[**Introduction To Ad-Hoc Commands**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id7)

**Topics**

* [Introduction To Ad-Hoc Commands](http://docs.ansible.com/ansible/latest/intro_adhoc.html#introduction-to-ad-hoc-commands)
  + [Parallelism and Shell Commands](http://docs.ansible.com/ansible/latest/intro_adhoc.html#parallelism-and-shell-commands)
  + [File Transfer](http://docs.ansible.com/ansible/latest/intro_adhoc.html#file-transfer)
  + [Managing Packages](http://docs.ansible.com/ansible/latest/intro_adhoc.html#managing-packages)
  + [Users and Groups](http://docs.ansible.com/ansible/latest/intro_adhoc.html#users-and-groups)
  + [Deploying From Source Control](http://docs.ansible.com/ansible/latest/intro_adhoc.html#deploying-from-source-control)
  + [Managing Services](http://docs.ansible.com/ansible/latest/intro_adhoc.html#managing-services)
  + [Time Limited Background Operations](http://docs.ansible.com/ansible/latest/intro_adhoc.html#time-limited-background-operations)
  + [Gathering Facts](http://docs.ansible.com/ansible/latest/intro_adhoc.html#gathering-facts)

The following examples show how to use */usr/bin/ansible* for running ad hoc tasks.

What’s an ad-hoc command?

An ad-hoc command is something that you might type in to do something really quick, but don’t want to save for later.

This is a good place to start to understand the basics of what Ansible can do prior to learning the playbooks language – ad-hoc commands can also be used to do quick things that you might not necessarily want to write a full playbook for.

Generally speaking, the true power of Ansible lies in playbooks. Why would you use ad-hoc tasks versus playbooks?

For instance, if you wanted to power off all of your lab for Christmas vacation, you could execute a quick one-liner in Ansible without writing a playbook.

For configuration management and deployments, though, you’ll want to pick up on using ‘/usr/bin/ansible-playbook’ – the concepts you will learn here will port over directly to the playbook language.

(See [Playbooks](http://docs.ansible.com/ansible/latest/playbooks.html) for more information about those)

If you haven’t read [Inventory](http://docs.ansible.com/ansible/latest/intro_inventory.html) already, please look that over a bit first and then we’ll get going.

[**Parallelism and Shell Commands**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id8)

Arbitrary example.

Let’s use Ansible’s command line tool to reboot all web servers in Atlanta, 10 at a time. First, let’s set up SSH-agent so it can remember our credentials:

$ ssh-agent bash

$ ssh-add ~/.ssh/id\_rsa

If you don’t want to use ssh-agent and want to instead SSH with a password instead of keys, you can with --ask-pass (-k), but it’s much better to just use ssh-agent.

Now to run the command on all servers in a group, in this case, *atlanta*, in 10 parallel forks:

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

/usr/bin/ansible will default to running from your user account. If you do not like this behavior, pass in “-u username”. If you want to run commands as a different user, it looks like this:

$ ansible atlanta -a "/usr/bin/foo" -u username

Often you’ll not want to just do things from your user account. If you want to run commands through privilege escalation:

$ ansible atlanta -a "/usr/bin/foo" -u username --become **[**--ask-become-pass**]**

Use --ask-become-pass (-K) if you are not using a passwordless privilege escalation method (sudo/su/pfexec/doas/etc). This will interactively prompt you for the password to use. Use of a passwordless setup makes things easier to automate, but it’s not required.

It is also possible to become a user other than root using --become-user:

$ ansible atlanta -a "/usr/bin/foo" -u username --become-user otheruser **[**--ask-become-pass**]**

**Note**

Rarely, some users have security rules where they constrain their sudo/pbrun/doas environment to running specific command paths only. This does not work with ansible’s no-bootstrapping philosophy and hundreds of different modules. If doing this, use Ansible from a special account that does not have this constraint. One way of doing this without sharing access to unauthorized users would be gating Ansible with [Ansible Tower](http://docs.ansible.com/ansible/latest/tower.html), which can hold on to an SSH credential and let members of certain organizations use it on their behalf without having direct access.

Ok, so those are basics. If you didn’t read about patterns and groups yet, go back and read [Patterns](http://docs.ansible.com/ansible/latest/intro_patterns.html).

The -f 10 in the above specifies the usage of 10 simultaneous processes to use. You can also set this in [Configuration file](http://docs.ansible.com/ansible/latest/intro_configuration.html) to avoid setting it again. The default is actually 5, which is really small and conservative. You are probably going to want to talk to a lot more simultaneous hosts so feel free to crank this up. 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. Feel free to push this value as high as your system can handle!

You can also select what Ansible “module” you want to run. Normally commands also take a -m for module name, but the default module name is ‘command’, so we didn’t need to specify that all of the time. We’ll use -m in later examples to run some other [About Modules](http://docs.ansible.com/ansible/latest/modules.html).

**Note**

The [command - Executes a command on a remote node](http://docs.ansible.com/ansible/latest/command_module.html#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 [About Modules](http://docs.ansible.com/ansible/latest/modules.html) page.

Using the [shell - Execute commands in nodes.](http://docs.ansible.com/ansible/latest/shell_module.html#shell) module looks like this:

$ ansible raleigh -m shell -a 'echo $TERM'

When running any command with the Ansible *ad hoc* CLI (as opposed to [Playbooks](http://docs.ansible.com/ansible/latest/playbooks.html)), pay particular attention to shell quoting rules, so the local shell doesn’t eat a variable before it gets passed to Ansible. For example, using double rather than single quotes in the above example would evaluate the variable on the box you were on.

So far we’ve been demoing simple command execution, but most Ansible modules are not simple imperative scripts. Instead, they use a declarative model, calculating and executing the actions required to reach a specified final state. Furthermore, they achieve a form of idempotence by checking the current state before they begin, and if the current state matches the specified final state, doing nothing. However, we also recognize that running arbitrary commands can be valuable, so Ansible easily supports both.

