

Introduction to ad hoc commands

An Ansible ad hoc command uses the `/usr/bin/ansible` command-line tool to automate a single task on one or more managed nodes. ad hoc commands are quick and easy, but they are not reusable. So why learn about ad hoc commands first? ad hoc commands demonstrate the simplicity and power of Ansible. The concepts you learn here will port over directly to the playbook language. Before reading and executing these examples, please read [ref: intro_inventory](#).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\ansible-devel) (docs) (docsite) (rst) (user_guide) intro_adhoc.rst, line 7); [backlink](#)

Unknown interpreted text role "ref".

- [Why use ad hoc commands?](#)
- [Use cases for ad hoc tasks](#)
 - [Rebooting servers](#)
 - [Managing files](#)
 - [Managing packages](#)
 - [Managing users and groups](#)
 - [Managing services](#)
 - [Gathering facts](#)
 - [Patterns and ad-hoc commands](#)

Why use ad hoc commands?

ad hoc commands are great for tasks you repeat rarely. For example, if you want to power off all the machines in your lab for Christmas vacation, you could execute a quick one-liner in Ansible without writing a playbook. An ad hoc command looks like this:

```
$ ansible [pattern] -m [module] -a "[module options]"
```

You can learn more about [ref: patterns<intro_patterns>](#) and [ref: modules<working_with_modules>](#) on other pages.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\ansible-devel) (docs) (docsite) (rst) (user_guide) intro_adhoc.rst, line 21); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\ansible-devel) (docs) (docsite) (rst) (user_guide) intro_adhoc.rst, line 21); [backlink](#)

Unknown interpreted text role "ref".

Use cases for ad hoc tasks

ad hoc tasks can be used to reboot servers, copy files, manage packages and users, and much more. You can use any Ansible module in an ad hoc task. ad hoc tasks, like playbooks, use a declarative model, calculating and executing the actions required to reach a specified final state. They achieve a form of idempotence by checking the current state before they begin and doing nothing unless the current state is different from the specified final state.

Rebooting servers

The default module for the `ansible` command-line utility is the [ref: ansible.builtin.command module<command_module>](#). You can use an ad hoc task to call the `command` module and reboot all web servers in Atlanta, 10 at a time. Before Ansible can do this, you must have all servers in Atlanta listed in a group called `[atlanta]` in your inventory, and you must have working SSH credentials for each machine in that group. To reboot all the servers in the `[atlanta]` group:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\ansible-devel) (docs) (docsite) (rst) (user_guide) intro_adhoc.rst, line 33); [backlink](#)

Unknown interpreted text role "ref".

```
$ ansible atlanta -a "/sbin/reboot"
```

By default Ansible uses only 5 simultaneous processes. If you have more hosts than the value set for the fork count, Ansible will talk to them, but it will take a little longer. To reboot the `[atlanta]` servers with 10 parallel forks:

```
$ ansible atlanta -a "/sbin/reboot" -f 10
```

`/usr/bin/ansible` will default to running from your user account. To connect as a different user:

```
$ ansible atlanta -a "/sbin/reboot" -f 10 -u username
```

Rebooting probably requires privilege escalation. You can connect to the server as `username` and run the command as the `root` user by using the `ref:become <become>` keyword:

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 51); backlink
```

Unknown interpreted text role "ref".

```
$ ansible atlanta -a "/sbin/reboot" -f 10 -u username --become [--ask-become-pass]
```

If you add `--ask-become-pass` or `-K`, Ansible prompts you for the password to use for privilege escalation (`sudo/su/pfexec/doas/etc`).

Note

The `ref:command module <command_module>` does not support extended shell syntax like piping and redirects (although shell variables will always work). If your command requires shell-specific syntax, use the `shell` module instead. Read more about the differences on the `ref:working_with_modules` page.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-
resources\ansible-devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs)
(docsite) (rst) (user_guide)intro_adhoc.rst, line 60); backlink
```

Unknown interpreted text role "ref".

