Red Hat Enterprise Linux Automation with Ansible

FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

Guided Exercise: Selecting Hosts with Host Patterns ☆

In this exercise, you will explore how to use host patterns to specify hosts from the inventory for plays or ad hoc commands. You will be provided with several example inventories to explore host patterns.

Outcomes

You will be able to use different host patterns to access various hosts in an inventory.

Log in to `workstation` as `student` using `student` as the password.

On `workstation`, run the `lab projects-host start` command. The script creates the `projects-host` project directory, and then downloads the Ansible configuration file and the host inventory file needed for this exercise.

```
[student@workstation ~]$ lab projects-host start
```

Procedure 6.1. Instructions

1. On `workstation`, change to the working directory for the exercise, `/home/student/projects-host`, and review the contents of the directory.

   ```
   [student@workstation ~]$ cd ~/projects-host
   [student@workstation projects-host]$
   ```

   1.1. List the contents of the directory.

   ```
   [student@workstation projects-host]$ ls
   ansible.cfg inventory1 inventory2 playbook.yml
   ```

   1.2. Inspect the example inventory file, `inventory1`. Notice how the inventory is organized. Explore which hosts and groups are in the inventory, and which domains are used.

```
srv1.example.com
srv2.example.com
s1.lab.example.com
s2.lab.example.com

[web]
jupiter.lab.example.com
saturn.example.com

[db]
db1.example.com
db2.example.com
db3.example.com

[lb]
lb1.lab.example.com
lb2.lab.example.com

[boston]
db1.example.com
jupiter.lab.example.com
lb2.lab.example.com

[london]
db2.example.com
db3.example.com
file1.lab.example.com
lb1.lab.example.com

[dev]
web1.lab.example.com
db3.example.com

[stage]
file2.example.com
db2.example.com

[prod]
lb2.lab.example.com
db1.example.com
jupiter.lab.example.com

[function:children]
web
db
lb
city

[city:children]
boston
london
environments

[environments:children]
dev
stage
prod
new

[new]
172.25.252.23
172.25.252.44
172.25.252.32
```

1.3. Inspect the example inventory file, `inventory2`. Notice how the inventory is organized. Explore which hosts and groups are in the inventory, and which domains are used.

```
workstation.lab.example.com

[london]
servera.lab.example.com

[berlin]
serverb.lab.example.com

[tokyo]
serverc.lab.example.com

[atlanta]
serverd.lab.example.com

[europe:children]
london
berlin
```

1.4. Lastly, inspect the contents of the playbook, `playbook.yml`. Notice how the playbook uses the `debug` module to display the name of each managed host.

```
---
- name: Resolve host patterns
  hosts:
  tasks:
    - name: Display managed host name
      debug:
        msg: "{{ inventory_hostname }}"
```

2. Using an ad hoc command, determine if the `db1.example.com` server is present in the `inventory1` inventory file.

```
[student@workstation projects-host]$ ansible db1.example.com -i inventory1 \
> --list-hosts
  hosts (1):
    db1.example.com
```

3. Using an ad hoc command, reference an IP address contained in the `inventory1` inventory with a host pattern.

```
[student@workstation projects-host]$ ansible 172.25.252.44 -i inventory1 \
> --list-hosts
 hosts (1):
    172.25.252.44
```

4. With an ad hoc command, use the `all` group to list all managed hosts in the `inventory1` inventory file.

```
[student@workstation projects-host]$ ansible all -i inventory1 --list-hosts
  hosts (17):
    srv1.example.com
    srv2.example.com
    s1.lab.example.com
    s2.lab.example.com
    jupiter.lab.example.com
    saturn.example.com
    db1.example.com
    db2.example.com
    db3.example.com
    lb1.lab.example.com
    lb2.lab.example.com
    file1.lab.example.com
    web1.lab.example.com
    file2.example.com
    172.25.252.23
    172.25.252.44
    172.25.252.32
```

5. With an ad hoc command, use the asterisk (*) character to list all hosts that end in `.example.com` in the `inventory1` inventory file.

```
[student@workstation projects-host]$ ansible '*.example.com' -i inventory1 \
> --list-hosts
  hosts (14):
    jupiter.lab.example.com
    saturn.example.com
    db1.example.com
    db2.example.com
    db3.example.com
    lb1.lab.example.com
    lb2.lab.example.com
    file1.lab.example.com
    web1.lab.example.com
    file2.example.com
    srv1.example.com
    srv2.example.com
    s1.lab.example.com
    s2.lab.example.com
```

