```
selinux_restore_dirs:
  - /srv/www
```

`selinux_ports` 变量取应当具有特定 SELinux 类型的端口的列表作为值。

```
selinux_ports:
  - ports: '82'
    setype: 'http_port_t'
    proto: 'tcp'
    state: 'present'
```

此角色还有其他的变量和选项。有关更多信息，请参见其 **`README.md`** 文件。

参考文献

红帽企业 Linux (RHEL) 系统角色
https://access.redhat.com/articles/3050101

Linux 系统角色
https://linux-system-roles.github.io/

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

指导练习

利用系统角色重用内容

在本练习中，您要将一个红帽企业 Linux 系统角色和普通任务结合使用，以配置受管主机上的时间同步和时区。

成果

您应能够：

- 安装红帽企业 Linux 系统角色。
- 查找并使用 RHEL 系统角色文档。
- 在 playbook 中使用 `rhel-system-roles.timesync` 角色来配置远程主机上的时间同步。

场景概述

您的组织维护着两个数据中心：一个在美国（芝加哥），另一个在芬兰（赫尔辛基）。为了协助数据中心之间数据库服务器的日志分析，需要确保利用时间同步协议同步每一主机上的系统时钟。为了协助数据中心之间进行时刻活动分析，需要确保每一数据库服务器的时区设置为与主机的数据中心位置对应。

时间同步具有如下要求：

- 使用位于 `classroom.example.com` 的 NTP 服务器。启用 **`iburst`** 选项，以加快初始时间同步。
- 将 chrony 软件包用于时间同步。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 **`lab role-system start`** 命令。这会创建工作目录 **`/home/student/role-system`**，并为它填充 Ansible 配置文件和主机清单。

```
[student@workstation ~]$ lab role-system start
```

1. 更改到 **`/home/student/role-system`** 工作目录。

```
[student@workstation ~]$ cd ~/role-system
[student@workstation role-system]$
```

2. 在控制节点 `workstation.lab.example.com` 上安装红帽企业 Linux 系统角色。验证控制节点上的角色安装位置。

2.1. 使用 **`ansible-galaxy`** 命令，验证 playbook 项目中最初没有可供使用的角色。

海量视频题库 myitpub.com QQ:5565462 www.52myit.com

```
[student@workstation role-system]$ ansible-galaxy list
# /home/student/role-system/roles
# /usr/share/ansible/roles
# /etc/ansible/roles
[student@workstation role-system]$
```

ansible-galaxy 命令将在三个目录中搜索角色，如 **ansible.cfg** 文件中的 roles_path 条目所示：

- **./roles**
- **/usr/share/ansible/roles**
- **/etc/ansible/roles**

以上输出表明这些目录中都没有角色。

2.2. 安装 rhel-system-roles 软件包。

```
[student@workstation role-system]$ sudo yum install rhel-system-roles
```

在提示安装软件包时，输入 **y**。

2.3. 使用 **ansible-galaxy** 命令验证系统角色现已可用。

```
[student@workstation role-system]$ ansible-galaxy list
# /home/student/role-system/roles
# /usr/share/ansible/roles
- linux-system-roles.kdump, (unknown version)
- linux-system-roles.network, (unknown version)
- linux-system-roles.postfix, (unknown version)
- linux-system-roles.selinux, (unknown version)
- linux-system-roles.timesync, (unknown version)
- rhel-system-roles.kdump, (unknown version)
- rhel-system-roles.network, (unknown version)
- rhel-system-roles.postfix, (unknown version)
- rhel-system-roles.selinux, (unknown version)
- rhel-system-roles.timesync, (unknown version)
# /etc/ansible/roles
```

这些角色位于 **/usr/share/ansible/roles** 目录中。以 **linux-system-roles** 开头的任何角色实际上都是对应的 **rhel-system-roles** 角色的符号链接。

3. 创建一个 playbook **configure_time.yml**，其中包含一个以 **database_servers** 主机组为目标的 play。将 rhel-system-roles.timesync 角色包含到 play 的 **roles** 部分。

```
---
- name: Time Synchronization
  hosts: database_servers

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

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

4. 角色文档中包含各个角色变量的描述，包括变量的默认值。确定要覆盖的角色变量，以满足时间同步的要求。

将角色变量值放在一个名为 **timesync.yml** 的文件中。由于这些变量值应用到清单中的所有主机，因此请将 **timesync.yml** 文件放到 **group_vars/all** 子目录中。

4.1. 检查 **README.md** 文件的 Role Variables 部分中的 rhel-system-roles.timesync 角色。

```
[student@workstation role-system]$ cat \
> /usr/share/doc/rhel-system-roles/timesync/README.md
...output omitted...
Role Variables
--------------

...output omitted...
# List of NTP servers
timesync_ntp_servers:
  - hostname: foo.example.com    # Hostname or address of the server
    minpoll: 4                   # Minimum polling interval (default 6)
    maxpoll: 8                   # Maximum polling interval (default 10)
    iburst: yes                  # Flag enabling fast initial synchronization
                                 # (default no)
    pool: no                     # Flag indicating that each resolved address
                                 # of the hostname is a separate NTP server
                                 # (default no)
...output omitted...
# Name of the package which should be installed and configured for NTP.
# Possible values are "chrony" and "ntp". If not defined, the currently active
# or enabled service will be configured. If no service is active or enabled, a
# package specific to the system and its version will be selected.
timesync_ntp_provider: chrony
...output omitted...
```

4.2. 创建 **group_vars/all** 子目录。

```
[student@workstation role-system]$ mkdir -pv group_vars/all
mkdir: created directory 'group_vars'
mkdir: created directory 'group_vars/all'
```

4.3. 使用文本编辑器创建新文件 **group_vars/all/timesync.yml**。添加变量定义，以满足时间同步要求。该文件现在包含：

```
---
#rhel-system-roles.timesync variables for all hosts

timesync_ntp_provider: chrony

timesync_ntp_servers:
  - hostname: classroom.example.com
    iburst: yes
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

5. 在 **configure_time.yml** 中添加一个任务，为各个主机设置时区。确保该任务使用 timezone 模块，并在 rhel-system-roles.timesync 角色之后执行。

