Red Hat Enterprise Linux 8

使用 RHEL 中的系统角色管理和配置任务

使用 Red Hat Ansible Automation Platform playbook 应用 RHEL 系统角色来执行系统管理任务。

Last Updated: 2020-12-23

Red Hat Enterprise Linux 8 使用 RHEL 中的系统角色管理和配置任务

使用 Red Hat Ansible Automation Platform playbook 应用 RHEL 系统角色来执行系统管理任务。

法律通告

摘要

本文档论述了在 Red Hat Enterprise Linux 8 上使用 Ansible 配置系统角色。本文档侧重于：RHEL System Roles 是 Ansible 角色、模块和 playbook 的集合，它提供了一个稳定的、一致的配置界面来管理和配置 Red Hat Enterprise Linux。它们旨在转发与 Red Hat Enterprise Linux 8 的多个主发行版本兼容。

目录

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始： master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。如需了解更多详细信息，请参阅 CTO Chris Wright 信息。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。要做到这一点：

- 关于特定内容的简单评论：
  1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
  2. 用鼠标指针高亮显示您想评论的文本部分。
  3. 点在高亮文本上弹出的 **Add Feedback**。
  4. 按照显示的步骤操作。
- 要提交更复杂的反馈，请创建一个 Bugzilla ticket：
  1. 进入 Bugzilla 网站。
  2. 在 Component 中选择 **Documentation**。
  3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
  4. 点 **Submit Bug**。

第 1 章 RHEL 系统角色入门

本节介绍 RHEL 系统角色是什么。另外，它介绍了如何通过 Ansible playbook 应用特定角色来执行各种系统管理任务。

1.1. RHEL 系统角色简介

RHEL 系统角色是 Ansible 角色和模块的集合。RHEL 系统角色提供了一个配置界面，用于远程管理多个 RHEL 系统。这个界面允许在多个 RHEL 版本间管理系统配置，以及处理新的主发行版本。

在 Red Hat Enterprise Linux 8 中，该接口目前由以下角色组成：

- kdump
- network
- selinux
- storage
- certificate
- kernel_settings
- logging
- metrics
- nbde_client and nbde_server
- timesync
- tlog

所有这些角色都由 **AppStream** 存储库中可用的 **rhel-system-roles** 软件包提供。

其它资源

- 有关 RHEL 系统角色概述，请查看红帽知识库中的 Red Hat Enterprise Linux(RHEL)系统角色 。
- 有关特定角色的详情请查看 **/usr/share/doc/rhel-system-roles** 目录中的文档。本文档会随 **rhel-system-roles** 软件包自动安装。
- SELinux 系统角色简介
- 存储角色简介

1.2. RHEL 系统角色术语

您可以在本文档中找到以下术语：

系统角色术语

Ansible playbook

Playbook 是 Ansible 的配置、部署和编配语言。它们可以描述您希望远程系统强制使用的策略，或者在一般的 IT 进程中选择一组步骤。

控制节点

安装了 Ansible 的任何机器。您可以从任何控制节点运行命令和 playbook，调用 /usr/bin/ansible 或 /usr/bin/ansible-playbook。您可以使用任意安装了 Python 的计算机作为控制节点 - 笔记本电脑、共享桌面和服务器都可以运行 Ansible。但是，您不能使用 Windows 机器作为控制节点。您可以拥有多个控制节点。

清单（Inventory）

受管节点列表。清单文件有时也称为"hostfile"。您的清单可以为每个受管节点指定像 IP 地址等信息。清单也可以管理受管节点，创建并嵌套组以更轻松地进行扩展。如需了解更多有关清单的信息，请参阅使用清单一 节。

受管节点

使用 Ansible 管理的网络设备、服务器或两者。受管节点有时也称为 "hosts（主机）"。Ansible 未安装到受管节点上。

1.3. 应用一个角色

以下流程描述了如何应用特定角色。

先决条件

- **rhel-system-roles** 软件包安装在您要用作控制节点的系统中：

```
# yum install rhel-system-roles
```

- 启用 Ansible Engine 软件仓库, **ansible** 软件包安装在您要用作控制节点的系统中。您需要 **ansible** 软件包来运行使用 RHEL 系统角色的 playbook。
  - 如果您没有 Red Hat Ansible Engine 订阅，可以使用 Red Hat Enterprise Linux 订阅提供的有限版本的 Red Hat Ansible Engine。在这种情况下，请按照以下步骤操作：
    1. 启用 RHEL Ansible Engine 存储库：

```
# subscription-manager refresh
# subscription-manager repos --enable ansible-2-for-rhel-8-x86_64-rpms
```

    2. 安装 Ansible Engine:

```
# yum install ansible
```

  - 如果您有 Red Hat Ansible Engine 订阅，请按照以下所述步骤操作 如何下载和安装 Red Hat Ansible Engine?。

- 您可以创建一个 Ansible playbook。
  Playbook 代表 Ansible 的配置、部署和编配语言。通过使用 playbook，您可以声明和管理远程机器的配置，部署多个远程机器，编配任何手动排序进程的步骤。

  playbook 是一个或多个 **plays** 的列表。每个 **play** 都可以包括 Ansible 变量、任务或角色。

  playbook 是人类可读的,以 **YAML** 格式表示。

  如需有关 playbook 的更多信息，请参阅 Ansible 文档。

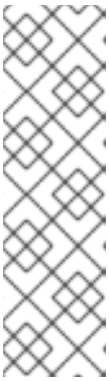
流程

1. 创建一个 Ansible playbook，包括所需角色。
   以下示例演示了如何通过给定 **roles:** 选项使用角色 **play**:

```
---
- hosts: webservers
  roles:
    - rhel-system-roles.network
    - rhel-system-roles.timesync
```

   如需有关在 playbook 中使用角色的更多信息，请参阅 Ansible 文档。

   请参阅 playbook 示例 Ansible 示例。



注意

每个角色都包括 README 文件，该文件记录如何使用角色和支持的参数值。您还可以在角色的文档目录中找到特定角色的示例 playbook。这些文档目录默认由 **rhel-system-roles** 软件包提供，并可在以下位置找到：

```
/usr/share/doc/rhel-system-roles/SUBSYSTEM/
```

使用所需角色的名称（如**selinux kdump**、**network**、**timesync**、或**storage**）替换 *SUBSYSTEM* 。

2. 运行以下 **ansible-playbook** 命令，在目标主机上执行 playbook:

```
# ansible-playbook -i name.of.the.inventory name.of.the.playbook
```

清单是 Ansible 支持的系统的列表。更多相关信息，请参阅 Ansible 文档。

*如果没有清单，您可以在运行 **ansible-playbook** 时创建它：*

如果您只有一个目标主机要针对其运行 playbook，请使用：

```
# ansible-playbook -i host1, name.of.the.playbook
```

如果您有多个要运行 playbook 的目标主机，请使用：

```
# ansible-playbook -i host1,host2,....,hostn name.of.the.playbook
```

其它资源

- *有关使用 **ansible-playbook** 命令的详情请参考 **ansible-playbook** man page。*

1.4. 其它资源

- *有关 RHEL 系统角色概述，请查看红帽知识库中的 Red Hat Enterprise Linux(RHEL)系统角色。*
- *使用 RHEL 系统角色管理本地存储*
- *在多个系统中使用 RHEL 系统角色部署相同的 SELinux 配置*

第 2 章 安装 RHEL 系统角色

在开始使用系统角色前，您必须在您的系统中安装它。

2.1. 在系统中安装 RHEL 系统角色

这一段是过程模块简介：对流程的简短描述。

先决条件

- 您有一个 Red Hat Ansible Engine 订阅。参阅 如何下载和安装 Red Hat Ansible Engine 的步骤？
- 您已在系统中安装了 Ansible 软件包，以用作控制节点：

流程

1. 在您要用作控制节点的系统中安装该 **rhel-system-roles** 软件包：

```
# yum install rhel-system-roles
```

如果您没有 Red Hat Ansible Engine 订阅，可以使用 Red Hat Enterprise Linux 订阅提供的有限版本的 Red Hat Ansible Engine。在这种情况下，请按照以下步骤操作：

a. 启用 RHEL Ansible Engine 存储库：

```
# subscription-manager refresh

# subscription-manager repos --enable ansible-2-for-rhel-8-x86_64-rpms
```

b. 安装 Ansible Engine:

```
# yum install ansible
```

因此,您可以创建一个 Ansible playbook。

其它资源

- 有关 RHEL 系统角色概述,请查看 Red Hat Enterprise Linux(RHEL)系统角色。
- 有关使用 ansible-playbook 命令的详情,请参阅 ansible-playbook 手册页。

第 3 章 使用 ANSIBLE 角色永久配置内核参数

作为熟悉 Red Hat Ansible Engine 经验丰富的用户,您可以使用角色 **kernel_settings** 同时在多个客户端中配置内核参数。这个解决方案：

- 提供带有有效输入设置的友好界面。
- 在同一个位置保留所有预期内核参数。

在您从控制机器运行 **kernel_settings** 角色后，内核参数会立即应用于受管系统，并在重启后保留。

3.1. 内核设置角色简介

RHEL 系统角色是 Ansible Automation Platform 中的角色和模块集合，可提供一致的配置界面来远程管理多个系统。

RHEL 系统角色用于使用 **kernel_settings** 系统角色自动配置内核。**rhel-system-roles** 软件包包含这个系统角色以及参考文档。

要自动在一个或多个系统中应用内核参数，请在 playbook 中使用您选择的 **kernel_settings** 一个或多个角色变量的角色。playbook 是一个或多个 play 的列表,可人类可读,并以 YAML 格式编写。

您可以使用清单文件来定义一组您希望 Ansible Engine 根据 playbook 配置的系统。

使用 **kernel_settings** 角色，您可以配置：

- 使用 **kernel_settings_sysctl** 角色变量的内核参数
- 使用 **kernel_settings_sysfs** 角色变量的各种内核子系统、硬件设备和设备驱动程序
- **systemd** 服务管理者的 CPU 关联性,并使用 **kernel_settings_systemd_cpu_affinity** 角色变量处理其 fork
- 内核内存子系统使用 **kernel_settings_transparent_hugepages** 和 **kernel_settings_transparent_hugepages_defrag** 角色变量透明巨页

其它资源

- 如需有关 **kernel_settings** 角色变量和示例 playbook 的详细引用,安装 **rhel-system-roles** 软件包,以及查看 **README.md README.html /usr/share/doc/rhel-system-roles/kernel_settings/** 目录中的文件。
- 如需有关 playbook 的更多信息，请参阅 Ansible 文档中的 使用 playbook。
- 如需有关创建和使用清单的更多信息，请参阅 Ansible 文档中的如何构建您的清单。