6. As you can see in the output of the preceeding command, there are 14 hosts in the `*.example.com` domain. Modify the host pattern in the previous ad hoc command so that hosts in the `*.lab.example.com` domain are ignored.

```
[student@workstation projects-host]$ ansible '*.example.com, !*.lab.example.com' \
> -i inventory1 --list-hosts
  hosts (7):
    saturn.example.com
    db1.example.com
    db2.example.com
    db3.example.com
    file2.example.com
    srv1.example.com
    srv2.example.com
```

7. Without accessing the groups in the `inventory1` inventory file, use an ad hoc command to list these three hosts: `lb1.lab.example.com`, `s1.lab.example.com`, and `db1.example.com`.

```
[student@workstation projects-host]$ ansible \
> lb1.lab.example.com,s1.lab.example.com,db1.example.com -i inventory1 \
> --list-hosts
  hosts (3):
    lb1.lab.example.com
    s1.lab.example.com
    db1.example.com
```

8. Use a wildcard host pattern in an ad hoc command to list hosts that start with a `172.25.` IP address in the `inventory1` inventory file.

```
[student@workstation projects-host]$ ansible '172.25.*' -i inventory1 --list-hosts
  hosts (3):
    172.25.252.23
    172.25.252.44
    172.25.252.32
```

9. Use a host pattern in an ad hoc command to list all hosts in the `inventory1` inventory file that start with the letter "s."

```
[student@workstation projects-host]$ ansible 's*' -i inventory1 --list-hosts
  hosts (7):
    saturn.example.com
    srv1.example.com
    srv2.example.com
    s1.lab.example.com
    s2.lab.example.com
    file2.example.com
    db2.example.com
```

Notice the `file2.example.com` and `db2.example.com` hosts in the output of the preceding command. They appear in the list because they are both members of a group called `stage`, which also begins with the letter "s."

0. Using a list and wildcard host patterns in an ad hoc command, list all hosts in the `inventory1` inventory in the `prod` group, those hosts with an IP address beginning with `172`, and hosts that contain `lab` in their name.

```
[student@workstation projects-host]$ ansible 'prod,172*,*lab*' -i inventory1 \
> --list-hosts
  hosts (11):
    lb2.lab.example.com
    db1.example.com
    jupiter.lab.example.com
    172.25.252.23
    172.25.252.44
    172.25.252.32
    lb1.lab.example.com
    file1.lab.example.com
    web1.lab.example.com
    s1.lab.example.com
    s2.lab.example.com
```

11. Use an ad hoc command to list all hosts that belong to both the `db` and `london` groups.

```
[student@workstation projects-host]$ ansible 'db,&london' -i inventory1 \
> --list-hosts
  hosts (2):
    db2.example.com
    db3.example.com
```

2. Modify the `hosts` value in the `playbook.yml` file so that all servers in the `london` group are targeted. Execute the playbook using the `inventory2` inventory file.

```
...output omitted...
  hosts: london
...output omitted...
```

```
[student@workstation projects-host]$ ansible-playbook -i inventory2 playbook.yml
...output omitted...
TASK [Gathering Facts] *********************************************
ok: [servera.lab.example.com]
...output omitted...
```

3. Modify the `hosts` value in the `playbook.yml` file so that all servers in the `europe` nested group are targeted. Execute the playbook using the `inventory2` inventory file.

```
...output omitted...
  hosts: europe
...output omitted...
```

```
[student@workstation projects-host]$ ansible-playbook -i inventory2 playbook.yml
...output omitted...
TASK [Gathering Facts] *********************************************
ok: [servera.lab.example.com]
ok: [serverb.lab.example.com]
...output omitted...
```

4. Modify the `hosts` value in the `playbook.yml` file so that all servers that do not belong to any group are targeted. Execute the playbook using the `inventory2` inventory file.

```
...output omitted...
  hosts: ungrouped
...output omitted...
```

```
[student@workstation projects-hosts]$ ansible-playbook -i inventory2 playbook.yml
...output omitted...
TASK [Gathering Facts] *********************************************
ok: [workstation.lab.example.com]
...output omitted...
```

Finish

On `workstation`, run the `lab projects-host finish` script to clean up this exercise.

```
[student@workstation ~]$ lab projects-host finish
```

This concludes the guided exercise.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH06) → NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH06S03)