由于主机不属于同一个时区，因此使用变量 (host_timezone) 作为时区名称。

5.1. 检查 timezone 模块文档的 Examples 部分。

```
[student@workstation role-system]$ ansible-doc timezone | grep -A 4 "EXAMPLES"
EXAMPLES:
- name: set timezone to Asia/Tokyo
  timezone:
    name: Asia/Tokyo
```

5.2. 在 **configure_time.yml** playbook 中，添加一个任务到 play 的 **post_tasks** 部分。按照文档中的示例为任务建模，但是将 host_timezone 变量用作时区名称。

ansible-doc timezone 中的文档建议在模块更改时区后重新启动 Cron 服务，以确保 Cron 作业在正确的时间运行。在任务中添加 **notify** 关键字，以及关联值 **restart crond**。Play 的 **post_tasks** 部分应如以下所示：

```
  post_tasks:
    - name: Set timezone
      timezone:
        name: "{{ host_timezone }}"
      notify: restart crond
```

5.3. 将 **restart crond** 处理程序添加到 **Time Synchronization** play 中。完整的 playbook 现在包含：

```
---
- name: Time Synchronization
  hosts: database_servers

  roles:
    - rhel-system-roles.timesync

  post_tasks:
    - name: Set timezone
      timezone:
        name: "{{ host_timezone }}"
      notify: restart crond

  handlers:
    - name: restart crond
      service:
        name: crond
        state: restarted
```

6. 对于每个数据中心，创建一个名为 **timezone.yml** 的文件，其中包含 host_timezone 变量的相应值。使用 **timedatectl list-timezones** 命令查找各个数据中心的有效时区字符串。

6.1. 为 na_datacenter 和 europe_datacenter 主机组创建 **group_vars** 子目录。

海量视频题库 my.pub.com QQ:5565462 www.52myit.com

```
[student@workstation role-system]$ mkdir -pv \
> group_vars/{na_datacenter,europe_datacenter}
mkdir: created directory 'group_vars/na_datacenter'
mkdir: created directory 'group_vars/europe_datacenter'
```

6.2. 使用 **timedatectl list-timezones** 命令来确定美国和欧洲数据中心的时区：

```
[student@workstation role-system]$ timedatectl list-timezones | grep Chicago
America/Chicago
[student@workstation role-system]$ timedatectl list-timezones | grep Helsinki
Europe/Helsinki
```

6.3. 为两个数据中心创建 **timezone.yml**：

```
[student@workstation role-system]$ echo "host_timezone: America/Chicago" > \
> group_vars/na_datacenter/timezone.yml
[student@workstation role-system]$ echo "host_timezone: Europe/Helsinki" > \
> group_vars/europe_datacenter/timezone.yml
```

7. 运行 playbook。

```
[student@workstation role-system]$ ansible-playbook configure_time.yml

PLAY [Time Synchronization] ***************************************************

TASK [Gathering Facts] ********************************************************
ok: [serverb.lab.example.com]
ok: [servera.lab.example.com]

TASK [rhel-system-roles.timesync : Check if only NTP is needed] **************
ok: [servera.lab.example.com]
ok: [serverb.lab.example.com]

...output omitted...

TASK [rhel-system-roles.timesync : Enable timemaster] ************************
skipping: [servera.lab.example.com]
skipping: [serverb.lab.example.com]

RUNNING HANDLER [rhel-system-roles.timesync : restart chronyd] ***************
changed: [servera.lab.example.com]
changed: [serverb.lab.example.com]

TASK [Set timezone] ***********************************************************
changed: [serverb.lab.example.com]
changed: [servera.lab.example.com]

RUNNING HANDLER [restart crond] ***********************************************
changed: [serverb.lab.example.com]
changed: [servera.lab.example.com]
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
PLAY RECAP *********************************************************************
servera.lab.example.com    : ok=17   changed=6    unreachable=0    failed=0
skipped=20   rescued=0    ignored=6
serverb.lab.example.com    : ok=17   changed=6    unreachable=0    failed=0
skipped=20   rescued=0    ignored=6
```

8. 验证每台服务器的时区设置。使用 Ansible 临时命令，查看所有数据库服务器上 **date** 命令的输出。

> **注意**
> 列出的实际时区将根据一年中的时间以及夏令时是否有效而有所不同。

```
[student@workstation role-system]$ ansible database_servers -m shell -a date
serverb.lab.example.com | CHANGED | rc=0 >>
Wed Apr  3 06:27:01 EEST 2019

servera.lab.example.com | CHANGED | rc=0 >>
Tue Apr  2 22:27:01 CDT 2019
```

每台服务器具有基于其地理位置的时区设置。

完成

运行 **lab role-system finish**命令，以清理受管主机。

```
[student@workstation ~]$ lab role-system finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

创建角色

培训目标

学完本节后，您应能够在 playbook 的项目目录中创建角色，并将其作为 playbook 中某一 play 的一部分运行。

角色创建流程

在 Ansible 中创建角色不需要特别的开发工具。创建和使用角色包含三个步骤：

1. 创建角色目录结构。
2. 定义角色内容。
3. 在 playbook 中使用角色。

创建角色目录结构

默认情况下，Ansible 在 Ansible Playbook 所在目录的 **roles** 子目录中查找角色。这样，您可以利用 playbook 和其他支持文件存储角色。

如果 Ansible 无法在该位置找到角色，它会按照顺序在 Ansible 配置设置 **roles_path** 所指定的目录中查找。此变量包含要搜索的目录的冒号分隔列表。此变量的默认值为：

```
~/.ansible/roles:/usr/share/ansible/roles:/etc/ansible/roles
```

这允许您将角色安装到由多个项目共享的系统上。例如，您可能将自己的角色安装在您的主目录下的 **~/.ansible/roles** 子目录中，而系统可能将所有用户的角色安装在 **/usr/share/ansible/roles** 目录中。

每个角色具有自己的目录，采用标准化的目录结构。例如，以下目录结构包含了定义 **motd** 角色的文件。