[**File Transfer**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id9)

Here’s another use case for the */usr/bin/ansible* command line. Ansible can SCP lots of files to multiple machines in parallel.

To transfer a file directly to many servers:

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

If you use playbooks, you can also take advantage of the template module, which takes this another step further. (See module and playbook documentation).

The file module allows changing ownership and permissions on files. These same options can be passed directly to the copy module as well:

$ ansible webservers -m file -a "dest=/srv/foo/a.txt mode=600"

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

[**Managing Packages**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id10)

There are modules available for yum and apt. Here are some examples with yum.

Ensure a package is installed, but don’t update it:

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

Ensure a package is installed to a specific version:

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

Ensure a package is at the latest version:

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

Ensure a package is not installed:

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

Ansible has modules for managing packages under many platforms. If there isn’t a module for your package manager, you can install packages using the command module or (better!) contribute a module for your package manager. Stop by the mailing list for info/details.

[**Users and Groups**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id11)

The ‘user’ module allows easy creation and manipulation of existing user accounts, as well as removal of user accounts that may exist:

$ ansible all -m user -a "name=foo password=<crypted password here>"

$ ansible all -m user -a "name=foo state=absent"

See the [About Modules](http://docs.ansible.com/ansible/latest/modules.html) section for details on all of the available options, including how to manipulate groups and group membership.

[**Deploying From Source Control**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id12)

Deploy your webapp straight from git:

$ ansible webservers -m git -a "repo=https://foo.example.org/repo.git dest=/srv/myapp version=HEAD"

Since Ansible modules can notify change handlers it is possible to tell Ansible to run specific tasks when the code is updated, such as deploying Perl/Python/PHP/Ruby directly from git and then restarting apache.

[**Managing Services**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id13)

Ensure a service is started on all webservers:

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

Alternatively, restart a service on all webservers:

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

Ensure a service is stopped:

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

[**Time Limited Background Operations**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id14)

Long running operations can be run in the background, and it is possible to check their status later. For example, to execute long\_running\_operationasynchronously in the background, with a timeout of 3600 seconds (-B), and without polling (-P):

$ ansible all -B 3600 -P 0 -a "/usr/bin/long\_running\_operation --do-stuff"

If you do decide you want to check on the job status later, you can use the async\_status module, passing it the job id that was returned when you ran the original job in the background:

$ ansible web1.example.com -m async\_status -a "jid=488359678239.2844"

Polling is built-in and looks like this:

$ ansible all -B 1800 -P 60 -a "/usr/bin/long\_running\_operation --do-stuff"

The above example says “run for 30 minutes max (-B 30\*60=1800), poll for status (-P) every 60 seconds”.

Poll mode is smart so all jobs will be started before polling will begin on any machine. Be sure to use a high enough --forks value if you want to get all of your jobs started very quickly. After the time limit (in seconds) runs out (-B), the process on the remote nodes will be terminated.

Typically you’ll only be backgrounding long-running shell commands or software upgrades. Backgrounding the copy module does not do a background file transfer. [Playbooks](http://docs.ansible.com/ansible/latest/playbooks.html) also support polling, and have a simplified syntax for this.

[**Gathering Facts**](http://docs.ansible.com/ansible/latest/intro_adhoc.html#id15)

Facts are described in the playbooks section and represent discovered variables about a system. These can be used to implement conditional execution of tasks but also just to get ad-hoc information about your system. You can see all facts via:

$ ansible all -m setup

# [Configuration file](http://docs.ansible.com/ansible/latest/intro_configuration.html#id81)

**Topics**

* [Configuration file](http://docs.ansible.com/ansible/latest/intro_configuration.html#configuration-file)
  + [Getting the latest configuration](http://docs.ansible.com/ansible/latest/intro_configuration.html#getting-the-latest-configuration)
  + [Environmental configuration](http://docs.ansible.com/ansible/latest/intro_configuration.html#environmental-configuration)
  + [Explanation of values by section](http://docs.ansible.com/ansible/latest/intro_configuration.html#explanation-of-values-by-section)
    - [General defaults](http://docs.ansible.com/ansible/latest/intro_configuration.html#general-defaults)
      * [action\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#action-plugins)
      * [allow\_unsafe\_lookups](http://docs.ansible.com/ansible/latest/intro_configuration.html#allow-unsafe-lookups)
      * [allow\_world\_readable\_tmpfiles](http://docs.ansible.com/ansible/latest/intro_configuration.html#allow-world-readable-tmpfiles)
      * [ansible\_managed](http://docs.ansible.com/ansible/latest/intro_configuration.html#ansible-managed)
      * [ask\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#ask-pass)
      * [ask\_sudo\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#ask-sudo-pass)
      * [ask\_vault\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#ask-vault-pass)
      * [bin\_ansible\_callbacks](http://docs.ansible.com/ansible/latest/intro_configuration.html#bin-ansible-callbacks)
      * [callback\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#callback-plugins)
      * [callback\_whitelist](http://docs.ansible.com/ansible/latest/intro_configuration.html#callback-whitelist)
      * [command\_warnings](http://docs.ansible.com/ansible/latest/intro_configuration.html#command-warnings)
      * [connection\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#connection-plugins)
      * [deprecation\_warnings](http://docs.ansible.com/ansible/latest/intro_configuration.html#deprecation-warnings)
      * [display\_args\_to\_stdout](http://docs.ansible.com/ansible/latest/intro_configuration.html#display-args-to-stdout)
      * [display\_skipped\_hosts](http://docs.ansible.com/ansible/latest/intro_configuration.html#display-skipped-hosts)
      * [error\_on\_undefined\_vars](http://docs.ansible.com/ansible/latest/intro_configuration.html#error-on-undefined-vars)
      * [executable](http://docs.ansible.com/ansible/latest/intro_configuration.html#executable)
      * [filter\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#filter-plugins)
      * [force\_color](http://docs.ansible.com/ansible/latest/intro_configuration.html#force-color)
      * [force\_handlers](http://docs.ansible.com/ansible/latest/intro_configuration.html#force-handlers)
      * [forks](http://docs.ansible.com/ansible/latest/intro_configuration.html#forks)
      * [fact\_caching](http://docs.ansible.com/ansible/latest/intro_configuration.html#fact-caching)
      * [fact\_caching\_connection](http://docs.ansible.com/ansible/latest/intro_configuration.html#fact-caching-connection)
      * [fact\_caching\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#fact-caching-timeout)
      * [fact\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#fact-path)
      * [gathering](http://docs.ansible.com/ansible/latest/intro_configuration.html#gathering)
      * [hash\_behaviour](http://docs.ansible.com/ansible/latest/intro_configuration.html#hash-behaviour)
      * [hostfile](http://docs.ansible.com/ansible/latest/intro_configuration.html#hostfile)
      * [host\_key\_checking](http://docs.ansible.com/ansible/latest/intro_configuration.html#host-key-checking)
      * [internal\_poll\_interval](http://docs.ansible.com/ansible/latest/intro_configuration.html#internal-poll-interval)
      * [inventory](http://docs.ansible.com/ansible/latest/intro_configuration.html#inventory)
      * [inventory\_ignore\_extensions](http://docs.ansible.com/ansible/latest/intro_configuration.html#inventory-ignore-extensions)
      * [jinja2\_extensions](http://docs.ansible.com/ansible/latest/intro_configuration.html#jinja2-extensions)
      * [library](http://docs.ansible.com/ansible/latest/intro_configuration.html#library)
      * [local\_tmp](http://docs.ansible.com/ansible/latest/intro_configuration.html#local-tmp)
      * [log\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#log-path)
      * [lookup\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#lookup-plugins)
      * [merge\_multiple\_cli\_tags](http://docs.ansible.com/ansible/latest/intro_configuration.html#merge-multiple-cli-tags)
      * [module\_lang](http://docs.ansible.com/ansible/latest/intro_configuration.html#module-lang)
      * [module\_name](http://docs.ansible.com/ansible/latest/intro_configuration.html#module-name)
      * [module\_set\_locale](http://docs.ansible.com/ansible/latest/intro_configuration.html#module-set-locale)
      * [module\_utils](http://docs.ansible.com/ansible/latest/intro_configuration.html#module-utils)
      * [nocolor](http://docs.ansible.com/ansible/latest/intro_configuration.html#nocolor)
      * [nocows](http://docs.ansible.com/ansible/latest/intro_configuration.html#nocows)
      * [pattern](http://docs.ansible.com/ansible/latest/intro_configuration.html#pattern)
      * [poll\_interval](http://docs.ansible.com/ansible/latest/intro_configuration.html#poll-interval)
      * [private\_key\_file](http://docs.ansible.com/ansible/latest/intro_configuration.html#private-key-file)
      * [remote\_port](http://docs.ansible.com/ansible/latest/intro_configuration.html#remote-port)
      * [remote\_tmp](http://docs.ansible.com/ansible/latest/intro_configuration.html#remote-tmp)
      * [remote\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#remote-user)
      * [retry\_files\_enabled](http://docs.ansible.com/ansible/latest/intro_configuration.html#retry-files-enabled)
      * [retry\_files\_save\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#retry-files-save-path)
      * [roles\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#roles-path)
      * [squash\_actions](http://docs.ansible.com/ansible/latest/intro_configuration.html#squash-actions)
      * [stdout\_callback](http://docs.ansible.com/ansible/latest/intro_configuration.html#stdout-callback)
      * [strategy\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#strategy-plugins)
      * [strategy](http://docs.ansible.com/ansible/latest/intro_configuration.html#strategy)
      * [sudo\_exe](http://docs.ansible.com/ansible/latest/intro_configuration.html#sudo-exe)
      * [sudo\_flags](http://docs.ansible.com/ansible/latest/intro_configuration.html#sudo-flags)
      * [sudo\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#sudo-user)
      * [system\_warnings](http://docs.ansible.com/ansible/latest/intro_configuration.html#system-warnings)
      * [timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#timeout)
      * [transport](http://docs.ansible.com/ansible/latest/intro_configuration.html#transport)
      * [vars\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#vars-plugins)
      * [vault\_password\_file](http://docs.ansible.com/ansible/latest/intro_configuration.html#vault-password-file)
    - [Privilege Escalation Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#privilege-escalation-settings)
      * [become](http://docs.ansible.com/ansible/latest/intro_configuration.html#become)
      * [become\_method](http://docs.ansible.com/ansible/latest/intro_configuration.html#become-method)
      * [become\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#become-user)
      * [become\_ask\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#become-ask-pass)
      * [become\_allow\_same\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#become-allow-same-user)
    - [Paramiko Specific Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#paramiko-specific-settings)
      * [record\_host\_keys](http://docs.ansible.com/ansible/latest/intro_configuration.html#record-host-keys)
      * [proxy\_command](http://docs.ansible.com/ansible/latest/intro_configuration.html#proxy-command)
    - [OpenSSH Specific Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#openssh-specific-settings)
      * [ssh\_args](http://docs.ansible.com/ansible/latest/intro_configuration.html#ssh-args)
      * [control\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#control-path)
      * [control\_path\_dir](http://docs.ansible.com/ansible/latest/intro_configuration.html#control-path-dir)
      * [retries](http://docs.ansible.com/ansible/latest/intro_configuration.html#retries)
      * [scp\_if\_ssh](http://docs.ansible.com/ansible/latest/intro_configuration.html#scp-if-ssh)
      * [pipelining](http://docs.ansible.com/ansible/latest/intro_configuration.html#pipelining)
      * [ssh\_executable](http://docs.ansible.com/ansible/latest/intro_configuration.html#ssh-executable)
    - [Accelerated Mode Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#accelerated-mode-settings)
      * [accelerate\_port](http://docs.ansible.com/ansible/latest/intro_configuration.html#accelerate-port)
      * [accelerate\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#accelerate-timeout)
      * [accelerate\_connect\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#accelerate-connect-timeout)
      * [accelerate\_daemon\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#accelerate-daemon-timeout)
      * [accelerate\_multi\_key](http://docs.ansible.com/ansible/latest/intro_configuration.html#accelerate-multi-key)
    - [Selinux Specific Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#selinux-specific-settings)
      * [special\_context\_filesystems](http://docs.ansible.com/ansible/latest/intro_configuration.html#special-context-filesystems)
      * [libvirt\_lxc\_noseclabel](http://docs.ansible.com/ansible/latest/intro_configuration.html#libvirt-lxc-noseclabel)
      * [show\_custom\_stats](http://docs.ansible.com/ansible/latest/intro_configuration.html#show-custom-stats)
    - [Galaxy Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#galaxy-settings)
      * [server](http://docs.ansible.com/ansible/latest/intro_configuration.html#server)
      * [ignore\_certs](http://docs.ansible.com/ansible/latest/intro_configuration.html#ignore-certs)

