Red Hat Ansible Automation Platform 2.4

Red Hat Ansible Automation Platform 操作指南

安装后配置，以确保平稳部署 Ansible Automation Platform 安装

Last Updated: 2024-06-17

Red Hat Ansible Automation Platform 2.4 Red Hat Ansible Automation Platform 操作指南

安装后配置，以确保平稳部署 Ansible Automation Platform 安装

法律通告

摘要

本指南提供有关 Red Hat Ansible Automation Platform 安装后活动的说明和指导。

目录

前言

安装 Red Hat Ansible Automation Platform 后，您的系统可能需要额外的配置，以确保您的部署平稳运行。本指南为配置任务提供在安装 Red Hat Ansible Automation Platform 后可以执行的操作。

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 激活 RED HAT ANSIBLE AUTOMATION PLATFORM

Red Hat Ansible Automation Platform 使用可用的订阅或订阅清单来授权使用 Ansible Automation Platform。要获取订阅，您可以执行以下操作之一：

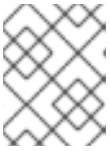
1. 启动 Ansible Automation Platform 时，请使用您的红帽客户或 Satellite 凭证。
2. 使用 Red Hat Ansible Automation Platform 界面或手动在 Ansible playbook 中上传订阅清单文件。

1.1. 使用凭证激活

当 Ansible Automation Platform 首次启动时，Ansible Automation Platform 订阅屏幕会自动显示。您可以使用您的红帽凭证直接检索您的订阅并将其导入 Ansible Automation Platform。

流程

1. 输入您的红帽用户名和密码。
2. 点 **Get Subscriptions**。



注意

如果集群节点通过 Subscription Manager 注册到 Satellite，也可以使用您的 Satellite 用户名和密码。

3. 查看最终用户许可证协议并选择 **我同意最终用户许可证协议**。
4. 默认检查跟踪和分析选项。这些选择可以帮助红帽通过提供更好的用户体验来改进产品。您可以通过取消选择选项来选择不再使用。
5. 点 **Submit**。
6. 接受订阅后，许可证屏幕会显示并进入 Ansible Automation Platform 界面的 Dashboard。您可以点 **Settings** 图标 ⚙，从设置界面中选择 **License** 标签页来返回到许可证屏幕。

1.2. 使用清单文件激活

如果您有订阅清单，可以使用 Red Hat Ansible Automation Platform 界面或手动在 Ansible playbook 中上传清单文件。

先决条件

您必须有一个从红帽客户门户网站导出的 Red Hat Subscription Manifest 文件。如需更多信息，请参阅获取清单文件。

使用接口上传

1. 完成生成和下载清单文件的步骤
2. 登录到 Red Hat Ansible Automation Platform。
3. 如果没有立即提示输入清单文件，请转至 **Settings → License**。

4. 确保 **Username** 和 **Password** 字段为空。
5. 点 **Browse** 并选择清单文件。
6. 点击 **Next**。

注意

如果在 License 页面中禁用了 **BROWSE** 按钮，请清除 **USERNAME** 和 **PASSWORD** 字段。

手动上传

如果您无法使用 Red Hat Ansible Automation Platform 界面应用或更新订阅信息，您可以使用 **ansible.controller** 集合中的 **license** 模块手动将订阅清单上传到 Ansible playbook 中。

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

第 2 章 获取清单文件

您可以在 Red Hat Subscription Management 的 Subscription Allocations 部分中获取订阅清单。获取订阅分配后，您可以下载其清单文件并上传该文件以激活 Ansible Automation Platform。

首先，使用您的管理员用户帐户登录到红帽客户门户网站，并按照本节中的步骤操作。

2.1. 创建订阅分配

通过创建新订阅分配，您可以为当前离线或 air-gapped 的系统设置侧订阅和权利。这是在下载清单并将其上传到 Ansible Automation Platform 之前所必需的。

流程

1. 在 Subscription Allocations 页面中，点 **New Subscription Allocation**。
2. 输入分配的名称，以便稍后找到它。
3. 选择 **Type: Satellite 6.8** 作为管理应用。
4. 点 **Create**。