```
[user@host ~]$ tree roles/
roles/
└── motd
    ├── defaults
    │   └── main.yml
    ├── files
    ├── handlers
    ├── meta
    │   └── main.yml
    ├── README.md
    ├── tasks
    │   └── main.yml
    └── templates
        └── motd.j2
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

README.md 提供人类可读的基本角色描述、有关如何使用该角色的文档和示例，以及其发挥作用所需要满足的任何非 Ansible 要求。**meta** 子目录包含一个 **main.yml** 文件，该文件指定有关模块的作者、许可证、兼容性和依赖项的信息。**files** 子目录包含固定内容的文件，而 **templates** 子目录则包含使用时可由角色部署的模板。其他子目录中可以包含 **main.yml** 文件，它们定义默认的变量值、处理程序、任务、角色元数据或变量，具体取决于所处的子目录。

如果某一子目录存在但为空，如本例中的 **handlers**，它将被忽略。如果某一角色不使用功能，则其子目录可以完全省略。例如，本例中的 **vars** 子目录已被省略。

创建角色框架

您可以使用标准 Linux 命令创建新角色所需的所有子目录和文件。此外，也可以通过命令行实用程序来自动执行新角色创建过程。

ansible-galaxy 命令行工具（本课程稍后会详细介绍）可用于管理 Ansible 角色，包括新角色的创建。您可以运行 **ansible-galaxy init** 来创建新角色的目录结构。指定角色的名称作为命令的参数，该命令在当前工作目录中为新角色创建子目录。

```
[user@host playbook-project]$ cd roles
[user@host roles]$ ansible-galaxy init my_new_role
- my_new_role was created successfully
[user@host roles]$ ls my_new_role/
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
```

定义角色内容

创建目录结构后，您必须编写角色的内容。**ROLENAME/tasks/main.yml** 任务文件是一个不错的起点，它是由角色运行的主要任务列表。

下列 **tasks/main.yml** 文件管理受管主机上的 **/etc/motd** 文件。它使用 **template** 模块将名为 **motd.j2** 的模板部署到受管主机上。因为 **template** 模块是在角色任务而非 playbook 任务内配置的，所以从角色的 **templates** 子目录中检索 **motd.j2** 模板。

```
[user@host ~]$ cat roles/motd/tasks/main.yml
---
# tasks file for motd

- name: deliver motd file
  template:
    src: motd.j2
    dest: /etc/motd
    owner: root
    group: root
    mode: 0444
```

下列命令显示 **motd** 角色的 **motd.j2** 模板的内容。它引用了 Ansible 事实和 **system_owner** 变量。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
[user@host ~]$ cat roles/motd/templates/motd.j2
This is the system {{ ansible_facts['hostname'] }}.

Today's date is: {{ ansible_facts['date_time']['date'] }}.

Only use this system with permission.
You can ask {{ system_owner }} for access.
```

该角色为 **system_owner** 变量定义一个默认值。角色目录结构中的 **defaults/main.yml** 文件就是设置这个值的位置。

下列 **defaults/main.yml** 文件将 **system_owner** 变量设置为 **user@host.example.com**。此电子邮件地址将写入到该角色所应用的受管主机上的 **/etc/motd** 文件中。

```
[user@host ~]$ cat roles/motd/defaults/main.yml
---
system_owner: user@host.example.com
```

角色内容开发的推荐做法

角色允许以模块化方式编写 playbook。为了最大限度地提高新开发角色的效率，请考虑在角色开发中采用以下推荐做法：

- 在角色自己的版本控制存储库中维护每个角色。Ansible 很适合使用基于 **git** 的存储库。
- 角色存储库中不应存储敏感信息，如密码或 SSH 密钥。敏感值应以变量的形式进行参数化，其默认值应不敏感。使用角色的 Playbook 负责通过 Ansible Vault 变量文件、环境变量或其他 **ansible-playbook** 选项定义敏感变量。
- 使用 **ansible-galaxy init** 启动您的角色，然后删除您不需要的任何目录和文件。
- 创建并维护 **README.md** 和 **meta/main.yml** 文件，以记录您的角色的用途、作者和用法。
- 让您的角色侧重于特定的用途或功能。您可以编写多个角色，而不是让一个角色承担许多任务。
- 经常重用和重构角色。避免为边缘配置创建新的角色。如果现有角色能够完成大部分的所需配置，请重构现有角色以集成新的配置方案。使用集成和回归测试技术来确保角色提供所需的新功能，并且不对现有的 playbook 造成问题。

定义角色依赖项

角色依赖项使得角色可以将其他角色作为依赖项包含在内。例如，一个定义文档服务器的角色可能依赖于另一个安装和配置 Web 服务器的角色。依赖关系在角色目录层次结构中的 **meta/main.yml** 文件内定义。

以下是一个示例 **meta/main.yml** 文件。

```
---
dependencies:
  - role: apache
    port: 8080
  - role: postgres
    dbname: serverlist
    admin_user: felix
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

默认情况下，角色仅作为依赖项添加到 playbook 中一次。若有其他角色也将它作为依赖项列出，它不会再次运行。此行为可以被覆盖，将 **meta/main.yml** 文件中的 **allow_duplicates** 变量设置为 **yes** 即可。

> **重要**
> 限制角色对其他角色的依赖。依赖项使得维护角色变得更加困难，尤其是当它具有许多复杂的依赖项时。

在 PLAYBOOK 中使用角色

要访问角色，可在 play 的 **roles:** 部分引用它。下列 playbook 引用了 **motd** 角色。由于没有指定变量，因此将使用默认变量值应用该角色。

```
[user@host ~]$ cat use-motd-role.yml
---
- name: use motd role playbook
  hosts: remote.example.com
  remote_user: devops
  become: true
  roles:
    - motd
```

执行该 playbook 时，因为角色而执行的任务可以通过角色名称前缀来加以识别。以下示例输出通过任务名称中的 **motd :** 前缀进行了演示：

```
[user@host ~]$ ansible-playbook -i inventory use-motd-role.yml

PLAY [use motd role playbook] ***************************************************

TASK [setup] ********************************************************************
ok: [remote.example.com]

TASK [motd : deliver motd file] *************************************************
changed: [remote.example.com]

PLAY RECAP **********************************************************************
remote.example.com         : ok=2    changed=1    unreachable=0    failed=0
```

