



(/rol/app/)

[Home\(/rol/app/\)](#)[Reports\(/rol/app/reports\)](#)[Community\(https://learn.redhat.com/\)](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/ch02)	3.4/pages/ch01)
(/rol/app/courses/rh294-8.4/pages/ch01s05)	(/rol/app/courses/rh294-8.4/pages/ch02s04)	(/rol/app/courses/rh294-8.4/pages/ch02s02)
(/rol/app/courses/rh294-8.4/pages/ch02s03)	3.4/pages/ch02)	(/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/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)	8.4/pages/ch03)	(/rol/app/courses/rh294-8.4/pages/ch03s08)
(/rol/app/courses/rh294-8.4/pages/ch04)	(/rol/app/courses/rh294-8.4/pages/ch04s02)	4 (/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/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)	8.4/pages/ch05)
(/rol/app/courses/rh294-8.4/pages/ch06s02)	(/rol/app/courses/rh294-8.4/pages/ch06s03)	(/rol/app/courses/rh294-8.4/pages/ch06)
(/rol/app/courses/rh294-8.4/pages/ch06s05)	(/rol/app/courses/rh294-8.4/pages/ch06s06)	(/rol/app/courses/rh294-8.4/pages/ch06s04)
(/rol/app/courses/rh294-8.4/pages/ch07s02)	(/rol/app/courses/rh294-8.4/pages/ch07s03)	8.4/pages/ch06)
(/rol/app/courses/rh294-8.4/pages/ch07s05)	(/rol/app/courses/rh294-8.4/pages/ch07s06)	(/rol/app/courses/rh294-8.4/pages/ch07)
(/rol/app/courses/rh294-8.4/pages/ch07s08)	(/rol/app/courses/rh294-8.4/pages/ch07s09)	(/rol/app/courses/rh294-8.4/pages/ch07s04)
(/rol/app/courses/rh294-8.4/pages/ch07s11)	(/rol/app/courses/rh294-8.4/pages/ch07s12)	8.4/pages/ch07)
(/rol/app/courses/rh294-8.4/pages/ch08s02)	(/rol/app/courses/rh294-8.4/pages/ch08s03)	(/rol/app/courses/rh294-8.4/pages/ch07s10)
(/rol/app/courses/rh294-8.4/pages/ch08s05)	(/rol/app/courses/rh294-8.4/pages/ch08s06)	8 (/rol/app/courses/rh294-8.4/pages/ch08)
(/rol/app/courses/rh294-8.4/pages/ch09s02)	(/rol/app/courses/rh294-8.4/pages/ch09s03)	(/rol/app/courses/rh294-8.4/pages/ch08s04)
(/rol/app/courses/rh294-8.4/pages/ch09s05)	(/rol/app/courses/rh294-8.4/pages/ch09s06)	8.4/pages/ch08)
(/rol/app/courses/rh294-8.4/pages/ch09s08)	(/rol/app/courses/rh294-8.4/pages/ch09s09)	(/rol/app/courses/rh294-8.4/pages/ch09)
(/rol/app/courses/rh294-8.4/pages/ch09s11)	(/rol/app/courses/rh294-8.4/pages/ch09s12)	(/rol/app/courses/rh294-8.4/pages/ch09s04)
(/rol/app/courses/rh294-8.4/pages/ch10s02)	(/rol/app/courses/rh294-8.4/pages/ch10s03)	8.4/pages/ch09)
A (/rol/app/courses/rh294-8.4/pages/apa)	(/rol/app/courses/rh294-8.4/pages/apa)	(/rol/app/courses/rh294-8.4/pages/ch09s07)
	(/rol/app/courses/rh294-8.4/pages/apb)	(/rol/app/courses/rh294-8.4/pages/ch09s10)
		10 (/rol/app/courses/rh294-8.4/pages/ch10)
		(/rol/app/courses/rh294-8.4/pages/ch10s04)
		8.4/pages/ch10)

[← PREVIOUS \(/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S04\)](#) [→ NEXT \(/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S06\)](#)

VIDEO CLASSROOM



# Running Ad Hoc Commands

## Objectives

After completing this section, you should be able to run a single Ansible automation task using an ad hoc command and explain some use cases for ad hoc commands.

## Running Ad Hoc Commands with Ansible

An *ad hoc command* is a way of executing a single Ansible task quickly, one that you do not need to save to run again later. They are simple, online operations that can be run without writing a playbook.