2.2. 在订阅分配中添加订阅

创建分配后，您可以添加 Ansible Automation Platform 正确运行所需的订阅。在下载清单并将其添加到 Ansible Automation Platform 之前，此步骤是必需的。

流程

1. 在 Subscription Allocations 页面中，点您要添加订阅的**订阅分配**名称。
2. 点 **Subscriptions** 选项卡。
3. 点 **Add Subscriptions**。
4. 输入您要添加的 Ansible Automation Platform 权利数量。
5. 点 **Submit**。

验证

接受订阅后，会显示订阅详情。*Compliant* 状态表示您的订阅符合您在订阅数中自动的主机数量。否则，您的状态将显示为 *Out of Compliance*，这表示您已超过订阅中的主机数量。

显示的其他重要信息包括：

主机自动化

由作业自动进行主机计数，这将消耗许可证计数

导入的主机

考虑所有清单源的主机计数（不会影响剩余的主机）

主机剩余

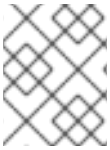
主机总数减主机自动

2.3. 下载清单文件

创建分配并拥有适当的订阅后，您可以从 Red Hat Subscription Management 下载清单。

流程

1. 在 Subscription Allocations 页面中，点您要生成清单的订阅**分配**名称。
2. 点 **Subscriptions** 选项卡。
3. 点 **Export Manifest** 以下载清单文件。



注意

该文件被保存到默认下载文件夹中，现在可以上传以激活 Red Hat Ansible Automation Platform。

第 3 章 安装后的步骤

无论您是希望开始使用自动化的新 Ansible Automation Platform 用户，还是希望将旧的 Ansible 内容迁移到最新版本的 Red Hat Ansible Automation Platform，都探索下一步以使用 Ansible Automation Platform 2.4 的新功能。

3.1. 将数据迁移到 ANSIBLE AUTOMATION PLATFORM 2.4 的步骤

对于希望完成升级到 Ansible Automation Platform 2.4 的平台管理员，可能需要额外步骤将数据迁移到新实例：

要完成升级到 Ansible Automation Platform 2.4，您必须迁移数据。将数据迁移到新实例需要额外的步骤。

3.1.1. 从旧的虚拟环境 (venvs) 迁移到自动化执行环境

Ansible Automation Platform 2.4 使您从自定义 Python 虚拟环境(venvs)改为使用自动化执行环境 - 容器化镜像包括了运行和扩展 Ansible 自动化所需的组件。这些组件包括 ansible-core、Ansible 内容集合、Python 依赖项、Red Hat Enterprise Linux UBI 8 以及任何其他软件包依赖项。

要将 venvs 迁移到执行环境，您必须使用 **awx-manage** 命令列出和导出来自原始实例的 venvs 列表，然后使用 **ansible-builder** 创建执行环境。

其他资源

- 升级到自动化执行环境指南
- 创建和恢复执行环境.

3.1.2. 使用 Ansible Builder 迁移 Ansible Engine 镜像

要迁移以前的 Ansible Engine 镜像以用于 Ansible Automation Platform 2.4，请使用 **ansible-builder** 工具自动重建镜像（包括其自定义插件和依赖项）的过程，以用于自动化执行环境。

其他资源

- 有关使用 Ansible Builder 构建执行环境的更多信息，请参阅 创建和恢复执行环境。

3.1.3. 迁移到 Ansible Core 2.13

升级到 ansible-core 2.13 时，您必须更新 playbook 和插件，或者 Ansible 基础架构的其他部分，使其被最新版本的 ansible-core 支持。

其他资源

有关为 ansible-core 2.13 兼容性更新 Ansible 内容的说明，请参阅 Ansible 内核 2.13 端口指南 。

3.2. 更新执行环境镜像位置

如果独立于 Ansible Automation Platform 安装私有自动化中心，您可以更新您的执行环境镜像位置以指向您的私有自动化中心。

流程

1. 进入包含 **setup.sh**的目录
2. 运行以下命令来创建 **./group_vars/automationcontroller** ：

```
touch ./group_vars/automationcontroller
```

