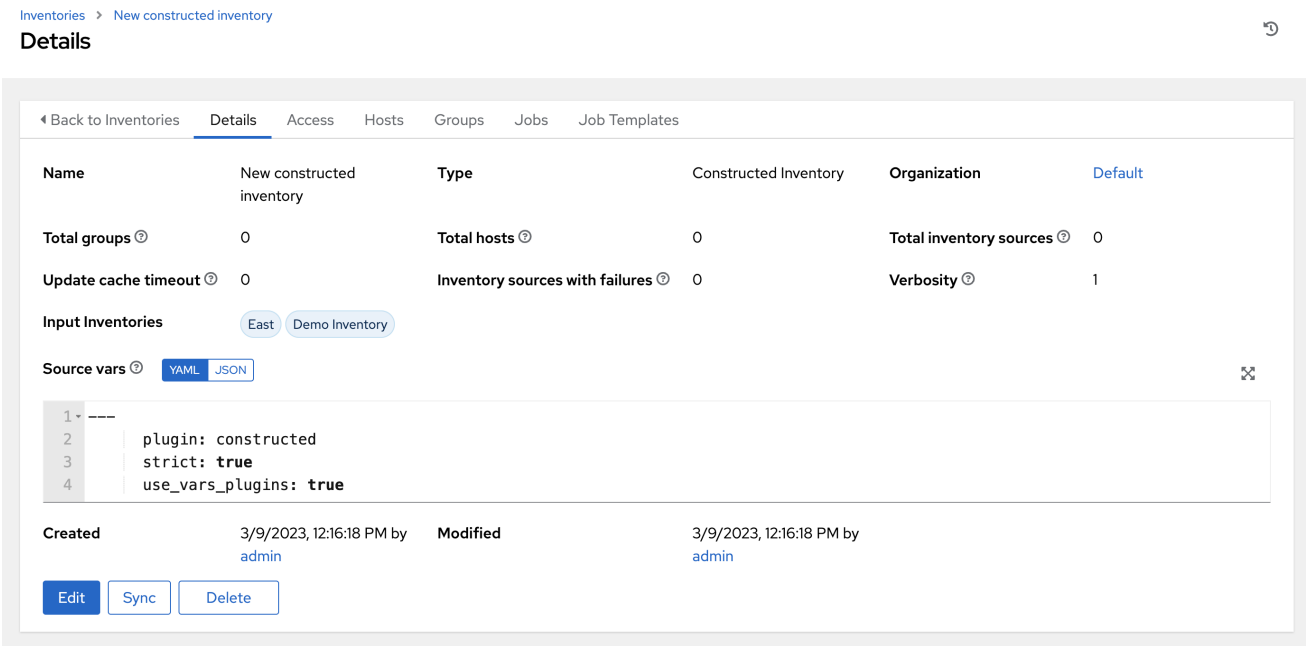
- 普通的 **Ansible hostvars** 命名空间可用
- 它们提供组

构建的清单取 **source_vars** 和 **limit** 作为输入，并将其 **input_inventories** 转换为新清单，随组一起完成。组（现有或构建）可以在 **limit** 字段中引用，以减少生成的主机数量。

您可以根据以下主机属性构建组：

- **RHEL** 主版本或次版本
- **Windows** 主机
- 在特定区域中标记的基于云的实例
- 其他

以下是构建的清单详情视图的示例：



后续小节中描述的示例由输入清单的结构进行组织。

18.2.1. 过滤组名称和变量

您可以根据组和变量的组合进行过滤。例如，您可以过滤与 **group** 变量值匹配的主机，同时匹配主机变量值。

执行此过滤器的方法有两种：

- 定义两个组：一个组来匹配 **group** 变量，另一个组与主机变量值匹配。使用 限制 模式返回两个组中的主机。这是推荐的方法。
- 定义一个组。在定义中，包含组和主机变量必须与特定值匹配的条件。使用 限制 模式返回新组中的所有主机。

Example:

以下清单文件定义了四个主机，并设置组和主机变量。它定义了一个产品组，一个 **keeping group**，它会将两个主机设置为关闭状态。

目标是创建一个仅返回关闭生产主机的过滤器。

```
[account_1234]
host1
host2 state=shutdown

[account_4321]
host3
host4 state=shutdown

[account_1234:vars]
account_alias=product_dev

[account_4321:vars]
account_alias=sustaining
```

此处的目标是仅返回与 **product_dev** 相等的 **account_alias** 变量所存在的主机。这种方法有两种，它们都以 **YAML** 格式显示。建议第一个建议。

1. 构造 **2** 个组，限制为交集 ：

   source_vars ：

   ```
   plugin: constructed
   strict: true
   groups:
     is_shutdown: state | default("running") == "shutdown"
     product_dev: account_alias == "product_dev"
   ```

   限制**:is_shutdown:&product_dev**

   这个构建的清单输入会为这两个类别创建一个组，并使用 限制 （主机模式）来仅返回位于这两个组的交集中的主机，如 **Patterns:targeting hosts 和 groups**。

   当变量为 或 未定义（取决于主机）时，您可以指定一个默认值。例如，如果您知道未定义值，请使用 | **default ("running")**。这有助于进行调试，如 调试提示 中所述。

2. 构造 **1** 组，限制为组 ：

source_vars ：

```
plugin: constructed
strict: true
groups:
  shutdown_in_product_dev: state | default("running") == "shutdown" and account_alias ==
"product_dev"
```

限制**:shutdown_in_product_dev**

此输入会创建一个组，它只包含匹配这两个条件的主机。然后，限制本身只是组名称，返回 **host2**。与之前的方法相同。

18.2.2. 调试提示

务必要将 **strict** 参数设置为 **true**，以便您可以调试模板的问题。如果模板无法呈现，则会在该构建的清单关联的清单更新中发生错误。

遇到错误时，会增加详细程度来获取更多详细信息。

提供默认值，如 **| default ("running")** 是 **Ansible** 中的 **Jinja2** 模板的通用使用。设置 **strict: true** 时，这可避免模板的错误。

您还可以设置 **strict: false**，因此启用模板生成错误，这会导致主机没有包含在该组中。但是，如果模板继续复杂性，则在以后无法调试问题。

如果没有生成预期的清单内容，您可能仍必须调试模板的预期功能。例如，如果 组 组有一个复杂的过滤器（如 **shutdown_in_product_dev**），但没有包含因构建的清单中的任何主机，则使用 **compose** 参数来帮助调试。

例如：

```
source_vars:

plugin: constructed
strict: true
groups:
  shutdown_in_product_dev: state | default("running") == "shutdown" and account_alias ==
```

```
"product_dev"
compose:
  resolved_state: state | default("running")
  is_in_product_dev: account_alias == "product_dev"

limit: ``
```

使用空白 限制 运行会返回所有主机。您可以使用此选项检查特定主机上的特定变量，从而深入了解 组中 的问题。

18.2.3. 嵌套组

嵌套组由两个组组成，其中一个是另一个组的子组。在以下示例中，子组在其中有另一个主机，父组定义了一个变量。

由于 **Ansible** 核心运行方式，父组的变量在命名空间中作为 **playbook** 正在运行，并可用于过滤。

以下示例清单文件 **nested.yml** 是 **YAML** 格式：

```
all:
  children:
    groupA:
      vars:
        filter_var: filter_val
      children:
        groupB:
          hosts:
            host1: {}
    ungrouped:
      hosts:
        host2: {}
```

由于 **host1** 位于 **groupB** 中，因此它也位于 **groupA** 中。

过滤嵌套组名称

使用以下 **YAML** 格式过滤嵌套组名称：

```
`source_vars`:

plugin: constructed

`limit`: `groupA`
```

–

过滤嵌套组属性

使用以下 **YAML** 格式过滤组变量，即使主机间接是该组的成员。

请注意，在清单内容中，**host2** 不应定义变量 **filter_var**，因为它不在任何组中。由于使用了 **strict: true**，因此请使用默认值，以便定义没有该变量的主机。使用此选项时，**host2** 会从表达式返回 **false**，而不是生成错误。**host1** 从其组中继承 变量，并且返回。

```
source_vars:

plugin: constructed
strict: true
groups:
  filter_var_is_filter_val: filter_var | default("") == "filter_val"

limit: filter_var_is_filter_val
```

18.2.4. Ansible 事实

要创建具有 **Ansible** 事实的清单，您必须针对具有设置 **gather_facts: true** 的清单运行一个 **playbook**。事实因 **system-to-system** 而异。以下示例并不能解决所有已知的场景。

18.2.4.1. 过滤环境变量

以下示例涉及使用 **YAML** 格式在环境变量上过滤：

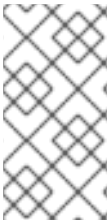
```
source_vars:

plugin: constructed
strict: true
groups:
  hosts_using_xterm: ansible_env.TERM == "xterm"

limit: hosts_using_xterm
```

18.2.4.2. 根据处理器类型过滤主机

以下示例涉及使用 **YAML** 格式根据处理器类型**(Intel)**过滤主机：

```
source_vars:

plugin: constructed
```

```
strict: true
groups:
  intel_hosts: "GenuineIntel" in ansible_processor

limit: intel_hosts
```



注意

构建中的主机不会根据您的许可证计算，因为它们引用原始清单主机。此外，原始清单中禁用的主机也不包含在构建的清单中。

使用 **ansible-inventory** 运行清单更新将创建构建的清单内容。

这始终配置为在作业前更新时启动，但您仍可以选择缓存超时值，以防这用时过长。

在创建构建的清单时，**API** 会确保它始终关联了一个清单源。所有清单更新都有一个关联的清单源，构建的清单**(source_vars** 和 **limit)**所需的字段都已存在于清单源模型中。

18.3. 清单插件

清单更新使用动态生成的 **YAML** 文件，这些文件由相应的清单插件解析。在自动化控制器 **v4.4** 中，您可以使用以下清单源的清单源 **source_vars** 将清单插件配置直接提供给自动化控制器：

- **Amazon Web Services EC2**
- **Google Compute Engine**
- **Microsoft Azure Resource Manager**
- **VMware vCenter**
- **Red Hat Satellite 6**

- Red Hat Insights
- OpenStack
- Red Hat Virtualization
- Red Hat Ansible Automation Platform

为清单源新创建的配置包含默认插件配置值。如果您希望新创建的清单源与旧源的输出匹配，您必须为该源应用一组特定的配置值。为确保向后兼容，自动化控制器对每个源都使用**"templates"**将清单插件的输出强制为旧格式。

有关源及其对应模板的更多信息，请参阅支持的清单插件模板。

包含插件的 **source_vars : foo.bar.baz** 作为顶层键会被在运行时基于 **InventorySource** 源的完全限定清单插件名称替代。例如，如果为 **InventorySource** 选择了 **ec2**，则在运行时将插件设置为 **amazon.aws.aws_ec2**。

18.4. 添加新清单

添加新清单涉及以下组件：

- 为清单添加权限
- 将组添加到清单
- 添加主机
- 添加源

- 查看完成的作业

使用以下步骤创建清单：

流程

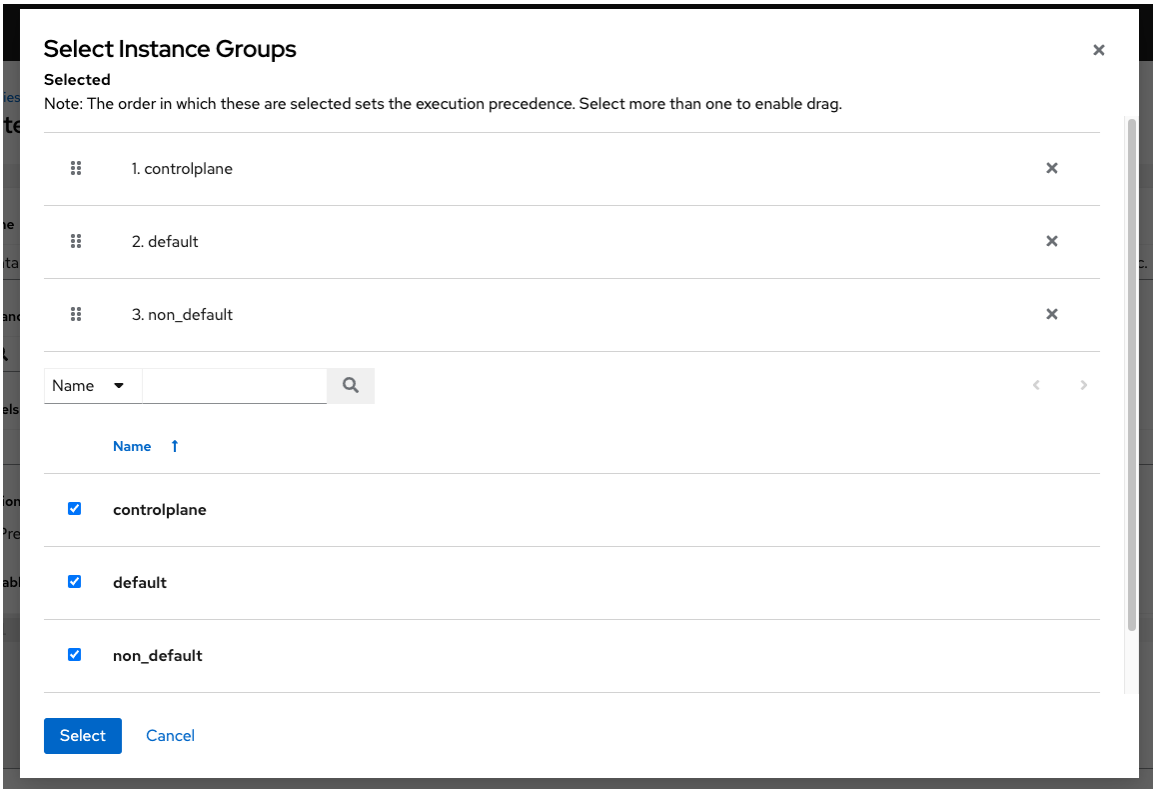
1. 在导航面板中，选择 **Resources** → **Inventories**。**Inventories** 窗口显示当前可用的清单列表。

2. 单击 **Add**，再选择要创建的清单类型。

3. 在以下字段中输入相关信息：

   - **Name** ：输入适合此清单的名称。
   - 可选： 描述 ：根据需要输入任意描述。
   - **Organization:** 必需。在可用的机构中进行选择。
   - 只适用于 智能清单：智能主机过滤器 ：点击  图标打开一个单独的窗口来过滤此清单的主机。这些选项基于您选择的机构。

     过滤器与标签类似，用于过滤包含这些名称的某些主机。因此，要填充 **Smart Host Filter** 字段，请指定包含您想要的主机的标签，而不是主机本身。在 **Search** 字段中输入标签，然后单击 **Enter**。过滤是区分大小写的。如需更多信息，请参阅 智能主机过滤器。
   - 实例组 ：点击  图标打开一个单独的窗口。选择要运行此清单的实例组或组。如果列表太长，请使用搜索来缩小选项范围。您可以选择多个实例组，并根据您想要运行的顺序对它们进行排序。



- 可选： 标签 ：选择描述此清单的标签，以便它们可用于对清单和作业进行分组和过滤。

- 仅适用于构建的清单：输入清单 ：指定要包含在此构建的清单中的源清单。 点 图标从可用清单中选择。来自输入清单的空组将复制到构建的清单中。

- 可选：（仅适用于构建的清单）： 缓存超时（秒） ：设置您希望缓存插件数据超时的时间长度。

- 仅适用于构建的清单：详细控制 **Ansible** 生成的输出级别，因为 **playbook** 执行与构建的清单关联的清单源相关的级别。 从 **Normal** 到各种 **Verbose** 或 **Debug** 设置中选择详细程度。这仅显示在**"details"**报告视图中。

  - 详细日志记录包括所有命令的输出。

  - 调试日志记录非常详细，包括对某些支持实例有用的 **SSH** 操作信息。大多数用户都不需要查看调试模式输出。

-

只适用于构建的清单： 限制 与构建的清单关联的清单源返回的主机数量。您可以将组名称粘贴到 **limit** 字段中，以仅包含该组中的主机。有关更多信息，请退出 **Source vars** 设置。

- 仅适用于标准清单： 选项 ：检查 **Prevent Instance Group Fallback** 选项，仅启用 **Instance Groups** 字段中列出的实例组来执行该作业。如果取消选中，则执行池中的所有可用实例将根据控制自动化控制器 *管理指南中的* 作业运行的 层次结构中所描述的层次结构使用。点
  ?
  图标了解更多信息。

注意

通过 **API** 为智能清单设置 **prevent_instance_group_fallback** 选项。

- 变量 （构建的清单 的源变量）：
  - 要应用到此清单中的所有主机 的变量 变量定义和值。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。
  - 构建的清单 的源变量会创建组，特别是在数据的 **groups** 键下。它接受 **Jinja2** 模板语法，呈现每个主机，进行 **true** 或 **false** 评估，如果结果为 **true**，则包括组中的主机（来自条目的密钥）。这特别有用，因为您可以将组名称粘贴到 **limit** 字段中，以仅包含该组中的主机。请参阅 智能主机过滤器中的示例 **1**。

4. 点击 **Save**。

保存新清单后，您可以继续配置权限、组、主机、源和查看已完成的作业（如果适用于相关清单类型）。

18.4.1. 为清单添加权限

使用以下步骤为清单添加权限：

流程

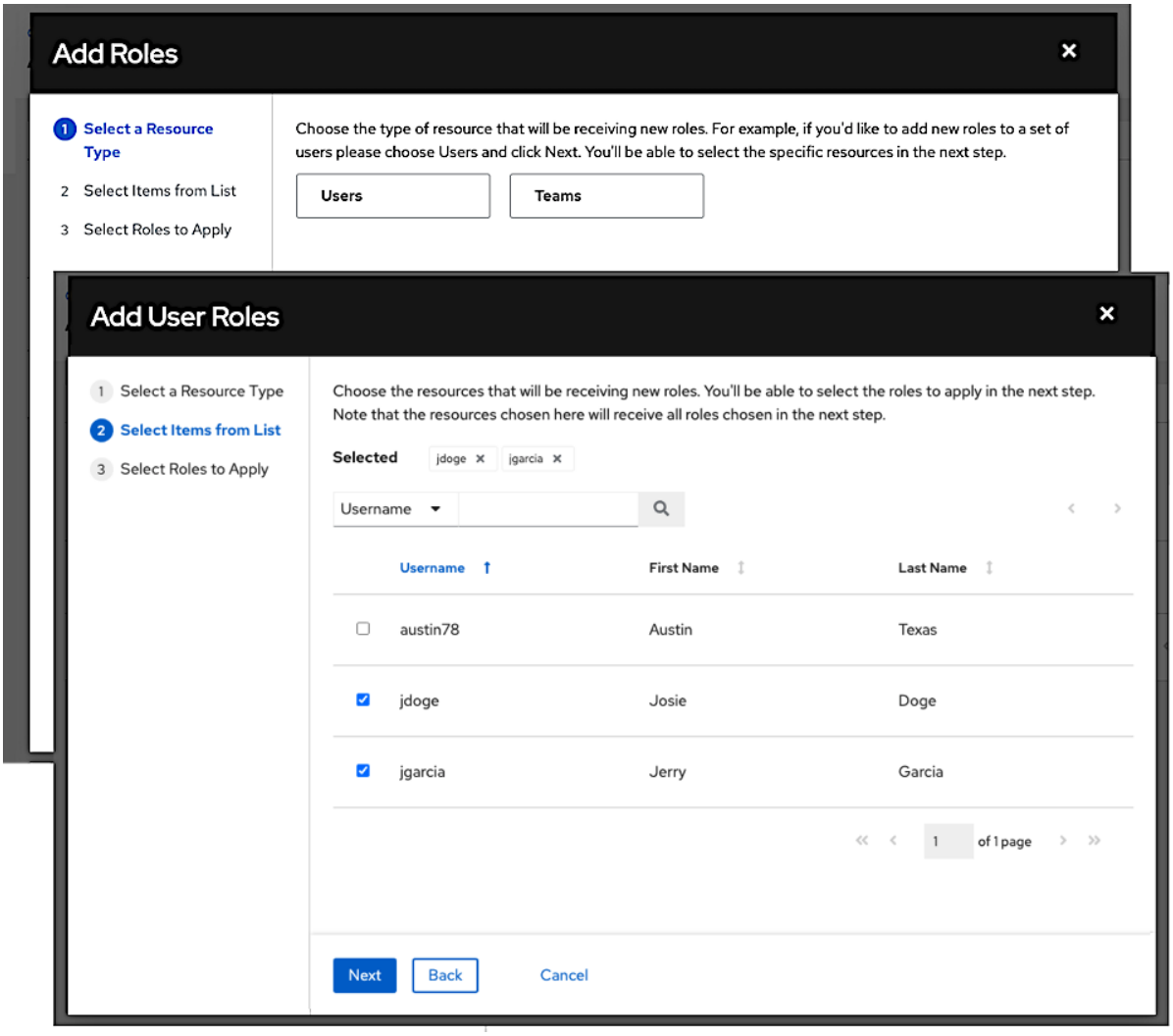
1. 在导航面板中，选择 **Resources** → **Inventories**。

2. 选择一个模板，并在 **Access** 选项卡中点 **Add**。

3. 选择要添加的用户或团队，然后点 **Next**。

4. 选择名称旁边的复选框，从列表中选择一个或多个用户或团队作为成员。

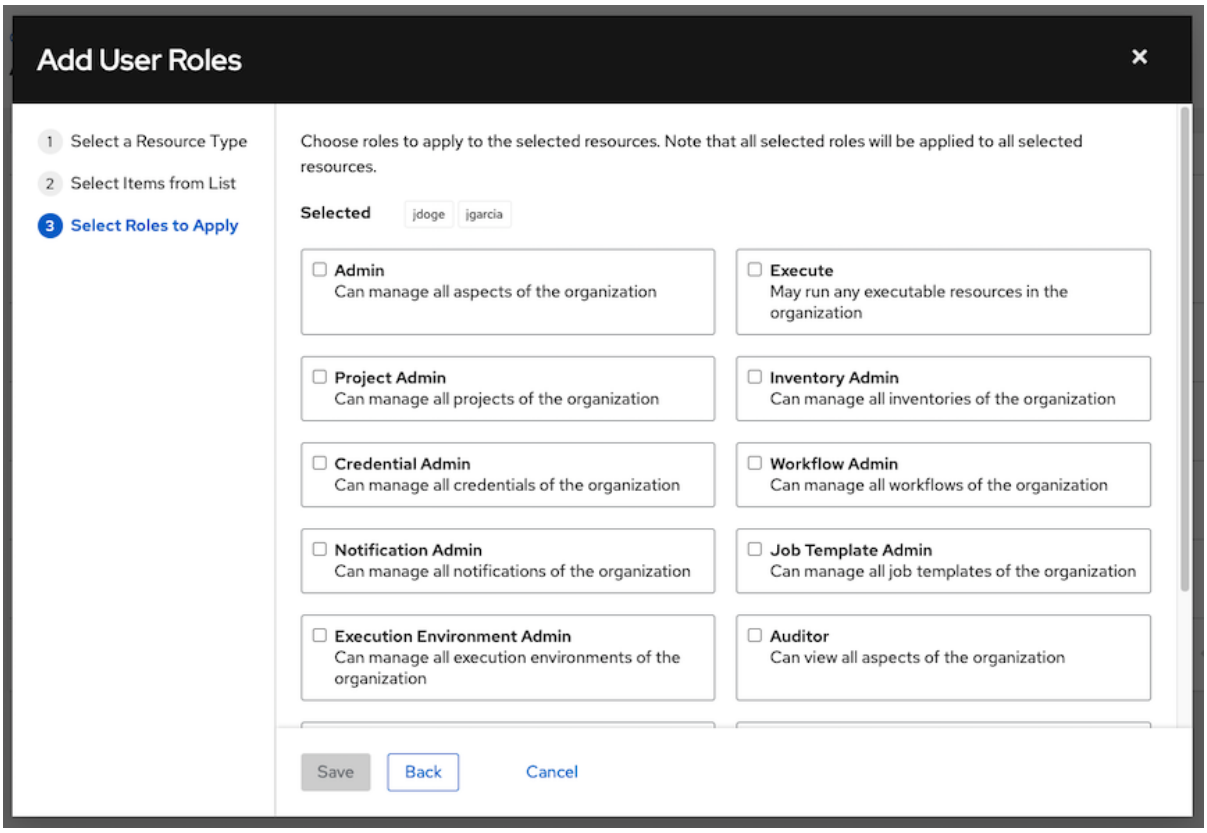
5. 点击 **Next**。



在这个示例中，选择了两个用户来添加。

6.

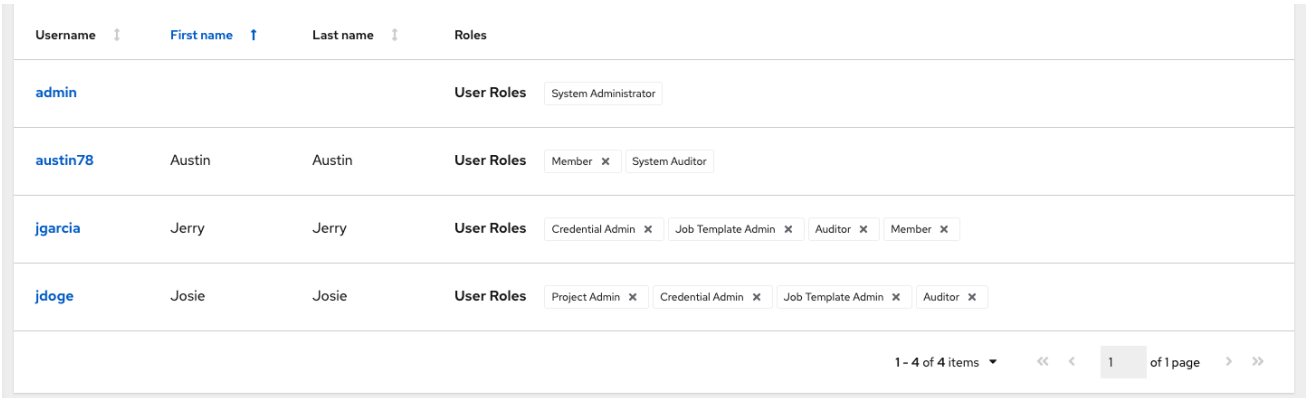
选择您希望所选用户或团队具有的角色。向下滚动以获得完整的角色列表。不同的资源有不同的可用选项。



7.

点 **Save** 将角色应用到所选用户或团队，并将它们添加为成员。

关闭 **Add Users or Teams** 窗口，以显示为每个用户和团队分配的更新角色。



删除权限

- 要删除特定用户的角色，请点其资源旁的
✖
图标。

这会启动一个确认窗口，要求您确认解除关联。

18.4.2. 将组添加到清单

清单被分成不同的组，这些组可以包含主机和其他组。组仅适用于标准清单，不能直接通过智能清单进行配置。您可以通过用于标准清单的主机关联现有组。

以下操作可用于标准清单：

- 创建新组
- 创建新主机
- 在所选清单上运行命令
- 编辑清单属性
- 查看组和主机的活动流
- 获取构建清单的帮助

注意

清单源不与组关联。生成的组为顶级组，仍然可以具有子组。所有这些生成的组都可以有主机。

使用以下步骤为清单创建新组：

流程

1. 选择您要添加组的清单名称。
2. 在 **Inventory Details** 页面中，选择 **Groups** 选项卡。
3. 单击 **Add** 以打开 **Create Group** 窗口。
4. 输入相关详情：
   - 名称 ：必需
   - 可选： 描述 ：根据需要输入描述。
   - 变量 ：输入要应用到此组中的所有主机的定义和值。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。
5. 点击 **Save**。
6. 将组添加到模板后，会显示 **Group** 详情页面。

18.4.2.1. 在组内添加组

使用以下步骤在组中添加组：

流程

1. 将组添加到模板后，会显示 **Group** 详情页面。
2.

2. 选择 **Related Groups** 选项卡。

3. 点**Add**。

4. 选择是添加您的配置中已存在的组还是创建新组。

5. 如果创建新组，请在必填和可选字段中输入相关详情：

- **Name** （必需）：
- 可选： 描述 ：根据需要输入描述。
- 变量 ：输入要应用到此组中的所有主机的定义和值。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。

6. 点击 **Save**。

7. **Create Group** 窗口关闭，在为之创建了组组关联的组列表中，新创建的组会显示为一个条目。

如果您选择添加现有组，可用组会出现在单独的选择窗口中。

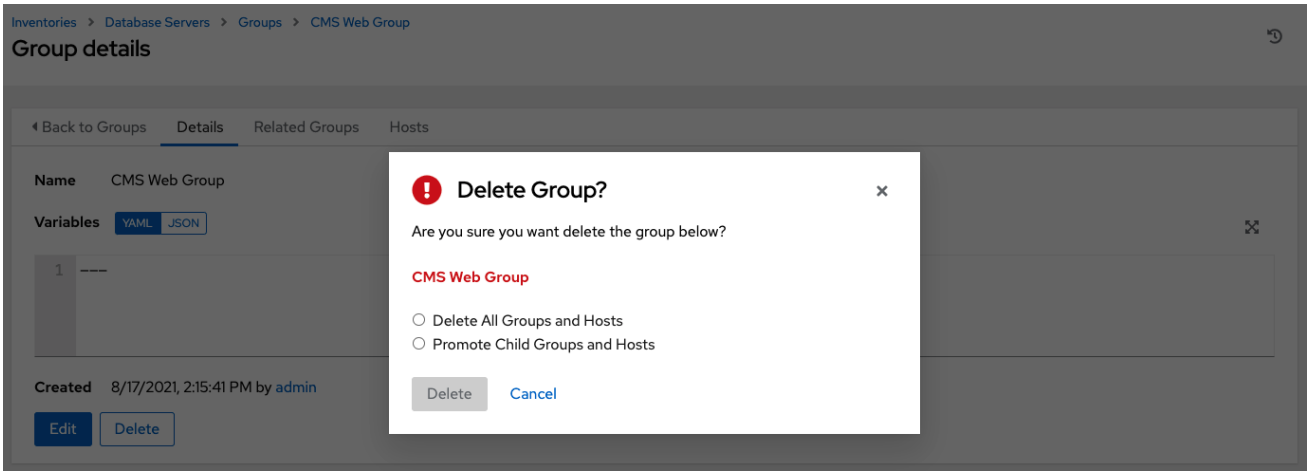
选择了一个组后，它将会显示在与组关联的组列表中。

- 要在子组下配置附加组和主机，请点击组列表中的子组名称，然后重复本节中列出的步骤。

18.4.2.2. 查看或编辑清单组

groups 列表视图会显示所有清单组，或者您可以将其过滤为仅显示根组。如果清单组不是另一个组的子集，则清单组被视为根组。

您可以删除子组而无需考虑依赖项，因为自动化控制器会查找依赖项，如子组或主机。如果存在，会显示一个确认窗口，供您选择是删除根组及其所有子组和主机，还是提升子组，以便它们及其主机成为顶层清单组。



18.4.3. 将主机添加到清单

您可以为清单以及组和组内的组配置主机。

使用以下步骤添加主机：

流程

1. 选择您要添加组的清单名称。
2. 在 **Inventory Details** 页面中，选择 **Hosts** 选项卡。
3. 点**Add**。
4. 选择是添加您的配置中已存在的主机还是创建新主机。
5. 如果创建新主机，请将切换设置为 **On**，以在运行作业时包含此主机。

6. 输入相关详情：

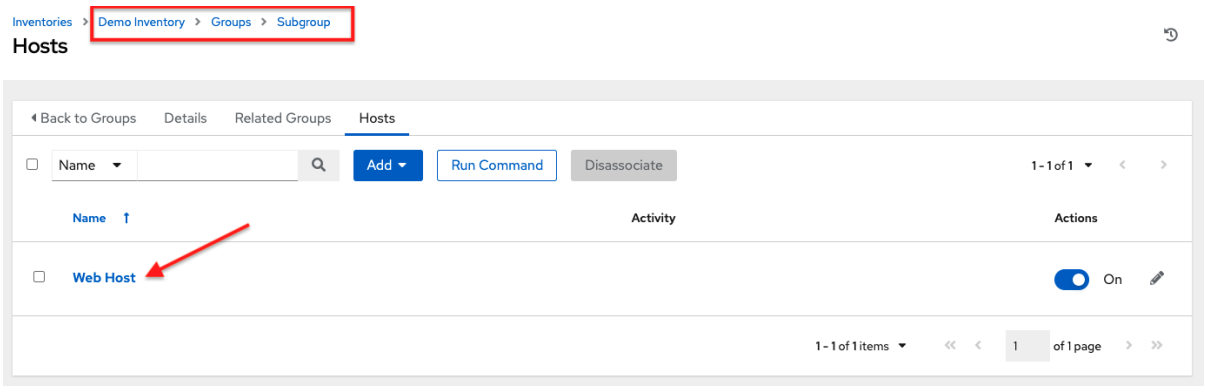
- 主机名(必需) ：
- 可选： 描述 ：根据需要输入描述。
- 变量 ：输入要应用到此组中所有主机的定义和值，如下例所示：

```
{
 ansible_user : <username to ssh into>
 ansible_ssh_pass : <password for the username>
 ansible_become_pass: <password for becoming the root>
}
```

使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。

7. 点击 **Save**。

8. **Create Host** 窗口关闭，新建的主机显示在为其为其创建的组关联的主机列表中。



如果您选择添加现有主机，可用主机会出现在单独的选择窗口中。

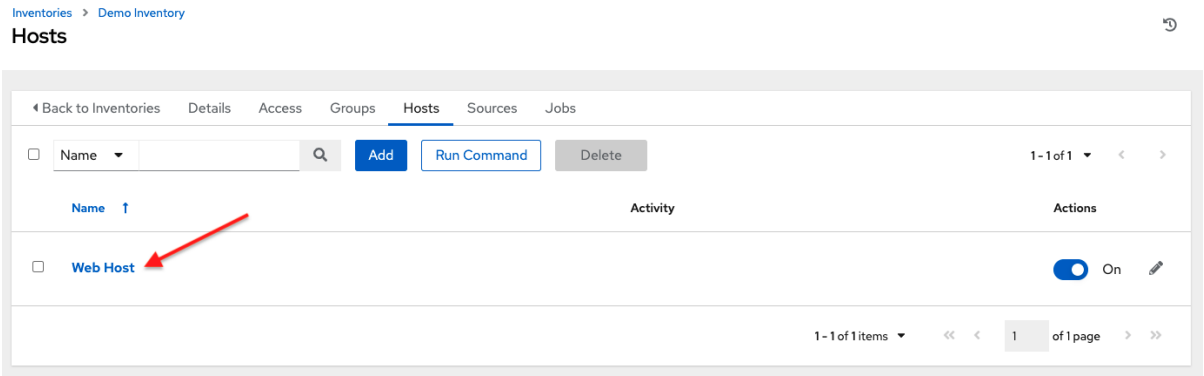
选择主机后，它将显示在与组关联的主机列表中。

9. 您可以通过选择主机并点
✖
图标从此屏幕取消关联主机。

注意

您也可以在此屏幕中运行临时命令。如需更多信息，请参阅 **[Running** 临时命令]。

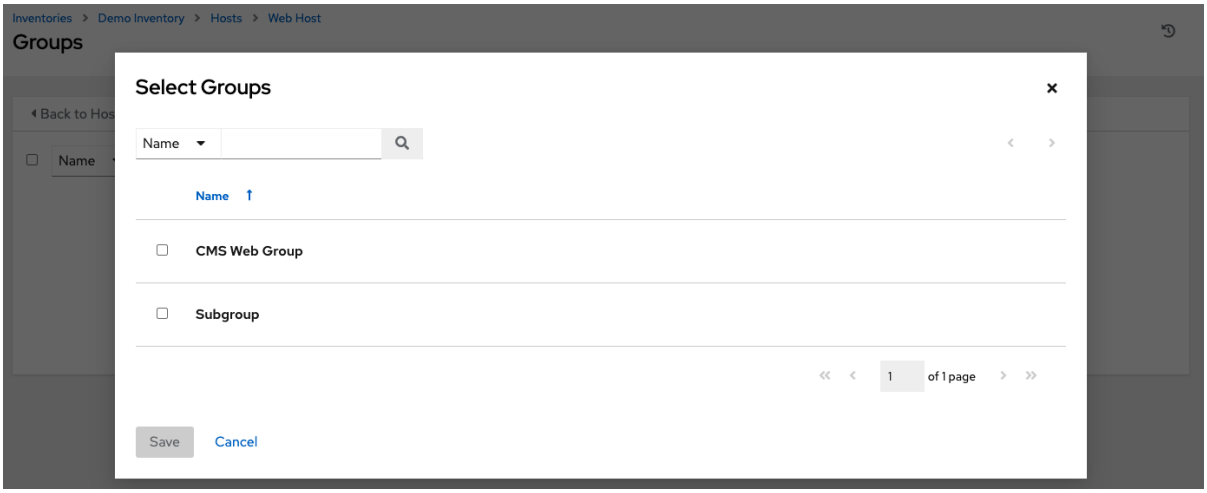
10. 要为主机配置额外的组，请点击主机列表中的主机名称。



这将打开所选主机的 **Details** 选项卡。

11. 选择 **Groups** 选项卡为主机配置组。

12. 点 **Add** 将主机与现有组关联。可用的组会出现在单独的选择窗口中。

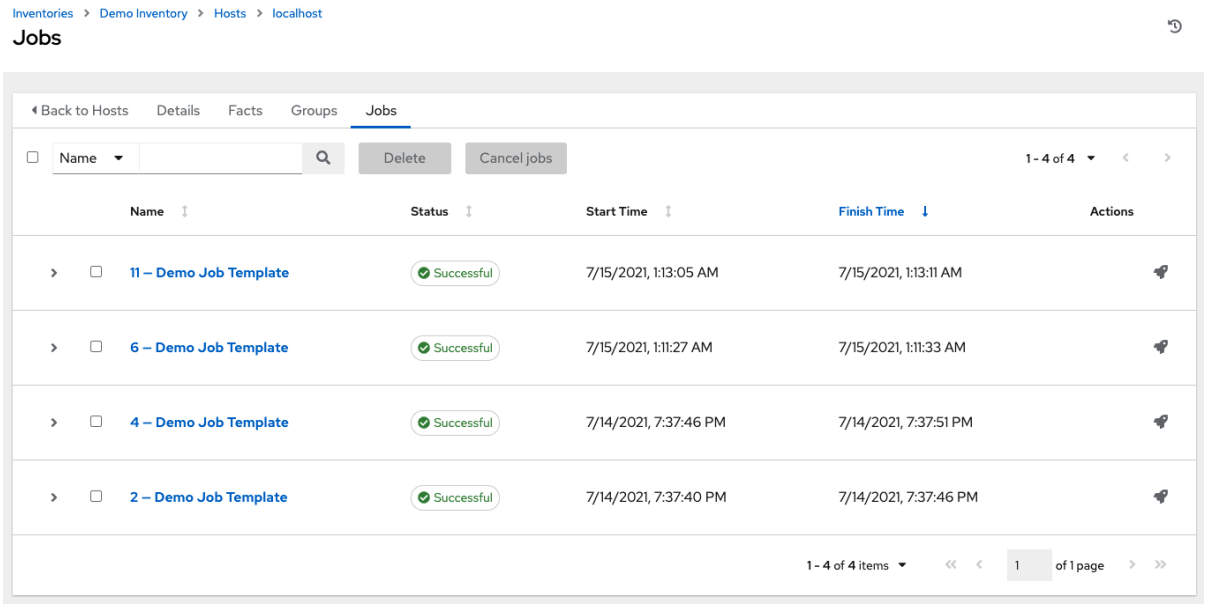


13. 选择要与主机关联的组，然后单击 **Save**。

    关联了一个组时，它将显示在与主机关联的组列表中。

14. 如果使用主机运行作业，您可以在主机的 **Completed Jobs** 选项卡中查看这些作业的详情。

15. 点 **Expanded** 查看每个作业的详情。



注意

您可以使用 **API** 中新添加的端点 **/api/v2/bulk/host_create** 来批量创建主机。此端点接受 **JSON**，您可以指定目标清单和要添加到清单的主机列表。这些主机在清单中必须是唯一的。添加所有主机，或者返回一个错误，指示操作无法完成的原因。使用 **OPTIONS** 请求返回相关模式。

如需更多信息，请参阅 *自动化控制器 **API** 指南中的 **Bulk** 端点*。

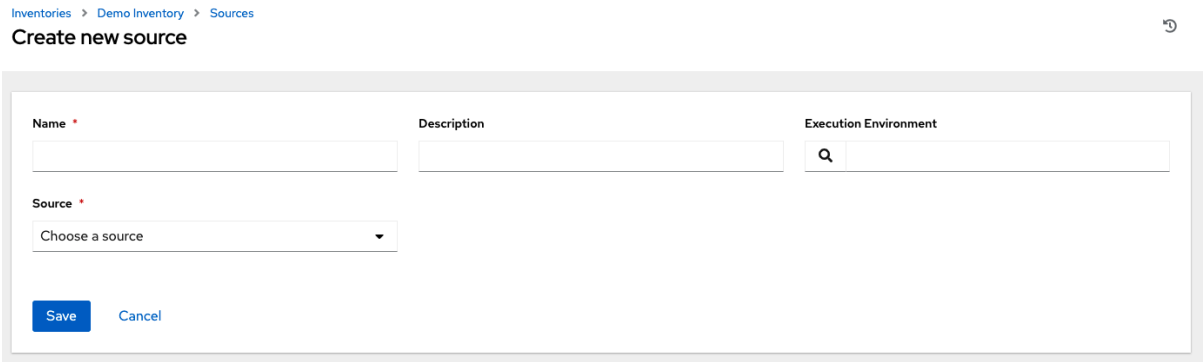
18.4.4. 添加源

清单源不与组关联。生成的组为顶级组，仍然可以具有子组。所有这些生成的组都可以有主机。在清单中添加源只适用于标准清单。智能清单从与其关联的标准清单中继承其源。

使用以下步骤为清单配置源：

流程

1. 选择您要添加源的清单名称。
2. 在 **Inventory Details** 页面中，选择 **Sources** 选项卡。
3. 点**Add**。这将打开 **Create Source** 窗口。



4. 输入相关详情：

- **Name** （必需）：
- 可选： 描述 ：根据需要输入描述。
- 可选： 点
  
  图标或输入您要运行清单导入的执行环境名称。有关构建执行环境的更多信息，请参阅执行环境。
- **Source:** 为您的清单选择一个源。有关源的更多信息，并提供适当的信息，请参阅 清单源。

5. 完成所选清单源的 信息后，您可以选择指定其他常用参数，如详细程度、主机过滤器和变量。

6. 使用 **Verbosity** 菜单选择任何清单源更新作业上的输出级别。

7. 使用 **Host Filter** 字段指定仅导入自动化控制器中的匹配主机名。

8. 在 **Enabled Variable** 字段中，指定自动化控制器从主机变量的字典中检索启用的状态。您可以使用点表示法指定为 **'foo.bar'** 指定启用的变量，在这种情况下，查找嵌套字典等同于：**from_dict.get ('foo', {}).get ('bar', default)**。

9. 如果您在 **Enabled Variable** 字段中指定主机变量字典，您可以提供一个值以便在导入时启用。例如，对于以下主机变量中的 **enabled_var='status.power_state'** 和 **'enabled_value='powered_on'**，主机被标记为 **enabled** ：

```
{
"status": {
"power_state": "powered_on",
"created": "2020-08-04T18:13:04+00:00",
"healthy": true
},
"name": "foobar",
"ip_address": "192.168.2.1"
}
```

如果 **power_state** 是 **powered_on** 以外的任何值，则主机在导入到自动化控制器时被禁用。如果没有找到密钥，则主机会被启用。

10. 所有云清单源都有以下更新选项：

- 覆盖 ：如果选中，以前存在于外部源上的但现已被删除的任何主机和组都将从自动化控制器清单中删除。未由清单源管理的主机和组将提升到下一个手动创建的组，如果没有手动创建的组来提升它们，则它们将保留在清单的**"all"**默认组中。

  如果没有选中，外部源上没有的本地子主机和组不会受到清单更新过程的影响。

- **overwrite Variables:** 如果选中，子组和主机的所有变量都将被删除，并替换为外部源上的变量。

  如果没有选中，就会执行合并，将本地变量与外部源上的变量合并。

- 启动时更新 ：每次使用此清单运行作业时，请在执行作业任务前从所选源中刷新清单。

  为了避免生成作业的速度比清单可以同步的速度快，请选择此选项可让您为以前的缓存清单同步配置缓存超时。

  Update on Launch 设置指的是项目和清单的依赖项系统，它不特别排除两个作业同时运行。

  如果指定了缓存超时，则创建第二个作业的依赖项，并使用第一个作业生成的项目和清单更新。

  然后，两个作业都会等待该项目或清单更新完成，然后继续操作。如果它们是不同的作业模板，则可以同时启动并运行它们（如果系统有能力）。如果要将自动化控制器的置备回调功能与动态清单源搭配使用，则必须为清单组 设置在启动时更新。

  如果您同步了使用设置了 **Update On Launch** 的项目的清单源，则项目可能会在清单更新开始前自动更新（根据缓存超时规则）。

  您可以创建一个作业模板，该模板使用来自模板使用的同一项目中的清单。在这种情况下，项目更新，然后清单更新（如果还没有进行更新，或者缓存超时还没有过期）。

11. 检查您的条目和选择。这可让您配置其他详情，如调度和通知。

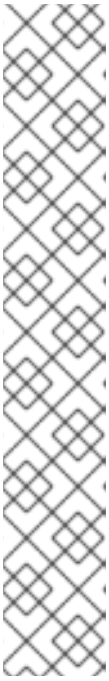
12. 要配置与此清单源关联的调度，请点 **Schedules** 选项卡：

- 如果已经设置了调度，请检查、编辑、启用或禁用您的调度首选项。
- 如果还没有设置调度，请参阅 **Schedules**。

18.4.5. 为源配置通知

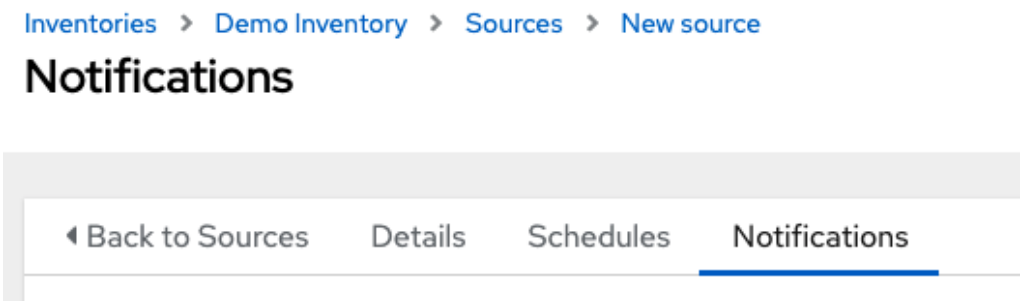
使用以下步骤为源配置通知：

1. 在 **Inventory Details** 页面中，选择 **Notifications** 选项卡。



注意

只有在保存新创建的源时，才会显示 **Notifications** 选项卡。



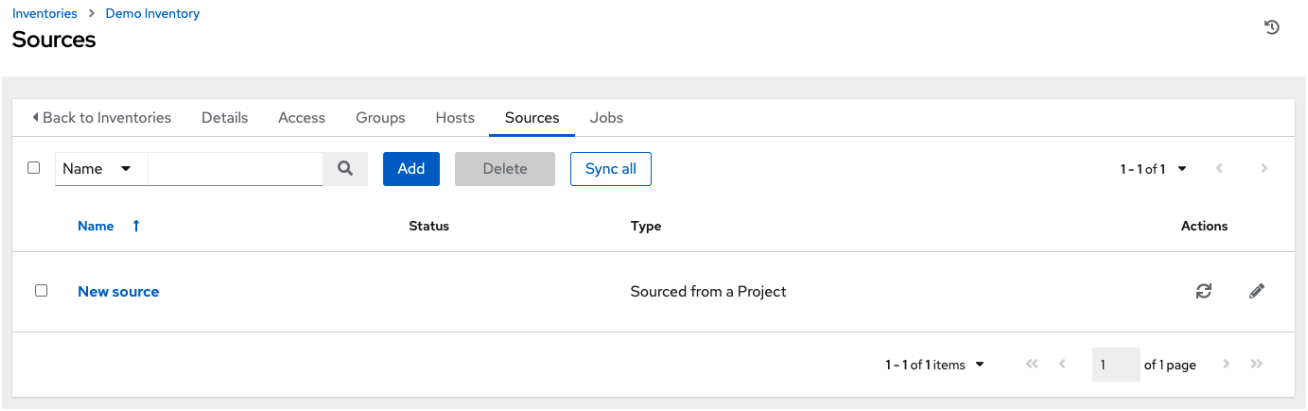
2. 如果已经设置了通知，请使用切换按钮启用或禁用要与特定源搭配使用的通知。如需更多信息，请参阅 启用和禁用通知。

3. 如果还没有设置通知，请参阅 通知 以了解更多信息。

4. 检查您的条目和选择。

5. 点击 **Save**。

定义源时，它将显示在与清单关联的源列表中。在 **Sources** 选项卡中，您可以对单个源执行同步，或者一次性同步它们。您还可以执行其他操作，如调度同步过程，以及编辑或删除源。



18.4.5.1. 清单源

选择与可以作为主机输入的清单类型匹配的源：

- 来自项目的源
- **Amazon Web Services EC2**
- **Google Compute Engine**
- **Microsoft Azure Resource Manager**
- **VMware vCenter**

- Red Hat Satellite 6
- Red Hat Insights
- OpenStack
- Red Hat Virtualization
- Red Hat Ansible Automation Platform

18.4.5.1.1. 来自项目的源

源于一个项目的清单意味着它使用它所关联的项目中的 **SCM** 类型。例如，如果项目的源来自 **GitHub**，则清单将使用相同的源。

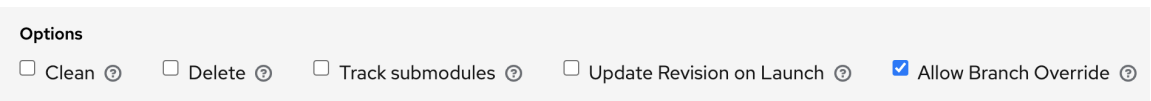
使用以下步骤配置项目的清单：

流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **Sourced from a Project**。
2. **Create Source** 窗口会展开更多字段。输入以下详情：

- 可选： **Source Control Branch/Tag/Commit** ：从源控制(**Git** 或 **Subversion**)输入 **SCM** 分支、标签、提交散列、任意 **refs** 或修订号（如果适用）。

只有在源项目检查 **Allow Branch Override** 选项时，才会显示此字段。如需更多信息，请参阅 **SCM** 类型 **- Git** 和 **Subversion**。

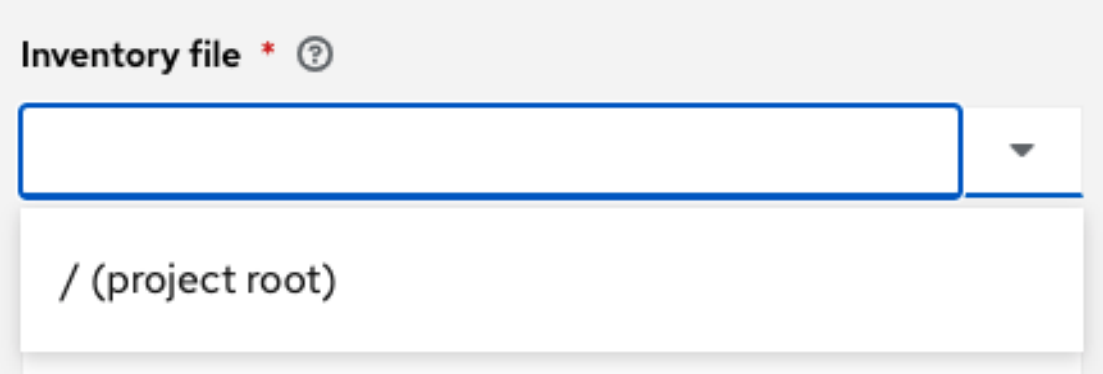


除非在下一字段中还提供了自定义 **refspec**，否则某些提交散列和 **refs** 可能不可用。如果留空，则默认为 **HEAD**，这是此项目最后一次签出的 **Branch/Tag/Commit**。

- 可选： **Credential:** 指定用于此源的凭证。

- **Project** （必需）：与默认项目预先填充，否则指定此清单将用作其源的项目。点 图标从项目列表中选择。如果列表太长，请使用搜索来缩小选项范围。

- 清单文件 （必需）：选择与源项目关联的清单文件。如果尚未填充，您可以在菜单中的文本字段中输入它，以过滤额外的文件类型。除了平面文件清单外，您还可以指向目录或清单脚本。



3. 可选： 您可以指定详细程度、主机过滤器、启用的变量/值和更新选项，如 添加源 中所述。

4. 可选： 要传递给自定义清单脚本，您可以在 **Environment Variables** 字段中设置环境变量。您还可以将清单脚本放在源控制中，然后从项目运行它。如需更多信息，请参阅 *自动化控制器管理指南中的* 清单文件导入。

注意

如果您要从 **SCM** 执行自定义清单脚本，请确保在上游源控制中为脚本设置执行位 **(chmod +x)**。

如果没有，自动化控制器会在执行时抛出 **[Errno 13] Permission denied** 错误。

18.4.5.1.2. Amazon Web Services EC2

使用以下步骤配置 **AWS EC2-sourced** 清单，

流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **Amazon EC2**。
2. **Create Source** 窗口会展开其他字段。输入以下详情：
   - 可选： 凭证 ：从现有 **AWS** 凭证中选择（如需更多信息，请参阅 凭证）。

     如果自动化控制器在带有分配的 **IAM** 角色的 **EC2** 实例上运行，则可以省略凭证，并且改为使用实例元数据中的安全凭证。有关使用 **IAM** 角色的更多信息，请参阅 **IAM_Roles_for_Amazon_EC2_documentation_at_Amazon**。
3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。
4. 使用 **Source Variables** 字段覆盖 **aws_ec2** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **aws** 清单插件文档。

注意

如果您只使用 **include_filters**，**AWS** 插件始终返回所有主机。要正确使用它，或 上的第一个条件必须为过滤器，然后在 **include_ filters** 列表中构建 **OR** 条件的其余部分。

18.4.5.1.3. Google Compute Engine

使用以下步骤配置 **Google-sourced** 清单。

流程

1. 在 **Create new source** 页面中，从 **Source** 中选择 **Google Compute Engine**。
2.

