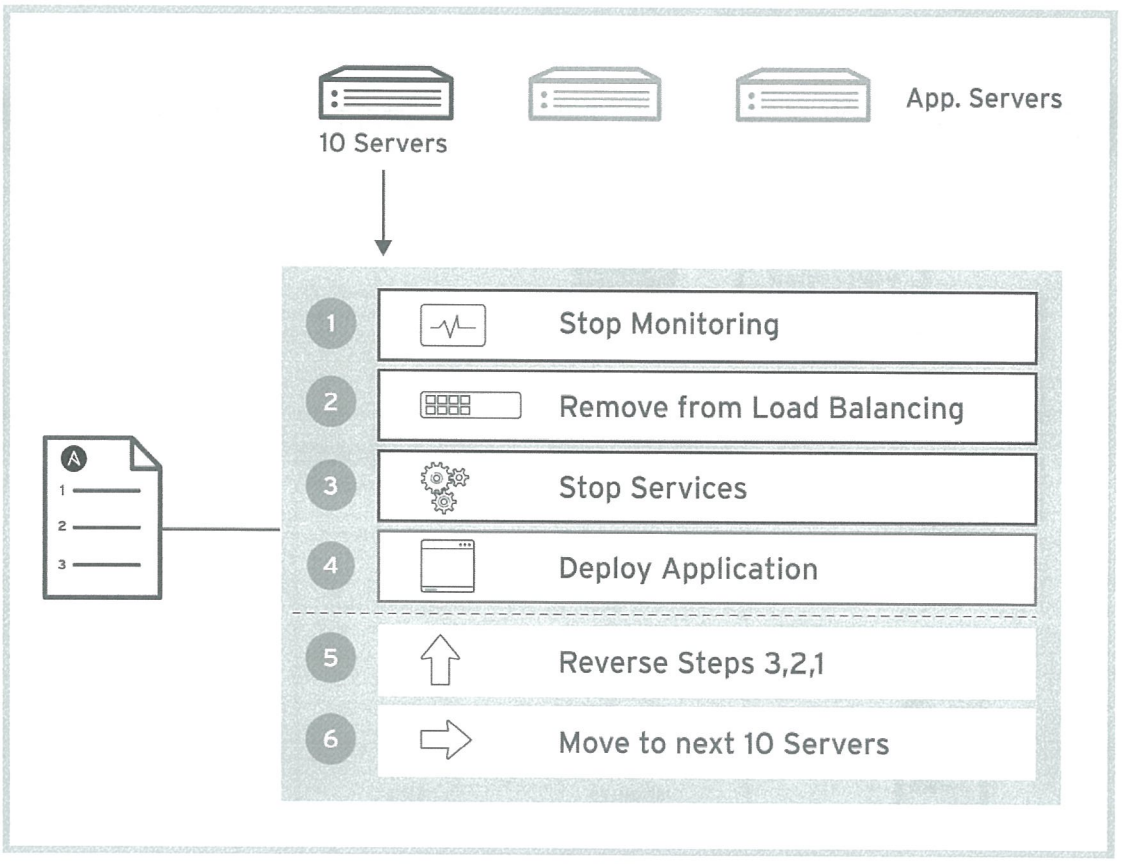

图 1.3: Ansible 提供完整的自动化

用例

与某些其他工具不同，Ansible 在一个易用的平台中，将编排与配置管理、调配和应用部署相结合。

Ansible 的一些用例包括：

配置管理
集中化配置文件管理和部署是 Ansible 的常见用例，很多超级用户也是通过这种方式了解 Ansible 自动化平台。

应用部署
通过 Ansible 定义应用，以及使用红帽 Ansible Tower 管理部署时，各团队可以更加有效地管理从开发到生产的整个应用生命周期。

调配
应用必须要部署或安装到系统上。Ansible 和红帽 Ansible Tower 有助于简化调配系统的流程，不论您是要 PXE 引导和 kickstart 安装裸机恢复服务器或虚拟机，还是从模板创建虚拟机或云实例。应用必须要部署或安装到系统上。

持续交付
创建 CI/CD 管道需要多个团队的协调和参与。如果没有组织内人人可用的简单自动化平台，就无法实现这个目标。Ansible Playbook 让您的应用可以在整个生命周期内得到正确部署（和管理）。

安全和合规性
当您在 Ansible Playbook 中定义安全策略时，也可以将扫描和修复整站安全策略集成到其他自动化流程中。确保安全应该是您所有部署中不可或缺的组成部分，而不是事后才去考虑的部分。

