Red Hat Enterprise Linux Automation with Ansible

DOWNLOAD EBOOK FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

P (/rol/app/courses/rh294-8.4/pages/pr01) (/rol/app/courses/rh294-8.4/pages/pr01)
(/rol/app/courses/rh294-8.4/pages/pr01s02)
1 (/rol/app/courses/rh294-8.4/pages/ch01) (/rol/app/courses/rh294-8.4/pages/ch01)
(/rol/app/courses/rh294-8.4/pages/ch01s02)
(/rol/app/courses/rh294-8.4/pages/ch01s03)
(/rol/app/courses/rh294-8.4/pages/ch01s04)
(/rol/app/courses/rh294-8.4/pages/ch01s05)
2 (/rol/app/courses/rh294-8.4/pages/ch02) (/rol/app/courses/rh294-8.4/pages/ch02)
(/rol/app/courses/rh294-8.4/pages/ch02s02)
(/rol/app/courses/rh294-8.4/pages/ch02s03)
(/rol/app/courses/rh294-8.4/pages/ch02s04)
(/rol/app/courses/rh294-8.4/pages/ch02s05)
(/rol/app/courses/rh294-8.4/pages/ch02s06)
(/rol/app/courses/rh294-8.4/pages/ch02s07)
(/rol/app/courses/rh294-8.4/pages/ch02s08)
(/rol/app/courses/rh294-8.4/pages/ch02s09)
(/rol/app/courses/rh294-8.4/pages/ch02s10)
(/rol/app/courses/rh294-8.4/pages/ch02s11)
(/rol/app/courses/rh294-8.4/pages/ch02s12)
3 (/rol/app/courses/rh294-8.4/pages/ch03) (/rol/app/courses/rh294-8.4/pages/ch03)
(/rol/app/courses/rh294-8.4/pages/ch03s02)
(/rol/app/courses/rh294-8.4/pages/ch03s03)
(/rol/app/courses/rh294-8.4/pages/ch03s04)
(/rol/app/courses/rh294-8.4/pages/ch03s05)
(/rol/app/courses/rh294-8.4/pages/ch03s06)
(/rol/app/courses/rh294-8.4/pages/ch03s07)
(/rol/app/courses/rh294-8.4/pages/ch03s08)
4 (/rol/a
8.4/pa
(/rol/app/courses/rh294-8.4/pages/ch04)
(/rol/app/courses/rh294-8.4/pages/ch04s02)
(/rol/app/courses/rh294-8.4/pages/ch04s03)
(/rol/app/courses/rh294-8.4/pages/ch04s04)
(/rol/app/courses/rh294-8.4/pages/ch04s05)
(/rol/app/courses/rh294-8.4/pages/ch04s06)
(/rol/app/courses/rh294-8.4/pages/ch04s07)
(/rol/app/courses/rh294-8.4/pages/ch04s08)
5 (/rol/app/courses/rh294-8.4/pages/ch05) (/rol/app/courses/rh294-8.4/pages/ch05)
(/rol/app/courses/rh294-8.4/pages/ch05s02)
(/rol/app/courses/rh294-8.4/pages/ch05s03)
(/rol/app/courses/rh294-8.4/pages/ch05s04)
(/rol/app/courses/rh294-8.4/pages/ch05s05)
(/rol/app/courses/rh294-8.4/pages/ch05s06)
6 (/rol/app/courses/rh294-8.4/pages/ch06) (/rol/app/courses/rh294-8.4/pages/ch06)
(/rol/app/courses/rh294-8.4/pages/ch06s02)
(/rol/app/courses/rh294-8.4/pages/ch06s03)
(/rol/app/courses/rh294-8.4/pages/ch06s04)
(/rol/app/courses/rh294-8.4/pages/ch06s05)
(/rol/app/courses/rh294-8.4/pages/ch06s06)
7 (/rol/app/courses/rh294-8.4/pages/ch07) (/rol/app/courses/rh294-8.4/pages/ch07)
(/rol/app/courses/rh294-8.4/pages/ch07s02)
(/rol/app/courses/rh294-8.4/pages/ch07s03)
(/rol/app/courses/rh294-8.4/pages/ch07s04)
(/rol/app/courses/rh294-8.4/pages/ch07s05)
(/rol/app/courses/rh294-8.4/pages/ch07s06)
(/rol/app/courses/rh294-8.4/pages/ch07s07)
(/rol/app/courses/rh294-8.4/pages/ch07s08)
(/rol/app/courses/rh294-8.4/pages/ch07s09)
(/rol/app/courses/rh294-8.4/pages/ch07s10)
(/rol/app/courses/rh294-8.4/pages/ch07s11)
(/rol/app/courses/rh294-8.4/pages/ch07s12)
8 (/rol/app/courses/rh294-8.4/pages/ch08) (/rol/app/courses/rh294-8.4/pages/ch08)
(/rol/app/courses/rh294-8.4/pages/ch08s02)
(/rol/app/courses/rh294-8.4/pages/ch08s03)
(/rol/app/courses/rh294-8.4/pages/ch08s04)
(/rol/app/courses/rh294-8.4/pages/ch08s05)
(/rol/app/courses/rh294-8.4/pages/ch08s06)
9 (/rol/app/courses/rh294-8.4/pages/ch09) (/rol/app/courses/rh294-8.4/pages/ch09)
(/rol/app/courses/rh294-8.4/pages/ch09s02)
(/rol/app/courses/rh294-8.4/pages/ch09s03)
(/rol/app/courses/rh294-8.4/pages/ch09s04)
(/rol/app/courses/rh294-8.4/pages/ch09s05)
(/rol/app/courses/rh294-8.4/pages/ch09s06)
(/rol/app/courses/rh294-8.4/pages/ch09s07)
(/rol/app/courses/rh294-8.4/pages/ch09s08)
(/rol/app/courses/rh294-8.4/pages/ch09s09)
(/rol/app/courses/rh294-8.4/pages/ch09s10)
(/rol/app/courses/rh294-8.4/pages/ch09s11)
(/rol/app/courses/rh294-8.4/pages/ch09s12)
10 (/rol/app/courses/rh294-8.4/pages/ch10) (/rol/app/courses/rh294-8.4/pages/ch10)
(/rol/app/courses/rh294-8.4/pages/ch10s02)
(/rol/app/courses/rh294-8.4/pages/ch10s03)
(/rol/app/courses/rh294-8.4/pages/ch10s04)
A (/rol/app/courses/rh294-8.4/pages/apa) (/rol/app/courses/rh294-8.4/pages/apa)
A (/rol/app/courses/rh294-8.4/pages/apb) (/rol/app/courses/rh294-8.4/pages/apb)

PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH09)

NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH09S03)

VIDEO CLASSROOM

Guided Exercise: Managing Software and Subscriptions

In this exercise you will configure a new Yum repository and install packages from it on your managed hosts.

Outcomes

You should be able to:

Configure a yum repository using the `yum_repository` module.

Manage RPM GPG keys using the `rpm_key` module.

Obtain information about the installed packages on a host using the `package_facts` module.

On `workstation`, run the lab start script to confirm that the environment is ready for the lab to begin. The script creates the working directory, called `system-software`, and populates it with an Ansible configuration file, a host inventory, and lab files.

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

Procedure 9.1. Instructions

Your organization requires that all hosts have the example-motd package installed. This package is provided by an internal Yum repository maintained by your organization to host internally developed software packages.

You are tasked with writing a playbook to ensure that the example-motd package is installed on the remote host. The playbook must ensure the configuration of the internal Yum repository.

The repository is located at `http://materials.example.com/yum/repository`. All RPM packages are signed with an organizational GPG key pair. The GPG public key is available at `http://materials.example.com/yum/repository/RPM-GPG-KEY-example`.

As the `student` user on `workstation`, change to the `/home/student/system-software` working directory.

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

Begin writing the `repo_playbook.yml` playbook. Define a single play in the playbook that targets all hosts. Add a `vars` clause that defines a single variable `custom_pkg` with a value of `example-motd`. Add the `tasks` clause to to the playbook.

The playbook now contains:

```
---
- name: Repository Configuration
  hosts: all
  vars:
    custom_pkg: example-motd
  tasks:
```

Add two tasks to the playbook.