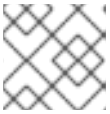
3.2. 使用内核设置角色应用所选内核参数

按照以下步骤准备并应用 Ansible playbook 来远程配置内核参数，从而对多个受管操作系统产生持久性。

先决条件

- 您的 Red Hat Ansible Engine 订阅已附加到系统,也称 控制机器,您要从其中运行 **kernel_settings** 角色。如需更多信息，请参阅如何下载和安装 Red Hat Ansible Engine 文章。

- 在控制机器上启用 Ansible Engine 软件仓库。
- Ansible Engine 已安装在控制机器上。



注意

您不需要在要配置内核参数的系统中（也称为 受管主机）安装 Ansible Engine。

- **rhel-system-roles** 软件包安装在控制机器上。
- 控制机器上存在受管主机的清单,Ansible Engine 能够连接到它们。

流程

1. 另外,还可查看 **inventory** 文件：

```
#  cat /home/jdoe/<ansible_project_name>/inventory
[testingservers]
pdoe@192.168.122.98
fdoe@192.168.122.226

[db-servers]
db1.example.com
db2.example.com

[webservers]
web1.example.com
web2.example.com
192.0.2.42
```

该文件定义 **[testingservers]** 组和其它组。它允许您针对特定系统集合更有效地运行 Ansible Engine。

2. 创建一个配置文件来为 Ansible Engine 操作设置默认和权限升级。

   a. 创建新 YAML 文件，并在文本编辑器中打开，例如：

```
#  vi /home/jdoe/<ansible_project_name>/ansible.cfg
```

   b. 将以下内容插入到文件中：

```
[defaults]
inventory = ./inventory

[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = true
```

   [defaults] 部分指定受管主机清单文件的路径。**[privilege_escalation]** 部分定义了将用户权限切换到指定的受管主机 **root** 上。这对成功配置内核参数是必需的。运行 Ansible playbook 时，会提示您输入用户密码。在连接到受管主机后，用户会通过 **sudo** 切换为 **root**。

3. 创建使用 **kernel_settings** 角色的 Ansible playbook。

a. 创建新 YAML 文件，并在文本编辑器中打开，例如：

```
# vi /home/jdoe/<ansible_project_name>/kernel_roles.yml
```

此文件代表一个 playbook，通常包含了一组有特定顺序的任务（也称为 play ）列表。这些任何会根据 **inventory** 文件中选择的特定管理主机进行。

b. 将以下内容插入到文件中：

```
---
- name: Configure kernel settings
  hosts: testingservers

  vars:
    kernel_settings_sysctl:
      - name: fs.file-max
        value: 400000
      - name: kernel.threads-max
        value: 65536
    kernel_settings_sysfs:
      - name: /sys/class/net/lo/mtu
        value: 65000
    kernel_settings_transparent_hugepages: madvise

  roles:
    - linux-system-roles.kernel_settings
```

name 键是可选的。它把任意字符串与这个 play 相关联,作为一个标签来标识这个 play 的作用。play 中的 **hosts** 键指定 play 被运行的主机。此键的值或值可以作为被管理的主机的单独名称提供，也可以作为 **inventory** 文件中定义的一组主机提供。

vars 部分代表包含所选内核参数名称和值的变量列表。

roles 键指定要配置 **vars** 部分中提到的参数和值的系统角色。

注意

您可以修改 playbook 中的内核参数及其值,以满足您的需要。

4. （可选）验证 play 中的语法是否正确。

```
# ansible-playbook --syntax-check kernel-roles.yml

playbook: kernel-roles.yml
```

本例演示了 playbook 成功验证。

5. 执行 playbook。

```
# ansible-playbook kernel-roles.yml
BECOME password:

PLAY [Configure kernel settings] ... PLAY RECAP **
fdoe@192.168.122.226      : ok=10   changed=4    unreachable=0    failed=0    skipped=6
```

```
rescued=0    ignored=0
pdoe@192.168.122.98        : ok=10   changed=4    unreachable=0    failed=0    skipped=6
rescued=0    ignored=0
```

在 Ansible Engine 运行 playbook 之前，会提示您输入密码，以便受管主机上的用户能够切换至 **root**，这在配置内核参数时是必需的。

recap 部分显示所有受管主机的 play 都成功完成(**failed=0**)且应用了 4 个内核参数(**changed=4**)。

6. 重启您的受管主机并检查受影响的内核参数,以验证是否应用了更改并在重启后保留。

其它资源

- 有关 RHEL 系统角色的更多信息，参阅开始使用 RHEL 系统角色。
- 有关目录中所有当前 **kernel_settings** 中支持的变量的更多信息，请参阅 **/usr/share/doc/rhel-system-roles/kernel_settings/** 目录中的 **README.html** 和 **README.md** 文件。
- 如需有关 Ansible 清单的更多信息，请参阅 Ansible 文档中的使用清单。
- 如需有关 Ansible 配置文件的更多信息,请参阅 Ansible 文档中的配置 Ansible。
- 如需有关 Ansible playbook 的更多信息，请参阅 Ansible 文档中的使用 Playbook。
- 如需有关 Ansible 变量的更多信息,请参阅 Ansible 文档中 的使用变量。
- 如需有关 Ansible 角色的更多信息,请参阅 Ansible 文档中的角色。

第 4 章 使用系统角色配置网络连接

RHEL 上的 **network** 系统角色可让管理员使用 Ansible 自动执行与网络相关的配置和管理任务。

4.1. 配置以太网连接

这部分论述了如何使用静态和动态 IP 地址配置以太网连接的不同方法。

4.1.1. 使用 RHEL 系统角色配置静态以太网连接

此流程描述了如何通过运行 Ansible playbook 来使用 RHEL 系统角色远程为带有以下设置的 **enp7s0** 接口添加以太网连接：

- 静态 IPv4 地址 - **192.0.2.1** 子网掩码 /**24**
- 静态 IPv6 地址 - **2001:db8:1::1** 子网掩码 /**64**
- IPv4 默认网关 - **192.0.2.254**
- IPv6 默认网关 - **2001:db8:1::fffe**
- IPv4 DNS 服务器 - **192.0.2.200**
- IPv6 DNS 服务器 - **2001:db8:1::ffbb**
- DNS 搜索域 - **example.com**

在 Ansible 控制节点上运行此步骤。

先决条件

- 在控制节点上安装 **ansible** 和 **rhel-system-roles** 软件包。
- 如果您运行 playbook 时使用了与 **root** 不同的远程用户，则此用户在受管节点上需要具有适当的 **sudo** 权限。
- 主机使用 NetworkManager 配置网络。

流程

1. 如果要执行 playbook 中说明的主机尚未撤离，请将此主机的 IP 或名称添加到 **/etc/ansible/hosts** Ansible 清单文件中：

```
node.example.com
```

2. 使用以下内容创建 ~/**ethernet-static-IP.yml** playbook:

```
---
- name: Configure an Ethernet connection with static IP
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network
```

```
vars:
  network_connections:
    - name: enp7s0
      type: ethernet
      autoconnect: yes
      ip:
        address:
          - 192.0.2.1/24
          - 2001:db8:1::1/64
        gateway4: 192.0.2.254
        gateway6: 2001:db8:1::fffe
        dns:
          - 192.0.2.200
          - 2001:db8:1::ffbb
        dns_search:
          - example.com
      state: up
```

3. 运行 playbook:

   - 以 root 用户身份连接到受管主机，输入：

     ```
     # ansible-playbook -u root ~/ethernet-static-IP.yml
     ```

   - 要以用户连接到受管主机，请输入：

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/ethernet-static-IP.yml
     ```

     --ask-become-pass 选项可定义该 ansible-playbook 命令提示输入 -u user_name 选项中定义的用户的 sudo 密码。

   如果没有指定 -u user_name 选项，请以当前登录到控制节点的用户 ansible-playbook 连接到受管主机。

其它资源

- 有关使用的参数 network_connections 以及该 network 系统角色的额外信息,请查看该 /usr/share/ansible/roles/rhel-system-roles.network/README.md 文件。
- 有关该 ansible-playbook 命令的详情请参考 ansible-playbook(1) man page。

4.1.2. 使用 RHEL 系统角色配置动态以太网连接

此流程描述了如何使用 RHEL 系统角色通过运行 Ansible playbook 为 enp7s0 接口远程添加动态以太网连接。在这个设置中，网络连接从 DHCP 服务器请求这个连接的 IP 设置。在 Ansible 控制节点上运行此步骤。

先决条件

- 网络中有 DHCP 服务器。
- 在控制节点上安装 ansible 和 rhel-system-roles 软件包。
- 如果您运行 playbook 时使用了与 root 不同的远程用户, 则此用户在受管节点上需要具有适当的 sudo 权限。

- 主机使用 NetworkManager 配置网络。

流程

1. 如果要执行 playbook 中说明的主机尚未撤离，请将此主机的 IP 或名称添加到 /etc/ansible/hosts Ansible 清单文件中：

```
node.example.com
```

2. 使用以下内容创建 ~/ethernet-dynamic-IP.yml playbook:

```
---
- name: Configure an Ethernet connection with dynamic IP
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        - name: enp7s0
          type: ethernet
          autoconnect: yes
          ip:
            dhcp4: yes
            auto6: yes
          state: up
```

3. 运行 playbook:

   - 以 root 用户身份连接到受管主机，输入：

```
# ansible-playbook -u root ~/ethernet-dynamic-IP.yml
```

   - 要以用户连接到受管主机，请输入：

```
# ansible-playbook -u user_name --ask-become-pass ~/ethernet-dynamic-IP.yml
```

     该 --ask-become-pass 选项可定义该 ansible-playbook 命令提示输入 -u user_name 选项中定义的用户的 sudo 密码。

   如果没有指定 -u user_name 选项，请以当前登录到控制节点的用户 ansible-playbook 连接到受管主机。

其它资源

- 有关使用的参数 network_connections 以及 network 系统角色的额外信息，请查看 /usr/share/ansible/roles/rhel-system-roles.network/README.md 文件。
- 有关 ansible-playbook 命令的详情，请参考 ansible-playbook(1) man page。

4.2. 使用 802.1X 标准向网络验证 RHEL 客户端

管理员通常使用基于 IEEE 802.1X 标准的基于端口的网络访问控制(NAC)来保护网络不受未授权 LAN 和 Wi-Fi 客户端的影响。本节中的步骤描述了配置网络身份验证的不同选项。