海量视频题库 hvitpub.com QQ:3565462
www.52hvit.com

编排

仅配置本身不足以定义您的环境。您还需要定义多个配置间应如何交互，并且确保以整体的方式管理各类分散的资源。

参考文献

Ansible
https://www.ansible.com

Ansible 的工作方式
https://www.ansible.com/how-ansible-works

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

小测验

使用 ANSIBLE 自动化 LINUX 管理任务

选择以下问题的正确答案：

1. 以下哪一个词语最适合描述 Ansible 架构？

a. 无代理
b. 客户端/服务器
c. 事件驱动
d. 无状态

2. Ansible 默认使用哪一种网络协议与受管节点通信？

a. HTTP
b. HTTPS
c. SNMP
d. SSH

3. 以下哪一种文件定义 Ansible 在受管节点上执行的操作？

a. 主机清单
b. 清单
c. Playbook
d. 脚本

4. 哪一种语法用于定义 Ansible Playbook？

a. Bash
b. Perl
c. Python
d. YAML

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

解决方案

使用 ANSIBLE 自动化 LINUX 管理任务

选择以下问题的正确答案：

1. 以下哪一个词语最适合描述 Ansible 架构？

a. 无代理
b. 客户端/服务器
c. 事件驱动
d. 无状态

2. Ansible 默认使用哪一种网络协议与受管节点通信？

a. HTTP
b. HTTPS
c. SNMP
d. SSH

3. 以下哪一种文件定义 Ansible 在受管节点上执行的操作？

a. 主机清单
b. 清单
c. Playbook
d. 脚本

4. 哪一种语法用于定义 Ansible Playbook？

a. Bash
b. Perl
c. Python
d. YAML

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

安装 ANSIBLE

培训目标

学完本节后，您应能够在控制节点上安装 Ansible 并描述社区 Ansible 和红帽 Ansible 引擎之间的区别。

ANSIBLE 或红帽 ANSIBLE 自动化?

红帽以特殊渠道提供 Ansible 软件，以方便红帽企业 Linux 订户，并且您可以正常使用这些软件包。

但是，如果您需要 Ansible 及其模块的正式支持，红帽为此提供了特殊订阅，即红帽 Ansible 自动化。此订阅包括对 Ansible 本身的支持，即红帽 Ansible 引擎。其中增加了 SLA 正式技术支持和 Ansible 及其核心模块已发布的覆盖范围。有关此支持范围的更多信息，请访问 红帽 Ansible 引擎生命周期 [https://access.redhat.com/support/policy/updates/ansible-engine]。

控制节点

Ansible 易于安装。Ansible 软件只需要安装到要运行它的一个（或多个）控制节点上。由 Ansible 管理的主机不需要安装 Ansible。这种安装涉及相对较少的步骤，其要求也最少。

控制节点应当是 Linux 或 UNIX 系统。不支持将 Microsoft Windows 用作控制节点，但 Windows 系统可以是受管主机。

控制节点上需要安装 Python 3（版本 3.5 或以上）或 Python 2（版本 2.7 或以上）。

重要

如果您运行红帽企业 Linux 8，Ansible 2.8 可以自动使用 platform-python 软件包，该软件包支持使用 Python 的系统实用程序。您不需要从 AppStream 安装 python36 或 python27 软件包。

```
[root@controlnode ~]# yum list installed platform-python
Loaded plugins: langpacks, search-disabled-repos
Installed Packages
platform-python.x86_64      3.6.8-2.el8_0     @anaconda
```

有关如何在红帽企业 Linux 系统上安装 ansible 软件包的信息，请参阅知识库文章 如何下载和安装红帽 Ansible 引擎？ [https://access.redhat.com/articles/3174981]。

Ansible 正处于上游快速发展阶段，因此红帽 Ansible 引擎具有快速的生命周期。 有关当前生命周期的更多信息，请访问 https://access.redhat.com/support/policy/updates/ansible-engine。

红帽为红帽企业 Linux 8 提供了 `ansible-2.8-for-rhel-8-x86_64-rpms` 频道。您还可以在 `ansible-2-for-rhel-8-x86_64-rpms` 频道中获得 RHEL 8 的红帽 Ansible 引擎 2 主要版本的最新更新。

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

您可以使用这些频道来安装 Ansible，并享有标准红帽企业 Linux 订阅的有限支持。 如果您需要更全面的 Ansible 支持，可以在启用频道之前购买完整的红帽 Ansible 引擎订阅并将其与您的系统相关联，如知识库文章中所述。

如果您有红帽 Ansible 引擎订阅，则红帽 Ansible 引擎 2 的安装过程如下所示：

> **警告**
> 您无需在课堂环境中运行这些步骤。

1. 将您的系统注册到红帽订阅管理工具。

```
[root@host ~]# subscription-manager register
```

2. 为您的系统设置角色。

```
[root@host ~]# subscription-manager role --set="Red Hat Enterprise Linux Server"
```

3. 附加您的红帽 Ansible 引擎订阅。 此命令可帮助您找到红帽 Ansible 引擎订阅：

```
[root@host ~]# subscription-manager list --available
```

4. 使用订阅的池 ID 将池附加到系统。

```
[root@host ~]# subscription-manager attach --pool=<engine-subscription-pool>
```

5. 启用红帽 Ansible 引擎存储库。

```
[root@host ~]# subscription-manager repos \
> --enable ansible-2-for-rhel-8-x86_64-rpms
```