Use the `package_facts` module in the first task to gather information about installed packages on the remote host. This task populates the `ansible_facts.packages` fact.

Use the `debug` module in the second task to print the installed version of the package referenced by the `custom_pkg` variable. Only execute this task if the custom package is found in the `ansible_facts.packages` fact.

Execute the `repo_playbook.yml` playbook.

3.1. Add the first task to the playbook. Configure the `manager` keyword of the `package_facts` module with a value of `auto`. The first task contains the following:

```
    - name: Gather Package Facts
      package_facts:
        manager: auto
```

3.2. Add a second task to the playbook that uses the `debug` module to display the value of the `ansible_facts.packages[custom_pkg]` variable. Add a `when` clause to the task to check if the value of the `custom_pkg` variable is contained in the `ansible_facts.packages` variable. The second task contains the following:

```
    - name: Show Package Facts for the custom package
      debug:
        var: ansible_facts.packages[custom_pkg]
      when: custom_pkg in ansible_facts.packages
```

3.3. Execute the playbook:

```
[student@workstation system-software]$ ansible-playbook repo_playbook.yml

PLAY [Repository Configuration] **********************************************

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

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

TASK [Show Package Facts for the custom package] *****************************
skipping: [servera.lab.example.com]

PLAY RECAP *******************************************************************
servera.lab.example.com    : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

The debug task is skipped because the example-motd package is not installed on the remote host.

Add a third task that uses the `yum_repository` module to ensure the configuration of the internal yum repository on the remote host. Ensure that:

The repository's configuration is stored in the file `/etc/yum.repos.d/example.repo`

The repository ID is `example-internal`

The base URL is `http://materials.example.com/yum/repository`

The repository is configured to check RPM GPG signatures

The repository description is `Example Inc. Internal YUM repo`

The third task contains the following:

```
    - name: Ensure Example Repo exists
      yum_repository:
        name: example-internal
        description: Example Inc. Internal YUM repo
        file: example
        baseurl: http://materials.example.com/yum/repository/
        gpgcheck: yes
```

Add a fourth task to the play that uses the `rpm_key` module to ensure that the repository public key is present on the remote host. The repository public key URL is `http://materials.example.com/yum/repository/RPM-GPG-KEY-example`.

The fourth task appears as follows:

```
    - name: Ensure Repo RPM Key is Installed
      rpm_key:
        key: http://materials.example.com/yum/repository/RPM-GPG-KEY-example
        state: present
```

Add a fifth task to ensure that the package referenced by the `custom_pkg` variable is installed on the remote host.

The fifth task appears as follows:

```
    - name: Install Example motd package
      yum:
        name: "{{ custom_pkg }}"
        state: present
```

The `ansible_facts.packages` fact is not updated when a new package is installed on a remote host.

Copy the second task and add it as the sixth task in the play. Execute the playbook and verify that the `ansible_facts.packages` fact does not contain information about the example-motd installed on the remote host.

7.1. The sixth task contains a copy of the second task:

```
    - name: Show Package Facts for the custom package
      debug:
        var: ansible_facts.packages[custom_pkg]
      when: custom_pkg in ansible_facts.packages
```

The entire playbook now looks as follows:

```
---
- name: Repository Configuration
  hosts: all
  vars:
    custom_pkg: example-motd
  tasks:
    - name: Gather Package Facts
      package_facts:
        manager: auto

    - name: Show Package Facts for the custom package
      debug:
        var: ansible_facts.packages[custom_pkg]
      when: custom_pkg in ansible_facts.packages

    - name: Ensure Example Repo exists
      yum_repository:
        name: example-internal
        description: Example Inc. Internal YUM repo
        file: example
        baseurl: http://materials.example.com/yum/repository/
        gpgcheck: yes

    - name: Ensure Repo RPM Key is Installed
      rpm_key:
        key: http://materials.example.com/yum/repository/RPM-GPG-KEY-example
        state: present

    - name: Install Example motd package
      yum:
        name: "{{ custom_pkg }}"
        state: present

    - name: Show Package Facts for the custom package
      debug:
        var: ansible_facts.packages[custom_pkg]
      when: custom_pkg in ansible_facts.packages
```

7.2. Execute the playbook.

