



Days remaining 76

Search

Red Hat Enterprise Linux Automation with Ansible

DOWNLOAD EBOOK ▾

FEEDBACK

TRANSLATIONS ▾

CERTIFICATE OF ATTENDANCE

⚡

★

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

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH08S02)

→ NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH08S04)

VIDEO CLASSROOM

Troubleshooting Ansible Managed Hosts



Objectives

After completing this section, you should be able to troubleshoot failures on managed hosts when running a playbook.

Using Check Mode as a Testing Tool

You can use the `ansible-playbook --check` command to run smoke tests on a playbook. This option executes the playbook without making changes to the managed hosts' configuration. If a module used within the playbook supports *check mode* then the changes that would have been made to the managed hosts are displayed but not performed. If check mode is not supported by a module then the changes are not displayed but the module still takes no action.

```
[student@demo ~]$ ansible-playbook --check playbook.yml
```

NOTE

The `ansible-playbook --check` command might not work properly if your tasks use conditionals.

You can also control whether individual tasks run in check mode with the `check_mode` setting. If a task has `check_mode: yes` set, it always runs in check mode, whether or not you passed the `--check` option to `ansible-playbook`. Likewise, if a task has `check_mode: no` set, it always runs normally, even if you pass `--check` to `ansible-playbook`.

The following task is always run in check mode, and does not make changes.

```
tasks:
  - name: task always in check mode
    shell: uname -a
    check_mode: yes
```

The following task is always run normally, even when started with `ansible-playbook --check`.

```
tasks:
  - name: task always runs even in check mode
    shell: uname -a
    check_mode: no
```

This can be useful because you can run most of a playbook normally while testing individual tasks with `check_mode: yes`. Likewise, you can make test runs in check mode more likely to provide reasonable results by running selected tasks that gather facts or set variables for conditionals but do not change the managed hosts with `check_mode: no`.

A task can determine if the playbook is running in check mode by testing the value of the magic variable `ansible_check_mode`. This Boolean variable is set to `true` if the playbook is running in check mode.

WARNING

Tasks that have `check_mode: no` set will run even when the playbook is run with `ansible-playbook --check`. Therefore, you cannot trust that the `--check` option will make no changes to managed hosts, without confirming this to be the case by inspecting the playbook and any roles or tasks associated with it.

NOTE

If you have older playbooks that use `always_run: yes` to force tasks to run normally even in check mode, you will have to replace that code with `check_mode: no` in Ansible 2.6 and later.

The `ansible-playbook` command also provides a `--diff` option. This option reports the changes made to the template files on managed hosts. If used with the `--check` option, those changes are displayed in the command's output but not actually made.

```
[student@demo ~]$ ansible-playbook --check --diff playbook.yml
```

Testing with Modules

Some modules can provide additional information about the status of a managed host. The following list includes some of the Ansible modules that can be used to test and debug issues on managed hosts.

The `uri` module provides a way to check that a RESTful API is returning the required content.

```
tasks:
  - uri:
      url: http://api.myapp.com
      return_content: yes
      register: apiresponse

  - fail:
      msg: 'version was not provided'
      when: "'version' not in apiresponse.content"
```

The `script` module supports executing a script on managed hosts, and fails if the return code for that script is nonzero. The script must exist on the control node and is transferred to and executed on the managed hosts.

```
tasks:
  - script: check_free_memory
```

The `stat` module gathers facts for a file much like the `stat` command. You can use it to register a variable and then test to determine if the file exists or to get other information about the file. If the file does not exist, the `stat` task will not fail, but its registered variable will report `false` for `*.stat.exists`.

In this example, an application is still running if `/var/run/app.lock` exists, in which case the play should abort.

```
tasks:
  - name: Check if /var/run/app.lock exists
    stat:
      path: /var/run/app.lock
      register: lock

  - name: Fail if the application is running
    fail:
      when: lock.stat.exists
```

The `assert` module is an alternative to the `fail` module. The `assert` module supports a `that` option that takes a list of conditionals. If any of those conditionals are false, the task fails. You can use the `success_msg` and `fail_msg` options to customize the message it prints if it reports success or failure.