6. 安装红帽 Ansible 引擎。

```
[root@host ~]# yum install ansible
```

如果您使用随红帽企业 Linux 订阅提供的有限支持版本，则使用以下过程：

1. 启用红帽 Ansible 引擎存储库。

```
[root@host ~]# subscription-manager refresh
[root@host ~]# subscription-manager repos \
> --enable ansible-2-for-rhel-8-x86_64-rpms
```

2. 安装红帽 Ansible 引擎。

```
[root@host ~]# yum install ansible
```

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

受管主机

Ansible 的一大优点是受管主机不需要安装特殊的代理。Ansible 控制节点使用标准的网络协议连接受管主机，从而确保系统处于指定的状态。

受管主机可能要满足一些要求，具体取决于控制节点连接它们的方式以及它们要运行的模块。

Linux 和 UNIX 受管主机需要安装有 Python 2（版本 2.6 或以上）或 Python 3（版本 3.5 或以上），这样才能运行大部分的模块。

对于红帽企业 Linux 8，您可以依靠 platform-python 软件包。您也可以启用并安装 python36 应用流（或 python27 应用流）。

```
[root@host ~]# yum module install python36
```

如果受管主机上启用了 SELinux，您还需要确保安装 python3-libselinux 软件包，然后才能使用与任何复制、文件或模板功能相关的模块。（请注意，如果安装了其他 Python 组件，您可以使用 `yum` 或 `package` 等 Ansible 模块来确保也安装了这一软件包）。

> **重要**
> 由于持续向 Python 3 迁移的缘故，一些软件包名称在红帽企业 Linux 7 和更早版本中可能有所不同。
>
> 对于红帽企业 Linux 7 和更早版够本，请安装 python 软件包，此软件包提供 Python 2。您需要确保安装了 libselinux-python 软件包，而非 python3-libselinux。

一些模块可能有自己的额外要求。例如，`dnf` 模块可用于在当前的 Fedora 系统上安装软件包，此模块需要安装 python3-dnf 软件包（RHEL 7 中为 python-dnf）。

> **注意**
> 一些模块不需要 Python。例如，传递到 Ansible `raw` 模块的参数直接通过配置的远程 shell 运行，而不通过模块子系统。这可在管理没有 Python 可用或无法安装 Python 的设备时使用，或者用于将 Python 引导到没有它的系统上。
>
> 不过，`raw` 模块难以通过安全的幂等方式使用。如果能够改用普通的模块，通常最好避免使用 `raw` 和类似的其他命令模块。这将在本课程后面进一步讨论。

基于 Microsoft Windows 的受管主机

Ansible 随附了多个专门为 Microsoft Windows 系统设计的模块。这些模块列在 Ansible 模块索引的 Windows Modules [https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html] 部分中。

大部分专门为 Microsoft Windows 受管主机设计的模块需要在受管主机上安装 PowerShell 3.0 或更高版本，而不是安装 Python。此外，受管主机也需要配置 PowerShell 远程连接。Ansible 还要求至少将 .NET Framework 4.0 或更高版本安装在 Windows 受管主机上。

本课程的示例中使用基于 Linux 的受管主机，不会深入阐述管理基于 Microsoft Windows 的受管主机时的具体差别和必要调整。可以在 Ansible 网站上查看更多信息，网址为 https://docs.ansible.com/ansible/latest/user_guide/windows.html。

海量视频题库 mytpub.com QQ:556462
www.52mvi.com

受管网络设备

您还可以使用 Ansible 自动化来配置受管网络设备，例如路由器和交换机。Ansible 包含大量专门为此目的而设计的模块。其中包括对 Cisco IOS、IOS XR 和 NX-OS 的支持；Juniper Junos；Arista EOS；以及基于 VyOS 的网络设备等。

您可以使用为服务器编写 playbook 时使用的相同基本技术为网络设备编写 Ansible Playbook。由于大多数网络设备无法运行 Python，因此 Ansible 在控制节点上运行网络模块，而不是在受管主机上运行。特殊连接方法也用于与网络设备通信，通常使用 SSH 上的 CLI、SSH 上的 XML 或 HTTP(S) 上的 API。

本课程不论述任何深度的网络设备管理自动化。有关此主题的更多信息，请参阅 Ansible 社区网站上的利用 Ansible 实现网络自动化 [https://docs.ansible.com/ansible/latest/network/index.html]，或参加我们的备选课程利用 Ansible 实现网络自动化 (DO457) [https://www.redhat.com/en/services/training/do457-ansible-network-automation]。

参考文献

`ansible`(1) 帮助手册

用于红帽 Ansible 自动化的首要支持策略
https://access.redhat.com/ansible-top-support-policies

安装指南 — Ansible 文档
http://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

Windows 指南 — Ansible 文档
https://docs.ansible.com/ansible/latest/user_guide/windows.html

利用 Ansible 实现网络自动化 — Ansible 文档
https://docs.ansible.com/ansible/latest/network/index.html