3. 将以下内容粘贴到 **./group_vars/automationcontroller** 中。调整设置以适应您的环境：

```
# Automation Hub Registry
registry_username: 'your-automation-hub-user'
registry_password: 'your-automation-hub-password'
registry_url: 'automationhub.example.org'
registry_verify_ssl: False

## Execution Environments
control_plane_execution_environment: 'automationhub.example.org/ee-supported-
rhel8:latest'

global_job_execution_environments:
  - name: "Default execution environment"
    image: "automationhub.example.org/ee-supported-rhel8:latest"
  - name: "Minimal execution environment"
    image: "automationhub.example.org/ee-minimal-rhel8:latest"
```

4. 运行 **./setup.sh** 脚本

```
$ ./setup.sh
```

验证

1. 以具有系统管理员访问权限的用户身份登录 Ansible Automation Platform。
2. 进入 **管理 → 执行环境**。
3. 在 **Image** 列中，确认执行环境镜像位置已从默认值 **< registry url>/ansible-automation-platform-<version>/<image name>:<tag&gt**; 改为 **& lt;automation hub url>/<tag&gt**;。

3.3. 自动化网格的好处

Red Hat Ansible Automation Platform 的自动化网格组件简化了在多站点部署之间分布自动化的过程。对于具有多个隔离的 IT 环境的企业，自动化网格提供了一个一致且可靠的方法，使用对等对网格通信网络在执行节点上部署和扩展自动化。

当从版本 1.x 升级到最新版本的 Ansible Automation Platform 时，您必须将旧隔离节点中的数据迁移到自动化网格所需的执行节点。您可以通过规划混合和控制节点网络来实施自动化中心，然后编辑 Ansible Automation Platform 安装程序中找到的清单文件，为每个执行节点分配与网格相关的值。

其他资源

- 有关如何从隔离节点迁移到执行节点的说明，请参阅 Red Hat Ansible Automation Platform 升级和迁移指南。
- 有关自动化网格以及为您的环境设计自动化网格的各种方法的信息：
  - 有关基于虚拟机的安装，请参阅基于虚拟机的安装 的 Red Hat Ansible Automation Platform

自动化网格指南。

- 有关基于 Operator 的安装，请参阅 Red Hat Ansible Automation Platform Automation mesh for operator 的安装。

第 4 章 配置 RED HAT ANSIBLE AUTOMATION PLATFORM 的代理支持

您可以配置 Red Hat Ansible Automation Platform，以使用代理与流量通信。代理服务器充当来自其他服务器的客户机用于请求查找资源的请求的中介。客户端连接到代理服务器，请求不同的服务器提供某些服务或可用资源，代理服务器会评估请求，以简化和控制其复杂性。以下小节描述了支持的代理配置以及如何设置它们。

4.1. 启用代理支持

为了提供代理服务器支持，自动化控制器通过自动化控制器设置中的 **REMOTE_HOST_HEADERS** 列表变量处理代理请求（如 ALB、NLB、HAProxy、Squid、Nginx 和 tinyproxy）。默认情况下，**REMOTE_HOST_HEADERS** 设置为 **["REMOTE_ADDR", "REMOTE_HOST"]**。

要启用代理服务器支持，请在自动化控制器的设置页面中编辑 **REMOTE_HOST_HEADERS** 字段：

流程

1. 在自动化控制器中，进入到 **Settings**。
2. 从系统选项列表中选择 **Miscellaneous System settings**。
3. 在 **REMOTE_HOST_HEADERS** 字段中输入以下值：

```
[
  "HTTP_X_FORWARDED_FOR",
  "REMOTE_ADDR",
  "REMOTE_HOST"
]
```