上述情形假定 **motd** 角色位于 **roles** 目录中。在本课程的后续部分中，您将了解如何使用位于远程版本控制存储库中的角色。

通过变量更改角色的行为

编写良好的角色利用默认变量来改变角色行为，使之与相关的配置场景相符。这有助于让角色变得更为通用，可在各种不同的上下文中重复利用。

如果通过以下方式定义了相同的变量，则角色的 **defaults** 目录中定义的变量的值将被覆盖：

- 在清单文件中定义，作为主机变量或组变量。
- 在 playbook 项目的 **group_vars** 或 **host_vars** 目录下的 YAML 文件中定义
- 作为变量嵌套在 play 的 **vars** 关键字中定义

海量视频题库 myitpub.com QQ:556462
www.52myit.com

· 在 play 的 **roles** 关键字中包含该角色时作为变量定义

下例演示了如何将 **motd** 角色与 **system_owner** 角色变量的不同值搭配使用。角色应用到受管主机时，指定的值 **someone@host.example.com** 将取代变量引用。

```
[user@host ~]$ cat use-motd-role.yml
---
- name: use motd role playbook
  hosts: remote.example.com
  remote_user: devops
  become: true
  vars:
    system_owner: someone@host.example.com
  roles:
    - role: motd
```

以这种方式定义时，**system_owner** 变量将替换同一名称的默认变量的值。嵌套在 **vars** 关键字内的任何变量定义不会替换在角色的 **vars** 目录中定义的同一变量的值。

下例也演示了如何将 **motd** 角色与 **system_owner** 角色变量的不同值搭配使用。指定的值 **someone@host.example.com** 将替换变量引用，不论是在角色的 **vars** 还是 **defaults** 目录中定义。

```
[user@host ~]$ cat use-motd-role.yml
---
- name: use motd role playbook
  hosts: remote.example.com
  remote_user: devops
  become: true
  roles:
    - role: motd
      system_owner: someone@host.example.com
```

重要

在 play 中使用角色变量时，变量的优先顺序可能会让人困惑。

· 几乎任何其他变量都会覆盖角色的默认变量，如清单变量、play **vars** 变量，以及内嵌的角色参数等。

· 较少的变量可以覆盖角色的 **vars** 目录中定义的变量。事实、通过 **include_vars** 加载的变量、注册的变量和角色参数是其中一些具备这种能力的变量。清单变量和 play **vars** 无此能力。这非常重要，因为它有助于避免您的 play 意外改变角色的内部功能。

· 不过，正如上述示例中最后一个所示，作为角色参数内嵌声明的变量具有非常高的优先级。它们可以覆盖角色的 **vars** 目录中定义的变量。如果某一角色参数的名称与 play **vars** 或角色 **vars** 中设置的变量或者清单变量或 playbook 变量的名称相同，该角色参数将覆盖另一个变量。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

参考文献

使用角色 — Ansible 文档
https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html#using-roles

使用变量 — Ansible 文档
https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

指导练习

创建角色

在本练习中，您将创建一个 Ansible 角色，该角色使用变量、文件、模板、任务和处理程序来部署网络服务。

成果

您应能够创建使用变量和参数的角色。

myvhost 角色将在主机上安装和配置 Apache 服务。我们已提供了一个模板 **vhost.conf.j2**，它将用于生成 **/etc/httpd/conf.d/vhost.conf**。

在你开始之前

以 student 用户身份并使用 student 作为密码登录 workstation。

在 workstation 上，运行 **lab role-create start** 命令。这会创建工作目录 **/home/student/role-create**，并为它填充 Ansible 配置文件和主机清单。

```
[student@workstation ~]$ lab role-create start
```

1. 更改到 **/home/student/role-create** 工作目录。

```
[student@workstation ~]$ cd ~/role-create
[student@workstation role-create]$
```

2. 为名为 **myvhost** 的角色创建目录结构。该角色将包含固定的文件、模板、任务和处理程序。

```
[student@workstation role-create]$ mkdir -v roles; cd roles
mkdir: created directory 'roles'
[student@workstation roles]$ ansible-galaxy init myvhost
- myvhost was created successfully
[student@workstation roles]$ rm -rvf myvhost/{defaults,vars,tests}
removed 'myvhost/defaults/main.yml'
removed directory: 'myvhost/defaults'
removed 'myvhost/vars/main.yml'
removed directory: 'myvhost/vars'
removed 'myvhost/tests/inventory'
removed 'myvhost/tests/test.yml'
removed directory: 'myvhost/tests'
[student@workstation roles]$ cd ..
[student@workstation role-create]$
```

3. 在角色的 **tasks** 子目录中编辑 **main.yml** 文件。该角色应当执行下列任务：

海量视频题库 myitpub.com QQ:556462 www.52myit.com

- httpd 软件包已经安装。
- **httpd** 服务已启动且已启用
- 已利用角色提供的模板安装 Web 服务器配置文件

3.1. 编辑 **roles/myvhost/tasks/main.yml** 文件。包含相应的代码，以使用 **yum** 模块安装 httpd 软件包。文件内容应当类似于下方所示：

```
---
# tasks file for myvhost

- name: Ensure httpd is installed
  yum:
    name: httpd
    state: latest
```

3.2. 在 **roles/myvhost/tasks/main.yml** 文件中添加额外的代码，以使用 **service** 模块启动并启用 **httpd** 服务。

```
- name: Ensure httpd is started and enabled
  service:
    name: httpd
    state: started
    enabled: true
```

3.3. 添加另一个小节，以使用 **template** 模块在受管主机上创建 **/etc/httpd/conf.d/vhost.conf**。它应当调用一个处理程序，从而在更新此文件时重新启动 **httpd** 守护程序。

```
- name: vhost file is installed
  template:
    src: vhost.conf.j2
    dest: /etc/httpd/conf.d/vhost.conf
    owner: root
    group: root
    mode: 0644
  notify:
    - restart httpd
```