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

指导练习

安装 ANSIBLE

在本练习中，您将在运行红帽企业 Linux 操作系统的控制节点上安装 Ansible。

成果

您应能够在控制节点上安装 Ansible。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`，并运行 **`lab intro-install start`**。此启动脚本配置控制节点。

```
[student@workstation ~]$ lab intro-install start
```

1. 在 `workstation` 上安装 Ansible，使它可以充当控制节点。

```
[student@workstation ~]$ sudo yum install ansible
[sudo] password for student:
Loaded plugins: langpacks, search-disabled-repos
Resolving Dependencies
--> Running transaction check
...output omitted...
Is this ok [y/d/N]: y
...output omitted...
```

2. 验证系统上是否安装了 Ansible。使用 **`--version`** 选项执行 **`ansible`** 命令。

```
[student@workstation ~]$ ansible --version
ansible 2.8.0
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/student/.ansible/plugins/modules', '/
usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.6.8 (default, Apr  3 2019, 17:26:03) [GCC 8.2.1 20180905 (Red
 Hat 8.2.1-3)]
```

3. 通过使用 setup 模块验证 localhost 上的 ansible_python_version。

```
[student@workstation ~]$ ansible -m setup localhost | grep ansible_python_version
    "ansible_python_version": "3.6.8",
```

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

完成

在 workstation 上，运行 **lab intro-install finish** 脚本来清理本练习。

```
[student@workstation ~]$ lab intro-install finish
```

本引导式练习到此结束。

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

总结

在本章中，您学到了：

- 自动化是减少人为错误并快速确保 IT 基础架构处于一致且正确的状态的关键工具。
- Ansible 是一种开源自动化平台，能够适应许多不同的工作流和环境。
- Ansible 可用于管理许多不同类型的系统，包括运行 Linux、Microsoft Windows 或 UNIX 的服务器以及网络设备。
- Ansible Playbook 是人类可读的文本文件，用于描述 IT 基础架构的所需状态。
- Ansible 围绕一种无代理架构构建，在控制节点上安装 Ansible，且客户端不需要任何特殊的代理软件。
- Ansible 使用 SSH 等标准协议连接受管主机，并在受管主机上运行代码或命令来确保它们处于 Ansible 指定的状态。

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

章 2

部署 ANSIBLE

目标

配置 Ansible 以管理主机和运行临时 Ansible 命令。

培训目标

- 描述 Ansible 清单概念并管理静态清单文件。
- 描述 Ansible 配置文件的位置、Ansible 如何选择这些文件，并编辑它们以对默认设置应用更改。
- 使用临时命令运行单个 Ansible 自动化任务，并解释临时命令的一些用例。

章节

- 构建 Ansible 清单（及引导式练习）
- 管理 Ansible 配置文件（及引导式练习）
- 运行临时命令（及引导式练习）

实验

- 部署 Ansible

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

构建 ANSIBLE 清单

培训目标

学完本章节后，您应该能够描述 Ansible 清单概念并且管理静态清单文件。

定义清单

清单定义 Ansible 将要管理的一批主机。这些主机也可以分配到组中，以进行集中管理。组可以包含子组，主机也可以是多个组的成员。清单还可以设置应用到它所定义的主机和组的变量。

可以通过两种方式定义主机清单。静态主机清单可以通过文本文件来定义。动态主机清单可以根据需要使用外部信息提供程序通过脚本或其他程序来生成。

使用静态清单指定受管主机

静态清单文件是指定 Ansible 目标受管主机的文本文件。您可以使用多种不同的格式编写此文件，包括 INI 样式或 YAML。INI 样式的格式很常见，将用于本课程中的大多数示例。

> **注意**
> Ansible 支持多种静态清单格式。在本节中，我们将着重介绍最常见的一种，即 INI 样式的格式。

在最简单的形式中，INI 样式的静态清单文件是受管主机的主机名或 IP 地址的列表，每行一个：

```
web1.example.com
web2.example.com
db1.example.com
db2.example.com
192.0.2.42
```

但通常而言，您可以将受管主机组织为主机组。通过主机组，您可以更加有效地对一系列系统运行 Ansible。这时，每一部分的开头为以方括号 ([]) 括起的主机组名称。其后为该组中每一受管主机的主机名或 IP 地址，每行一个。

在以下示例中，主机清单定义了两个主机组：`webservers` 和 `db-servers`。

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

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

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

主机可以在多个组中。事实上，建议您将主机组织为多个组，可以根据主机的角色、其物理位置以及是否在生产环境中等因素以不同的方式加以整理。这可让您轻松地将 Ansible play 应用到特定的主机。

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

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

[east-datacenter]
web1.example.com
db1.example.com

[west-datacenter]
web2.example.com
db2.example.com

[production]
web1.example.com
web2.example.com
db1.example.com
db2.example.com

[development]
192.0.2.42
```

重要

两个主机组始终存在：

- `all` 主机组含有清单中明确列出的每一个主机。
- `ungrouped` 主机组含有清单中明确列出、但不属于任何其他组的每一个主机。

定义嵌套组

Ansible 主机清单可以包含由多个主机组构成的组。这通过创建后缀为 `:children` 的主机组名称来实现。以下示例创建一个名为 `north-america` 的新组，它包含来自 `usa` 和 `canada` 组的所有主机。