Create Source 窗口会展开所需的 **Credential** 字段。从现有的 **GCE** 凭证中选择。如需更多信息，请参阅 凭证。

3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。

4. 使用 **Source Variables** 字段覆盖 **gcp_compute** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **gcp_compute** 清单插件文档。

18.4.5.1.4. Microsoft Azure 资源管理器

使用以下步骤配置 **Azure Resource Manager-sourced** 清单：

流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **Microsoft Azure Resource Manager**。

2. **Create Source** 窗口会展开所需的 **Credential** 字段。从现有的 **Azure** 凭证中选择。如需更多信息，请参阅 凭证。

3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。

4. 使用 **Source Variables** 字段覆盖 **azure_rm** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **azure_rm** 清单插件文档。

18.4.5.1.5. VMware vCenter

使用以下步骤配置 **VMWare** 源的清单。

流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **VMware vCenter**。

2. **Create Source** 窗口会展开所需的 **Credential** 字段。从现有的 **VMware** 凭证中选择。如需更多信息，请参阅 凭证。

3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。

4. 使用 **Source Variables** 字段覆盖 **vmware_inventory** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **vmware_inventory** 清单插件。

注意

VMware 属性已从小写改为 **camelCase**。自动化控制器为顶级键提供别名，但不再使用嵌套属性中的小写键。如需有效且支持的属性列表，请参阅在 **VMware** 动态插件中使用虚拟机属性。

18.4.5.1.6. Red Hat Satellite 6

使用以下步骤配置 **Red Hat Satellite** 源于 **Red Hat Satellite** 的清单。

流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **Red Hat Satellite**。

2. **Create Source** 窗口会展开所需的 **Credential** 字段。从现有的 **Satellite** 凭证中选择。如需更多信息，请参阅 凭证。

3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。

4. 使用 **Source Variables** 字段指定 **foreman** 清单源使用的参数。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **Ansible** 文档中的 **Foreman** 清单源。

如果您遇到自动化控制器清单没有 **Satellite** 的"相关组"的问题，您可能需要在清单源中定义这些变量。如需更多信息，请参阅 **Red Hat Satellite 6**。

如果您在 同步清单时看到消息"no foreman.id"变量，请参阅红帽客户门户网站中的解决方案：https://access.redhat.com/solutions/5826451。https://access.redhat.com/solutions/5826451请确定使用您的客户凭证登录以访问完整文章。

18.4.5.1.7. Red Hat Insights

使用以下步骤配置 **Red Hat Insights-sourced** 清单。

流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **Red Hat Insights**。
2. **Create Source** 窗口会展开所需的 **Credential** 字段。从现有的 **GCE** 凭证中选择。如需更多信息，请参阅 凭证。
3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。
4. 使用 **Source Variables** 字段覆盖 **gcp_compute** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **Insights** 清单插件。

18.4.5.1.8. OpenStack

使用以下步骤配置 **OpenStack** 源的清单。

流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **Openstack**。
2. **Create Source** 窗口会展开所需的 **Credential** 字段。从现有的 **GCE** 凭证中选择。如需更多信息，请参阅 凭证。
3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。

4. 使用 **Source Variables** 字段覆盖 **gcp_compute** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **openstack** 清单插件。

18.4.5.1.9. 红帽虚拟化

使用以下步骤配置 **Red Hat virtualization-sourced** 清单。

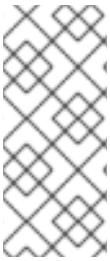
流程

1. 在 **Create new source** 页面中，从 **Source** 列表中选择 **Red Hat Virtualization**。

2. **Create Source** 窗口会展开所需的 **Credential** 字段。从现有的 **GCE** 凭证中选择。如需更多信息，请参阅 凭证。

3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。

4. 使用 **Source Variables** 字段覆盖 **gcp_compute** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 **ovirt** 清单插件



注意

Red Hat Virtualization (ovirt)清单源请求默认是安全的。要更改此默认设置，请在 **source_variables** 中将键 **ovirt_insecure** 设置为 **true**，这只在 **/api/v2/inventory_sources/N/** 端点的清单源的 **API** 详情中可用。

18.4.5.1.10. Red Hat Ansible Automation Platform

使用以下步骤配置自动化控制器提供的清单。

流程

1. 在 **Create new source** 页面中，从 ***Source** 列表中选择 **Red Hat Ansible Automation Platform**。

2. **Create Source** 窗口会展开所需的 **Credential** 字段。从现有的 **GCE** 凭证中选择。如需更多信息，请参阅 凭证。

3. 可选： 您可以指定详细程度、主机过滤器、启用的变量或值以及更新选项，如 添加源 中所述。

4. 使用 **Source Variables** 字段覆盖 **gcp_compute** 清单插件使用的变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。有关这些变量的更多信息，请参阅 控制器清单插件。这需要您的红帽客户登录。

18.4.5.2. 导出旧的清单脚本

尽管删除了自定义清单脚本 **API**，但脚本仍然保存在数据库中。本节中描述的命令可让您以适合您随后检查源控制的格式从数据库中恢复脚本。

使用以下命令：

```
$ awx-manage export_custom_scripts --filename=my_scripts.tar

Dump of old custom inventory scripts at my_scripts.tar
```

使用输出：

```
$ mkdir my_scripts
$ tar -xf my_scripts.tar -C my_scripts
```

脚本的名称具有以下形式： **< *pk*>_ &*lt*;name>**。这是用于项目文件夹的命名方案。

```
$ ls my_scripts
10inventory_script_rawhook _19 _30inventory_script_listenhospital _11inventory_script_upperorder
_1inventory_script_commercialinternet45 _4inventory_script_whitestring
_12inventory_script_eastplant _22inventory_script_pinexchange
_5inventory_script_literaturepossession _13inventory_script_governmentculture
_23inventory_script_brainluck _6inventory_script_opportunitytelephone
_14inventory_script_bottomguess _25inventory_script_buyerleague _7inventory_script_letjury
_15inventory_script_wallisland _26inventory_script_lifesport _8random_inventory_script
16inventory_script_wallisland _27inventory_script_exchangesomewhere _9random_inventory_script
_17nventory_script_bidstory          _28inventory_script_boxchild _18p
_29__inventory_script_wearstress
```

每个文件包含一个脚本。脚本可以是 **bash/python/ruby/more**，因此不会包含扩展。它们都是直接执行的。执行脚本可转储清单数据。

```
$ ./my_scripts/11__inventory_script_upperorder
{"group\ud801\udcb0\uc20e\u7b0e\ud81c\udfeb\ub12b\ub4d0\u9ac6\ud81e\udf07\u6ff9\uc17b":
{"hosts":
["host_\ud821\udcad\u68b6\u7a51\u93b4\u69cf\uc3c2\ud81f\uddbe\ud820\udc92\u3143\u62c7",
"host_\u6057\u3985\u1f60\ufefb\u1b22\ubd2d\ua90c\ud81a\udc69\u1344\u9d15",
"host_\u78a0\ud820\udef3\u925e\u69da\ua549\ud80c\ude7e\ud81e\udc91\ud808\uddd1\u57d6\ud801\
ude57",
"host_\ud83a\udc2d\ud7f7\ua18a\u779a\ud800\udf8b\u7903\ud820\udead\u4154\ud808\ude15\u9711",

"host_\u18a1\u9d6f\u08ac\u74c2\u54e2\u740e\u5f02\ud81d\uddee\ufbd6\u4506"], "vars":
{"ansible_host": "127.0.0.1", "ansible_connection":
"local"}}}
```

您可以使用 **ansible-inventory** 验证功能。这提供了相同的数据，但重新格式化。

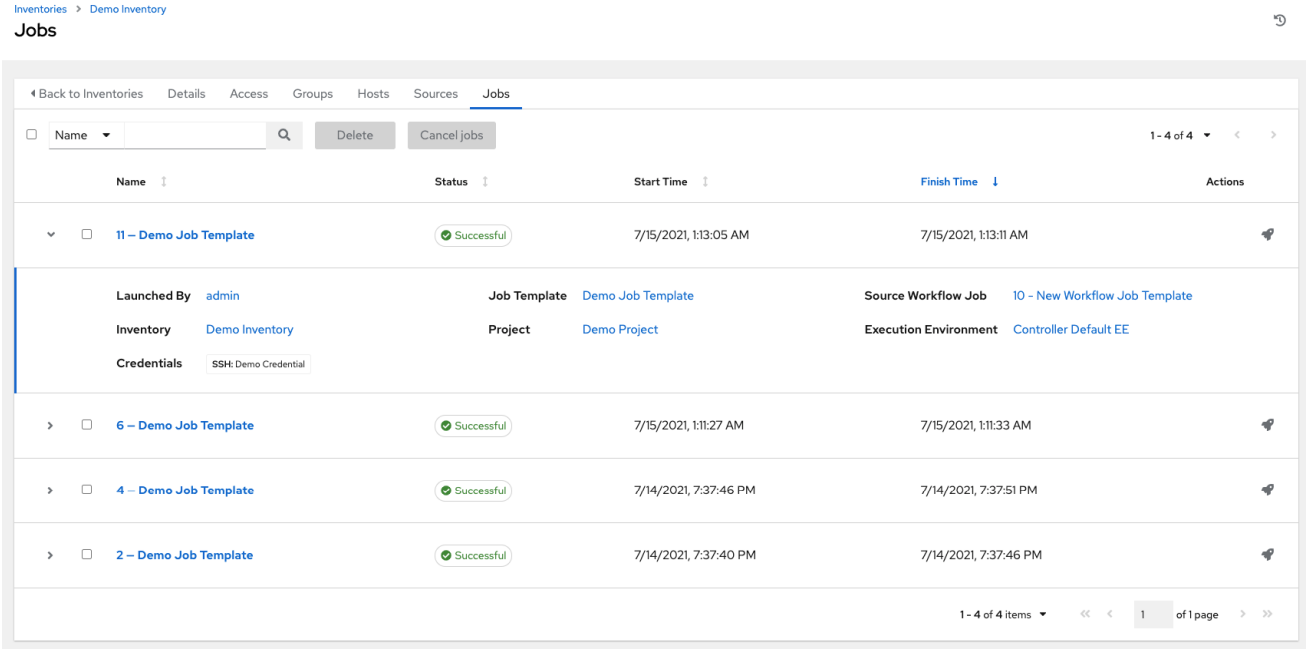
```
$ ansible-inventory -i ./my_scripts/_11__inventory_script_upperorder --list --export
```

在前面的示例中，您可以 **cd** 到 **my_scripts**，然后发出 **git init** 命令，添加您想要的脚本，将其推送到源控制，然后在用户界面中创建 **SCM** 清单源。

有关同步或使用 自定义清单脚本的更多信息，请参阅 *自动化控制器管理指南中的* 清单文件导入。

18.5. 查看完成的作业

如果使用清单来运行作业，您可以在清单的 **Completed Jobs** 选项卡中查看这些作业的详情，点 **Expanded** 查看每个作业的详情。



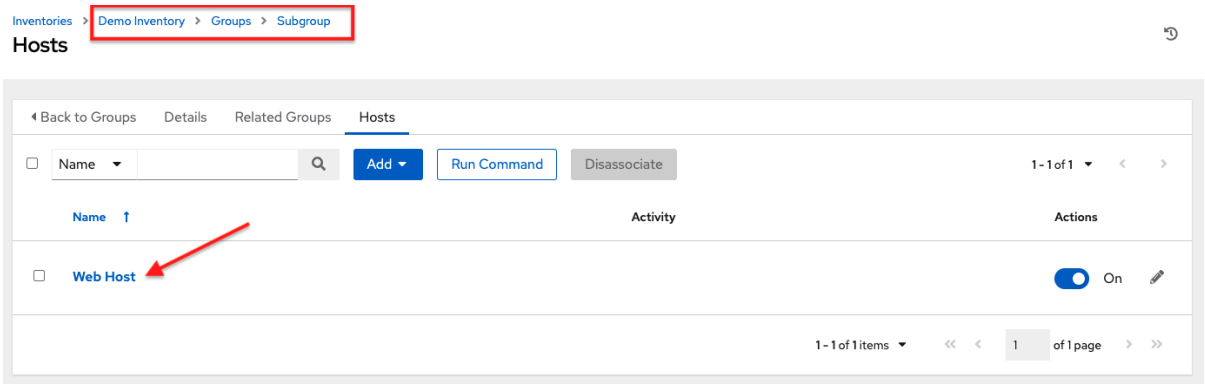
18.6. 运行临时命令

临时命令 使用 **Ansible** 执行快速命令，使用 **/usr/bin/ansible**，而不是编配语言，即 **/usr/bin/ansible-playbook**。一个临时命令的示例可能会在您的基础架构中重新引导 **50** 个机器。您可以编写 **Playbook** 来完成任何操作。**playbook** 也可以将许多其他操作组合在一起。

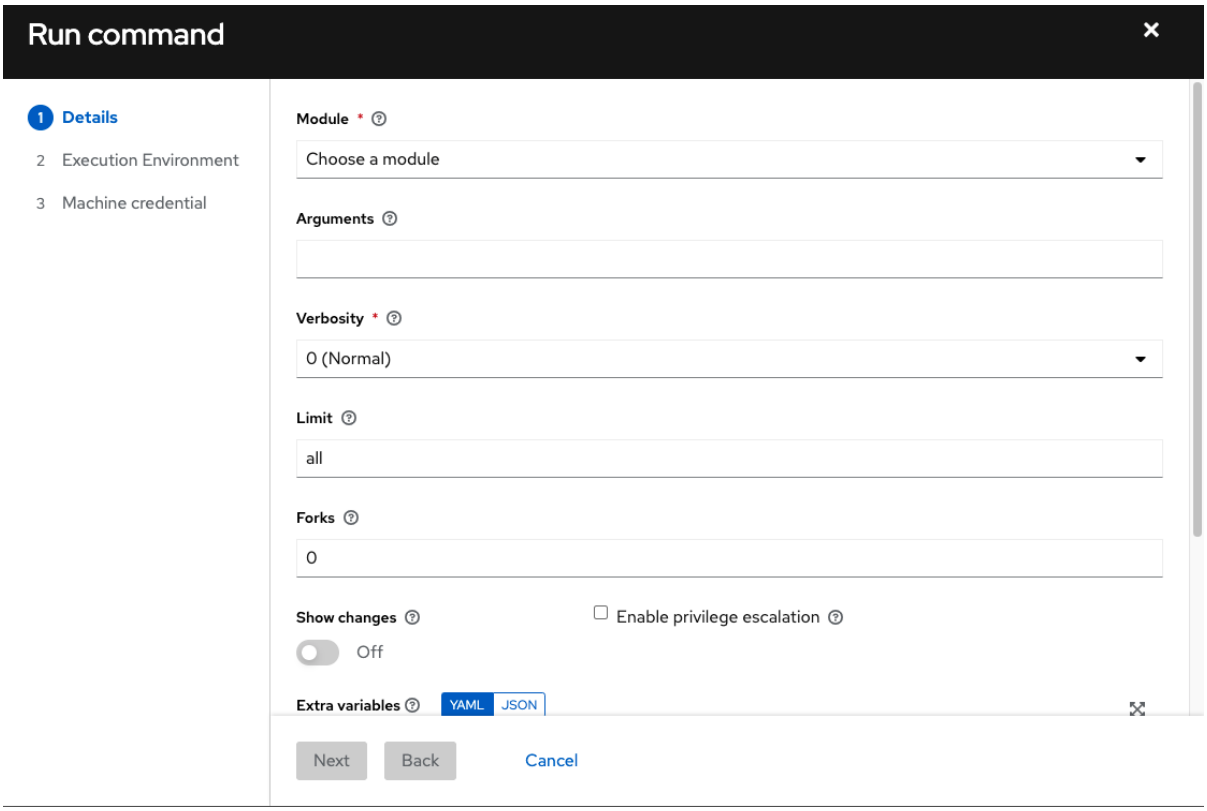
使用以下步骤运行临时命令：

流程

1. 从主机或组列表中选择清单源。该清单源可以是单个组或主机，也可以是特定的多个主机，也可以是特定的多个组。



2. 单击 **Run Command**。这将打开 **Run** 命令窗口。



3. 输入以下信息：

- 模块 ：选择支持运行命令的模块之一。

命令	apt_repository	mount	win_service
shell	apt_rpm	ping	win_updates
yum	service	selinux	win_group
apt	group	setup	win_user
apt_key	user	win_ping	win_user

- 参数 ：提供要与您选择的模块搭配使用的参数。

- 限制 ：输入用于清单中目标主机的限制。要以清单中的所有主机为目标，请输入 **all** 或

*，或者将该字段留空。在点启动按钮之前，这会自动填充上一次视图中选择的任何信息。

- **Machine Credential:** 选择访问远程主机时要使用的凭据，以运行该命令。选择包含 **Ansible** 需要登录远程主机所需的用户名和 **SSH** 密钥或密码的凭据。

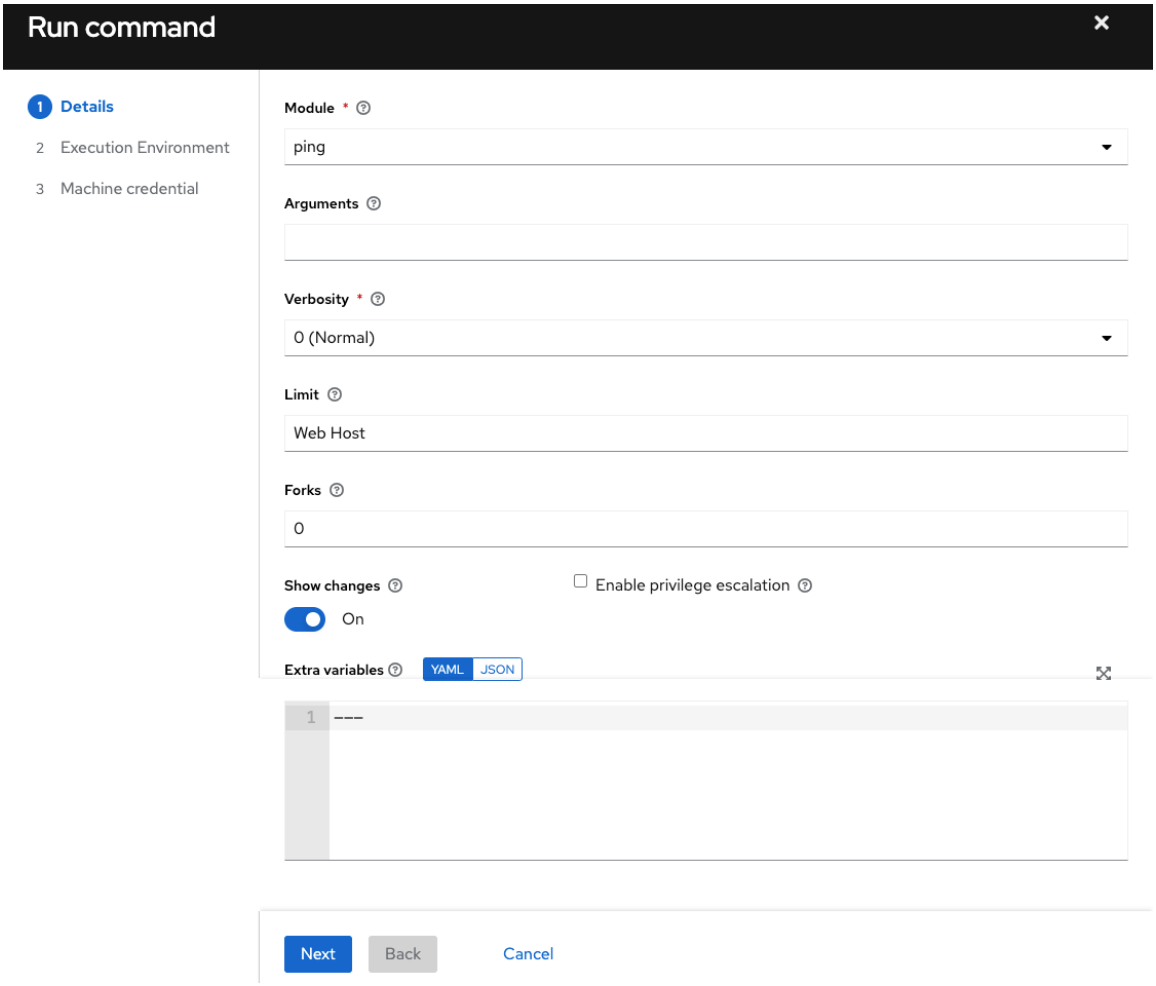
- 详细程度 ：为标准输出选择详细程度。

- **fork** ：如果需要，请选择执行命令时使用的并行或同步进程数量。

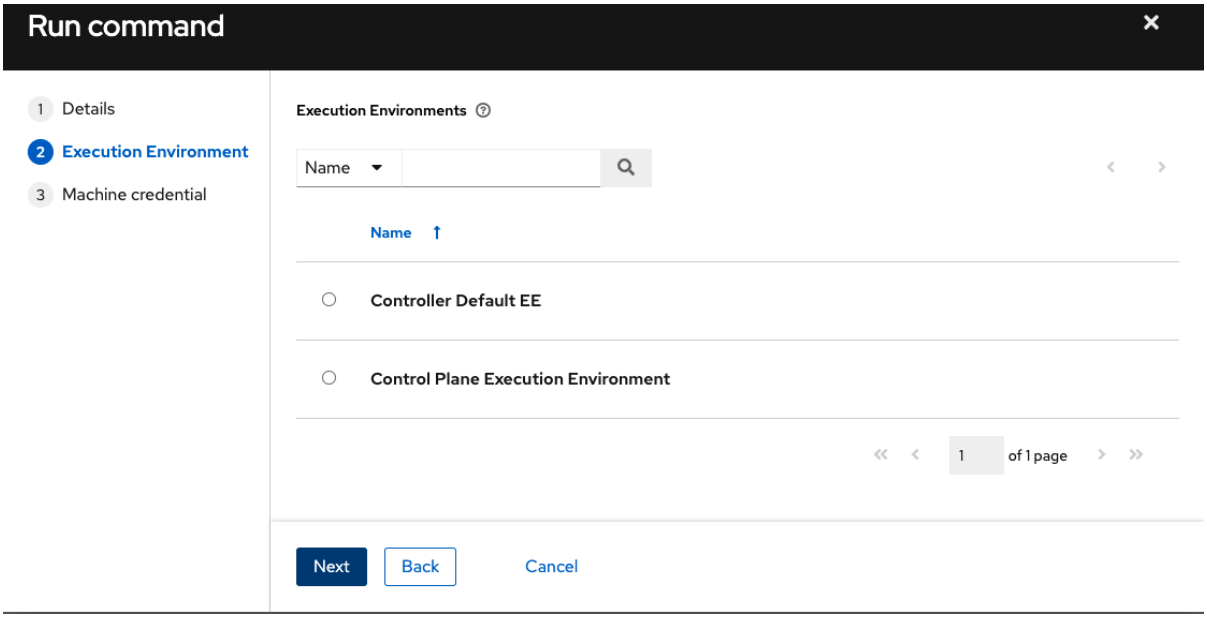
- 显示更改 ：选择此项可在标准输出中显示 **Ansible** 更改。默认值为 **OFF**。

- 启用权限升级 ：如果启用，则使用管理员权限运行 **playbook**。这等同于将**-- become** 选项传递给 **ansible** 命令。

- 额外变量 ：提供在运行此清单时要应用的额外命令行变量。使用 **JSON** 或 **YAML** 语法输入变量。使用单选按钮在两者之间切换。

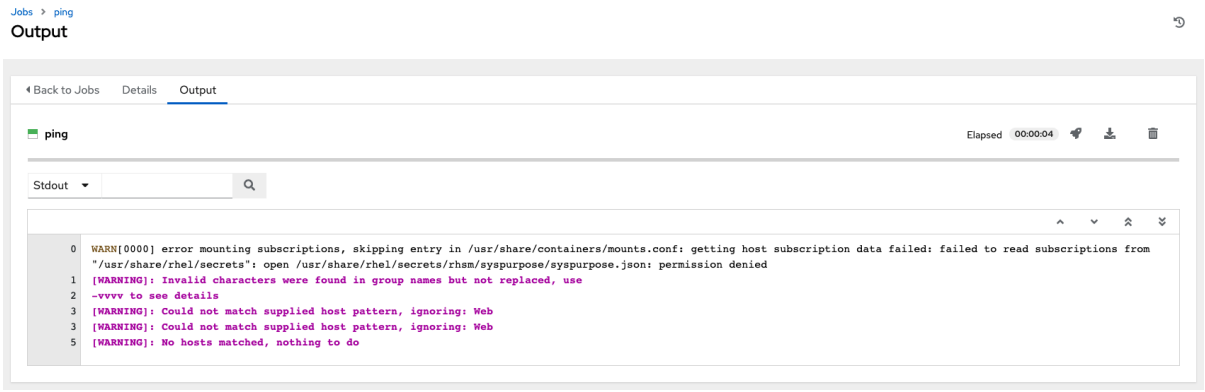


4. 单击 **Next** 以选择要对其运行临时命令的执行环境。



5. 点 **Next** 以选择要使用的凭证。

6. 点 **Launch**。结果会显示在模块作业窗口的 **Output** 选项卡中。



第 19 章 支持的清单插件模板

升级到 4.x 后，现有配置将迁移到生成向后兼容的清单输出的新格式。使用以下模板来帮助将清单迁移到新风格的清单插件输出。

- Amazon Web Services EC2
- Google Compute Engine
- Microsoft Azure Resource Manager
- VMware vCenter
- Red Hat Satellite 6
- OpenStack
- Red Hat Virtualization
- Red Hat Ansible Automation Platform

19.1. AMAZON WEB SERVICES EC2

```
compose:
  ansible_host: public_ip_address
  ec2_account_id: owner_id
  ec2_ami_launch_index: ami_launch_index | string
  ec2_architecture: architecture
  ec2_block_devices: dict(block_device_mappings | map(attribute='device_name') | list |
zip(block_device_mappings | map(attribute='ebs.volume_id') | list))
  ec2_client_token: client_token
  ec2_dns_name: public_dns_name
  ec2_ebs_optimized: ebs_optimized
  ec2_eventsSet: events | default("")
  ec2_group_name: placement.group_name
  ec2_hypervisor: hypervisor
  ec2_id: instance_id
```

```
  ec2_image_id: image_id
  ec2_instance_profile: iam_instance_profile | default("")
  ec2_instance_type: instance_type
  ec2_ip_address: public_ip_address
  ec2_kernel: kernel_id | default("")
  ec2_key_name: key_name
  ec2_launch_time: launch_time | regex_replace(" ", "T") | regex_replace("(\+)(\d\d):(\d)(\d)$",
".\g<2>\g<3>Z")
  ec2_monitored: monitoring.state in ['enabled', 'pending']
  ec2_monitoring_state: monitoring.state
  ec2_persistent: persistent | default(false)
  ec2_placement: placement.availability_zone
  ec2_platform: platform | default("")
  ec2_private_dns_name: private_dns_name
  ec2_private_ip_address: private_ip_address
  ec2_public_dns_name: public_dns_name
  ec2_ramdisk: ramdisk_id | default("")
  ec2_reason: state_transition_reason
  ec2_region: placement.region
  ec2_requester_id: requester_id | default("")
  ec2_root_device_name: root_device_name
  ec2_root_device_type: root_device_type
  ec2_security_group_ids: security_groups | map(attribute='group_id') | list |  join(',')
  ec2_security_group_names: security_groups | map(attribute='group_name') | list |  join(',')
  ec2_sourceDestCheck: source_dest_check | default(false) | lower | string
  ec2_spot_instance_request_id: spot_instance_request_id | default("")
  ec2_state: state.name
  ec2_state_code: state.code
  ec2_state_reason: state_reason.message if state_reason is defined else ""
  ec2_subnet_id: subnet_id | default("")
  ec2_tag_Name: tags.Name
  ec2_virtualization_type: virtualization_type
  ec2_vpc_id: vpc_id | default("")
filters:
  instance-state-name:
  - running
groups:
  ec2: true
hostnames:
  - network-interface.addresses.association.public-ip
  - dns-name
  - private-dns-name
keyed_groups:
  - key: image_id | regex_replace("[^A-Za-z0-9\_]", "_")
    parent_group: images
    prefix: ''
    separator: ''
  - key: placement.availability_zone
    parent_group: zones
    prefix: ''
    separator: ''
  - key: ec2_account_id | regex_replace("[^A-Za-z0-9\_]", "_")
    parent_group: accounts
    prefix: ''
    separator: ''
  - key: ec2_state | regex_replace("[^A-Za-z0-9\_]", "_")
```

```
    parent_group: instance_states
    prefix: instance_state
  - key: platform | default("undefined") | regex_replace("[^A-Za-z0-9\_]", "_")
    parent_group: platforms
    prefix: platform
  - key: instance_type | regex_replace("[^A-Za-z0-9\_]", "_")
    parent_group: types
    prefix: type
  - key: key_name | regex_replace("[^A-Za-z0-9\_]", "_")
    parent_group: keys
    prefix: key
  - key: placement.region
    parent_group: regions
    prefix: ''
    separator: ''
  - key: security_groups | map(attribute="group_name") | map("regex_replace", "[^A-Za-z0-
9\_]", "_") | list
    parent_group: security_groups
    prefix: security_group
  - key: dict(tags.keys() | map("regex_replace", "[^A-Za-z0-9\_]", "_") | list | zip(tags.values()
      | map("regex_replace", "[^A-Za-z0-9\_]", "_") | list))
    parent_group: tags
    prefix: tag
  - key: tags.keys() | map("regex_replace", "[^A-Za-z0-9\_]", "_") | list
    parent_group: tags
    prefix: tag
  - key: vpc_id | regex_replace("[^A-Za-z0-9\_]", "_")
    parent_group: vpcs
    prefix: vpc_id
  - key: placement.availability_zone
    parent_group: '{{ placement.region }}'
    prefix: ''
    separator: ''
plugin: amazon.aws.aws_ec2
use_contrib_script_compatible_sanitization: true
```

19.2. GOOGLE COMPUTE ENGINE

```
auth_kind: serviceaccount
compose:
  ansible_ssh_host: networkInterfaces[0].accessConfigs[0].natIP |
default(networkInterfaces[0].networkIP)
  gce_description: description if description else None
  gce_id: id
  gce_image: image
  gce_machine_type: machineType
  gce_metadata: metadata.get("items", []) | items2dict(key_name="key", value_name="value")
  gce_name: name
  gce_network: networkInterfaces[0].network.name
  gce_private_ip: networkInterfaces[0].networkIP
  gce_public_ip: networkInterfaces[0].accessConfigs[0].natIP | default(None)
  gce_status: status
  gce_subnetwork: networkInterfaces[0].subnetwork.name
  gce_tags: tags.get("items", [])
  gce_zone: zone
```

```
hostnames:
- name
- public_ip
- private_ip
keyed_groups:
- key: gce_subnetwork
  prefix: network
- key: gce_private_ip
  prefix: ''
  separator: ''
- key: gce_public_ip
  prefix: ''
  separator: ''
- key: machineType
  prefix: ''
  separator: ''
- key: zone
  prefix: ''
  separator: ''
- key: gce_tags
  prefix: tag
- key: status | lower
  prefix: status
- key: image
  prefix: ''
  separator: ''
plugin: google.cloud.gcp_compute
retrieve_image_info: true
use_contrib_script_compatible_sanitization: true
```

19.3. MICROSOFT AZURE RESOURCE MANAGER

```
conditional_groups:
  azure: true
default_host_filters: []
fail_on_template_errors: false
hostvar_expressions:
  computer_name: name
  private_ip: private_ipv4_addresses[0] if private_ipv4_addresses else None
  provisioning_state: provisioning_state | title
  public_ip: public_ipv4_addresses[0] if public_ipv4_addresses else None
  public_ip_id: public_ip_id if public_ip_id is defined else None
  public_ip_name: public_ip_name if public_ip_name is defined else None
  tags: tags if tags else None
  type: resource_type
keyed_groups:
- key: location
  prefix: ''
  separator: ''
- key: tags.keys() | list if tags else []
  prefix: ''
  separator: ''
- key: security_group
  prefix: ''
  separator: ''
```

```
- key: resource_group
  prefix: ''
  separator: ''
- key: os_disk.operating_system_type
  prefix: ''
  separator: ''
- key: dict(tags.keys() | map("regex_replace", "^(.*)$", "\1_") | list | zip(tags.values() | list)) if
tags else []
  prefix: ''
  separator: ''
plain_host_names: true
plugin: azure.azcollection.azure_rm
use_contrib_script_compatible_sanitization: true
```

19.4. VMWARE VCENTER

```
compose:
  ansible_host: guest.ipAddress
  ansible_ssh_host: guest.ipAddress
  ansible_uuid: 99999999 | random | to_uuid
  availablefield: availableField
  configissue: configIssue
  configstatus: configStatus
  customvalue: customValue
  effectiverole: effectiveRole
  guestheartbeatstatus: guestHeartbeatStatus
  layoutex: layoutEx
  overallstatus: overallStatus
  parentvapp: parentVApp
  recenttask: recentTask
  resourcepool: resourcePool
  rootsnapshot: rootSnapshot
  triggeredalarmstate: triggeredAlarmState
filters:
- runtime.powerState == "poweredOn"
keyed_groups:
- key: config.guestId
  prefix: ''
  separator: ''
- key: '"templates" if config.template else "guests"'
  prefix: ''
  separator: ''
plugin: community.vmware.vmware_vm_inventory
properties:
- availableField
- configIssue
- configStatus
- customValue
- datastore
- effectiveRole
- guestHeartbeatStatus
- layout
- layoutEx
- name
- network
```

```
- overallStatus
- parentVApp
- permission
- recentTask
- resourcePool
- rootSnapshot
- snapshot
- triggeredAlarmState
- value
- capability
- config
- guest
- runtime
- storage
- summary
strict: false
with_nested_properties: true
```

19.5. RED HAT SATELLITE 6

```
group_prefix: foreman_
keyed_groups:
- key: foreman['environment_name'] | lower | regex_replace(' ', '') | regex_replace('[^A-Za-z0-
9_]', '_') | regex_replace('none', '')
  prefix: foreman_environment_
  separator: ''
- key: foreman['location_name'] | lower | regex_replace(' ', '') | regex_replace('[^A-Za-z0-9_]',
'_')
  prefix: foreman_location_
  separator: ''
- key: foreman['organization_name'] | lower | regex_replace(' ', '') | regex_replace('[^A-Za-z0-
9_]', '_')
  prefix: foreman_organization_
  separator: ''
- key: foreman['content_facet_attributes']['lifecycle_environment_name'] | lower |
regex_replace(' ', '') | regex_replace('[^A-Za-z0-9_]', '_')
  prefix: foreman_lifecycle_environment_
  separator: ''
- key: foreman['content_facet_attributes']['content_view_name'] | lower | regex_replace(' ', '') |
regex_replace('[^A-Za-z0-9_]', '_')
  prefix: foreman_content_view_
  separator: ''
legacy_hostvars: true
plugin: theforeman.foreman.foreman
validate_certs: false
want_facts: true
want_hostcollections: false
want_params: true
```

19.6. OPENSTACK

```
expand_hostvars: true
fail_on_errors: true
```

```
inventory_hostname: uuid
plugin: openstack.cloud.openstack
```

19.7. RED HAT VIRTUALIZATION

```
compose:
  ansible_host: (devices.values() | list)[0][0] if devices else None
keyed_groups:
- key: cluster
  prefix: cluster
  separator: _
- key: status
  prefix: status
  separator: _
- key: tags
  prefix: tag
  separator: _
ovirt_hostname_preference:
- name
- fqdn
ovirt_insecure: false
plugin: ovirt.ovirt.ovirt
```

19.8. RED HAT ANSIBLE AUTOMATION PLATFORM

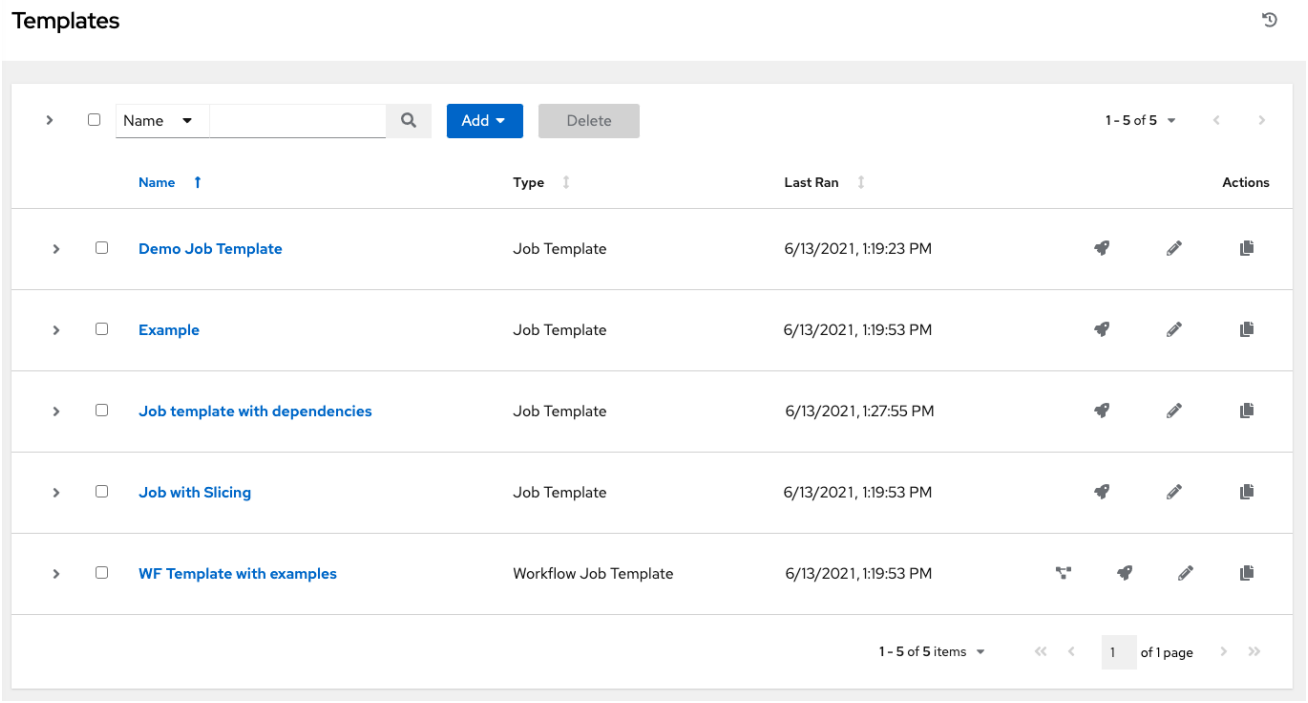
```
include_metadata: true
inventory_id: <inventory_id or url_quoted_named_url>
plugin: awx.awx.tower
validate_certs: <true or false>
```

第 20 章 作业模板

作业模板是用于运行 **Ansible** 作业的定义和一组参数。作业模板可用于多次运行同一作业。它们还鼓励重复使用 **Ansible playbook** 内容以及团队间的协作。

Templates 列表视图显示当前可用的作业模板。默认视图为折叠状态**(Compact)**，显示模板名称、模板类型和使用该模板运行的最后一个作业的时间戳。您可以点每个条目旁边的箭头

图标展开并查看更多信息。此列表按名称按字母顺序排序，但您可以根据其他条件排序，或者根据模板的不同字段和属性进行搜索。



在此屏幕中，您可以启动

，编辑

，并复制

工作流作业模板。

注意

作业模板可用于构建工作流模板。显示 工作流可视化工具

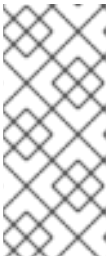
图标的模板是工作流模板。点图标允许您以图形方式构建工作流。作业模板中的多个参数允许您选择 **Prompt on Launch**，您可以在工作流级别更改，而不影响作业模板级别分配的值。具体步骤请查看 工作流可视化工具 部分。

20.1. 创建作业模板

流程

1. 在 **Templates** 列表视图中，从 **Add** 列表中选择 **Add Job template**。

2. 在以下字段中输入相关信息：



注意

如果某个字段选择了 **Prompt on launch** 复选框，则启动作业时会提示您输入该字段的值。大多数提示的值将覆盖作业模板中设置的任何值。下表中会记录例外情况。

字段	选项	启动时提示
Name	输入作业的名称。	N/A
描述	根据需要输入任意描述（可选）。	N/A
任务类型	选择作业类型： ● 运行：启动时启动 playbook，在选定的主机上运行 Ansible 任务。 ● check：执行 playbook 的"dry run"并报告将要进行的更改，而无需实际进行更改。不支持检查模式的任务将丢失，且不会报告潜在的更改。 如需有关作业类型的更多信息，请参阅 Ansible 文档中的 Playbook 部分。	是
清单（Inventory）	从可供登录的用户可用的清单中选择要用于此作业模板的清单。 系统管理员必须授予您或团队权限，才能在作业模板中使用某些清单。	是。 清单提示会在后续提示窗口中以自己的步骤的形式显示。

字段	选项	启动时提示
项目	从可供登录的用户可用的项目中选择要用于此作业模板的项目。	N/A
SCM 分支	只有在您选择了允许分支覆盖的项目时，才会显示此字段。指定要在任务运行中使用的覆盖分支。如果留空，则使用项目中的指定 SCM 分支（或提交散列或标签）。 如需更多信息，请参阅 作业分支覆盖。	是
执行环境	选择要运行此任务的容器镜像。您必须选择一个项目，然后才能选择执行环境。	是。 执行环境提示会在后续提示窗口中以自己的步骤的形式显示。
Playbook	从可用的 playbook 中选择要使用此作业模板启动的 playbook。此字段自动填充所选项目的项目基本路径中找到的 playbook 名称。另外，如果未列出 playbook，则可以输入 playbook 的名称，如您要使用该 playbook 运行的文件的名称（如 foo.yml）。如果您输入无效的文件名，模板会显示一个错误，或者导致作业失败。	N/A
凭证	选择 图标打开一个单独的窗口。 从用于此作业模板的可用选项中选择凭据。 如果列表太长，请使用下拉菜单列表根据凭证类型过滤。某些凭证类型不会被列出，因为它们不适用于某些作业模板。	• 如果选择，在启动具有默认凭证的作业模板时，如果相同类型，则提供另一个凭证会替换默认凭证。以下是这个消息的示例： **作业模板默认凭据必须替换为同一类型之一。请为以下类型选择一个凭证，以便继续： Machine。** • 或者，您可以在看到适合的情况下添加更多凭证。 • 凭证提示会在后续提示窗口中以自己的步骤的形式显示。

字段	选项	启动时提示
标签	• （可选）提供描述此作业模板的标签，如 **dev** 或 **test**。 • 使用标签对显示中的作业模板和完成的作业进行分组和过滤。 • 标签在添加到作业模板时创建。标签使用作业模板中提供的项目与单个机构关联。如果机构的成员具有编辑权限（如 admin 角色），则机构的成员可以在作业模板上创建标签。 • 保存作业模板后，标签会显示在 Expanded 视图中的 **作业模板** 概述中。 • 选择标签旁边的✖将其删除。删除标签后，它不再与该特定作业或作业模板关联，但它与引用它的任何其他作业关联。 • 在启动时，作业会继承作业模板中的标签。如果您从作业模板中删除标签，它也会从作业中删除。	• 如果选择，即使提供了默认值，也会在启动时提示您提供附加标签（如果需要）。 • 您无法删除现有标签，选择✖只删除新添加的标签，而不是现有的默认标签。
变量	• 向 playbook 传递额外的命令行变量。这是 ansible-playbook 的"-e"或"-extra-vars"命令行参数，记录在 Ansible 文档中 在运行时定义变量的 Ansible 文档中。 • 使用 YAML 或 JSON 提供键或值对。这些变量具有最大优先级值，并覆盖其他位置指定的其他变量。以下是一个值： **git_branch: production** **release_version: 1.5**	是。 如果要能够在调度中指定 **extra_vars**，您必须在作业模板中为 Variables 选择 **Prompt on launch**，或者在作业模板上启用问卷调查。那些回答的问卷调查问题将变为 **extra_vars**。
Forks	执行 playbook 时使用的并行或同步进程数量。值为零使用 Ansible 默认设置，即五个并行进程，除非在 **/etc/ansible/ansible.cfg** 中被覆盖。	是

字段	选项	启动时提示
限制	用于进一步限制受 playbook 管理或影响的主机列表的主机模式。您可以通过冒号(:)分隔多个模式。与核心 Ansible 一样： ● a:b 表示"在组 a 或 b 中" ● a:b:&c 表示"在 a 或 b 中，但必须在 c 中"。 ● 答：！b 表示"在 a 中，一定不要在 b 中" 如需更多信息，请参阅 Ansible 文档中的 Patterns：以主机和组为目标。	是 如果未选中，则作业模板针对清单中的所有节点，或者仅在 **Limit** 字段中预定义节点执行。作为工作流的一部分运行时，会使用工作流作业模板限制。
详细程度	控制 Ansible 在 playbook 执行时生成的输出级别。从 Normal 到各种 Verbose 或 Debug 设置中选择详细程度。这只会出现在 **详情** 报告视图中。详细日志记录包括所有命令的输出。调试日志记录非常详细，包括对某些支持实例有用的 SSH 操作信息。 详细程度 **5** 会导致自动化控制器在作业运行时大量阻断，这可能会延迟报告作业已完成（即使它已经完成），并可能导致浏览器标签页锁定。	是
任务分片	指定您希望此作业模板运行的分片数量。每个片段针对清单的一部分运行相同的任务。有关作业分片的更多信息，请参阅 作业分片。	是

字段	选项	启动时提示
Timeout（超时）	这可让您指定在作业被取消前可以运行的时间长度（以秒为单位）。在设置超时值时请考虑以下几点： ● 在设置中定义了一个全局超时，默认为 0，表示没有超时。 ● 作业模板上的一个负超时 (<0)是作业中的 true "no timeout"。 ● 作业模板上的超时为 0，将作业默认为全局超时（默认为没有超时）。 ● 一个正超时会设置该作业模板的超时时间。	是
显示更改	允许您查看 Ansible 任务所做的更改。	是
实例组	选择 Instance 和 Container Groups 来与此作业模板关联。如果列表太长，使用 图标缩小选项范围。作业模板实例组贡献作业调度条件，请参阅 作业运行时 行为 和控制针对规则运行的作业。系统管理员必须授予您或团队权限，才能在作业模板中使用实例组。使用容器组需要 admin 权限。	● 是。 如果选择，您将按首选顺序提供作业首选实例组。如果第一个组没有容量，则会考虑列表中的后续组，直到有容量可用为止，该组被选择来运行作业。 ● 如果您提示输入实例组，则输入的内容会替换正常的实例组层次结构，并覆盖所有机构和清单实例组。 ● 实例组提示在后续提示窗口中以自己的步骤的形式显示。
作业标签	键入 并选择 **Create** 菜单，以指定应执行 playbook 的哪些部分。有关更多信息和示例，请参阅 Ansible 文档中的 标签。	是
跳过标签	键入 并选择 **Create** 菜单，以指定要跳过的 playbook 的某些任务或部分。有关更多信息和示例，请参阅 Ansible 文档中的 标签。	是

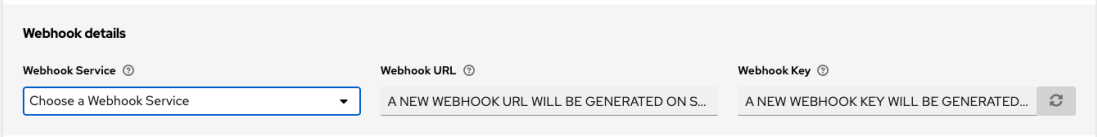
3. 如果需要，指定启动此模板的以下选项：

- **Privilege Escalation** ：如果选中，您可以使此 **playbook** 以管理员身份运行。这等同于将**-- become** 选项传递给 **ansible-playbook** 命令。

- **provisioning Callbacks** ：如果选中，您可以启用主机通过 **REST API** 调用自动化控制器，并从此作业模板启动作业。如需更多信息，请参阅 置备回调。

- 启用 **Webhook** ：如果选中，您可以打开与用于启动作业模板的预定义 **SCM** 系统 **Web** 服务进行接口的功能。**GitHub** 和 **GitLab** 是支持的 **SCM** 系统。

  - 如果您启用 **Webhook**，会显示其他字段，提示输入更多信息：



  - **Webhook Service** ：选择要从哪个服务侦听 **Webhook**。

  - **Webhook URL** ：自动填充将 **POST** 请求发送到的 **Webhook** 服务的 **URL**。

  - **Webhook Key:** 生成共享 **secret**，供 **Webhook** 服务用来签署发送到自动化控制器的有效负载。您必须在 **Webhook** 服务上的设置中对此进行配置，以便自动化控制器接受来自该服务的 **Webhook**。

  - **Webhook** 凭证 ：（可选）提供 **GitHub** 或 **GitLab** 个人访问令牌**(PAT)**作为凭证，用来向 **webhook** 服务发回状态更新。在选择它前，凭证必须存在。请参阅 凭证类型 来创建。

  - 有关设置 **Webhook** 的更多信息，请参阅使用 **Webhook**。

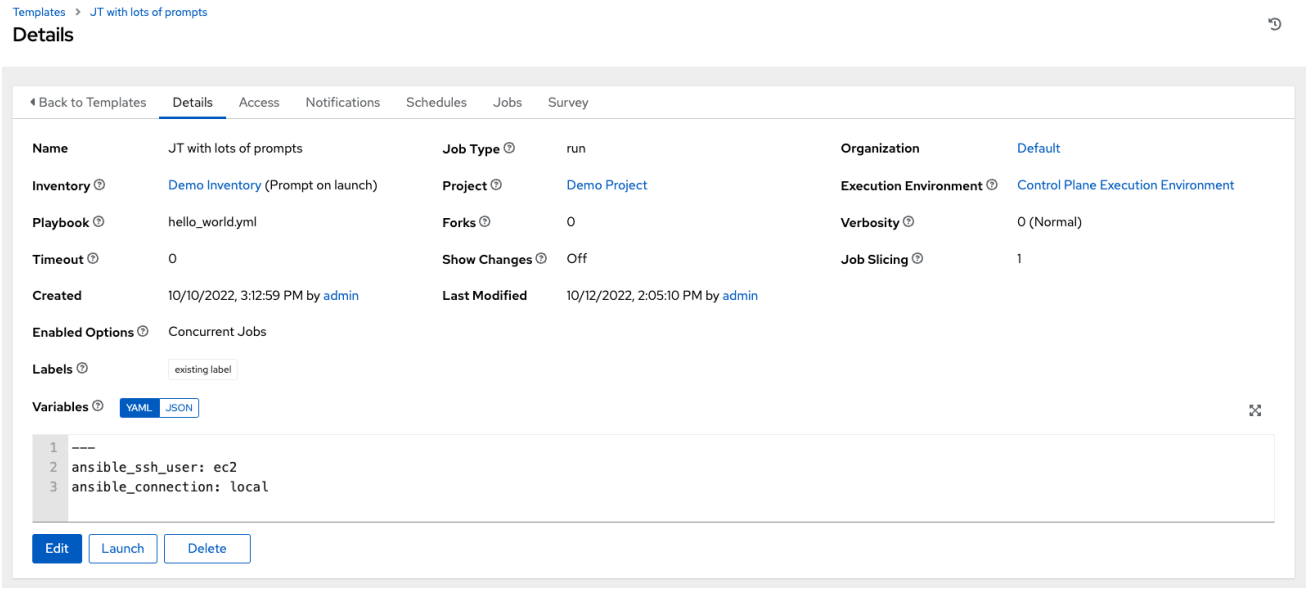
- 并发作业 ：如果选中，则允许队列中的作业如果不依赖于另一个作业，则同时运行。如果要同时运行作业分片，请选中此框。如需更多信息，请参阅自动控制器容量确定和作业影响。

- 启用事实存储 ：如果选中，自动化控制器将收集到与作业运行相关的清单中所有主机的事实。

- 防止实例组 **Fallback:** 检查此选项，仅允许 **Instance Groups** 字段中列出的实例组来运行作业。如果清除，则执行池中的所有可用实例都会根据控制 作业运行中所述的层次结构使用。

4. 当您完成作业模板详情配置后，点 **Save**。

保存模板不会退出作业模板页面，而是提前进入 作业模板详情选项卡。保存模板后，您可以点 **Launch** 来启动作业，或者点 **Edit** 添加或更改模板的属性，如权限、通知、查看完成的作业，并添加问卷调查（如果作业类型不是扫描）。在启动前，您必须首先保存模板，否则 **Launch** 仍被禁用。



验证

1. 在导航面板中，选择 **Resources** → **Templates**。

2. 验证新创建的模板是否出现在 **Templates** 列表视图中。

20.2. 为模板添加权限

使用以下步骤为团队添加权限。

流程

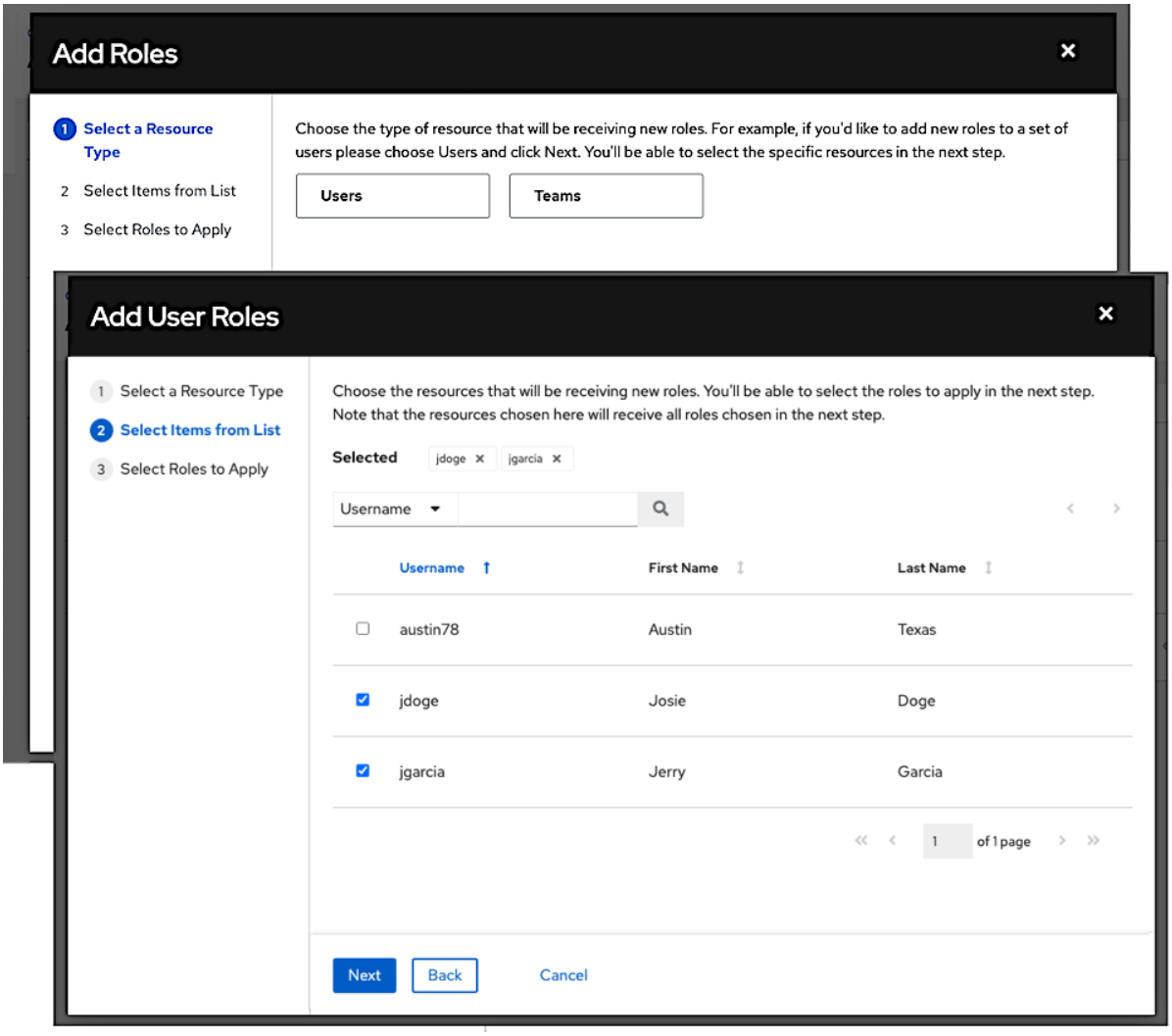
1. 在导航面板中，选择 **Resources** → **Templates**。

2. 选择一个模板，并在 **Access** 选项卡中点 **Add**。

3. 选择 "用户" 或 "团队"，然后单击下一步。

4. 点名称旁边的复选框从列表中选择一个或多个用户或团队，将它们添加为成员，然后单击 **Next**。

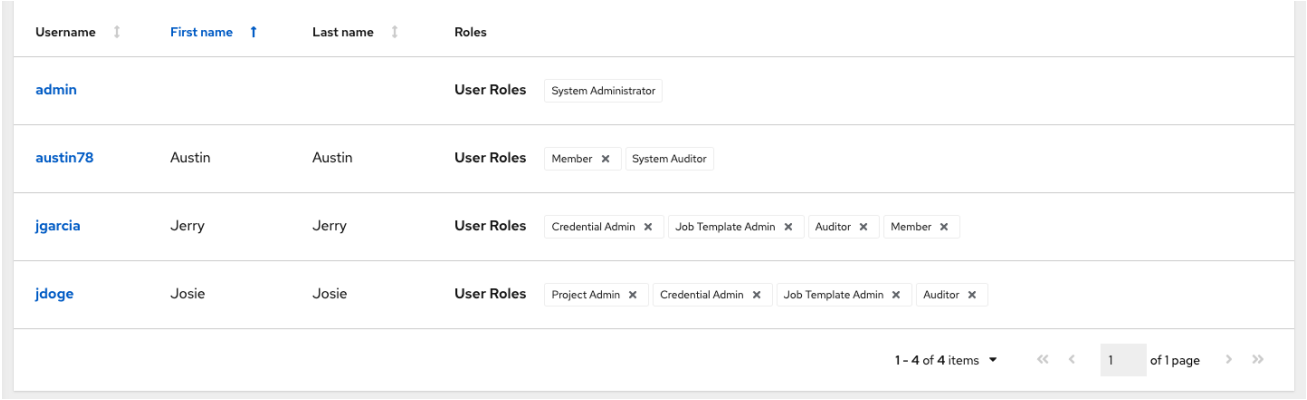
   以下示例显示了要添加的两个用户：



5. 选择您希望用户或团队具有的角色。确保向下滚动以获得完整的角色列表。每种资源具有不同的可用选项。

6. 点 **Save** 将角色应用到所选用户或团队，并将它们添加为成员。

添加用户和团队关闭的窗口，以显示为每个用户和团队分配的更新角色：



要删除特定用户的角色，请点其资源旁的 ✖ 图标。

这会出现确认对话框，要求您确认解除关联。

20.3. 删除作业模板

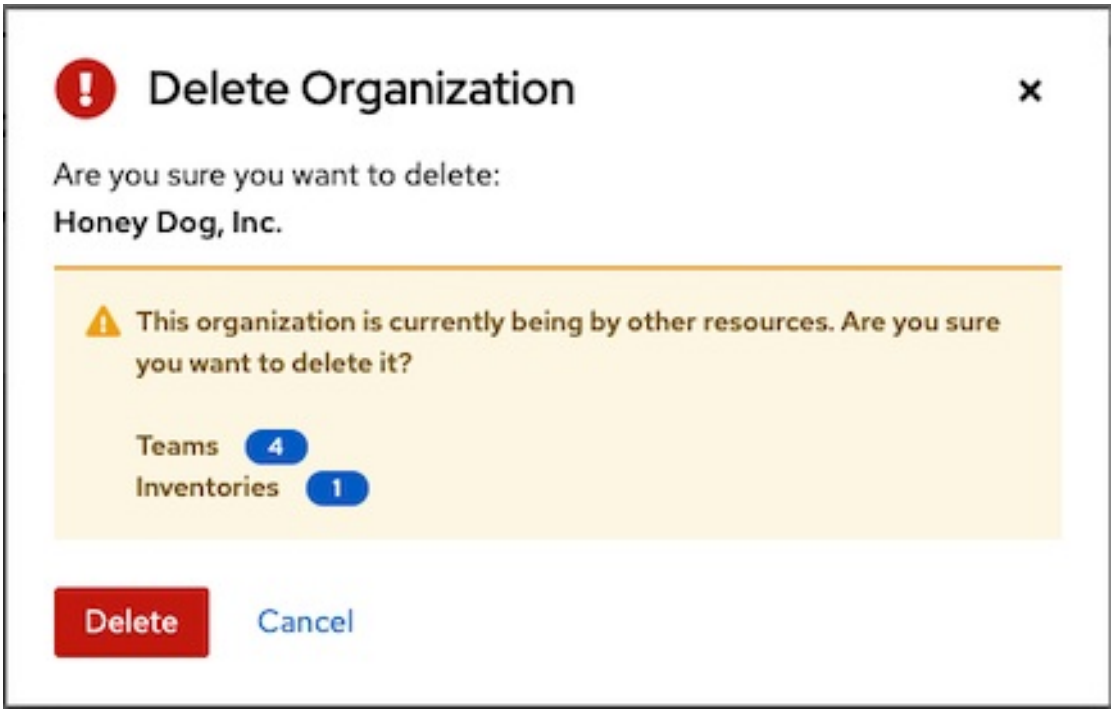
在删除作业模板前，请确保它不在工作流作业模板中使用。

流程

1. 使用以下方法之一删除作业模板：
   - 选中一个或多个作业模板旁边的复选框，再单击删除。
   - 单击所需的作业模板，再单击 **Details** 页面中的 **Delete**。