自动化控制器通过搜索 **REMOTE_HOST_HEADERS** 中的标头列表来确定远程主机的 IP 地址，直到第一个 IP 地址所在的 IP 地址为止。

4.2. 已知的代理

当自动化控制器配置有 **REMOTE_HOST_HEADERS = ['HTTP_X_FORWARDED_FOR', 'REMOTE_ADDR', 'REMOTE_HOST']** 时，它假定 **X-Forwarded-For** 的值源自 Tower 前面的代理/负载均衡器。如果自动化控制器可以在不使用代理/负载均衡器的情况下访问，或者代理没有验证标头，那么 **X-Forwarded-For** 的值可以被断断为原始 IP 地址。在 **REMOTE_HOST_HEADERS** 设置中使用 **HTTP_X_FORWARDED_FOR** 可能会存在安全漏洞。

要避免这种情况，您可以在自动化控制器上的设置菜单中的 PROXY_IP_ALLOWED_LIST 字段中配置允许使用 **PROXY_IP_ALLOWED_LIST** 字段的已知代理列表。不在已知代理列表上的负载均衡器和主机将导致请求被拒绝。

4.2.1. 配置已知的代理

要为自动化控制器配置已知代理列表，请将代理 IP 地址添加到自动化控制器设置页面中的 **PROXY_IP_ALLOWED_LIST** 字段中。

流程

1. 在自动化控制器中，进入到 **Settings**，然后从 **System** 选项**列表中**选择 **Miscellaneous System 设置**。
2. 在 **PROXY_IP_ALLOWED_LIST** 字段中，输入允许连接到您的自动化控制器的 IP 地址，如下例所示：

PROXY_IP_ALLOWED_LIST 条目示例

```
[
  "example1.proxy.com:8080",
  "example2.proxy.com:8080"
]
```

重要

- **PROXY_IP_ALLOWED_LIST** 需要列表中的代理正确清理标头输入，并正确将 **X-Forwarded-For** 的值设置为客户端的实际源 IP。自动化控制器可以依赖 **PROXY_IP_ALLOWED_LIST** 中的 IP 地址和主机名为 **X-Forwarded-For** 字段提供非欺骗的值。
- 不要将 **HTTP_X_FORWARDED_FOR** 配置为 'REMOTE_HOST_HEADERS' 中的项目，除非以下**所有**条件都满足：
  - 您使用带有 ssl 终止的代理环境；
  - 代理提供 **X-Forwarded-For** 标头的清理或验证处理，以防止客户端欺骗；
  - **/etc/tower/conf.d/remote_host_headers.py** 定义 **PROXY_IP_ALLOWED_LIST**，它只包含可信代理或负载均衡器的原始 IP 地址。

4.3. 配置一个反向代理

您可以通过将 **HTTP_X_FORWARDED_FOR** 添加到自动化控制器设置中的 **REMOTE_HOST_HEADERS** 字段来支持反向代理服务器配置。**X-Forwarded-For** (XFF) HTTP 标头字段标识通过 HTTP 代理或负载平衡器连接到 Web 服务器的客户端的原始 IP 地址。

流程

1. 在自动化控制器中，进入到 **Settings**，然后从 **System** 选项**列表中**选择 **Miscellaneous System 设置**。
2. 在 **REMOTE_HOST_HEADERS** 字段中输入以下值：

```
[
  "HTTP_X_FORWARDED_FOR",
  "REMOTE_ADDR",
  "REMOTE_HOST"
]
```

3. 将下面的行添加到 **/etc/tower/conf.d/custom.py** 中，以确保应用程序使用正确的标头：

```
USE_X_FORWARDED_PORT = True
USE_X_FORWARDED_HOST = True
```

4.4. 启用粘性会话

默认情况下，Application Load Balancer 根据所选的负载平衡算法将每个请求独立路由到注册的目标。为了避免在负载均衡器后面运行多个自动化中心实例时出现身份验证错误，您必须启用粘性会话。启用粘性会话会设置自定义应用程序 Cookie，它与负载均衡器中配置的 Cookie 匹配，以启用粘性。此自定义 Cookie 可以包含应用程序所需的任何 Cookie 属性。

其他资源

- 有关启用粘性会话的更多信息，请参阅 Application Load Balancer 的 Sticky 会话。

第 5 章 配置自动化控制器 WEBSOCKET 连接

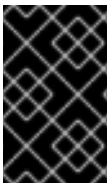
您可以配置自动化控制器，使 websocket 配置与 nginx 或负载均衡器配置保持一致。