```
[usa]
washington1.example.com
washington2.example.com

[canada]
ontario01.example.com
ontario02.example.com
```

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

```
[north-america:children]
canada
usa
```

一个组可以同时包含受管主机和子组作为其成员。例如，在上面的清单中，您可以添加一个拥有自己的受管主机列表的 **[north-america]** 部分。这一列表中的主机将与 north-america 组从其子组中继承的其他主机合并。

通过范围简化主机规格

可以通过指定主机名称或 IP 地址的范围来简化 Ansible 主机清单。您可以指定数字或字母范围。范围具有下列语法：

```
[START:END]
```

范围匹配从 START 到 END（含）的所有值。请思考以下示例：

- 192.168.[4:7].[0:255] 匹配 192.168.4.0/22 网络中的所有 IPv4 地址（192.168.4.0 到 192.168.7.255）。
- server[01:20].example.com 匹配名为 server01.example.com 到 server20.example.com 的所有主机。
- [a:c].dns.example.com 匹配名为 a.dns.example.com、b.dns.example.com 和 c.dns.example.com 的主机。
- 2001:db8::[a:f] 匹配从 2001:db8::a 到 2001:db8::f 的所有 IPv6 地址。

如果数字范围中包含前置零，其模式中会使用它们。上方第二个示例不匹配 server1.example.com，但匹配 server07.example.com。若要进行演示，可参见以下例子；它使用范围来简化前面示例中的 **[usa]** 和 **[canada]** 组定义：

```
[usa]
washington[1:2].example.com

[canada]
ontario[01:02].example.com
```

验证清单

若有疑问，可使用 **ansible** 命令验证计算机是否存在于清单中：

```
[user@controlnode ~]$ ansible washington1.example.com --list-hosts
  hosts (1):
    washington1.example.com
[user@controlnode ~]$ ansible washington01.example.com --list-hosts
 [WARNING]: provided hosts list is empty, only localhost is available

  hosts (0):
```

您可以运行以下命令来列出组中的所有主机：

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

```
[user@controlnode ~]$ ansible canada --list-hosts
  hosts (2):
    ontario01.example.com
    ontario02.example.com
```

重要

如果清单中含有名称相同的主机和主机组，**ansible** 命令将显示警告并以主机作为其目标。主机组则被忽略。

应对这种情况的方法有多种，其中最简单的是确保主机组不使用与清单中主机相同的名称。

覆盖清单的位置

/etc/ansible/hosts 文件被视为系统的默认静态清单文件。不过，通常的做法是不使用该文件，而是在 Ansible 配置文件中为清单文件定义一个不同的位置。这将在下一节介绍。

稍后您会在本课程中利用 **ansible** 和 **ansible-playbook** 命令来运行 Ansible 临时命令和 playbook；这两个命令也可通过 **--inventory PATHNAME** 或 **-i PATHNAME** 选项在命令行中指定清单文件的位置，其中 **PATHNAME** 是所需清单文件的路径。

在清单中定义变量

可以在主机清单文件中指定 playbook 使用的变量值。这些变量仅应用到特定的主机或主机组。通常，最好在特殊目录中定义这些库存变量，而不要直接在清单文件中定义。本课程的其他章节将对这个主题进行更加深入的探讨。

描述动态清单

也可以使用外部数据库提供的信息动态生成 Ansible 清单信息。开源社区编写了许多可从上游 Ansible 项目获取的动态清单脚本。如果这些脚本不能满足您的需求，您也可以编写自己的脚本。

例如，动态清单程序可以联系您的红帽卫星服务器或 Amazon EC2 帐户，并使用那里存储的信息来构造 Ansible 清单。由于该程序在您运行 Ansible 时进行此操作，因此它可以在新主机添加和旧主机移除时为清单填充该服务所提供的最新信息。

本课程的后续部分将对此主题进行更为详细的探讨。

参考文献

清单：Ansible 文档

http://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

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

指导练习

构建 ANSIBLE 清单

在本练习中，您将创建一个包含主机和组的新静态清单。

成果

您应该能够创建默认和自定义静态清单。

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 deploy-inventory start** 命令。此起始脚本将确保受管主机 servera、serverb、serverc 和 serverd 可在网络上访问。

```
[student@workstation ~]$ lab deploy-inventory start
```

1. 修改 **/etc/ansible/hosts** 以包括 servera.lab.example.com 作为受管主机。

   1.1. 将 servera.lab.example.com 添加到默认清单文件 **/etc/ansible/hosts** 的末尾。

```
[student@workstation ~]$ sudo vim /etc/ansible/hosts
...output omitted...
## db-[99:101]-node.example.com

servera.lab.example.com
```

   1.2. 继续编辑 **/etc/ansible/hosts** 清单文件，将 [webservers] 组添加到该文件的底部，并将 serverb.lab.example.com 服务器作为组成员。