4.2.1. 使用 RHEL 系统角色通过 802.1X 网络身份验证配置静态以太网连接

使用 RHEL 系统角色，您可以自动创建使用 802.1X 标准验证客户端的以太网连接。此流程描述了如何通过运行 Ansible playbook 来远程为带有以下设置的 **enp1s0** 接口添加以太网连接：

- 静态 IPv4 地址 - **192.0.2.1** 子网掩码 **/24**
- 静态 IPv6 地址 - **2001:db8:1::1** 子网掩码 **/64**
- IPv4 默认网关 - **192.0.2.254**
- IPv6 默认网关 - **2001:db8:1::fffe**
- IPv4 DNS 服务器 - **192.0.2.200**
- IPv6 DNS 服务器 - **2001:db8:1::ffbb**
- DNS 搜索域 - **example.com**
- 802.1X 网络验证使用 **TLS** 可扩展验证协议(EAP)

在 Ansible 控制节点上运行此步骤。

先决条件

- 在控制节点上安装 **ansible** 和 **rhel-system-roles** 软件包。
- 如果您运行 playbook 时使用了与 **root** 不同的远程用户，则此用户在受管节点上需要具有适当的 **sudo** 权限。
- 网络支持 802.1X 网络身份验证。
- 受管节点使用 NetworkManager。
- control 节点上存在 TLS 身份验证所需的以下文件：
  - 保存在 **/srv/data/client.key** 文件中的客户端密钥。
  - 存储在 **/srv/data/client.crt** 文件中的客户端证书。
  - 存储在 **/srv/data/ca.crt** 文件中的证书颁发机构(CA)证书。

流程

1. 如果要执行 playbook 中说明的主机尚未撤离，请将主机的 IP 地址或名称添加到 **/etc/ansible/hosts** Ansible 清单文件中：

```
node.example.com
```

2. 使用以下内容创建 **~/enable-802.1x.yml** playbook:

```
---
- name: Configure an Ethernet connection with 802.1X authentication
```

```
hosts: node.example.com
become: true
tasks:
 - name: Copy client key for 802.1X authentication
   copy:
     src: "/srv/data/client.key"
     dest: "/etc/pki/tls/private/client.key"
     mode: 0600

 - name: Copy client certificate for 802.1X authentication
   copy:
     src: "/srv/data/client.crt"
     dest: "/etc/pki/tls/certs/client.crt"

 - name: Copy CA certificate for 802.1X authentication
   copy:
     src: "/srv/data/ca.crt"
     dest: "/etc/pki/ca-trust/source/anchors/ca.crt"

 - include_role:
     name: linux-system-roles.network
   vars:
     network_connections:
       - name: enp1s0
         type: ethernet
         autoconnect: yes
         ip:
           address:
             - 192.0.2.1/24
             - 2001:db8:1::1/64
           gateway4: 192.0.2.254
           gateway6: 2001:db8:1::fffe
           dns:
             - 192.0.2.200
             - 2001:db8:1::ffbb
           dns_search:
             - example.com
         ieee802_1x:
           identity: user_name
           eap: tls
           private_key: "/etc/pki/tls/private/client.key"
           private_key_password: "password"
           client_cert: "/etc/pki/tls/certs/client.crt"
           ca_cert: "/etc/pki/ca-trust/source/anchors/ca.crt"
           domain_suffix_match: example.com
         state: up
```

3. 运行 playbook:

   - 以 root 用户身份连接到受管主机，输入：

     ```
     # ansible-playbook -u root ~/enable-802.1x.yml
     ```

   - 要以用户连接到受管主机，请输入：

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/ethernet-static-IP.yml
     ```

- **--ask-become-pass** 选项可定义该 **ansible-playbook** 命令提示输入 **-u user_name** 选项中定义的用户的 **sudo** 密码。

如果没有指定 **-u user_name** 选项，请以当前登录到控制节点的用户 **ansible-playbook** 连接到受管主机。

其它资源

- 有关使用的参数 **network_connections** 以及 **network** 系统角色的额外信息，请查看 **/usr/share/ansible/roles/rhel-system-roles.network/README.md** 文件。
- 有关 802.1X 参数的详情,请查看文件中的 **ieee802_1x** 章节 **/usr/share/ansible/roles/rhel-system-roles.network/README.md**。
- 有关 **ansible-playbook** 命令的详情请参考 **ansible-playbook(1)** man page。

第 5 章 使用系统角色配置 SELINUX

5.1. SELINUX 系统角色简介

RHEL 系统角色是 Ansible 角色和模块的集合，可为远程管理多个 RHEL 系统提供一致的配置界面。SELinux 系统角色启用以下操作：

- *清理与 SELinux 布尔值、文件上下文、端口和登录相关的本地策略修改。*
- *设置 SELinux 策略布尔值、文件上下文、端口和登录。*
- *在指定文件或目录中恢复文件上下文。*

下表提供了 SELinux 系统角色中可用的输入变量概述。

表 5.1. SELinux 系统角色变量

角色变量	描述	CLI 备选
selinux_policy	选择保护目标进程或多级别安全保护策略。	**/etc/selinux/config** 中的 **SELINUXTYPE**
selinux_state	切换 SELinux 模式。请查看 **ansible-doc selinux**	**setenforce** 以及 **/etc/selinux/config** 下的**SELINUX**。
selinux_booleans	启用和禁用 SELinux 布尔值。请参阅 **ansible-doc seboolean**。	**setsebool**
selinux_fcontexts	添加或删除 SELinux 文件上下文映射。请参阅 **ansible-doc sefcontext**。	**semanage fcontext**
selinux_restore_dirs	在文件系统树中恢复 SELinux 标签。	**restorecon -R**
selinux_ports	在端口上设置 SELinux 标签。请参阅 **ansible-doc seport**。	**semanage port**
selinux_logins	将用户设置为 SELinux 用户映射。请参阅 **ansible-doc selogin**。	**semanage login**

***rhel-system-roles** 软件包安装的 playbook **/usr/share/doc/rhel-system-roles/selinux/example-selinux-playbook.yml** 示例演示了如何在 enforcing 模式中设置目标策略。playbook 还应用一些本地策略修改，并在 **/tmp/test_dir/** 目录中恢复文件上下文。*

其它资源

- *有关 SELinux 角色变量的详情，安装 **rhel-system-roles**，并参阅 **/usr/share/doc/rhel-system-roles/selinux/** 目录中的 **README.md** 或 **README.html**。*
- *有关 RHEL 系统角色的更多信息，请参阅 RHEL 系统角色简介*

5.2. 使用 SELINUX 系统角色在多个系统中应用 SELINUX 设置

按照以下步骤，在已验证的 SELinux 设置中准备并应用 Ansible playbook。

先决条件

- *您的 Red Hat Ansible Engine 订阅已附加到系统。如需更多信息，请参阅如何下载和安装 Red Hat Ansible Engine 文章。*

流程

1. *启用 RHEL Ansible 存储库，例如：*

```
# subscription-manager repos --enable ansible-2-for-rhel-8-x86_64-rpms
```

2. *安装 Ansible Engine:*

```
# yum install ansible
```

3. *安装 RHEL 系统角色：*

```
# yum install rhel-system-roles
```

4. *使用 SELinux 系统角色应用您的 playbook。*
   以下命令应用一个 playbook 示例，它是 rhel-system-roles 软件包的一部分。您可以使用此 playbook 作为模板：

```
# ansible-playbook -i host1,host2,host3 /usr/share/doc/rhel-system-roles/selinux/example-selinux-playbook.yml
```

其它资源

- *如需更多信息，安装 rhel-system-roles 软件包以及查看 /usr/share/doc/rhel-system-roles/selinux/ 和 /usr/share/ansible/roles/rhel-system-roles.selinux/ 目录。*

第 6 章 使用日志记录系统角色

作为系统管理员，您可以使用日志记录系统角色将 RHEL 主机配置为日志服务器，从很多客户端系统收集日志。

6.1. 日志系统角色

使用日志记录系统角色，您可以在本地和远程主机上部署日志配置。

要在一个或多个系统中应用日志记录系统角色，您可以在 playbook 中定义日志配置。playbook 是一个或多个 play 的列表。playbook 是人类可读的，它们采用 YAML 格式编写。如需有关 playbook 的更多信息，请参阅 Ansible 文档中的 使用 playbook。

您希望 Ansible 根据 playbook 配置的系统集合已在 清单文件中定义。如需有关创建和使用清单的更多信息，请参阅 Ansible 文档中的如何构建您的清单。

日志记录解决方案提供多种读取日志和多个日志记录输出的方法。

例如，日志记录系统可接受以下输入：

- 本地文件、
- **systemd/journal,**
- 网络中的另一个日志记录系统。

另外，日志记录系统还可有以下输出：

- 日志存储在 **/var/log** 目录中的本地文件中,
- 日志发送到 Elasticsearch,
- 日志转发到另一个日志系统。

使用日志记录系统角色,您可以组合输入和输出来满足您的需要。例如,您可以配置日志解决方案,它存储来自本地文件 **journal** 中的输入,而从文件读取的输入都会转发到另一个日志记录系统,并存储在本地日志文件中。

6.2. 日志记录系统角色参数

在日志记录系统角色 playbook 中,您可以在 **logging_inputs** 参数中定义输入、**logging_outputs** 参数中的输出以及 **logging_flows** 参数中输入和输出之间的关系。Logging 系统角色使用附加选项处理这些变量来配置日志记录系统。您还可以启用加密。

注意

目前，日志记录系统角色中唯一可用的日志记录系统是 Rsyslog。

- **logging_inputs** - 日志记录解决方案的输入列表。
  - **name** - 输入的唯一名称。在 **logging_flows** 输入列表和生成 **config** 文件名称的一部分中使用。
  - **type** - 输入元素的类型。这个类型指定与目录名称对应的任务类型 **roles/rsyslog/{tasks,vars}/inputs/**。

- **basics** - 输入配置从 **systemd** 日志或 **unix** 套接字输入。
  - **kernel_message** - 如果设置为 **true**，则会加载 **imklog**。默认为 **false**。
  - **use_imuxsock** - 使用 **imuxsock** 替代 **imjournal**。默认为 **false**。
  - **ratelimit_burst** - 可在其中发送的最大信息数 **ratelimit_interval**。默认 **20000** **use_imuxsock** 为 false。默认 **200** **use_imuxsock** 为 true。
  - **ratelimit_interval** - 评估 **ratelimit_burst** 的时间间隔。如果 **use_imuxsock** 为 false，则默认为 600 秒。如果 **use_imuxsock** 为 true，则默认为 0。0 表示关闭速率限制。
  - **persist_state_interval** - Journal 状态会保留每个 **value** 信息。默认为 **10**。仅在 **use_imuxsock** 为 false 时有效。
- **files** - 输入配置本地文件输入。
- **remote** - 输入通过网络配置其他日志记录系统的输入。