5.1. 用于自动化控制器的 WEBSOCKET 配置

自动化控制器节点通过 websocket 互连，以在整个系统中分发所有 websocket 的消息。此配置设置可让任何浏览器客户端 websocket 订阅任何可能在自动化控制器节点上运行的任何作业。websocket 客户端不路由到特定的自动化控制器节点。任何自动化控制器节点都可以处理任何 websocket 请求，每个自动化控制器节点必须了解所有目标于所有客户端的 websocket 消息。

您可以在所有自动化控制器节点的 **/etc/tower/conf.d/websocket_config.py** 中配置 websocket，更改会在服务重启后生效。

自动化控制器将通过数据库中的实例记录自动处理其他自动化控制器节点的发现。



重要

您的自动化控制器节点旨在在多个私有、可信子网（而不是开放的互联网）中广播 websocket 流量。因此，如果您为 websocket 广播关闭 HTTPS，websocket 流量（大部分是 Ansible playbook stdout）会在自动化控制器节点之间进行发送，未加密。

5.1.1. 配置其他自动化控制器节点的自动发现

您可以配置 websocket 连接，使自动化控制器能够通过数据库中的实例记录自动处理其他自动化控制器节点的发现。

1. 编辑端口和协议的自动化控制器 websocket 信息，并确认在建立 websocket 连接时是否使用 **True** 或 **False** 验证证书：

```
BROADCAST_WEBSOCKET_PROTOCOL = 'http'
BROADCAST_WEBSOCKET_PORT = 80
BROADCAST_WEBSOCKET_VERIFY_CERT = False
```

2. 使用以下命令重启自动化控制器：

```
$ automation-controller-service restart
```

第 6 章 从自动化控制器管理可用性分析和数据收集

您可以通过在自动化控制器用户界面中不使用或更改您的设置，改变来自自动化控制器控制器的可用性分析和数据收集。

6.1. 可用性分析和数据收集

自动化控制器包括可用性数据收集，以便更好地了解自动化控制器用户如何与自动化控制器进行交互，以帮助增强未来版本，并持续简化您的用户体验。

只有安装自动化控制器试用或全新安装自动化控制器的用户才会选择使用此数据收集。

其他资源

- 如需更多信息，请参阅红帽隐私政策。

6.1.1. 控制自动化控制器的数据收集

您可以通过在 **Settings** 菜单中的**用户界面**选项卡中设置参与级别来控制自动化控制器如何收集数据。

流程

1. 登录到您的自动化控制器。
2. 导航到 **Settings**，然后从 **User Interface** 选项**中选择 User Interface** 设置。
3. 从 **User Tracking State** 下拉列表中选择所需的数据收集级别：
   - **Off**：防止数据收集。
   - **Anonymous**：启用数据收集功能，而不包括特定于您的用户数据.
   - **Detailed**：启用数据收集功能，包括特定于您的用户数据。
4. 点 **Save** 以应用设置或 **Cancel** 以丢弃更改。

第 7 章 在自动化控制器配置文件中加密明文密码

存储在自动化控制器配置文件中的密码以纯文本形式保存。有权访问 **/etc/tower/conf.d/** 目录的用户可以查看用于访问数据库的密码。对目录的访问权限是通过权限进行控制的，因此受到保护，但有些安全发现导致这种保护无法受到保护。解决方法是单独加密密码。

7.1. 创建 POSTGRESQL 密码哈希

流程

1. 在自动化控制器节点上运行以下命令：

```
# awx-manage shell_plus
```

2. 然后，从 python 提示符运行以下命令：

```
>>> from awx.main.utils import encrypt_value, get_encryption_key \
>>> postgres_secret = encrypt_value('$POSTGRES_PASS') \
>>> print(postgres_secret)
```