3.4. 保存更改并退出 **roles/myvhost/tasks/main.yml** 文件。

4. 创建用于重新启动 **httpd** 服务的处理程序。编辑 **roles/myvhost/handlers/main.yml** 文件，以包含相应的代码来使用 **service** 模块，然后保存并退出。文件内容应当类似于下方所示：

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
---
# handlers file for myvhost

- name: restart httpd
  service:
    name: httpd
    state: restarted
```

5. 将 **vhost.conf.j2** 模板从项目目录移到角色的 **templates** 子目录。

```
[student@workstation role-create]$ mv -v vhost.conf.j2 roles/myvhost/templates/
renamed 'vhost.conf.j2' -> 'roles/myvhost/templates/vhost.conf.j2'
```

6. 创建将由 Web 服务器提供的 HTML 内容。

6.1. 创建 **files/html/** 目录来存储内容。

```
[student@workstation role-create]$ mkdir -pv files/html
mkdir: created directory 'files/html'
```

6.2. 在该目录下创建 **index.html** 文件，文件包含以下内容：**simple index**。

```
[student@workstation role-create]$ echo \
> 'simple index' > files/html/index.html
```

7. 测试 **myvhost** 角色，确保它正确工作。

7.1. 编写使用该角色的 playbook，名为 **use-vhost-role.yml**。包括一个复制 **files/html/** 中 HTML 内容的任务。使用 **copy** 模块，并在源目录名称后面加上一个尾斜杠。它应当包含以下内容：

```
---
- name: Use myvhost role playbook
  hosts: webservers
  pre_tasks:
    - name: pre_tasks message
      debug:
        msg: 'Ensure web server configuration.'

  roles:
    - myvhost

  post_tasks:
    - name: HTML content is installed
      copy:
        src: files/html/
        dest: "/var/www/vhosts/{{ ansible_hostname }}"
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
  - name: post_tasks message
    debug:
      msg: 'Web server is configured.'
```

7.2. 在运行该 playbook 前，使用 **--syntax-check** 通过运行 **ansible-playbook** 来验证其语法是否正确。如果报告任何错误，请更正后再继续下一步。您应看到类似于下文的输出：

```
[student@workstation role-create]$ ansible-playbook use-vhost-role.yml \
> --syntax-check

playbook: use-vhost-role.yml
```

7.3. 运行 playbook。检查输出，确认 Ansible 对 Web 服务器 servera 执行了相应的操作。

```
[student@workstation role-create]$ ansible-playbook use-vhost-role.yml

PLAY [Use myvhost role playbook] ********************************************

TASK [Gathering Facts] ******************************************************
ok: [servera.lab.example.com]

TASK [pre_tasks message] ****************************************************
ok: [servera.lab.example.com] => {
    "msg": "Ensure web server configuration."
}

TASK [myvhost : Ensure httpd is installed] **********************************
changed: [servera.lab.example.com]

TASK [myvhost : Ensure httpd is started and enabled] ************************
changed: [servera.lab.example.com]

TASK [myvhost : vhost file is installed] ************************************
changed: [servera.lab.example.com]

RUNNING HANDLER [myvhost : restart httpd] ***********************************
changed: [servera.lab.example.com]

TASK [HTML content is installed] ********************************
changed: [servera.lab.example.com]

TASK [post_tasks message] ***************************************************
ok: [servera.lab.example.com] => {
    "msg": "Web server is configured."
}

PLAY RECAP ******************************************************************
servera.lab.example.com    : ok=8    changed=5    unreachable=0    failed=0
```

7.4. 运行临时命令来确认角色发挥了作用。httpd 软件包应当已安装，并且 httpd 服务应当在运行中。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[student@workstation role-create]$ ansible webservers -a \
> 'systemctl is-active httpd'
servera.lab.example.com | CHANGED | rc=0 >>
active

[student@workstation role-create]$ ansible webservers -a \
> 'systemctl is-enabled httpd'
servera.lab.example.com | CHANGED | rc=0 >>
enabled
```

7.5. Apache 配置应当已使用展开的模板变量进行了安装。

```
[student@workstation role-create]$ ansible webservers -a \
> 'cat /etc/httpd/conf.d/vhost.conf'
servera.lab.example.com | CHANGED | rc=0 >>
# Ansible managed:

<VirtualHost *:80>
    ServerAdmin webmaster@servera.lab.example.com
    ServerName servera.lab.example.com
    ErrorLog logs/servera-error.log
    CustomLog logs/servera-common.log common
    DocumentRoot /var/www/vhosts/servera/

    <Directory /var/www/vhosts/servera/>
        Options +Indexes +FollowSymlinks +Includes
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

7.6. 名为 **/var/www/vhosts/servera** 的目录下应当可找到 HTML 内容。**index.html** 文件应当包含字符串 “simple index”。

```
[student@workstation role-create]$ ansible webservers -a \
> 'cat /var/www/vhosts/servera/index.html'
servera.lab.example.com | CHANGED | rc=0 >>
simple index
```

7.7. 在一个临时命令中使用 `uri` 模块来检查 Web 内容是否在本地可用。将 `return_content` 参数设为 **true**，使服务器的响应内容添加至输出中。服务器内容应当是字符串 “**simple index\n**”。

```
[student@workstation role-create]$ ansible webservers -m uri \
> -a 'url=http://localhost return_content=true'
servera.lab.example.com | SUCCESS => {
    "accept_ranges": "bytes",
    "changed": false,
    "connection": "close",
    "content": "simple index\n",
...output omitted...
```

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
    "status": 200,
    "url": "http://localhost"
}
```

7.8. 确认 Web 服务器内容对远程客户端可用。

```
[student@workstation role-create]$ curl http://servera.lab.example.com
simple index
```

