



(/rol/app/)

[Home\(/rol/app/\)](#)[Reports\(/rol/app/reports\)](#)[Community\(https://learn.redhat.com/\)](#)Days remaining **76**

Search

# Red Hat Enterprise Linux Automation with Ansible

FEEDBACK

TRANSLATIONS ▾

CERTIFICATE OF ATTENDANCE



P	(/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/ch01s02)	(/rol/app/courses/rh294-8.4/pages/ch01s03)		(/rol/app/courses/rh294-8.4/pages/ch01s04)
8.4/pages/pr01)	(/rol/app/courses/rh294-8.4/pages/ch01s05)	2	(/rol/app/courses/rh294-8.4/pages/ch02)	3.4/pages/ch01)
	(/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)	3.4/pages/ch02)		(/rol/app/courses/rh294-8.4/pages/ch02s08)
	(/rol/app/courses/rh294-8.4/pages/ch02s09)	(/rol/app/courses/rh294-8.4/pages/ch02s07)		(/rol/app/courses/rh294-8.4/pages/ch02s11)
	(/rol/app/courses/rh294-8.4/pages/ch02s12)	(/rol/app/courses/rh294-8.4/pages/ch02s10)		(/rol/app/courses/rh294-8.4/pages/ch03s02)
	(/rol/app/courses/rh294-8.4/pages/ch03s03)	3	(/rol/app/courses/rh294-8.4/pages/ch03)	(/rol/app/courses/rh294-8.4/pages/ch03s05)
	(/rol/app/courses/rh294-8.4/pages/ch03s06)	(/rol/app/courses/rh294-8.4/pages/ch03s04)		(/rol/app/courses/rh294-8.4/pages/ch03s08)
	(/rol/app/courses/rh294-8.4/pages/ch04)	8.4/pages/ch03)		4
	(/rol/app/courses/rh294-8.4/pages/ch04s04)	(/rol/app/courses/rh294-8.4/pages/ch03s07)		(/rol/app/courses/rh294-8.4/pages/ch04s03)
	(/rol/app/courses/rh294-8.4/pages/ch04s07)	(/rol/app/courses/rh294-8.4/pages/ch04s02)		(/rol/app/courses/rh294-8.4/pages/ch04s06)
	(/rol/app/courses/rh294-8.4/pages/ch05s02)	(/rol/app/courses/rh294-8.4/pages/ch04s05)		(/rol/app/courses/rh294-8.4/pages/ch04s06)
	(/rol/app/courses/rh294-8.4/pages/ch05s05)	(/rol/app/courses/rh294-8.4/pages/ch04s08)	5	(/rol/app/courses/rh294-8.4/pages/ch05)
	(/rol/app/courses/rh294-8.4/pages/ch06s02)	(/rol/app/courses/rh294-8.4/pages/ch05s03)		(/rol/app/courses/rh294-8.4/pages/ch05s04)
	(/rol/app/courses/rh294-8.4/pages/ch06s05)	(/rol/app/courses/rh294-8.4/pages/ch05s06)		6.4/pages/ch05)
	(/rol/app/courses/rh294-8.4/pages/ch07s02)	(/rol/app/courses/rh294-8.4/pages/ch06s03)		(/rol/app/courses/rh294-8.4/pages/ch06)
	(/rol/app/courses/rh294-8.4/pages/ch07s05)	(/rol/app/courses/rh294-8.4/pages/ch06s06)		(/rol/app/courses/rh294-8.4/pages/ch06s04)
	(/rol/app/courses/rh294-8.4/pages/ch07s08)	(/rol/app/courses/rh294-8.4/pages/ch07s03)	8.4/pages/ch06)	(/rol/app/courses/rh294-8.4/pages/ch07)
	(/rol/app/courses/rh294-8.4/pages/ch07s11)	(/rol/app/courses/rh294-8.4/pages/ch07s06)		(/rol/app/courses/rh294-8.4/pages/ch07s04)
	(/rol/app/courses/rh294-8.4/pages/ch08s02)	(/rol/app/courses/rh294-8.4/pages/ch07s09)	8.4/pages/ch07)	(/rol/app/courses/rh294-8.4/pages/ch07s07)
	(/rol/app/courses/rh294-8.4/pages/ch08s05)	(/rol/app/courses/rh294-8.4/pages/ch07s12)		(/rol/app/courses/rh294-8.4/pages/ch07s10)
	(/rol/app/courses/rh294-8.4/pages/ch09s02)	(/rol/app/courses/rh294-8.4/pages/ch08s03)	8	(/rol/app/courses/rh294-8.4/pages/ch08)
	(/rol/app/courses/rh294-8.4/pages/ch09s05)	(/rol/app/courses/rh294-8.4/pages/ch08s06)		(/rol/app/courses/rh294-8.4/pages/ch08s04)
	(/rol/app/courses/rh294-8.4/pages/ch09s08)	(/rol/app/courses/rh294-8.4/pages/ch09s03)	8.4/pages/ch08)	(/rol/app/courses/rh294-8.4/pages/ch09)
	(/rol/app/courses/rh294-8.4/pages/ch09s11)	(/rol/app/courses/rh294-8.4/pages/ch09s06)		(/rol/app/courses/rh294-8.4/pages/ch09s04)
	(/rol/app/courses/rh294-8.4/pages/ch10s02)	(/rol/app/courses/rh294-8.4/pages/ch09s09)	8.4/pages/ch09)	(/rol/app/courses/rh294-8.4/pages/ch09s07)
A (/rol/app/courses/rh294-8.4/pages/apa)	(/rol/app/courses/rh294-8.4/pages/ch10s03)	(/rol/app/courses/rh294-8.4/pages/ch09s12)		(/rol/app/courses/rh294-8.4/pages/ch09s10)
	(/rol/app/courses/rh294-8.4/pages/apa)	(/rol/app/courses/rh294-8.4/pages/ch10s04)	10	(/rol/app/courses/rh294-8.4/pages/ch10)
	(/rol/app/courses/rh294-8.4/pages/apb)	(/rol/app/courses/rh294-8.4/pages/apb)		(/rol/app/courses/rh294-8.4/pages/ch10s04)
			8.4/pages/ch10)	(/rol/app/courses/rh294-8.4/pages/apb)