注意

将 **$POSTGRES_PASS** 变量替换为您要加密的实际纯文本密码。

输出应类似以下示例：

```
$encrypted$UTF8$AESCBC$Z0FBQUFBQmtLdGNRWXFjZGtkV1ZBR3hkNGVVbFFIU3hhY
21UT081eXFkR09aUWZLcG9TSmpndmZYQXFyRHVFQ3ZYSE15OUFuM1RHZHBqTFU3S
0MyNEo2Y2JWUURSYktsdmc9PQ==
```

3. 复制这些哈希的完整值并保存它们。

- 哈希值以 **$encrypted$** 开头，而不仅仅是字符字符串，如下例所示：

```
$encrypted$AESCBC$Z0FBQUFBQmNONU9BbGQ1VjJyNDJRVTRKaFRIR09lb2U5TGd
aYVRfcXFXRjlmdmpZNjdoZVpEZ21QRWViMmNDOGJaM0dPeHN2b194NUxvQ1M5X3d
Sc1gxQ29TdDBKRkljWHc9PQ==
```

请注意，**$*_PASS** 值已在清单文件中以纯文本形式。

这些步骤提供哈希值替换自动化控制器配置文件中的纯文本密码。

7.2. 加密 POSTGRES 密码

以下流程将纯文本密码替换为加密值。在集群的每个节点中执行以下步骤：

流程

1. 使用以下命令编辑 **/etc/tower/conf.d/postgres.py** ：

```
$ vim /etc/tower/conf.d/postgres.py
```

2. 将以下行添加到文件的顶部：

```
from awx.main.utils import decrypt_value, get_encryption_key
```

3. 删除 'PASSWORD' 后列出的 password 值，并将它替换为以下行，将提供的 **$encrytpted..** 替换为您自己的哈希值：

```
decrypt_value(get_encryption_key('value'),'$encrypted$AESCBC$Z0FBQUFBQmNONU9BbGQ1VjJyNDJRVTRKaFRIR09Ib2U5TGdaYVRfcXFXRjlmdmpZNjdoZVpEZ21QRWViMmNDOGJaM0dPeHN2b194NUxvQ1M5X3dSc1gxQ29TdDBKRkljWHc9PQ==='),
```

注意

此步骤中的哈希值是 **postgres_secret** 的输出值。

4. 完整的 **postgres.py** 类似如下：

```
# Ansible Automation platform controller database settings. from awx.main.utils import decrypt_value, get_encryption_key DATABASES = { 'default': { 'ATOMIC_REQUESTS': True, 'ENGINE': 'django.db.backends.postgresql', 'NAME': 'awx', 'USER': 'awx', 'PASSWORD': decrypt_value(get_encryption_key('value'),'$encrypted$AESCBC$Z0FBQUFBQmNONU9BbGQ1VjJyNDJRVTRKaFRIR09Ib2U5TGdaYVRfcXFXRjlmdmpZNjdoZVpEZ21QRWViMmNDOGJaM0dPeHN2b194NUxvQ1M5X3dSc1gxQ29TdDBKRkljWHc9PQ==='), 'HOST': '127.0.0.1', 'PORT': 5432, } }
```

7.3. 重启自动化控制器服务

流程

1. 当加密在所有节点上完成时，使用以下方法在集群中执行服务重启：

```
# automation-controller-service restart
```

2. 进入到 UI，并验证您是否能够在所有节点中运行作业。

第 8 章 续订和更改 SSL 证书

如果您的当前 SSL 证书已过期或很快过期，您可以续订或替换 Ansible Automation Platform 使用的 SSL 证书。

如果需要使用新主机等新证书重新生成 SSL 证书，则必须续订 SSL 证书。

如果要使用由内部证书颁发机构签名的 SSL 证书，则必须替换 SSL 证书。

8.1. 续订自签名 SSL 证书

以下步骤为自动化控制器和自动化中心重新生成新的 SSL 证书。

流程

1. 将 **aap_service_regen_cert=true** 添加到 **[all:vars]** 部分中的 inventory 文件中：

```
[all:vars]
aap_service_regen_cert=true
```

