Linux Automation with Ansible 8.0

Contents

Defining the Inventory

An *inventory* defines a collection of hosts that Ansible will manage. These hosts can also be assigned to *groups*, which can be managed collectively. Groups can contain child groups, and hosts can be members of multiple groups. The inventory can also set variables that apply to the hosts and groups that it defines.

Host inventories can be defined in two different ways. A *static* host inventory can be defined by a text file. A *dynamic* host inventory can be generated by a script or other program as needed, using external information providers.

Specifying Managed Hosts with a Static Inventory

A static inventory file is a text file that specifies the managed hosts that Ansible targets. You can write this file using a number of different formats, including INI-style or YAML. The INI-style format is very common and will be used for most examples in this course.

There are multiple static inventory formats supported by Ansible. In this section, we are focusing on the most common one, INI-style format.

In its simplest form, an INI-style static inventory file is a list of host names or IP addresses of managed hosts, each on a single line:

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

Normally, however, you organize managed hosts into *host groups*. Host groups allow you to more effectively run Ansible against a collection of systems. In this case, each section starts with a host group name enclosed in square brackets ([]). This is followed by the host name or an IP address for each managed host in the group, each on a single line.

In the following example, the host inventory defines two host groups: `webservers` and `db-servers`.

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

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

Hosts can be in multiple groups. In fact, recommended practice is to organize your hosts into multiple groups, possibly organized in different ways depending on the role of the host, its physical location, whether it is in production or not, and so on. This allows you to easily apply Ansible plays to specific hosts.

```
[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
```

Two host groups always exist:

- The `all` host group contains every host explicitly listed in the inventory.
- The `ungrouped` host group contains every host explicitly listed in the inventory that is not a member of any other group.

Defining Nested Groups

Ansible host inventories can include groups of host groups. This is accomplished by creating a host group name with the `:children` suffix. The following example creates a new group called `north-america`, which includes all hosts from the `usa` and `canada` groups.

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

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

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

A group can have both managed hosts and child groups as members. For example, in the previous inventory you could add a `[north-america]` section that has its own list of managed hosts. That list of hosts would be merged with the additional hosts that the `north-america` group inherits from its child groups.

Simplifying Host Specifications with Ranges

You can specify ranges in the host names or IP addresses to simplify Ansible host inventories. You can specify either numeric or alphabetic ranges. Ranges have the following syntax:

```
[START:END]
```

Ranges match all values from *`START`* to *`END`*, inclusively. Consider the following examples:

- 192.168.[4:7].[0:255] matches all IPv4 addresses in the 192.168.4.0/22 network (192.168.4.0 through 192.168.7.255).
- server[01:20].example.com matches all hosts named server01.example.com through server20.example.com.
- [a:c].dns.example.com matches hosts named a.dns.example.com, b.dns.example.com, and c.dns.example.com.
- 2001:db8::[a:f] matches all IPv6 addresses from 2001:db8::a through 2001:db8::f.

If leading zeros are included in numeric ranges, they are used in the pattern. The second example above does not match `server1.example.com` but does match `server07.example.com`. To illustrate this, the following example uses ranges to simplify the `[usa]` and `[canada]` group definitions from the earlier example:

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

[canada]
```

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

Verifying the Inventory

When in doubt, use the **ansible** command to verify a machine's presence in the inventory:

```
[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):
```

You can run the following command to list all hosts in a group:

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

If the inventory contains a host and a host group with the same name, the **ansible** command prints a warning and targets the host. The host group is ignored.

There are various ways to deal with this situation, the easiest being to ensure that host groups do not use the same names as hosts in the inventory.

Overriding the Location of the Inventory

The `/etc/ansible/hosts` file is considered the system's default static inventory file. However, normal practice is not to use that file but to define a different location for inventory files in your Ansible configuration file. This is covered in the next section.

The **ansible** and **ansible-playbook** commands that you use to run Ansible ad hoc commands and playbooks later in the course can also specify the location of an inventory file on the command line with the `--inventory` *`PATHNAME`* or `-i` *`PATHNAME`* option, where `PATHNAME` is the path to the desired inventory file.

Defining Variables in the Inventory

Values for variables used by playbooks can be specified in host inventory files. These variables only apply to specific hosts or host groups. Normally it is better to define these *inventory variables* in special directories and not directly in the inventory file. This topic is discussed in more depth elsewhere in the course.

Describing a Dynamic Inventory

Ansible inventory information can also be dynamically generated, using information provided by external databases. The open source community has written a number of dynamic inventory scripts that are available from the upstream Ansible project. If those scripts do not meet your needs, you can also write your own.

For example, a dynamic inventory program could contact your RedHat Satellite server or Amazon EC2 account, and use information stored there to construct an Ansible inventory. Because the program does this when you run Ansible, it can populate the inventory with up-to-date information provided by the service as new hosts are added and old hosts are removed.

Source: https://rha.ole.redhat.com/rha/app/courses/rh294-8.0/pages/ch02