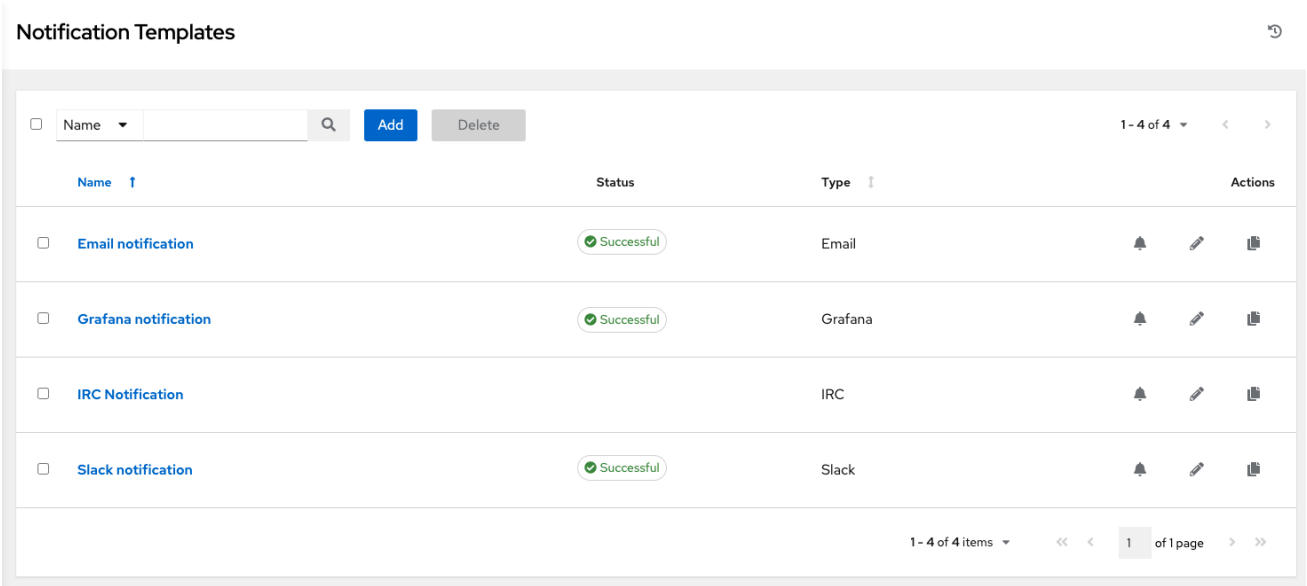
注意

如果删除由其他工作项目使用的项目，则会打开一个信息，列出受删除影响的项目，并提示您确认删除。有些屏幕包含无效的或之前删除的项目，且无法运行。以下是该消息的示例：



20.4. 使用通知

在导航面板中，选择 **Administration → Notifications**。这可让您查看您设置的任何通知集成及其状态（如果已运行）。



使用切换按钮启用或禁用要与特定模板搭配使用的通知。如需更多信息，请参阅 启用和禁用通知。

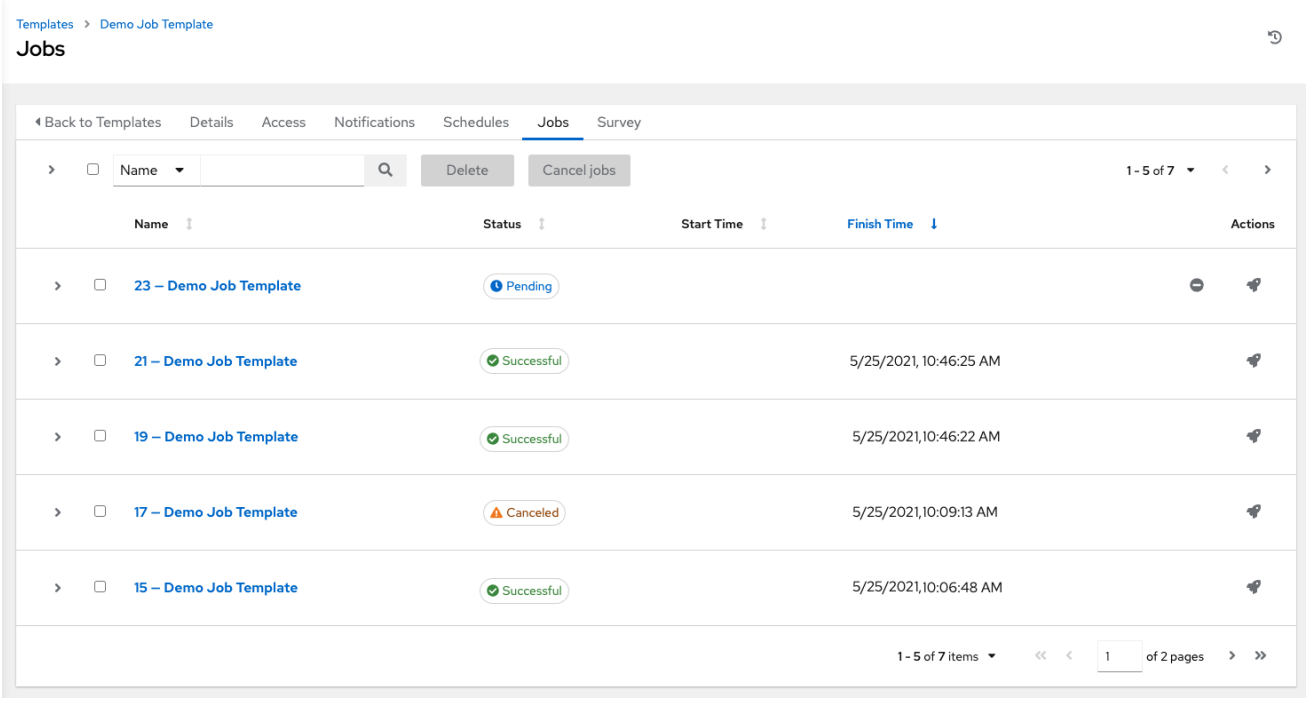
如果没有设置通知，请点击 **Add** 来创建新通知。有关配置各种通知类型和扩展消息传递的更多信息，请参阅 通知类型。

20.5. 查看完成的作业

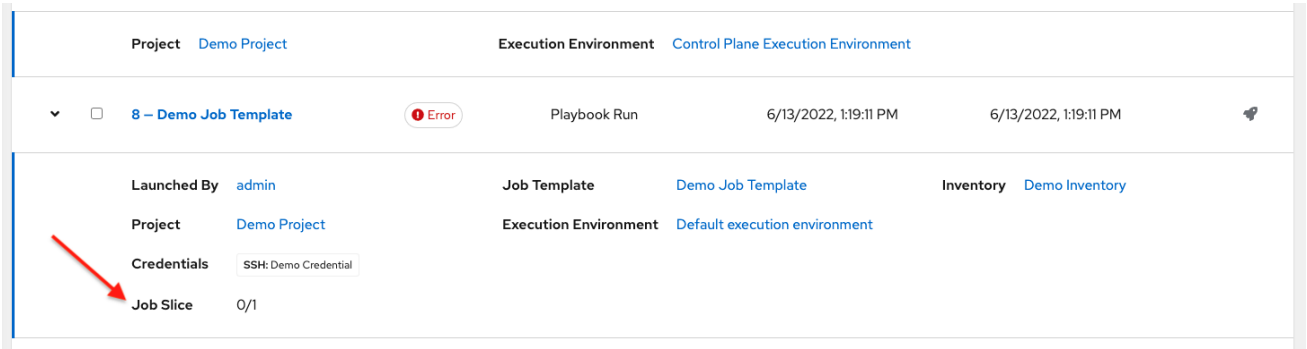
Jobs 选项卡提供已运行的作业模板列表。点每个作业旁的展开图标查看以下详情：

- **Status**
- **ID** 和名称
- 作业类型
- 开始和完成的时间
- 启动了作业，以及使用哪个模板、清单、项目和凭证。

您可以使用任何这些条件过滤已完成的作业列表。

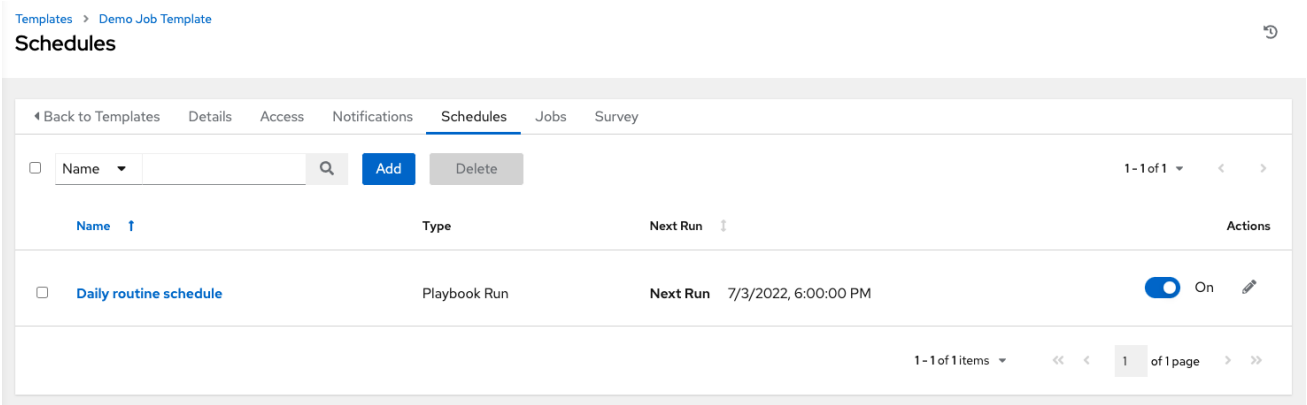


在此列表中显示的切片作业会进行相应标记，包含已运行的分片作业数量：



20.6. 调度作业模板

从 **Schedules** 选项卡访问特定作业模板的计划。

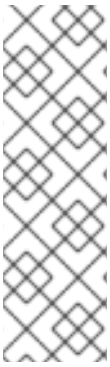


流程

- 要调度作业模板，请选择 **Schedules** 选项卡，然后选择适当的方法：
  - 如果已经设置了调度，请检查、编辑、启用或禁用您的调度首选项。
  - 如果还没有设置调度，请参阅 **Schedules** 以了解更多信息。

如果您为 **Credentials** 字段 选择了 **Prompt on Launch**，并且您创建或编辑作业模板的调度信息，则会在 **Schedules** 表单上显示一个 **Prompt** 选项。

在保存前，您无法在 **Prompt** 对话框中删除默认机器凭证，而无需将其替换为其他机器凭证。



注意

要在调度上设置 **extra_vars**，您必须在作业模板中为 变量 选择 **Prompt on Launch**，或者在作业模板上配置和启用问卷调查。

然后，回答的问卷调查问题将变为 **extra_vars**。

20.7. 任务模板中的问卷调查

运行或检查 的作业类型 提供了一种在 作业模板 创建或编辑屏幕中设置问卷调查的方法。问卷调查为 **playbook** 设置额外变量，类似于 **Prompt for Extra Variables**，但采用用户友好的问题和答案方式。调查还允许验证用户输入。选择 **Survey** 选项卡来创建问卷调查。

Example

调查可用于多种情况。例如，操作希望为开发人员提供一个 **"push to stage"** 按钮，无需提高 **Ansible** 知识即可运行。启动后，此任务可以提示输入问题的回答，如"应该如何发布标签？"。

可以询问很多类型的问题，包括多项选择题。

20.7.1. 创建问卷调查

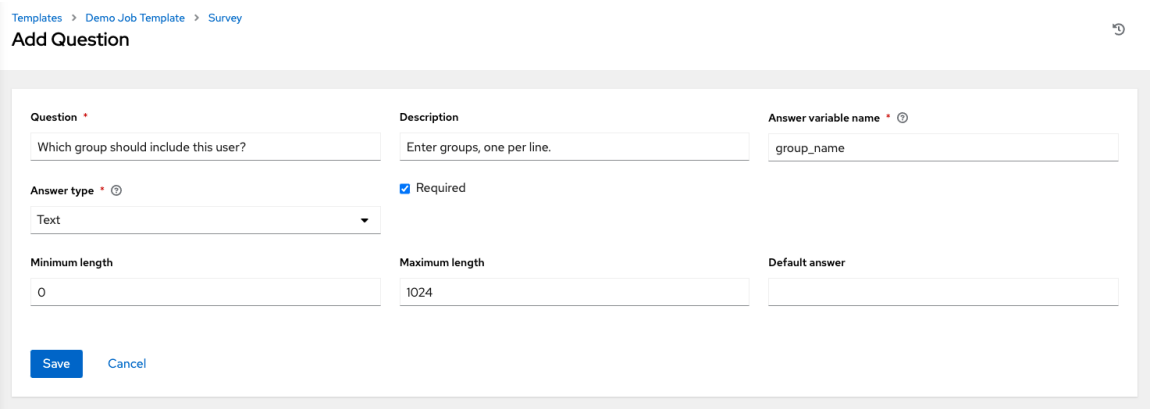
流程

1. 从 调查 选项卡中，单击 添加。

2. 问卷调查可由任意数量的问题组成。对于每个问题，输入以下信息：

   - 问题 ：询问用户的问题。
   - 可选： 描述**:** 描述用户被要求的内容。
   - 回答变量名称 ：用于存储用户响应的 **Ansible** 变量名称。这是 **playbook** 要使用的变量。变量名称不能包含空格。
   - 回答类型 ：从以下问题类型中选择：
     - 文本**:** 单行文本。您可以为此回答设置最小和最大长度（字符数）。
     - 文本域：多行文本字段。 您可以为此回答设置最小和最大长度（字符数）。
     - 密码 ：响应被视为敏感信息，就像实际密码一样被处理。您可以为此回答设置最小和最大长度（字符数）。
     - 多项选择（单选 ）：选项列表，一次只能选择一个。在 **Multiple Choice Options** 字段中输入选项（每行一个）。
     - 多项选择（多选） ：选项列表，一次可以选择任意数量。在 **Multiple Choice Options** 字段中输入选项（每行一个）。
     - 整数 ：整数。您可以为此回答设置最小和最大长度（字符数）。
     - 浮点 数：十进制数。您可以为此回答设置最小和最大长度（字符数）。

- 必需 ：用户是否需要回答这个问题。
- 最小长度和最大长度 ：指定是否需要回答中的某个长度。
- 默认回答 ：问题的默认回答。这个值在界面中预先填充，并在用户未提供回答时使用。



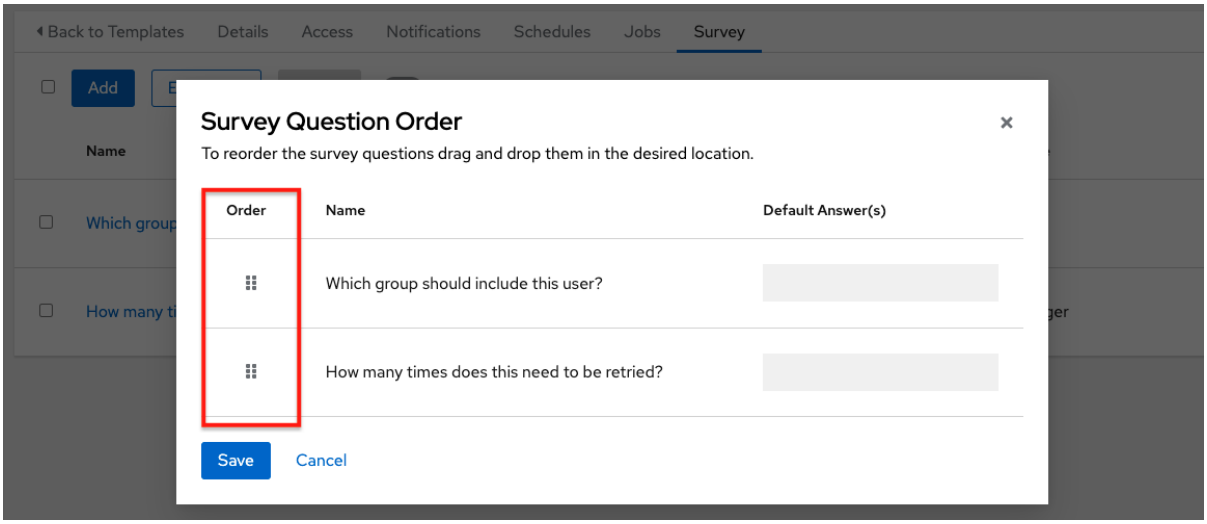
3. 输入问题信息后，单击 **Save** 以添加问题。

调查问题显示在 **Survey** 列表中。对于任何问题，您可以点

来编辑它。

选中每个问题旁边的框，然后单击 **Delete** 以删除问题，或者使用菜单栏中的切换选项启用或禁用调查提示。

如果您有多个调查问题，点 **Edit Order** 来通过单击并拖动网格图标来重新安排问题的顺序。



4. 要添加更多问题，请点 添加。

20.7.2. 可选的问卷调查问题

问卷调查问题中的 **Required** 设置决定了对于与之交互的用户是否是可选的。

可选的问卷调查变量也可以传递给 **extra_vars** 中的 **playbook**。

- 如果非文本变量（输入类型）标记为可选，且没有填写，则不会将任何问卷调查 **extra_var** 传递给 **playbook**。
- 如果文本输入或文本区域输入标记为可选，未填写，且 最小长度 **> 0**， 则不会将问卷调查 **extra_var** 传递给 **playbook**。
- 如果文本输入或文本区域输入标记为可选，未填写，并且具有 最小长度 **=== 0**， 则调查 **extra_var** 将传递给 **playbook**，并将值设为空字符串**("")**。

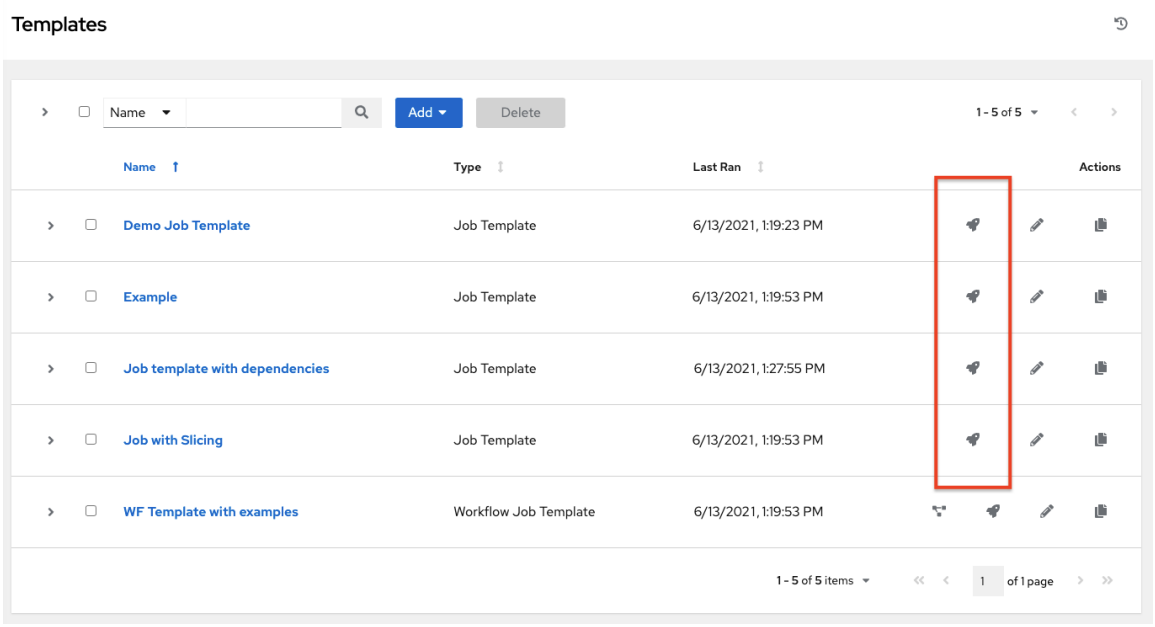
20.8. 启动作业模板

自动化控制器的一个优点是 **Ansible playbook** 的按钮式部署。您可以配置模板，以存储您通常在命令行上传递给 **Ansible playbook** 的所有参数。除了 **playbook** 外，模板还会传递清单、凭证、额外变量以及您可以在命令行上指定的所有选项和设置。

更简单的部署会在每次运行时以相同的方式运行 **playbook** 来提高一致性，并允许您委派职责。

流程

- 使用以下方法之一启动作业模板：
  - 在导航面板中，选择 **Resources** → **Templates**，再单击作业模板的 **Launch**。



- 在您要启动的作业模板的作业模板视图中，单击 **Launch**。

作业可能需要额外信息才能运行。启动时可以请求以下数据：

- 设置的凭证
- 为任意参数选择了 **Prompt on Launch** 选项
- 已设置为 **Ask**的密码或密码短语
- 问卷调查（如果已经为作业模板配置了问卷调查）
- 额外变量（如果作业模板要求提供）

注意

如果作业有用户提供的值，则重启时会考虑这些值。如果用户没有指定值，则作业将使用作业模板中的默认值。作业不会按原样重新启动。它们通过用户提示重新应用到作业模板。

如果您在一个标签页中提供值，返回上一个标签页，请继续下一个标签页会导致在剩余标签页中重新提供值。确保您按照提示出现的顺序填写标签页。

启动后，自动化控制器会在 **Jobs** 选项卡下自动将 **Web** 浏览器重定向到此作业的 **Job Status** 页面。

您可以从列表视图中重新启动最新的作业，以便针对指定清单中的所有主机或仅仅是失败的主机重新运行。如需更多信息，请参阅 作业 部分。

当分片作业正在运行时，作业列表会显示工作流和作业分片，以及用于单独查看其详情的链接。

注意

您可以使用 **API** 中新添加的端点 **/api/v2/bulk/job_launch** 来批量启动作业。此端点接受 **JSON**，您可以指定用于启动的统一作业模板（如作业模板和项目更新）的列表。用户必须具有启动所有作业的适当权限。如果没有启动所有作业，则返回错误，指示操作无法完成的原因。使用 **OPTIONS** 请求返回相关模式。如需更多信息，请参阅自动化控制器 **API** 指南中的参考部分中的 **Bulk** 端点。

20.9. 复制作业模板

如果您复制作业模板，它不会复制任何关联的调度、通知或权限。用户必须由用户或管理员创建作业模板副本重新创建调度和通知。复制作业模板的用户被授予管理员权限，但没有将权限分配给作业模板。

流程

1. 在导航面板中，选择 **Resources** → **Templates**。
2. 点击与您要复制的模板关联的 图标。
   - 在模板列表中会显示带有作为复制来源的新模板的名称和一个时间戳。
3. 单击以打开新模板，然后单击 编辑。

4. 将 **Name** 字段的内容替换为新名称，并提供或者修改其他字段中的条目以完成此页面。

5. 点击 **Save**。

20.10. 扫描作业模板

从自动化控制器 **3.2** 开始，不再支持扫描作业。此系统跟踪功能用作捕获和存储事实作为历史数据的方法。现在，事实通过事实缓存存储在控制器中。如需更多信息，请参阅 事实缓存。

在自动化控制器 **3.2** 之前，系统中的作业模板扫描作业将转换为运行类型，如正常的作业模板。它们保留其相关资源，如清单和凭证。默认情况下，没有相关项目的作业模板扫描作业会被分配一个特殊的 **playbook**。您还可以使用自己的扫描 **playbook** 指定项目。为每个指向 **awx-facts-playbooks** 的机构创建一个项目，并将作业模板设置为 **playbook: https://github.com/ansible/tower-fact-modules/blob/master/scan_facts.yml**。

20.10.1. 事实扫描 playbook

扫描作业 **playbook scan_facts.yml** 包含三种 事实扫描模块 （软件包、服务和文件）的调用，以及 **Ansible** 的标准事实收集。**scan_facts.yml playbook** 文件类似如下：

```
- hosts: all
  vars:
    scan_use_checksum: false
    scan_use_recursive: false
  tasks:
    - scan_packages:
    - scan_services:
    - scan_files:
        paths: '{{ scan_file_paths }}'
        get_checksum: '{{ scan_use_checksum }}'
        recursive: '{{ scan_use_recursive }}'
      when: scan_file_paths is defined
```

scan_files 事实模块是唯一接受参数的模块，通过扫描作业模板上的 **extra_vars** 传递：

scan_file_paths:/tmp/scan_use_checksum: true scan_use_recursive: true

- **scan_file_paths** 参数可以有多个设置（如 **/tmp/** 或 **/var/log**）。

- **scan_use_checksum** 和 **scan_use_recursive** 参数也可以设置为 **false** 或省略。省略与 **false** 设置相同。

扫描作业模板应启用 **become**，并使用 成为可能 的凭证。您可以通过从选项列表中检查 **Privilege Escalation** 来启用 **become** ：

20.10.2. 支持的 scan_facts.yml OS

如果您使用带有使用事实缓存的 **scan_facts.yml playbook**，请确保使用以下支持的操作系统之一：

- **Red Hat Enterprise Linux 5、6、7、8 和 9**
- **Ubuntu 23.04** （支持 **Ubuntu** 已被弃用，将在以后的版本中删除）
- **OEL 6 和 7**
- **SLES 11 和 12**
- **Debian 6、7、9、10、11 和 12**
- **Fedora 22、23 和 24**
- **Amazon Linux 2023.1.20230912**

其中一些操作系统需要初始配置来运行 **python**，或者有权访问 **python** 软件包，如 **python-apt**，扫描模块依赖。

20.10.3. 预扫描设置

以下是配置某些发行版本的 **playbook** 示例，以便可以针对它们运行扫描作业：

```
Bootstrap Ubuntu (16.04)
---
- name: Get Ubuntu 16, and on ready
 hosts: all
 sudo: yes
 gather_facts: no
 tasks:
 - name: install python-simplejson
   raw: sudo apt-get -y update
   raw: sudo apt-get -y install python-simplejson
   raw: sudo apt-get install python-apt

Bootstrap Fedora (23, 24)
---
- name: Get Fedora ready
 hosts: all
 sudo: yes
 gather_facts: no
 tasks:
 - name: install python-simplejson
   raw: sudo dnf -y update
   raw: sudo dnf -y install python-simplejson
   raw: sudo dnf -y install rpm-python
```

20.10.4. 自定义事实扫描

自定义事实扫描的 **playbook** 与 **Fact 扫描 playbook** 部分中的示例类似。例如，只使用自定义 **scan_foo Ansible** 事实模块的 **playbook** 类似如下：

```
scan_foo.py:
def main():
    module = AnsibleModule(
        argument_spec = dict())


    foo = [
      {
        "hello": "world"
      },
      {
        "foo": "bar"
      }
    ]
    results = dict(ansible_facts=dict(foo=foo))
    module.exit_json(**results)


main()
```

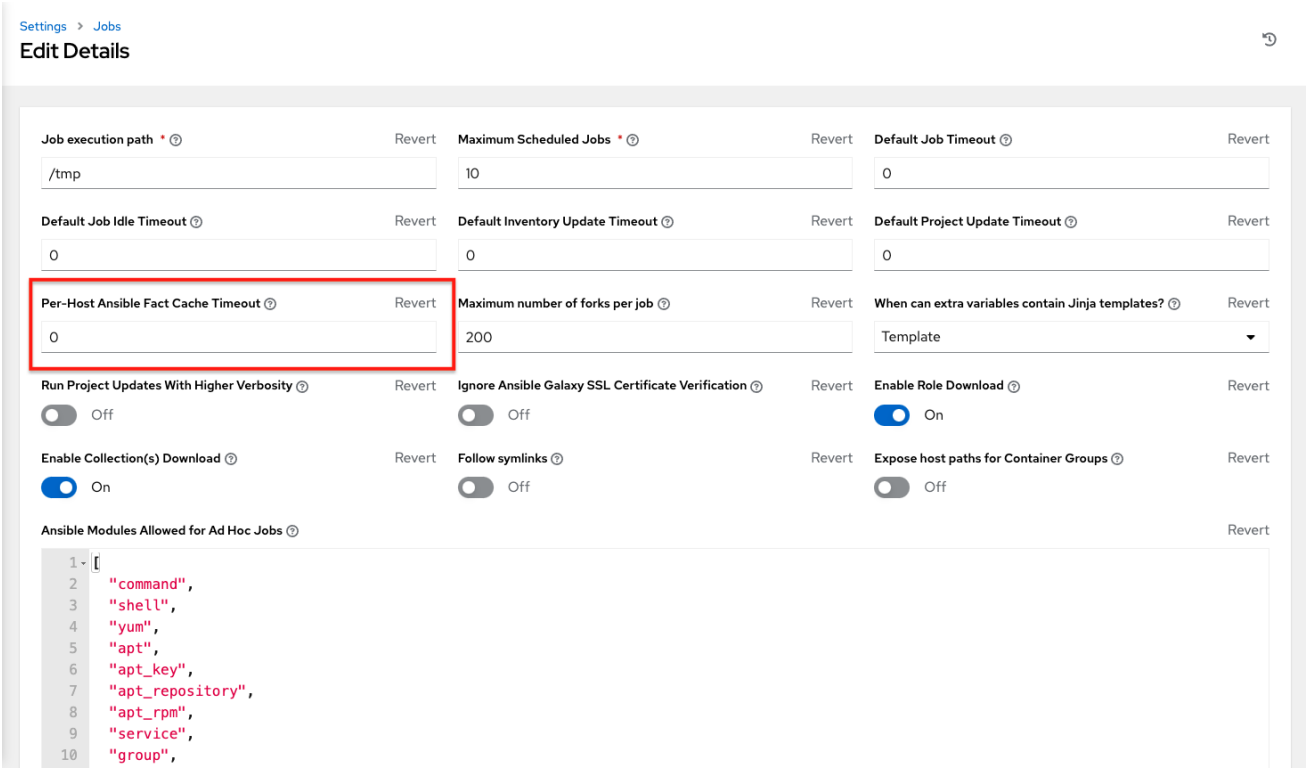
要使用自定义事实模块，请确保它位于扫描作业模板中使用的 **Ansible** 项目的 **/library/** 子目录中。这个事实扫描模块返回一组硬编码的事实：

```
[
  {
    "hello": "world"
  },
  {
    "foo": "bar"
  }
]
```

有关更多信息，请参阅 **Ansible** 文档中的 开发模块 部分。

20.10.5. 事实缓存

自动化控制器可以通过 **Ansible** 事实缓存插件来基于每个主机存储和检索事实。这个行为可根据每个作业模板进行配置。默认情况下，事实缓存会被关闭，但可以启用来为与作业运行相关的清单中所有主机提供事实请求。这可让您使用带有**- limit** 的作业模板，同时仍可访问整个主机事实清单。通过进入 **Settings** 并从 **Jobs** 选项中选择 作业设置，可指定插件强制在每个主机的全局超时设置：



在启动使用事实缓存（**use_fact_cache=True**的作业后，每个主机的 **ansible_facts** 均由控制器存储在作业的清单中。

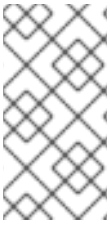
自动化控制器附带的 **Ansible** 事实缓存插件在启用了事实缓存的作业上启用 **(use_fact_cache=True)**。

当一个启用了事实缓存**(use_fact_cache=True)**的作业正在运行时，自动化控制器会恢复清单中主机的所有记录。任何比每个主机当前存储事实更新时间更新的记录都会在数据库中更新。

新的和更改的事实通过自动化控制器的日志记录功能记录。特别是 **system_tracking** 命名空间或日志记录 器。日志记录有效负载包括以下字段：

- **host_name**
- **inventory_id**
- **ansible_facts**

Ansible 事实是自动化控制器清单 **inventory_id** 中 **host_name** 的所有 **Ansible** 事实的字典。

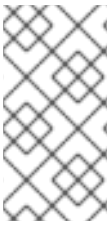


注意

如果主机名包含正斜杠**(/)**，事实缓存不适用于该主机。如果您的清单有 **100** 个主机，且一个主机的名称中有一个 /，则剩余的 **99** 个主机仍然收集事实。

20.10.6. 事实缓存的好处

事实缓存可让您通过运行事实收集来节省时间。如果您在某个作业中有一个针对一千个主机和分叉运行的 **playbook**，您可以花费 **10** 分钟在所有这些主机上收集事实。但是，如果您定期运行作业，第一次运行会缓存这些事实，下一次运行会从数据库中拉取它们。这可减少针对大型清单（包括智能清单）的作业运行时。



注意

不要修改 **ansible.cfg** 文件以应用事实缓存。自定义事实缓存可能会与控制器的事实缓存功能冲突。您必须使用与自动化控制器附带的事实缓存模块。

您可以选择在作业模板窗口的 **Options** 字段中启用缓存的事实来在作业中使用缓存的事实。

若要清除事实，请运行 **Ansible clear_facts** meta 任务。以下是使用 **Ansible clear_facts meta** 任务的示例 **playbook**。

```
- hosts: all
  gather_facts: false
  tasks:
    - name: Clear gathered facts from all currently targeted hosts
      meta: clear_facts
```

您可以在以下找到事实缓存的 **API** 端点：

http://<controller server name>/api/v2/hosts/x/ansible_facts

20.11. 将云凭证与云清单搭配使用

在同步云清单时，可以使用云凭证。它们也可以与作业模板关联，并包含在运行时环境中，供 **playbook** 使用。支持以下云凭证：

- **Openstack**
- **Amazon Web Services**
- **Google**
- **Azure**
- **VMware**

20.11.1. OpenStack

以下示例 **playbook** 调用 **nova_compute Ansible OpenStack** 云模块并需要凭证：

- **auth_url**
- **username**
- **password**
- 项目名称

这些字段通过环境变量 **OS_CLIENT_CONFIG_FILE** 提供给 **playbook**，它指向控制器基于云凭证内容的 **YAML** 文件。以下示例 **playbook** 将 **YAML** 文件加载到 **Ansible** 变量空间中：

- **OS_CLIENT_CONFIG_FILE** 示例：

```
clouds:
  devstack:
    auth:
      auth_url: http://devstack.yoursite.com:5000/v2.0/
      username: admin
      password: your_password_here
      project_name: demo
```

- **Playbook** 示例：

```
- hosts: all
  gather_facts: false
  vars:
    config_file: "{{ lookup('env', 'OS_CLIENT_CONFIG_FILE') }}"
    nova_tenant_name: demo
    nova_image_name: "cirros-0.3.2-x86_64-uec"
    nova_instance_name: autobot
    nova_instance_state: 'present'
    nova_flavor_name: m1.nano


    nova_group:
      group_name: antarctica
      instance_name: deceptacon
      instance_count: 3
```

```
tasks:
  - debug: msg="{{ config_file }}"
  - stat: path="{{ config_file }}"
    register: st
  - include_vars: "{{ config_file }}"
    when: st.stat.exists and st.stat.isreg


  - name: "Print out clouds variable"
    debug: msg="{{ clouds|default('No clouds found') }}"


  - name: "Setting nova instance state to: {{ nova_instance_state }}"
    local_action:
      module: nova_compute
      login_username: "{{ clouds.devstack.auth.username }}"
      login_password: "{{ clouds.devstack.auth.password }}"
```

20.11.2. Amazon Web Services

Amazon Web Services (AWS)云凭证在 **playbook** 执行过程中作为以下环境变量公开（在作业模板中，选择您的设置所需的云凭证）：

- **AWS_ACCESS_KEY_ID**
- **AWS-SECRET_ACCESS_KEY**

每个 **AWS** 模块在通过控制器运行时都会隐式使用这些凭证，而无需设置 **aws_access_key_id** 或 **aws_secret_access_key** 模块选项。

20.11.3. Google

Google 云凭证在 **playbook** 执行过程中作为以下环境变量公开（在作业模板中，选择您的设置所需的云凭证）：

- **GCE_EMAIL**
- **GCE_PROJECT**

- **GCE_CREDENTIALS_FILE_PATH**

每个 **Google** 模块在通过控制器运行时都会隐式使用这些凭证，而无需设置 **service_account_email**、**project_id** 或 **pem_file** 模块选项。

20.11.4. Azure

Azure 云凭证在 **playbook** 执行过程中作为以下环境变量公开（在作业模板中，选择您的设置所需的云凭证）：

- **AZURE_SUBSCRIPTION_ID**
- **AZURE_CERT_PATH**

每个 **Azure** 模块在通过控制器运行时都会隐式使用这些凭证，而无需设置 **subscription_id** 或 **management_cert_path** 模块选项。

20.11.5. VMware

VMware 云凭证在 **playbook** 执行过程中作为以下环境变量公开（在作业模板中，选择您的设置所需的云凭证）：

- **VMWARE_USER**
- **VMWARE_PASSWORD**
- **VMWARE_HOST**

以下示例 **playbook** 演示了这些凭证的使用情况：

```
- vsphere_guest:
    vcenter_hostname: "{{ lookup('env', 'VMWARE_HOST') }}"
    username: "{{ lookup('env', 'VMWARE_USER') }}"
    password: "{{ lookup('env', 'VMWARE_PASSWORD') }}"
```

```
guest: newvm001
from_template: yes
template_src: linuxTemplate
cluster: MainCluster
resource_pool: "/Resources"
vm_extra_config:
  folder: MyFolder
```

20.12. 置备回调

置备回调是自动化控制器的一项功能，使主机能够启动针对自身运行的 **playbook**，而不是等待用户从自动化控制器控制台管理主机。

置备回调仅用于在调用主机上运行 **playbook**，用于云突发。云突发是一种云计算配置，它允许私有云在计算需求激增时通过"突发"进入公共云资源来访问公共云资源。

Example

需要客户端到服务器通信（如传输授权密钥）的新实例，而不是对另一主机运行作业。这为自动配置以下内容提供：

- 另一个系统置备后的系统（如 **AWS** 自动扩展，或操作系统置备系统，如 **kickstart** 或预装）。
- 以编程方式启动作业，而不直接调用自动化控制器 **API**。

启动的作业模板仅针对请求调配的主机运行。

这通常通过 **firstboot** 类型脚本或从 **cron** 访问。

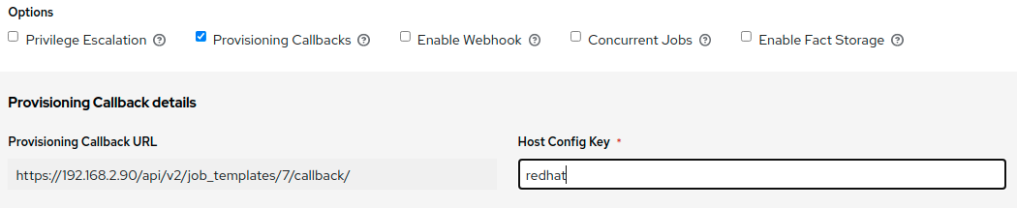
20.12.1. 启用部署回调

流程

- 要启用回调，请选中作业模板中的 **Provisioning Callbacks** 复选框。这将显示作业模板的 **Provisioning Callback URL**。

注意

如果要将自动化控制器的置备回调功能与动态清单搭配使用，请为作业模板中使用的清单组设置 **Update on Launch**。



回调还需要主机配置密钥，以确保具有 **URL** 的外部主机无法请求配置。为主机配置密钥提供自定义值。主机密钥可以在多个主机间重复使用，从而将此作业模板应用到多个主机。如果要控制哪些主机可以请求配置，可以随时更改该密钥。

使用 **REST** 手动回调：

流程

1. 查看 **UI** 中的回调 **URL**，格式为：**https://<CONTROLLER_SERVER_NAME>/api/v2/job_templates/7/callback/**
   - 示例 **URL** 中的 **"7"** 是自动化控制器中的作业模板 **ID**。
2. 确保来自主机的请求是 **POST**。以下是使用 **curl** 的示例（全部在一行）：

```
curl -k -f -i -H 'Content-Type:application/json' -XPOST -d '{"host_config_key": "redhat"}' \
        https://<CONTROLLER_SERVER_NAME>/api/v2/job_templates/7/callback/
```

3. 确保在清单中定义了请求的主机，以便回调成功。

故障排除

如果自动化控制器无法按名称或 **IP** 地址在您定义的一个清单中根据名称或 **IP** 地址查找主机，则请求将被拒绝。以这种方式运行作业模板时，请确保针对自身启动 **playbook** 的主机位于清单中。如果清单中缺少主机，则作业模板会失败，并显示 **No Hosts Matched type** 错误消息。

如果您的主机不在清单中，并且为清单组自动化控制器设置了 **Update on Launch**，则在运行回调前尝试更新基于云的清单源。

验证

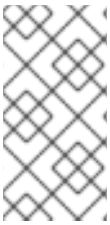
成功请求会在 **Jobs** 标签页中生成一个条目，您可以在其中查看结果和历史记录。您可以使用 **REST** 访问回调，但推荐的使用回调方法是使用自动化控制器附带的示例脚本之一：

- **/usr/share/awx/request_tower_configuration.sh (Linux/UNIX)**
- **/usr/share/awx/request_tower_configuration.ps1 (Windows)**

通过传递 **-h** 标志，其用法在文件的源代码中进行了描述，如下所示：

```
./request_tower_configuration.sh -h
Usage: ./request_tower_configuration.sh <options>

Request server configuration from Ansible Tower.

OPTIONS:
 -h      Show this message
 -s      Controller server (e.g. https://ac.example.com) (required)
 -k      Allow insecure SSL connections and transfers
 -c      Host config key (required)
 -t      Job template ID (required)
 -e      Extra variables
```

这个脚本可以重试命令，因此比简单的 **curl** 请求相比，使用回调更为可靠的方法。脚本每分钟重试一次，最多十分钟。



注意

这是示例脚本。如果您在检测到失败情况时需要更多的动态行为，请编辑这个脚本，因为任何非 **200** 错误代码都可能不是需要重试的临时错误。

您可以在自动化控制器中将回调与动态清单搭配使用。例如，从其中一个受支持的云提供商拉取云清单时。在这些情况下，除了设置 **Update On Launch** 之外，请确保为清单源配置清单缓存超时，以避免

对云的 **API** 端点造成影响。由于 **request_tower_configuration.sh** 脚本每分钟轮询一次，因此建议为清单（在清单源本身上配置）的缓存无效时间为一两分钟。

不建议从 **cron** 任务运行 **request_tower_configuration.sh** 脚本，但建议每 **30** 分钟运行推荐的 **cron** 间隔。重复的配置可以通过调度自动化控制器来处理，以便大多数用户使用回调的主要用途是启用在上线时引导至最新配置的基础镜像。在第一次引导时运行是最佳实践。首次启动脚本是通常自我删除的初始化脚本，因此您设置了一个调用 **request_tower_configuration.sh** 脚本副本的 **init** 脚本，并将其设置为自动扩展镜像。

20.12.2. 将额外变量传递给置备回调

您可以在 **Provisioning Callback** 中传递 **extra_vars**，其方式与在常规作业模板中相同。要传递 **extra_vars**，发送的数据必须是 **POST** 正文的一部分，作为应用程序或 **JSON**，作为内容类型。

流程

- 使用以下方法之一传递额外变量：
  - 在添加您自己的 **extra_vars** 时要传递时，使用以下 **JSON** 格式作为示例：

```
'{"extra_vars": {"variable1":"value1","variable2":"value2",...}}'
```

  - 使用 **curl** 将额外变量传递给作业模板调用：

```
root@localhost:~$ curl -f -H 'Content-Type: application/json' -XPOST \
-d '{"host_config_key": "redhat", "extra_vars": "{\"foo\": \"bar\"}"}' \
https://<CONTROLLER_SERVER_NAME>/api/v2/job_templates/7/callback
```

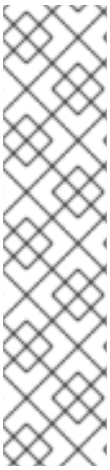
如需更多信息，请参阅 *自动化控制器管理指南中的使用* **Curl** 启动作业。

20.13. 额外变量

当您传递问卷调查变量时，它们作为自动化控制器中的额外变量**(extra_vars)**传递。但是，将额外变量传递给作业模板（就像对问卷调查的操作一样）可能会覆盖从清单和项目传递的其他变量。

默认情况下，**extra_vars** 标记为 **!unsafe**，除非您在作业模板的 **Extra Variables** 部分中指定它们。这些是信任的，因为它们只能由具有足够特权的用户添加或编辑作业模板。例如，嵌套变量在作为提示符输

入时不会扩展，因为 **Jinja** 方括号被视为字符串。有关不安全变量的更多信息，请参阅 **Unsafe 或 raw string**。



注意

只有在以下条件之一被满足时，传递给作业启动 **API** 的 **extra_vars** 才有效：

- 它们与启用的问卷调查中的变量对应。
- **ask_variables_on_launch** 设置为 **True**。

Example

为 **debug = true** 定义了清单的变量。此变量 **debug = true** 可能会在作业模板问卷调查中被覆盖。

要确保您传递的变量不会被覆盖，请在问卷调查中重新定义变量来确保包括它们。可以在清单、组和主机级别上定义额外的变量。

如果您要指定 **ALLOW_JINJA_IN_EXTRA_VARS** 参数，请参阅 *自动化控制器管理指南中的控制器* 提示和 **Tricks** 部分，以便在控制器 **UI** 的 **Jobs Settings** 屏幕中进行配置。

作业模板额外变量字典与问卷调查变量合并。

以下是 **YAML** 和 **JSON** 格式的 **extra_vars** 的一些简化示例：

- **YAML** 格式的配置：

```
launch_to_orbit: true
satellites:
  - sputnik
  - explorer
  - satcom
```

- **JSON** 格式的配置：

```
{
  "launch_to_orbit": true,
  "satellites": ["sputnik", "explorer", "satcom"]
}
```

下表记录了自动化控制器中变量优先级的行为（层次结构）与 **Ansible** 中的变量优先级比较。

表 **20.1.** 自动化控制器变量优先级层次结构（最后列出的优先）

Ansible	自动化控制器
角色默认值	角色默认值
动态清单变量	动态清单变量
清单变量	自动化控制器清单变量
清单 **group_vars**	自动化控制器组变量
清单 **host_vars**	自动化控制器主机变量
Playbook **group_vars**	Playbook **group_vars**
playbook **host_vars**	playbook **host_vars**
主机事实	主机事实
注册的变量	注册的变量
设置事实	设置事实
Play 变量	Play 变量
play **vars_prompt**	（不支持）
play **vars_files**	play **vars_files**
role 和 include 变量	role 和 include 变量
块变量	块变量
任务变量	任务变量
额外变量	作业模板额外变量
	作业模板调查(defaults)

Ansible	自动化控制器
	任务启动额外变量

20.13.1. 重新启动作业模板

通过设置 **launch_type** 以重新启动作业来表示重新启动作业，而不是手动 重新启动作业。重新启动行为与启动行为不同，其不会继承 **extra_vars**。

作业重新启动不会通过继承逻辑。它使用为重新启动的作业计算的相同 **extra_vars**。

Example

您启动了一个没有 **extra_vars** 的作业模板，这导致创建名为 **j1** 的作业。然后，您编辑作业模板并添加 **extra_vars** （如添加 **"{ "hello": "world" }"**）。

重新启动 **j1** 会导致创建 **j2**，但由于没有继承逻辑，并且 **j1** 没有 **extra_vars**，**j2** 没有任何 **extra_vars**。

如果您使用您在创建 **j1** 后添加的 **extra_vars** 启动作业模板，则创建重新启动作业**(j3)**包含 **extra_vars**。重新启动 **j3** 会导致创建 **j4**，这还包括 **extra_vars**。

第 21 章 作业分片

分片作业指的是分布式作业的概念。分布式作业用于在大量主机上运行作业，允许您运行多个 **ansible-playbooks**，各自在清单的子集上运行，它们可以在集群中并行调度。

默认情况下，**Ansible** 从单个控制实例运行作业。对于不需要跨主机编配的作业，作业分片可以利用自动化控制器将工作分发到集群中的多个节点。

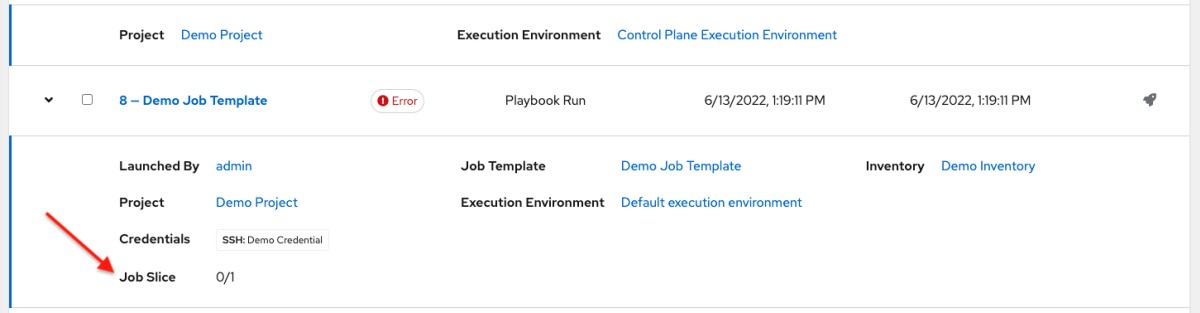
作业分片的工作原理方法是添加一个作业模板字段 **job_slice_count**，它指定要将 **Ansible** 运行分片到的作业数量。当这个数字大于 **1** 时，自动化控制器会从作业模板而不是作业生成工作流。清单在分片作业之间平均分配。然后，工作流作业启动，并像正常工作流一样继续。

在启动作业时，**API** 将返回作业资源（如果 **job_slice_count = 1**）或工作流作业资源。对应的用户界面 **(UI)**重定向到适当的屏幕，以显示运行的状态。

21.1. 作业分片注意事项

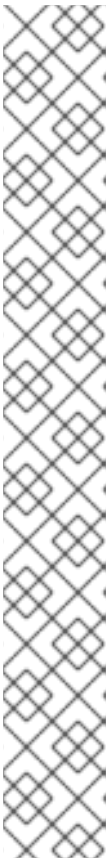
在设置作业分片时，请考虑以下几点：

- 分片作业会创建一个工作流作业，然后创建作业。
- 作业分片由作业模板、清单和分片计数组成。
- 执行后，分片作业会将每个清单分成多个"分片大小"块。然后，它将 **ansible-playbook** 的作业排队在相应清单的每个块上运行。引入 **ansible-playbook** 的清单是原始清单的简化版本，仅包含该特定片段中的主机。作业列表中显示的已完成分片作业会相应地标记，以及已运行的分片作业数量：



- 这些分片作业遵循常规调度行为（**fork** 数量，根据容量进行排队，根据清单映射分配给实例

组）。



注意

作业分片旨在水平扩展作业执行。在作业模板上启用作业分片会将清单划分成在启动时配置的分片数量，然后为每个分片启动作业。

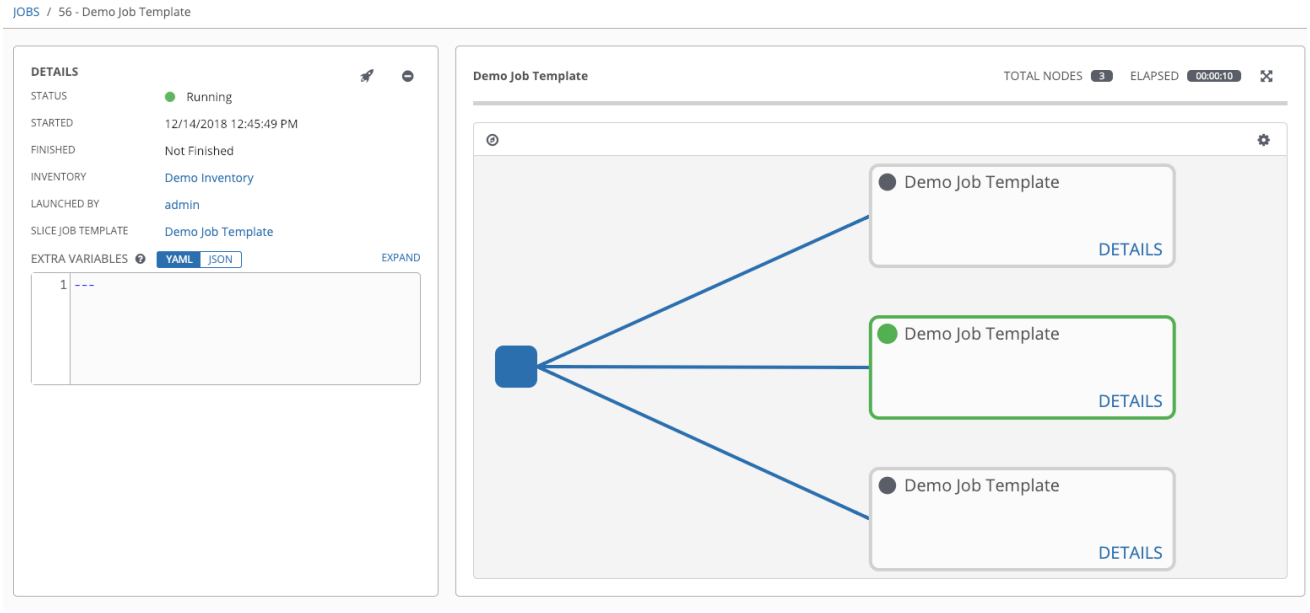
通常，分片的数量等于或小于自动化控制器节点的数量。设置大量作业分片（如数千作业分片）可能会导致性能下降，因为作业调度程序不会同时调度数千个工作流节点，这是分片作业已变得是什么。

- 带有提示或额外变量的分片作业模板的行为与标准作业模板相同，将所有变量和限制应用到生成的工作流作业集合。但是，当将限制传递给分片作业时，如果限制导致分片没有分配主机，则这些分片将失败，从而导致整个作业失败。
- 计算分布式作业的作业分片作业状态的方式与工作流作业相同。如果其子作业中存在未处理的失败，则会失败。

- 任何计划在多个主机间编配（而不是只对单个主机应用更改）的作业都不能配置为分片作业。
- 任何作业都可能会失败，自动化控制器也不会试图发现或考虑作为分片作业运行时失败的 **playbook**。

21.2. 作业分片执行行为

当作业被分片时，它们可以在任何节点上运行。系统中的容量不足可能会导致某些在不同时间运行。当分片作业运行时，作业详情会显示当前运行的工作流和作业分片，以及用于单独查看其详情的链接。



默认情况下，作业模板通常不配置为同时执行（必须在 **API** 中检查**allow_simultaneous**，或在 **UI** 中启用 **Concurrent** 作业 ）。分片会覆盖此行为，即使该设置是明确的，也意味着 **allow_simultaneous**。如需有关如何指定此功能以及 作业模板 配置中的作业分片数量的信息，请参阅作业模板。

Job templates 部分提供了在 **UI** 中执行以下操作的更多详情：

- 使用分片数量大于一的作业模板启动工作流作业。
- 启动分片作业模板后，取消整个工作流或单独的作业。
- 在分片完成运行后，重新启动整个工作流或单独的作业。
- 在启动作业模板后，查看工作流和分片作业的详情。
- 按照后续部分"搜索作业分片"部分创建后，专门搜索分片作业。