Certain settings in Ansible are adjustable via a configuration file. The stock configuration should be sufficient for most users, but there may be reasons you would want to change them.

Changes can be made and used in a configuration file which will be processed in the following order:

\* ANSIBLE\_CONFIG **(**an environment variable**)**

\* ansible.cfg **(**in the current directory**)**

\* .ansible.cfg **(**in the home directory**)**

\* /etc/ansible/ansible.cfg

Prior to 1.5 the order was:

\* ansible.cfg **(**in the current directory**)**

\* ANSIBLE\_CONFIG **(**an environment variable**)**

\* .ansible.cfg **(**in the home directory**)**

\* /etc/ansible/ansible.cfg

Ansible will process the above list and use the first file found. Settings in files are not merged.

**Note**

Comments The configuration file is one variant of an INI format. Both the hash sign (“#”) and semicolon (”;”) are allowed as comment markers when the comment starts the line. However, if the comment is inline with regular values, only the semicolon is allowed to introduce the comment. For instance:

*# some basic default values...*

inventory **=** /etc/ansible/hosts ; This points to the file that lists your hosts

## [Getting the latest configuration](http://docs.ansible.com/ansible/latest/intro_configuration.html#id82)

If installing ansible from a package manager, the latest ansible.cfg should be present in /etc/ansible, possibly as a ”.rpmnew” file (or other) as appropriate in the case of updates.

If you have installed from pip or from source, however, you may want to create this file in order to override default settings in Ansible.

You may wish to consult the [ansible.cfg in source control](https://raw.github.com/ansible/ansible/devel/examples/ansible.cfg) for all of the possible latest values.

## [Environmental configuration](http://docs.ansible.com/ansible/latest/intro_configuration.html#id83)

Ansible also allows configuration of settings via environment variables. If these environment variables are set, they will override any setting loaded from the configuration file. These variables are defined in [constants.py](https://github.com/ansible/ansible/blob/devel/lib/ansible/constants.py).

## [Explanation of values by section](http://docs.ansible.com/ansible/latest/intro_configuration.html#id84)

The configuration file is broken up into sections. Most options are in the “general” section but some sections of the file are specific to certain connection types.

### [General defaults](http://docs.ansible.com/ansible/latest/intro_configuration.html#id85)

In the [defaults] section of ansible.cfg, the following settings are tunable:

#### [action\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#id86)

Actions are pieces of code in ansible that enable things like module execution, templating, and so forth.

This is a developer-centric feature that allows low-level extensions around Ansible to be loaded from different locations:

action\_plugins **=** ~/.ansible/plugins/action\_plugins/:/usr/share/ansible\_plugins/action\_plugins

Most users will not need to use this feature. See [Developing Plugins](http://docs.ansible.com/ansible/latest/dev_guide/developing_plugins.html) for more details.

#### [allow\_unsafe\_lookups](http://docs.ansible.com/ansible/latest/intro_configuration.html#id87)

New in version 2.2.3,: 2.3.1

When enabled, this option allows lookup plugins (whether used in variables as {{lookup(‘foo’)}} or as a loop as with\_foo) to return data that is **not**marked “unsafe”. By default, such data is marked as unsafe to prevent the templating engine from evaluating any jinja2 templating language, as this could represent a security risk.

This option is provided to allow for backwards-compatibility, however users should first consider adding allow\_unsafe=True to any lookups which may be expected to contain data which may be run through the templating engine later. For example:

**{{**lookup**(**'pipe', '/path/to/some/command', allow\_unsafe**=**True**)}}**

#### [allow\_world\_readable\_tmpfiles](http://docs.ansible.com/ansible/latest/intro_configuration.html#id88)

New in version 2.1.

This makes the temporary files created on the machine to be world readable and will issue a warning instead of failing the task.

It is useful when becoming an unprivileged user:

allow\_world\_readable\_tmpfiles**=**True

#### [ansible\_managed](http://docs.ansible.com/ansible/latest/intro_configuration.html#id89)

The ansible\_managed string can be inserted into files written by Ansible’s config templating system:

**{{** ansible\_managed **}}**

The default value indicates that Ansible is managing a file:

ansible\_managed **=** Ansible managed

This string can be helpful to indicate that a file should not be directly edited because Ansible may overwrite the contents of the file.