Ad hoc commands are useful for quick tests and changes. For example, you can use an ad hoc command to make sure that a certain line exists in the `/etc/hosts` file on a group of servers. You could use another ad hoc command to efficiently restart a service on many different machines, or to ensure that a particular software package is up-to-date.

Ad hoc commands are very useful for quickly performing simple tasks with Ansible. They do have their limits, and in general you will want to use Ansible Playbooks to realize the full power of Ansible. In many situations, however, ad hoc commands are exactly the tool you need to perform simple tasks quickly.

### Running Ad Hoc Commands

Use the `ansible` command to run ad hoc commands:

```
ansible host-pattern -m module [-a 'module arguments'] [-i inventory]
```

The *host-pattern* argument is used to specify the managed hosts on which the ad hoc command should be run. It could be a specific managed host or host group in the inventory. You have already seen this used in conjunction with the `--list-hosts` option, which shows you which hosts are matched by a particular host pattern. You have also already seen that you can use the `-i` option to specify a different inventory location to use than the default in the current Ansible configuration file.

The `-m` option takes as an argument the name of the *module* that Ansible should run on the targeted hosts. Modules are small programs that are executed to implement your task. Some modules need no additional information, but others need additional arguments to specify the details of their operation. The `-a` option takes a list of those arguments as a quoted string.

One of the simplest ad hoc commands uses the `ping` module. This module does not do an ICMP ping, but checks to see if you can run Python-based modules on managed hosts. For example, the following ad hoc command determines whether all managed hosts in the inventory can run standard modules:

```
[user@controlnode ~]$ ansible all -m ping
servera.lab.example.com | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
```

## Performing Tasks with Modules Using Ad Hoc Commands

Modules are the tools that ad hoc commands use to accomplish tasks. Ansible provides hundreds of modules which do different things. You can usually find a tested, special-purpose module that does what you need as part of the standard installation.

The `ansible-doc -l` command lists all modules installed on a system. You can use `ansible-doc` to view the documentation of particular modules by name, and find information about what arguments the modules take as options. For example, the following command displays documentation for the `ping` module:

```
[user@controlnode ~]$ ansible-doc ping
> PING      (/usr/lib/python3.6/site-packages/ansible/modules/system/ping.py)

    A trivial test module, this module always returns `pong' on successful contact. It does not make sense in playbooks, but it is useful from `/usr/bin/ansible' to
    verify the ability to login and that a usable Python is configured. This is NOT ICMP ping, this is just a trivial test module that requires Python on the
    remote-node. For Windows targets, use the [win_ping] module instead. For Network targets, use the [net_ping] module instead.

    * This module is maintained by The Ansible Core Team
OPTIONS (= is mandatory):

- data
    Data to return for the `ping' return value.
    If this parameter is set to `crash', the module will cause an exception.
    [Default: pong]
    type: str

SEE ALSO:
    * Module net_ping
      The official documentation on the net_ping module.
      https://docs.ansible.com/ansible/2.9/modules/net_ping_module.html
    * Module win_ping
      The official documentation on the win_ping module.
      https://docs.ansible.com/ansible/2.9/modules/win_ping_module.html

AUTHOR: Ansible Core Team, Michael DeHaan
METADATA:
    status:
    - stableinterface
    supported_by: core

EXAMPLES:

# Test we can logon to 'webservers' and execute python with json lib.
# ansible webservers -m ping

# Example from an Ansible Playbook
- ping:

# Induce an exception to see what happens
- ping:
    data: crash

RETURN VALUES:

ping:
    description: value provided with the data parameter
    returned: success
    type: str
    sample: pong
```

To learn more about modules, access the online Ansible documentation at [http://docs.ansible.com/ansible/2.9/modules/modules\\_by\\_category.html](http://docs.ansible.com/ansible/2.9/modules/modules_by_category.html) ([http://docs.ansible.com/ansible/2.9/modules/modules\\_by\\_category.html](http://docs.ansible.com/ansible/2.9/modules/modules_by_category.html)).

The following table lists a number of useful modules as examples. Many others exist.