完成

运行 **lab role-create finish** 命令，以清理受管主机。

```
[student@workstation ~]$ lab role-create finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

使用 ANSIBLE GALAXY 部署角色

培训目标

学完本节后，您应能够从 Ansible Galaxy 或其他来源（例如 Git 存储库）中选择和检索角色，并在 playbook 中使用它们。

介绍 ANSIBLE GALAXY

Ansible Galaxy [https://galaxy.ansible.com] 是一个 Ansible 内容公共资源库，这些内容由许许多多 Ansible 管理员和用户编写。它包含数千个 Ansible 角色，具有可搜索的数据库，可帮助 Ansible 用户确定或许有助于他们完成管理任务的角色。Ansible Galaxy 含有面向新的 Ansible 用户和角色开发人员的文档和视频链接。

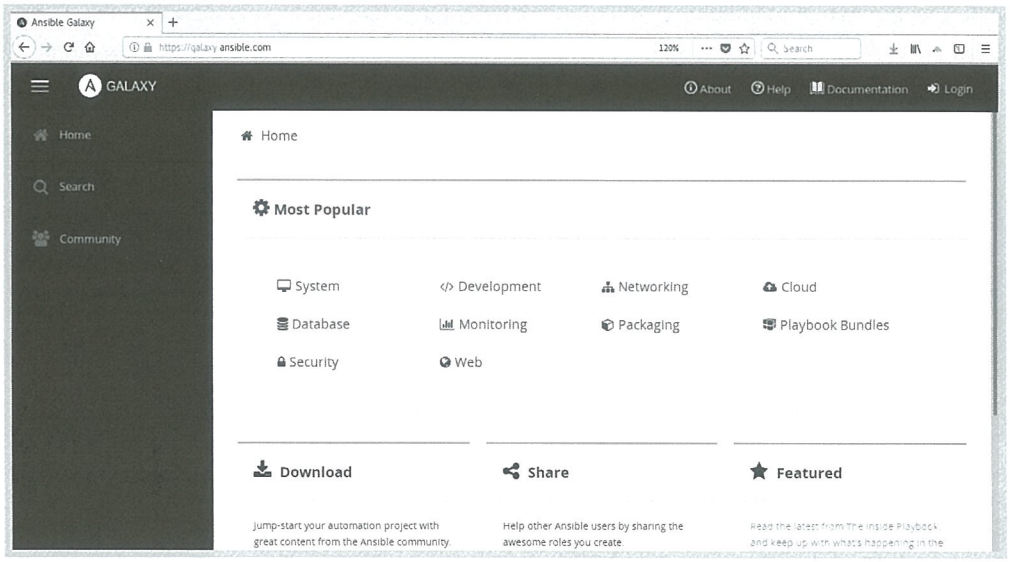

图 8.1: Ansible Galaxy 主页

此外，您用于从 Ansible Galaxy 获取和管理角色的 **`ansible-galaxy`** 命令也可用于为您的项目获取和管理您自有的 Git 存储库中的角色。

获取 Ansible Galaxy 帮助

通过 Ansible Galaxy 网站主页上的 Documentation 标签，可以进入描述如何使用 Ansible Galaxy 的页面。其中包含了介绍如何从 Ansible Galaxy 下载和使用角色的内容。该页面也提供关于如何开发角色并上传到 Ansible Galaxy 的说明。

浏览 Ansible Galaxy 中的角色

通过 Ansible Galaxy 网站主页上左侧的 Search 标签，用户可以访问关于 Ansible Galaxy 上发布的角色的信息。您可以使用标记通过角色的名称或通过其他角色属性来搜索 Ansible 角色。结果按照 **`Best Match`** 分数降序排列，此分数依据角色质量、角色受欢迎程度和搜索条件计算而得。

海量视频题库 myitpub.com QQ:565462
www.52mvit.com

> **注意**
> 文档中的内容评分 [https://galaxy.ansible.com/docs/contributing/content_scoring.html] 更详细地介绍了 Ansible Galaxy 如何对角色进行评分。

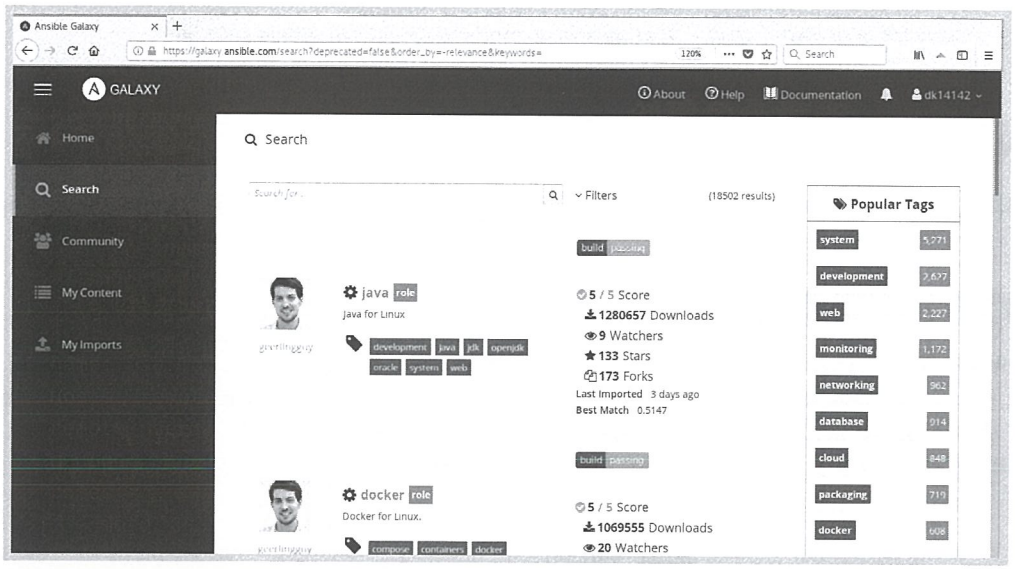

图 8.2: Ansible Galaxy 搜索屏幕

Ansible Galaxy 报告各个角色从 Ansible Galaxy 下载的次数。此外，Ansible Galaxy 也报告角色的 GitHub 存储库拥有的观察者、分叉和星星的数量。用户可以使用此信息来帮助确定角色的开发活跃程度，以及角色在社区中的受欢迎程度。

下图显示了在执行 **redis** 关键字搜索后 Ansible Galaxy 显示的搜索结果。注意第一个结果的 **Best Match** 分数为 **0.9009**。

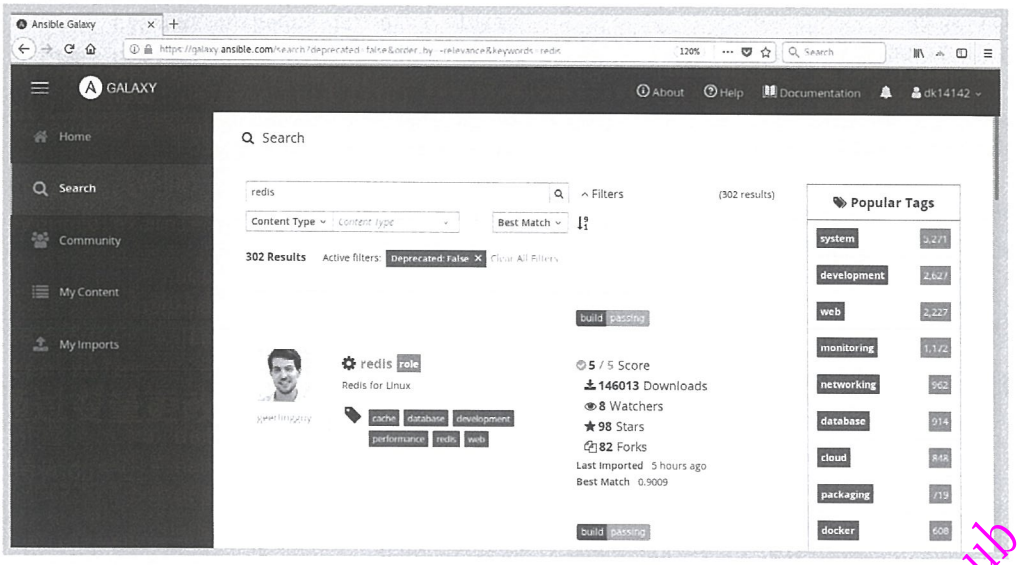

图 8.3: Ansible Galaxy 搜索结果示例

通过搜索框右侧的 Filters 下拉菜单，可以按照关键字、作者 ID、平台和标记来执行搜索。可能的平台值包括 **EL**（代表红帽企业 Linux，及 CentOS 等紧密相关的分发）以及 **Fedora** 等。

海量视频题库 itpub.com QQ:5565462
www.52mvit.com

标记是角色作者设置的任意单词字符串，用于描述和分类角色。用户可以使用标记查找相关的角色。可能的标记值包括 **system**、**development**、**web** 和 **monitoring** 等。在 Ansible Galaxy 中，一个角色最多可具有 20 个标记。

> **重要**
> 在 Ansible Galaxy 搜索界面中，关键字搜索匹配 **README** 文件、内容名称或内容描述中的词语或词组。相反，标记搜索则专门匹配作者为角色设置的标记值。

ANSIBLE GALAXY 命令行工具

ansible-galaxy 命令行工具可以用于搜索角色，显示角色相关的信息，以及安装、列举、删除或初始化角色。

从命令行搜索角色

ansible-galaxy search 子命令在 Ansible Galaxy 中搜索角色。如果以参数形式指定了字符串，则可用于按照关键字在 Ansible Galaxy 中搜索角色。您可以使用 **--author**、**--platforms** 和 **--galaxy-tags** 选项来缩小搜索结果的范围。也可以将这些选项用作主要的搜索键。例如，命令 **ansible-galaxy search --author geerlingguy** 将显示由用户 geerlingguy 提交的所有角色。

结果按照字母顺序显示，而不是 **Best Match** 分数降序排列。下例显示了包含 **redis** 并且适用于企业 Linux (**EL**) 平台的角色的名称。