[← PREVIOUS \(/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S09\)](#) [→ NEXT \(/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S11\)](#)

VIDEO CLASSROOM

## Guided Exercise: Implementing Multiple Plays



In this exercise, you will create a playbook containing multiple plays, then use it to perform configuration tasks on managed hosts.

### Outcomes

You should be able to construct and execute a playbook to manage configuration and perform administration of a managed host.

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

On workstation, run the `lab playbook-multi start` command. This function ensures that the managed host, `servera.lab.example.com`, is reachable on the network. It also ensures that the correct Ansible configuration file and inventory file are installed on the control node.

```
[student@workstation ~]$ lab playbook-multi start
```

### Procedure 2.5. Instructions

1. A working directory, `/home/student/playbook-multi`, has been created on workstation for the Ansible project. The directory has already been populated with an `ansible.cfg` configuration file and an inventory file, `inventory`. The managed host, `servera.lab.example.com`, is already defined in this inventory file. Create a new playbook, `/home/student/playbook-multi/intranet.yml`, and add the lines needed to start the first play. It should target the managed host `servera.lab.example.com` and enable privilege escalation.

- 1.1. Change directory into the `/home/student/playbook-multi` working directory.

```
[student@workstation ~]$ cd ~/playbook-multi  
[student@workstation playbook-multi]$
```

- 1.2. Create and open a new playbook, `/home/student/playbook-multi/intranet.yml`, and add a line consisting of three dashes to the beginning of the file to indicate the start of the YAML file.

```
---
```

- 1.3. Add the following line to the `/home/student/playbook-multi/intranet.yml` file to denote the start of a play with a name of `Enable intranet services`.

```
- name: Enable intranet services
```

- 1.4. Add the following line to indicate that the play applies to the `servera.lab.example.com` managed host. Be sure to indent the line with two spaces (aligning with the `name` keyword above it) to indicate that it is part of the first play.

```
hosts: servera.lab.example.com
```

- 1.5. Add the following line to enable privilege escalation. Be sure to indent the line with two spaces (aligning with the keywords above it) to indicate it is part of the first play.

```
become: yes
```

- 1.6. Add the following line to define the beginning of the `tasks` list. Indent the line with two spaces (aligning with the keywords above it) to indicate that it is part of the first play.

```
tasks:
```

2. As the first task in the first play, define a task that ensures that the `httpd` and `firewalld` packages are up to date.