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-
resources\ansible-devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs)
(docsite) (rst) (user_guide)intro_adhoc.rst, line 60); backlink
```

Unknown interpreted text role "ref".

So far all our examples have used the default 'command' module. To use a different module, pass `-m` for module name. For example, to use the `ref:ansible.builtin.shell module <shell_module>`:

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 65); backlink
```

Unknown interpreted text role "ref".

```
$ ansible raleigh -m ansible.builtin.shell -a 'echo $TERM'
```

When running any command with the Ansible *ad hoc* CLI (as opposed to `ref:Playbooks <working_with_playbooks>`), pay particular attention to shell quoting rules, so the local shell retains the variable and passes it to Ansible. For example, using double rather than single quotes in the above example would evaluate the variable on the box you were on.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 71); backlink
```

Unknown interpreted text role "ref".

Managing files

An *ad hoc* task can harness the power of Ansible and SCP to transfer many files to multiple machines in parallel. To transfer a file directly to all servers in the `[atlanta]` group:

```
$ ansible atlanta -m ansible.builtin.copy -a "src=/etc/hosts dest=/tmp/hosts"
```

If you plan to repeat a task like this, use the `ref:ansible.builtin.template<template_module>` module in a playbook.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 88); backlink
```

Unknown interpreted text role "ref".

The `ref:ansible.builtin.file<file_module>` module allows changing ownership and permissions on files. These same options can be passed directly to the `copy` module as well:

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 90); backlink
```

Unknown interpreted text role "ref".

```
$ ansible webservers -m ansible.builtin.file -a "dest=/srv/foo/a.txt mode=600"
$ ansible webservers -m ansible.builtin.file -a "dest=/srv/foo/b.txt mode=600 owner=mdehaan group=mdehaan"
```

The `file` module can also create directories, similar to `mkdir -p`:

```
$ ansible webservers -m ansible.builtin.file -a "dest=/path/to/c mode=755 owner=mdehaan group=mdehaan state=directory"
```

As well as delete directories (recursively) and delete files:

```
$ ansible webservers -m ansible.builtin.file -a "dest=/path/to/c state=absent"
```

Managing packages

You might also use an ad hoc task to install, update, or remove packages on managed nodes using a package management module like `yum`. To ensure a package is installed without updating it:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme state=present"
```

To ensure a specific version of a package is installed:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme-1.5 state=present"
```

To ensure a package is at the latest version:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme state=latest"
```

To ensure a package is not installed:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme state=absent"
```

Ansible has modules for managing packages under many platforms. If there is no module for your package manager, you can install packages using the `command` module or create a module for your package manager.

Managing users and groups

You can create, manage, and remove user accounts on your managed nodes with ad hoc tasks:

```
$ ansible all -m ansible.builtin.user -a "name=foo password=<crypt> password here>"
$ ansible all -m ansible.builtin.user -a "name=foo state=absent"
```

See the [ref:ansible.builtin.user <user_module>](#) module documentation for details on all of the available options, including how to manipulate groups and group membership.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 154); [backlink](#)
Unknown interpreted text role "ref".

Managing services

Ensure a service is started on all web servers:

```
$ ansible webservers -m ansible.builtin.service -a "name=httpd state=started"
```

Alternatively, restart a service on all web servers:

```
$ ansible webservers -m ansible.builtin.service -a "name=httpd state=restarted"
```

Ensure a service is stopped:

```
$ ansible webservers -m ansible.builtin.service -a "name=httpd state=stopped"
```

Gathering facts

Facts represent discovered variables about a system. You can use facts to implement conditional execution of tasks but also just to get ad hoc information about your systems. To see all facts:

```
$ ansible all -m ansible.builtin.setup
```

You can also filter this output to display only certain facts, see the [ref:ansible.builtin.setup <setup_module>](#) module documentation for details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 191); [backlink](#)
Unknown interpreted text role "ref".

Patterns and ad-hoc commands

See the [ref:patterns <intro_patterns>](#) documentation for details on all of the available options, including how to limit using patterns

in ad-hoc commands.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 196); backlink
Unknown interpreted text role "ref".
```

Now that you understand the basic elements of Ansible execution, you are ready to learn to automate repetitive tasks using [ref: Ansible Playbooks <playbooks_intro>](#).

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 199); backlink
Unknown interpreted text role "ref".
```

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-
devel\docs\docsite\rst\user_guide\ (ansible-devel) (docs) (docsite) (rst)
(user_guide)intro_adhoc.rst, line 201)
```

Unknown directive type "seealso".

```
.. seealso::

:ref:`intro_configuration`
    All about the Ansible config file
:ref:`list_of_collections`
    Browse existing collections, modules, and plugins
:ref:`working_with_playbooks`
    Using Ansible for configuration management & deployment
`Mailing List <https://groups.google.com/group/ansible-project>`_
    Questions? Help? Ideas? Stop by the list on Google Groups
:ref:`communication_irc`
    How to join Ansible chat channels
```