- **state** - 配置文件的状态。**present** 或者 **absent**。默认为 **present**。

- **logging_outputs** - 日志解决方案的输出列表。
  - **files** - 输出配置输出到本地文件。
  - **forwards** - 输出配置输出到另一个日志记录系统。
  - **remote_files** - 输出将输出配置为另一个日志记录系统到本地文件。
- **logging_flows** - 定义 **logging_inputs** 和 **logging_outputs** 之间关系的流程列表。**logging_flows** 变量有以下键：
  - **name** - 流的唯一名称
  - **inputs** - **logging_inputs** 名称值列表
  - **outputs** - **logging_outputs** 名称值列表。

其它资源

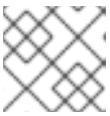
- 安装 **rhel-system-roles** 软件包，文档位于 **/usr/share/ansible/roles/rhel-system-roles.logging/README.html**

6.3. 应用本地日志记录系统角色

按照以下步骤准备和应用 Red Hat Ansible Engine playbook，在一组单独机器上配置日志记录解决方案。每台机器都会在本地记录日志。

先决条件

- 您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。



注意

您不必在要部署日志记录解决方案的系统中安装 Red Hat Ansible Engine。

- 您在要从中运行 **playbook** 的系统中具有 **rhel-system-roles** 软件包。

注意

*您不必安装 **rsyslog**，因为在部署后系统角色会安装 **rsyslog**。*

- 您有一个清单文件，它列出要配置日志记录解决方案的系统。

流程

1. 创建定义所需角色的 **playbook:**

   a. 创建新 **YAML** 文件，并在文本编辑器中打开，例如：

   ```
   # vi logging-playbook.yml
   ```

   b. 插入以下内容：

   ```
   ---
   - name: Deploying basics input and implicit files output
     hosts: all
     roles:
       - linux-system-roles.logging
     vars:
       logging_inputs:
         - name: system_input
           type: basics
       logging_outputs:
         - name: files_output
           type: files
       logging_flows:
         - name: flow1
           inputs: [system_input]
           outputs: [files_output]
   ```

2. 在特定清单上执行 **playbook:**

   ```
   # ansible-playbook -i inventory-file /path/to/file/logging-playbook.yml
   ```

   其中：

   - ***inventory-file*** 是清单文件。
   - ***logging-playbook.yml*** 是您使用的 **playbook**。

验证

1. 测试 **/etc/rsyslog.conf** 文件的语法：

   ```
   # rsyslogd -N 1
   rsyslogd: version 8.1911.0-6.el8, config validation run (level 1), master config
   /etc/rsyslog.conf
   rsyslogd: End of config validation run. Bye.
   ```

2. 验证系统是否向日志发送信息：

   a. 发送测试信息：

   ```
   # logger test
   ```

   b. 查看 **/var/log/messages** 日志，例如：

   ```
   # cat /var/log/messages
   Aug  5 13:48:31 hostname root[6778]: test
   ```

   其中 **'hostname'** 是客户端系统的主机名。请注意，该日志包含输入 **logger** 命令的用户的用户名，在本例中是 **root**。

6.4. 使用日志记录系统角色应用远程日志解决方案

按照以下步骤准备和应用 **Red Hat Ansible Engine playbook** 来配置远程日志记录解决方案。在这个 **playbook** 中，一个或多个客户端从 **systemd-journal** 中提取日志，并将其转发到远程服务器。服务器从 **remote_rsyslog** 和 **remote_files** 接收远程输入，并输出日志到由远程主机名命名的目录的的本地文件中。

先决条件

- 您已在要运行 **playbook** 的系统中安装了 **Red Hat Ansible Engine**。

  注意

  您不必在要部署日志记录解决方案的系统中安装 **Red Hat Ansible Engine**。

- 您在要从中运行 **playbook** 的系统中具有 **rhel-system-roles** 软件包。

  注意

  您不必安装 **rsyslog**，因为在部署后系统角色会安装 **rsyslog**。

- 您至少有两个系统：
  - 至少一个是日志服务器。
  - 至少一个是日志记录客户端。

流程

1. 创建定义所需角色的 **playbook:**

   a. 创建新 **YAML** 文件，并在文本编辑器中打开，例如：

   ```
   # vi logging-playbook.yml
   ```

   b. 将以下内容插入到文件中：

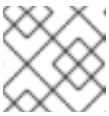
   ```
   ---
   - name: Deploying remote input and remote_files output
   ```

```
  hosts: server
  roles:
    - linux-system-roles.logging
  vars:
    logging_inputs:
      - name: remote_udp_input
        type: remote
        udp_ports: [ 601 ]
      - name: remote_tcp_input
        type: remote
        tcp_ports: [ 601 ]
    logging_outputs:
      - name: remote_files_output
        type: remote_files
    logging_flows:
      - name: flow_0
        inputs: [remote_udp_input, remote_tcp_input]
        outputs: [remote_files_output]

- name: Deploying basics input and forwards output
  hosts: clients
  roles:
    - linux-system-roles.logging
  vars:
    logging_inputs:
      - name: basic_input
        type: basics
    logging_outputs:
      - name: forward_output0
        type: forwards
        severity: info
        target: host1.example.com
        udp_port: 601
      - name: forward_output1
        type: forwards
        facility: mail
        target: host1.example.com
        tcp_port: 601
    logging_flows:
      - name: flows0
        inputs: [basic_input]
        outputs: [forward_output0, forward_output1]

[basic_input]
[forward_output0, forward_output1]
```

*其中 **host1.example.com** 是日志服务器。*



注意

*您可以修改 **playbook** 中的参数以符合您的需要。*

警告

日志解决方案只适用于在服务器或者客户端系统的 SELinux 策略中定义的端口并在防火墙中打开。默认 SELinux 策略包括端口 601、514、6514、10514 和 20514。要使用不同的端口,请 在客户端系统和服务器系统中修改 SELinux 策略。目前还不支持通过系统角色配置防火墙。

2. *创建列出您的服务器和客户端的清单文件：*

   a. *创建新文件并在文本编辑器中打开该文件，例如：*

   ```
   # vi inventory.ini
   ```

   b. *将以下内容插入到清单文件中：*

   ```
   [servers]
   server ansible_host=host1.example.com
   [clients]
   client ansible_host=host2.example.com
   ```

   其中： ***host1.example.com** 是日志服务器。* ***host2.example.com** 是日志记录客户端。*

3. *在清单上执行 **playbook**。*

   ```
   # ansible-playbook -i /path/to/file/inventory.ini /path/to/file/_logging-playbook.yml
   ```

   其中：

   - ***inventory.ini** 是清单文件。*
   - ***logging-playbook.yml** 是您创建的 playbook。*

验证步骤

1. *在客户端和服务器系统中测试 **/etc/rsyslog.conf** 文件的语法：*

   ```
   # rsyslogd -N 1
   rsyslogd: version 8.1911.0-6.el8, config validation run (level 1), master config
   /etc/rsyslog.conf
   rsyslogd: End of config validation run. Bye.
   ```

2. *验证客户端系统向服务器发送信息：*

   a. *在客户端系统中发送测试信息：*

   ```
   # logger test
   ```

   b. *在服务器系统中查看 **/var/log/messages** 日志，例如：*

```
# cat /var/log/messages
Aug  5 13:48:31 host2.example.com root[6778]: test
```

其中 **host2.example.com** 是客户端系统的主机名。请注意，该日志包含输入 **logger** 命令的用户的用户名，在本例中是 **root**。

其它资源

- RHEL 系统角色入门
- 安装 **rhel-system-roles** 软件包，文档位于 **/usr/share/ansible/roles/rhel-system-roles.logging/README.html**
- RHEL 系统角色知识库文章

6.5. 其它资源

- RHEL 系统角色入门
- 安装 **rhel-system-roles** 软件包，文档位于 **/usr/share/ansible/roles/rhel-system-roles.logging/README.html**
- RHEL 系统角色知识库文章

第 7 章 使用 CLEVIS 和 TANG 系统角色

7.1. CLEVIS 和 TANG 系统角色介绍

RHEL 系统角色是 Ansible 角色和模块的集合，可为远程管理多个 RHEL 系统提供一致的配置界面。

RHEL 8.3 引入了 Ansible 角色，用于使用 Clevis 和 Tang 自动部署基于策略解密(PBD)解决方案。**rhel-system-roles** 软件包包含这些系统角色、相关示例以及参考文档。

nbde_client 系统角色可让您自动部署多个 Clevis 客户端。请注意，**nbde_client** 角色只支持 Tang 绑定，您目前不能在 TPM2 绑定中使用它。

通过 **nbde_server** 角色，您可以部署和管理 Tang 服务器作为自动磁盘加密解决方案的一部分。此角色支持以下功能：

- 轮转 Tang 密钥
- 部署和备份 Tang 密钥

其它资源

- 有关 Network-Bound Disk Encryption(NBDE)角色变量的详细参考，安装 **rhel-system-roles** 软件包，查看 **README.md** 和 **README.html** 文件 **/usr/share/doc/rhel-system-roles/nbde_client/** 和 **/usr/share/doc/rhel-system-roles/nbde_server/** 目录里的文件。
- 例如 system-roles playbook，安装 **rhel-system-roles** 软件包并查看 **/usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/** 目录。
- 有关 RHEL 系统角色的更多信息，请参阅 RHEL 系统角色简介

7.2. 使用 NBDE_SERVER 系统角色设置多个 TANG 服务器

按照以下步骤准备并应用包含您的 Tang-server 设置的 Ansible playbook。

先决条件

- 您的 Red Hat Ansible Engine 订阅已附加到系统。如需更多信息，请参阅如何下载和安装 Red Hat Ansible Engine 文章。

流程

1. 启用 RHEL Ansible 存储库，例如：

```
# subscription-manager repos --enable ansible-2-for-rhel-8-x86_64-rpms
```

2. 安装 Ansible Engine:

```
# yum install ansible
```

3. 安装 RHEL 系统角色：

```
# yum install rhel-system-roles
```

4. 准备包含 Tang 服务器设置的 playbook。您可以从头开始，或使用 **/usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/** 目录中的示例 playbook。

```
# cp /usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/simple_deploy.yml
./my-tang-playbook.yml
```

5. 在您选择的文本编辑器中编辑 playbook，例如：

```
# vi my-tang-playbook.yml
```

6. 添加所需参数。以下示例 playbook 可确保部署 Tang 服务器和密钥轮转：

```
---
- hosts: all

  vars:
    nbde_server_rotate_keys: yes

  roles:
    - linux-system-roles.nbde_server
```