**Table 2.3. Ansible Modules**

Module category	Modules
Files modules	<ul style="list-style-type: none"> <li>• <code>copy</code>: Copy a local file to the managed host</li> <li>• <code>file</code>: Set permissions and other properties of files</li> <li>• <code>lineinfile</code>: Ensure a particular line is or is not in a file</li> <li>• <code>synchronize</code>: Synchronize content using <code>rsync</code></li> </ul>
Software package modules	<ul style="list-style-type: none"> <li>• <code>package</code>: Manage packages using autodetected package manager native to the operating system</li> <li>• <code>yum</code>: Manage packages using the YUM package manager</li> <li>• <code>apt</code>: Manage packages using the APT package manager</li> <li>• <code>dnf</code>: Manage packages using the DNF package manager</li> <li>• <code>gem</code>: Manage Ruby gems</li> <li>• <code>pip</code>: Manage Python packages from PyPI</li> </ul>
System modules	<ul style="list-style-type: none"> <li>• <code>firewalld</code>: Manage arbitrary ports and services using <code>firewalld</code></li> <li>• <code>reboot</code>: Reboot a machine</li> <li>• <code>service</code>: Manage services</li> <li>• <code>user</code>: Add, remove, and manage user accounts</li> </ul>
Net Tools modules	<ul style="list-style-type: none"> <li>• <code>get_url</code>: Download files over HTTP, HTTPS, or FTP</li> <li>• <code>nmcli</code>: Manage networking</li> <li>• <code>uri</code>: Interact with web services</li> </ul>

Most modules take arguments. You can find the list of arguments available for a module in the module's documentation. Ad hoc commands pass arguments to modules using the `-a` option. When no argument is needed, omit the `-a` option from the ad hoc command. If multiple arguments need to be specified, supply them as a quoted space-separated list.

For example, the following ad hoc command uses the `user` module to ensure that the `newbie` user exists and has UID 4000 on `servera.lab.example.com`:

```
[user@controlnode ~]$ ansible -m user -a 'name=newbie uid=4000 state=present' \
> servera.lab.example.com
servera.lab.example.com | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": true,
  "comment": "",
  "createhome": true,
  "group": 4000,
  "home": "/home/newbie",
  "name": "newbie",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 4000
}
```

Most modules are *idempotent*, which means that they can be run safely multiple times, and if the system is already in the correct state, they do nothing. For example, if you run the previous ad hoc command again, it should report no change:

```
[user@controlnode ~]$ ansible -m user -a 'name=newbie uid=4000 state=present' \
> servera.lab.example.com
servera.lab.example.com | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "append": false,
  "changed": false
  "comment": "",
  "group": 4000,
  "home": "/home/newbie",
  "move_home": false,
  "name": "newbie",
  "shell": "/bin/bash",
  "state": "present",
  "uid": 4000
}
```

## Running Arbitrary Commands on Managed Hosts

The `command` module allows administrators to run arbitrary commands on the command line of managed hosts. The command to be run is specified as an argument to the module using the `-a` option. For example, the following command runs the `hostname` command on the managed hosts referenced by the `mymanagedhosts` host pattern.

```
[user@controlnode ~]$ ansible mymanagedhosts -m command -a /usr/bin/hostname
host1.lab.example.com | CHANGED | rc=0 >>
host1.lab.example.com
host2.lab.example.com | CHANGED | rc=0 >>
host2.lab.example.com
```

The previous ad hoc command example returned two lines of output for each managed host. The first line is a status report, showing the name of the managed host that the ad hoc operation ran on, as well as the outcome of the operation. The second line is the output of the command executed remotely using the Ansible `command` module.

For better readability and parsing of ad hoc command output, administrators might find it useful to have a single line of output for each operation performed on a managed host. Use the `-o` option to display the output of Ansible ad hoc commands in a single line format.

```
[user@controlnode ~]$ ansible mymanagedhosts -m command -a /usr/bin/hostname -o
host1.lab.example.com | CHANGED | rc=0 >> (stdout) host1.lab.example.com
host2.lab.example.com | CHANGED | rc=0 >> (stdout) host2.lab.example.com
```

The `command` module allows administrators to quickly execute remote commands on managed hosts. These commands are not processed by the shell on the managed hosts. As such, they cannot access shell environment variables or perform shell operations, such as redirection and piping.

### NOTE

If an ad hoc command does not specify which module to use with the `-m` option, Ansible uses the `command` module by default.

For situations where commands require shell processing, administrators can use the `shell` module. Like the `command` module, you pass the commands to be executed as arguments to the module in an ad hoc command. Ansible then executes the command remotely on the managed hosts. Unlike the `command` module, the commands are processed through a shell on the managed hosts. Therefore, shell environment variables are accessible and shell operations such as redirection and piping are also available for use.