```
[student@workstation ~]$ sudo vim /etc/ansible/hosts
...output omitted...
## db-[99:101]-node.example.com

servera.lab.example.com

[webservers]
serverb.lab.example.com
```

2. 验证 **/etc/ansible/hosts** 清单文件中的受管主机。

   2.1. 使用 **ansible all --list-hosts** 命令列出默认清单文件中的所有受管主机。

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

```
[student@workstation ~]$ ansible all --list-hosts
  hosts (2):
    servera.lab.example.com
    serverb.lab.example.com
```

2.2. 使用 **ansible ungrouped --list-hosts** 命令，仅列出不属于某个组的受管主机。

```
[student@workstation ~]$ ansible ungrouped --list-hosts
  hosts (1):
    servera.lab.example.com
```

2.3. 使用 **ansible webservers --list-hosts** 命令，仅列出属于 webservers 组的受管主机。

```
[student@workstation ~]$ ansible webservers --list-hosts
  hosts (1):
    serverb.lab.example.com
```

3. 在 **/home/student/deploy-inventory** 工作目录中，创建一个名为 **inventory** 的自定义静态清单文件。

下表列出了有关四个受管主机的信息。您将根据主机的用途、所在的城市以及它所属的部署环境，将每个主机分配给多个组以进行管理。

此外，美国城市组（罗利和山景城）必须设为组 us 的子项，这样美国的主机就可以作为一个组进行管理。

服务器清单规格

主机名称	用途	位置	运行环境
servera.lab.example.com	Web 服务器	罗利	开发
serverb.lab.example.com	Web 服务器	罗利	测试
serverc.lab.example.com	Web 服务器	山景城	生产
serverd.lab.example.com	Web 服务器	伦敦	生产

3.1. 创建 **/home/student/deploy-inventory** 工作目录。

```
[student@workstation ~]$ mkdir ~/deploy-inventory
```

3.2. 在 **/home/student/deploy-inventory** 工作目录中创建 **inventory** 文件。使用“服务器清单规格”表作为指南。编辑 **inventory** 文件并添加以下内容：

```
[student@workstation ~]$ cd ~/deploy-inventory
[student@workstation deploy-inventory]$ vim inventory
[webservers]
server[a:d].lab.example.com
```

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

```
[raleigh]
servera.lab.example.com
serverb.lab.example.com

[mountainview]
serverc.lab.example.com

[london]
serverd.lab.example.com

[development]
servera.lab.example.com

[testing]
serverb.lab.example.com

[production]
serverc.lab.example.com
serverd.lab.example.com

[us:children]
raleigh
mountainview
```

4. 使用 **ansible** host-or-group **-i inventory --list-hosts** 命令的变体验证自定义 **/home/student/deploy-inventory/inventory** 清单文件中的受管主机和组。

> **重要**
> 您的 **ansible** 命令必须包括 **-i inventory** 选项。这使得 **ansible** 在当前工作目录中使用您的 **inventory** 文件，而不是系统 **/etc/ansible/hosts** 清单文件。

4.1. 使用 **ansible all -i inventory --list-hosts** 命令列出所有受管主机。

```
[student@workstation deploy-inventory]$ ansible all -i inventory --list-hosts
  hosts (4):
    servera.lab.example.com
    serverb.lab.example.com
    serverc.lab.example.com
    serverd.lab.example.com
```

4.2. 使用 **ansible ungrouped -i inventory --list-hosts** 命令列出清单文件中列出的、但不属于某个组的所有受管主机。此清单文件中没有未分组的受管主机。

```
[student@workstation deploy-inventory]$ ansible ungrouped -i inventory \
> --list-hosts
 [WARNING]: No hosts matched, nothing to do

  hosts (0):
```

4.3. 使用 **ansible development -i inventory --list-hosts** 命令列出 development 组中列出的所有受管主机。

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

```
[student@workstation deploy-inventory]$ ansible development -i inventory \
> --list-hosts
  hosts (1):
    servera.lab.example.com
```

4.4. 使用 **ansible testing -i inventory --list-hosts** 命令列出 testing 组中列出的所有受管主机。

```
[student@workstation deploy-inventory]$ ansible testing -i inventory \
> --list-hosts
  hosts (1):
    serverb.lab.example.com
```

4.5. 使用 **ansible production -i inventory --list-hosts** 命令列出 production 组中列出的所有受管主机。

```
[student@workstation deploy-inventory]$ ansible production -i inventory \
> --list-hosts
  hosts (2):
    serverc.lab.example.com
    serverd.lab.example.com
```

4.6. 使用 **ansible us -i inventory --list-hosts** 命令列出 us 组中列出的所有受管主机。