21.3. 搜索作业分片

要更轻松地查找分片作业，请使用搜索功能将搜索过滤器应用到：

- 作业列表仅显示分片作业
- 作业列表仅显示作业分片的父工作流作业
- 作业模板列表仅显示生成分片作业的作业模板

流程

- 使用以下方法之一搜索分片作业：
  - 要只显示作业列表中的分片作业，就像大多数情况一样，您可以根据类型（此处的作业）或 **unified_jobs** 过滤：

    ```
    /api/v2/jobs/?job_slice_count__gt=1
    ```

  - 仅显示作业分片的父工作流作业：

    ```
    /api/v2/workflow_jobs/?job_template__isnull=false
    ```

  - 仅显示生成分片作业的作业模板：

    ```
    /api/v2/job_templates/?job_slice_count__gt=1
    ```

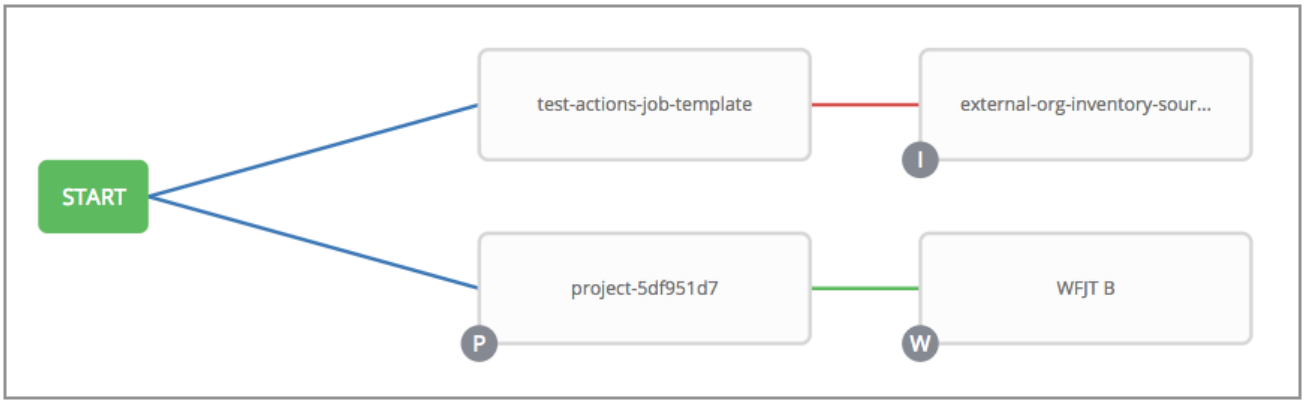
第 22 章 自动化控制器中的工作流

工作流允许您配置一系列不同的作业模板（或工作流模板），它们可能或不共享清单、playbook 或权限。

工作流具有 管理员 和执行权限，类似于作业模板。工作流完成的任务是将属于发布过程一部分的完整作业集合作为一个单元来跟踪。

作业或工作流模板使用类似图形的结构（称为节点）链接在一起。这些节点可以是作业、项目同步或清单同步。模板可以是不同工作流的一部分，也可以在同一工作流中多次使用。在启动工作流时，图形结构的副本会保存到工作流作业中。

以下示例显示了包含所有三个工作流以及工作流作业模板的工作流：



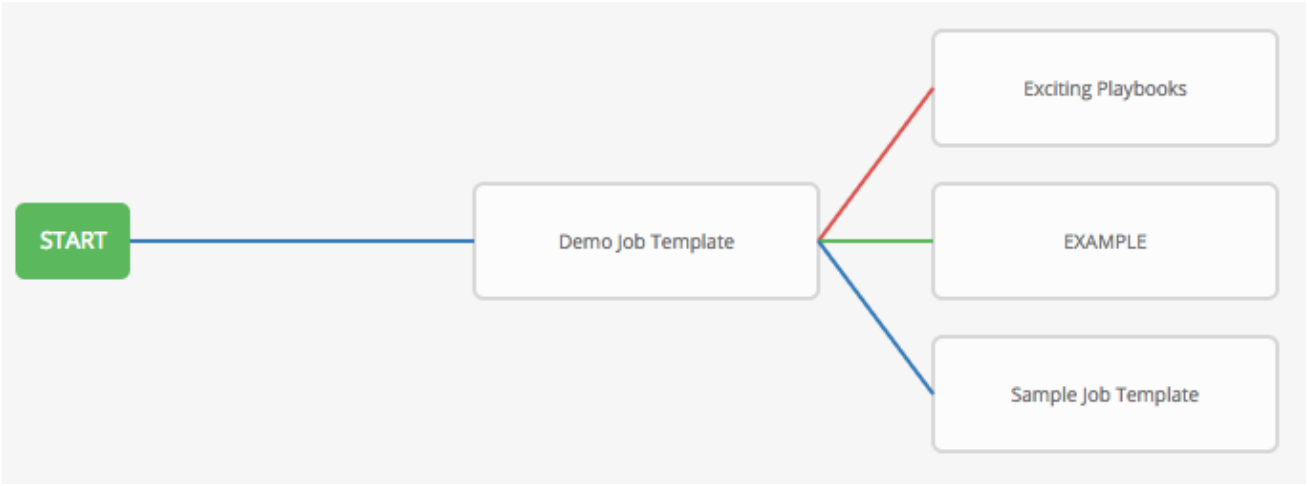
当工作流运行时，作业会从节点的链接模板生成。链接到具有提示驱动的字段(job_type、job_tags、skip_tags、limit)的节点可以包含这些字段，且不会在启动时提示。提示凭证或清单的作业模板没有默认值，无法包含在工作流中。

22.1. 工作流场景和注意事项

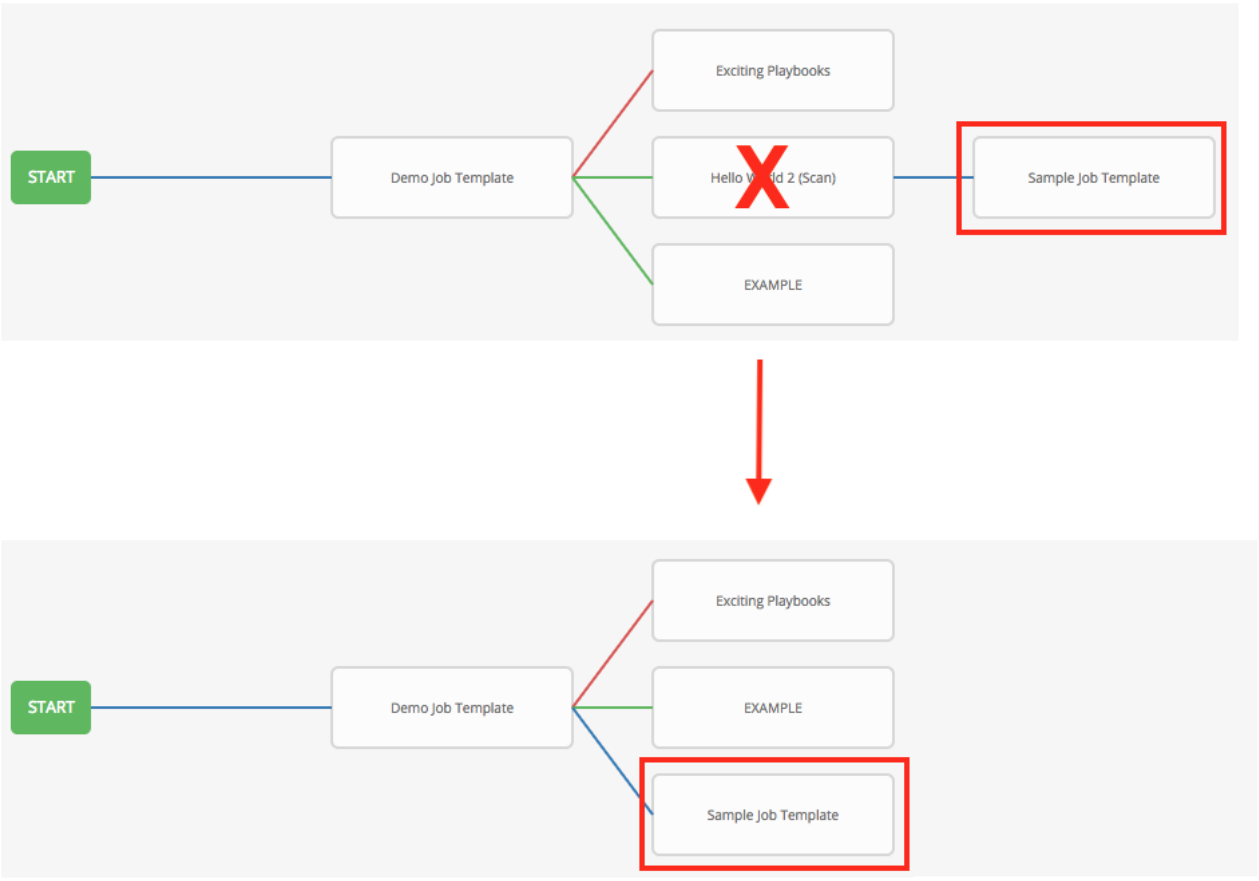
在构建工作流时，请考虑以下事项：

- 根节点默认设置为 ALWAYS，且无法编辑。

- 节点可以有多个父项，子项可以链接到任何 **success, failure,** 或 **always** 状态。如果为 **always**，则状态既不会成功，也没有失败。**State** 在节点级别应用，而不是在工作流作业模板级别应用。工作流作业标记为成功，除非被取消或遇到错误。



- 如果您删除工作流中的作业或工作流模板，则之前连接到删除的节点会自动连接上游，并保留边缘类型，如下例所示：

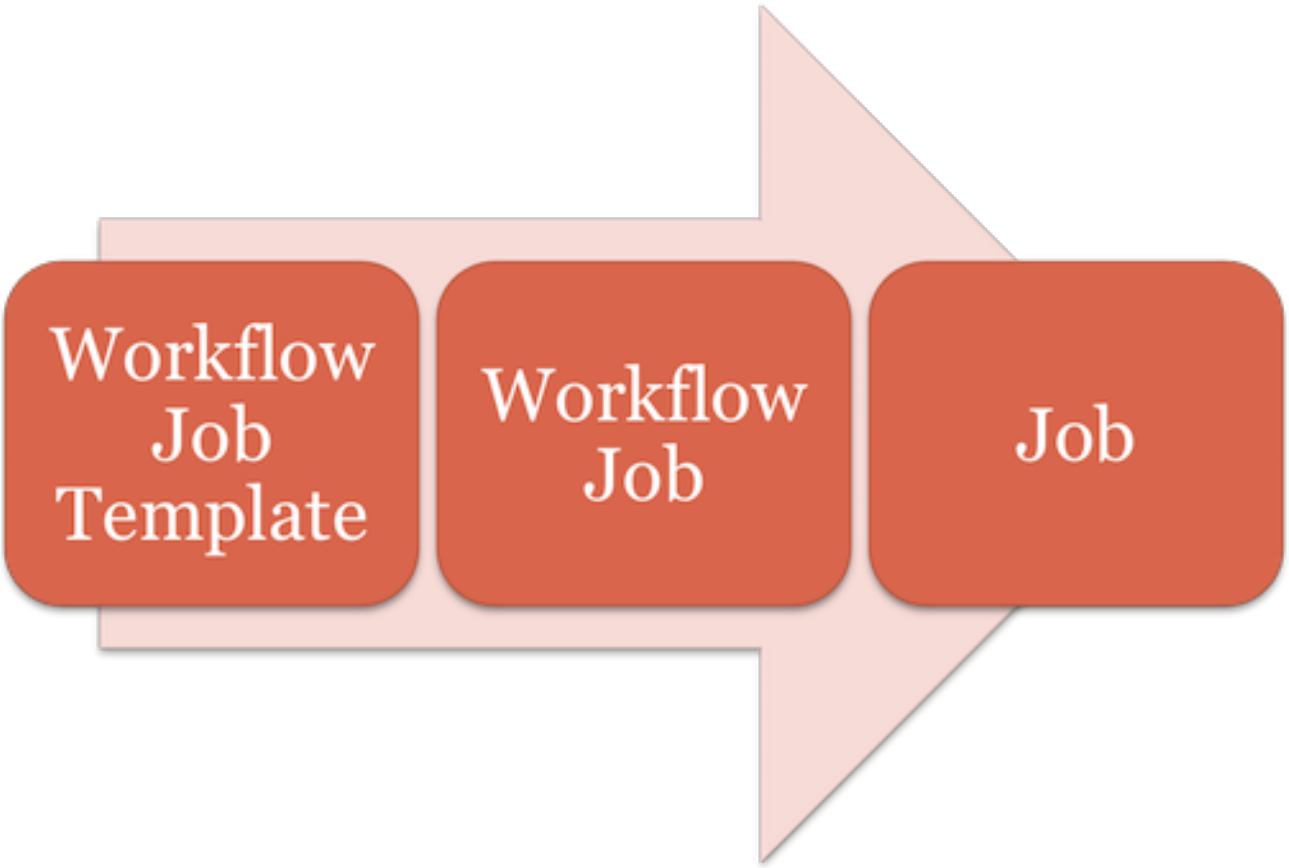


- 您可以有一个聚合的工作流，其中多个作业聚合成一个。在这种情况下，任何作业或所有作业都必须在下一个运行前完成，如下例所示：

  - 在本例中，自动化控制器会并行运行前两个作业模板。当它们都按照指定完成并成功时，第三个下游（聚合节点）会触发。

- 应用于工作流作业模板中的工作流节点的清单和问卷调查提示。

- 如果从 **API** 启动，运行 **get** 命令会显示一个警告列表并突出显示缺少的组件。下图演示了工作流作业模板的基本工作流：



- 可以同时启动多个工作流，并为启动它们的时间设置一个调度。您可以在工作流上设置通知，如作业完成时，类似于作业模板的通知。

注意

作业分片旨在水平扩展作业执行。

如果您在作业模板上启用作业分片，它会将清单划分成在启动时配置的分片数量。然后为每个分片启动一个作业。

如需更多信息，请参阅 作业分片 部分。

- 您可以构建递归工作流，但如果自动化控制器检测到错误，它会在嵌套工作流尝试运行时停止。

- 在子工作流的作业中收集到的工件会被传递给下游节点。

- 清单可以在工作流级别设置，或在启动时提示设置清单。
- 启动后，带有 **ask_inventory_on_launch=true** 的工作流中的所有作业模板都使用工作流级别清单。
- 不提示清单的作业模板会忽略工作流清单，并针对自己的清单运行。
- 如果工作流提示提供清单，调度和其他工作流节点可以提供清单。
- 在工作流聚合场景中，**set_stats** 数据会以未定义的方式合并，因此您必须设置唯一的键。

22.2. 工作流额外变量

工作流使用调查来指定工作流中 **playbook** 中使用的变量，称为 **extra_vars**。问卷调查变量与工作流作业模板上定义的 **extra_vars** 合并，并保存到工作流作业 **extra_vars** 中。在工作流内生成作业时，工作流作业中的 **extra_vars** 与作业模板变量合并。

工作流使用与作业模板相同的变量优先级行为（层次结构），但有三个额外变量除外。请参阅作业模板的额外变量部分中的 自动化控制器变量优先级层次结构。三个额外变量包括：

- 工作流作业模板额外变量
- 工作流作业模板问卷调查(defaults)
- 工作流作业启动额外变量

工作流中包含的工作流遵循相同的变量优先级，它们仅在特别提示或定义为调查的一部分时继承变量。

除了工作流 **extra_vars** 外，作为工作流的一部分运行的作业和工作流还可继承工作流中父作业的工件字典中的变量（也与分支中上游的祖先合并）。它们可由 **set_stats** **Ansible** 模块 定义。

如果在 **playbook** 中使用 **set_stats** 模块，您可以生成可供下游其他作业使用的结果。

Example

向用户通知用户集成运行是成功还是失败。在本例中，工作流中可以合并两个 **playbook**，用于操作工件传递：

- **invoke_set_stats.yml:** 工作流中的第一个 **playbook**：

```
---
- hosts: localhost
  tasks:
    - name: "Artifact integration test results to the web"
      local_action: 'shell curl -F "file=@integration_results.txt" https://file.io'
      register: result


    - name: "Artifact URL of test results to Workflows"
      set_stats:
        data:
          integration_results_url:  "{{ (result.stdout|from_json).link }}"
```

- **use_set_stats.yml: second playbook in the workflow:**

```
---
- hosts: localhost
  tasks:
    - name: "Get test results from the web"
      uri:
        url: "{{ integration_results_url }}"
        return_content: true
      register: results


    - name: "Output test results"
      debug:
        msg: "{{ results.content }}"
```

set_stats 模块处理此工作流，如下所示：

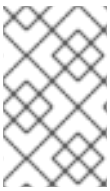
1. 集成结果的内容将上传到 **web**。

2.

通过 **invoke_set_stats playbook**，调用 **set_stats** 将上传的 **integration_results.txt** 的 **URL** 打包到 **Ansible** 变量 **"integration_results_url"**。

3.

工作流中的第二个 **playbook** 使用 **Ansible** 额外变量 **"integration_results_url"**。它使用 **uri** 模块调用 **Web**，以获取上一作业模板作业上传的文件内容。然后，它会输出获取的 文件的内容。



注意

要使工件正常工作，请在 **set _stats** 模块中保留每个**_host = False** 的默认设置。

22.3. 工作流状态

工作流可以具有以下状态（无**"Failed"**状态）：

- **Waiting**
- **Running**
- **Success (finished)**
- **Cancel**
- 错误
- **Failed**

在工作流方案中，取消作业会取消分支，而取消工作流作业会取消整个工作流。

22.4. 基于角色的访问控制

要编辑和删除工作流作业模板，您必须具有管理员角色。要创建工作流作业模板，您必须是机构管理员或系统管理员。但是，您可以运行包含您没有权限的作业模板的工作流作业模板。系统管理员可以创建空

白工作流，然后将 **admin_role** 授予低级用户，之后他们可以委派更多访问权限并构建图形。您必须具有对 作业模板的执行 访问权限，才能将其添加到工作流作业模板中。

您还可以执行其他任务，如复制复制或重新启动工作流，具体取决于向用户授予哪些权限。在重新启动或制作副本前，您必须对工作流中使用的所有资源（如作业模板）具有权限。

如需更多信息，请参阅基于角色的访问控制。

有关执行本节所述任务的更多信息，请参阅管理指南。

第 23 章 工作流作业模板

工作流作业模板将一系列不同资源链接，这些资源将属于发布过程一部分的完整作业集合作为一个单元来跟踪。这些资源包括以下内容：

- 作业模板
- 工作流作业模板
- 项目同步
- 清单源同步

Templates 列表视图显示当前可用的工作流和作业模板。默认视图为折叠状态**(Compact)**，显示模板名称、模板类型和使用该模板运行的作业的状态。您可以单击每个条目旁边的箭头，以展开和查看更多信息。此列表按名称按字母顺序排序，但您可以根据其他条件排序，或者根据模板的不同字段和属性进行搜索。在此屏幕中，您可以启动

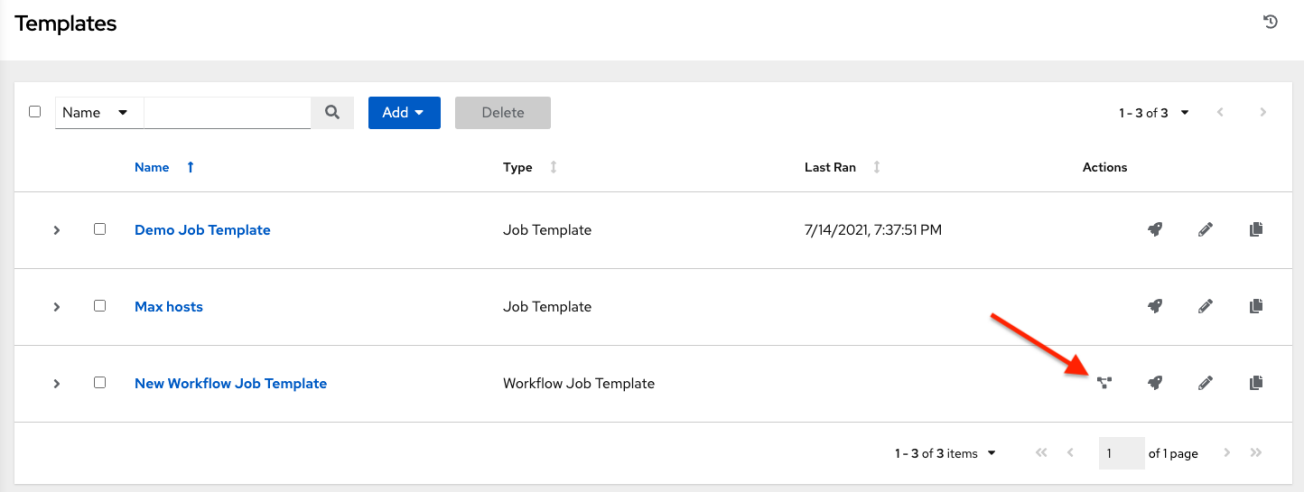
，**编辑**

，并复制

工作流作业模板。

只有工作流模板带有工作流可视化工具

图标作为访问工作流编辑器的快捷方式。



注意

工作流模板可用作另一个工作流模板的构建块。您可以通过在工作流模板中设置多个设置来启用 **Prompt on Launch**，您可以在工作流作业模板一级编辑该设置。它们不会影响在各个工作流模板级别分配的值。具体步骤请查看 工作流可视化工具 部分。

23.1. 创建工作流模板

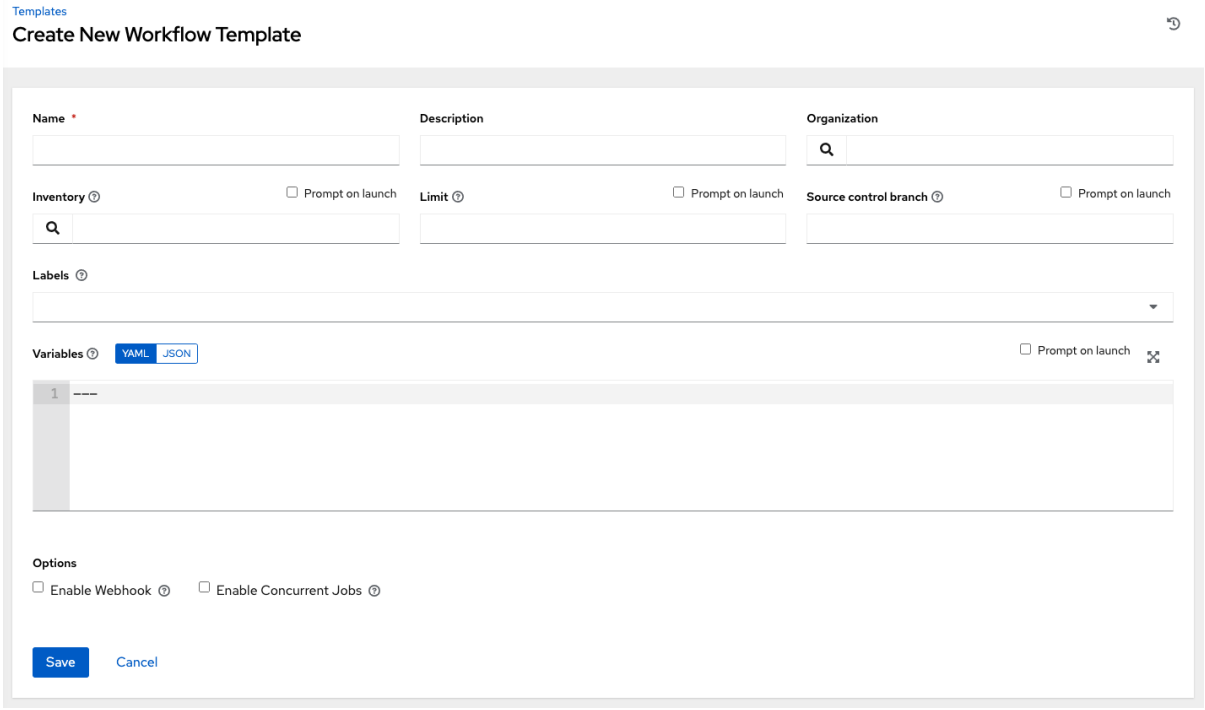
要创建新工作流作业模板，请完成以下步骤：

重要

如果将限制设置为工作流模板，则不会传递给作业模板，除非您检查了 **Prompt on launch** 的限制。如果运行的 **playbook** 是必需的，这可能会导致 **playbook** 失败。

流程

1. 在 **Templates** 列表视图中，从 **Add** 列表中选择 **Add workflow template**。



2. 在以下字段中输入相关信息：

注意

如果某个字段选择了 **Prompt on launch** 复选框，则启动工作流模板，或者在另一个工作流模板中使用工作流模板，则会提示您输入该字段的值。大多数提示的值将覆盖作业模板中设置的任何值。下表中会记录例外情况。

字段	选项	启动时提示
Name	输入作业的名称。	N/A
描述	根据需要输入任意描述（可选）。	N/A
机构（Organization）	从可供登录的用户可用的机构中选择用于此模板的组织。	N/A
清单（Inventory）	（可选）从登录的用户可用的清单中选择要与此模板搭配使用的清单。	是

字段	选项	启动时提示
限制	用于进一步限制受 playbook 管理或影响的主机列表的主机模式。您可以通过冒号(:)分隔多个模式。与核心 Ansible 一样： ● a:b 表示"在组 a 或 b 中" ● a:b:&c 表示"在 a 或 b 中，但必须在 c 中"。 ● 答：！b 表示"在 a 中，一定不要在 b 中" 如需更多信息，请参阅 Ansible 文档中的 Patterns：以主机和组为目标。	是 如果选择，即使提供了默认值，也会在启动时提示您选择一个限制。
源控制分支	为工作流选择分支。此分支应用于提示分支的所有工作流作业模板节点。	是
标签	● （可选）提供描述此工作流作业模板的标签，如 **dev** 或 **test**。使用标签对显示中的工作流作业模板和完成的作业进行分组和过滤。 ● 标签在添加到工作流模板时创建。标签使用工作流模板中提供的项目与单个机构关联。如果机构的成员具有编辑权限（如 admin 角色），则机构成员可以在工作流模板上创建标签。 ● 保存作业模板后，标签会显示在工作流作业模板 **详情视图中**。 ● 标签仅应用于工作流模板，而不是工作流中使用的作业模板节点。 ● 选择标签旁边的✖将其删除。删除标签后，它不再与该特定作业或作业模板关联，但它与引用它的任何其他作业关联。	是 如果选择，即使提供了默认值，也会在启动时提示您提供附加标签（如果需要）。- 您无法删除现有标签，选择✖仅会删除新添加的标签，而不是现有的默认标签。

字段	选项	启动时提示
变量	• 向 playbook 传递额外的命令行变量。 这是 ansible-playbook 的"-e"或"-extra-vars"命令行参数，记录在 Ansible 文档中，控制 Ansible 的行为：优先级规则. - 使用 YAML 或 JSON 提供键或值对。这些变量具有最大优先级值，并覆盖其他位置指定的其他变量。以下是一个值： **git_branch: production** **release_version: 1.5**	是 如果要能够在调度中指定 extra_vars，您必须在工作流作业模板中为 Variables 选择 **Prompt on launch**，或者在作业模板上启用问卷调查。那些回答的问卷调查问题将变为 **extra_vars**。有关额外变量的更多信息，请参阅 额外变量。
作业标签	键入并选择 **Create** 下拉菜单来指定应运行 playbook 的哪个部分。有关更多信息和示例，请参阅 Ansible 文档中的 标签。	是
跳过标签	键入并选择 **Create** 下拉菜单来指定要跳过的 playbook 的某些任务或部分。有关更多信息和示例，请参阅 Ansible 文档中的 标签。	是

3. 如果需要，指定启动此模板的以下选项：

- 选中 **Enable Webhooks** 以打开与用于启动工作流作业模板的预定义 **SCM** 系统 **Web** 服务进行接口的功能。**GitHub** 和 **GitLab** 是支持的 **SCM** 系统。
  - 如果您启用 **Webhook**，会显示其他字段，提示输入更多信息：
    - **Webhook Service** ：选择要从哪个服务侦听 **Webhook**。
    - **Webhook** 凭证 ：（可选）提供 **GitHub** 或 **GitLab** 个人访问令牌**(PAT)**作为凭证，用来向 **webhook** 服务发回状态更新。如需更多信息，请参阅 凭证类型 来创建。
  - 当您点 **Save** 时，会填充其他字段，工作流可视化工具会自动打开。

- **Webhook URL** ：自动填充将 **POST** 请求发送到的 **Webhook** 服务的 **URL**。

- **Webhook Key:** 生成共享 **secret**，供 **Webhook** 服务用来签署发送到自动化控制器的有效负载。您必须在 **Webhook** 服务上的设置中配置此功能，以便在自动化控制器中接受此服务的 **Webhook**。有关设置 **Webhook** 的更多信息，请参阅使用 **Webhook**。

选中 **Enable Concurrent Jobs** 以允许同时运行此工作流。如需更多信息，请参阅自动控制器容量确定和作业影响。

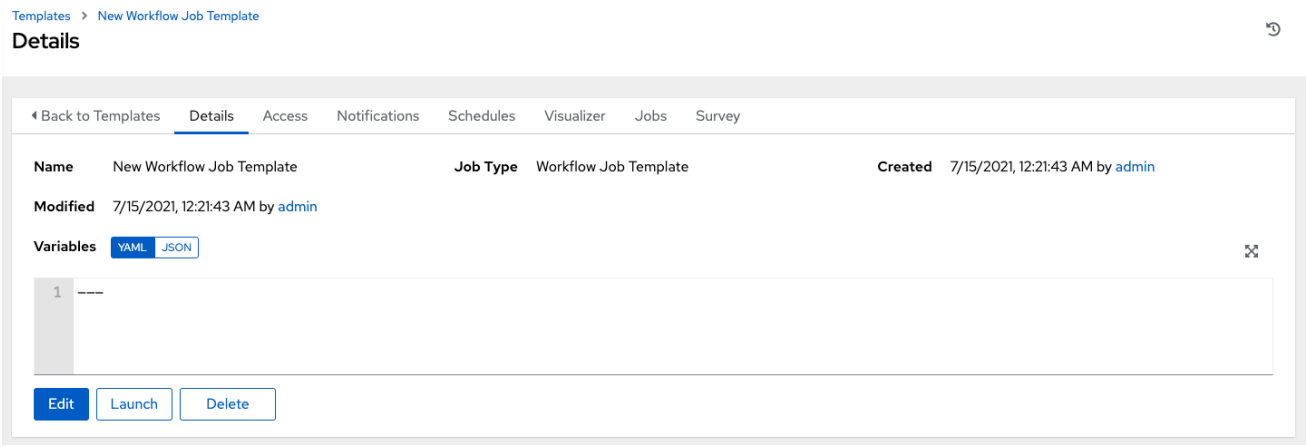
4. 配置完工作流模板后，点 **Save**。

保存模板会退出工作流模板页面，并打开工作流可视化工具，以便您构建工作流。如需更多信息，请参阅 工作流可视化工具 部分。否则，请选择以下方法之一：

- 关闭工作流可视化工具，以返回到新保存的模板的 **Details** 选项卡。您可以完成以下任务：
  - 查看、编辑、添加权限、通知、计划和调查
  - 查看完成的作业
  - 构建工作流模板
- 点 **Launch** 启动工作流。

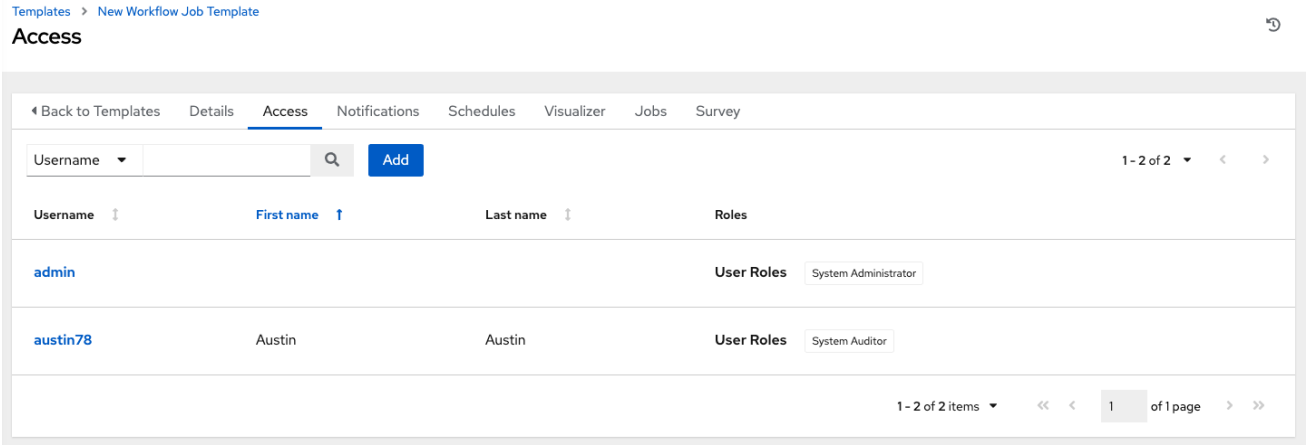
注意

在启动前保存模板，或者 **Launch** 仍被禁用。**Notifications** 选项卡只有在您保存模板后才会显示。



23.2. 使用权限

点 **Access** 选项卡来查看、获取、编辑和删除用户和团队成员的相关权限。



点 **Add** 以为此工作流模板创建新权限，按照提示相应地分配它们。

23.3. 使用通知

有关在工作流作业模板中使用通知的详情，请参考使用通知。

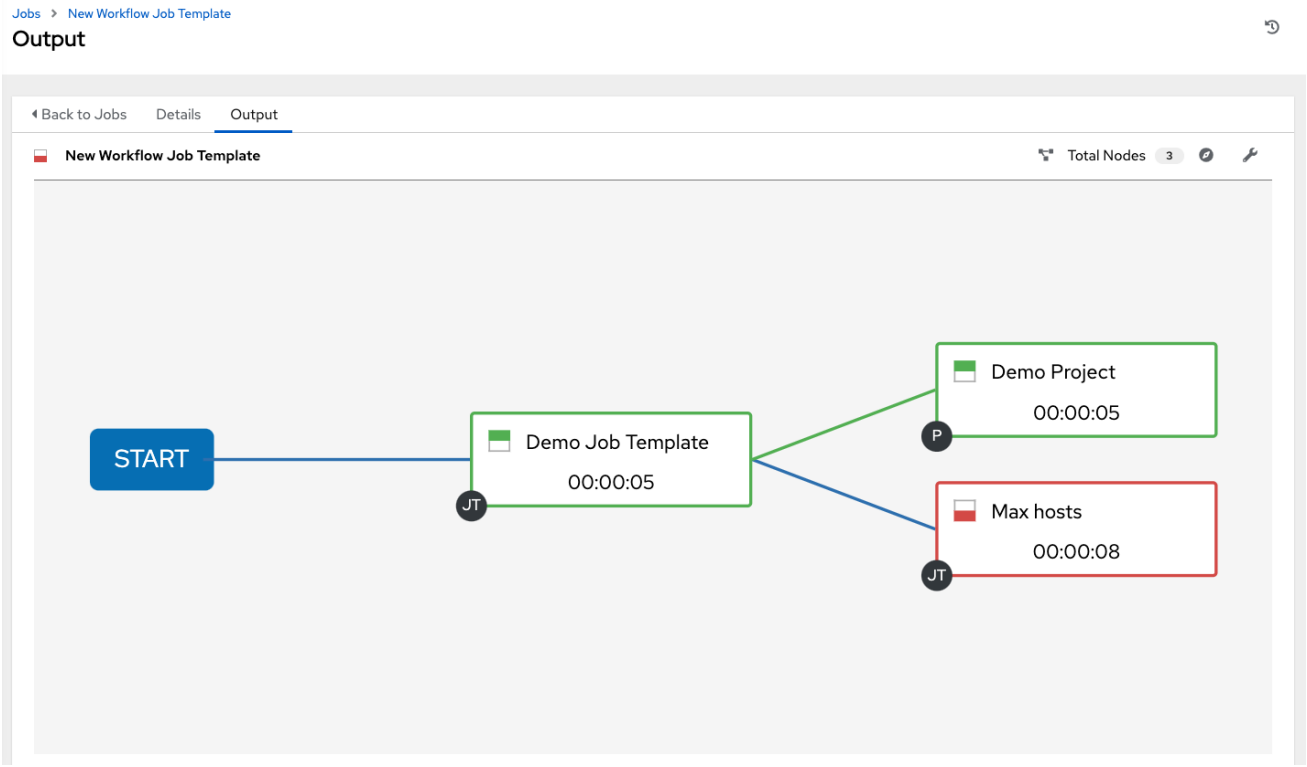
23.4. 查看完成的工作流作业

Jobs 选项卡提供已运行的作业模板列表。点每个作业旁的

图标查看每个作业的详情。

从此视图中，您可以点作业 **ID**，工作流作业的名称并查看其图形表示。以下示例显示了工作流作业的

作业详情：



节点使用标签标记，以帮助您识别它们。如需更多信息，请参阅 工作流可视化工具 部分中的图例。

23.5. 调度工作流作业模板

选择 **Schedules** 选项卡，以访问特定工作流作业模板的计划。

有关调度工作流作业模板运行的更多信息，请参阅 调度作业模板 部分。

如果嵌套工作流中使用的工作流作业模板有问卷调查，或者为 **inventory** 选项选择了 **Prompt on Launch**，则调度表单上的 **SAVE** 和 **CANCEL** 选项旁边会显示 **PROMPT** 选项。单击 **PROMPT** 以显示可选的 **INVENTORY** 步骤，您可以在其中提供或删除清单，或者在没有任何更改的情况下跳过这一步。

23.6. 工作流作业模板中的问卷调查

包含 运行 或检查 的作业类型 的工作流提供了一种在工作流作业模板创建或编辑屏幕中设置问卷调查的方法。

有关作业调查的更多信息，包括如何在工作流作业模板中创建问卷调查和可选问卷调查问题，请参阅

作业模板中的调查 部分。

23.7. 工作流可视化工具

工作流可视化工具提供了一种图形方式，用于将作业模板、工作流模板、项目同步和清单同步链接到构建工作流模板。在构建工作流模板前，请参阅 工作流 部分，以了解与父、子和同级节点上各种场景相关的注意事项。

23.7.1. 构建工作流

您可以设置以下两个或更多节点类型的组合来构建工作流：

- 模板（作业模板或工作流作业模板）
- 项目同步
- 清单同步
- 批准

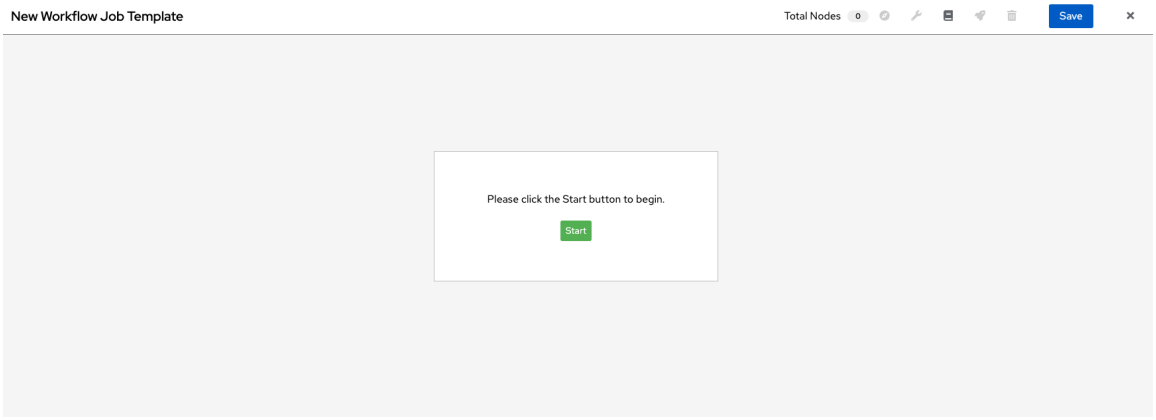
每个节点都由一个 **rectangle** 表示，而关系及其关联的边缘类型由连接它们的行（或链接）表示。

流程

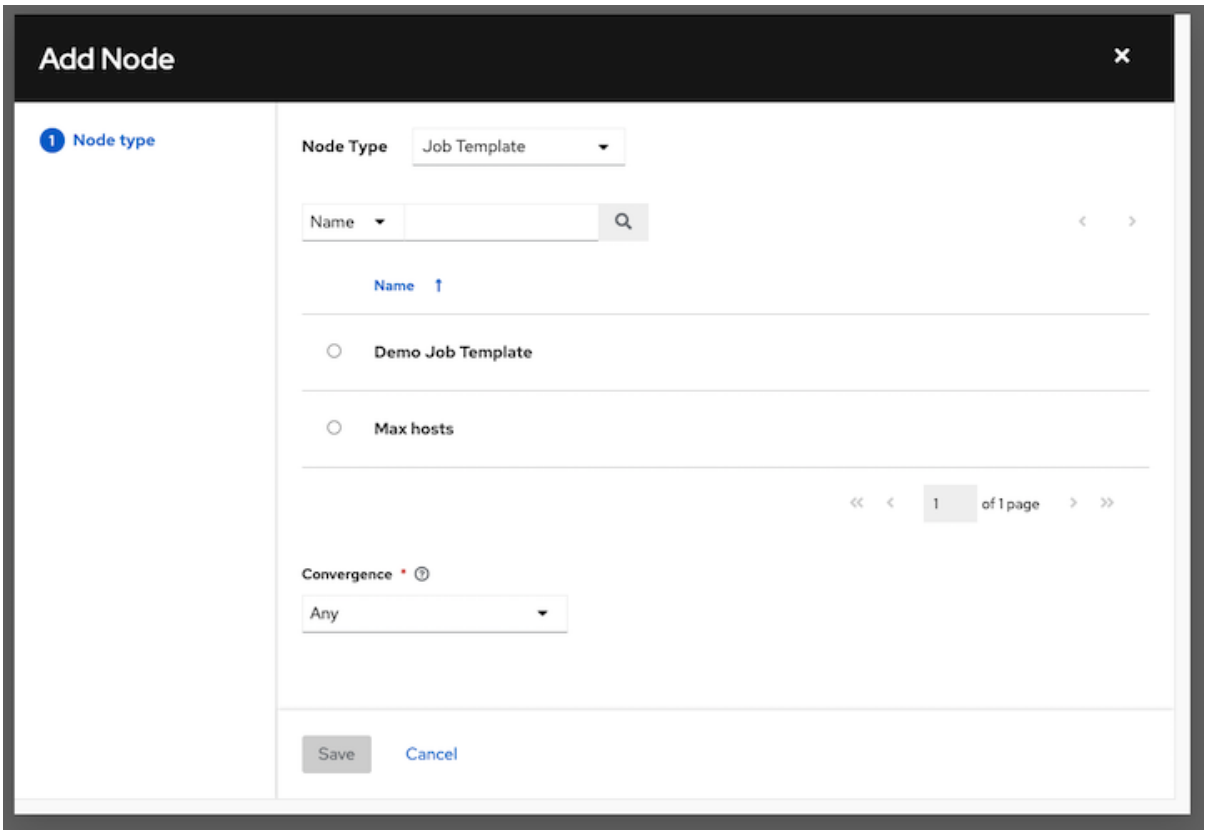
1. 要启动工作流可视化工具，请使用以下方法之一：
   a. 在导航面板中，选择 **Resources** → **Templates**。
      i. 在 **Details** 选项卡中，选择工作流模板，点 **Edit**。
      ii. 选择 **Visualizer** 选项卡。
   b.

在 **Templates** 列表中点

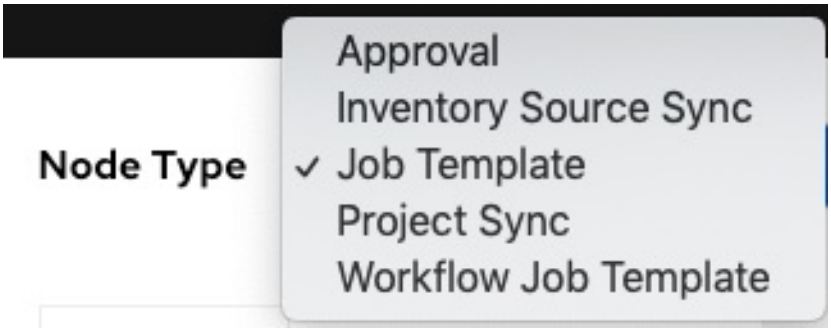
图标。



2. 单击 **Start** 以显示要添加到工作流的节点列表。



3. 从 **Node Type** 列表中，选择要添加的节点类型：



- 如果选择了 **Approval node**，请参阅 **Approval nodes** 以了解更多信息。

  选择节点会提供与其关联的可用的有效选项。

  注意

  如果您在填充工作流图形时选择了没有默认清单的作业模板，则会使用父工作流的清单。虽然作业模板中不需要凭证，但如果具有需要密码的凭证，则无法为工作流选择作业模板，除非凭证被提示的凭证替代。

4. 当您选择节点类型时，工作流开始构建，且您必须指定要为所选节点执行的操作类型。此操作也称为边缘类型。

5. 如果节点是根节点，则边缘类型默认为 **Always**，不可编辑。对于后续节点，您可以选择以下场景（类型）之一以应用到每个节点：

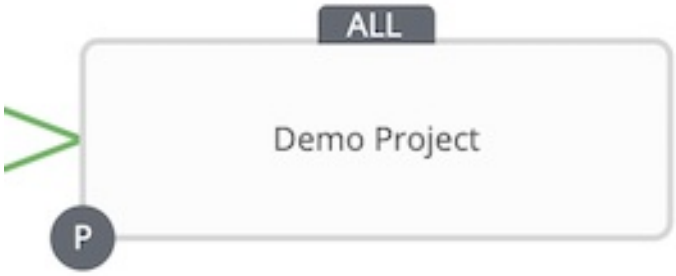
   - **Always:** 无论成功或失败都继续执行。
   - 成功后 ：成功完成后，执行下一个模板。
   - **On Failure:** 在失败时执行不同的模板。

6. 如果节点是 **Convergence** 字段中的聚合节点，请选择节点的行为：

   - 任何 是默认行为，允许任何节点按照指定完成，然后再触发下一个聚合节点。只要一个父状态满足其中一个运行条件，就会 运行任何 子节点。任何节点都 需要所有节点都完成，但只有一个节点必须以预期结果完成。

- 选择 **All** 以确保所有节点都按指定完成，然后再聚合并触发下一个节点。所有* 节点的目的是确保每个父节点都满足其预期结果，以运行子节点。工作流检查以确保每个父项都如预期运行子节点。否则，它不会运行子节点。

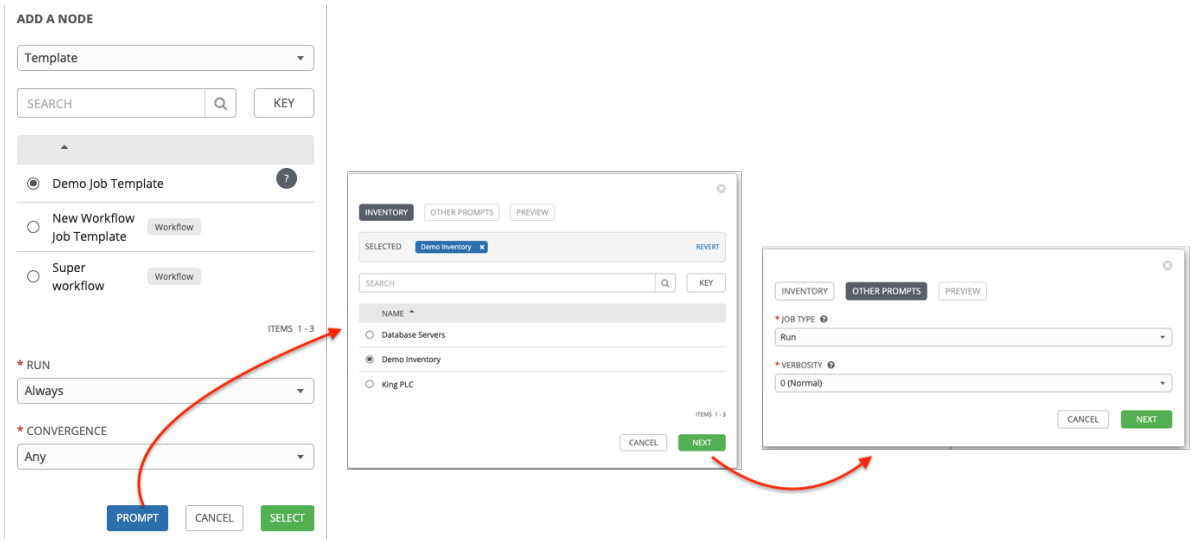
  如果选择，在图形视图中将节点标记为 **ALL** ：



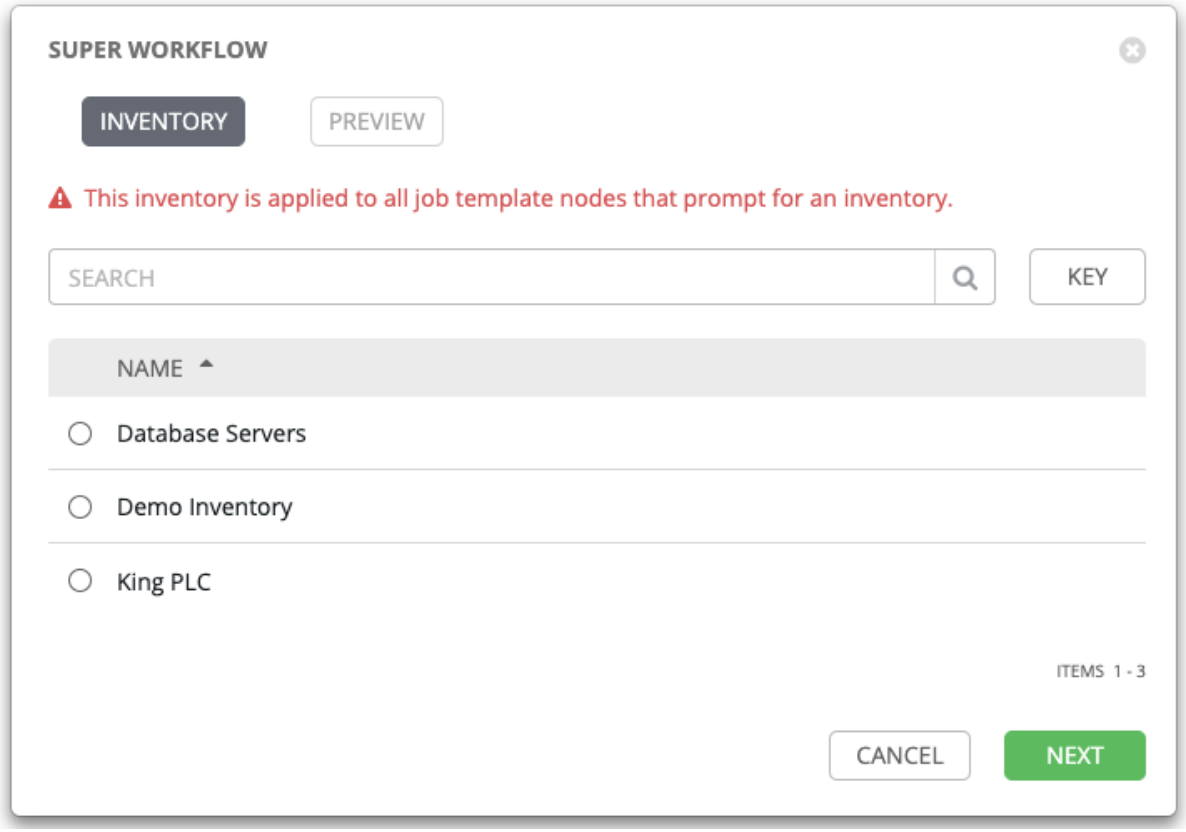
注意

如果节点是根节点，或者没有与其聚合的节点，设置 **Convergence** 规则不会应用，因为它的行为是由触发节点的操作决定的。

7. 如果工作流中使用的作业模板为其任何参数选择了 **Prompt on Launch**，则会出现 **PROMPT** 选项，允许您在节点级别更改这些值。使用向导更改每个标签页中的值，然后点击 **Preview** 选项卡中的 **Confirm**。



如果工作流中使用的工作流模板为清单选项选择了 **Prompt on Launch**，请使用向导在提示符处提供清单。如果父工作流有自己的清单，它会覆盖此处提供的任何清单。



注意

对于带有提示详情但没有默认值的工作流作业模板，您必须在启用 **Select** 选项前提供这些值。

以下两个情况会禁用 **SELECT** 选项，直到 **PROMPT** 选项提供值前：

a. 当您在工作流作业模板中选择 **Prompt on Launch** 复选框时，但不提供默认值。

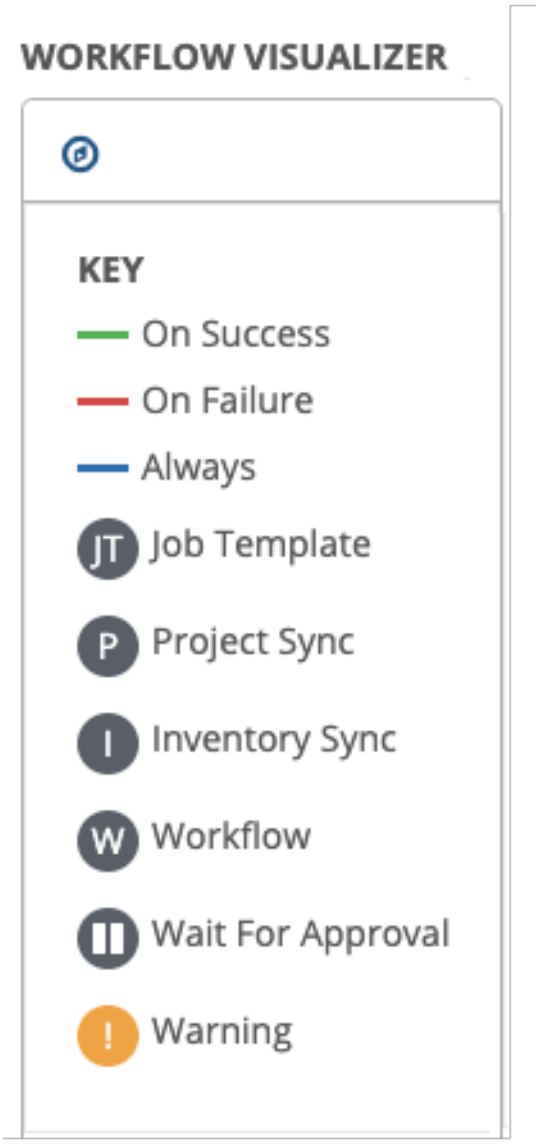
b. 当您创建需要但不提供默认答案的问卷调查问题时。

但是，凭证并不如此。创建工作流节点时不允许在启动时需要密码的凭证，因为在创建节点时必须提供启动该节点所需的所有内容。如果您在工作流作业模板中提示输入凭证，则无法选择在自动化控制器中需要密码的凭证。

当提示向导关闭时，您还必须单击 **SELECT**，以在该节点上应用更改。否则，您所做的任何更改都会恢复到作业模板中设置的值。

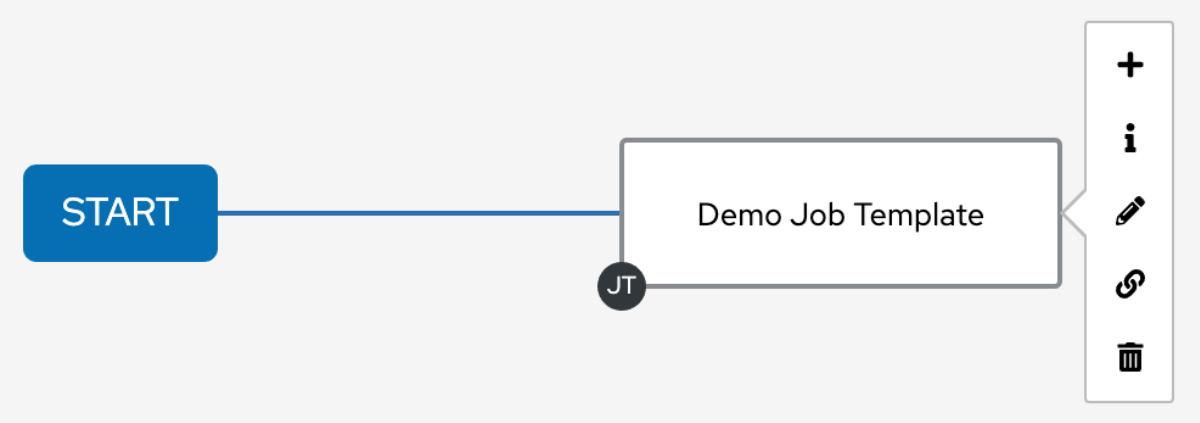
创建节点时，会使用其作业类型进行标记。与每个工作流节点关联的模板，会根据所选运行场景运行。点 **compass (**

)图标显示每个运行场景及其作业类型的图例。



8.