Be sure to indent the first line of the task with four spaces. Under the `tasks` keyword in the first play, add the following lines.

```
- name: latest version of httpd and firewalld installed
  yum:
    name:
      - httpd
      - firewalld
    state: latest
```

The first line provides a descriptive name for the task. The second line is indented with six spaces and calls the `yum` module. The next line is indented eight spaces and is a `name` keyword. It specifies which packages the `yum` module should ensure are up-to-date. The `yum` module's `name` keyword (which is different from the task name) can take a list of packages, which is indented ten spaces on the two following lines. After the list, the 8-space indented `state` keyword specifies that the `yum` module should ensure that the latest version of the packages is installed.

3. Add a task to the first play's list that ensures that the correct content is in `/var/www/html/index.html`.

Add the following lines to define the content for `/var/www/html/index.html`. Be sure to indent the first line with four spaces.

```
- name: test html page is installed
  copy:
    content: "Welcome to the example.com intranet!\n"
    dest: /var/www/html/index.html
```

The first entry provides a descriptive name for the task. The second entry is indented with six spaces and calls the `copy` module. The remaining entries are indented with eight spaces and pass the necessary arguments to ensure that the correct content is in the web page.

4. Define two more tasks in the play to ensure that the `firewalld` service is running and will start on boot, and will allow connections to the `httpd` service.

- 4.1. Add the following lines to ensure that the `firewalld` service is enabled and running. Be sure to indent the first line with four spaces.

```
- name: firewalld enabled and running
  service:
    name: firewalld
    enabled: true
    state: started
```

The first entry provides a descriptive name for the task. The second entry is indented with eight spaces and calls the `service` module. The remaining entries are indented with ten spaces and pass the necessary arguments to ensure that the `firewalld` service is enabled and started.

- 4.2. Add the following lines to ensure that `firewalld` allows HTTP connections from remote systems. Be sure to indent the first line with four spaces.

```
- name: firewalld permits access to httpd service
  firewalld:
    service: http
    permanent: true
    state: enabled
    immediate: yes
```

The first entry provides a descriptive name for the task. The second entry is indented with six spaces and calls the `firewalld` module. The remaining entries are indented with eight spaces and pass the necessary arguments to ensure that remote HTTP connections are permanently allowed.

5. Add a final task to the first play that ensures that the `httpd` service is running and will start at boot.

Add the following lines to ensure that the `httpd` service is enabled and running. Be sure to indent the first line with four spaces.

```
- name: httpd enabled and running
  service:
    name: httpd
    enabled: true
    state: started
```

The first entry provides a descriptive name for the task. The second entry is indented with six spaces and calls the `service` module. The remaining entries are indented with eight spaces and pass the necessary arguments to ensure that the `httpd` service is enabled and running.

6. In `/home/student/playbook-multi/intranet.yml`, define a second play targeted at `localhost` which will test the intranet web server. It does not need privilege escalation.

- 6.1. Add the following line to define the start of a second play. Note that there is no indentation.

```
- name: Test intranet web server
```

- 6.2. Add the following line to indicate that the play applies to the `localhost` managed host. Be sure to indent the line with two spaces to indicate that it is contained by the second play.

```
  hosts: localhost
```

- 6.3. Add the following line to disable privilege escalation. Be sure to align the indentation with the `hosts` keyword above it.

```
    become: no
```

- 6.4. Add the following line to the `/home/student/playbook-multi/intranet.yml` file to define the beginning of the tasks list. Be sure to indent the line with two spaces to indicate that it is contained by the second play.

```
  tasks:
```

7. Add a single task to the second play, and use the `uri` module to request content from `http://servera.lab.example.com`. The task should verify a return HTTP status code of `200`. Configure the task to place the returned content in the task results variable.

Add the following lines to create the task for verifying the web service from the control node. Be sure to indent the first line with four spaces.

```
- name: connect to intranet web server
  uri:
    url: http://servera.lab.example.com
    return_content: yes
    status_code: 200
```

The first line provides a descriptive name for the task. The second line is indented with six spaces and calls the `uri` module. The remaining lines are indented with eight spaces and pass the necessary arguments to execute a query for web content from the control node to the managed host and verify the status code received. The `return_content` keyword ensures that the server's response is added to the task results.