```
[user@host ~]$ ansible-galaxy search 'redis' --platforms EL

Found 124 roles matching your search:

 Name                                  Description
 ----                                  -----------
 1it.sudo                              Ansible role for managing sudoers
 AerisCloud.librato                    Install and configure the Librato Agent
 AerisCloud.redis                      Installs redis on a server
 AlbanAndrieu.java                     Manage Java installation
 andrewrothstein.redis                 builds Redis from src and installs
...output omitted...
 geerlingguy.php-redis                 PhpRedis support for Linux
 geerlingguy.redis                     Redis for Linux
 gikoluo.filebeat                      Filebeat for Linux.
...output omitted...
```

ansible-galaxy info 子命令显示与角色相关的更多详细信息。Ansible Galaxy 从多个位置获取这一信息，包括角色的 **meta/main.yml** 文件及其 GitHub 存储库。以下命令显示了 Ansible Galaxy 提供的 geerlingguy.redis 角色的相关信息。

```
[user@host ~]$ ansible-galaxy info geerlingguy.redis

Role: geerlingguy.redis
        description: Redis for Linux
        active: True
...output omitted...
        download_count: 146209
```

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

```
        forks_count: 82
        github_branch: master
        github_repo: ansible-role-redis
        github_user: geerlingguy
...output omitted...
        license: license (BSD, MIT)
        min_ansible_version: 2.4
        modified: 2018-11-19T14:53:29.722718Z
        open_issues_count: 11
        path: [u'/etc/ansible/roles', u'/usr/share/ansible/roles']
        role_type: ANS
        stargazers_count: 98
...output omitted...
```

从 Ansible Galaxy 安装角色

ansible-galaxy install 子命令从 Ansible Galaxy 下载角色，并将它安装到控制节点本地。

默认情况下，角色安装到用户的 roles_path 下的第一个可写目录中。根据为 Ansible 设置的默认 roles_path，角色通常将安装到用户的 **~/.ansible/roles** 目录。默认的 roles_path 可能会被您的当前 Ansible 配置文件或环境变量 ANSIBLE_ROLES_PATH 覆盖，这将影响 **ansible-galaxy** 的行为。

您也可以通过使用 **-p DIRECTORY** 选项，指定具体的目录来安装角色。

在下例中，**ansible-galaxy** 将 geerlingguy.redis 角色安装到 playbook 项目的 **roles** 目录中。命令的当前工作目录是 **/opt/project**。