7. 应用完成的 playbook:

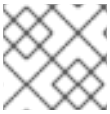
```
# ansible-playbook -i host1,host2,host3 my-tang-playbook.yml
```

其它资源

- 如需更多信息，请安装 **rhel-system-roles** 软件包并查看 **/usr/share/ansible/roles/rhel-system-roles.nbde_server/** 目录和 **/usr/share/doc/rhel-system-roles/nbde_server/** 目录。

7.3. 使用 NBDE_CLIENT 系统角色设置多个 CLEVIS 客户端

按照以下步骤准备并应用包含 Clevis-client 设置的 Ansible playbook。



注意

nbde_client 系统角色只支持 Tang 绑定。这意味着您目前无法将其用于 TPM2 绑定。

先决条件

- 您的 Red Hat Ansible Engine 订阅已附加到系统。如需更多信息，请参阅*如何下载和安装 Red Hat Ansible Engine* 文章。

流程

1. 启用 RHEL Ansible 存储库，例如：

```
# subscription-manager repos --enable ansible-2-for-rhel-8-x86_64-rpms
```

2. 安装 Ansible Engine:

```
# yum install ansible
```

3. 安装 **RHEL** 系统角色：

```
# yum install rhel-system-roles
```

4. 准备包含 **Clevis** 客户端设置的 **playbook**。您可以从头开始,或使用 **/usr/share/ansible/roles/rhel-system-roles.nbde_client/examples/** 目录中的示例 **playbook**。

```
# cp /usr/share/ansible/roles/rhel-system-roles.nbde_client/examples/high_availability.yml
./my-clevis-playbook.yml
```

5. 在您选择的文本编辑器中编辑 **playbook**，例如：

```
# vi my-clevis-playbook.yml
```

6. 添加所需参数。以下示例 **playbook** 配置 **Clevis** 客户端,以便在两个 **Tang** 服务器中至少有一个可用时自动解锁两个 **LUKS** 加密卷：

```
---
- hosts: all

  vars:
    nbde_client_bindings:
      - device: /dev/rhel/root
        key_file: /etc/luks/keyfile
        servers:
          - http://server1.example.com
          - http://server2.example.com
      - device: /dev/rhel/swap
        key_file: /etc/luks/keyfile
        servers:
          - http://server1.example.com
          - http://server2.example.com

  roles:
    - linux-system-roles.nbde_client
```

7. 应用完成的 **playbook:**

```
# ansible-playbook -i host1,host2,host3 my-clevis-playbook.yml
```

其它资源

- 如需了解有关该角色的参数和附加信息 **nbde_client**,安装 **rhel-system-roles** 软件包并查看 **/usr/share/doc/rhel-system-roles/nbde_client/** /**usr/share/ansible/roles/rhel-system-roles.nbde_client/** 目录的详情。

第 8 章 使用 RHEL 系统角色请求证书

通过证书系统角色，您可以使用 Red Hat Ansible Engine 发布和管理证书。

本章涵盖了以下主题：

- 证书系统角色
- 使用证书系统角色请求新的自签名证书
- 使用证书系统角色从 IdM CA 请求一个新证书

8.1. 证书系统角色

通过使用证书系统角色，您可以使用 Red Hat Ansible Engine 管理并更新 TLS 和 SSL 证书。

该角色使用 **certmonger** 作为证书提供商，目前支持发布和更新自签名证书，并使用 IdM 集成证书颁发机构 (CA)。

您可以将 Ansible playbook 中的以下变量与证书系统角色结合使用：

- **certificate_wait** 指定任务是否应该等待签发证书。
- 代表要发布的每个证书及其参数的**certificate_requests**。

其它资源

- 有关 **certificate_requests** 变量使用的参数的详情，以及 **certificate** 系统角色的附加信息，请参阅该 **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md** 文件。
- 有关 RHEL 系统角色以及如何应用它们的详情，请参阅 RHEL 系统角色入门。

8.2. 使用证书系统角色请求新的自签名证书

使用证书系统角色，您可以使用 Red Hat Ansible Engine 发布自签名证书。

此过程使用 **certmonger** 供应商并通过 **getcert** 命令请求证书。

注意

默认情况下，在证书过期前 **certmonger** 自动尝试更新该证书。您可以通过将 Ansible playbook 中的 **auto_renew** 参数设置为 **no** 来禁用此功能。

先决条件

- 您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。

注意

您不必在要部署 **certificate** 解决方案的系统中安装 Ansible。

- 已安装该系统中要运行 playbook 的 **rhel-system-roles** 软件包。
  有关 RHEL 系统角色以及如何应用它们的详情，请参阅 RHEL 系统角色入门。

流程

1. 可选：创建一个清单文件，例如 **inventory.file**：

```
$ touch inventory.file
```

2. 打开清单文件并定义要请求证书的主机，例如：

```
[webserver]
server.idm.example.com
```

3. 创建 playbook 文件，例如 **request-certificate.yml**：
   - 设置 **hosts** 为包含您要请求证书的主机，如 **webserver**。
   - 将 **certificate_requests** 变量设置为包含以下内容：
     - 将参数设置 **name** 为证书的所需名称，如 **mycert**。
     - 将 **dns** 参数设置为证书中包含的域，如 ***.example.com**。
     - 将参数 **ca** 设置为 **self-sign**。
   - 在 **roles** 下设置 **rhel-system-roles.certificate** 角色。
     这是本例的 playbook 文件：

```
---
- hosts: webserver

  vars:
    certificate_requests:
      - name: mycert
        dns: *.example.com
        ca: self-sign

  roles:
    - rhel-system-roles.certificate
```

4. 保存该文件。

5. 运行 playbook：

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

其它资源

- 有关 **certificate_requests** 变量使用的参数的详情，以及 **certificate** 系统角色的附加信息，请参阅该 **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md** 文件。
- 有关 **ansible-playbook** 命令的详情请参考 **ansible-playbook(1)** man page。

8.3. 使用证书系统角色从 IDM CA 请求一个新证书

使用证书系统角色时,您可以使用 Red Hat Ansible Engine 在使用带有集成证书颁发机构(CA)的 IdM 服务器时,使用 Red Hat Ansible Engine 发布证书。因此,在使用 IdM 作为 CA 时,您可以迅速、持续地管理多个系统的证书信任链。

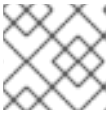
此过程使用 **certmonger** 供应商并通过 **getcert** 命令请求证书。

注意

默认情况下，在证书过期前 **certmonger** 自动尝试更新该证书。您可以通过将 Ansible playbook 中的 **auto_renew** 参数设置为 **no** 来禁用此功能。

先决条件

- 您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。



注意

您不必在要部署 **certificate** 解决方案的系统中安装 Ansible。

- 已安装该系统中要运行 playbook 的 **rhel-system-roles** 软件包。
  有关 RHEL 系统角色以及如何应用它们的详情，请参阅 RHEL 系统角色入门。

流程

1. 可选：创建一个清单文件，例如 **inventory.file**：

```
$ touch inventory.file
```

2. 打开清单文件并定义要请求证书的主机，例如：

```
[webserver]
server.idm.example.com
```

3. 创建 playbook 文件，例如 **request-certificate.yml**：
   - 设置 **hosts** 为包含您要请求证书的主机，如 **webserver**。
   - 将 **certificate_requests** 变量设置为包含以下内容：
     - 将参数设置 **name** 为证书的所需名称，如 **mycert**。
     - 将 **dns** 参数设置为证书中包含的域，如 **www.example.com**。
     - 将 **principal** 参数设置为特定的 Kerberos 主体，如 **HTTP/www.example.com@EXAMPLE.COM**。
     - 将参数 **ca** 设置为 **ipa**。
   - 在 **roles** 下设置 **rhel-system-roles.certificate** 角色。
     这是本例的 playbook 文件：

```
---
- hosts: webserver
  vars:
```

```
certificate_requests:
  - name: mycert
    dns: www.example.com
    principal: HTTP/www.example.com@EXAMPLE.COM
    ca: ipa

roles:
  - rhel-system-roles.certificate
```

4. 保存该文件。

5. 运行 playbook:

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

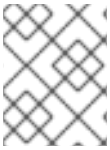
其它资源

- 有关 **certificate_requests** 变量使用的参数的详情，以及 **certificate** 系统角色的附加信息，请参阅该 **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md** 文件。
- 有关 **ansible-playbook** 命令的详情请参考 **ansible-playbook(1)** man page。

8.4. 指定在使用证书系统角色前或之后要运行的命令

使用证书系统角色，您可以使用 Red Hat Ansible Engine 在签发或更新证书之前和之后执行命令。

在以下示例中，管理员确保在为 **www.example.com** 发布或更新自签名证书前停止 **httpd** 服务，然后再重启该服务。

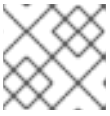


注意

默认情况下，在证书过期前 **certmonger** 自动尝试更新该证书。您可以通过将 Ansible playbook 中的 **auto_renew** 参数设置为**no**来禁用此功能。

先决条件

- 您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。



注意

您不必在要部署 **certificate** 解决方案的系统中安装 Ansible。

- 已安装该系统中要运行 playbook 的 **rhel-system-roles** 软件包。
  有关 RHEL 系统角色以及如何应用它们的详情，请参阅 RHEL 系统角色入门。

流程

1. 可选：创建一个清单文件，例如 **inventory.file**:

```
$ touch inventory.file
```

2. 打开清单文件并定义要请求证书的主机，例如：

```
[webserver]
server.idm.example.com
```

3. 创建 playbook 文件，例如 **request-certificate.yml**:

   - 设置 **hosts** 为包含您要请求证书的主机，如 **webserver**。
   - 将 **certificate_requests** 变量设置为包含以下内容：
     - 将参数设置 **name** 为证书的所需名称，如 **mycert**。
     - 将 **dns** 参数设置为证书中包含的域，如 **www.example.com**。
     - 将 **ca** 参数设置为您要用来发布证书的 CA，如 **self-sign**。
     - 将 **run_before** 参数设置为在签发或续订证书前要执行的命令，如 **systemctl stop httpd.service**。
     - 将 **run_after** 参数设置为发布或续订这个证书后要执行的命令，如 **systemctl start httpd.service**。
   - 在 **roles** 下设置 **rhel-system-roles.certificate** 角色。
     这是本例的 playbook 文件：

```
---
- hosts: webserver
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
        ca: self-sign
        run_before: systemctl stop httpd.service
        run_after: systemctl start httpd.service

  roles:
    - linux-system-roles.certificate
```

4. 保存该文件。

5. 运行 playbook:

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

其它资源

- 有关 **certificate_requests** 变量使用的参数的详情，以及 **certificate** 系统角色的附加信息，请参阅该 **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md** 文件。
- 有关 **ansible-playbook** 命令的详情请参考 **ansible-playbook(1)** man page。

第 9 章 使用 RHEL 系统角色配置 KDUMP

要使用 Ansible 管理 kdump，您可以使用 **kdump** 角色，该角色是 RHEL 8 中可用的 RHEL 系统角色之一。

使用 **kdump**，您可以指定保存系统内存内容的位置以便稍后进行分析。

有关 RHEL 系统角色以及如何应用它们的更多信息，请参阅 RHEL 系统角色简介。

9.1. KDUMP RHEL 系统角色

kdump 系统角色允许您在多个系统中设置基本内核转储参数。

9.2. KDUMP 角色参数

kdump RHEL 系统角色使用的参数有：

角色变量	描述
kdump_path	**vmcore** 写入的路径。如果 **kdump_target** 不是 null，则路径相对于那个转储目标。否则，它必须是 root 文件系统的绝对路径。

其它资源

- 请参阅 makedumpfile(8)man page。
- 有关 **kdump** 中使用的参数详情，以及 **kdump** 系统角色的信息，请参阅 **/usr/share/ansible/roles/rhel-system-roles.tlog/README.md** 文件。

9.3. 使用 RHEL 系统角色配置 KDUMP

您可以通过运行 Ansible playbook 在多个系统中使用 **kdump** 系统角色设置基本内核转储参数。

警告

kdump 角色通过替换该文件来完全取代受管主机的 kdump 配置 **/etc/kdump.conf**。另外，如果应用了 kdump 角色，则之前的所有 kdump 设置也会被替换，即使它们没有被角色变量指定，也会替换 **/etc/sysconfig/kdump** 文件。

先决条件

- 您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。

注意

*您不必在要部署 **kdump** 解决方案的系统中安装 Red Hat Ansible Automation Platform。*

- *已安装该系统中要运行 playbook 的 **rhel-system-roles** 软件包。*
- *您有一个清单文件，它列出了您要在其上部署的系统 **kdump**。*

流程

1. *使用以下内容 **playbook.yml** 创建新文件：*

```
---
- hosts: kdump-test
  vars:
    kdump_path: /var/crash
  roles:
    - rhel-system-roles.kdump
```

2. *可选：验证 **playbook** 语法。*

```
# ansible-playbook --syntax-check playbook.yml
```

3. *在清单文件上运行 **playbook:***

```
# ansible-playbook -i inventory_file /path/to/file/playbook.yml
```

其它资源

- *有关 kdump 角色变量的详情，请查看 **/usr/share/doc/rhel-system-roles/kdump** 目录中的 **README.md** 或 **README.html** 文件。*
- *请参阅 第 1.3 节 "应用一个角色"。*
- *安装 **rhel-system-roles** 软件包的文档 **/usr/share/ansible/roles/rhel-system-roles.kdump/README.html***

第 10 章 使用 RHEL 系统角色配置存储

要使用 Ansible 管理 LVM 和本地文件系统(FS),您可以使用 **storage** 角色,该角色是 RHEL 8 中可用的 RHEL 系统角色之一。

使用 **storage role** 可让您从 RHEL 7.7 开始在多台机器和 RHEL 的所有版本中自动管理磁盘和逻辑卷中的文件系统。

有关 RHEL 系统角色以及如何应用它们的更多信息，请参阅 RHEL 系统角色简介。

10.1. 存储角色简介

storage 角色可以管理：

- 磁盘上未被分区的文件系统
- 完整的 LVM 卷组，包括其逻辑卷和文件系统

使用 **storage** 角色，您可以执行以下任务：

- 创建文件系统
- 删除文件系统
- 挂载文件系统
- 卸载文件系统
- 创建 LVM 卷组
- 删除 LVM 卷组
- 创建逻辑卷
- 删除逻辑卷
- 创建 RAID 卷
- 删除 RAID 卷
- 创建带有 RAID 的 LVM 池
- 删除带有 RAID 的 LVM 池

10.2. 在存储设备角色中识别存储设备的参数

您的 **storage** 角色配置只会影响您在以下变量中列出的文件系统、卷和池。

storage_volumes

在所有要管理的未分区磁盘中的文件系统列表。
当前不支持的分区。

storage_pools

要管理的池列表。

目前唯一支持的池类型是 LVM。在 LVM 中,池代表卷组(VG)。每个池中都有一个要由角色管理的卷列表。使用 LVM,每个卷对应一个带文件系统的逻辑卷(LV)。

10.3. 使用 RHEL 系统角色在块设备中创建 XFS 文件系统

这部分论述了如何使用 **storage** 角色在多目标机器的块设备中创建 XFS 文件系统。

先决条件

- 存在一个使用 **storage** 角色的 Ansible playbook。
  如需有关如何应用此 playbook 的信息，请参阅 应用角色。

10.3.1. 在块设备中创建 XFS 文件系统的 Ansible playbook 示例

本节提供了一个 Ansible playbook 示例。此 playbook 应用 **storage** 角色，以使用默认参数在块设备中创建 XFS 文件系统。

警告

storage 角色只能在未分区、整个磁盘或者逻辑卷(LV)中创建文件系统。它不能在分区中创建文件系统。

例 10.1. 在 /dev/sdb 上创建 XFS 的 playbook

```
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
  roles:
    - rhel-system-roles.storage
```

- 卷名称（示例中是 **barefs** ）目前是任意卷。**storage** 角色根据 **disks:** 属性中列出的磁盘设备识别卷。
- 您可以省略 **fs_type: xfs** 行，因为 XFS 是 RHEL 8 中的默认文件系统。
- 要在 LV 中创建文件系统，在 **disks:** 属性中提供 LVM 设置，包括保护卷组。详情请参阅 管理逻辑卷的 Ansible playbook 示例。
  不要提供到 LV 设备的路径。

其它资源

- 有关 **storage** 系统角色中使用的参数的详情，请查看 **/usr/share/ansible/roles/rhel-system-roles.storage/README.md** 文件。

10.4. 使用 RHEL 系统角色永久挂载文件系统

这部分论述了如何使用角色永久挂载文件系统 **storage**。

先决条件

- 存在一个使用 **storage** 角色的 Ansible playbook。
  如需有关如何应用此 playbook 的信息，请参阅 应用角色。

10.4.1. 永久挂载文件系统的 Ansible playbook 示例

本节提供了一个 Ansible playbook 示例。此 playbook 应用 **storage** 角色来立即和永久挂载 **XFS** 文件系统。

例 10.2. 在 /dev/sdb 上将文件系统挂载到 /mnt/data 的 playbook

```
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
        mount_point: /mnt/data
  roles:
    - rhel-system-roles.storage
```

- 此 playbook 将文件系统添加到 **/etc/fstab** 文件中，并立即挂载文件系统。
- 如果 **/dev/sdb** 设备或挂载点目录中的文件系统不存在，则 playbook 会创建它们。

其它资源

- 有关 **storage** 系统角色中使用的参数的详情，请查看 **/usr/share/ansible/roles/rhel-system-roles.storage/README.md** 文件。

10.5. 使用 RHEL 系统角色启用在线块丢弃

这部分论述了如何使用角色启用在线块丢弃 **storage**。

先决条件

- 已存在包括 **storage** 角色的 Ansible playbook。

如需有关如何应用此 playbook 的信息，请参阅 应用角色。

10.5.1. 启用在线块丢弃的 Ansible playbook 示例

本节提供了一个 Ansible playbook 示例。此 playbook 应用 **storage** 角色来挂载启用了在线块丢弃的 XFS 文件系统。

例 10.3. 在 /mnt/data/ 上启用在线块丢弃的 playbook

```
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
        mount_point: /mnt/data
        mount_options: discard
  roles:
    - rhel-system-roles.storage
```

其它资源

- 此 playbook 还执行 Example Ansible playbook 中描述的持久性挂载示例的所有操作,以永久挂载文件系统。
- 有关 **storage** 系统角色中使用的参数的详情，请查看 **/usr/share/ansible/roles/rhel-system-roles.storage/README.md** 文件。

10.6. 使用 RHEL 系统角色创建并挂载 EXT3 文件系统

这部分论述了如何在磁盘中使用给定标签创建 ext3 文件系统，并使用角色永久挂载文件系统 **storage**。

先决条件

- 已存在包括 **storage** 角色的 Ansible playbook。

如需有关如何应用此 playbook 的信息，请参阅 应用角色。

10.6.1. 创建和挂载 ext3 文件系统的 Ansible playbook 示例

本节提供了一个 Ansible playbook 示例。此 playbook 应用 **storage** 角色来创建和挂载 Ext3 文件系统。

例 10.4. 在 /dev/sdb 上创建 Ext3 并挂载到 /mnt/data 的 playbook

```
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: ext3
```

```
      fs_label: label-name
      mount_point: /mnt/data
  roles:
    - rhel-system-roles.storage
```

- *playbook 在 **/dev/sdb** 磁盘上创建文件系统。*
- *playbook 会永久将文件系统挂载到 **/mnt/data** 目录中。*
- *文件系统的标签是 **label-name**。*

其它资源

- *有关 **storage** 系统角色中使用的参数的详情，请查看 **/usr/share/ansible/roles/rhel-system-roles.storage/README.md** 文件。*

10.7. 使用 RHEL 系统角色创建并挂载 EXT4 文件系统

*这部分论述了如何在磁盘中使用给定标签创建 ext4 文件系统，并使用角色永久挂载文件系统 **storage**。*

先决条件

- *已存在包括 **storage** 角色的 Ansible playbook。*

如需有关如何应用此 playbook 的信息，请参阅 应用角色。

10.7.1. 创建和挂载 Ext4 文件系统的 Ansible playbook 示例

*本节提供了一个 Ansible playbook 示例。此 playbook 应用 **storage** 角色来创建和挂载 Ext4 文件系统。*

例 10.5. 在 /dev/sdb 上创建 Ext4 并挂载到 /mnt/data 的 playbook

```
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: ext4
        fs_label: label-name
        mount_point: /mnt/data
  roles:
    - rhel-system-roles.storage
```

- *playbook 在 **/dev/sdb** 磁盘上创建文件系统。*
- *playbook 会永久将文件系统挂载到 **/mnt/data** 目录中。*
- *文件系统的标签是 **label-name**。*

其它资源

- 有关 **storage** 系统角色中使用的参数的详情，请查看 **/usr/share/ansible/roles/rhel-system-roles.storage/README.md** 文件。