There are several special placeholder values that can be placed in the ansible\_managed string. These are not in the default ansible\_managed string because they can cause Ansible to behave as though the entire template has changed when only the ansible\_managed string has changed.

These placeholder values, along with the situations which can lead Ansible to report a template as changed when they are used, are listed below:

* Standard directives that can be used with :func:~time.strftime:. The time referred to is the modification time of the template file. Many revision control systems timestamp files according to when they are checked out, not the last time the file was modified. That means Ansible will think the file has been modified anytime there is a fresh checkout.
* {file}: expands to the name of the full path to the template file. If Ansible is run with multiple checkouts of the same configuration repository (for instance, in each sysadmin’s home directory), then the path will differ in each checkout causing Ansible to behave as though the file has been modified.
* {host}: expands to the hostname of the machine **ansible** is run on. If **ansible** is invoked on multiple machines (for instance, each sysadmin can checkout the configuration repository on their workstation and run **ansible** there), then the host will vary on each of these machines. This will cause Ansible to behave as though the file has been modified.
* {uid}: expands to the owner of the template file. If Ansible is run on checkouts of the configuration repository made by separate users (for instance, if multiple system administrators are making checkouts of the repository with their own accounts) then this will cause Ansible to behave as if the file has been modified.

#### [ask\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#id90)

This controls whether an Ansible playbook should prompt for a password by default. The default behavior is no:

ask\_pass **=** True

If using SSH keys for authentication, it’s probably not needed to change this setting.

#### [ask\_sudo\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#id91)

Similar to ask\_pass, this controls whether an Ansible playbook should prompt for a sudo password by default when sudoing. The default behavior is also no:

ask\_sudo\_pass **=** True

Users on platforms where sudo passwords are enabled should consider changing this setting.

#### [ask\_vault\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#id92)

This controls whether an Ansible playbook should prompt for the vault password by default. The default behavior is no:

ask\_vault\_pass **=** True

#### [bin\_ansible\_callbacks](http://docs.ansible.com/ansible/latest/intro_configuration.html#id93)

New in version 1.8.

Controls whether callback plugins are loaded when running /usr/bin/ansible. This may be used to log activity from the command line, send notifications, and so on. Callback plugins are always loaded for /usr/bin/ansible-playbook if present and cannot be disabled:

bin\_ansible\_callbacks **=** False

Prior to 1.8, callbacks were never loaded for /usr/bin/ansible.

#### [callback\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#id94)

Callbacks are pieces of code in ansible that get called on specific events, permitting to trigger notifications.

This is a developer-centric feature that allows low-level extensions around Ansible to be loaded from different locations:

callback\_plugins **=** ~/.ansible/plugins/callback:/usr/share/ansible/plugins/callback

Most users will not need to use this feature. See [Developing Plugins](http://docs.ansible.com/ansible/latest/dev_guide/developing_plugins.html) for more details

#### [callback\_whitelist](http://docs.ansible.com/ansible/latest/intro_configuration.html#id95)

New in version 2.0.

Now ansible ships with all included callback plugins ready to use but they are disabled by default. This setting lets you enable a list of additional callbacks. This cannot change or override the default stdout callback, use [stdout\_callback](http://docs.ansible.com/ansible/latest/intro_configuration.html#stdout-callback) for that:

callback\_whitelist **=** timer,mail

#### [command\_warnings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id96)

New in version 1.8.

By default since Ansible 1.8, Ansible will issue a warning when the shell or command module is used and the command appears to be similar to an existing Ansible module. For example, this can include reminders to use the ‘git’ module instead of shell commands to execute ‘git’. Using modules when possible over arbitrary shell commands can lead to more reliable and consistent playbook runs, and also easier to maintain playbooks:

command\_warnings **=** False

These warnings can be silenced by adjusting the following setting or adding warn=yes or warn=no to the end of the command line parameter string, like so:

- name: usage of git that could be replaced with the git module

shell: git update foo warn**=**yes

#### [connection\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#id97)

Connections plugin permit to extend the channel used by ansible to transport commands and files.

This is a developer-centric feature that allows low-level extensions around Ansible to be loaded from different locations:

connection\_plugins **=** ~/.ansible/plugins/connection\_plugins/:/usr/share/ansible\_plugins/connection\_plugins

Most users will not need to use this feature. See [Developing Plugins](http://docs.ansible.com/ansible/latest/dev_guide/developing_plugins.html) for more details

#### [deprecation\_warnings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id98)

New in version 1.3.

Allows disabling of deprecating warnings in ansible-playbook output:

deprecation\_warnings **=** True

Deprecation warnings indicate usage of legacy features that are slated for removal in a future release of Ansible.

#### [display\_args\_to\_stdout](http://docs.ansible.com/ansible/latest/intro_configuration.html#id99)

New in version 2.1.0.

By default, ansible-playbook will print a header for each task that is run to stdout. These headers will contain the name: field from the task if you specified one. If you didn’t then ansible-playbook uses the task’s action to help you tell which task is presently running. Sometimes you run many of the same action and so you want more information about the task to differentiate it from others of the same action. If you set this variable to True in the config then ansible-playbook will also include the task’s arguments in the header.

This setting defaults to False because there is a chance that you have sensitive values in your parameters and do not want those to be printed to stdout:

display\_args\_to\_stdout **=** False

If you set this to True you should be sure that you have secured your environment’s stdout (no one can shoulder surf your screen and you aren’t saving stdout to an insecure file) or made sure that all of your playbooks explicitly added the no\_log: True parameter to tasks which have sensistive values See [How do I keep secret data in my playbook?](http://docs.ansible.com/ansible/latest/faq.html#keep-secret-data) for more information.