```
[user@host project]$ ansible-galaxy install geerlingguy.redis -p roles/
- downloading role 'redis', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/...output omitted...
- extracting geerlingguy.redis to /opt/project/roles/geerlingguy.redis
- geerlingguy.redis (1.6.0) was installed successfully
[user@host project]$ ls roles/
geerlingguy.redis
```

使用要求文件安装角色

您也可以使用 **ansible-galaxy**，根据某一文本文件中的定义来安装一个角色列表。例如，如果您的一个 playbook 需要安装特定的角色，您可以在项目目录中创建一个 **roles/requirements.yml** 文件来指定所需的角色。此文件充当 playbook 项目的依赖项清单，使得 playbook 的开发和测试能与任何支持角色分开进行。

例如，一个用于安装 geerlingguy.redis 的简单 **requirements.yml** 可能类似于如下：

```
- src: geerlingguy.redis
  version: "1.5.0"
```

src 属性指定角色的来源，本例中为来自 Ansible Galaxy 的 geerlingguy.redis 角色。version 属性是可选的，指定要安装的角色版本，本例中为 **1.5.0**。

海量视频题库 myitpub.com QQ:5565462 www.52myit.com

> **重要**
> 您应当在 **requirements.yml** 文件中指定角色版本，特别是生产环境中的 playbook。
>
> 如果不指定版本，您将获取角色的最新版本。如果上游作者对角色做出了更改，并与您的 playbook 不兼容，这可能会造成自动化失败或其他问题。

若要使用角色文件来安装角色，可使用 **-r REQUIREMENTS-FILE** 选项：

```
[user@host project]$ ansible-galaxy install -r roles/requirements.yml \
> -p roles
- downloading role 'redis', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-redis/
archive/1.6.0.tar.gz
- extracting geerlingguy.redis to /opt/project/roles/geerlingguy.redis
- geerlingguy.redis (1.6.0) was installed successfully
```

您可以使用 **ansible-galaxy** 来安装不在 Ansible Galaxy 中的角色。您可以在私有的 Git 存储库或 Web 服务器上托管自有的专用或内部角色。下例演示了如何利用各种远程来源配置要求文件。

```
[user@host project]$ cat roles/requirements.yml
# from Ansible Galaxy, using the latest version
- src: geerlingguy.redis

# from Ansible Galaxy, overriding the name and using a specific version
- src: geerlingguy.redis
  version: "1.5.0"
  name: redis_prod

# from any Git-based repository, using HTTPS
- src: https://gitlab.com/guardianproject-ops/ansible-nginx-acme.git
  scm: git
  version: 56e00a54
  name: nginx-acme

# from any Git-based repository, using SSH
- src: git@gitlab.com:guardianproject-ops/ansible-nginx-acme.git
  scm: git
  version: master
  name: nginx-acme-ssh

# from a role tar ball, given a URL;
#   supports 'http', 'https', or 'file' protocols
- src: file:///opt/local/roles/myrole.tar
  name: myrole
```

src 关键字指定 Ansible Galaxy 角色名称。如果角色没有托管在 Ansible Galaxy 中，则 **src** 关键字将指明角色的 URL。

如果角色托管在来源控制存储库中，则需要使用 **scm** 属性。**ansible-galaxy** 命令能够从基于 Git 或 Mercurial 的软件存储库下载和安装角色。基于 Git 的存储库要求 **scm** 值为 **git**，而托管在

海量视频题库 myitpub.com QQ:556462
www.53ayit.com

Mercurial 存储库中的角色则要求值为 **hg**。如果角色托管在 Ansible Galaxy 中，或者以 tar 存档形式托管在 Web 服务器上，则省略 **scm** 关键字。

name 关键字用于覆盖角色的本地名称。**version** 关键字用于指定角色的版本。**version** 关键字可以是与来自角色的软件存储库的分支、标记或提交哈希对应的任何值。

若要安装与 playbook 项目关联的角色，可执行 **ansible-galaxy install** 命令：

```
[user@host project]$ ansible-galaxy install -r roles/requirements.yml \
> -p roles
- downloading role 'redis', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-redis/
archive/1.6.0.tar.gz
- extracting geerlingguy.redis to /opt/project/roles/geerlingguy.redis
- geerlingguy.redis (1.6.0) was installed successfully
- downloading role 'redis', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-redis/
archive/1.5.0.tar.gz
- extracting redis_prod to /opt/project/roles/redis_prod
- redis_prod (1.5.0) was installed successfully
- extracting nginx-acme to /opt/project/roles/nginx-acme
- nginx-acme (56e00a54) was installed successfully
- extracting nginx-acme-ssh to /opt/project/roles/nginx-acme-ssh
- nginx-acme-ssh (master) was installed successfully
- downloading role from file:///opt/local/roles/myrole.tar
- extracting myrole to /opt/project/roles/myrole
- myrole was installed successfully
```

管理下载的角色

ansible-galaxy 命令也可管理本地的角色，如位于 playbook 项目的 **roles** 目录中的角色。**ansible-galaxy list** 子命令列出本地找到的角色。

```
[user@host project]$ ansible-galaxy list
- geerlingguy.redis, 1.6.0
- myrole, (unknown version)
- nginx-acme, 56e00a54
- nginx-acme-ssh, master
- redis_prod, 1.5.0
```

可以使用 **ansible-galaxy remove** 子命令本地删除角色。

```
[user@host ~]$ ansible-galaxy remove nginx-acme-ssh
- successfully removed nginx-acme-ssh
[user@host ~]$ ansible-galaxy list
- geerlingguy.redis, 1.6.0
- myrole, (unknown version)
- nginx-acme, 56e00a54
- redis_prod, 1.5.0
```

在 playbook 中使用下载并安装的角色的方式与任何其他角色都一样。在 **roles** 部分中利用其下载的角色名称来加以引用。如果角色不在项目的 **roles** 目录中，则将检查 roles_path 来查看角

海量视频题库 mvitpub.com QQ:556462
www.32mvit.com