将鼠标悬停在节点上以添加另一个节点，查看节点信息、编辑节点详情、编辑现有链接或删除所选节点：



9. 添加或编辑节点后，单击 **SELECT** 以保存任何修改并将其呈现在图形视图中。有关构建工作流的可能方法，请参阅 构建节点场景。

10. 构建工作流作业模板后，点 **Save** 保存整个工作流模板并返回到新的工作流作业模板详情页面。

重要

点 **Close** 不会保存您的工作，而是关闭整个工作流可视化工具，因此您必须再次启动。

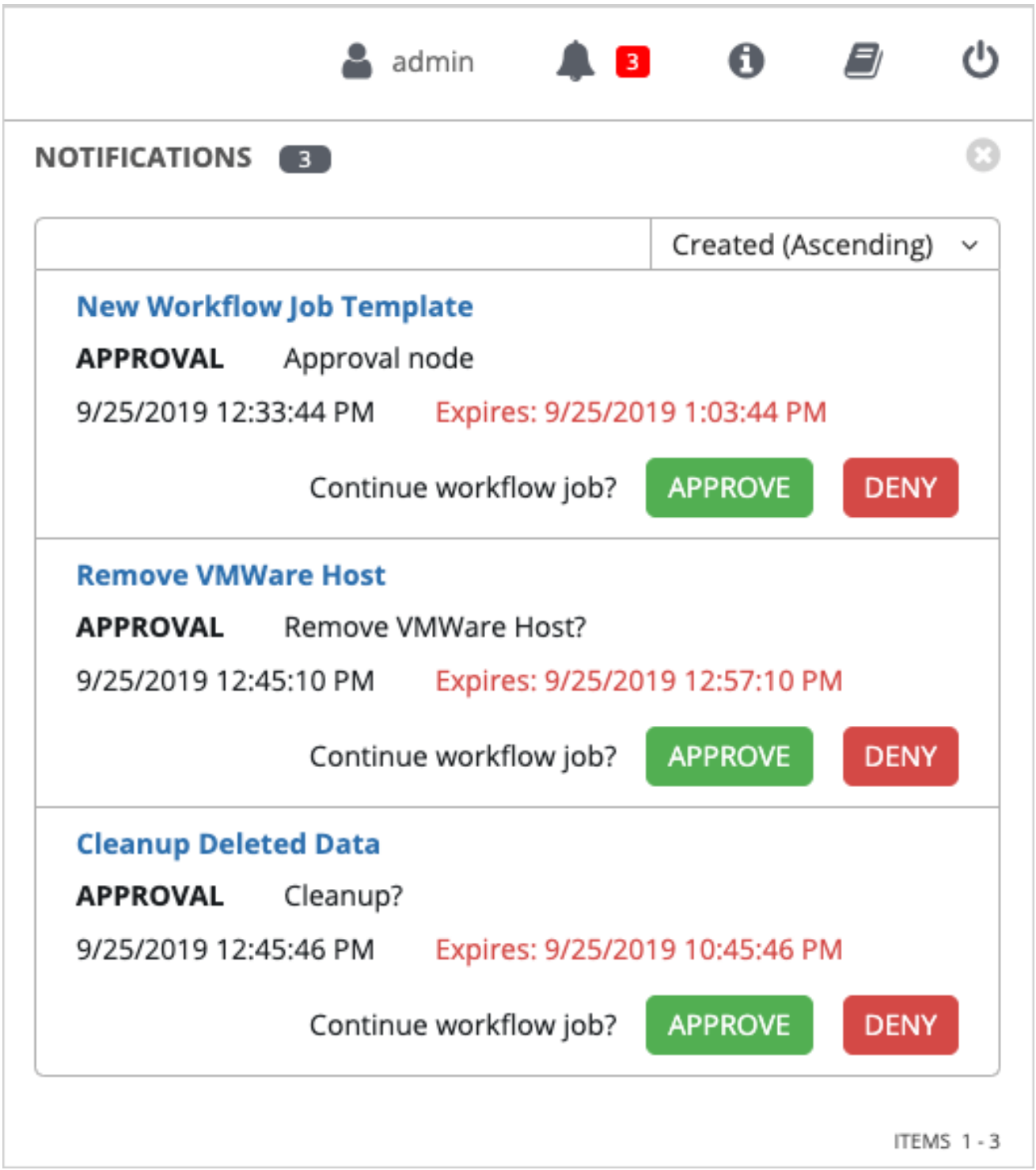
23.7.2. 批准节点

选择 批准节点 需要您的干预才能推进工作流。此功能为在 **playbook** 间暂停工作流，以便您可以授予继续工作流中的下一个 **playbook** 的批准。这可让用户指定的时间干预，但也允许您尽快继续，而无需等待其他触发器。

超时的默认值为 **none**，但您可以指定请求过期前的时间长度，并自动拒绝。选择并提供批准节点的信息后，它会显示在图形视图中，其中包含一个暂停图标。

批准者是满足以下条件的任何人：

- 可以执行包含批准节点的工作流作业模板的用户。
- 具有机构管理员或以上特权的用户（用于与该工作流作业模板关联的机构）。
- 在该特定工作流作业模板中明确为其分配了 **Approve** 权限的用户。



如果待处理的批准节点没有在指定的时间限制内批准（如果分配了过期时间），或者被拒绝，则它们

被标记为 **"timed out"** 或 **"failed"**，并移到下一个 **"on fail node"** 或 **"always node"**。如果批准，则使用 **"on success"** 路径。如果您试图将 **API** 中的 **POST** 发布到已批准、被拒绝或超时的节点，则会显示错误消息通知您此操作冗余，且不会执行进一步的步骤。

下表显示了在批准工作流中允许的各种权限级别：

SCOPE	ROLE	CREATE WORKFLOW APPROVAL	GRANT APPROVAL	VIEW WORKFLOW APPROVAL	APPROVE/DENY	VIEW WORKFLOW APPROVAL IN ACTIVITY STREAM
Organization	Organization Admin	Yes	Yes	Yes	Yes	Yes
Organization Workflow Job Template	Workflow Admin	Yes	Yes (*)	Yes	Yes	Yes
	Workflow Executor	No	No	Yes	No	Yes
	Workflow Approver	No	No	Yes	Yes	Yes
	Read on Workflow	No	No	Yes (**)	No	Yes (***)
System	System Admin	Yes	Yes	Yes	Yes	Yes
	System Auditor	No	No	Yes	No	Yes
Random user in Organization		No	No	No	No	No
Random user outside Organization		No	No	No	No	No

* Exception: A User with WF Admin permission at the organization level would not be able to grant approval.

** Exception: A User with Read on WF permission at the organization level would not be able to view WF approvals.

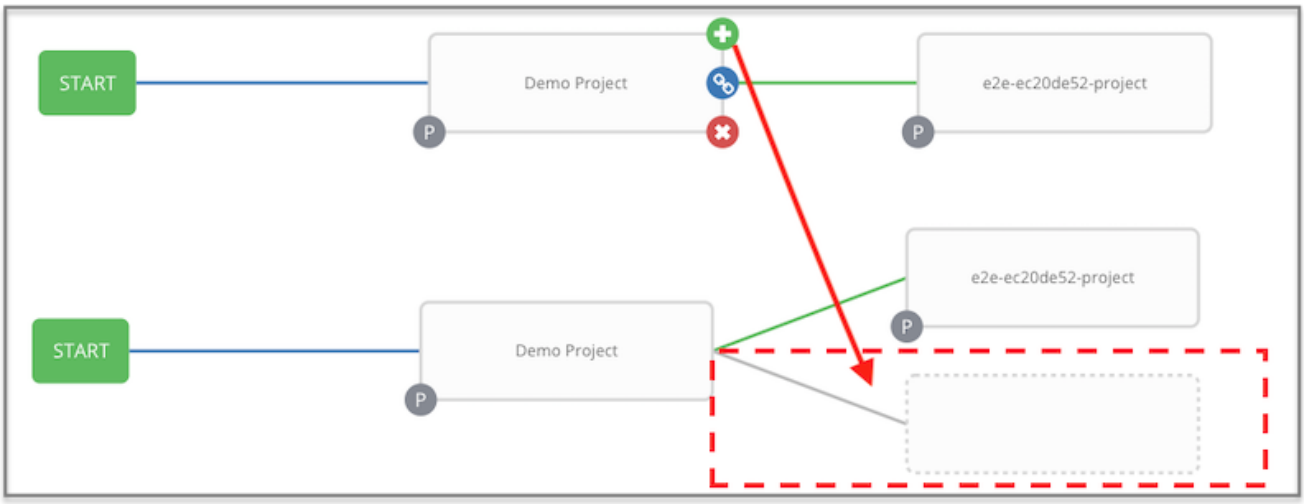
*** Exception: A User with Read on WF permission at the organization level would not be able to view approval jobs in the Activity Stream.

23.7.3. 构建节点场景

了解如何在以下场景中管理节点。

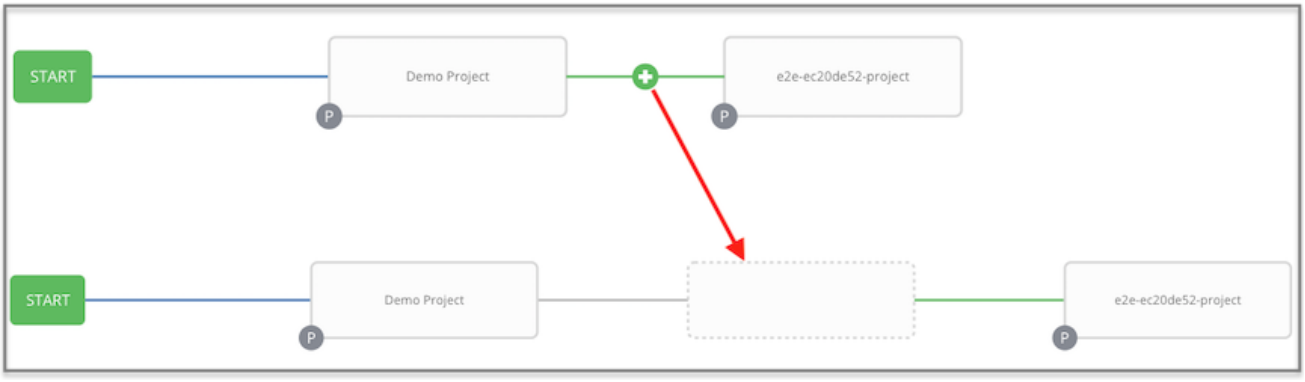
流程

- 点击父节点上的(
  +
)图标添加同级节点：

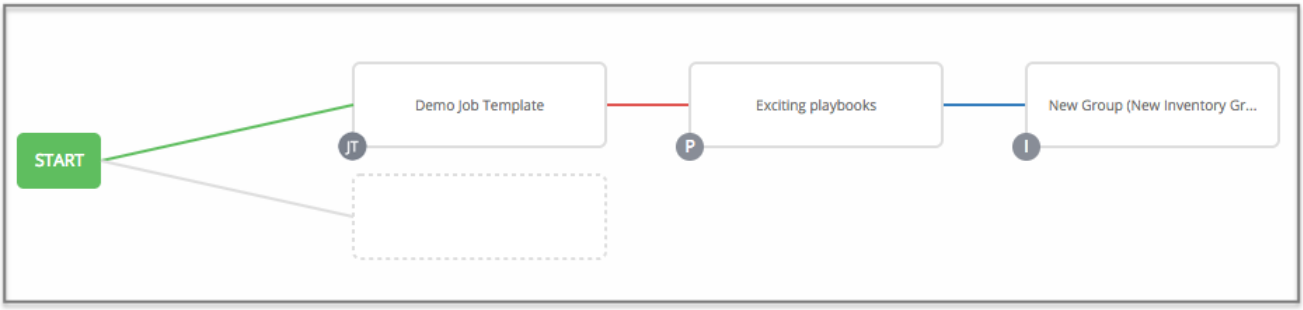


- 将鼠标悬停在连接两个节点的行上，并点击加号(
  +
)在节点间插入另一个节点。点击加号(

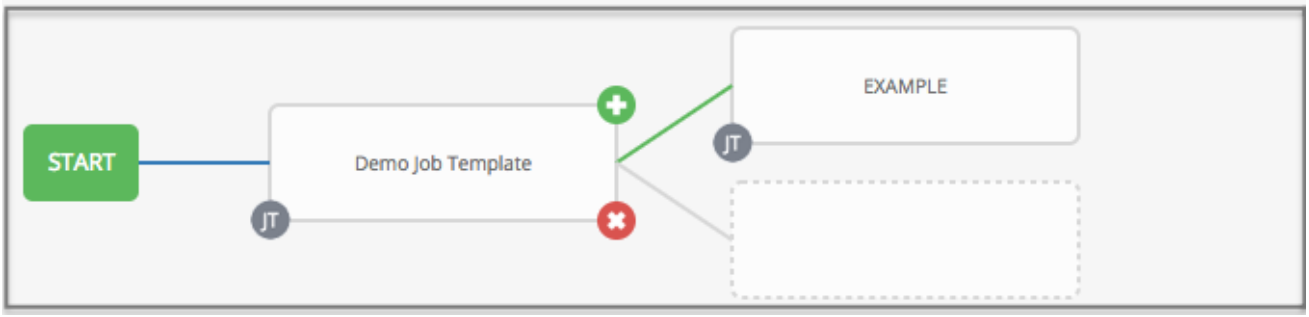
+)图标在两个节点之间自动插入节点：



- 再次单击 **START**，以添加根节点来描述分割场景：



- 在您要创建分割场景的任何节点上，将鼠标悬停在分割场景开始的节点上，然后点击加号(+)图标。这会从同一父节点添加多个节点，创建同级节点：

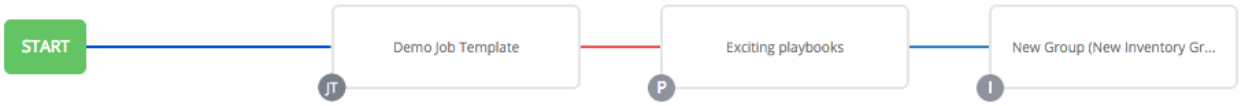


注意

在添加新节点时，**PROMPT** 选项也适用于工作流模板。工作流模板提示输入清单和问卷调查。

- 您可以使用以下方法之一撤销最后一个插入的节点：
  - 在不做出选择的情况下点另一个节点。
  - 单击 取消。

以下示例工作流包含作业模板启动的所有三种类型的作业。如果无法运行，您必须保护同步任务。无论它是否失败还是成功，都继续执行清单同步作业：

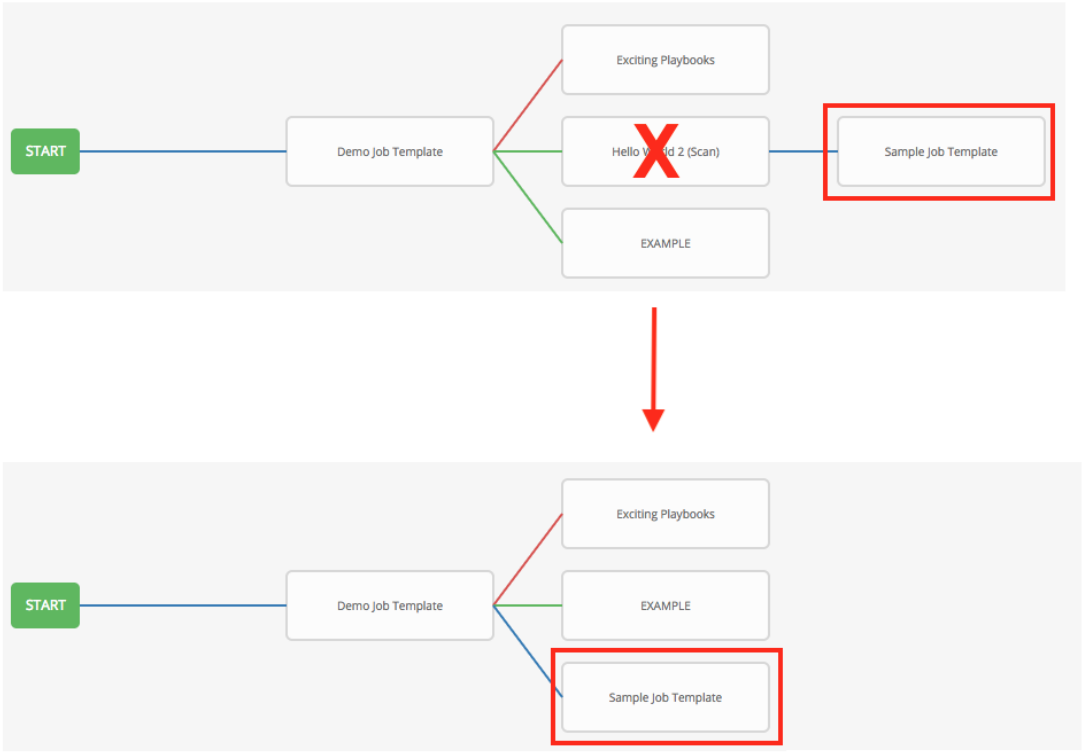


点击 **compass (**

)图标来识别与图形描述相关的符号和颜色的含义，以引用该密钥。

注意

如果您在带有一组具有不同边缘类型的同级节点的工作流中删除附加有后续节点的节点，则附加节点会自动加入同级节点组并保留其边缘类型：

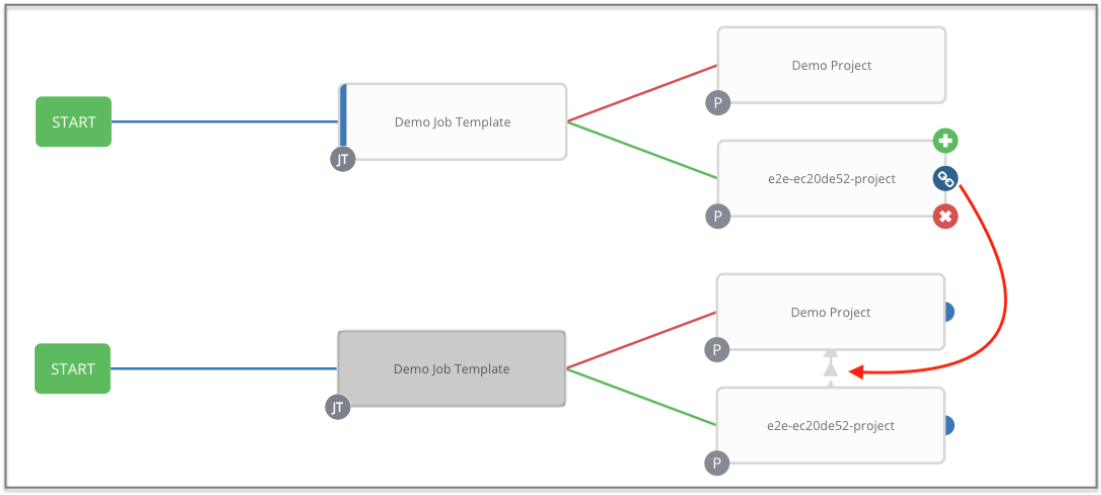


23.7.4. 编辑节点

流程

- 使用以下方法之一编辑节点：
  - 如果要编辑节点，点您要编辑的节点。窗格中显示当前的选择。进行更改并点击 **Select** 将它们应用到图形视图。
  - 要编辑现有链接的边缘类型(成功，失败，始终为)，请单击 链接。窗格中显示当前选择。进行更改并点 **Save** 将它们应用到图形视图。
  - 点击每个节点中出现的链接(
)图标，将一个新链接从一个节点添加到另一个节点。这样做会突出显示可以链接到的节点。这些选项通过点号行来指示。无效的选项由禁用框（节点）表示，否则会生成无效的

链接。以下示例显示 **e2e-ec20de52-project** 链接的 **Demo Project** 作为可能的选项，由箭头表示：



- 要删除链接，请单击链接并单击 **UNLINK**。只有在目标或子节点有多个父节点时，此选项才会出现在窗格中。所有节点都必须始终链接到另一个其他节点，因此您必须在删除旧链接前创建新链接。

- 使用以下方法之一编辑工作流图的视图：

  - 点设置图标缩放、**pan** 或重新组成视图。

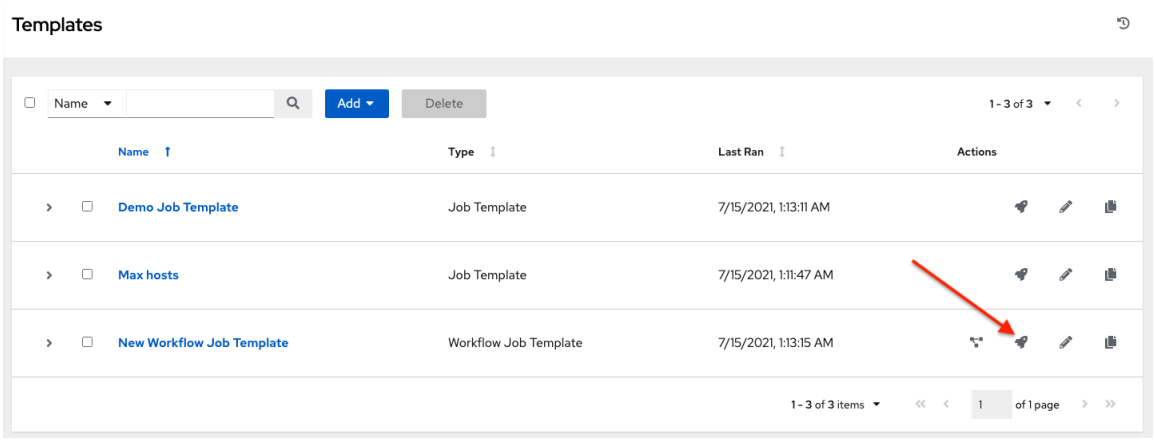
  - 拖动工作流图以在屏幕上进行重新定位，或使用鼠标上的滚动来缩放。

23.8. 启动工作流作业模板

流程

- 使用以下方法之一启动工作流作业模板：

  - 在导航面板中，选择 **Resources** → **Templates** 并点作业模板旁的 **Launch** ：



- 点击您要启动的工作流作业模板的 **Details** 选项卡中的 **Launch**。

在启动时，为工作流作业模板添加的变量会在自动化控制器中自动添加，以及工作流作业模板和问卷调查中设置的额外变量。

与工作流上的批准相关的事件显示在活动流(
)中，其中包含有关批准请求的详细信息（若有）。

23.9. 复制工作流作业模板

使用自动化控制器，您可以复制工作流作业模板。当您复制工作流作业模板时，它不会复制任何关联的调度、通知或权限。用户必须由用户或系统管理员创建工作流模板副本重新创建调度和通知。复制工作流模板的用户被授予管理员权限，但没有将权限分配给工作流模板。

流程

1. 使用以下方法之一打开您要复制的工作流作业模板：
   - 在导航面板中，选择 **Resources** → **Templates**。
   - 在工作流作业模板 详情视图中，向下滚动以从模板列表中访问它。
     - 点复制(
)图标。

此时会打开一个新模板，其中包含您复制的模板名称和时间戳：

2. 选择复制的模板，并将 **Name** 字段的内容替换为新名称，并提供或者修改其他字段中的条目以完成此模板。

3. 点击 **Save**。

注意

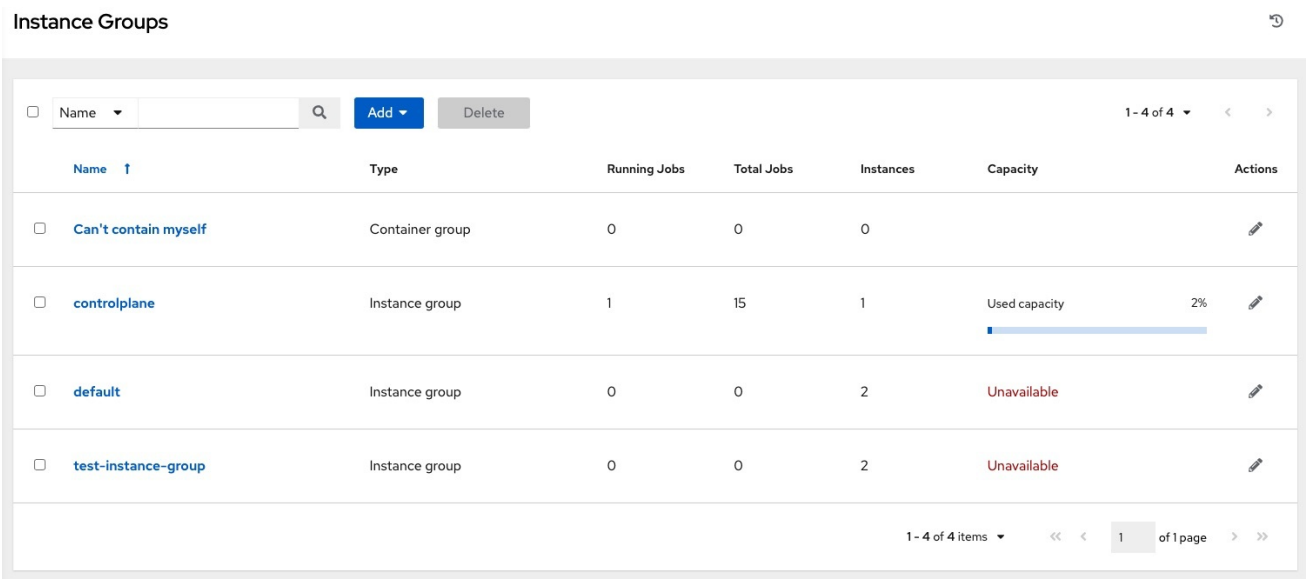
如果资源具有您没有正确权限级别的相关资源，则无法复制该资源。例如，如果项目使用了当前用户仅具有 **Read** 访问权限的凭据。但是，对于工作流作业模板，如果其任何节点使用未授权的作业模板、清单或凭证，则工作流模板仍然可以复制。但是在复制的工作流作业模板中，工作流模板节点中的对应字段不存在。

23.10. 工作流作业模板额外变量

如需更多信息，请参阅 **Extra variables** 部分。

第 24 章 管理实例组

实例组可让您在集群环境中对实例进行分组。策略指定实例组的行为方式以及任务的执行方式。以下视图显示基于策略算法的容量级别：



其他资源

- 有关与实例组关联的策略或规则的更多信息，请参阅 *自动化控制器管理指南中的* 实例组 部分。
- 有关将 实例组连接到容器的更多信息，请参阅容器组。

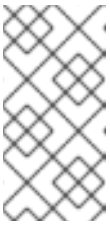
24.1. 创建实例组

使用以下步骤创建新实例组。

流程

1. 在导航面板中，选择 **Administration → Instance Groups**。
2. 从 **Add instance group** 列表中选择 **Add**。
3. 在以下字段中输入相关信息：

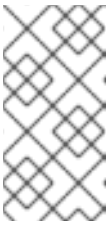
- 名称 ：名称必须是唯一的，且不能命名为 **"controller"**。
- 策略实例最小值 ：在新实例上线时，自动分配给此组的最少实例数量。
- 策略实例百分比 ：使用滑块来选择在新实例上线时自动分配给此组的最小实例百分比。



注意

创建新实例组时不需要策略实例字段。如果没有指定值，则 **Policy** 实例最小值 和 **Policy** 实例百分比 默认为 **0**。

- 最大并发作业 ：指定可为任何给定作业运行的最大 **fork** 数量。
- 最大 **fork** ：指定可为任何给定作业运行的最大并发作业数。



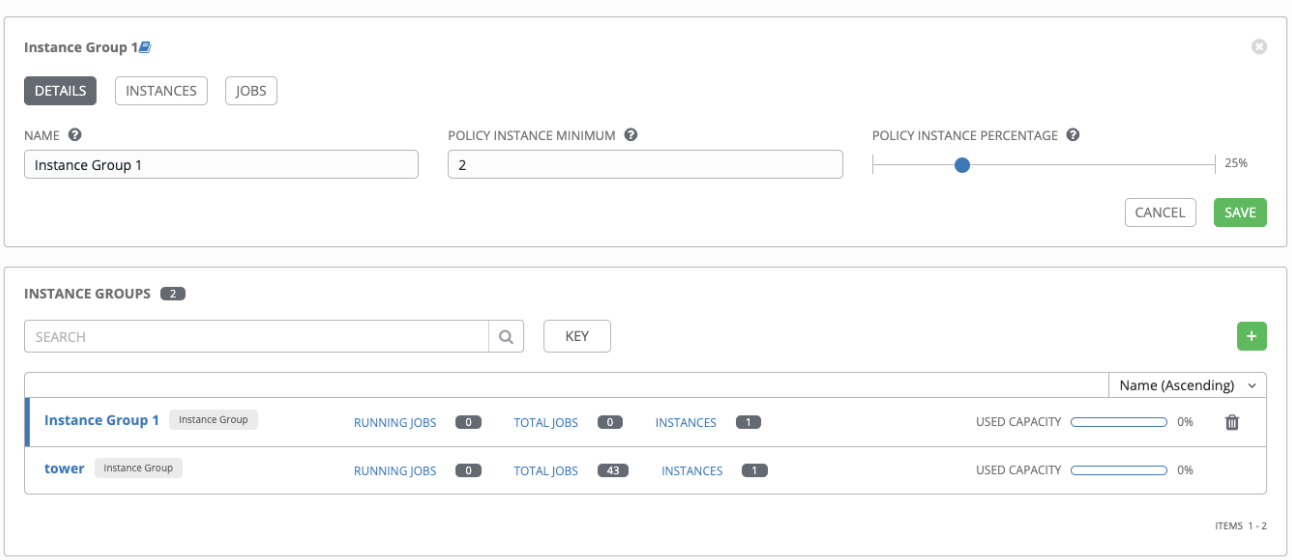
注意

Max concurrent jobs 和 **Max forks** 的默认值 **0** 表示没有限制。如需更多信息，请参阅 *自动化控制器管理指南中的* 实例组容量限制。

4. 点击 **Save**。

当您成功创建实例组时，新创建的实例组的 **Details** 选项卡将保留，允许您查看和编辑您的实例组信息。当您点击 **Instance Groups** 列表视图中的 **Edit**

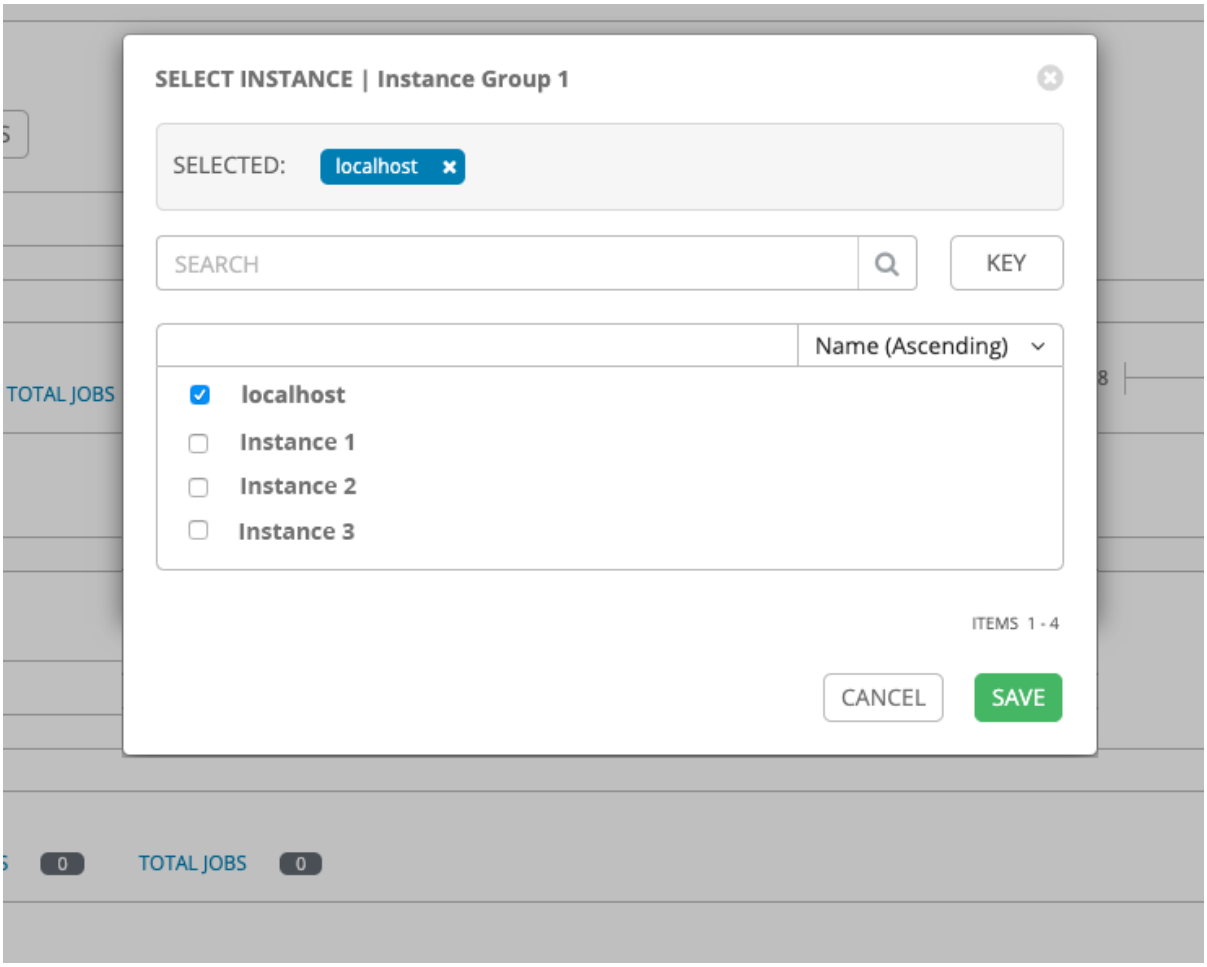
图标时，就会打开这个界面。您还可以编辑 实例， 并查看与此实例组关联的 作业 ：



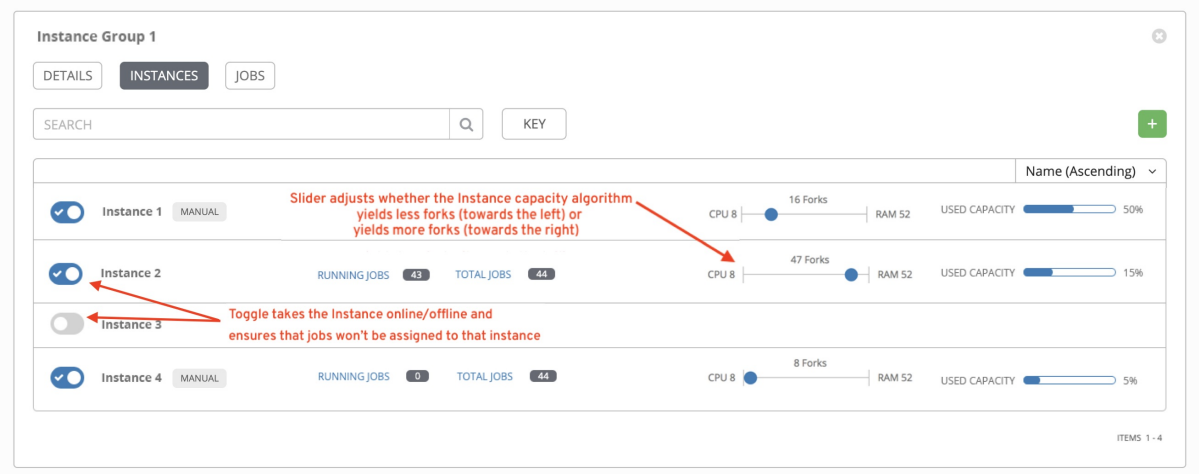
24.1.1. 将实例关联到实例组

流程

1. 选择 **Instance Groups** 窗口的 **Instances** 选项卡。
2. 点关联。
3. 点击列表中一个或多个可用实例旁边的复选框，以选择要与实例组关联的实例：



4. 在以下示例中，添加到实例组中的实例以及其容量的信息：



24.1.2. 查看与实例组关联的作业

流程

1. 选择 **Instance Group** 窗口中的 **Jobs** 选项卡。

2. 点击作业旁边的箭头

图标展开视图并显示每个作业的详情。

每个作业都显示以下详情：

- 作业状态
- **ID** 和名称
- 作业类型
- 它启动和完成的时间
- 谁启动了作业以及与其关联的可用资源，如模板、清单、项目和执行环境

其他资源

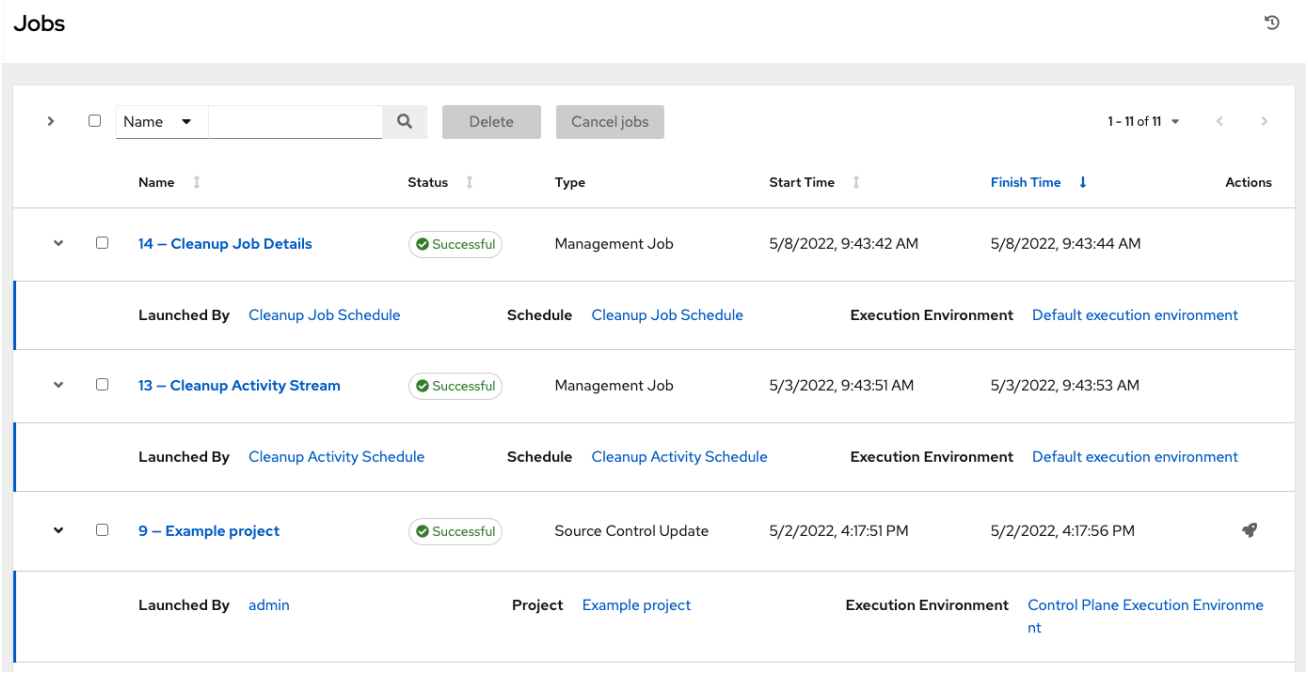
实例会根据实例组的策略运行。如需更多信息，请参阅 *自动化控制器管理指南中的* 实例组策略。

第 25 章 自动化控制器中的作业

作业是为主机清单启动 **Ansible playbook** 的一个自动化控制器实例。

Jobs 列表视图显示作业列表及其状态，显示为成功完成、失败或活跃（正在运行）作业。默认视图为折叠状态**(Compact)**，作业名称、状态、作业类型、启动和完成时间。您可以点箭头

图标展开并查看更多信息。您可以根据各种条件对列表进行排序，并执行搜索来过滤感兴趣的作业。

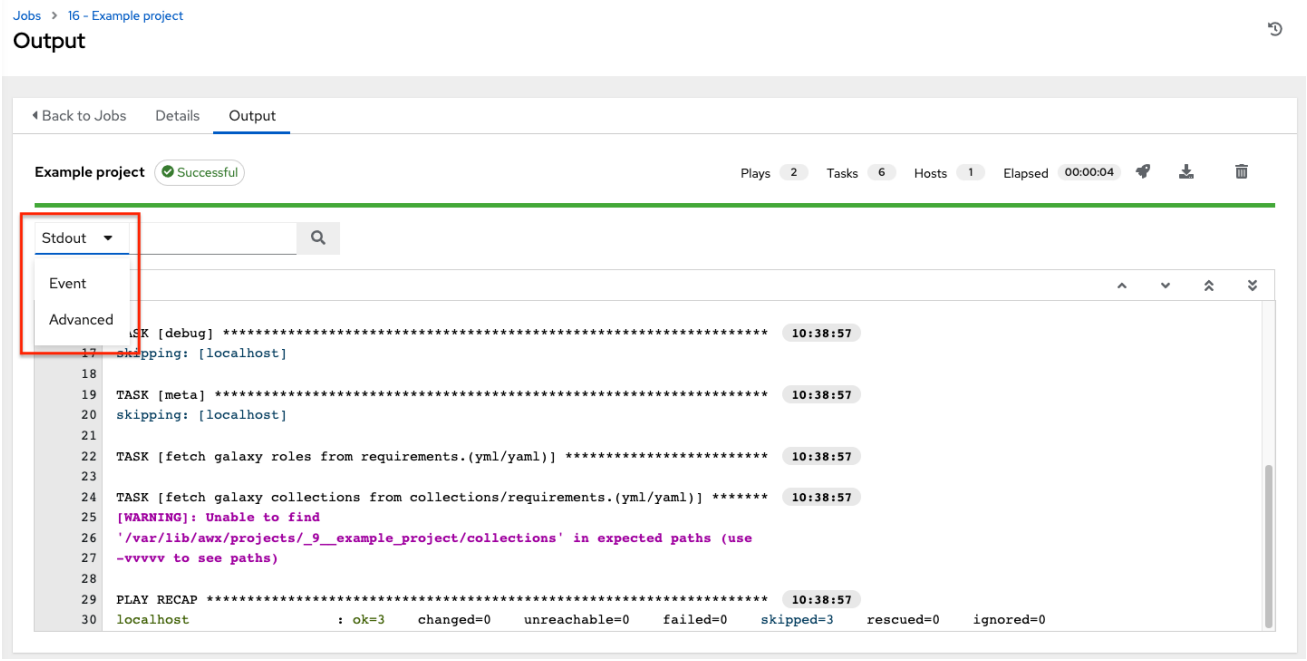


在此屏幕中，您可以完成以下任务：

- 查看特定作业的详情和标准输出
- 重启

  作业
- 删除所选作业

重新启动操作只适用于重新启动 **playbook** 运行，不适用于项目或清单更新、系统作业和工作流作业。当作业重启时，会显示 **Jobs Output** 视图。选择任何类型的作业也会带您进入该作业的作业输出视图，您可以根据各种条件过滤作业：



- **Stdout** 选项是默认显示，显示作业进程和输出。

- **Event** 选项允许您根据感兴趣的事件进行过滤，如错误、主机失败、主机重试和跳过的项目。您可以根据需要在过滤器中包含多个事件。

- **Advanced** 选项是一个优化的搜索，可让您组合包含或排除条件、按键搜索或查找类型。有关使用搜索的更多信息，请参阅 搜索 部分。

25.1. 清单同步作业

执行清单同步时，结果会显示在 **Output** 选项卡中。如果使用，**Ansible CLI** 会显示相同的信息。这对调试非常有用。**ANSIBLE_DISPLAY_ARGS_TO_STDOUT** 参数设置为 **False**，适用于所有 **playbook** 运行。此参数与 **Ansible** 的默认行为匹配，且不会在作业详情界面的任务标头 中显示任务 参数，以避免将某些敏感模块参数泄漏到 **stdout**。要恢复之前的行为，请通过 **AWX_TASK_ENV** 配置设置将 **ANSIBLE_DISPLAY_ARGS_TO_STDOUT** 设置为 **True**。

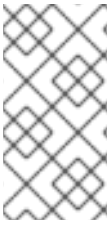
如需更多信息，请参阅 **ANSIBLE_DISPLAY_ARGS_TO_STDOUT**。

使用图标重新启动

，下载

作业输出或删除

作业。

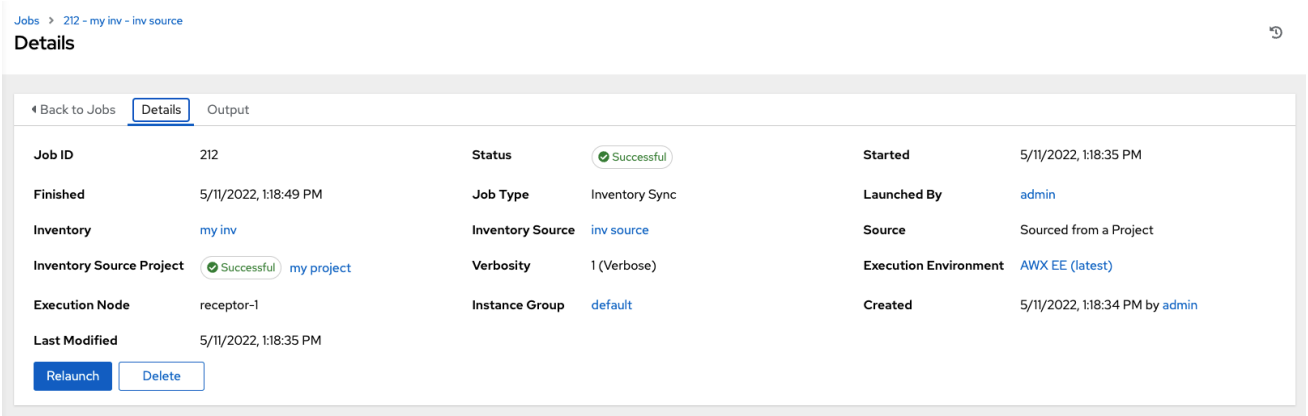


注意

您可以在相关作业运行时执行清单更新。如果您有大型项目（大约 **10 GB**），**/tmp** 上的磁盘空间可能会出现问题。

25.1.1. 清单同步详情

访问 **Details** 选项卡，以查看作业执行的详情：



您可以查看已执行作业的以下详情：

- 状态 ：可以是以下任意一种：
  - 待定 ：清单同步已创建，但尚未排队或启动。任何作业（不仅仅是清单源同步）都会处于待处理状态，直到系统准备好运行为止。清单源同步没有就绪的原因包括：
    - 当前正在运行的依赖项（所有依赖项都必须完成才能执行下一个步骤）。
    - 在为其配置的位置运行不足以达到容量。
  - 等待 ：清单同步处于等待执行的队列中。
  - **Running:** 清单同步当前正在进行中。
  - 成功 ：清单同步作业成功。
  - 失败 ：清单同步作业失败。
- **Inventory** ：关联的清单组的名称。
- **Source:** 云清单的类型。
- 清单源项目 ：用作此清单同步作业源的项目。
- 执行环境 ：使用的执行环境。
- 执行节点 ：用于执行该作业的节点。
- 实例组 ：与此作业使用的实例组的名称（自动化控制器是默认实例组）。

选择这些项目可让您查看对应的作业模板、项目和其他对象。

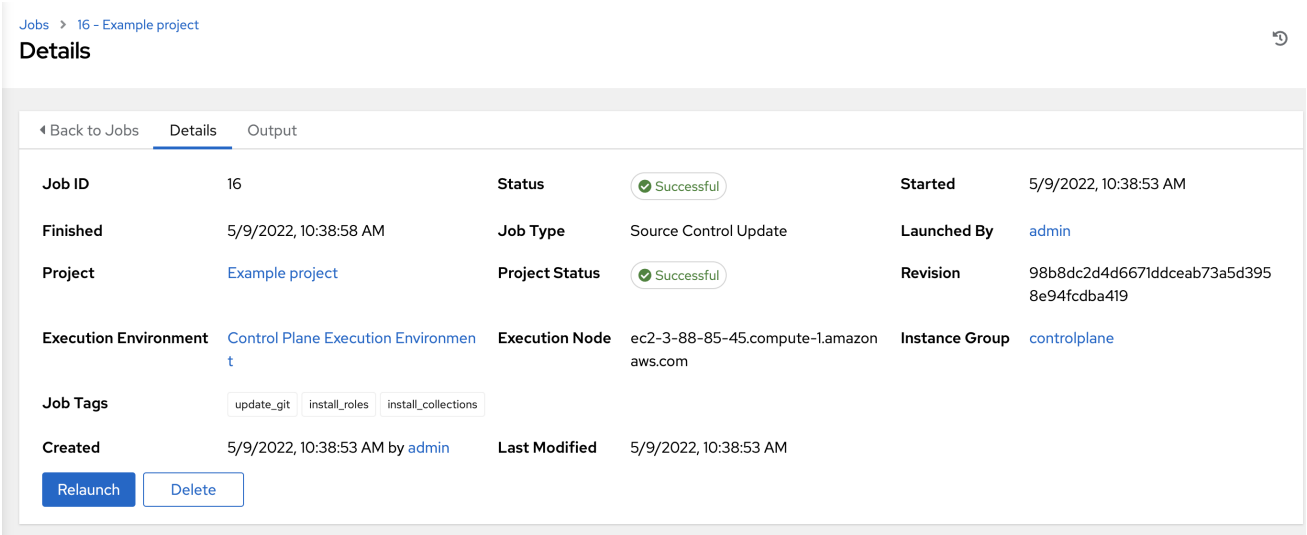
25.2. SCM 清单作业

当执行来自 **SCM** 的清单（如 **git**）时，其结果将显示在 **Output** 选项卡中。如果使用，**Ansible CLI** 会显示相同的信息。这对调试非常有用。使用导航菜单中的图标重新启动(

)、下载(

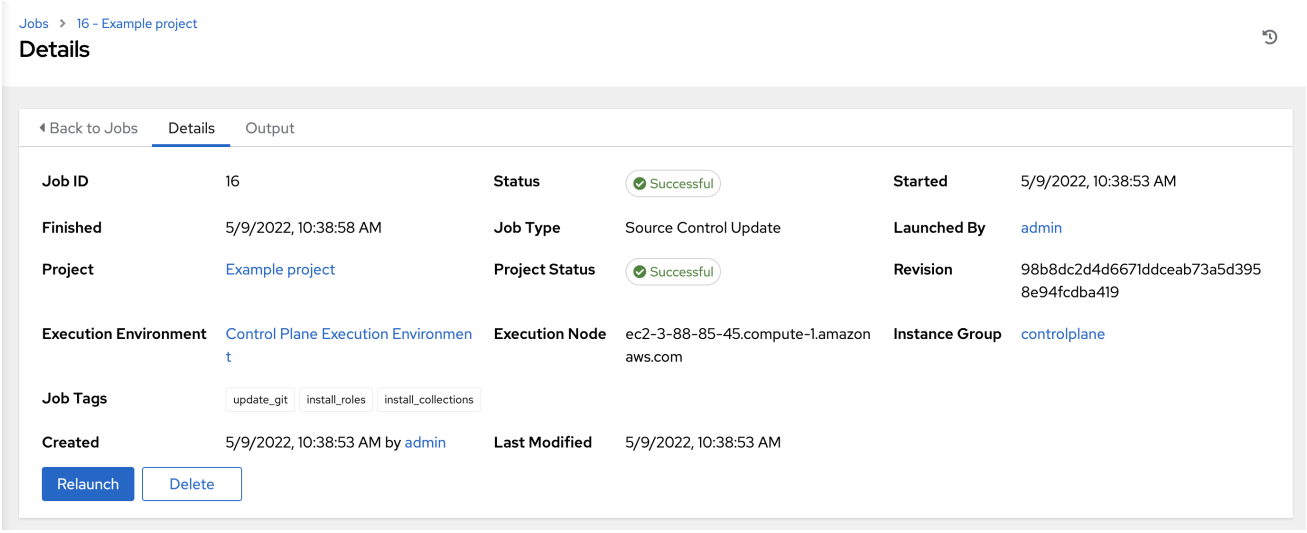
)作业输出，或删除(

)作业。



25.2.1. SCM 清单脚本

要查看作业执行及其关联的项目的详细信息，请选择 **Access** 选项卡。



您可以查看已执行作业的以下详情：

- 状态 ：可以是以下任意一种：
  - **Pending: SCM** 作业已创建，但尚未排队或启动。任何作业（不仅仅是 **SCM** 作业）会一直处于待处理状态，直到系统准备好运行为止。**SCM** 作业未准备就绪的原因包括：依赖项当前正在运行（所有依赖项都必须已完成才能执行下一个步骤），或者其配置的位置没有足够的运行容量。
  - **waiting : SCM** 作业处于等待执行的队列中。
  - 运行 ：**SCM** 作业当前正在进行中。
  - 成功 ：最后一个 **SCM** 作业成功。
  - 失败 ：最后一个 **SCM** 作业失败。
- 作业类型 ：**SCM** 作业显示源控制更新。
- 项目 ：项目名称。
- 项目状态 ：指示关联的项目是否已成功更新。
- 修订 ：指示此作业中使用的源项目的修订号。
- 执行环境 ：指定用于运行此作业的执行环境。
- 执行节点 ：指示作业运行的节点。

- 实例组 ：指示作业运行的实例组（如果指定）。
- **Job Tags** ：标签显示执行的各种作业操作。

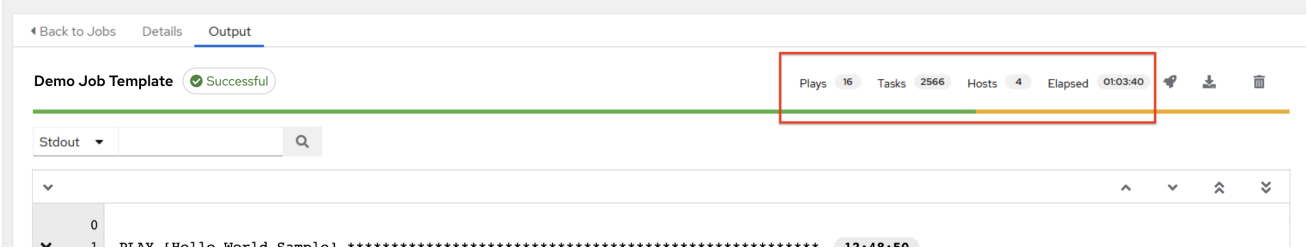
选择这些项目可让您查看对应的作业模板、项目和其他对象。

25.3. PLAYBOOK 运行任务

执行 **playbook** 时，结果会显示在 **Output** 选项卡中。如果使用，**Ansible CLI** 会显示相同的信息。这对调试非常有用。

事件摘要显示作为此 **playbook** 一部分运行的以下事件：

- 此 **playbook** 运行的次数显示在 **Plays** 字段中
- 与此 **playbook** 关联的任务数量显示在 **Tasks** 字段中
- 与此 **playbook** 关联的主机数量显示在 **Hosts** 字段中
- 完成 **playbook** 运行所需的时间显示在 **Elapsed** 字段中



使用事件旁边的图标重新启动(
)、下载(
)作业输出，或删除(
)作业。

将鼠标悬停在 **Output** 视图中的主机状态栏的部分上，并且显示与该状态关联的主机数量。

在 **Job Templates** 页的 **Jobs** 选项卡中启动作业后，也提供了 **playbook** 作业的输出。单击输出中的行项目任务，以查看其主机详细信息。

25.3.1. 搜索

使用 **Search** 查找特定的事件、主机名及其状态。要只过滤具有特定状态的某些主机，请指定以下有效状态之一：

ok

表示任务成功完成，但没有在主机上执行任何更改。

changed

playbook 任务已执行。由于 **Ansible** 任务应该具有幂等性，因此任务可能会成功退出，而不必在主机上执行任何内容。在这些情况下，该任务返回 **ok**，但未 更改。