10.8. 使用 RHEL 系统角色管理 LVM 逻辑卷

本节论述了如何应用 **storage** 角色来执行以下任务：

- 在由多个磁盘组成的卷组中创建 LVM 逻辑卷。
- 在逻辑卷中创建一个带给定标签的 ext4 文件系统。
- 永久挂载 ext4 文件系统。

先决条件

- 包括 **storage** 角色的 Ansible playbook

如需有关如何应用 Ansible playbook 的信息，请参阅 应用角色。

10.8.1. 管理逻辑卷的 Ansible playbook 示例

本节提供了一个 Ansible playbook 示例。这个 playbook 应用 **storage** 角色在卷组中创建 LVM 逻辑卷。

例 10.6. 在 myvg 卷组中创建 mylv 逻辑卷的 playbook

```
- hosts: all
  vars:
    storage_pools:
      - name: myvg
        disks:
          - sda
          - sdb
          - sdc
        volumes:
          - name: mylv
            size: 2G
            fs_type: ext4
            mount_point: /mnt
  roles:
    - rhel-system-roles.storage
```

- **myvg** 卷组由以下磁盘组成：
  - **/dev/sda**
  - **/dev/sdb**
  - **/dev/sdc**
- 如果 **myvg** 卷组已存在，则 playbook 会将逻辑卷添加到卷组中。
- 如果 **myvg** 卷组不存在，则 playbook 会创建它。
- playbook 在 **mylv** 逻辑卷中创建 Ext4 文件系统，并在其中永久挂载文件系统 **/mnt**。

其它资源

- 有关 **storage** 系统角色中使用的参数的详情，请查看 **/usr/share/ansible/roles/rhel-system-roles.storage/README.md** 文件。

10.8.2. 其它资源

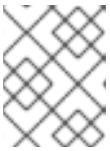
- 有关 **storage** 角色的更多信息，请参阅使用 RHEL 系统角色管理本地存储。

10.9. 使用存储系统角色配置 RAID 卷

使用 **storage** 系统角色，您可以使用 Red Hat Ansible Automation Platform 在 RHEL 上配置 RAID 卷。在本小节中，您将了解如何使用可用参数设置 Ansible playbook，以配置 RAID 卷以满足您的要求。

先决条件

- 您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。



注意

您不必在要部署 **storage** 解决方案的系统中安装 Red Hat Ansible Automation Platform。

- 已安装该系统中要运行 playbook 的 **rhel-system-roles** 软件包。
- 您有一个清单文件详细描述了您要使用 **storage** 系统角色部署 RAID 卷的系统。

流程

1. 使用以下内容 **playbook.yml** 创建新文件：

```
- hosts: all
  vars:
    storage_safe_mode: false
    storage_volumes:
      - name: data
        type: raid
        disks: [sdd, sde, sdf, sdg]
        raid_level: raid0
        raid_chunk_size: 32 KiB
        mount_point: /mnt/data
        state: present
  roles:
    - name: rhel-system-roles.storage
```

警告

设备名称在某些情况下可能会改变，例如：当您在系统中添加新磁盘时。因此，为了避免数据丢失,我们不建议在 playbook 中使用特定的磁盘名称。

2. *可选。验证 playbook 语法。*

```
# ansible-playbook --syntax-check playbook.yml
```

3. *在清单文件上运行 playbook:*

```
# ansible-playbook -i inventory.file /path/to/file/playbook.yml
```

其它资源

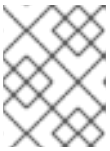
- *有关 RAID 的详情，请参阅管理 RAID。*
- *有关存储系统角色中使用的参数的详情，请查看 /usr/share/ansible/roles/rhel-system-roles.storage/README.md 文件。*

10.10. 使用存储系统角色使用 RAID 配置 LVM 池

使用 storage 系统角色,您可以使用 Red Hat Ansible Automation Platform 在 RHEL 上配置带有 RAID 的 LVM 池。在本小节中，您将了解如何使用可用参数设置 Ansible playbook，以配置使用 RAID 的 LVM 池。

先决条件

- *您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。*



注意

您不必在要部署 storage 解决方案的系统中安装 Red Hat Ansible Automation Platform。

- *已安装该系统中要运行 playbook 的 rhel-system-roles 软件包。*
- *您有一个清单文件详细描述了您要使用系统角色配置带 RAID 的 LVM 池的 storage 系统。*

流程

1. *使用以下内容 playbook.yml 创建新文件：*

```
- hosts: all
  vars:
    storage_safe_mode: false
    storage_pools:
      - name: my_pool
```

```
    type: lvm
    disks: [sdh, sdi]
    raid_level: raid1
    volumes:
      - name: my_pool
        size: "1 GiB"
        mount_point: "/mnt/app/shared"
        fs_type: xfs
        state: present
roles:
  - name: rhel-system-roles.storage
```

注意

*要使用具有 RAID 的 LVM 池，您必须使用 **raid_level** 参数指定 RAID 类型。*

2. *可选。验证 **playbook** 语法。*

```
# ansible-playbook --syntax-check playbook.yml
```

3. *在清单文件上运行 **playbook:***

```
# ansible-playbook -i inventory.file /path/to/file/playbook.yml
```

其它资源

- *有关 RAID 的详情，请参阅管理 RAID。*
- *有关存储系统角色中使用的参数的详情，请查看 **/usr/share/ansible/roles/rhel-system-roles.storage/README.md** 文件。*

10.11. 使用 RHEL 系统角色管理 LUKS 加密的卷

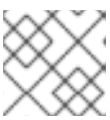
*使用 **storage** 系统角色,您可以使用 Red Hat Ansible Automation Platform 管理 RHEL 中的 Linux Unified Key Setup-on-disk-format(LUKS)加密的卷。*

10.11.1. 使用存储角色创建 LUKS 加密卷

*您可以通过运行 Ansible playbook，使用 **storage** 角色创建并配置使用 LUKS 加密的卷。*

先决条件

- *您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。*



注意

您不必在要创建卷的系统中安装 Red Hat Ansible Automation Platform。

- *您已在 Ansible 控制器上安装了 **rhel-system-roles** 软件包。*
- *您有一个清单文件详细描述了您要使用存储系统角色部署 LUKS 加密卷的系统。*

流程

1. *使用以下内容* ***playbook.yml*** *创建新文件：*

```
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
         - sdb
        fs_type: xfs
        fs_label: label-name
        mount_point: /mnt/data
        encryption: true
        encryption_password: your-password
  roles:
   - rhel-system-roles.storage
```

2. *可选。验证* ***playbook*** *语法：*

```
# ansible-playbook --syntax-check playbook.yml
```

3. *在清单文件上运行* ***playbook:***

```
# ansible-playbook -i inventory.file /path/to/file/playbook.yml
```

其它资源

- *如需了解更多与 LUKS 相关的信息，请参阅 17。使用 LUKS 加密块设备。*
- *有关* ***storage*** *系统角色中使用的参数的详情，请查看* ***/usr/share/ansible/roles/rhel-system-roles.storage/README.md*** *文件。*

其它资源

如需更多信息，请安装 ***rhel-system-roles*** *软件包以及查看* ***/usr/share/ansible/roles/rhel-system-roles.storage/*** *目录和* ***/usr/share/doc/rhel-system-roles/storage/*** *目录。*

第 11 章 使用 RHEL 系统角色配置时间同步

使用 **timesync** RHEL 系统角色，您可以使用 Red Hat Ansible Automation Platform 在 RHEL 中的多个目标机器上管理时间同步。

11.1. TIMESYNC 系统角色

您可以使用 **timesync** RHEL 系统角色在多个目标机器上管理时间同步。

timesync 角色安装并配置 NTP 或 PTP 实现，以作为 NTP 客户端或 PTP 副本操作，以便将系统时钟与 PTP 域中的 NTP 服务器或 Powermasters 同步。

请注意，使用 **timesync** 角色还可允许迁移到 chrony，因为您可以在从 RHEL 6 开始的所有 Red Hat Enterprise Linux 版本上使用相同的 playbook，无论系统是使用 **ntp** 还是 **chrony** 实施 NTP 协议。

11.2. 为单一服务器池应用 TIMESYNC 系统角色

以下示例演示了如何在只有一个服务器池的情况下应用 **timesync** 角色。

警告

timesync 角色替换了受管主机上给定或检测到的供应商服务的配置。之前的设置即使没有在角色变量中指定，也会丢失。如果没有定义 **timesync_ntp_provider** 变量，唯一保留的设置就是供应商选择。

先决条件

- 您已在要运行 playbook 的系统中安装了 Red Hat Ansible Engine。

注意

您不必在要部署 **timesync** 解决方案的系统中安装 Red Hat Ansible Automation Platform。

- 已安装该系统中要运行 playbook 的 **rhel-system-roles** 软件包。
- 您有一个清单文件，其中包含您要在其上部署 **timesync** 系统角色的系统。

流程

1. 使用以下内容 **playbook.yml** 创建新文件：

```
---
- hosts: timesync-test
  vars:
    timesync_ntp_servers:
      - hostname: 2.rhel.pool.ntp.org
        pool: yes
```

```
      iburst: yes
  roles:
    - rhel-system-roles.timesync
```

2. *可选：验证* ***playbook*** *语法。*

```
# ansible-playbook --syntax-check playbook.yml
```

3. *在清单文件上运行* ***playbook:***

```
# ansible-playbook -i inventory_file /path/to/file/playbook.yml
```

11.3. TIMESYNC 系统角色变量

您可以将以下变量传递给 ***timesync*** *角色：*

- ***timesync_ntp_servers:***

角色变量设置	描述
hostname: host.example.com	服务器的主机名或地址
minpoll: *number*	最小轮询间隔。默认：6
maxpoll: *number*	最大轮询间隔。默认：10
iburst: yes	标志启用快速初始同步。默认： no
pool: yes	指示每个主机名解析地址都是一个单独的 NTP 服务器的标志。默认： no

其它资源

- *有关* ***timesync*** *角色变量的详细参考，安装* ***rhel-system-roles*** *软件包，并参阅* ***/usr/share/doc/rhel-system-roles/timesync*** *目录中的* ***README.md*** *或者* ***README.html*** *文件。*

第 12 章 使用 RHEL 系统角色监控性能

12.1. 指标系统角色简介

RHEL 系统角色是 Ansible 角色和模块的集合，可为远程管理多个 RHEL 系统提供一致的配置界面。指标系统角色为本地系统配置性能分析服务，可选包括由本地系统监控的远程系统列表。借助 metrics 系统角色，您可以使用 **pcp** 来监控系统性能，而无需单独配置 **pcp**，因为 playbook 处理 **pcp** 设置和部署。