2. 运行安装程序。

验证

- 验证自动化控制器上的 CA 文件和 server.crt 文件：

```
openssl verify -CAfile ansible-automation-platform-managed-ca-cert.crt /etc/tower/tower.cert
openssl s_client -connect <AUTOMATION_HUB_URL>:443
```

- 验证自动化中心上的 CA 文件和 server.crt 文件：

```
openssl verify -CAfile ansible-automation-platform-managed-ca-cert.crt
/etc/pulp/certs/pulp_webserver.crt
openssl s_client -connect <AUTOMATION_CONTROLLER_URL>:443
```

8.2. 更改 SSL 证书

要更改 SSL 证书，您可以编辑清单文件并运行安装程序。安装程序验证所有 Ansible Automation Platform 组件是否正常工作。安装程序可能需要很长时间才能运行。

或者，您可以手动更改 SSL 证书。这速度更快，但没有自动验证。

红帽建议您使用安装程序更改 Ansible Automation Platform 实例。

8.2.1. 先决条件

- 如果存在中间证书颁发机构，您必须将它附加到服务器证书中。
- 自动化控制器和自动化中心都使用 NGINX，因此服务器证书必须采用 PEM 格式。
- 为证书使用正确的顺序：服务器证书首先，后跟中间证书颁发机构。

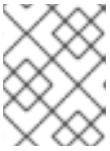
如需更多信息，请参阅 NGINX 文档中的 ssl 证书部分。

8.2.2. 使用安装程序更改 SSL 证书和密钥

以下流程描述了如何更改清单文件中的 SSL 证书和密钥。

流程

1. 将新的 SSL 证书和密钥复制到相对于 Ansible Automation Platform 安装程序的路径。
2. 将 SSL 证书和密钥的绝对路径添加到清单文件。有关设置这些变量的指导，请参阅 Red Hat AnsibleAutomation Platform 安装指南中的 Automation controller 变量、Automation hub 变量和 Event-Driven Ansible 控制器变量部分。https://docs.redhat.com/en/documentation/red_hat_ansible_automation_platform/2.4/html/hub-variables
   - 自动化控制器： **web_server_ssl_cert,web_server_ssl_key,custom_ca_cert**
   - 自动化中心：**automationhub_ssl_cert,automationhub_ssl_key,custom_ca_cert**
   - event-Driven Ansible 控制器： **automationedacontroller_ssl_cert,automationedacontroller_ssl_key,custom_ca_cert**

   注意

   custom_ca_cert 必须是签署中间证书颁发机构的根证书颁发机构。此文件安装在 **/etc/pki/ca-trust/source/anchors** 中。
3. 运行安装程序。

8.2.3. 手动更改 SSL 证书

8.2.3.1. 在自动化控制器中手动更改 SSL 证书和密钥

以下流程描述了如何在 Automation Controller 上手动更改 SSL 证书和密钥。

流程

1. 备份当前的 SSL 证书：

   ```
   cp /etc/tower/tower.cert /etc/tower/tower.cert-$(date +%F)
   ```
2. 备份当前密钥文件：

   ```
   cp /etc/tower/tower.key /etc/tower/tower.key-$(date +%F)+
   ```
3. 将新 SSL 证书复制到 **/etc/tower/tower.cert**。
4. 将新密钥复制到 **/etc/tower/tower.key**。
5. 恢复 SELinux 上下文：

   ```
   restorecon -v /etc/tower/tower.cert /etc/tower/tower.key
   ```
6. 为证书和密钥文件设置适当的权限：

```
chown root:awx /etc/tower/tower.cert /etc/tower/tower.key
chmod 0600 /etc/tower/tower.cert /etc/tower/tower.key
```

7. 测试 NGINX 配置：

```
nginx -t
```

8. 重新载入 NGINX:

```
systemctl reload nginx.service
```

9. 验证是否安装了新的 SSL 证书和密钥：

```
true | openssl s_client -showcerts -connect ${CONTROLLER_FQDN}:443
```