失败

任务失败。此主机停止了进一步的 **playbook** 执行。

unreachable

主机无法从网络访问，或者具有另一个与之关联的致命错误。

跳过

playbook 任务跳过，因为主机不需要更改即可达到目标状态。

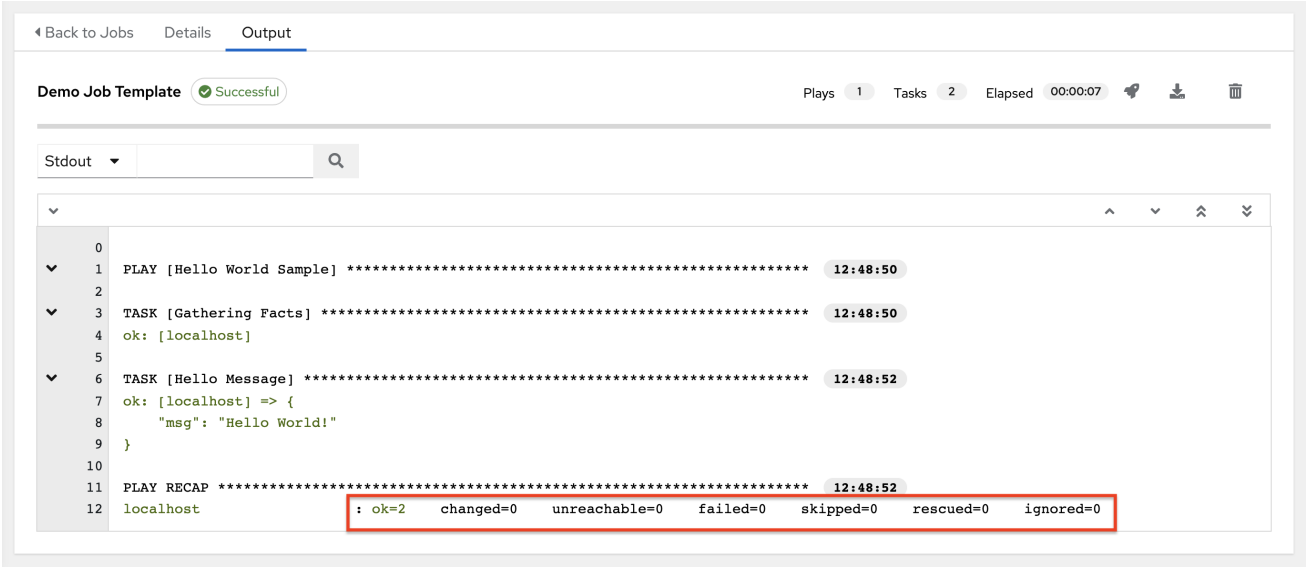
rescued

这将显示失败的任务，然后执行 **rescue** 部分。

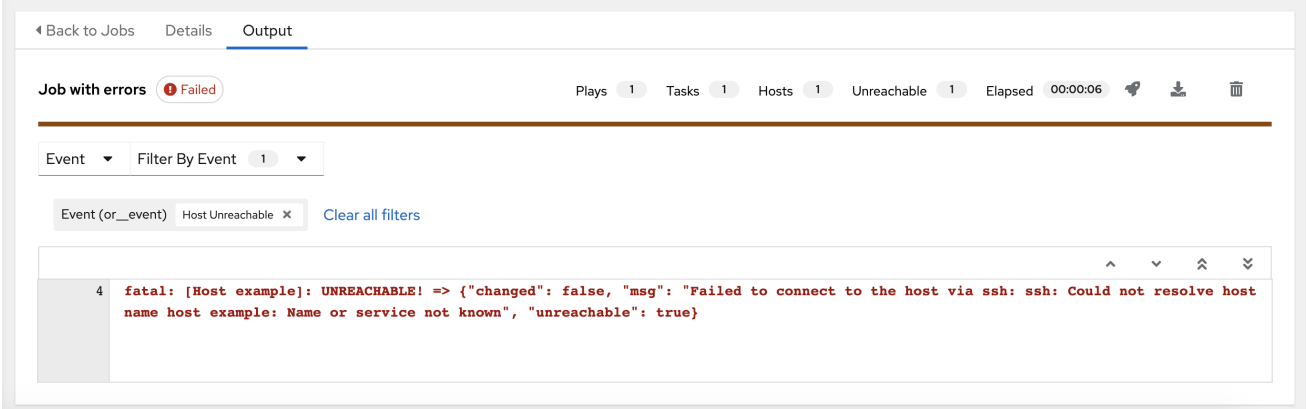
忽略

这将显示失败的任务，并且配置了 **ignore_errors: yes**。

这些状态也显示在每个 **Stdout** 窗格中，它们称为主机概述字段的一组"统计数据"：



以下示例显示一个只包含无法访问主机的搜索：



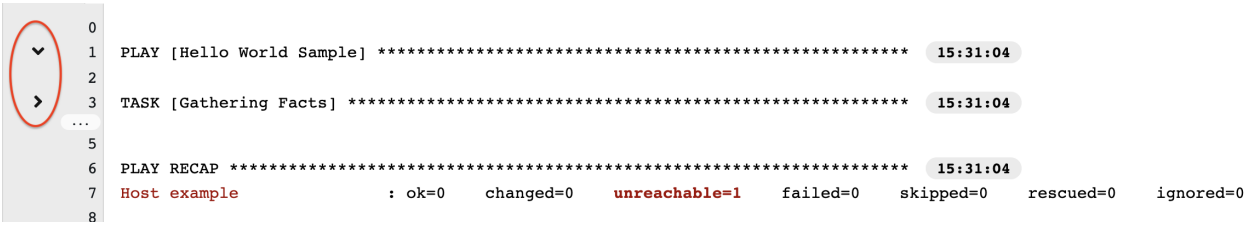
有关使用搜索的更多信息，请参阅 搜索 部分。

标准输出视图显示特定作业上发生的事件。默认情况下，所有行都会被扩展，以便显示详情。使用折叠功能(

)图标切换到仅包含 **play** 和任务的标头的视图。点击加号(

)图标查看标准输出的所有行。

您可以通过单击特定 **play** 或任务旁的箭头图标来显示特定 **play** 或任务的所有详情。点侧边的箭头到下键，以展开与该 **play** 或任务关联的行。点箭头回到侧边位置，以折叠和隐藏行。



在展开或折叠模式中查看详情时，请注意以下几点：

- 每个没有折叠的显示行都有对应的行号和开始时间。
- 在 **play** 或任务完成后，展开或折叠图标位于任何 **play** 或任务开始时。
- 如果查询特定的 **play** 或任务，它会在其完成进程结束时出现折叠状态。
- 在某些情况下会出现错误消息，表示输出可能太大而无法显示。当存在超过 **4000** 事件时会出现这种情况。使用搜索和过滤特定事件来绕过错误。

单击 **Stdout** 窗格中的事件行，并在单独的窗口中显示 **Host Events** 窗口。此窗口显示受该特定事件影响的主机。

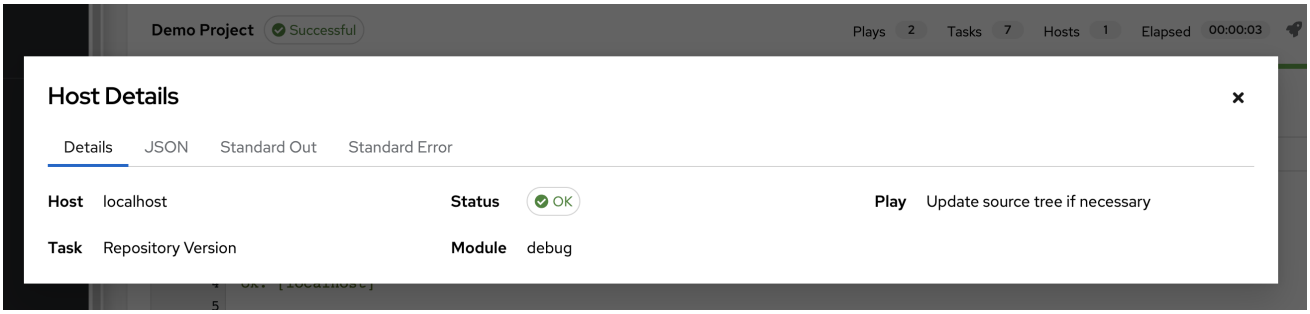
注意

升级到最新版本的 **Ansible Automation Platform** 涉及逐渐迁移所有历史 **playbook** 输出和事件。这个迁移过程是逐步的，在安装完成后自动在后台进行。在迁移完成前，带有大量历史作业输出（几十或几百 **GB** 的输出）的安装可能会缺少作业输出。最新的数据显示在输出的顶部，后跟旧的事件。

25.3.2. 类型详情

Host Details 窗口显示受所选事件及其关联的 **play** 和任务影响的主机的以下信息：

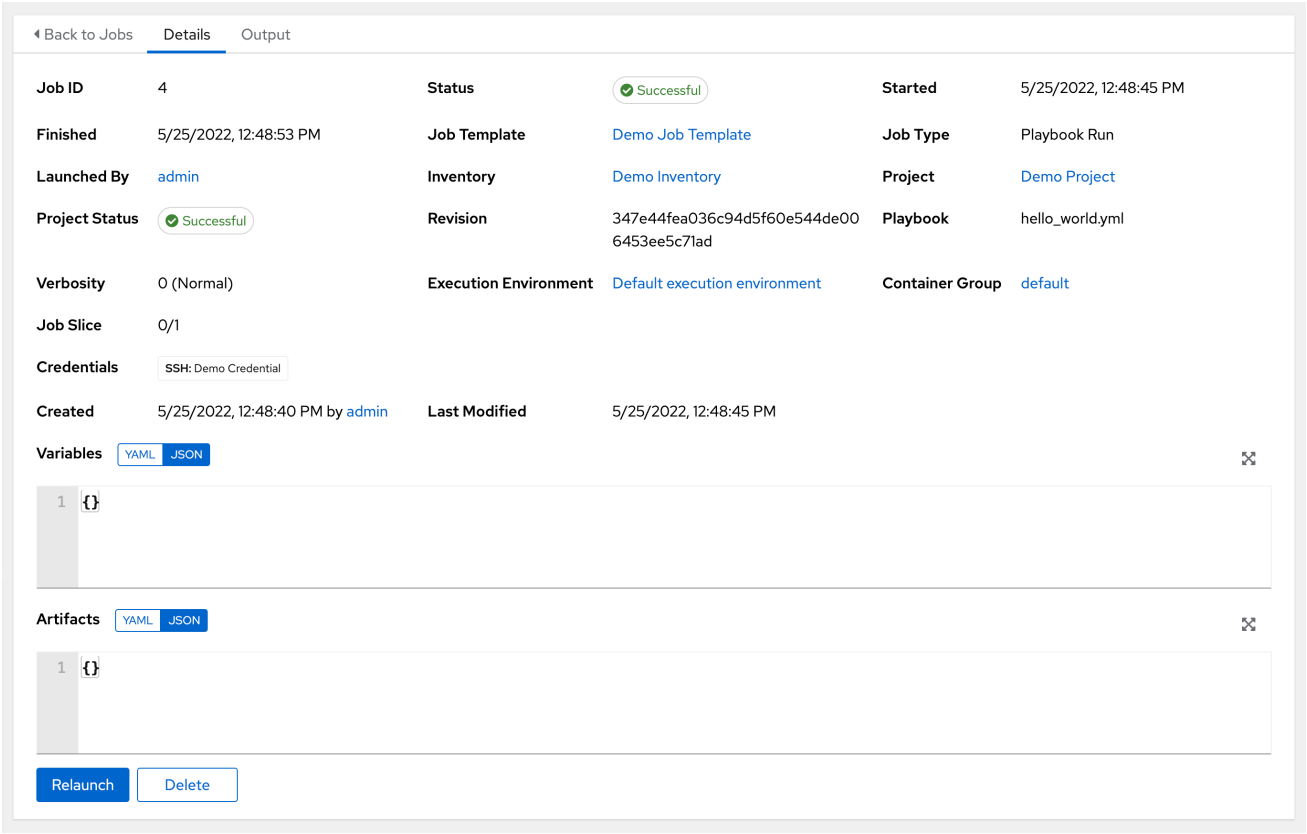
- 主机.
- 状态 ：
- 在 **Play** 字段中运行的类型。
- 任务 的类型。
- 如果适用，**Ansible Module** 任务以及该模块的任何参数。



若要以 **JSON** 格式查看结果，请单击 **JSON** 选项卡。要查看任务的输出，请单击 **Standard Out**。要查看输出中的错误，请点 **Standard Error**。

25.3.3. Playbook 运行详情

访问 **Details** 选项卡，以查看作业执行的详情：



您可以查看已执行作业的以下详情：

- 状态 ：可以是以下任意一种：
  - **Pending: playbook** 运行已创建，但尚未排队或启动。任何作业（不仅仅是 **playbook** 运行）会一直处于待处理状态，直到系统准备好运行为止。**playbook** 运行未准备就绪的原因包括：依赖项当前正在运行（所有依赖项都必须已完成才能执行下一个步骤），或者其配置的位置没有足够的运行容量。
  - 等待 ：**playbook** 运行处于等待执行的队列中。
  - **Running: playbook** 运行当前正在进行中。
  - 成功 ：最后一个 **playbook** 运行成功。
  - 失败 ：最后一个 **playbook** 运行失败。

- 作业模板 ：从中启动此作业的作业模板的名称。
- 清单 ：选择针对此任务运行此清单。
- 项目 ：与启动的作业关联的项目名称。
- 项目状态 ：与启动作业关联的项目的状态。
- **Playbook:** 用于启动此作业的 **playbook**。
- 执行环境 ：此作业中使用的执行环境的名称。
- 容器组 ：此作业中使用的容器组的名称。
- **credentials** ：此作业中使用的凭证。
- 额外变量 ：创建作业模板时传递的任何额外变量都会在此处显示。

选择其中一个项目来查看对应的作业模板、项目和其他对象。

25.4. 自动化控制器容量确定和作业影响

自动化控制器容量系统根据实例可用的资源量以及正在运行的作业的大小（称为影响）来确定实例可在实例上运行的作业数量。用于确定这一点的算法基于以下两个方面：

- 系统可使用多少内存**(mem_capacity)**
- 系统可使用多少个处理容量**(cpu_capacity)**

容量也会影响实例组。由于组由实例组成，因此实例也可以分配到多个组。这意味着，对一个实例的影

响可能会影响其他组的整体容量。

实例组而不是实例本身，可以分配给不同级别的作业使用。如需更多信息，请参阅 *自动化控制器管理指南中的* 集群。

当任务管理器准备其图形来确定作业运行的组时，它会将实例组的容量提交到尚未准备好启动的作业。

在较小的配置中，如果只有一个实例可用于某个作业运行，任务管理器可让该作业在实例上运行，即使它会使实例超额。这样可保证作业不会因为置备系统而卡住。

其他资源

- 有关 容器组的详情，请参考 *自动化控制器管理指南中的* 容器容量限制。
- 有关分片作业及其对容量的影响的详情，请参考 作业分片执行行为。

25.4.1. 容量算法的资源确定

容量算法确定系统可以同时运行多少个 **fork**。这些算法控制 **Ansible** 可以同时与多少个系统通信。增加自动化控制器系统运行的 **fork** 数量，使作业可以更快地运行，方法是并行执行更多工作。但是，这会增加系统的负载，这可能导致工作变慢。

默认 **mem_capacity** 可让您超额提交处理资源，同时防止系统内存不足。如果您的大多数工作不是处理器密集型，则选择此模式可最大化 **fork** 数量。

25.4.1.1. 内存相对容量

mem_capacity 相对于每个 **fork** 所需的内存量计算。考虑到内部组件的开销，每个分叉大约需要 **100MB**。在考虑 **Ansible** 作业可用的内存量时，容量算法保留 **2GB** 内存，以考虑存在其他服务。其算法公式是：

```
(mem - 2048) / mem_per_fork
```

以下是一个示例：

```
(4096 - 2048) / 100 == ~20
```

-

具有 **4GB** 内存的系统可以运行 **20** 个分叉。**mem_per_fork** 值通过设置 **SYSTEM_TASK_FORKS_MEM** 的值来控制，默认值为 **100**。

25.4.1.2. CPU 相对容量

Ansible 工作负载通常是处理器密集型。在这种情况下，您可以减少同步工作负载，使更多任务可以更快地运行，并减少这些作业的平均完成时间。

就像 **mem_capacity** 算法调整每个 **fork** 所需的内存量一样，**cpu_capacity** 算法会调整每个 **fork** 所需的处理资源量。这个基准值是每个内核的四个 **fork**。其算法公式是：

```
cpus * fork_per_cpu
```

例如，**4** 核系统类似如下：

```
4 * 4 == 16
```

您可以通过将 **SYSTEM_TASK_FORKS_CPU** 的值设置为 **4** 来控制 **fork_per_cpu** 的值。

25.4.2. 容量作业影响

在选择容量时，了解每个作业类型对容量的影响非常重要。

Ansible 的默认 **fork** 值为 **5**。但是，如果您将自动化控制器设置为针对更少的系统运行，则实际的并发值会较低。

当在自动化控制器中运行作业时，所选的 **fork** 数量会递增 **1**，以补充 **Ansible** 父进程。

Example

如果您针对 **5** 的五个系统运行 **playbook**，则从作业影响角度来看，实际的 **fork** 值为 **6**。

25.4.2.1. 自动化控制器中作业类型的影响

作业和临时作业遵循前面的模型，分叉 **+1**。如果在作业模板上设置了 **fork** 值，则您的作业容量值是提供的 **forks** 值的最小值，以及您拥有的主机数量，再加上 **1**。**+1** 是考虑父 **Ansible** 进程。

实例容量决定了哪些作业被分配给任何特定实例。如果作业和临时命令具有更高的 **fork** 值，则使用更多容量。

作业类型包括以下内容，具有固定影响：

- 清单更新：**1**
- 项目更新：**1**
- 系统作业：**5**

注意

如果您没有在作业模板上设置 **fork** 值，则您的作业将使用 **Ansible** 的默认 **fork** 值 **5**。但是，如果您的作业少于五个主机，它会使用较少的主机。通常，设置 **fork** 值高于系统能够使问题出现内存不足或过量提交 **CPU** 时的问题。您使用的作业模板 **fork** 值必须适合于系统。如果您有使用 **1000** 个 **fork** 的 **playbook**，但您的系统都没有足够容量，则您的系统会降低，并面临性能或资源问题的风险。

25.4.2.2. 选择正确的容量

从 **CPU** 密集型或内存密集型容量限制中选择容量会在最小或最大分叉数量之间进行选择。在上例中，**CPU** 容量最多允许 **16** 个 **fork**，而内存容量允许 **20** 个。对于某些系统，它们之间的差别可能较大，您可能希望在两者之间保持平衡。

instance 字段 **capacity_adjustment** 允许您选择要考虑多少。它表示为 **0.0** 到 **1.0** 之间的值。如果设置为 **1.0**，则使用最大值。上例涉及内存容量，因此可以选择 **20** 个 **fork**。如果设置为 **0.0**，则使用最小值。值 **0.5** 是两个算法之间的 **50/50** 平衡，即 **18**：

```
16 + (20 - 16) * 0.5 = 18
```

流程

查看或编辑容量：

1. 从 **Instances Groups** 列表视图中，选择所需的实例。

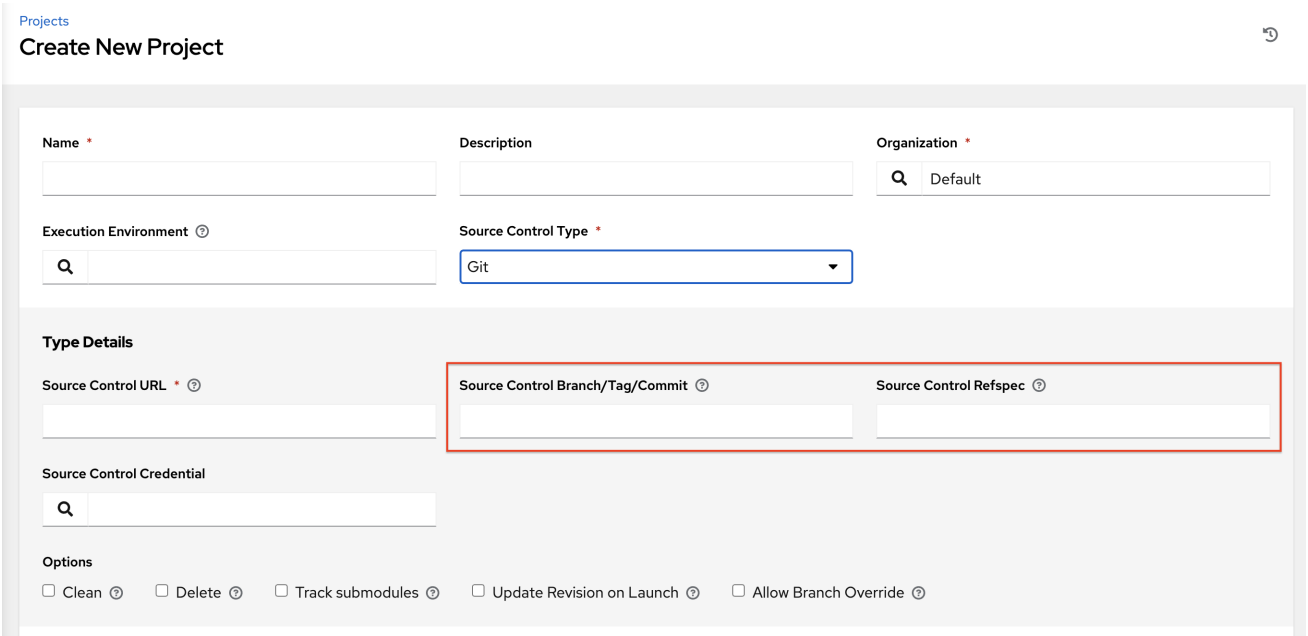
2. 选择 **Instances** 选项卡，并调整 **Capacity Adjustment** 滑块。

注意

滑块调整实例容量算法是否产生较少的分叉（在左侧）或产生更多分叉（在右侧）。

25.5. 作业分支覆盖

项目在 **scm_branch** 字段中指定要从源控制使用的分支、标签或引用。它们由 **Type Details** 字段中指定的值表示：



在创建或编辑作业时，您可以选择 **Allow Branch Override**。选中此选项时，项目管理员可以将分支选择委托给使用该项目的作业模板，只需要项目 **use_role**。

25.5.1. 源树复制行为

每个作业运行都有自己的私有数据目录。此目录包含作业运行的给定 **scm_branch** 的项目源树的副本。作业可以自由地更改项目文件夹，并在仍在运行时使用这些更改。此文件夹是临时的，在作业运行结

束时被删除。

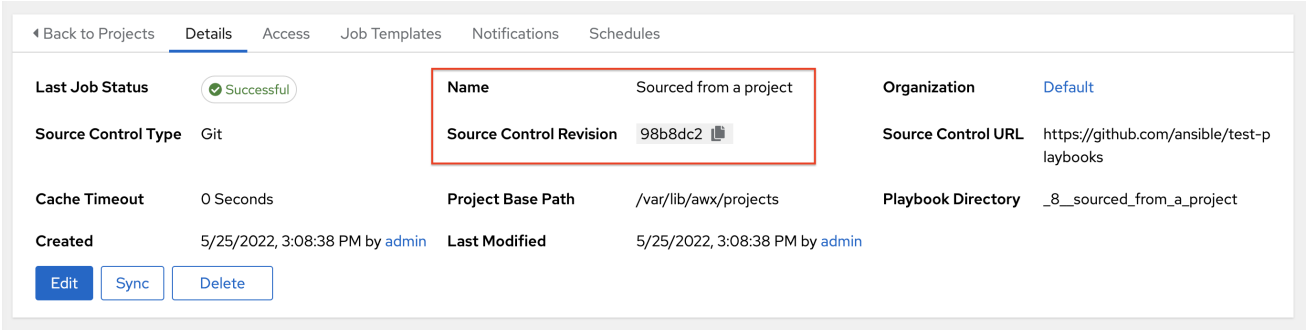
如果您检查 **Clean** 选项，则在自动化控制器的本地副本中删除修改后的文件。这可以通过在与 **git** 或 **Subversion** 相关的相应 **Ansible** 模块中使用 **force** 参数来完成。

其他资源

有关更多信息，请参阅 **Ansible** 文档中的 **Parameters** 部分。

25.5.2. 项目修订行为

在项目更新过程中，默认分支的修订版本（在项目的 **SCM** 分支字段中指定 ）会在更新时存储。如果在作业中提供非默认 **SCM** 分支 （而不是提交散列或标签），则在作业启动前会立即从源控制远程拉取最新的修订版本。此修订版本显示在作业的 **Source Control Revision** 字段中，及其项目更新。



因此，非默认分支无法离线作业运行。为确保作业从源控制运行静态版本，请使用标签或提交哈希。项目更新不会保存所有分支，而只有项目默认分支。

SCM 分支 字段未验证，因此项目必须更新以确保其有效。如果提供或提示了此字段，则不会验证作业模板的 **Playbook** 字段，您必须启动作业模板以验证预期的 **playbook** 是否存在。

25.5.3. Git Refspec

SCM Refspec 字段指定更新应该从远程下载的额外引用。示例包括以下内容：

- **refs/:refs/remotes/origin/** ：这将获取所有引用，包括远程的远程
- **refs/pull/:refs/remotes/origin/pull/ (GitHub-specific)**：这将获取所有拉取请求的所有 **refs**

- **refs/pull/62/head:refs/remotes/origin/pull/62/head** ：这会获取一个 **GitHub** 拉取请求的 **ref**

对于大型项目，在使用第一个或第二个示例时请考虑性能影响。

SCM Refspec 参数会影响项目分支的可用性，并可启用对不可用的引用的访问。前面的示例允许您提供 **SCM** 分支 的拉取请求，在没有 **SCM Refspec** 字段的情况下无法实现。

默认情况下，**Ansible git** 模块获取 **refs/heads/**。这意味着，如果 **SCM Refspec** 为空，项目的分支、标签和提交散列可用作 **SCM** 分支。**SCM Refspec** 字段中指定的值会影响哪些 **SCM** 分支 字段可用作覆盖。项目更新（任何类型的）执行额外的 **git fetch** 命令来从远程拉取该 **refspec**。

Example

您可以设置使用第一个或第二个 **refspec** 示例启用分支覆盖的项目。在提示 **SCM** 分支 的作业模板中使用此选项。然后，客户端可以在创建新拉取请求时启动作业模板，提供分支 **pull/N/head**，并且作业模板可以根据提供的 **GitHub** 拉取请求引用运行。

其他资源

有关更多信息，请参阅 **Ansible git** 模块。

第 26 章 使用 WEBHOOK

Webhook 可让您通过 **Web** 在应用程序间执行指定的命令。自动化控制器目前提供 **webhook** 与 **GitHub** 和 **GitLab** 的集成。

使用以下服务设置 **webhook**：

- 设置 **GitHub Webhook**
- 设置 **GitLab Webhook**
- 查看有效负载输出

webhook 的 **GitHub** 和 **GitLab** 的 **post-status-back** 功能旨在仅在某些 **CI** 事件下工作。在服务日志中接收其他类型的事件会导致信息类似如下：

awx.main.models.mixins Webhook 事件没有关联状态 **API** 端点并跳过。

26.1. 设置 GITHUB WEBHOOK

自动化控制器可根据触发的 **webhook** 事件运行作业。作业状态信息（待定、错误、成功）只能针对拉取请求事件发回。如果您不需要自动化控制器将作业状态回 **webhook** 服务，请直接转至第 **3** 步。

流程

1. 生成用于自动化控制器 *的个人访问令牌* **(PAT)**：
   a. 在 **GitHub** 帐户的配置集设置中，选择 **Settings**。
   b. 在导航面板中，选择 **<&gt ; Developer Settings**。

c. 在 **Developer Settings** 页面中，选择 **Personal access token**。

d. 在 个人访问令牌 屏幕中，单击 **Generate a personal access token**。

e. 提示时，请输入您的 **GitHub** 帐户密码以继续。

f. 在 **Note** 字段中，输入有关此 **PAT** 用途的简要描述。

g. 在 **Select scopes** 字段中，选中 **repo:status**、**repo_deployment** 和 **public_repo** 旁边的框。自动化 **Webhook** 只需要存储库范围访问权限，但邀请除外。如需更多信息，请参阅 **OAuth** 应用文档的范围。

h. 点 **Generate token**。

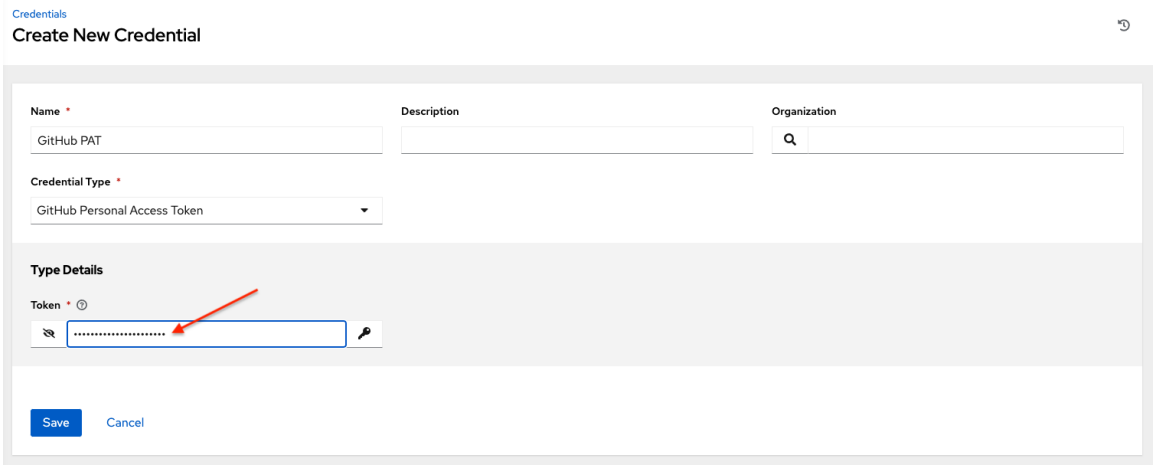
重要

生成令牌时，请确保复制 **PAT**，因为您在第 **2** 步中需要它。您无法在 **GitHub** 中再次访问此令牌。

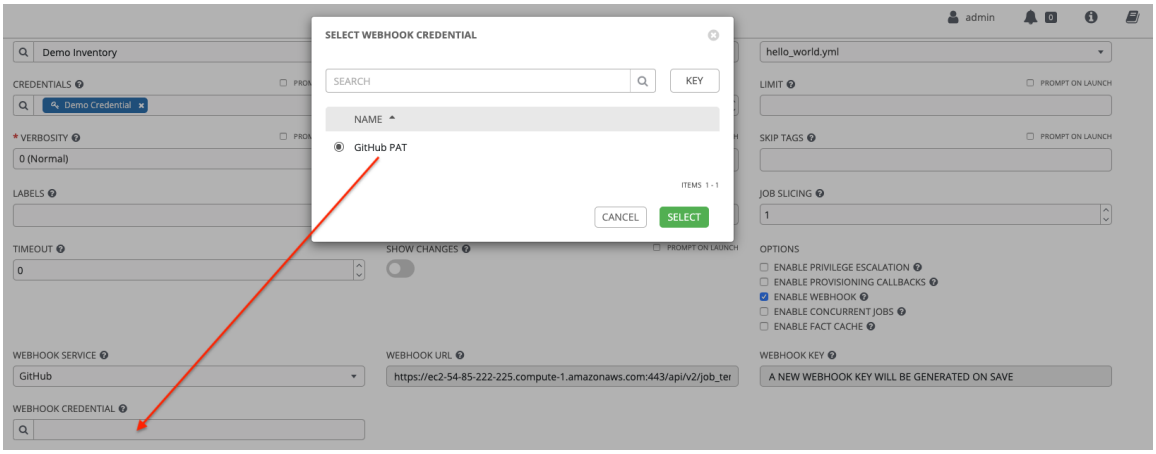
2. 使用 **PAT** 创建 **GitHub** 凭证（可选）：

a. 进入您的实例，并使用生成的令牌为 **GitHub PAT** 创建新凭证。

b. 记录此凭据的名称，因为您在回发到 **GitHub** 的作业模板中使用它。



c. 进入您要启用 **webhook** 的作业模板，然后选择 **webhook** 服务和您在上一步中创建的凭证。



d. 点击 **Save**。您的作业模板已设置为回 **GitHub**。

3. 进入您要配置 **webhook** 的 **GitHub** 存储库，然后选择 **Settings**。

4. 在导航面板中，选择 **Webhooks** → **Add webhook**。

5. 要完成 **Add webhook** 页面，您必须检查作业模板或工作流作业模板中的 **Enable Webhook** 选项。如需更多信息，请参阅创建 作业模板和 创建工作流模板 中的第 **3** 步。

6. 完成以下字段：

- **payload URL** ：从作业模板中复制 **Webhook URL** 的内容，并将它粘贴。结果从 **GitHub** 发送到此地址。

- 内容类型 ：将其设置为 **application/json**。

- **Secret** ：从作业模板中复制 **Webhook** 密钥的内容并将其粘贴到此处。

- 您要触发此 **Webhook** 的事件类型？ ：选择您要触发 **Webhook** 的事件类型。任何这样的事件都会触发作业或工作流。要让作业状态**(pending, error, success)**发送到 **GitHub**，您必须在 **Let me select individual events** 部分中选择 **Pull requests** 部分。

- **Active** ：保留此检查。

7. 点击 **Add webhook**。

8. 配置 **webhook** 时，它会显示在您的仓库活跃的 **webhook** 列表中，并能够编辑或删除它。单击 **webhook**，前往 **Manage webhook** 屏幕。

9. 滚动以查看向 **webhook** 发出的发送尝试，以及它们是成功还是失败。

其他资源

如需更多信息，请参阅 **Webhooks** 文档。

26.2. 设置 GITLAB WEBHOOK

自动化控制器可根据触发的 **webhook** 事件运行作业。作业状态信息（待定、错误、成功）只能针对拉取请求事件发回。如果自动化控制器不需要将作业状态回 **webhook** 服务，请直接转至第 **3** 步。

流程

1. 生成用于自动化控制器 *的个人访问令牌* **(PAT)**：

   a. 在 **GitLab** 中的导航面板中，选择您的 **avatar** 和 **Edit profile**。

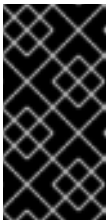
   b. 在导航面板中，选择 **Access token**。

   c. 完成以下字段：

   - 令牌名称 ：输入有关此 **PAT** 用途的简要描述。
   - 过期日期 ：跳过此字段，除非您想为您的 **webhook** 设置过期日期。
   - 选择范围 ：选择适用于您的集成的范围。对于自动化控制器，**api** 是唯一需要的选择。

d.

单击 **Create personal access token**。



重要

生成令牌时，请确保复制 **PAT**，因为您在第 **2** 步中需要它。您无法在 **GitLab** 中再次访问此令牌。

2.

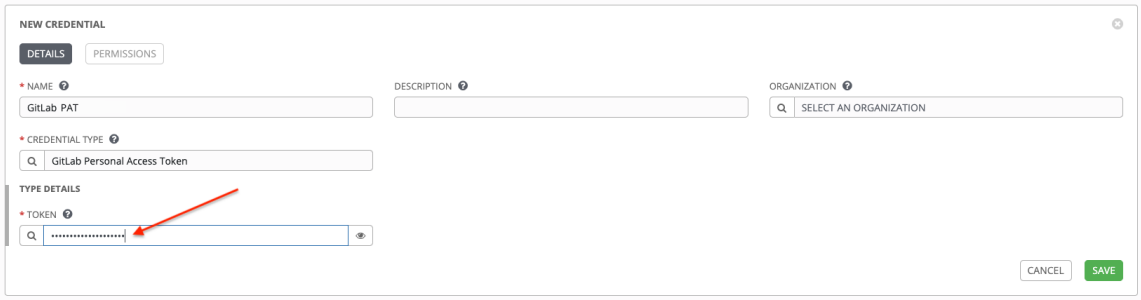
使用 **PAT** 创建 **GitLab** 凭证（可选）：

a.

进入您的实例，并使用生成的令牌为 **GitLab PAT** 创建新凭证。

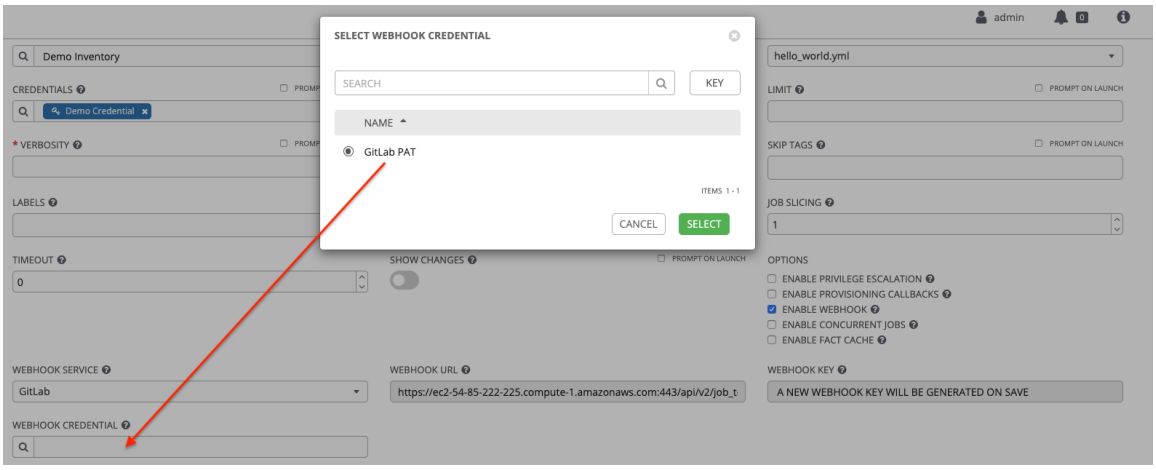
b.

记录此凭据的名称，因为您在回发到 **GitLab** 的作业模板中使用它。



c.

进入您要启用 **webhook** 的作业模板，然后选择 **webhook** 服务和您在上一步中创建的凭证。



d.

点击 **Save**。您的作业模板设置为回发到 **GitLab**。

3. 进入您要配置 **webhook** 的 **GitLab** 存储库。

4. 在导航面板中，选择 **Settings** → **Integrations**。

5. 要完成 **Add webhook** 页面，您必须检查作业模板或工作流作业模板中的 **Enable Webhook** 选项。如需更多信息，请参阅创建 作业模板和 创建工作流模板 中的第 **3** 步。

6. 完成以下字段：

   - **URL** ：从作业模板中复制 **Webhook URL** 的内容，并将它粘贴。结果从 **GitLab** 发送到此地址。
   - **Secret Token** ：从作业模板中复制 **Webhook** 密钥的内容并粘贴它。
   - **trigger** ：选择您要触发 **Webhook** 的事件类型。 任何这样的事件都会触发作业或工作流。要让作业状态**(pending, error, success)**发送到 **GitLab**，您必须在 **Trigger** 部分中选择 **Merge request** 事件。
   - **SSL** 验证 ：使 启用 **SSL** 验证 被选择。

7. 点击 **Add webhook**。

8. 配置 **webhook** 后，它会显示在存储库的列表 **Project Webhooks** 中，并能够测试事件、编辑或删除 **webhook**。测试 **Webhook** 事件会显示每个页面的结果是成功还是失败。

其他资源

如需更多信息，请参阅 **Webhooks**。

26.3. 查看有效负载输出

您可以查看作为额外变量公开的整个有效负载。

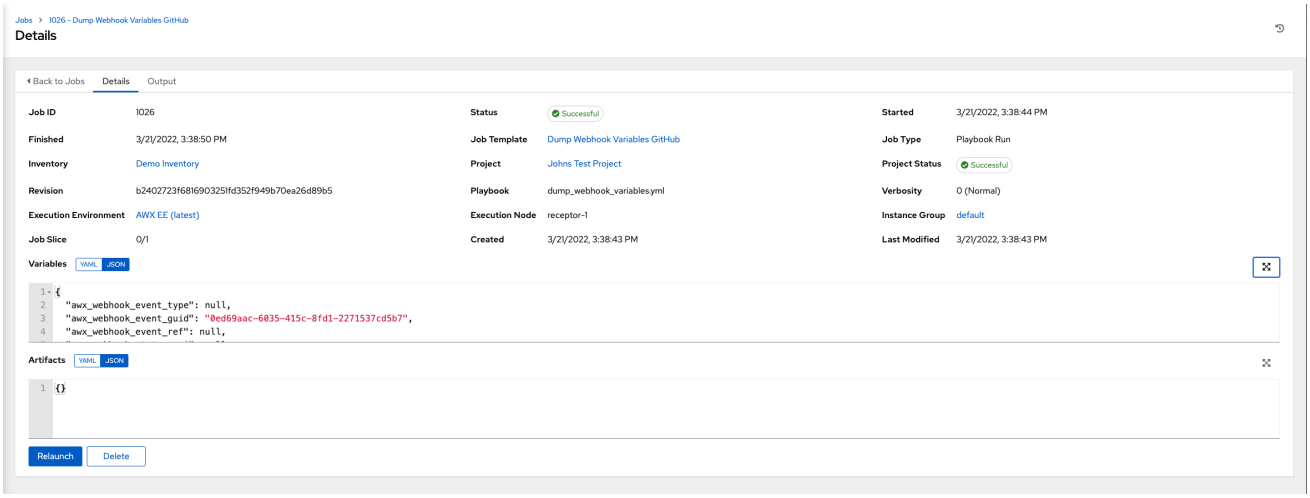
流程

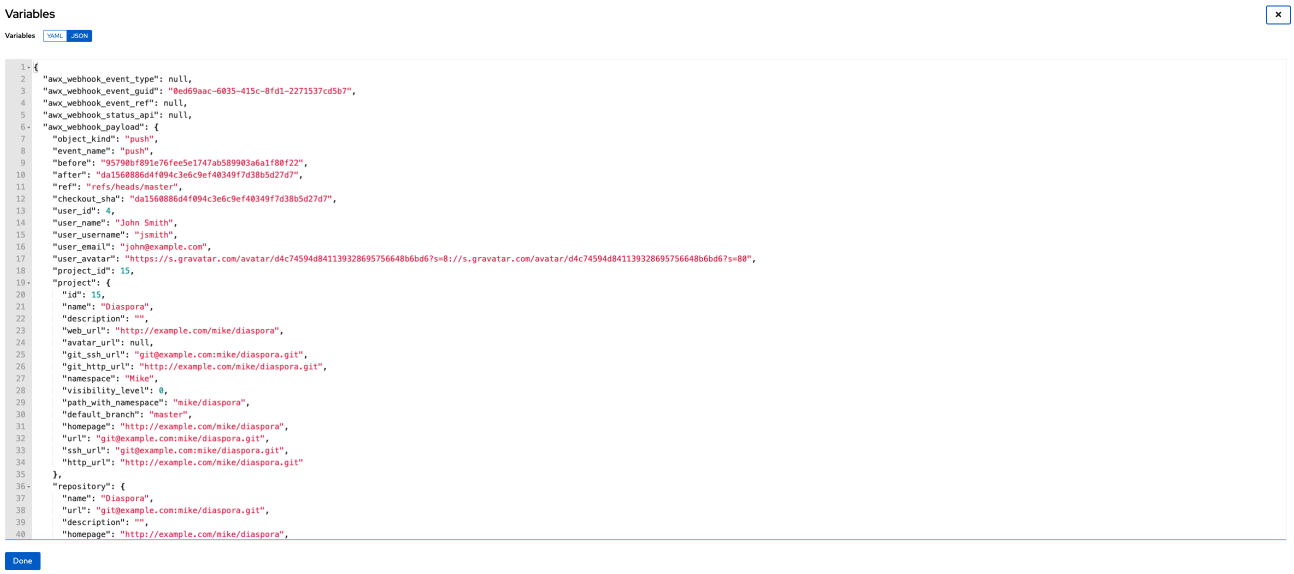
1. 在导航面板中，选择 **Views** → **Jobs**。

2. 选择启用了 **webhook** 的作业模板。

3. 选择 **Details** 选项卡。

4. 在 **Extra Variables** 字段中，查看 **awx_webhook_payload** 变量中的有效负载输出，如下例所示：





第 27 章 通知

通知类型 （如 **Email**、**Slack** 或 **Webhook**）是通知模板的实例，具有通知模板中定义的名称、描述和配置。

以下是添加通知模板所需的详情示例：

- 电子邮件通知模板需要用户名、密码、服务器和收件人
- **Slack** 通知模板需要令牌和频道列表
- **Webhook** 通知模板需要 **URL** 和标头

当作业失败时，将使用您在通知模板中定义的配置发送通知。

以下显示了通知系统的典型流程：

- 您可以通过 **API** 或 **UI** 创建指向 **/api/v2/notification_templates** 端点 的 **REST API** 的通知模板。
- 您可以将通知模板分配给支持它的各种对象（所有作业模板变体以及机构和项目）以及您想要通知的适当触发器级别（启动、成功或错误）。例如，您可能希望分配特定的通知模板，以便在作业模板 **1** 失败时触发。在这种情况下，您可以将通知模板与 **/api/v2/job_templates/n/notification_templates_error API** 端点的作业模板关联。
- 您可以在作业启动和作业结束时设置通知。用户和团队也可以定义他们自己的通知，这些通知可以附加到任意作业。

27.1. 通知层次结构

通知模板继承父对象上定义的模板，如下所示：

- 作业模板使用为它们定义的通知模板。此外，他们可以从作业模板使用的项目中继承通知模

板，以及它列在下面列出的机构中。

- 项目更新使用项目上定义的通知模板，并从与其关联的机构中继承通知模板。
- 清单更新使用在下面列出的机构上定义的通知模板。
- 临时命令使用与清单关联的组织上定义的通知模板。

27.2. 通知工作流

当作业成功或失败时，错误或成功处理程序使用 **Notifications** 部分中定义的步骤拉取相关通知模板列表。

然后，它会为每个对象创建一个通知对象，其中包含作业的相关详情并将其发送到目的地。这包括电子邮件地址、**slack** 频道和 **SMS** 号。

这些通知对象作为作业类型上的相关资源（作业、清单更新、项目更新、项目更新）以及 **/api/v2/notifications** 提供。您还可以通过检查其相关资源来查看从通知模板发送了哪些通知。

如果通知失败，它不会影响与其关联的作业，或导致它失败。通知的状态可以在其详情端点 **/api/v2/notifications/<n>** 中查看。

27.3. 创建通知模板

使用以下步骤创建通知模板。

流程

1. 在导航面板中，选择 **Administration** → **Notifications**。
2. 点**Add**。

3. 完成以下字段：

   - **Name** ：输入通知的名称。
   - 描述 ：输入通知的描述。此字段是可选的。
   - **Organization:** 指定通知所属的机构。
   - 类型 ：从下拉菜单中选择通知类型。如需更多信息，请参阅 通知类型 部分。

4. 点击 **Save**。

27.4. 通知类型

自动化控制器支持以下通知类型：

- 电子邮件
- **Grafana**
- **IRC**
- **Mattermost**
- **PagerDuty**
- **Rocket.Chat**

- Slack
- Twilio
- Webhook
  - Webhook 有效负载

每个通知类型都有自己的配置和行为语义。您可能需要以不同的方式进行测试。另外，您可以将每种通知类型自定义为特定的详情，或一组用于触发通知的条件。

其他资源

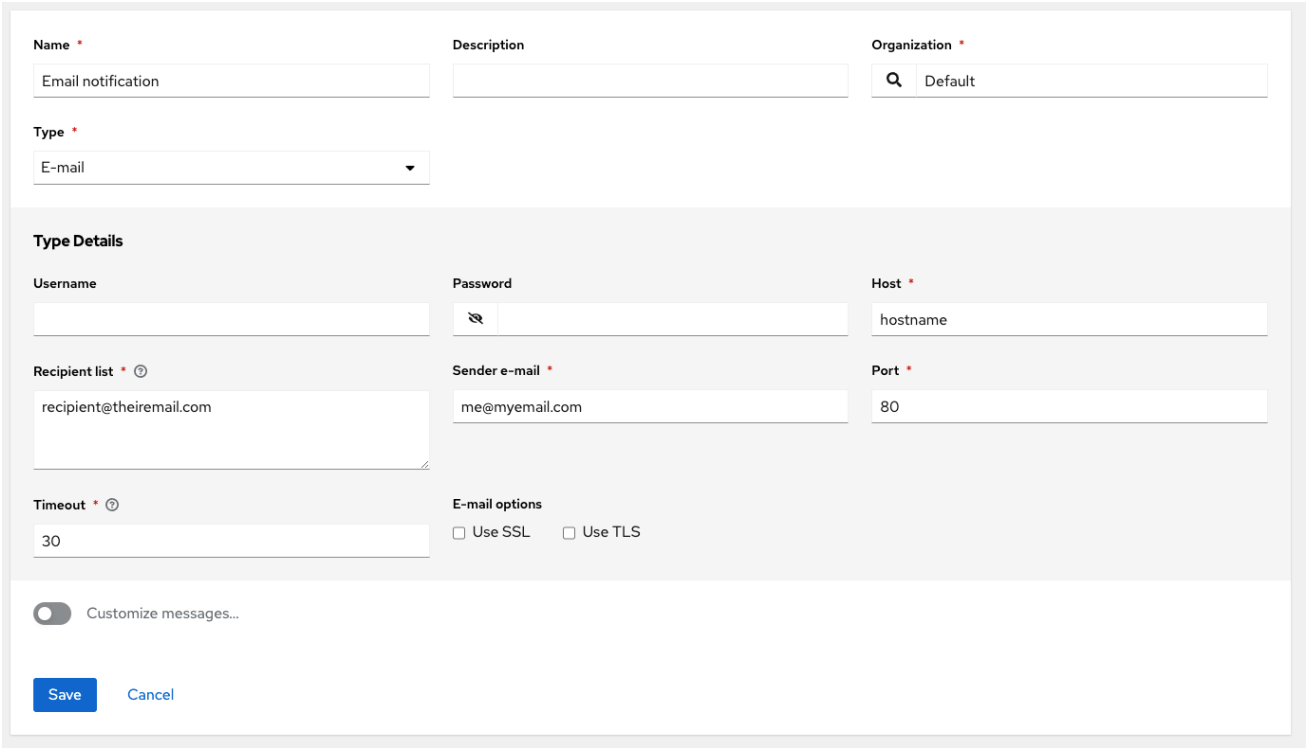
有关配置自定义通知的更多信息，请参阅创建自定义通知。以下小节详细介绍了每种通知类型。

27.4.1. 电子邮件

电子邮件通知类型支持各种 **SMTP** 服务器，并支持 **SSL/TLS** 连接。

提供以下详情来设置电子邮件通知：

- **Host**
- 接收者列表
- 发件人电子邮件
- **Port**
- **Timeout** （以秒为单位）：使您能够指定最多 **120** 秒，自动化控制器在失败前尝试连接到电子邮件服务器的时间长度。



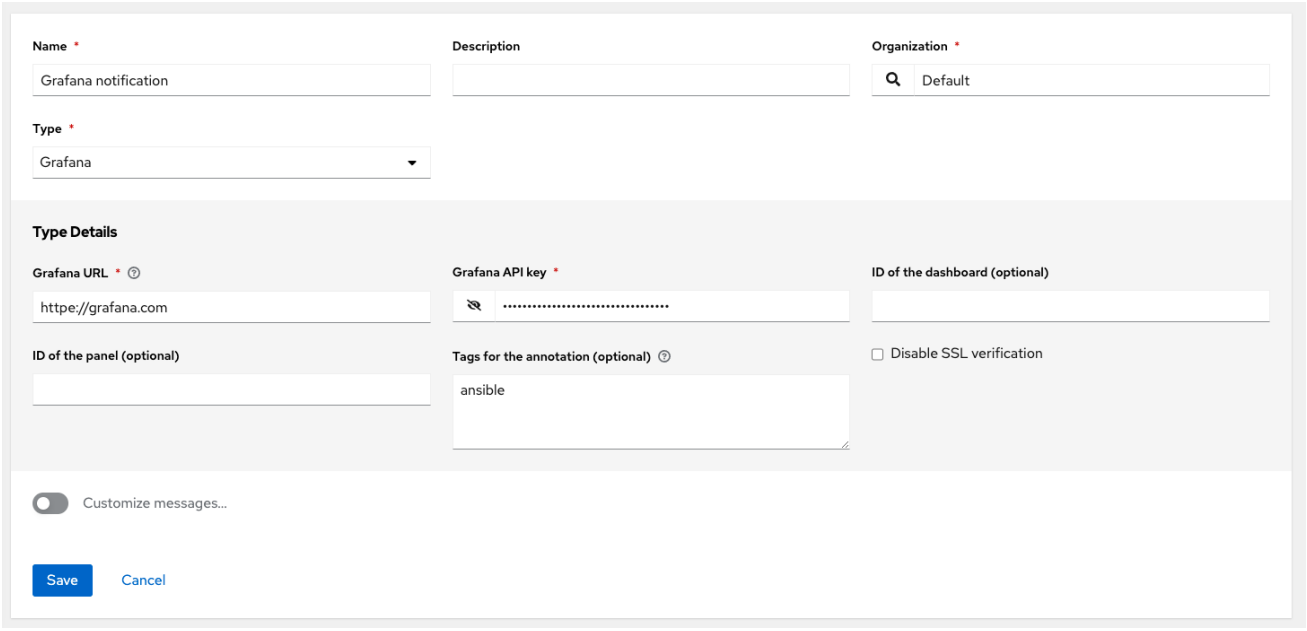
27.4.2. Grafana

要集成 **Grafana**，您必须首先在 **Grafana** 系统中创建 **API** 密钥。这是提供给自动化控制器的令牌。

提供以下详情来设置 **Grafana** 通知：

- **Grafana URL** ：**Grafana API** 服务的 **URL**，例如： **http://yourcompany.grafana.com**。
- **Grafana API** 密钥 ：您必须首先在 **Grafana** 系统中创建一个 **API** 密钥。
- 可选： 仪表板 **ID** ：当您为 **Grafana** 帐户创建 **API** 密钥时，您可以使用自己的唯一 **ID** 设置仪表板。
- 可选： 面板 **ID** ：如果您向 **Grafana** 界面添加了面板和图形，您可以在这里指定其 **ID**。
- 可选： 注解 的标签 **:** 输入关键字，以帮助识别您要配置的通知类型。
-

禁用 **SSL** 验证 ：默认情况下 **SSL** 验证是开启的，但您可以选择关闭验证目标证书真实性的功能。选择这个选项来禁用使用内部或私有 **CA** 的环境验证。



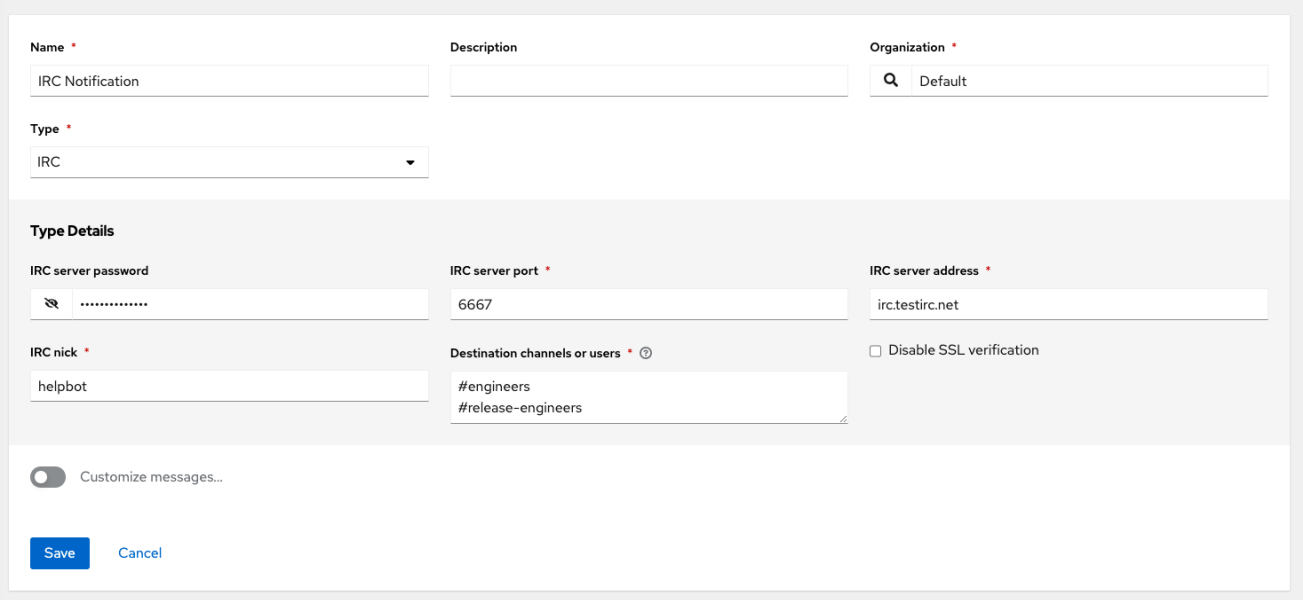
27.4.3. IRC

IRC 通知采用 **IRC bot** 的形式，它连接，将消息传送到频道或单个用户，然后断开连接。通知 **bot** 还支持 **SSL** 身份验证。**bot** 目前不支持 **Nickserv** 身份识别。如果频道或用户不存在或者没有在线，通知会失败。故障场景是为连接而保留的。

提供以下详情来设置 **IRC** 通知：

- 可选： **IRC** 服务器密码 ：**IRC** 服务器可能需要密码才能连接。如果服务器不需要，请将其留空。**IRC** 服务器端口 ：**IRC** 服务器端口。**IRC** 服务器地址 ：**IRC** 服务器的主机名或地址。**IRC nick** ：**bot** 在连接到服务器后的别名。目标频道或用户 ：发送通知的用户或频道列表。