表 12.1. 指标系统角色变量

角色变量	描述	用法示例
metrics_monitored_hosts	要被目标主机分析的远程主机列表。这些主机将在目标主机上记录指标，以确保每个主机的 **/var/log** 中有足够的磁盘空间。	**metrics_monitored_hosts: ["*webserver.example.com*", "*database.example.com*"]**
metrics_retention_days	在删除前配置性能数据保留的天数。	**metrics_retention_days: 14**
metrics_graph_service	布尔值标志，可让主机与服务一起通过 **pcp** 和通过和视觉化性能数据进行设置 **grafana**。默认设置为 false。	**metrics_graph_service: false**
metrics_query_service	布尔值标志,使用时间序列查询服务来通过 **pcp** 查询记录的指标数据来设置主机 **redis**。默认设置为 false。	**metrics_query_service: false**
metrics_provider	指定要用于提供指标的指标收集器。目前, **pcp** 是唯一支持的指标提供程序。	**metrics_provider: "pcp"**

其它资源

- 有关使用的参数 **metrics_connections** 以及有关 metrics 系统角色的额外信息，请参阅该 **/usr/share/ansible/roles/rhel-system-roles.metrics/README.md** 文件。

12.2. 使用指标系统角色以可视化方式监控本地系统

此流程描述了如何在同时置备数据视觉化时使用 metrics RHEL 系统角色监控您的本地系统 **grafana**。

先决条件

- 您已在要监控的机器上安装了 Red Hat Ansible Engine。
- 您需要在要监控的机器中安装 **rhel-system-roles** 软件包。

流程

1. 通过 **localhost** 将以下内容添加到清单中，在 **/etc/ansible/hosts** Ansible 清单中配置：

```
localhost ansible_connection=local
```

2. 使用以下内容创建一个 **Ansible playbook:**

```
---
- hosts: localhost
  vars:
    metrics_graph_service: yes
  roles:
    - rhel-system-roles.metrics
```

3. 运行 **Ansible playbook:**

```
# ansible-playbook name_of_your_playbook.yml
```

注意

因为 **metrics_graph_service** 布尔值被设置为 **value="yes", grafana** 它会被自动安装并置备并将 **pcp** 添加为数据源。

4. 要查看您机器上收集的指标的视觉化，请访问 **Grafana grafana Web UI** 中的 **Web** 界面。

12.3. 使用 METRICS 系统角色设置监控其自身的独立系统

此流程描述了如何使用 **metrics** 系统角色设置一组机器来监控其自身。

先决条件

- 您已在要用来运行 **playbook** 的机器上安装了 **Red Hat Ansible Engine**。
- 已安装要用来运行 **playbook** 的机器上的 **rhel-system-roles** 软件包。

流程

1. 将您要通过 **playbook** 监控的机器的名称或 **IP** 添加到 **/etc/ansible/hosts** Ansible 清单文件中，其名称在由括号括起来的标识组名称下：

```
[remotes]
webserver.example.com
database.example.com
```

2. 使用以下内容创建一个 **Ansible playbook:**

```
---
- hosts: remotes
  vars:
    metrics_retention_days: 0
  roles:
    - rhel-system-roles.metrics
```

3. 运行 Ansible playbook:

```
# ansible-playbook name_of_your_playbook.yml
```

12.4. 使用 METRICS 系统角色通过本地机器集中监控机器的数量

此流程描述了如何使用 metrics 系统角色设置本地机器来集中监控机器数量，同时通过 **grafana** 进行数据视觉化，通过 **redis** 查询数据。

先决条件

- 您已在要用来运行 playbook 的机器上安装了 Red Hat Ansible Engine。
- 已安装要用来运行 playbook 的机器上的 **rhel-system-roles** 软件包。

流程

1. 使用以下内容创建一个 Ansible playbook:

```
---
- hosts: localhost
  vars:
    metrics_graph_service: yes
    metrics_query_service: yes
    metrics_retention_days: 10
    metrics_monitored_hosts: ["database.example.com", "webserver.example.com"]
  roles:
    - rhel-system-roles.metrics
```

2. 运行 Ansible playbook:

```
# ansible-playbook name_of_your_playbook.yml
```

注意

因为 **metrics_graph_service** 和**metrics_query_service** 被设置为 **value="yes"**，**grafana** 会被自动安装，被置备为带有 **pcp** 作为 **pcp** 数据源索引到 **redis**，从而可以使用 **pcp** 查询语言对数据进行复杂的查询。

3. 要查看机器集中收集的指标的图形显示并查询数据,请访问访问 Grafana **grafana** Web UI 中的 Web 界面。

第 13 章 配置系统以使用 TLOG RHEL 系统角色记录会话记录

使用 **tlog** RHEL 系统角色,您可以使用 Red Hat Ansible Automation Platform 为 RHEL 上的终端会话记录配置系统。

13.1. TLOG 系统角色

您可以使用 RHEL 系统角色为 RHEL 中的终端会话记录配置 **tlog** RHEL 系统。**tlog** 软件包及其关联的 web 控制台会话播放器为您提供记录和回溯用户终端会话的功能。

您可以将录制配置为通过 **SSSD** 服务在每个用户或用户组中进行。所有终端输入和输出都会捕获并保存在系统日志中的文本格式。

其它资源

- 有关 RHEL 中会话记录的详情，请参阅 记录会话

13.2. TLOG 系统角色的组件和参数

Session Recording 的解决方案由以下组件组成：

- **tlog** 工具
- 系统安全性服务守护进程（**SSSD**）
- 可选：**Web** 控制台界面

用于 **tlog** RHEL 系统角色的参数有：

角色变量	描述
tlog_use_sssd (default: yes)	使用 SSSD 配置会话记录，这是管理记录的用户或组的首选方法
tlog_scope_sssd (default: none)	配置 SSSD 记录范围 - all / some / none
tlog_users_sssd (default: [])	要记录的用户的 YAML 列表
tlog_groups_sssd (default: [])	要记录的组的 YAML 列表

- 有关使用的参数 **tlog** 以及 tlog 系统角色的附加信息，请参考 **/usr/share/ansible/roles/rhel-system-roles.tlog/README.md** 文件。

13.3. 部署 TLOG RHEL 系统角色

按照以下步骤准备并应用 Ansible playbook 以配置 RHEL 系统将数据记录到 **systemd** 日志中。

先决条件

- 您已设置了从控制节点访问系统角色的目标系统（**tlog** 系统角色在其中配置的系统）的 SSH 密钥。

- 您有一个控制节点，这是 Ansible Engine 配置其他系统的系统。
- 您已在控制节点上安装了 Red Hat Ansible Engine，您要从该节点上运行 playbook。
- 已在要从其中运行 playbook 的控制节点上安装了 **rhel-system-roles** 软件包。
- 您至少有一个要配置 **tlog** 系统角色的系统。您不必在要部署 **tlog** 解决方案的系统中安装 Red Hat Ansible Automation Platform。

流程

1. 使用以下内容 **playbook.yml** 创建新文件：

```
---
- name: Deploy session recording
  hosts: all
  vars:
    tlog_scope_sssd: some
    tlog_users_sssd:
      - recordeduser

  roles:
    - rhel-system-roles.tlog
```

其中，

- **tlog_scope_sssd**:
  - **some** 指定您只记录某些用户和组群，不是 **all** 或 **none**。
- **tlog_users_sssd**:
  - **recordeduser** 指定要记录会话的用户。请注意，这不会为您添加用户。您必须自行设置该用户。

2. 另外，还可以验证 **playbook** 语法。

```
# ansible-playbook --syntax-check playbook.yml
```

3. 在清单文件上运行 **playbook**:

```
# ansible-playbook -i IP_Address /path/to/file/playbook.yml -v
```

playbook 会在您指定的系统中安装 **tlog** 角色。它还会创建一个 SSSD 配置丢弃文件，可供您定义的用户和组使用。SSSD 解析并读取这些用户和组以 shell 用户的身份覆盖 **tlog** 会话。另外，如果 **cockpit** 软件包安装在系统中，playbook 也会安装 **cockpit-session-recording** 软件包，它是一个 **Cockpit** 模块，供您在 web 控制台界面中查看和播放记录。

验证步骤

要验证 SSSD 配置文件是否在系统中创建了，请执行以下步骤：

1. 进入创建 SSSD 配置丢弃文件的文件夹：

```
# cd /etc/sssd/conf.d
```

2. 检查文件内容：

```
# cat /etc/sssd/conf.d/sssd-session-recording.conf
```

您可以看到该文件包含您在 **playbook** 中设置的参数。

13.4. 使用在 CLI 中部署的 TLOG 系统角色记录会话

当您在指定的系统中部署了 **tlog** 系统角色，就可以使用命令行界面（CLI）记录用户终端会话。

先决条件

- 您已在目标系统中部署了 **tlog** 系统角色。
- SSSD 配置丢弃文件在文件 **/etc/sssd/conf.d** 中创建。

流程

1. 创建一个用户并为这个用户分配密码：

```
# useradd recordeduser
# passwd recordeduser
```

2. 以您刚刚创建的用户身份登录到该系统：

```
# ssh recordeduser@localhost
```

3. 当系统提示您输入 **yes** 或 **no** 进行身份验证时请输入 **"yes"**。

4. 插入 **recorduser** 的密码。
系统提示一条信息通知您的会话被记录。

```
ATTENTION! Your session is being recorded!
```

5. 记录完会话后，请键入：

```
# exit
```

系统从用户注销并关闭与本地主机的连接。

用户会话会被记录，并被保存，您可以使用 **journal** 进行播放。

验证步骤

要在日志中查看您记录的会话，请执行以下步骤：

1. 运行以下命令：

```
# journalctl -o verbose -r
```

2. 搜索 **tlog-rec** 记录日志条目中的 **MESSAGE** 字段。

13.5. 使用 CLI 监视记录的会话

您可以使用命令行界面（CLI）从日志中执行用户会话记录。

先决条件

- 您已经记录了一个用户会话。请查看 第 13.4 节 “使用在 CLI 中部署的 tlog 系统角色记录会话”

流程

1. 在 CLI 终端中，播放用户会话记录：

```
# journalctl -o verbose -r
```

2. 搜索 **tlog** 记录：

```
$ /tlog-rec
```

   您可以查看详情，例如：

   - 用户会话记录的用户名
   - **out_txt** 字段是记录的会话的原始输出编码
   - 标识符号 **TLOG_REC=ID_number**

3. 复制标识符号 **TLOG_REC=ID_number**。
4. 使用标识符号 **TLOG_REC=ID_number** 回放记录。

```
# tlog-play -r journal -M TLOG_REC=ID_number
```

您可以看到记录的用户会话被回放。