The following example repeats the preceding one, but uses `assert` instead of `fail`.

```
tasks:
  - name: Check if /var/run/app.lock exists
    stat:
      path: /var/run/app.lock
      register: lock

  - name: Fail if the application is running
    assert:
      that:
        - not lock.stat.exists
```

Troubleshooting Connections

Many common problems when using Ansible to manage hosts are associated with connections to the host and with configuration problems around the remote user and privilege escalation.

If you are having problems authenticating to a managed host, make sure that you have `remote_user` set correctly in your configuration file or in your play. You should also confirm that you have the correct SSH keys set up or are providing the correct password for that user.

Make sure that `become` is set properly, and that you are using the correct `become_user` (this is `root` by default). You should confirm that you are entering the correct `sudo` password and that `sudo` on the managed host is configured correctly.

A more subtle problem has to do with inventory settings. For a complex server with multiple network addresses, you may need to use a particular address or DNS name when connecting to that system. You might not want to use that address as the machine's inventory name for better readability. You can set a host inventory variable, `ansible_host`, that will override the inventory name with a different name or IP address and be used by Ansible to connect to that host. This variable could be set in the `host_vars` file or directory for that host, or could be set in the inventory file itself.

For example, the following inventory entry configures Ansible to connect to `192.0.2.4` when processing the host `web4.phx.example.com`:

```
web4.phx.example.com ansible_host=192.0.2.4
```

This is a useful way to control how Ansible connects to managed hosts. However, it can also cause problems if the value of `ansible_host` is incorrect.

Testing Managed Hosts Using Ad Hoc Commands

The following examples illustrate some of the checks that can be made on a managed host through the use of ad hoc commands.

You have used the `ping` module to test whether you can connect to managed hosts. Depending on the options you pass, you can also use it to test whether privilege escalation and credentials are correctly configured.

```
[student@demo ~]$ ansible demohost -m ping
demohost | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
[student@demo ~]$ ansible demohost -m ping --become
demohost | FAILED! => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "module_stderr": "sudo: a password is required\n",
  "module_stdout": "",
  "msg": "MODULE FAILURE\nSee stdout/stderr for the exact error",
  "rc": 1
}
```

This example returns the currently available space on the disks configured in the `demohost` managed host. That can be useful to confirm that the file system on the managed host is not full.

```
[student@demo ~]$ ansible demohost -m command -a 'df'
```

This example returns the currently available free memory on the `demohost` managed host.

```
[student@demo ~]$ ansible demohost -m command -a 'free -m'
```

The Correct Level of Testing

Ansible is designed to ensure that the configuration included in playbooks and performed by its modules is correct. It monitors all modules for reported failures, and stops the playbook immediately if any failure is encountered. This helps ensure that any task performed before the failure has no errors.

Because of this, there is usually no need to check if the result of a task managed by Ansible has been correctly applied on the managed hosts. It makes sense to add some health checks either to playbooks, or run those directly as ad hoc commands, when more direct troubleshooting is required. But, you should be careful about adding too much complexity to your tasks and plays in an effort to double check the tests performed by the modules themselves.

REFERENCES

Check Mode ("Dry Run") – Ansible Documentation (https://docs.ansible.com/ansible/2.9/user_guide/playbooks_checkmode.html)

Testing Strategies – Ansible Documentation (https://docs.ansible.com/ansible/2.9/reference_appendices/test_strategies.html)

← PREVIOUS (</ROL/APP/COURSES/RH294-8.4/PAGES/CH08S02>)

→ NEXT (</ROL/APP/COURSES/RH294-8.4/PAGES/CH08S04>)



Privacy Policy (http://s.bl-1.com/h/cZrgWbQn?url=https://www.redhat.com/en/about/privacy-policy?extIdCarryOver=true&sc_cid=701f20000001D8QoAAK)

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>)

Cookie Preferences