- 可选： 禁用 **SSL** 验证 ：检查是否希望 **bot** 在连接时使用 **SSL**。

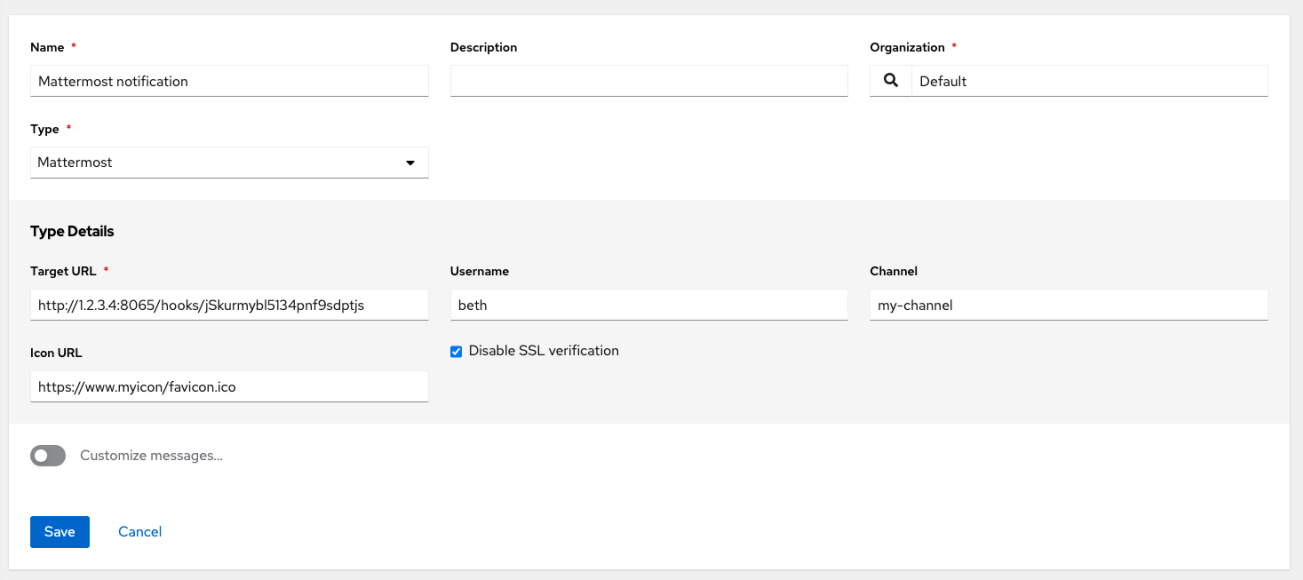


27.4.4. Mattermost

Mattermost 通知类型为 **Mattermost** 的消息和协作工作区提供了一个简单的接口。

提供以下详情来设置 **Mattermost** 通知：

- 目标 **URL** ：发布的完整 **URL**。
- 可选： 用户名 ：输入通知的用户名。
- 可选 ： 输入通知频道。
- **Icon URL:** 指定为这个通知显示的图标。
- 禁用 **SSL** 验证 ：关闭验证目标证书真实性的功能。选择这个选项来禁用使用内部或私有 **CA** 的环境验证。

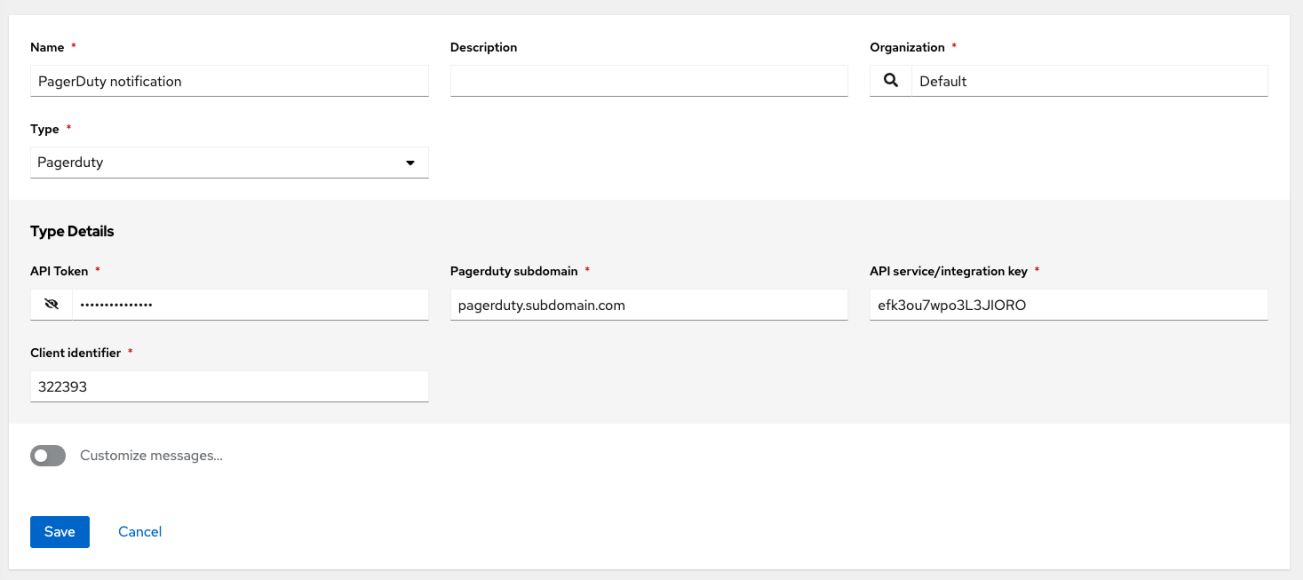


27.4.5. PagerDuty

要集成 **PagerDuty**，您必须首先在 **PagerDuty** 系统中创建一个 **API** 密钥。这是提供给自动化控制器的令牌。然后创建一个 提供 自动化控制器的 **Integration Key** 的服务。

提供以下详情来设置 **PagerDuty** 通知：

- **API Token** ：您必须首先在 **PagerDuty** 系统中创建一个 **API** 密钥。这是提供给自动化控制器的令牌。
- **PagerDuty** 子域 ：当您注册 **PagerDuty** 帐户时，您会收到一个用来进行通信的唯一子域。例如，如果您以**"testuser"**身份注册，**Web** 仪表板位于 **testuser.pagerduty.com**，并且为您提供 **API testuser** 作为子域，而不是完整的域。
- **API** 服务/集成 密钥 ：输入 **PagerDuty** 中创建的 **API** 服务/集成密钥。
- 客户端标识符 ：这与警报内容一同发送到 **PagerDuty** 服务，以帮助识别使用 **API** 密钥和服务的服务。如果多个集成使用相同的 **API** 密钥和服务，这非常有用。

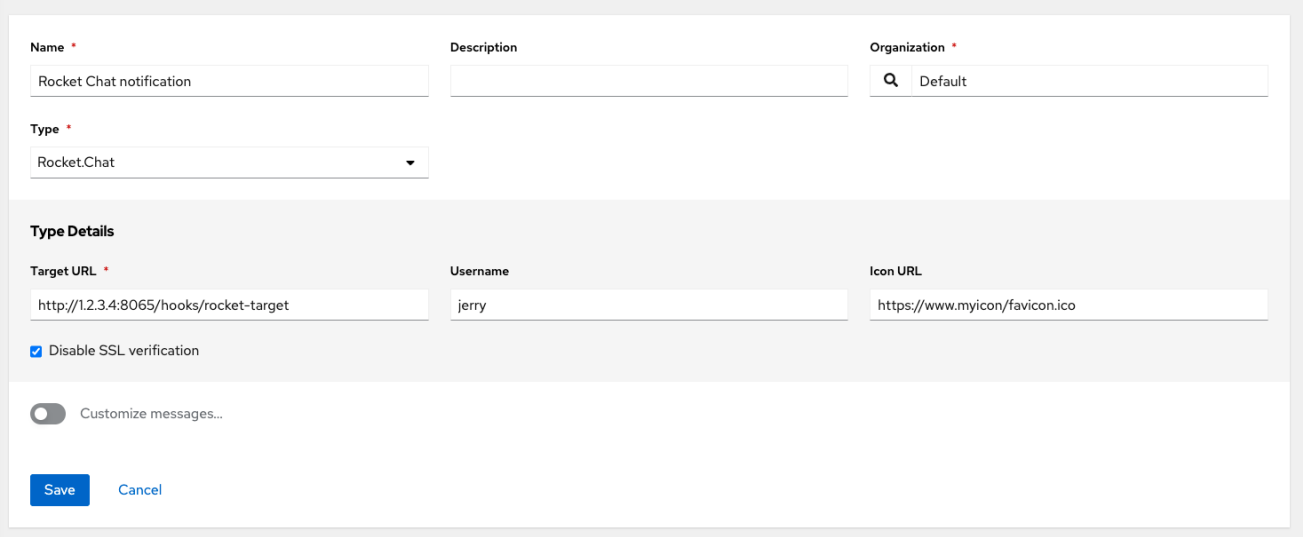


27.4.6. Rocket.Chat

Rocket.Chat 通知类型为 **Rocket.Chat** 的协作和通信平台提供了一个接口。

提供以下详情来设置 **Rocket.Chat** 通知：

- 目标 **URL** ： **POST** 到的完整 **URL**。
- 可选： 用户名 ：输入用户名。
- 可选： **Icon URL:** 指定为这个通知显示的图标
- **Disable SSL Verification** ：关闭验证目标证书真实性的功能。选择这个选项来禁用使用内部或私有 **CA** 的环境验证。



27.4.7. Slack

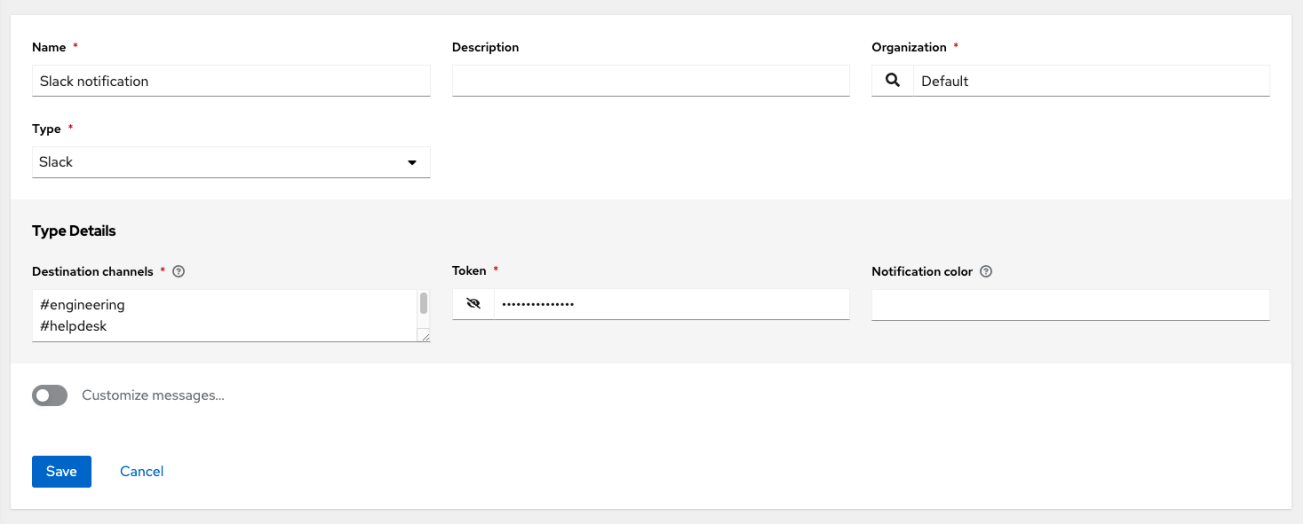
Slack 是一个协作团队通信和消息传递工具。

提供以下详情来设置 **Slack** 通知：

- **Slack** 应用程序。如需更多信息，请参阅 **Slack** 文档中的 **Quickstart** 页面。
- 令牌。如需更多信息，请参阅 **Current token type** 文档 页中的 **Legacy bots** 以及 **bot** 令牌令牌的特定详情。

当您设置 **bot** 或应用程序时，您必须完成以下步骤：

1. 导航到 **Apps**。
2. 点新创建的应用程序，然后进入 **Add features and functionality**，它可让您配置传入的 **webhook**、**bot** 和 **permissions**，并将您的应用程序安装到工作区中。



27.4.8. Twilio

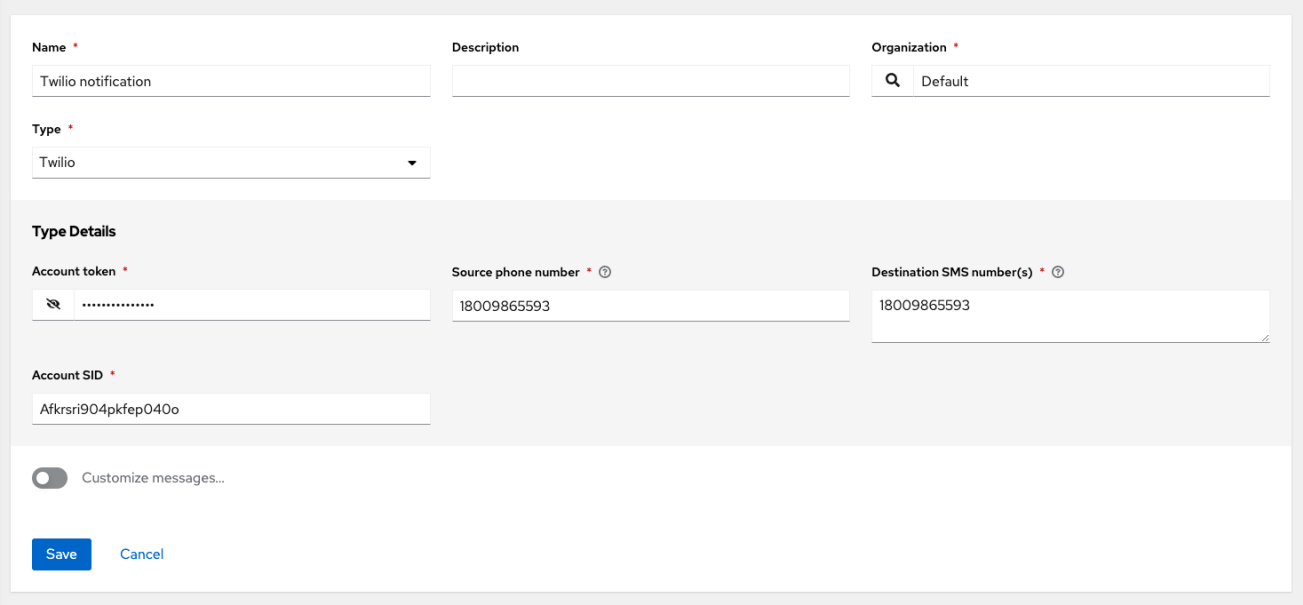
Twilio 是一个语音和 **SMS** 自动化服务。当您签名后，您必须创建一个从其中发送消息的电话号码。然后，您可以在 **Programmable SMS** 下定义一个 消息传递服务，并将之前创建的号码与其相关联。

在被允许使用它发送到任何数字之前，您可能需要验证这个数字或一些其他信息。消息传递服务 不需要状态回调 **URL**，不需要处理入站消息。

在您的单独（或子）帐户设置下，您有 **API** 凭证。**Twilio** 使用两个凭证来确定来自哪个帐户。帐户 **SID** 充当用户名，以及作为密码的 **Auth Token**。

提供以下详情来设置 **Twilio** 通知：

- 帐户令牌 ：输入帐户令牌。
- 源电话号码 ：以**"+15556667777"**的形式输入与消息传递服务关联的号码。
- **Destination SMS number (s):** 输入您要接收 **SMS** 的数字列表。它必须是 **10** 位的电话号码。
- 帐户 **SID** ：输入帐户 **SID**。



27.4.9. Webhook

webhook 通知类型提供了一个简单接口，用于将 **POST** 发送到预定义的 **Web** 服务。自动化控制器使用应用程序和 **JSON** 内容类型以及包含 **JSON** 格式的相关详情的数据有效负载 **POST** 到此地址。有些 **Web** 服务 **API** 预期 **HTTP** 请求采用特定格式，带有特定字段。

使用以下命令配置 **webhook** 通知：

- 使用 **POST** 或 **PUT** 配置 **HTTP** 方法。
- 传出请求的正文。
- 使用基本身份验证配置身份验证。

提供以下详情来设置 **Webhook** 通知：

- 可选： 用户名 ：输入用户名。
- 可选： 基本身份验证密码 ：

- 目标 **URL** ：输入 **webhook** 通知为 **PUT** 或 **POSTed** 的完整 **URL**。

- **Disable SSL Verification** ：默认情况下 **SSL** 验证是开启的，但您可以选择关闭验证目标证书真实性的功能。选择这个选项来禁用使用内部或私有 **CA** 的环境验证。

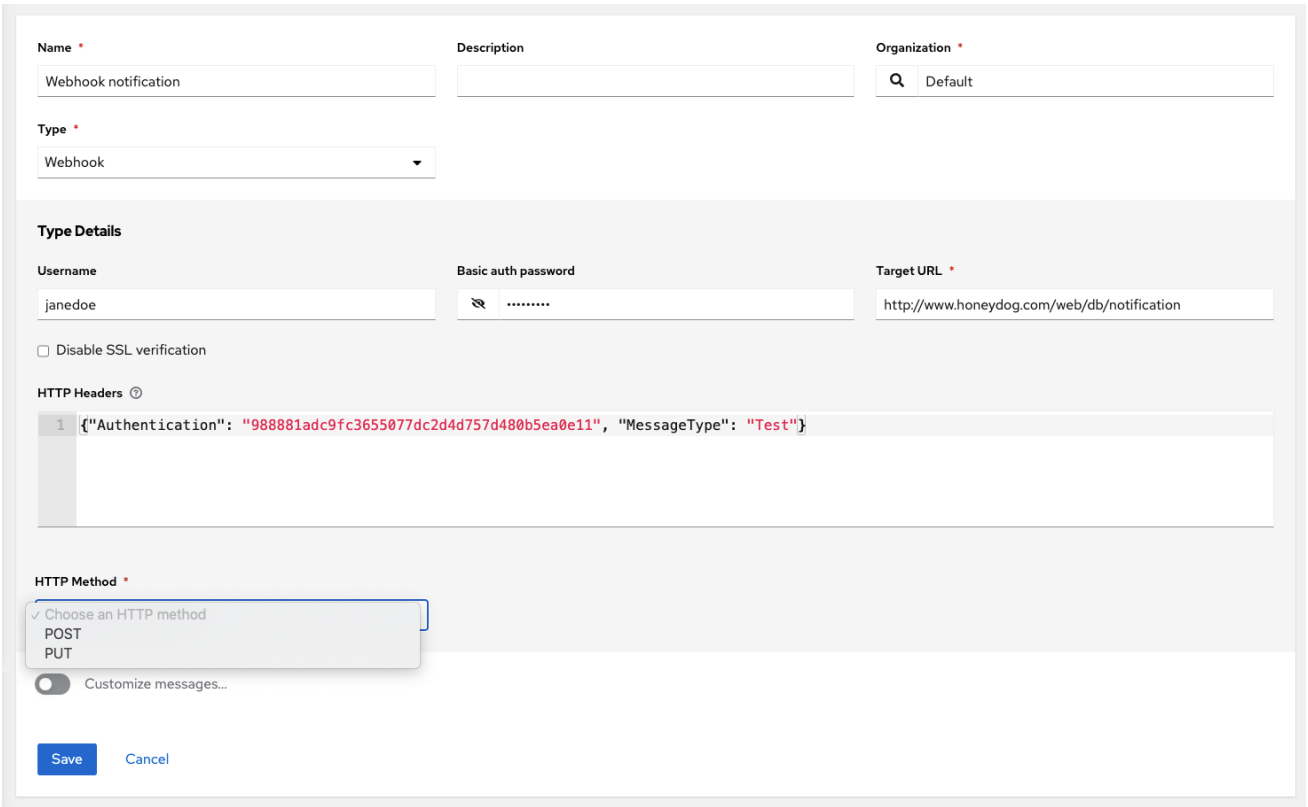
- **HTTP Headers** ：以 **JSON** 格式输入标头，其中键和值是字符串。例如：

```
{"Authentication": "988881adc9fc3655077dc2d4d757d480b5ea0e11", "MessageType": "Test"}`.
```

- **HTTP 方法** ：选择 **Webhook** 的方法：

- **POST** ：创建新资源。它还作为操作的一个捕获全部，它们不适合于其他类别。除非您知道 **Webhook** 服务需要 **PUT**，否则您可能需要 **POST**。

- **PUT** ：更新特定资源（按标识符）或资源集合。如果事先知道资源标识符，也可以使用 **PUT** 来创建特定资源。



27.4.9.1. Webhook 有效负载

自动化控制器在 **Webhook** 端点发送以下数据：

```
job id
name
url
created_by
started
finished
status
traceback
inventory
project
playbook
credential
limit
extra_vars
hosts
http method
```

以下是通过自动化控制器返回的 **webhook** 消息的 启动 通知示例：

```
{"id": 38, "name": "Demo Job Template", "url": "https://host/#/jobs/playbook/38",
"created_by": "bianca", "started":
"2020-07-28T19:57:07.888193+00:00", "finished": null, "status": "running", "traceback": "",
"inventory": "Demo Inventory",
"project": "Demo Project", "playbook": "hello_world.yml", "credential": "Demo Credential",
"limit": "", "extra_vars": "{}",
"hosts": {}}POST / HTTP/1.1
```

自动化控制器会在 **Webhook** 端点中 返回成功/失败状态 的数据：

```
job id
name
url
created_by
started
finished
status
traceback
inventory
project
playbook
credential
limit
extra_vars
hosts
```

以下是自动化控制器通过 **Webhook** 消息返回的 成功/失败 通知示例：

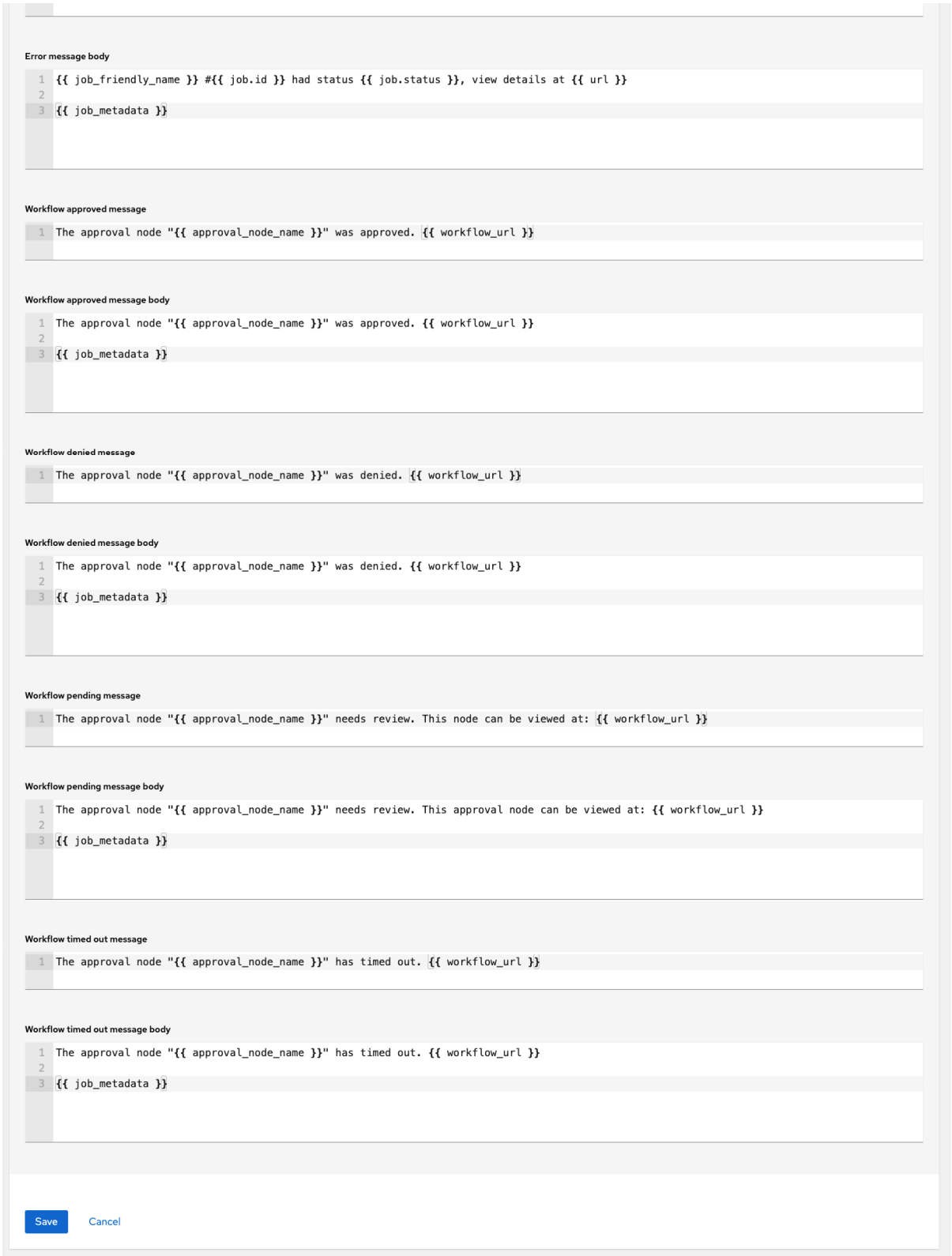
```
{"id": 46, "name": "AWX-Collection-tests-awx_job_wait-long_running-XVFBGRSAvUUlrYKn",
"url": "https://host/#/jobs/playbook/46",
"created_by": "bianca", "started": "2020-07-28T20:43:36.966686+00:00", "finished": "2020-07-
28T20:43:44.936072+00:00", "status": "failed",
"traceback": "", "inventory": "Demo Inventory", "project": "AWX-Collection-tests-
awx_job_wait-long_running-JJSlgInwtsRJyQmw", "playbook":
"fail.yml", "credential": null, "limit": "", "extra_vars": "{\"sleep_interval\": 300}", "hosts":
{"localhost": {"failed": true, "changed": 0,
"dark": 0, "failures": 1, "ok": 1, "processed": 1, "skipped": 0, "rescued": 0, "ignored": 0}}}
```

27.5. 创建自定义通知

您可以在通知表单上 自定义每个 通知类型 的文本内容。

流程

1. 在 **Notification Templates** 列表视图中，单击 **Add**。
2. 从 **Type** 列表中选择通知类型。
3. 使用切换启用 自定义消息。



4. 您可以为各种作业事件提供自定义消息，如下所示：

- 开始消息
- 成功消息正文

- 错误信息
- 工作流已批准消息
- 工作流拒绝消息
- 工作流运行消息
- 工作流待处理消息
- 工作流超时消息

消息表单因您要配置的通知类型而异。例如，电子邮件和 **PagerDuty** 通知的消息显示为典型的电子邮件，其中包含正文和主题，在这种情况下，自动化控制器会将字段显示为 **Message** 和 **Message Body**。其他通知类型仅针对每种事件要求 消息。

Message 字段预先填充一个模板，其中包含顶层变量，并与属性合并，如 **id** 或 **name**。模板用大括号括起，可以从自动化控制器提供的固定字段集合中提取，如预先填充的消息字段所示：

这个预先填充的字段建议通常向接收事件通知的接收者显示信息。您可以根据需要为作业添加自己的属性来自定义这些消息。自定义通知消息使用 **Jinja** 呈现；**Ansible playbook** 使用的同一模板引擎。

消息和消息正文具有不同的内容类型，如下所示：

- 信息始终只是字符串，仅一行。不支持新行。
- 消息正文可以是字典或文本块：
  -

Webhooks 和 **PagerDuty** 的消息正文使用字典定义。这些的默认消息正文是 **{{ job_metadata }}**，您可以保留原样的值，或者提供自己的字典。

- 电子邮件的消息正文使用文本块或多行字符串。默认消息正文为：

> **{{ job_friendly_name }} #{{ job.id }} had status {{ job.status }}, view details at {{ url }} {{ job_metadata }}**

您可以编辑此文本，使 **{{ job_metadata }}** 留在，或者丢弃 **{{ job_metadata }}**。由于正文是文本块，它可以是您想要的任何字符串。**{{ job_metadata }}** 呈现为包含描述正在执行作业的字段的字典。在所有情况下，**{{ job_metadata }}** 包括以下字段：

- **id**
- **name**
- **url**
- **created_by**
- **started**
- **finished**
- **status**
- 追溯

注意

您无法查询 **{{ job_metadata }}** 中的各个字段。当您在通知模板中使用 **{{ job_metadata }}** 时，所有数据都会被返回。

生成的字典类似如下：

```
{"id": 18,
 "name": "Project - Space Procedures",
 "url": "https://host/#/jobs/project/18",
 "created_by": "admin",
 "started": "2019-10-26T00:20:45.139356+00:00",
 "finished": "2019-10-26T00:20:55.769713+00:00",
 "status": "successful",
 "traceback": ""
}
```

如果 **{{ job_metadata }}** 在作业中呈现，它将包括以下附加字段：

- 清单（**inventory**）
- **project**
- **playbook**
- **credential**
- **limit**
- **extra_vars**
- 主机

生成的字典类似如下：

```
{"id": 12,
 "name": "JobTemplate - Launch Rockets",
 "url": "https://host/#/jobs/playbook/12",
 "created_by": "admin",
 "started": "2019-10-26T00:02:07.943774+00:00",
 "finished": null,
 "status": "running",
```

```
 "traceback": "",
 "inventory": "Inventory - Fleet",
 "project": "Project - Space Procedures",
 "playbook": "launch.yml",
 "credential": "Credential - Mission Control",
 "limit": "",
 "extra_vars": "{}",
 "hosts": {}
}
```

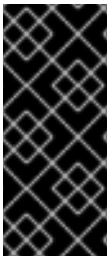

如果 **{{ job_metadata }}** 在工作流作业中呈现，它将包括以下附加字段：

- **body** （此枚举工作流作业中的节点，并包含与每个节点关联的作业描述）

生成的字典类似如下：


```
{"id": 14,
 "name": "Workflow Job Template - Launch Mars Mission",
 "url": "https://host/#/workflows/14",
 "created_by": "admin",
 "started": "2019-10-26T00:11:04.554468+00:00",
 "finished": "2019-10-26T00:11:24.249899+00:00",
 "status": "successful",
 "traceback": "",
 "body": "Workflow job summary:

        node #1 spawns job #15, \"Assemble Fleet JT\", which finished with status
successful.
        node #2 spawns job #16, \"Mission Start approval node\", which finished
with status successful.\n
        node #3 spawns job #17, \"Deploy Fleet\", which finished with status
successful."
}
```


如果您创建使用无效语法或引用不可用字段的通知模板，则会显示一条错误消息，指出错误的性质。如果您删除通知的自定义消息，则会在其位置显示默认消息。



重要

如果您在不编辑自定义消息（或编辑并恢复到默认值）的情况下保存通知模板，**Details** 屏幕会假定默认值，且不会显示自定义消息表。如果您编辑并保存任何值，则整个表会显示在 **Details** 屏幕中。

其他资源

- 如需更多信息，请参阅使用带有 **Jinja2** 的变量。

- 自动化控制器需要有效的语法来检索正确的数据来显示消息。有关支持的属性和正确的语法构建列表，请参阅 **Custom Notifications** 支持的属性 部分。

27.6. 启用和禁用通知

您可以将通知设置为在特定作业启动时通知您，并在作业运行结束时出现成功或失败。请注意以下行为：

- 如果工作流作业模板启动时启用了通知，并且该工作流中的作业模板也启用了启动时通知，您会收到这两者的通知。

- 您可以启用在工作流作业模板中的多个作业模板上运行通知。

- 您可以启用在分片作业模板启动中运行的通知，每个分片都会生成通知。

- 当您启用在作业启动时运行通知并且通知被删除时，作业模板将继续运行，但会生成错误消息。

您可以从以下资源的 **Notifications** 选项卡中启用作业启动、作业成功和作业失败时通知，或者它们的组合：

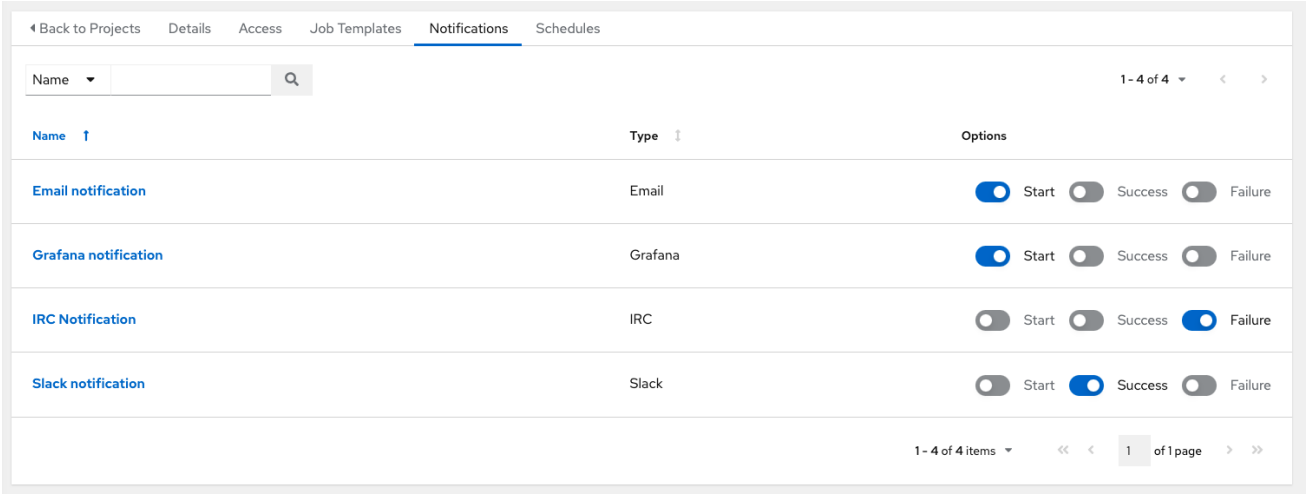
- 作业模板

- 工作流模板

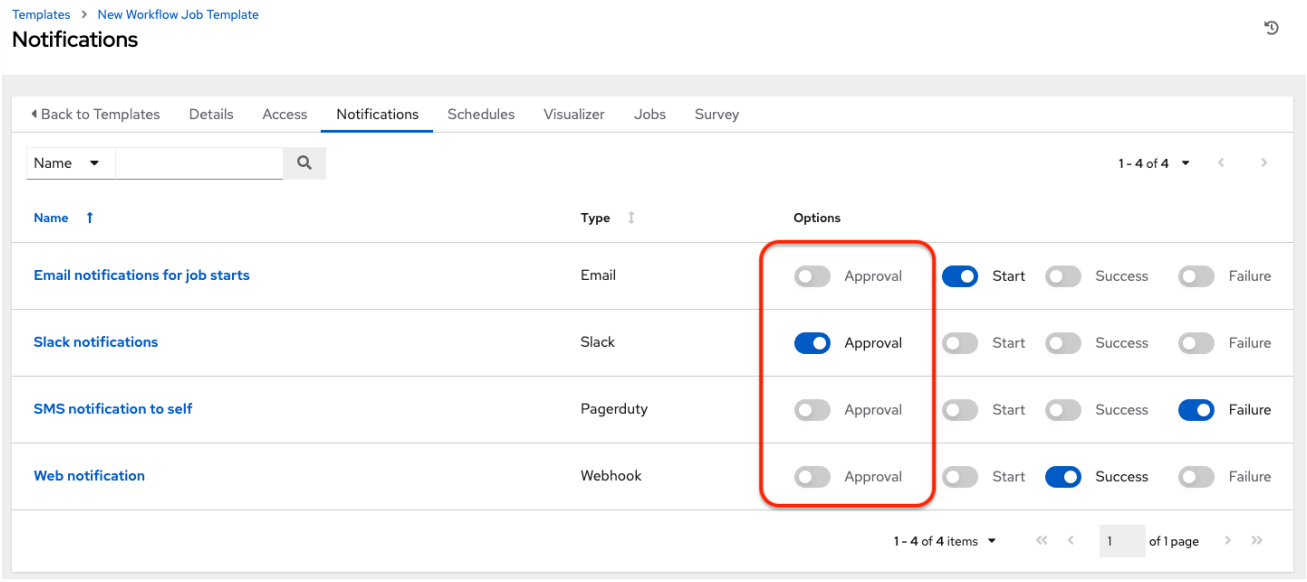
- 项目（如下例所示）

- 清单源

- 机构



对于具有批准节点的工作流模板，除了 启动、成功 和 **Failure** 外，您还可以启用或禁用某些与批准相关的事件：

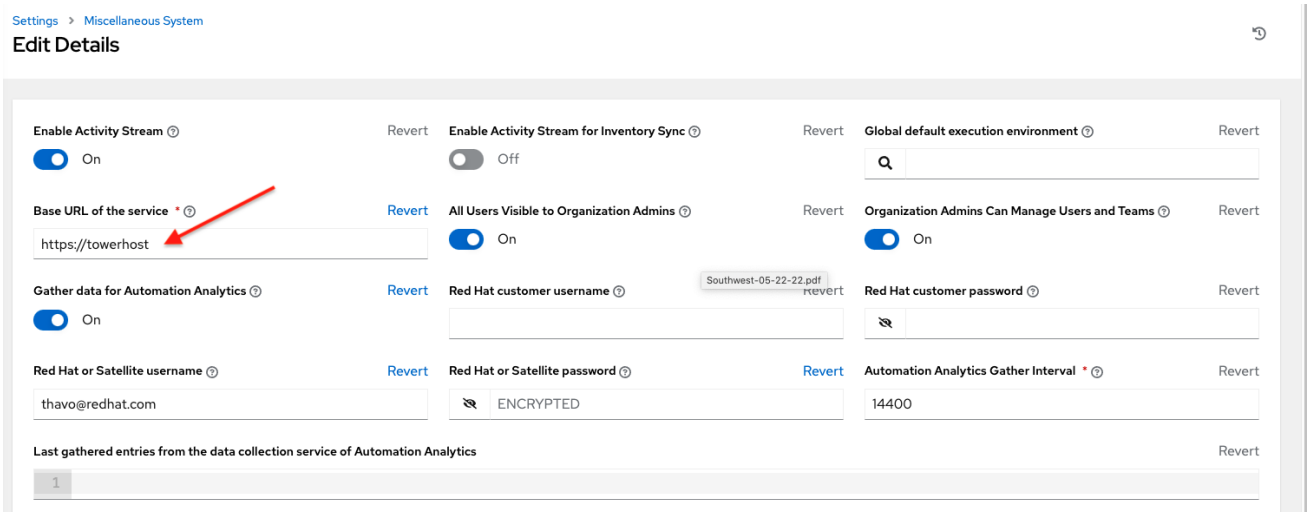


其他资源

有关使用这些节点类型的更多信息，请参阅 批准节点。

27.7. 为通知配置主机主机名

在 **System** 设置中，您可以将 **service** 字段的 **Base URL** 中的默认值替换为您首选的主机名，以更改通知主机名。



刷新您的许可证也会更改通知主机名。新的自动化控制器安装不必为通知设置主机名。

27.7.1. 重置 TOWER_URL_BASE

自动化控制器通过查看传入请求并根据该传入请求设置服务器地址来确定如何定义基本 **URL (TOWER_URL_BASE)**。

自动化控制器首先从数据库获取设置值。如果没有找到设置值，它将使用设置文件中的值。如果您通过进入到自动化控制器主机的 **IP** 地址发布许可证，发布的许可证将写入数据库中的设置条目。

如果选择了错误的地址，请使用以下步骤重置 **TOWER_URL_BASE** ：

流程

1. 在导航面板中，选择 **Settings**。
2. 从系统选项中选择 **Miscellaneous System settings**。
3. 点 **Edit**。
4. 在您要出现在通知的 **DNS** 条目的服务字段的 **Base URL** 中输入地址。
5. 在设置 订阅设置中 重新添加您的许可证。

27.8. 通知 API

使用已启动的 、 **success** 或 **error** 端点：

```
/api/v2/organizations/N/notification_templates_started/
/api/v2/organizations/N/notification_templates_success/
/api/v2/organizations/N/notification_templates_error/
```

另外，**../ .././N/notification_templates_started** 端点具有 **GET** 和 **POST** 操作：

- 机构
- 项目
- 清单源
- 作业模板
- 系统作业模板
- 工作流任务模板

第 28 章 自定义通知支持的属性

了解支持的作业属性列表以及构建通知消息文本的正确语法。

以下是支持的作业属性：

- **allow_simultaneous -** （布尔值）指示多个作业是否可以从与此作业关联的作业模板同时运行。
- **controller_node -** （字符串）管理隔离执行环境的实例。
- **created -** （日期时间）创建此作业时的时间戳。
- **custom_virtualenv -** （字符串）用于执行作业的自定义虚拟环境。
- **description -** （字符串）作业的可选描述。
- **diff_mode -** （布尔值）如果启用，标准输出中会显示对主机上任何模板文件进行的文本更改。
- **elapsed -(decimal)**作业运行经过的时间（以秒为单位）。
- **execution_node -** （字符串）作业执行的节点。
- **failed -** （布尔值）如果作业失败，则为 **True**。
- **finished -** （日期时间）作业完成执行的日期和时间。
- **force_handlers -** （布尔值）当处理程序被强制运行时，它们也会在通知时运行，即使该主机上的任务失败也是如此。请注意，一些条件（如不可访问的主机）仍然可以阻止处理程序运行。

- **forks -** （整数）此作业请求的 **fork** 数量。
- **id -** （整数）此作业的数据库 **ID**。
- **job_explanation -** （字符串）在无法运行和捕获 **stdout** 时指示作业状态的 **status** 字段。
- **job_slice_count -** （整数）如果作为分片作业的一部分运行，则这是分片（如果为 **1**，则作业不是分片作业的一部分）的总数。
- **job_slice_number -** （整数）如果作为分片作业的一部分运行，这是在其上操作的清单分片的 **ID** （如果不是分片作业的一部分，则不使用属性）。
- **job_tags -** （字符串）仅执行具有指定标签的任务。
- **job_TYPE -**（选择）这可以 可以运行、检查 或 **scan**。
- **launch_type -** （选择）这可以是 手动、重新启动、回调、计划、依赖项、工作流、同步 或 **scm**。
- **limit -** （字符串）如果指定，则 **playbook** 执行仅限于这组主机。
- **modified -** （日期时间）最后一次修改此作业的时间戳。
- **name -** （字符串）此作业的名称。
- **playbook -**（字符串）执行的 **playbook**。
- **scm_revision -** （字符串）用于此作业的项目中的 **scm** 修订（如果可用）。

- **skip_tags -** （字符串）如果指定，**playbook** 执行将跳过此组标签。
- **start_at_task -** （字符串）如果指定，**playbook** 执行从与此名称匹配的任务开始。
- **started-** （日期时间）作业加入启动队列的日期和时间。
- 状态 **-**（选择）可以是 新的、**pending**、**waiting**、**running**、**successful**、**failed**、**error** 或 **Canceled**。
- **timeout -** （整数）取消任务前运行的时间（以秒为单位）。
- **type -** （选择）此作业的数据类型。
- **url -** （字符串）此作业的 **URL**。
- **use_fact_cache -** （布尔值）如果已为作业启用，自动化控制器会在 **playbook** 运行到数据库和缓存事实的末尾充当 **Ansible** 事实缓存插件。
- **verbosity -** （选择）**0** 到 **5** （与 **Normal** 到 **WinRM Debug** 相对应）。
  - **host_status_counts** （分配给每个状态的唯一主机数量）
    - **skipped** （整数）
    - **ok** （整数）
    - **changed** （整数）
    - **failures** （整数）

- **dark** （整数）
- **processed** （整数）
- **rescued** （整数）
- **ignored** （整数）
- **failed** （布尔值）

- **summary_fields:**
  - 清单（**inventory**）
    - **id -** （整数）清单的数据库 **ID**。
    - **name -** （字符串）清单的名称。
    - **description -** （字符串）清单的可选描述。
    - **has_active_failures-** （布尔值）（已弃用）指示此清单中是否有主机失败的标记。
    - **total_hosts -** （已弃用）（整数）此清单中的主机总数。
    - **hosts_with_active_failures -** （已弃用）（整数）此清单中有活跃故障的主机数量。
    - **total_groups -** （已弃用）（整数）此清单中的组总数。

- **groups_with_active_failures -** （已弃用）（整数）此清单中有活跃故障的主机数量。
- **has_inventory_sources -** （已弃用）（布尔值）指明此清单是否具有外部清单源的标记。
- **total_inventory_sources -**（整数）在此清单中配置的外部清单源总数。
- **inventory_sources_with_failures -**（整数）此清单中有故障的外部清单源数量。
- **organization_id -(id)**包含此清单的机构。
- **kind -**（选择）（空字符串）（代表主机与清单有直接链接）或 **smart**

■ **project**

- **id -** （整数）项目的数据库 **ID**。
- **name -** （字符串）项目名称。
- **description** （字符串）项目的可选描述。
- **status -** （选择）
新、**pending**、**waiting**、**running**、**successful**、**failed**、**error**、**canceled**、**never updated**、**ok** 或 **missing** 之一。
- **scm_type** （选择） 其中一个（空字符串）、**git**、**hg**、**svn**、**insights.**

■ **job_template**

- **id -** （整数）作业模板的数据库 **ID**。
- **description -** （字符串）项目的可选描述。
- **status -** （选择）
新、**pending**、**waiting**、**running**、**successful**、**failed**、**error**、**canceled**、**never updated**、**ok** 或 **missing** 之一。

■ **job_template**

- **id-** （整数）作业模板的数据库 **ID**。
- **name-** （字符串）作业模板的名称。
- **description-** （字符串）作业模板的可选描述。

■ **unified_job_template**

- **id -**（整数）统一的作业模板的数据库 **ID**。
- **name -** （字符串）统一的作业模板的名称。
- **description -** （字符串）统一的作业模板的可选描述。
- **unified_job_type -(choice)**统一作业类型，如 作业、**workflow_job** 或 **project_update**。

■ **instance_group**

- **id -** （整数）实例组的数据库 **ID**。

- **name-** （字符串）实例组的名称。

- **created_by**
  - **id -(int)**启动操作的用户的数据库 **ID**。
  - **username -** （字符串）启动操作的用户名。
  - **first_name -** （字符串）名。
  - **last_name -(string)**姓氏。
- **labels**
  - **count -(int)**标签数。
  - **results -** 代表标签的字典列表。例如： **{"id": 5, "name": "database jobs"}**。

您可以使用分组大括号 **{{ }}** 在自定义通知消息中引用有关作业的信息。使用点表示法访问特定作业属性，如 **{{ job.summary_fields.inventory.name }}**。您可以添加在大括号或周围使用的任何字符，或纯文本，如 **":"** 用于作业 **ID**，单引号用于表示某些描述符。自定义消息可在整个消息中包含多个变量：

```
{{ job_friendly_name }} {{ job.id }} ran on {{ job.execution_node }} in {{ job.elapsed }} seconds.
```

以下是可添加到模板中的额外变量：

- **approval_node_name -** （字符串）批准节点名称。

- **approval_status -**（选择） 批准的、**denied** 和 **timed_out** 之一。

- **URL-** （字符串）发出通知的作业 **URL** （这适用于 启动、成功、失败 和 批准通知）。

- **workflow_url -** （字符串）相关批准节点的 **URL**。这允许通知接收者进入相关的工作流作业页面来检查这种情况。例如，可在以下位置查看此节点：**{{workflow_url }}**。在与批准相关的通知中，**url** 和 **workflow_url** 都相同。

- **job_friendly_name -** （字符串）作业的友好名称。

- **job_metadata -(string)**作业元数据作为 **JSON** 字符串，例如：

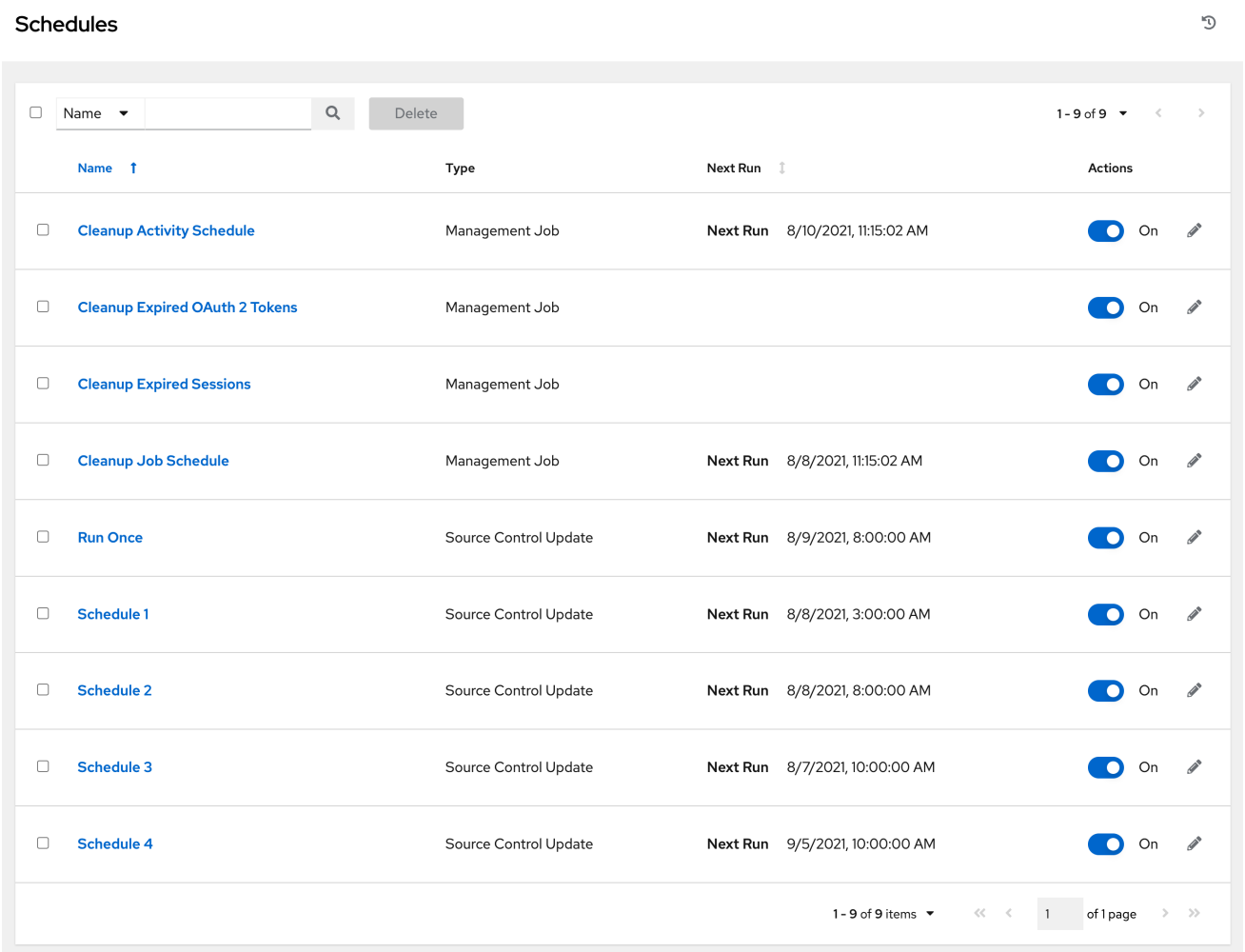
```
{'url': 'https://towerhost/$/jobs/playbook/13',
 'traceback': '',
 'status': 'running',
 'started': '2019-08-07T21:46:38.362630+00:00',
 'project': 'Stub project',
 'playbook': 'ping.yml',
 'name': 'Stub Job Template',
 'limit': '',
 'inventory': 'Stub Inventory',
 'id': 42,
 'hosts': {},
 'friendly_name': 'Job',
 'finished': False,
 'credential': 'Stub credential',
 'created_by': 'admin'}
```

第 29 章 调度

在导航面板中，单击 **Views** → **Schedules** 以访问您配置的计划。调度列表可以根据每个列中的任何属性使用方向箭头进行排序。您还可以按名称、日期或调度运行的月份名称进行搜索。

每个调度都有对应的 **Actions** 列，具有使用调度名称旁边的 **On** 或 **Off** 切换来启用或禁用该调度的选项。点 **Edit**

图标编辑调度。



如果要设置模板、项目或清单源，请点击 **Schedules** 选项卡来为这些资源配置调度。当您创建调度时，会通过以下方法列出它们：

Name

点计划名称打开其详情。

类型

这标识调度是否与源控制更新关联或系统管理的作业调度。

下次运行

此任务的下一次调度运行。

29.1. 添加新调度

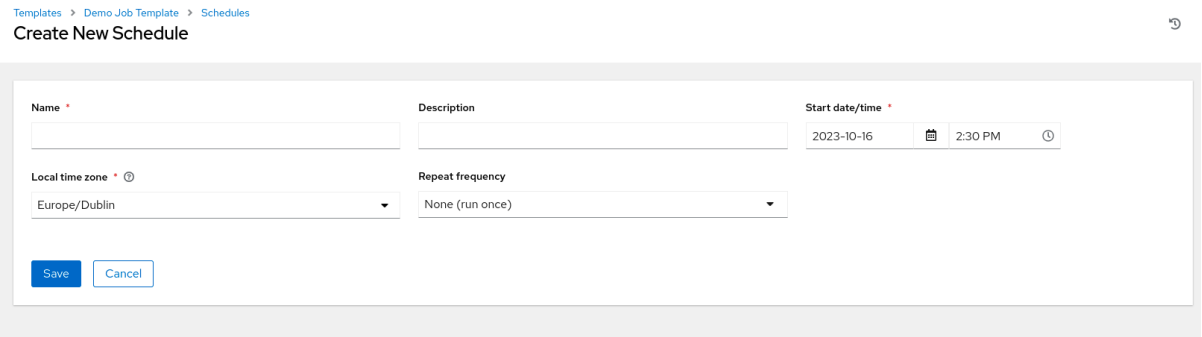
您只能从模板、项目或清单源创建调度，而不直接在主 **Schedules** 屏幕上创建。

创建新时调度：

流程

1. 单击您要配置的资源的 **Schedules** 选项卡。这可以是模板、项目或清单源。

2. 点**Add**。这将打开 **Create New Schedule** 窗口。



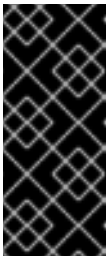
3. 在以下字段中输入相关信息：

- 名称 ：输入名称。
- 可选： 描述 ：输入描述。
- 开始日期/时间 ：输入开始计划的日期和时间。

- 本地时区 ：您输入的开始时间必须在此时区中。

- 重复频率 ：根据您选择的频率显示适当的调度选项。

Schedule Details 在建立调度时显示，供您查看调度设置以及所选本地时区中调度的发生次数列表。



重要

作业以 **UTC** 的形式调度。当夏时制发生时，在一天的特定时间运行的重复作业可能会针对本地时区有相应变化。在保存调度时，系统会将基于本地时区的时间解析为 **UTC**。要确保正确创建了您的调度，以 **UTC** 时间设置调度。

4. 点击 **Save**。

使用 **On** 或 **Off** 切换停止活跃的调度或激活已停止的调度。

第 30 章 为 RED HAT ANSIBLE AUTOMATION PLATFORM 修复设置 RED HAT INSIGHTS

自动化控制器支持与 **Red Hat Insights** 集成。

当使用 **Red Hat Insights** 注册主机时，它会持续扫描漏洞和已知的配置冲突。识别的每个问题都可以以 **Ansible playbook** 的形式有一个关联的修复。

Red Hat Insights 用户创建一个维护计划来对修复进行分组，并您可以创建一个 **playbook** 来缓解问题。自动化控制器通过 **Red Hat Insights** 项目跟踪维护计划 **playbook**。

通过基本授权向 **Red Hat Insights** 进行身份验证由特殊凭证支持，它必须首先在自动化控制器中建立。要运行 **Red Hat Insights** 维护计划，您需要一个 **Red Hat Insights** 项目和清单。