```
[student@workstation system-software]$ ansible-playbook repo_playbook.yml
PLAY [Repository Configuration] **********************************************

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

TASK [Gather Package Facts] **************************************************
ok: [servera.lab.example.com]❶

TASK [Show Package Facts for the custom package] *****************************
skipping: [servera.lab.example.com]

TASK [Ensure Example Repo exists] ********************************************
changed: [servera.lab.example.com]

TASK [Ensure Repo RPM Key is Installed] **************************************
changed: [servera.lab.example.com]

TASK [Install Example motd package] ******************************************
changed: [servera.lab.example.com]

TASK [Show Package Facts for the custom package] *****************************
skipping: [servera.lab.example.com]❷

PLAY RECAP *******************************************************************
servera.lab.example.com    : ok=5    changed=3    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

❶ The `Gather Package Facts` task determines the data contained in the `ansible_facts.packages` fact.

❷ The task is skipped because the example-motd package is installed after the `Gather Package Facts` task.

Insert a task immediately after the `Install Example motd package` task using the `package_facts` module to update the package facts. Set the module's `manager` keyword with a value of `auto`.

The complete playbook is shown below:

```
---
- name: Repository Configuration
  hosts: all
  vars:
    custom_pkg: example-motd
  tasks:
    - name: Gather Package Facts
      package_facts:
        manager: auto

    - name: Show Package Facts for the custom package
      debug:
        var: ansible_facts.packages[custom_pkg]
      when: custom_pkg in ansible_facts.packages

    - name: Ensure Example Repo exists
      yum_repository:
        name: example-internal
        description: Example Inc. Internal YUM repo
        file: example
        baseurl: http://materials.example.com/yum/repository/
        gpgcheck: yes

    - name: Ensure Repo RPM Key is Installed
      rpm_key:
        key: http://materials.example.com/yum/repository/RPM-GPG-KEY-example
        state: present

    - name: Install Example motd package
      yum:
        name: "{{ custom_pkg }}"
        state: present

    - name: Gather Package Facts
      package_facts:
        manager: auto

    - name: Show Package Facts for the custom package
      debug:
        var: ansible_facts.packages[custom_pkg]
      when: custom_pkg in ansible_facts.packages
```

Use an Ansible ad hoc command to remove the example-motd package installed during the previous execution of the playbook. Execute the playbook with the inserted `package_facts` task and use the output to verify that the installation of the example-motd package.

9.1. To remove the example-motd package from all hosts, use the `ansible all` command with the `-m yum` and `-a 'name=example-motd state=absent'` options.

```
[student@workstation system-software]$ ansible all -m yum \
> -a 'name=example-motd state=absent'
servera.lab.example.com | CHANGED => {
...output omitted...
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Removed: example-motd-1.0-1.el7.x86_64"
    ]
...output omitted...
```

9.2. Execute the playbook.

```
[student@workstation system-software]$ ansible-playbook repo_playbook.yml

PLAY [Repository Configuration] ***********************************************

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

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

TASK [Show Package Facts for the custom package] *****************************
skipping: [servera.lab.example.com]❶

...output omitted...

TASK [Install Example motd package] ******************************************
changed: [servera.lab.example.com]❷

TASK [Gather Package Facts] ***************************************************
ok: [servera.lab.example.com]❸

TASK [Show Package Facts for example-motd] ***********************************
ok: [servera.lab.example.com] => {
    "ansible_facts.packages[custom_pkg]": [❹
        {
            "arch": "x86_64",
            "epoch": null,
            "name": "example-motd",
            "release": "1.el7",
            "source": "rpm",
            "version": "1.0"
        }
    ]
}

PLAY RECAP ********************************************************************
servera.lab.example.com    : ok=7    changed=1    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

❶ No package fact exists for the example-motd package because the package is not installed on the remote host.

❷ The example-motd package is installed as a result of this task, as indicated by the `changed` status.

❸ This task updates the package facts with information about the example-motd package.

❹ The example-motd package fact exists and indicates only one example-motd package is installed. The installed package is at version `1.0`.

Finish

On `workstation`, run the `lab system-software finish` script to clean up the resources created in this exercise.

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

This concludes the guided exercise.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH09)

→ NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH09S03)