8.2.3.2. 在 OpenShift Container Platform 上更改自动化控制器中的 SSL 证书和密钥

以下流程描述了如何为 OpenShift Container Platform 上运行的自动化控制器更改 SSL 证书和密钥。

流程

1. 将签名的 SSL 证书和密钥复制到安全位置。
2. 在 OpenShift 中创建 TLS secret：

```
oc create secret tls ${CONTROLLER_INSTANCE}-certs-$(date +%F) --cert=/path/to/ssl.crt --
key=/path/to/ssl.key
```

3. 修改自动化控制器自定义资源，将 **route_tls_secret** 和新 secret 的名称添加到 spec 部分。

```
oc edit automationcontroller/${CONTROLLER_INSTANCE}
```

```
...
spec:
  route_tls_secret: automation-controller-certs-2023-04-06
...
```

TLS secret 的名称是任意的。在本例中，它的时间戳为创建 secret 的日期，以便将其与应用到自动化控制器实例的其他 TLS secret 进行区分。

1. 等待几分钟，以便应用更改。
2. 验证是否安装了新的 SSL 证书和密钥：

```
true | openssl s_client -showcerts -connect ${CONTROLLER_FQDN}:443
```

8.2.3.3. 在 Event-Driven Ansible 控制器上更改 SSL 证书和密钥

以下流程描述了如何在 Event-Driven Ansible 控制器上手动更改 SSL 证书和密钥。

流程

1. 备份当前的 SSL 证书：

```
cp /etc/ansible-automation-platform/eda/server.cert /etc/ansible-automation-platform/eda/server.cert-$(date +%F)
```

2. 备份当前密钥文件：

```
cp /etc/ansible-automation-platform/eda/server.key /etc/ansible-automation-platform/eda/server.key-$(date +%F)
```

3. 将新 SSL 证书复制到 **/etc/ansible-automation-platform/eda/server.cert**。
4. 将新密钥复制到 **/etc/ansible-automation-platform/eda/server.key**。
5. 恢复 SELinux 上下文：

```
restorecon -v /etc/ansible-automation-platform/eda/server.cert /etc/ansible-automation-platform/eda/server.key
```

6. 为证书和密钥文件设置适当的权限：

```
chown root:eda /etc/ansible-automation-platform/eda/server.cert /etc/ansible-automation-platform/eda/server.key
```

```
chmod 0600 /etc/ansible-automation-platform/eda/server.cert /etc/ansible-automation-platform/eda/server.key
```

7. 测试 NGINX 配置：

```
nginx -t
```

8. 重新载入 NGINX:

```
systemctl reload nginx.service
```

9. 验证是否安装了新的 SSL 证书和密钥：

```
true | openssl s_client -showcerts -connect ${CONTROLLER_FQDN}:443
```

8.2.3.4. 在自动化中心中手动更改 SSL 证书和密钥

以下流程描述了如何在自动化中心中手动更改 SSL 证书和密钥。

流程

1. 备份当前的 SSL 证书：

```
cp /etc/pulp/certs/pulp_webserver.crt /etc/pulp/certs/pulp_webserver.crt-$(date +%F)
```

2. 备份当前密钥文件：

```
cp /etc/pulp/certs/pulp_webserver.key /etc/pulp/certs/pulp_webserver.key-$(date +%F)
```

3. 将新 SSL 证书复制到 **/etc/pulp/certs/pulp_webserver.crt**。
4. 将新密钥复制到 **/etc/pulp/certs/pulp_webserver.key**。
5. 恢复 SELinux 上下文：

```
restorecon -v /etc/pulp/certs/pulp_webserver.crt /etc/pulp/certs/pulp_webserver.key
```

6. 为证书和密钥文件设置适当的权限：

```
chown root:pulp /etc/pulp/certs/pulp_webserver.crt /etc/pulp/certs/pulp_webserver.key
```

```
chmod 0600 /etc/pulp/certs/pulp_webserver.crt /etc/pulp/certs/pulp_webserver.key
```

7. 测试 NGINX 配置：

```
nginx -t
```

8. 重新载入 NGINX:

```
systemctl reload nginx.service
```

9. 验证是否安装了新的 SSL 证书和密钥：

```
true | openssl s_client -showcerts -connect ${CONTROLLER_FQDN}:443
```