```
[student@workstation deploy-inventory]$ ansible us -i inventory --list-hosts
  hosts (3):
    servera.lab.example.com
    serverb.lab.example.com
    serverc.lab.example.com
```

4.7. 建议您尝试使用其他变体来确认自定义清单文件中的受管主机条目。

完成

在 workstation 上，运行 **lab deploy-inventory finish** 脚本来清理本练习。

```
[student@workstation ~]$ lab deploy-inventory finish
```

本引导式练习到此结束。

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

管理 ANSIBLE 配置文件

培训目标

学完本章节后，您应能够描述 Ansible 配置文件的所在位置、Ansible 如何选择这些文件，并编辑这些文件以将更改应用于默认设置。

配置 ANSIBLE

可以通过修改 Ansible 配置文件中的设置来自定义 Ansible 安装的行为。Ansible 从控制节点上多个可能的位置之一选择其配置文件。

使用 /etc/ansible/ansible.cfg

ansible 软件包提供一个基本的配置文件，它位于 **/etc/ansible/ansible.cfg**。如果找不到其他配置文件，则使用此文件。

使用 ~/.ansible.cfg

Ansible 在用户的主目录中查找 **.ansible.cfg** 文件。如果存在此配置并且当前工作目录中也没有 **ansible.cfg** 文件，则使用此配置取代 **/etc/ansible/ansible.cfg**。

使用 ./ansible.cfg

如果执行 **ansible** 命令的目录中存在 **ansible.cfg** 文件，则使用它，而不使用全局文件或用户的个人文件。这样，管理员可以创建一种目录结构，将不同的环境或项目存储在单独的目录中，并且每一目录包含为独特的一组设置而定制的配置文件。

> **重要**
> 推荐的做法是在您要运行 Ansible 命令的目录中创建 **ansible.cfg** 文件。此目录中也将包含任何供您的 Ansible 项目使用的文件，如清单和 playbook。这是用于 Ansible 配置文件的最常用位置。实践中不常使用 **~/.ansible.cfg** 或 **/etc/ansible/ansible.cfg** 文件。

使用 ANSIBLE_CONFIG 环境变量

您可以通过将不同的配置文件放在不同的目录中，然后从适当目录执行 Ansible 命令，以此利用配置文件；但是，随着配置文件数量的增加，这种方法存在局限并且难以管理。有一个更加灵活的选项，即通过 ANSIBLE_CONFIG 环境变量定义配置文件的位置。定义了此变量时，Ansible 将使用变量所指定的配置文件，而不用上文中提及的任何配置文件。

配置文件优先级

配置文件的搜索顺序与上述列表相反。位于搜索顺序中的第一个文件是 Ansible 选择的文件。Ansible 仅使用它找到的第一个文件中的配置设置。

ANSIBLE_CONFIG 环境变量指定的任何文件将覆盖所有其他配置文件。如果没有设置该变量，则接下来检查运行 **ansible** 命令的目录中是否有 **ansible.cfg** 文件。如果不存在该文件，则检查用户的主目录中是否有 **.ansible.cfg** 文件。只有在找不到其他配置文件时，才使用全局 **/etc/**

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

ansible/ansible.cfg 文件。如果 **/etc/ansible/ansible.cfg** 配置文件不存在，Ansible 包含它使用的默认值。

由于 Ansible 配置文件可以放入的位置有多种，因此 Ansible 当前使用哪一个配置文件可能会令人困惑。您可以运行 **ansible --version** 命令来清楚地确认所安装的 Ansible 版本，以及正在使用的配置文件。

```
[user@controlnode ~]$ ansible --version
ansible 2.8.0
  config file = /etc/ansible/ansible.cfg
...output omitted...
```

显示活动的 Ansible 配置文件还有一种方式，那就是在命令行执行 Ansible 命令时使用 **-v** 选项。

```
[user@controlnode ~]$ ansible servers --list-hosts -v
Using /etc/ansible/ansible.cfg as config file
...output omitted...
```

Ansible 仅使用具有最高优先级的配置文件中的设置。即使存在优先级较低的其他文件，其设置也会被忽略，不会与选定配置文件中的设置结合。因此，如果您选择自行创建配置文件来取代全局 **/etc/ansible/ansible.cfg** 配置文件，您需要将该文件中所有需要的设置复制到自己的用户级配置文件中。用户级配置文件中未定义的设置将保持设为内置默认值，即使已在全局配置文件中设为某个其他值也是如此。

管理配置文件中的设置

Ansible 配置文件由几个部分组成，每一部分含有以键值对形式定义的设置。部分的标题以方括号括起。对于基本操作，请使用以下两部分：

- **[defaults]** 部分设置 Ansible 操作的默认值
- **[privilege_escalation]** 配置 Ansible 如何在受管主机上执行特权升级

例如，下面是典型的 **ansible.cfg** 文件：