#### [display\_skipped\_hosts](http://docs.ansible.com/ansible/latest/intro_configuration.html#id100)

If set to False, ansible will not display any status for a task that is skipped. The default behavior is to display skipped tasks:

display\_skipped\_hosts **=** True

Note that Ansible will always show the task header for any task, regardless of whether or not the task is skipped.

#### [error\_on\_undefined\_vars](http://docs.ansible.com/ansible/latest/intro_configuration.html#id101)

On by default since Ansible 1.3, this causes ansible to fail steps that reference variable names that are likely typoed:

error\_on\_undefined\_vars **=** True

If set to False, any ‘{{ template\_expression }}’ that contains undefined variables will be rendered in a template or ansible action line exactly as written.

#### [executable](http://docs.ansible.com/ansible/latest/intro_configuration.html#id102)

This indicates the command to use to spawn a shell under a sudo environment. Users may need to change this to /bin/bash in rare instances when sudo is constrained, but in most cases it may be left as is:

executable **=** /bin/bash

Starting in version 2.1 this can be overridden by the inventory var ansible\_shell\_executable.

#### [filter\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#id103)

Filters are specific functions that can be used to extend the template system.

This is a developer-centric feature that allows low-level extensions around Ansible to be loaded from different locations:

filter\_plugins **=** ~/.ansible/plugins/filter\_plugins/:/usr/share/ansible\_plugins/filter\_plugins

Most users will not need to use this feature. See [Developing Plugins](http://docs.ansible.com/ansible/latest/dev_guide/developing_plugins.html) for more details

#### [force\_color](http://docs.ansible.com/ansible/latest/intro_configuration.html#id104)

This options forces color mode even when running without a TTY:

force\_color **=** 1

#### [force\_handlers](http://docs.ansible.com/ansible/latest/intro_configuration.html#id105)

New in version 1.9.1.

This option causes notified handlers to run on a host even if a failure occurs on that host:

force\_handlers **=** True

The default is False, meaning that handlers will not run if a failure has occurred on a host. This can also be set per play or on the command line. See [Handlers and Failure](http://docs.ansible.com/ansible/latest/playbooks_error_handling.html#handlers-and-failure) for more details.

#### [forks](http://docs.ansible.com/ansible/latest/intro_configuration.html#id106)

This is the default number of parallel processes to spawn when communicating with remote hosts. Since Ansible 1.3, the fork number is automatically limited to the number of possible hosts at runtime, so this is really a limit of how much network and CPU load you think you can handle. Many users may set this to 50, some set it to 500 or more. If you have a large number of hosts, higher values will make actions across all of those hosts complete faster. The default is very very conservative:

forks **=** 5

#### [fact\_caching](http://docs.ansible.com/ansible/latest/intro_configuration.html#id107)

This option allows you to configure fact caching. When a fact cache is enabled and there is valid data for a host, Ansible will use that rather than running an implicit setup job on a remote host.

The value of this option should be the name of a cache plugin. Current versions of Ansible include redis and jsonfile:

fact\_caching **=** jsonfile

#### [fact\_caching\_connection](http://docs.ansible.com/ansible/latest/intro_configuration.html#id108)

This option tells Ansible where to cache facts. The value is plugin dependent. For the jsonfile plugin, it should be a path to a local directory. For the redis plugin, the value is a host:port:database triplet:

fact\_caching\_connection **=** localhost:6379:0

#### [fact\_caching\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#id109)

This option tells Ansible when to expire values from the cache. Setting this value to 0 effectively disables expiry, and a positive value is a TTL in seconds:

fact\_caching\_timeout **=** 86400

#### [fact\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#id110)

This option allows you to globally configure a custom path for [Local Facts (Facts.d)](http://docs.ansible.com/ansible/latest/playbooks_variables.html#local-facts) for the implied setup task when using implied fact gathering.

fact\_path = /home/centos/ansible\_facts.d

The default is to use the default from the [setup module](https://docs.ansible.com/ansible/setup_module.html): /etc/ansible/facts.d This ONLY affects fact gathering triggered by a play when gather\_facts: True.

#### [gathering](http://docs.ansible.com/ansible/latest/intro_configuration.html#id111)

New in 1.6, the ‘gathering’ setting controls the default policy of facts gathering (variables discovered about remote systems).

The value ‘implicit’ is the default, which means that the fact cache will be ignored and facts will be gathered per play unless ‘gather\_facts: False’ is set. The value ‘explicit’ is the inverse, facts will not be gathered unless directly requested in the play. The value ‘smart’ means each new host that has no facts discovered will be scanned, but if the same host is addressed in multiple plays it will not be contacted again in the playbook run. This option can be useful for those wishing to save fact gathering time. Both ‘smart’ and ‘explicit’ will use the fact cache:

gathering **=** smart

New in version 2.1.

You can specify a subset of gathered facts, via the play’s gather\_facts directive, using the following option:

gather\_subset **=** all

|  |  |
| --- | --- |
| **all:** | gather all subsets (the default) |
| **network:** | gather network facts |
| **hardware:** | gather hardware facts (longest facts to retrieve) |
| **virtual:** | gather facts about virtual machines hosted on the machine |
| **ohai:** | gather facts from ohai |
| **facter:** | gather facts from facter |

You can combine them using a comma separated list (ex: network,virtual,facter)

You can also disable specific subsets by prepending with a ! like this:

*# Don't gather hardware facts, facts from chef's ohai or puppet's facter*

gather\_subset **=** !hardware,!ohai,!facter

A set of basic facts are always collected no matter which additional subsets are selected. If you want to collect the minimal amount of facts, use !all:

gather\_subset **=** !all

#### [hash\_behaviour](http://docs.ansible.com/ansible/latest/intro_configuration.html#id112)

Ansible by default will override variables in specific precedence orders, as described in [Variables](http://docs.ansible.com/ansible/latest/playbooks_variables.html). When a variable of higher precedence wins, it will replace the other value.