8. Verify that the final `/home/student/playbook-multi/intranet.yml` playbook reflects the following structured content, then save and close the file.

```
---
- name: Enable intranet services
  hosts: servera.lab.example.com
  become: yes
  tasks:
    - name: latest version of httpd and firewalld installed
      yum:
        name:
          - httpd
          - firewalld
        state: latest

    - name: test html page is installed
      copy:
        content: "Welcome to the example.com intranet!\n"
        dest: /var/www/html/index.html

    - name: firewalld enabled and running
      service:
        name: firewalld
        enabled: true
        state: started

    - name: firewalld permits access to httpd service
      firewalld:
        service: http
        permanent: true
        state: enabled
        immediate: yes

    - name: httpd enabled and running
      service:
        name: httpd
        enabled: true
        state: started

- name: Test intranet web server
  hosts: localhost
  become: no
  tasks:
    - name: connect to intranet web server
      uri:
        url: http://servera.lab.example.com
        return_content: yes
        status_code: 200
```

9. Run the `ansible-playbook --syntax-check` command to verify the syntax of the `/home/student/playbook-multi/intranet.yml` playbook.

```
[student@workstation playbook-multi]$ ansible-playbook --syntax-check intranet.yml

playbook: intranet.yml
```

10. Execute the playbook using the `-v` option to output detailed results for each task. Read through the output generated to ensure that all tasks completed successfully. Verify that an HTTP GET request to `http://servera.lab.example.com` provides the correct content.

```
[student@workstation playbook-multi]$ ansible-playbook -v intranet.yml
...output omitted...

PLAY [Enable intranet services] *****

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

TASK [latest version of httpd and firewallld installed] *****
changed: [servera.lab.example.com] => {"changed": true, ...output omitted...

TASK [test html page is installed] *****
changed: [servera.lab.example.com] => {"changed": true, ...output omitted...

TASK [firewalld enabled and running] *****
ok: [servera.lab.example.com] => {"changed": false, ...output omitted...

TASK [firewalld permits http service] *****
changed: [servera.lab.example.com] => {"changed": true, ...output omitted...

TASK [httpd enabled and running] *****
changed: [servera.lab.example.com] => {"changed": true, ...output omitted...

PLAY [Test intranet web server] *****

TASK [Gathering Facts] *****
ok: [localhost]

TASK [connect to intranet web server] *****
ok: [localhost] => {"accept_ranges": "bytes", "changed": false, "connection": "close", "content": "Welcome to the example.com intranet!\n", "content_length": "137", "content_type": "text/html; charset=UTF-8", "cookies": {}, "cookies_string": "", "date": "...output omitted...", "etag": "\"25-5790ddb5a48\"", "last_modified": "...output omitted...", "msg": "OK (37 bytes)", "redirected": false, "server": "Apache/2.4.6 (Red Hat Enterprise Linux)", "status": 200, "url": "http://servera.lab.example.com"}

PLAY RECAP *****
localhost                : ok=2    changed=0    unreachable=0    failed=0
servera.lab.example.com  : ok=6    changed=4    unreachable=0    failed=0
```

❶ The server responded with the desired content, Welcome to the example.com intranet!\n.

❷ The server responded with an HTTP status code of 200.

## Finish

On workstation, run the `lab playbook-multi finish` command to clean up the resources created in this exercise.

```
[student@workstation ~]$ lab playbook-multi finish
```

This concludes the guided exercise.

◀ PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S09) → NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S11)

Privacy Policy ([http://s.bl-1.com/h/cZrgWbQn?url=https://www.redhat.com/en/about/privacy-policy?extIdCarryOver=true&sc\\_cid=701f2000001D8QoAAK](http://s.bl-1.com/h/cZrgWbQn?url=https://www.redhat.com/en/about/privacy-policy?extIdCarryOver=true&sc_cid=701f2000001D8QoAAK))

Red Hat Training Policies (<http://s.bl-1.com/h/cZrb2DXG?url=https://www.redhat.com/en/about/red-hat-training-policies>)



Terms of Use (<https://www.redhat.com/en/about/terms-use>)

All policies and guidelines (<https://www.redhat.com/en/about/all-policies-guidelines>)

Release Notes (<https://learn.redhat.com/t5/Red-Hat-Learning-Subscription/Red-Hat-Learning-Subscription-Release-Notes/ba-p/22952>)