```
[defaults]
inventory = ./inventory
remote_user = user
ask_pass = false

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

下表说明了此文件中的指令：

Ansible 配置

指令	描述
inventory	指定清单文件的路径。

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

指令	描述
remote_user	要在受管主机上登录的用户的名称。如果未指定，则使用当前用户的名称。
ask_pass	是否提示输入 SSH 密码。如果使用 SSH 公钥身份验证，则可以是 **false**。
become	连接后是否自动在受管主机上切换用户（通常切换为 root）。这也可以通过 play 来指定。
become_method	如何切换用户（通常为 **sudo**，这也是默认设置，但也可选择 **su**）。
become_user	要在受管主机上切换到的用户（通常是 root，这也是默认值）。
become_ask_pass	是否需要为 **become_method** 提示输入密码。默认为 **false**。

配置连接

Ansible 需要知道如何与其受管主机通信。更改配置文件的一个最常见原因是为了控制 Ansible 使用什么方法和用户来管理受管主机。需要的一些信息包括：

- 列出受管主机和主机组的清单的位置
- 要使用哪一种连接协议来与受管主机通信（默认为 SSH），以及是否需要非标准网络端口来连接服务器
- 要在受管主机上使用哪一远程用户；这可以是 root 用户或者某一非特权用户
- 如果远程用户为非特权用户，Ansible 需要知道它是否应尝试将特权升级为 root 以及如何进行升级（例如，通过使用 **sudo**）
- 是否提示输入 SSH 密码或 **sudo** 密码以进行登录或获取特权

清单位置

在 **[defaults]** 部分中，**inventory** 指令可以直接指向某一静态清单文件，或者指向含有多个静态清单文件和动态清单脚本的某一目录。

```
[defaults]
inventory = ./inventory
```

连接设置

默认情况下，Ansible 使用 SSH 协议连接受管主机。控制 Ansible 如何连接受管主机的最重要参数在 **[defaults]** 部分中设置。

默认情况下，Ansible 尝试连接受管主机时使用的用户名与运行 Ansible 命令的本地用户相同。若要指定不同的远程用户，请将 **remote_user** 参数设置为该用户名。

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

如果为运行 Ansible 的本地用户配置了 SSH 私钥，使得它们能够在受管主机上进行远程用户的身份验证，则 Ansible 将自动登录。如果不是这种情况，您可以通过设置指令 **ask_pass = true**，将 Ansible 配置为提示本地用户输入由远程用户使用的密码。

```
[defaults]
inventory = ./inventory

remote_user = root
ask_pass = true
```

假设您在使用一个 Linux 控制节点，并对受管主机使用 OpenSSH，如果您可以使用密码以远程用户身份登录，那么您或许可以设置基于 SSH 密钥的身份验证，从而能够设置 **ask_pass = false**。

第一步是确保在 **~/.ssh** 中为控制节点上的用户配置了 SSH 密钥对。您可以运行 **ssh-keygen** 命令来实现这个目标。

如果是单一现有受管主机，您可以在受管主机上安装公钥，并使用 **ssh-copy-id** 命令在本地的 **~/.ssh/known_hosts** 文件中填充其主机密钥，如下所示：

```
[user@controlnode ~]$ ssh-copy-id root@web1.example.com
The authenticity of host 'web1.example.com (192.168.122.181)' can't be
 established.
ECDSA key fingerprint is 70:9c:03:cd:de:ba:2f:11:98:fa:a0:b3:7c:40:86:4b.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
 now it is to install the new keys
root@web1.example.com's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@web1.example.com'"
and check to make sure that only the key(s) you wanted were added.
```

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

注意

您也可以通过 `authorized_key` 模块，使用 Ansible Playbook 将您的公钥部署到所有受管主机上的 `remote_user` 帐户。

本课程尚未详细论述 Ansible Playbook。确保您的公钥部署到受管主机 `root` 帐户的 play 内容可能如下所示：

```
- name: Public key is deployed to managed hosts for Ansible
  hosts: all

  tasks:
    - name: Ensure key is in root's ~/.ssh/authorized_hosts
      authorized_key:
        user: root
        state: present
        key: '{{ item }}'
      with_file:
        - ~/.ssh/id_rsa.pub
```

由于还没有为受管主机配置基于 SSH 密钥的身份验证，您必须要使用 **`ansible-playbook`** 命令及 **`--ask-pass`** 选项来运行 playbook，才让该命令进行远程用户的身份验证。

升级特权

鉴于安全性和审计原因，Ansible 可能需要先以非特权用户身份连接远程主机，然后再通过特权升级获得 `root` 用户身份的管理权限。这可以在 Ansible 配置文件的 **`[privilege_escalation]`** 部分中设置。

要默认启用特权升级，可在配置文件中设置指令 **`become = true`**。即使默认为该设置，也可以在运行临时命令或 Ansible Playbook 时通过各种方式覆盖它。（例如，有时候您可能要运行不需要特权升级的任务或 play。）

`become_method` 指令指定如何升级特权。有多个选项可用，但默认为使用 **`sudo`**。类似地，**`become_user`** 指令指定要升级到的用户，但默认为 `root`。

如果所选的 **`become_method`** 机制要求用户输入密码才能升级特权，您可以在配置文件中设置 **`become_ask_pass = true`** 指令。

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