Some users prefer that variables that are hashes (aka ‘dictionaries’ in Python terms) are merged. This setting is called ‘merge’. This is not the default behavior and it does not affect variables whose values are scalars (integers, strings) or arrays. We generally recommend not using this setting unless you think you have an absolute need for it, and playbooks in the official examples repos do not use this setting:

hash\_behaviour **=** replace

The valid values are either ‘replace’ (the default) or ‘merge’.

New in version 2.0.

If you want to merge hashes without changing the global settings, use the combine filter described in [Filters](http://docs.ansible.com/ansible/latest/playbooks_filters.html).

#### [hostfile](http://docs.ansible.com/ansible/latest/intro_configuration.html#id113)

This is a deprecated setting since 1.9, please look at [inventory](http://docs.ansible.com/ansible/latest/intro_configuration.html#inventory-file) for the new setting.

#### [host\_key\_checking](http://docs.ansible.com/ansible/latest/intro_configuration.html#id114)

As described in [Getting Started](http://docs.ansible.com/ansible/latest/intro_getting_started.html), host key checking is on by default in Ansible 1.3 and later. If you understand the implications and wish to disable it, you may do so here by setting the value to False:

host\_key\_checking **=** True

#### [internal\_poll\_interval](http://docs.ansible.com/ansible/latest/intro_configuration.html#id115)

New in version 2.2.

This sets the interval (in seconds) of Ansible internal processes polling each other. Lower values improve performance with large playbooks at the expense of extra CPU load. Higher values are more suitable for Ansible usage in automation scenarios, when UI responsiveness is not required but CPU usage might be a concern. Default corresponds to the value hardcoded in Ansible ≤ 2.1:

internal\_poll\_interval**=**0.001

#### [inventory](http://docs.ansible.com/ansible/latest/intro_configuration.html#id116)

This is the default location of the inventory file, script, or directory that Ansible will use to determine what hosts it has available to talk to:

inventory **=** /etc/ansible/hosts

It used to be called hostfile in Ansible before 1.9

#### [inventory\_ignore\_extensions](http://docs.ansible.com/ansible/latest/intro_configuration.html#id117)

Comma-separated list of file extension patterns to ignore when Ansible inventory is a directory with multiple sources (static and dynamic):

inventory\_ignore\_extensions **=** ~, .orig, .bak, .ini, .cfg, .retry, .pyc, .pyo

This option can be overridden by setting [ANSIBLE\_INVENTORY\_IGNORE](http://docs.ansible.com/ansible/latest/config.html#envvar-ANSIBLE_INVENTORY_IGNORE) environment variable.

#### [jinja2\_extensions](http://docs.ansible.com/ansible/latest/intro_configuration.html#id118)

This is a developer-specific feature that allows enabling additional Jinja2 extensions:

jinja2\_extensions **=** jinja2.ext.do,jinja2.ext.i18n

If you do not know what these do, you probably don’t need to change this setting :)

#### [library](http://docs.ansible.com/ansible/latest/intro_configuration.html#id119)

This is the default location Ansible looks to find modules:

library **=** /usr/share/ansible

Ansible can look in multiple locations if you feed it a colon separated path, and it also will look for modules in the ./library directory alongside a playbook.

This can be used to manage modules pulled from several different locations. For instance, a site wishing to checkout modules from several different git repositories might handle it like this:

$ mkdir -p /srv/modules

$ cd /srv/modules

$ git checkout https://vendor\_modules .

$ git checkout ssh://custom\_modules .

$ export ANSIBLE\_LIBRARY**=**/srv/modules/custom\_modules:/srv/modules/vendor\_modules

$ ansible **[**...**]**

In case of modules with the same name, the library paths are searched in order and the first module found with that name is used.

#### [local\_tmp](http://docs.ansible.com/ansible/latest/intro_configuration.html#id120)

New in version 2.1.

When Ansible gets ready to send a module to a remote machine it usually has to add a few things to the module: Some boilerplate code, the module’s parameters, and a few constants from the config file. This combination of things gets stored in a temporary file until ansible exits and cleans up after itself. The default location is a subdirectory of the user’s home directory. If you’d like to change that, you can do so by altering this setting:

local\_tmp **=** ~/.ansible/tmp

Ansible will then choose a random directory name inside this location.

#### [log\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#id121)

If present and configured in ansible.cfg, Ansible will log information about executions at the designated location. Be sure the user running Ansible has permissions on the logfile:

log\_path**=**/var/log/ansible.log

This behavior is not on by default. Note that ansible will, without this setting, record module arguments called to the syslog of managed machines. Password arguments are excluded.

For Enterprise users seeking more detailed logging history, you may be interested in [Ansible Tower](http://docs.ansible.com/ansible/latest/tower.html).

#### [lookup\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#id122)

This is a developer-centric feature that allows low-level extensions around Ansible to be loaded from different locations:

lookup\_plugins **=** ~/.ansible/plugins/lookup\_plugins/:/usr/share/ansible\_plugins/lookup\_plugins

Most users will not need to use this feature. See [Developing Plugins](http://docs.ansible.com/ansible/latest/dev_guide/developing_plugins.html) for more details

#### [merge\_multiple\_cli\_tags](http://docs.ansible.com/ansible/latest/intro_configuration.html#id123)

New in version 2.3.

This allows changing how multiple [--tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-tags) and [--skip-tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-skip-tags) arguments are handled on the command line. Specifying [--tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-tags) more than once merges all of the [--tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-tags) options together. If you want the pre-2.4.x behaviour where only the last value of [--tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-tags) is used, then set this to False. The same holds true for [--skip-tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-skip-tags).