The following example illustrates the difference between the `command` and `shell` modules. If you try to execute the built-in Bash command `set` with these two modules, it only succeeds with the `shell` module.

```
[user@controlnode ~]$ ansible localhost -m command -a set
localhost | FAILED | rc=2 >>
[Errno 2] No such file or directory
[user@controlnode ~]$ ansible localhost -m shell -a set
localhost | CHANGED | rc=0 >>
BASH=/bin/sh
BASHOPTS=cmdhist:extquote:force_ignore:hostcomplete:interact
ive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
...output omitted...
```

Both `command` and `shell` modules require a working Python installation on the managed host. A third module, `raw`, can run commands directly using the remote shell, bypassing the module subsystem. This is useful when managing systems that cannot have Python installed (for example, a network router). It can also be used to install Python on a host.

## IMPORTANT

In most circumstances, it is a recommended practice that you avoid the `command`, `shell`, and `raw` "run command" modules.

Most other modules are idempotent and can perform change tracking automatically. They can test the state of systems and do nothing if those systems are already in the correct state. By contrast, it is much more complicated to use "run command" modules in a way that is idempotent. Depending upon them makes it harder for you to be confident that rerunning an ad hoc command or playbook would not cause an unexpected failure. When a `shell` or `command` module runs, it typically reports a `CHANGED` status based on whether it thinks it affected machine state.

There are times when "run command" modules are valuable tools and a good solution to a problem. If you do need to use them, it is probably best to try to use the `command` module first, resorting to `shell` or `raw` modules only if you need their special features.

## Configuring Connections for Ad Hoc Commands

The directives for managed host connections and privilege escalation can be configured in the Ansible configuration file, and they can also be defined using options in ad hoc commands. When defined using options in ad hoc commands, they take precedence over the directive configured in the Ansible configuration file. The following table shows the analogous command-line options for each configuration file directive.

**Table 2.4. Ansible Command-line Options**

Configuration file directives	Command-line option
inventory	-i
remote_user	-u
become	--become, -b
become_method	--become-method
become_user	--become-user
become_ask_pass	--ask-become-pass, -K

Before configuring these directives using command-line options, their currently defined values can be determined by consulting the output of `ansible --help`.

```
[user@controlnode ~]$ ansible --help
...output omitted...
-b, --become                run operations with become (nopasswd implied)
--become-method=BECOME_METHOD
                           privilege escalation method to use (default=sudo),
                           valid choices: [ sudo | su | pbrun | pfexec | runas |
                           doas ]
--become-user=BECOME_USER
...output omitted...
-u REMOTE_USER, --user=REMOTE_USER
                           connect as this user (default=None)
```

## REFERENCES

[ansible\(1\) man page](#)

[Working with Patterns: Ansible Documentation](#)

([https://docs.ansible.com/ansible/2.9/user\\_guide/intro\\_patterns.html](https://docs.ansible.com/ansible/2.9/user_guide/intro_patterns.html))

[Introduction to Ad-Hoc Commands: Ansible Documentation](#)

([http://docs.ansible.com/ansible/2.9/user\\_guide/intro\\_adhoc.html](http://docs.ansible.com/ansible/2.9/user_guide/intro_adhoc.html))

[Module Index: Ansible Documentation](#)

([http://docs.ansible.com/ansible/2.9/modules/modules\\_by\\_category.html](http://docs.ansible.com/ansible/2.9/modules/modules_by_category.html))

[command - Executes a command on a remote node: Ansible Documentation](#)

([http://docs.ansible.com/ansible/2.9/modules/command\\_module.html](http://docs.ansible.com/ansible/2.9/modules/command_module.html))

[shell - Execute commands in nodes: Ansible Documentation](#)

([http://docs.ansible.com/ansible/2.9/modules/shell\\_module.html](http://docs.ansible.com/ansible/2.9/modules/shell_module.html))

◀ PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S04)    ➔ NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S06)

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

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

[Terms of Use \(https://www.redhat.com/en/about/terms-use\)](https://www.redhat.com/en/about/terms-use)

[All policies and guidelines \(https://www.redhat.com/en/about/all-policies-guidelines\)](https://www.redhat.com/en/about/all-policies-guidelines)

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

[Cookie Preferences](#)