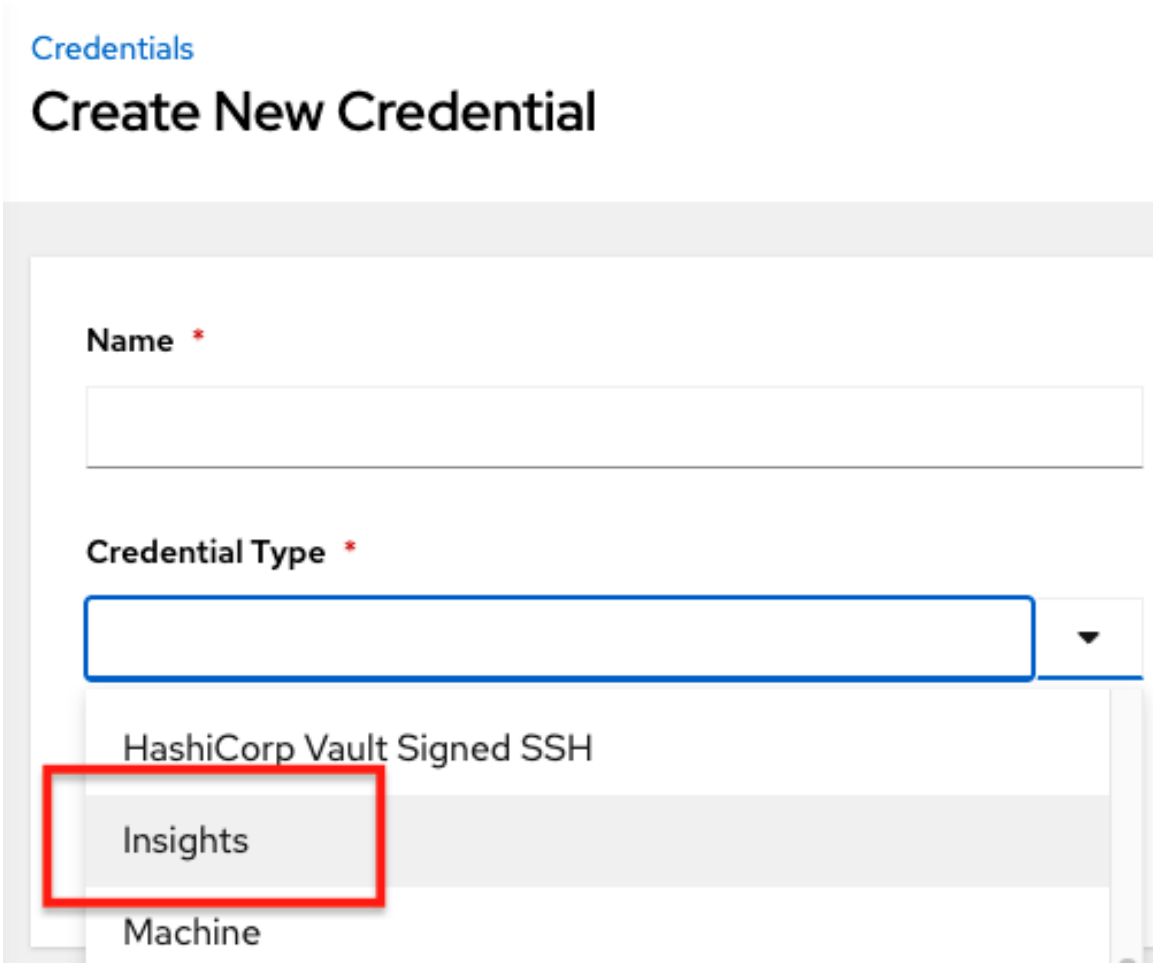
30.1. 创建 RED HAT INSIGHTS 凭证

使用以下步骤创建新凭证以用于 **Red Hat Insights**：

流程

1. 在导航面板中，选择 **Resources** → **Credentials**。
2. 点**Add**。
3. 在以下字段中输入相关信息：
   - 名称 ：输入凭证的名称。
   - 可选： 描述 ：输入凭证的描述。
   - 可选： 机构 ：输入与凭证关联的机构名称，或者点击搜索 图标并从 **Select organization** 窗口中选择它。

- 凭证类型 ：输入 **Insights** 或从列表中选择它。



- 用户名 ：输入有效的 **Red Hat Insights** 凭证。

- **Password:** 输入有效的 **Red Hat Insights** 凭证。**Red Hat Insights** 凭证是用户的 红帽客户门户网站帐户 用户名和密码。

4. 点击 **Save**。

30.2. 创建 RED HAT INSIGHTS 项目

使用以下步骤创建新项目以用于 **Red Hat Insights**：

流程

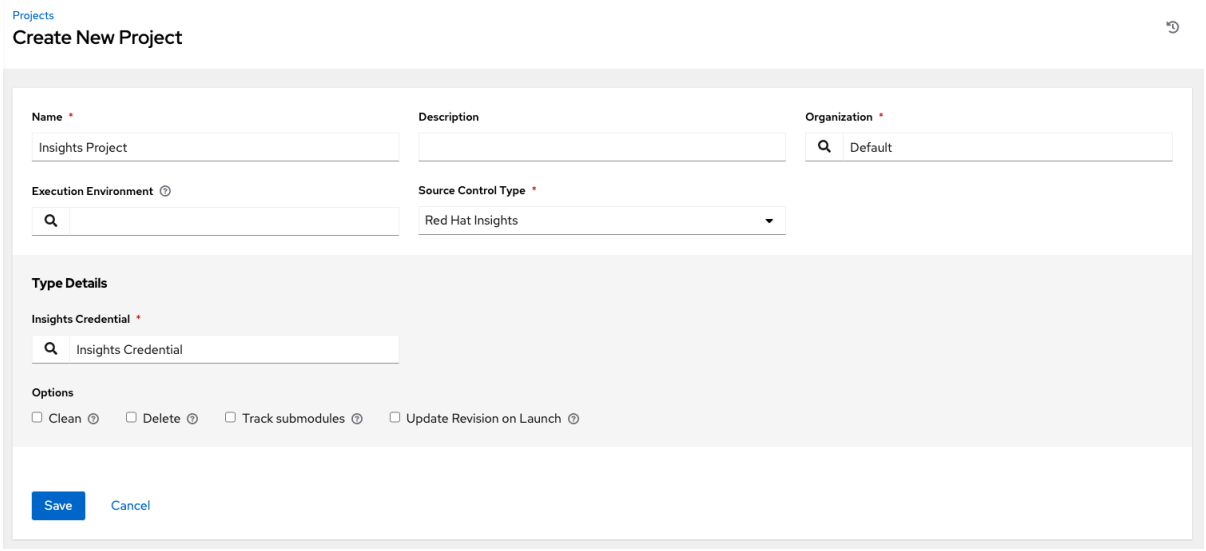
1. 在导航面板中，选择 **Resources** → **Projects**。

2. 点**Add**。

3. 在以下字段中输入相关详情。请注意，以下字段需要特定的 **Red Hat Insights** 相关条目：

- 名称 ：输入 **Red Hat Insights** 项目的名称。
- 可选： 描述 ：输入项目的描述。
- 机构 ：输入与凭证关联的机构名称，或者点击搜索()图标并从 **Select organization** 窗口中选择它。
- 可选： 执行环境 ：用于使用这个项目的作业的执行环境。
- 源控制类型 ：选择 **Red Hat Insights**。
- 可选： **Content Signature Validation Credential:** 启用内容签名以验证内容在项目同步时是否保持安全。
- **Insights** 凭证 ：这会预先填充您之前创建的 **Red Hat Insights** 凭证。如果没有，输入凭证，或者点击搜索图标并从 **Select Insights Credential** 窗口中选择它。

4. 从 **Options** 字段中选择此项目的更新选项，并提供任何其他值（如果适用）。有关每个选项的更多信息，请点每个选项旁的工具提示  图标。



5. 点击 **Save**。

您第一次保存新项目时，所有 **SCM** 和项目同步会自动进行。如果您希望它们被更新为 **Red Hat Insights** 中的当前内容，点项目可用操作下的更新

图标手动更新基于 **SCM** 的项目。

此过程会将 **Red Hat Insights** 项目与 **Red Hat Insights** 帐户解决方案同步。请注意，当同步运行后，项目名称旁的状态点会更新。

30.3. 创建 INSIGHTS 清单

Insights playbook 包含一个 **hosts:** 行，其中值是提供给红帽 **insights** 的主机名，它与提供给自动化控制器的主机名不同。

要创建用于 **Red Hat Insights** 的新清单，请参阅创建 **Insights** 凭证。

30.4. 修复 RED HAT INSIGHTS 清单

修复 **Red Hat Insights** 清单可让自动化控制器使用单个点击运行 **Red Hat Insights playbook**。您可以通过创建一个作业模板来运行 **Red Hat Insights** 修复。

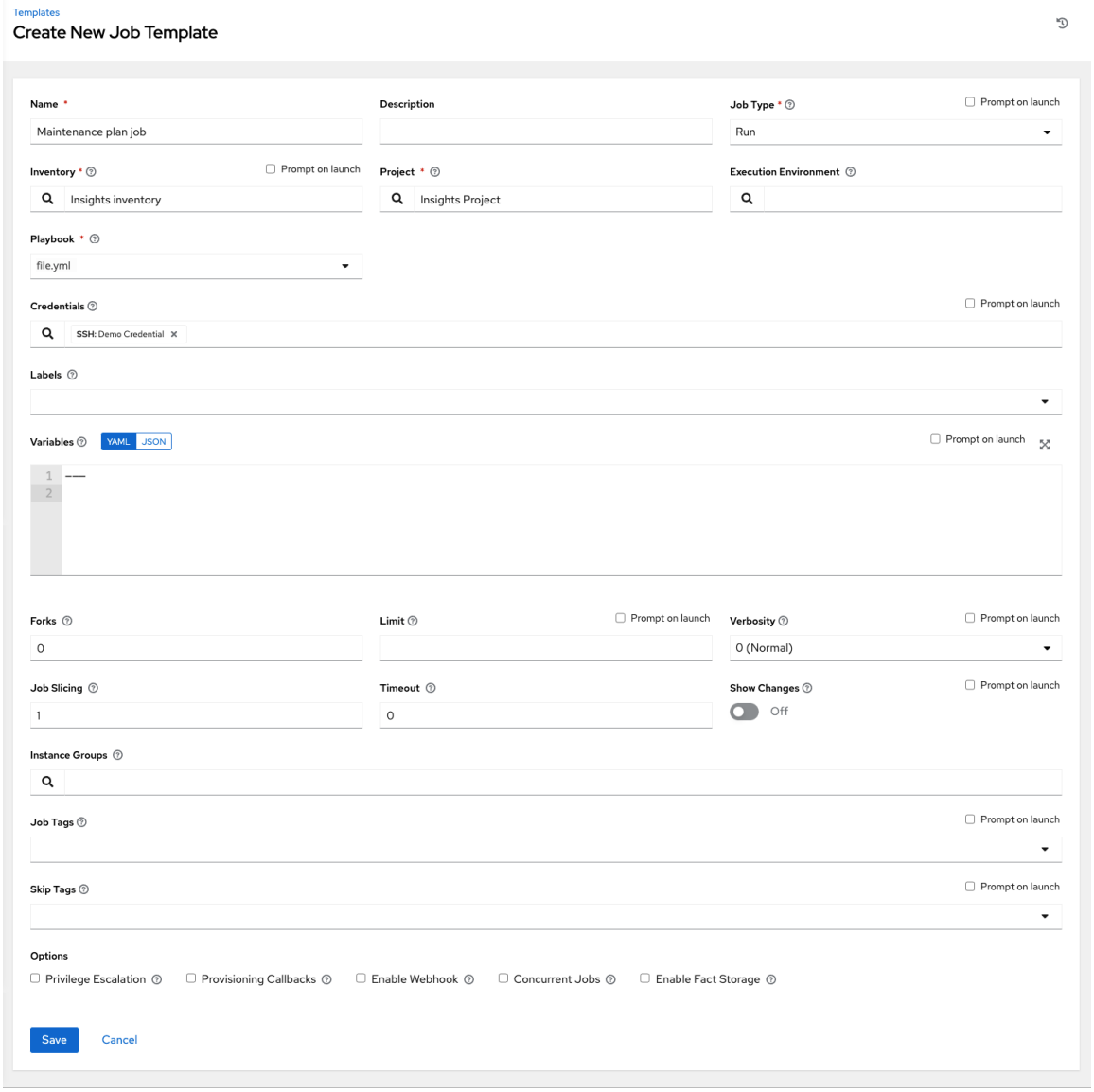
流程

1. 在导航菜单中，选择 **Resources** → **Templates**。

2. 在 **Templates** 列表视图中，从 **Add job template** 列表中选择 **Add**。

3. 在以下字段中输入相关详情。请注意，以下字段需要特定的 **Red Hat Insights** 相关条目：

- 名称 ：输入维护计划的名称。
- 可选： 描述 ：输入作业模板的描述。
- **Job Type:** 如果还没有填充，请从作业类型列表中选择 **Run**。
- 清单 ：选择您之前创建的 **Red Hat Insights** 清单。
- 项目 ：选择您之前创建的 **Red Hat Insights** 项目。
- 可选： 执行环境 ：用于执行的容器镜像。
- **Playbook:** 从 **playbook** 列表中选择一个与您要运行的维护计划关联的 **playbook**。
- 可选：凭证 ：输入要用于此项目的凭证，或者点击搜索(
)图标并从弹出窗口中选择它。 凭证不必是 **Red Hat Insights** 凭证。
- 详细程度 ：保持默认设置，或者从列表中选择所需的详细程度。



4. 点击 **Save**。

5. 点击启动

图标启动作业模板。

完成后，作业将生成 **Job Details** 页面。

第 31 章 自动化控制器的最佳实践

下面描述了使用自动化控制器的最佳实践：

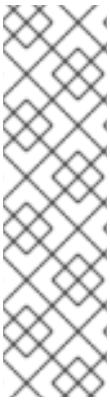
31.1. 使用源控制

自动化控制器支持直接存储在服务器上的 **playbook**。因此，您必须将 **playbook**、角色和任何关联的详情存储在源控制中。这样，您就能获得一个审计跟踪，用于描述您何时和为什么更改了自动化基础架构的规则。另外，它允许与基础架构或团队的其他部分共享 **playbook**。

31.2. ANSIBLE 文件和目录结构

如果您要创建跨项目使用的通用角色集合，则应该通过源控制子模块或一个通用位置（如 **/opt** ）访问它们。项目不应预期从其他项目导入角色或内容。

有关更多信息，请参阅 **Ansible** 文档中的链接 常规提示。



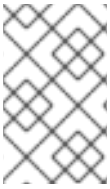
注意

- 避免使用 **playbooks vars_prompt** 功能，因为自动化控制器不以交互方式允许 **vars_prompt** 问题。如果您无法使用 **vars_prompt**，请参阅 调查 功能。
- 避免使用 **playbook** 暂停 功能时没有超时，因为自动化控制器不允许以交互方式取消暂停。如果无法使用 **pause**，则必须设置超时。

作业使用 **playbook** 目录作为当前工作目录，尽管必须编写作业来使用 **playbook_dir** 变量，而不必依赖于此操作。

31.3. 使用动态清单源（DYNAMIC INVENTORY SOURCES）

如果您的基础架构有外部的数据源，无论是云供应商还是本地 **CMDB**，最好定义一个清单同步过程，并使用对动态清单（包括云清单源）的支持。这样可确保您的清单始终为最新版本。



注意

只要 未设置 **- overwrite_vars**，在清单同步后编辑和添加清单主机变量仍然有效。

31.4. 清单的变量管理

使用主机和组定义保留变量数据（请参阅清单编辑器），而不是使用 **group_vars/** 和 **host_vars/**。如果使用动态清单源，只要未设置 **Overwrite Variables** 选项，自动化控制器就可以将这些变量与数据库同步。

31.5. 自动缩放

使用 **"callback"** 功能允许新引导实例请求配置自动扩展或置备集成。

31.6. 大量主机

将作业模板上的 **"forks"** 设置为较大的值，以增加执行运行的并行性。有关调整 **Ansible** 的更多信息，请参阅 **Ansible** 博客。

31.7. 持续集成/持续部署

对于持续集成系统（如 **Jenkins**）来生成作业，它必须向作业模板发出 **curl** 请求。作业模板的凭证不需要提示输入任何特定密码。有关配置和使用说明，请参阅 **Ansible** 文档中的安装。

第 32 章 安全性

以下小节描述了自动化控制器如何处理并可让您控制文件系统安全性。

所有 **playbook** 都是通过 **awx** 文件系统用户执行的。对于运行作业，自动化控制器通过使用 **Linux** 容器提供作业隔离。这种保护可确保作业只能从该作业模板的项目目录访问 **playbook**、角色和数据。

为了获得凭证安全性，您可以选择上传锁定的 **SSH** 密钥，并将解锁密码设置为**"ask"**。您还可以选择让系统提示输入 **SSH** 凭证或 **sudo** 密码，而不是让系统将其存储在数据库中。

32.1. PLAYBOOK 访问和信息共享

自动化控制器使用自动化执行环境和 **Linux** 容器可防止 **playbook** 读取其项目目录之外的文件。

默认情况下，公开给容器内 **ansible-playbook** 进程的唯一数据是当前使用的项目。

您可以在 **Job Settings** 中自定义此功能，并将主机中的其他目录公开给容器中。

32.1.1. 隔离功能和变量

自动化控制器使用容器技术将作业相互隔离。默认情况下，只有当前项目公开给运行作业模板的容器。

如果需要公开其他目录，您必须自定义 **playbook** 运行。要配置作业隔离，您可以设置变量。

默认情况下，自动化控制器使用系统的 **tmp** 目录（默认为**/tmp** ）作为其暂存区域。这可以在 **Jobs** 设置 页面的 **Job Execution Path** 字段中更改，也可以在位于 **/api/v2/settings/jobs** 的 **REST API** 中进行更改：

```
AWX_ISOLATION_BASE_PATH = "/opt/tmp"
```

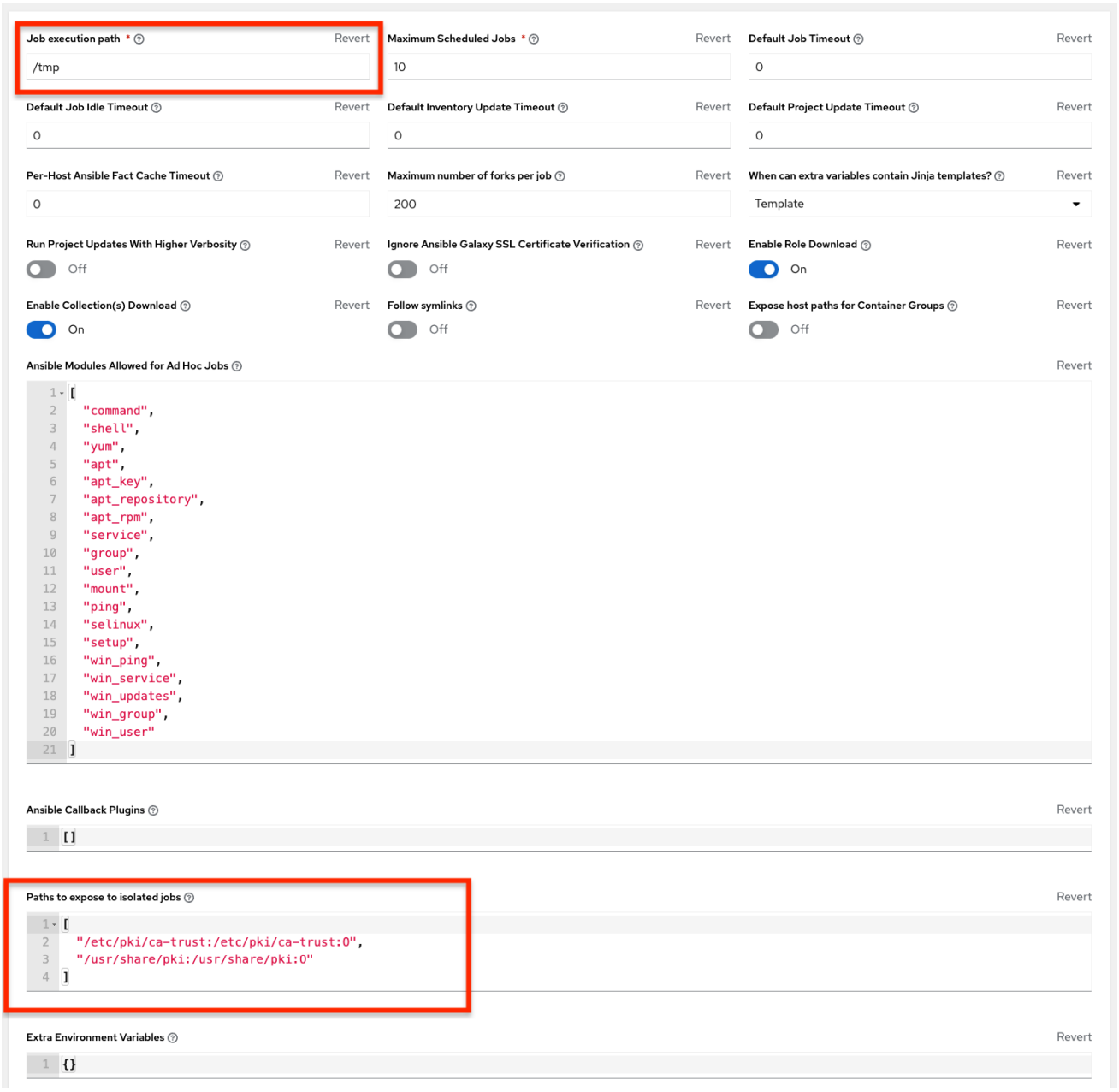
如果应该从主机向运行 **playbook** 的容器公开任何其他目录，您可以在 **Jobs** 设置页面的 **Paths to Expose to Isolated Jobs** 字段中指定，或者在位于 **/api/v2/settings /jobs** 的 **REST API** 中指定它们：

```
AWX_ISOLATION_SHOW_PATHS = ['/list/of/', '/paths']
```

注意

如果您的 **playbook** 需要使用 **AWX_ISOLATION_SHOW_PATHS** 中定义的密钥或设置，请将此文件添加到 **/var/lib/awx/.ssh** 中。

此处描述的字段可在 **Jobs** 设置 页面中找到：



32.2. 基于角色的访问控制

基于角色的访问控制 **(RBAC)**内置在自动化控制器中，并允许管理员委托对服务器清单、机构等的访问权限。管理员也可以集中管理各种凭据，允许用户在不向用户公开该机密的情况下使用所需的机密。您可以使用 **RBAC** 启用自动化控制器来提高安全性和简化管理。

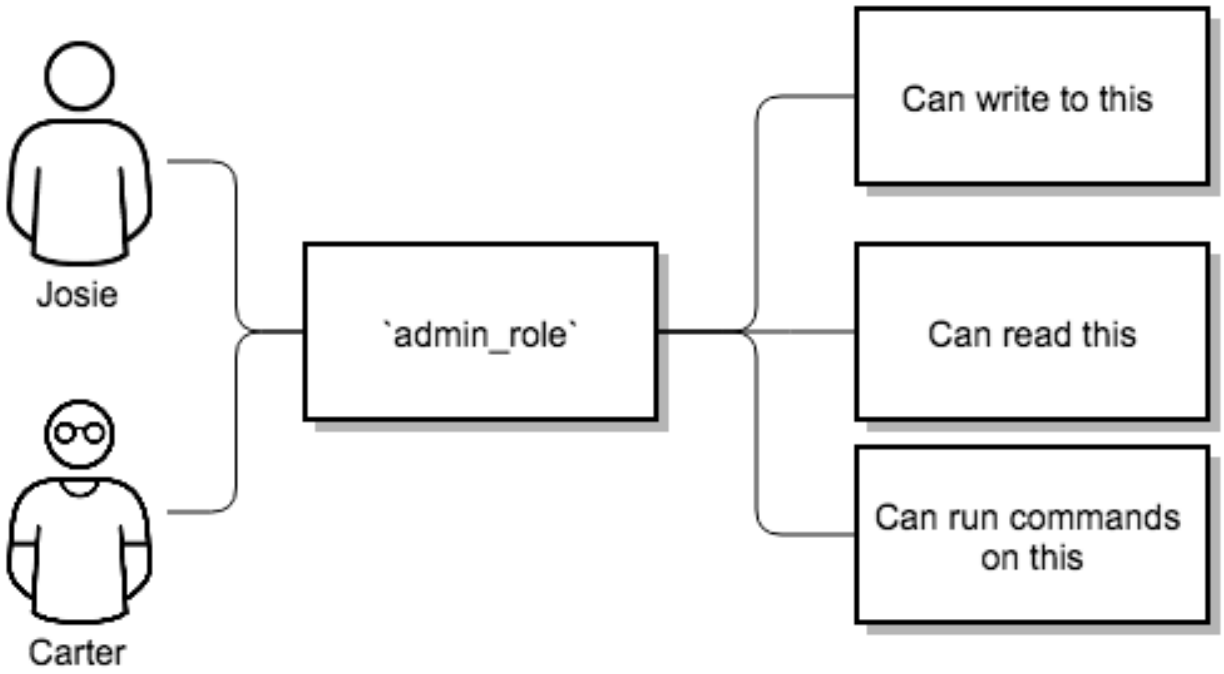
RBAC 是向用户或团队授予角色的方法。**RBAC** 可以被认为是角色，它精确定义了谁或什么可以看到、更改或删除要为其设置特定功能的"对象"。

自动化控制器的 **RBAC** 设计角色、资源和用户的主要概念如下：

- 用户可以是一个角色的成员，授予他们对与该角色关联的任何资源或与"子代"角色关联的任何资源的访问权限。
- 角色是能力的集合。
- 用户通过为其分配的角色或通过从角色层次结构继承的角色获得对这些权限和自动化控制器资源的访问权限。
- 角色将一组能力与一组用户相关联。所有功能都源自角色内的成员资格。用户仅通过为其分配的角色或通过角色层次结构继承的角色获得权限。角色的所有成员都具有授予该角色的所有权限。在一个机构中，角色相对稳定，而用户和能力有很多且可能会快速变化。
- 用户可以有许多角色。

32.2.1. 角色层次结构和访问权限继承

假设您有一个名为**"SomeCompany"**的机构，并想给两个人**"Josie"**和**"Carter"**，以管理与该机构关联的所有设置。为此，您必须使两个人都成为 组织的 **admin_role** 的成员。

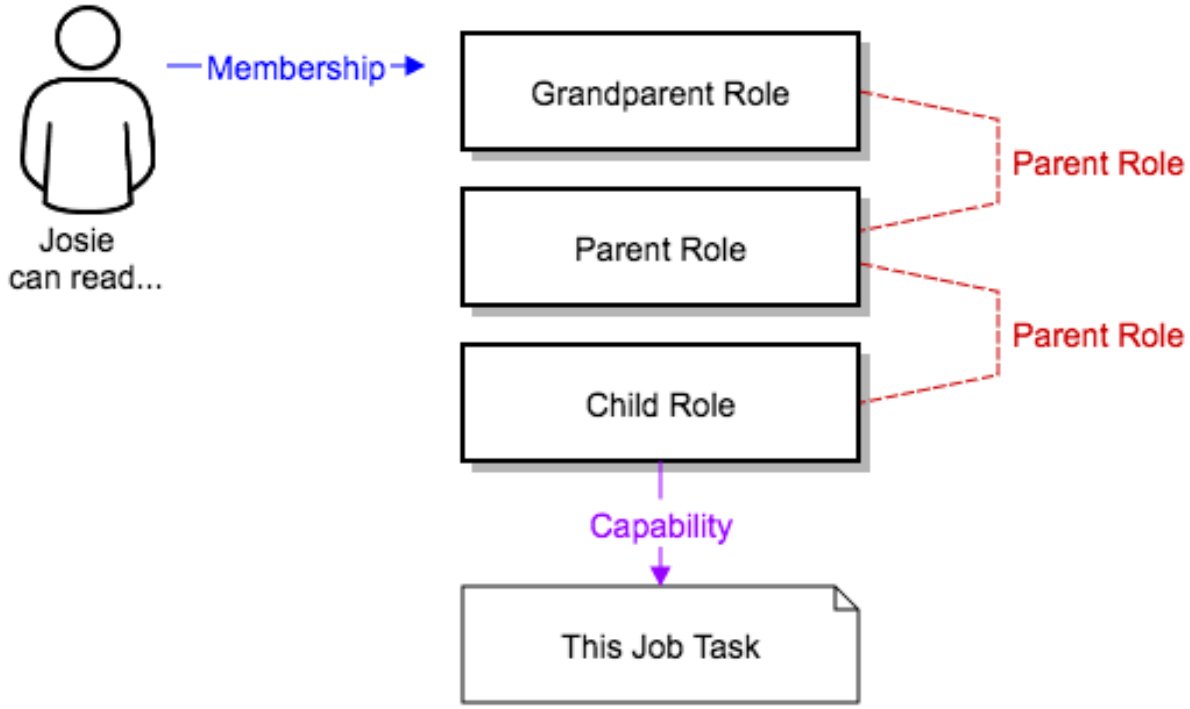


系统中通常会有许多角色，其中一些您要包含其他角色的所有功能。例如，您可能希望系统管理员可以访问机构管理员可访问的所有内容，而机构管理员具有项目管理员可访问的所有内容。

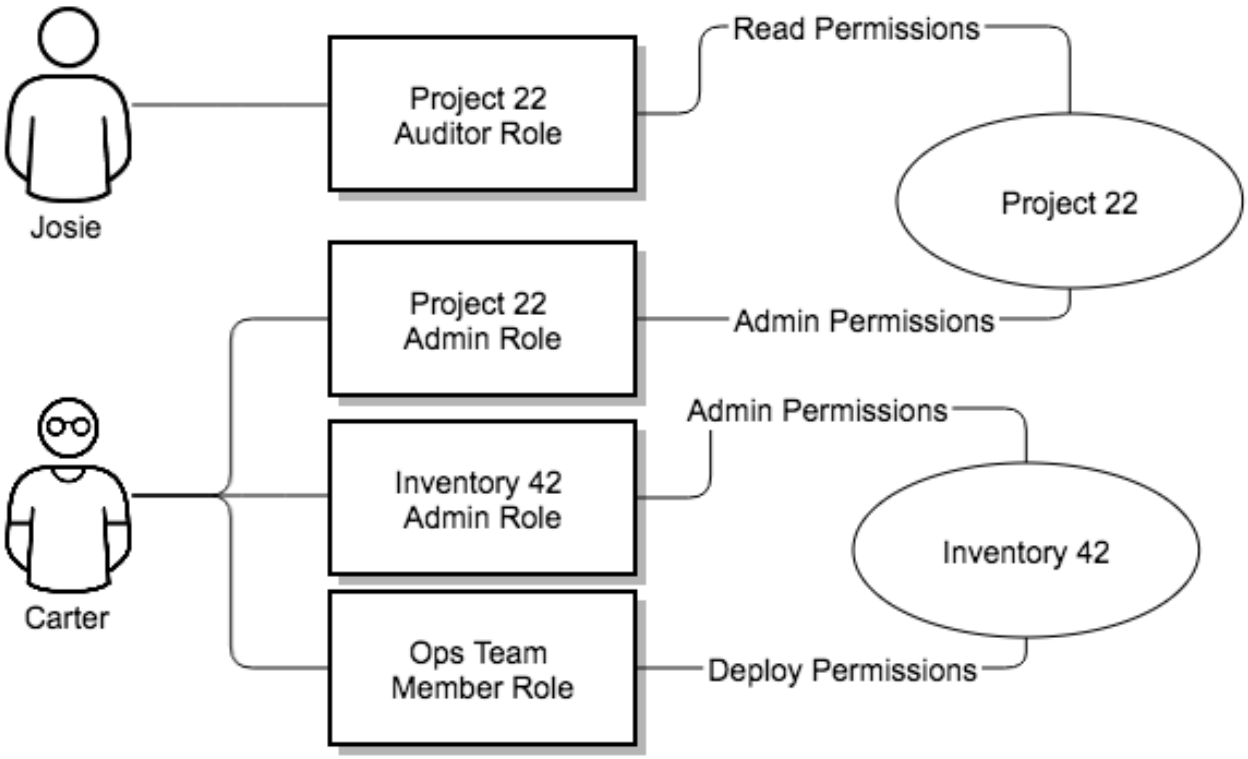
这个概念被称为 **"Role Hierarchy"**：

- 父角色获取与任何子角色相关的所有功能。
- 角色的成员可以自动获取他们所属角色以及任何子角色的所有权限。

角色层次结构通过允许角色具有"父角色"来表示。角色具有的任何权限都会被隐式授予任何父角色（或那些父角色的父级）。



角色可以有多个父角色，并且为所有父项隐式授予了能力。



RBAC 还允许您明确允许用户和用户团队针对特定主机组运行 **playbook**。用户和团队仅限于被授予了能力的 **playbook** 和主机组。使用自动化控制器，您可以根据需要创建多个用户和团队，手动创建用户和团队，或者从 **LDAP** 或 **Active Directory** 导入它们。

32.2.1.1. 使用 RBAC

下面介绍如何在您的环境中应用自动化控制器的 **RBAC** 系统。

32.2.1.1.1. 编辑用户

在编辑用户时，自动化控制器系统管理员可以将用户指定为 *系统管理员* （也称超级用户）或 ***System Auditor*** ：

- 系统管理员会隐式地继承环境中所有对象的所有权限（读取/写入/执行）。
- 系统审核员隐式继承环境中所有对象的只读权限。

32.2.1.1.2. 编辑机构

在编辑机构时，系统管理员可以指定以下角色：

- 一个或多个用户作为机构管理员
- 一个或多个用户作为机构审核员
- 一个或多个用户（或团队）作为机构成员

作为机构成员的用户和团队可以查看其机构管理员。

作为机构管理员的用户隐式继承了该机构内所有对象的所有权限。

作为机构审核员的用户隐式继承了该机构内所有对象的只读权限。

32.2.1.1.3. 编辑机构中的项目

在编辑机构中的项目时，系统管理员和机构管理员可指定：

- 一个或多个作为项目管理员的用户或团队
- 一个或多个作为项目成员的用户或团队
- 一个或多个可从 **SCM** 更新项目的用户或团队（来自属于该机构成员的用户和团队）。

作为项目成员的用户可以查看其项目管理员。

项目管理员隐式继承了从 **SCM** 更新项目的权限。

管理员也可以指定一个或多个可在作业模板中使用该项目的用户或团队（来自属于该项目成员的用户或团队）。

32.2.1.1.4. 在机构中创建清单和凭证

授予使用、读取或写入凭证的所有访问权限都通过角色处理，该角色使用自动化控制器的 **RBAC** 系统授予所有权、审核员或使用角色。

系统管理员和机构管理员可根据其管理功能在机构内创建清单和凭证。

无论是编辑清单还是凭证，系统管理员和机构管理员都可以指定一个或多个用户或团队（来自属于该机构成员的用户或团队）来授予该清单或凭证的用量权限。

系统管理员和机构管理员可以指定一个或多个用户或团队（来自属于该机构成员的用户或团队），以便具有（动态或手动）清单更新（动态或手动）。管理员也可以为清单执行临时命令。

32.2.1.1.5. 编辑作业模板

系统管理员、机构管理员和项目管理员在其管理功能下的项目中可以创建和修改该项目的新作业模板。

在编辑作业模板时，管理员（自动化控制器、机构和项目）可以在他们具有使用权限的机构中选择清单和凭证，或者他们可以将这些字段留空以便在运行时选择。

另外，他们可以指定具有该作业模板的执行权限的一个或多个用户或团队（来自属于该项目成员的用户或团队）。无论用户或团队针对清单或作业模板中指定的凭证授予了任何显式功能，执行能力都是有效的。

32.2.1.1.6. 用户视图

用户可以：

- 查看他们所属的任何机构或项目
- 创建只属于他们自己的凭证对象
- 查看并执行他们被授予执行权限的任何作业模板

如果赋予了执行功能的作业模板没有指定清单或凭证，则在运行时会提示用户在运行时选择他们拥有的机构中的清单和凭证，或被授予了使用能力。

作为作业模板管理员的用户可以对作业模板进行更改。但是，若要更改作业模板中使用的清单、项目、**playbook**、凭证或实例组，用户还必须具有当前使用或正在设置的项目和清单的"使用"角色。

32.2.1.2. 角色

授予使用、读取或写入凭证的所有访问权限都通过角色处理，并且为资源定义角色。

32.2.1.2.1. 内置角色

下表列出了 **RBAC** 系统角色，它包括了如何根据自动化控制器中的权限定义角色的定义描述：

系统角色	它可以执行什么操作
系统管理员 (System Administrator) - 系统范围单例	管理系统的所有方面
系统审核员 (System Auditor) - 系统范围单例	查看系统的所有方面
临时角色 (Ad Hoc Role) - 清单	对清单运行临时命令
管理员角色 (Admin Role) - 机构、团队、清单、项目、作业模板	管理定义的机构、团队、清单、项目或作业模板的所有方面
审核员角色 (Auditor Role) - 所有	查看定义的机构、项目、清单或作业模板的所有方面
执行角色 (Execute Role) - 作业模板	运行分配的作业模板
成员角色 (Member Role) - 机构、团队	用户 是定义的机构或团队的成员。
读取角色 (Read Role) - 机构、团队、清单、项目、作业模板	查看定义的机构、团队、清单、项目或作业模板的所有方面
更新角色 (Update Role) - 项目	从配置的源控制管理系统更新项目
更新角色 (Update Role) - 清单	使用云源更新系统更新清单
所有者角色 (Owner Role) - 凭证	拥有并管理此凭证的所有方面
Use Role - Credential, Inventory, Project, IGs, CGs	在作业模板中使用凭证、清单、项目、IG 或 CG

单例角色是授予系统范围权限的特殊角色。自动化控制器目前提供两个内置单例角色，但目前不支持创建或自定义单例角色。

32.2.1.3. 常见团队角色 -“Personas”

自动化控制器支持人员通常能够确保自动化控制器可用，并以平衡可支持性和用户易于使用的方式进行管理。自动化控制器支持人员通常会向用户分配 *机构所有者* 或 *管理员角色*，以便他们能够创建新机构或添加其团队中所需的成员。这可最小化支持人员的数量，并专注于保持服务的正常运行时间，并协助用户使用自动化控制器的用户。

下表列出了自动化控制器机构管理的一些常见角色：

系统角色（用于机构）	常见用户角色	描述
所有者	团队领导 - 技术领导	此用户可控制其机构中其他用户的访问权限。他们可以添加、删除和授予用户对项目、清单和作业模板的特定访问权限。这种类型的用户也可以创建、删除或修改机构项目、模板、清单、团队和凭证的任何方面。
审核员 (Auditor)	安全工程师 - 项目管理器	这个帐户可以在只读模式下查看机构的所有方面。对于检查和维护合规性的用户，这可能是一个不错的角色。对于管理或将自动化控制器的作业数据发送到其他数据收集器的服务帐户，这可能是一个很好的角色。
成员 - 团队	所有其他用户	默认情况下，作为机构成员的这些用户不会收到对机构任何方面的任何访问权限。要授予他们访问对应的机构所有者，必须将它们添加到其各自团队中，并为机构的项目、清单和作业模板的每个组件授予管理员、执行、使用、更新和临时权限。
成员 - 团队"所有者"	超级用户 - 领导开发人员	机构所有者可以通过组接口（包括项目、清单和作业模板）的团队接口提供"admin"。这些用户能够修改和使用所给访问权限的相应组件。
成员 - 团队"执行"	开发人员 - 工程师	这是最常见的角色，使机构成员能够执行作业模板和对特定组件的读取权限。此权限适用于模板。
成员 - 团队"使用"	开发人员 - 工程师	此权限适用于机构的凭证、清单和项目。此权限可让用户使用其作业模板中的相应组件。
成员 - 团队"更新"	开发人员 - 工程师	此权限适用于项目。允许用户在项目上运行 SCM 更新。

32.3. 角色的功能：编辑和创建

机构"资源角色"功能特定于特定资源类型，如工作流。作为此类角色的成员通常提供两种类型的权限：如果用户被授予机构 **"Default"** 的 **"workflow admin role"**，则具有以下权限：

- 此用户可以在机构 **"Default"** 中创建新工作流
- 此用户可编辑 **"Default"** 机构中的所有工作流

一个例外是作业模板，其中拥有角色独立于创建权限。如需更多信息，请参阅 作业模板。

32.3.1. 资源角色和机构成员资格角色的独立性

特定于资源的机构角色独立于管理员和成员的组织角色。拥有**"Default"**机构的**"workflow** 管理员角

色"不允许用户查看机构中所有用户，但"Default"机构中具有 "member" 角色。两种角色互相独立委托。

32.3.1.1. 编辑作业模板所需的权限

用户可以仅使用作业模板管理员角色单独编辑不会影响作业运行的字段（非敏感字段）。但是，要编辑影响作业模板中运行的字段，用户必须具有以下内容：

- 作业模板和容器组的 **admin** 角色
- 相关项目的 使用 角色
- 相关清单的 使用 角色
- 相关实例组的 使用 角色

引入了"organization job template admin"角色，但如果用户没有项目、清单或 *实例组使用* 角色，则此角色本身不足以编辑该机构内的作业模板。

要将 *完整的* 作业模板控制（位于机构中）委派给用户或团队，您必须授予团队或用户所有三个机构级角色：

- 作业模板管理员
- 项目管理员
- 清单管理员

这样可确保用户（或属于具有这些角色的团队成员的所有用户）具有修改机构中作业模板的完整访问权限。如果作业模板使用另一个机构的清单或项目，则具有这些机构角色的用户仍然可以具有修改该作业模板的权限。为了清晰起见，请不要混合来自不同机构的项目或清单。

32.3.1.2. RBAC 权限

每个角色都必须有一个内容对象，例如，机构管理员角色具有机构的内容对象。要委派角色，您必须具有内容对象的管理员权限，但有些例外情况会导致您可以重置用户的密码。

parent 是组织。

allow 是这个新权限将明确允许的内容。

范围 是创建此新角色的父资源。例如： **Organization.project_create_role**。

假设资源的创建者被授予该资源的管理员角色。资源创建并不意味着明确指定资源管理的实例。

与每种管理员类型关联的规则如下：

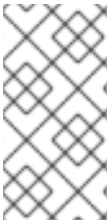
项目管理员

- **Allow**：创建、读取、更新、删除任何项目
- **Scope**：机构
- **User Interface**： *项目添加屏幕 - 机构*

清单管理员

- **Parent**：机构管理员
- **Allow**：创建、读取、更新、删除任何清单
- **Scope**：机构

- **User Interface**： *清单添加屏幕 - 机构*



注意

与 **Use** 角色一样，如果您为用户分配了 **Project Administrator** 和 **Inventory Administrator** 角色，它允许他们为您的机构创建作业模板（而非工作流）。

凭证管理员

- **Parent**：机构管理员
- **Allow**：创建、读取、更新、删除共享凭证
- **Scope**：机构
- **User Interface**： *凭证添加屏幕 - 机构*

通知管理员

- **Parent**：机构管理员
- **Allow**：通知的分配
- **Scope**：机构

工作流管理员

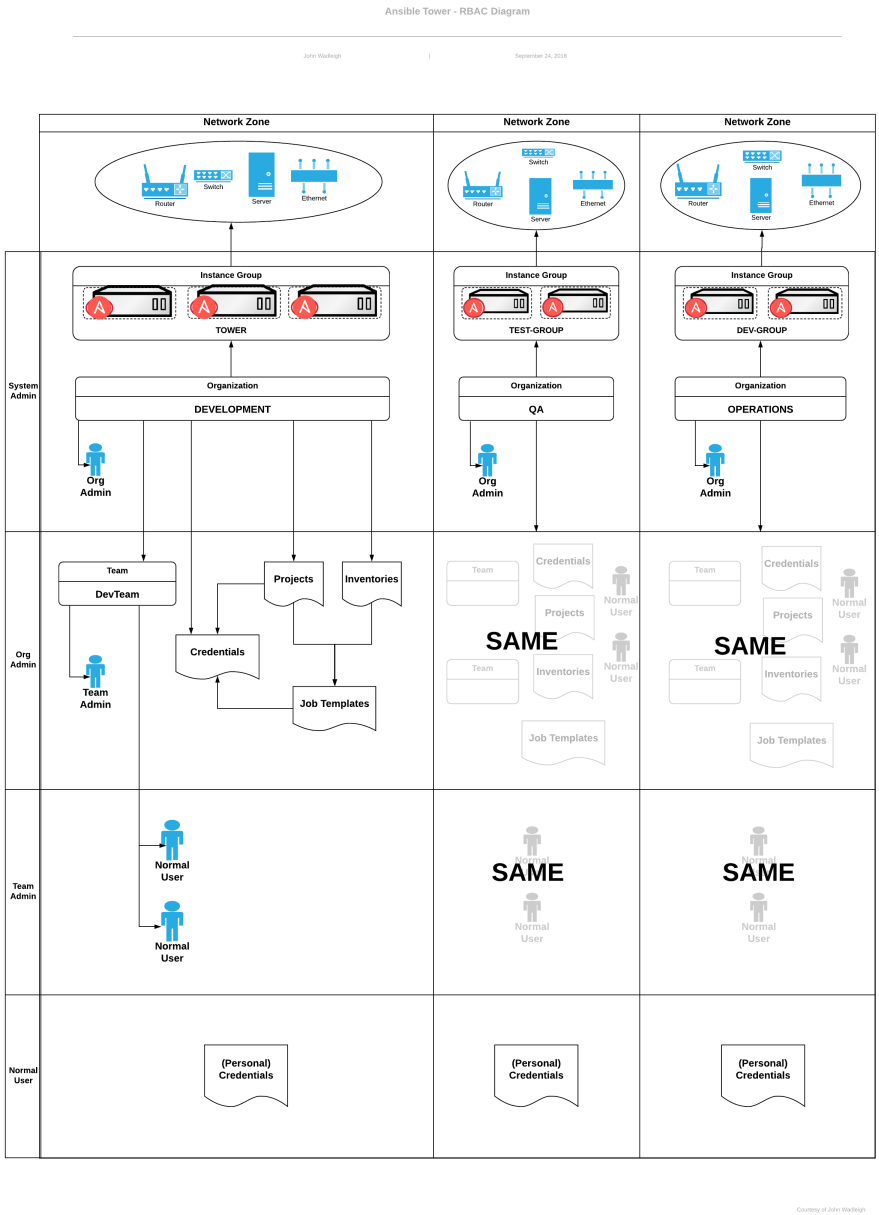
- **Parent**：机构管理员

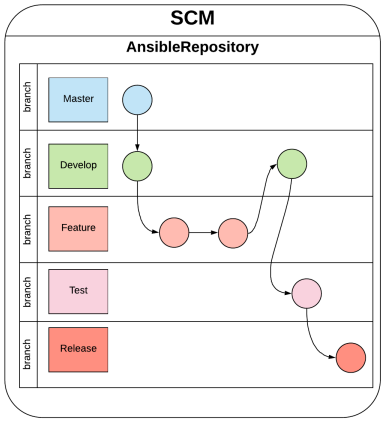
- **Allow**：创建工作流
- **Scope**：机构

机构执行

- **Parent**：机构管理员
- **Allow**：执行作业模板和工作流作业模板
- **Scope**：机构

以下是一个示例场景，显示了一个机构及其角色，以及每个角色可以访问哪些资源：


Ansible Tower - RBAC Diagram
John Wadleigh | September 24, 2016
Network Zone
Network Zone
Network Zone
Router
Switch
Server
Ethernet
Instance Group
TOWER
TEST-GROUP
DEV-GROUP
System Admin
Organization
DEVELOPMENT
QA
OPERATIONS
Org Admin
Org Admin
Org Admin
Org Admin
Team
DevTeam
Projects
Inventories
Credentials
Job Templates
Team Admin
SAME
SAME
Team Admin
Normal User
Normal User
Normal User
Normal User
Normal User
Normal User
(Personal) Credentials
(Personal) Credentials
(Personal) Credentials
Courtesy of John Wadleigh


SCM
AnsibleRepository
branch
Master
Develop
Feature
Test
Release

第 33 章 术语表

临时（**Ad Hoc**）

临时命令 使用 **Ansible** 执行快速命令，使用 **/usr/bin/ansible**，而不是编配语言，即 **/usr/bin/ansible-playbook**。一个临时命令的示例可能会在您的基础架构中重新引导 **50** 个机器。您可以编写 **Playbook** 来完成任何操作。**playbook** 也可以将许多其他操作组合在一起。

回调插件（**Callback Plugin**）

是指用户编写的代码，这些代码可以从 **Ansible** 截获结果并对其执行操作。**GitHub** 项目中的一些示例执行自定义日志记录、发送电子邮件或播放声音效果。

控制组群

也称为 **'*cgroups*'**，控制组是 **Linux** 内核中的一个功能，使资源可以被分组并分配以运行进程。除了将资源分配给进程外，**cgroups** 还可以报告 **cgroup** 内运行的所有进程使用资源。

检查模式

使用**- check** 选项运行 **Ansible**，该选项不会在远程系统上进行任何更改，而是仅输出命令在没有此标志的情况下运行时可能发生的更改。这与其它系统中所谓的**"dry run"**模式类似。但是，这不会考虑意外的命令故障或级联效果（在其它系统中有类似模式）。使用 **Check** 模式了解可能发生的情况，但它并不是一个好的暂存环境的替代品。

容器组

容器组是实例组的一种类型，用于指定在运行任务的 **Kubernetes** 或 **OpenShift** 集群中置备 **pod** 的配置。这些 **pod** 按需置备，且仅在 **playbook** 运行期间存在。

凭证

身份验证详情，供自动化控制器用于针对机器启动作业、与清单源同步以及从版本控制系统中导入项目内容。如需更多信息，请参阅 凭证。

凭证插件

Python 代码，包含外部凭证类型、其元数据字段以及与 **secret** 管理系统交互所需代码的定义。

分布式作业

由作业模板、清单和分片大小组成的作业。执行后，分布式作业会将每个清单划分为多个**"**分片大小**"**块，然后用于运行较小的作业分片。

外部凭证类型

用于与 **secret** 管理系统进行身份验证的受管凭证类型。

事实

事实是发现与远程节点相关的内容。虽然它们可以像变量一样在 **playbook** 和模板中使用，但事实是推断出来的，而不是设置。通过在远程节点上执行内部设置模块，在运行 **play** 时自动发现事实。您不必显式调用 **setup** 模块：只需要运行。如果不需要，可以禁用它来节省时间。为方便从其他配置管理系统切换的用户，事实模块也会从 **ohai** 和 **facter** 工具中提取事实（如果分别安装了事实库）和 **Puppet**。

Forks

Ansible 和自动化控制器并行与远程节点通信。可以在创建或编辑作业模板的过程中通过传递**- forks** 或编辑配置文件中的默认设置，以多种方式设置并行级别。默认值为非常保守的五个 **fork**，但如果您有大量 **RAM**，您可以将它设置为更高的值，如 **50**，以增加并行性。

Group

Ansible 中的一组主机，可以作为集合进行寻址，其中很多主机可以存在于单个清单中。

组变量

group_vars/ 文件是存储在具有清单文件的目录中的文件，具有以每个组命名的可选文件名。这是放置为给定组提供的变量（特别是复杂的数据结构）的便捷位置，因此这些变量不必嵌入到清单文件或 **playbook** 中。

处理程序（**handler**）

处理程序类似于 **Ansible playbook** 中的常规任务（请参阅 任务），但只有任务包含**"notify"**指令并且指示它更改了内容时才运行。例如，如果更改了配置文件，则引用配置文件模板操作的任务可能会通知服务重启处理程序。这意味着，只有在需要重启服务时，才能退回服务。处理程序可用于服务重启以外的内容，但服务重启是最常见的用途。

主机

由自动化控制器管理的系统，可能包含物理、虚拟或基于云的服务器，或其他设备（通常是操作系统实例）。主机包含在清单中。有时被称为**"**节点**"**。

主机指定符

Ansible 中的每个 **Play** 将一系列任务（定义系统的角色、目的或顺序）映射到一组系统。每个 **play** 中的这个**"hosts:"**指令通常称为主机指定符。它可以选择一个系统、多个系统、一个或多个组，或者一个组中的主机，并且明确不在另一个组中。

实例组

包含用于在集群环境中使用的实例的组。实例组提供根据策略对实例进行分组的功能。

清单（**Inventory**）

可对其启动任务的主机集合。

清单脚本

查找主机、组成员资格以及来自外部资源的变量信息的程序，无论是 **SQL** 数据库、**CMDB** 解决方案还是 **LDAP**。这个概念由 **Puppet** （其中称为"外部节点分类器"）调整，并以类似的方式工作。

清单源

有关要合并到当前清单组中的云或其他脚本的信息，从而自动填充组、主机以及有关这些组和主机的变量。

作业

自动化控制器启动的许多后台任务之一，这通常是实例化作业模板来启动 **Ansible playbook**。其他类型的作业包括清单导入、从源控制进行项目同步或管理清理操作。

作业详情

运行特定作业的历史记录，包括其输出和成功/失败状态。

作业分片

请参阅 分布式作业。

任务模板

Ansible playbook 以及启动它所需的一组参数的组合。如需更多信息，请参阅 作业模板。

JSON

JSON 是基于文本的格式，用于根据 **JavaScript** 对象语法表示结构化数据。**Ansible** 和自动化控制器使用 **JSON** 从远程模块返回数据。这可让使用任何语言编写的模块，而不只是 **Python** 编写的模块。

Mesh（网格）

描述由节点组成的网络。节点之间的通信通过 **TCP**、**UDP** 或 **Unix** 套接字等协议在传输层建立。

另请参阅 节点。

元数据

身份验证后，用于在外部系统中查找 **secret** 的信息。用户在将外部凭证链接到目标凭证字段时提供此信息。

节点

节点对应于实例数据库模型或 **/api/v2/instances/** 端点中的条目，是参与集群或网格的机器。统一作业 **API** 报告 **controller_node** 和 **execution_node** 字段。执行节点是作业运行的位置，以及作业和服务器功能之间的控制器节点接口。

节点类型	描述
控制	运行持久服务的节点，并将作业委托给混合节点和执行节点。
混合	运行持久服务和执行作业的节点。
hop	仅用于跨网格转发。
æ‰§è¡Œ	用于运行从控制节点交付的作业的节点（从用户的 Ansible 自动化提交的作业）

通知模板

通知类型的实例（电子邮件、**Slack** 、**Webhook** 等），其名称、描述和定义的配置。

通知

通知（如 **Email**、**Slack** 或 **Webhook**）在通知模板中定义名称、描述和配置。例如，当作业失败时，将使用通知模板定义的配置发送通知。

通知

任务注册更改事件并通知处理器任务在 **play** 结束时需要运行另一个操作。如果处理程序由多个任务通知，它仍然仅运行一次。处理程序按照列出的顺序运行，而不是按照通知的顺序运行。

机构（**Organization**）

用户、团队、项目和清单的逻辑集合。机构是对象层次结构中的最高级别。

Organization Administrator（机构管理员）

具有修改机构成员资格和设置权限的用户，包括在该机构中创建新用户和项目。机构管理员也可以向该机构内的其他用户授予权限。

权限

分配给用户和团队的权限集，它提供读取、修改和管理项目、清单和其他对象的能力。

Play

一个 **play** 在由主机指定符选择的一组主机（通常由组选择，但有时由主机名 **glob** 选择）与那些主机上运行的任务之间的映射最小，以定义这些系统执行的角色。**playbook** 是 **play** 的列表。**playbook** 中可以有一个或多个 **play**。

Playbook

一个 **Ansible playbook**。有关更多信息，请参阅 **Ansible playbook**。

policy

策略指定实例组的行为方式以及任务的执行方式。

项目

自动化控制器中代表 **Ansible playbook** 的逻辑集合。

角色

角色是 **Ansible** 和自动化控制器中的机构单元。将角色分配给一组主机（或一组组或主机模式等）意味着它们实施特定的行为。角色可以包含应用变量值、任务和处理程序，或者这些因素的组合。由于与角色关联的文件结构，角色成为可重新分发的单元，允许您在 **playbook** 或其他用户之间共享行为。

Secret 管理系统

用于安全存储并控制对令牌、密码、证书、加密密钥和其他敏感数据的访问的服务器或服务。

调度

作业应自动运行的日期和时间日历。

分片作业

请参阅 分布式作业。

源凭证

链接到目标凭证字段的外部凭证。

Sudo

Ansible 不需要 **root** 登录，因为它是无守护进程的，因此不需要 **root** 级别守护进程（可能是敏感环境

中的安全问题）。**Ansible** 可以登录并执行嵌套在 **sudo** 命令中的许多操作，并可处理无密码和基于密码的 **sudo**。在以 **sudo** 模式运行时，可以使用 **Ansible** *复制*、*模板* 和 *获取* 模块来实现一些通常无法使用 **sudo** 的操作（如 **scp** 文件传输）。

超级用户

具有编辑系统中任何对象权限的服务器的管理员，无论它是否与任何机构相关联。超级用户可以创建机构和其他超级用户。

问卷调查

在作业启动时由作业模板询问的问题，可在作业模板上配置。

目标凭证

一个非外部的凭证，它带有一个输入字段，用来链接到外部凭证。

Team

具有关联用户、项目、凭证和权限的机构子部门。团队提供了一种方式来实现基于角色的访问控制方案，并跨机构委派职责。

用户

具有相关权限和凭证的操作器。

Webhook

Webhook 支持应用程序之间的通信和信息共享。它们用于响应推送到 **SCM** 的提交以及启动作业模板或工作流模板。

工作流作业模板

由任何作业模板组合、项目同步和清单同步组成的集合，将它们链接在一起，以便以单个单元的形式执行。

YAML

人类可读的语言，通常用于编写配置文件。**Ansible** 和自动化控制器使用 **YAML** 定义 **playbook** 配置语言和变量文件。**YAML** 具有最小的语法，非常干净，用户可以轻松地。它是配置文件和人类的数据格式，但也是机器可读。**YAML** 在动态语言社区中很常见，格式具有可用于多种语言序列化的库。示例包括 **Python**、**Perl** 或 **Ruby**。