**Note**

The default value for this in 2.3 is False. In 2.4, the default value is True. After 2.8, the option will be removed. Multiple [--tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-tags) and multiple [--skip-tags](http://docs.ansible.com/ansible/latest/ansible-playbook.html#cmdoption-ansible-playbook-skip-tags) will always be merged together.

#### [module\_lang](http://docs.ansible.com/ansible/latest/intro_configuration.html#id124)

This is to set the default language to communicate between the module and the system. By default, the value is value LANG on the controller or, if unset, en\_US.UTF-8 (it used to be C in previous versions):

module\_lang **=** en\_US.UTF-8

**Note**

This is only used if [module\_set\_locale](http://docs.ansible.com/ansible/latest/intro_configuration.html#module-set-locale) is set to True.

#### [module\_name](http://docs.ansible.com/ansible/latest/intro_configuration.html#id125)

This is the default module name (-m) value for /usr/bin/ansible. The default is the ‘command’ module. Remember the command module doesn’t support shell variables, pipes, or quotes, so you might wish to change it to ‘shell’:

module\_name **=** command

#### [module\_set\_locale](http://docs.ansible.com/ansible/latest/intro_configuration.html#id126)

This boolean value controls whether or not Ansible will prepend locale-specific environment variables (as specified via the [module\_lang](http://docs.ansible.com/ansible/latest/intro_configuration.html#module-lang) configuration option). If enabled, it results in the LANG, LC\_MESSAGES, and LC\_ALL being set when the module is executed on the given remote system. By default this is disabled.

**Note**

The module\_set\_locale option was added in Ansible-2.1 and defaulted to True. The default was changed to False in Ansible-2.2

#### [module\_utils](http://docs.ansible.com/ansible/latest/intro_configuration.html#id127)

This is the default location Ansible looks to find module\_utils:

module\_utils **=** /usr/share/ansible/my\_module\_utils

module\_utils are python modules that Ansible is able to combine with Ansible modules when sending them to the remote machine. Having custom module\_utils is useful for extracting common code when developing a set of site-specific modules.

Ansible can look in multiple locations if you feed it a colon separated path, and it also will look for modules in the ./module\_utils directory alongside a playbook.

#### [nocolor](http://docs.ansible.com/ansible/latest/intro_configuration.html#id128)

By default ansible will try to colorize output to give a better indication of failure and status information. If you dislike this behavior you can turn it off by setting ‘nocolor’ to 1:

nocolor **=** 0

#### [nocows](http://docs.ansible.com/ansible/latest/intro_configuration.html#id129)

By default ansible will take advantage of cowsay if installed to make /usr/bin/ansible-playbook runs more exciting. Why? We believe systems management should be a happy experience. If you do not like the cows, you can disable them by setting ‘nocows’ to 1:

nocows **=** 0

#### [pattern](http://docs.ansible.com/ansible/latest/intro_configuration.html#id130)

This is the default group of hosts to talk to in a playbook if no “hosts:” stanza is supplied. The default is to talk to all hosts. You may wish to change this to protect yourself from surprises:

hosts **=** \*

Note that /usr/bin/ansible always requires a host pattern and does not use this setting, only /usr/bin/ansible-playbook.

#### [poll\_interval](http://docs.ansible.com/ansible/latest/intro_configuration.html#id131)

For asynchronous tasks in Ansible (covered in [Asynchronous Actions and Polling](http://docs.ansible.com/ansible/latest/playbooks_async.html)), this is how often to check back on the status of those tasks when an explicit poll interval is not supplied. The default is a reasonably moderate 15 seconds which is a tradeoff between checking in frequently and providing a quick turnaround when something may have completed:

poll\_interval **=** 15

#### [private\_key\_file](http://docs.ansible.com/ansible/latest/intro_configuration.html#id132)

If you are using a pem file to authenticate with machines rather than SSH agent or passwords, you can set the default value here to avoid re-specifying --private-key with every invocation:

private\_key\_file**=**/path/to/file.pem

#### [remote\_port](http://docs.ansible.com/ansible/latest/intro_configuration.html#id133)

This sets the default SSH port on all of your systems, for systems that didn’t specify an alternative value in inventory. The default is the standard 22:

remote\_port **=** 22

#### [remote\_tmp](http://docs.ansible.com/ansible/latest/intro_configuration.html#id134)

Ansible works by transferring modules to your remote machines, running them, and then cleaning up after itself. In some cases, you may not wish to use the default location and would like to change the path. You can do so by altering this setting:

remote\_tmp **=** ~/.ansible/tmp

The default is to use a subdirectory of the user’s home directory. Ansible will then choose a random directory name inside this location.

#### [remote\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#id135)

This is the default username ansible will connect as for /usr/bin/ansible-playbook. Note that /usr/bin/ansible will always default to the current user if this is not defined:

remote\_user **=** root

#### [retry\_files\_enabled](http://docs.ansible.com/ansible/latest/intro_configuration.html#id136)

This controls whether a failed Ansible playbook should create a .retry file. The default setting is True:

retry\_files\_enabled **=** False

#### [retry\_files\_save\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#id137)

The retry files save path is where Ansible will save .retry files when a playbook fails and retry\_files\_enabled is True (the default). The default location is adjacent to the play (~/ in versions older than 2.0) and can be changed to any writeable path:

retry\_files\_save\_path **=** ~/.ansible/retry-files

The directory will be created if it does not already exist.

#### [roles\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#id138)

New in version 1.4.

The roles path indicate additional directories beyond the ‘roles/’ subdirectory of a playbook project to search to find Ansible roles. For instance, if there was a source control repository of common roles and a different repository of playbooks, you might choose to establish a convention to checkout roles in /opt/mysite/roles like so:

roles\_path **=** /opt/mysite/roles

Additional paths can be provided separated by colon characters, in the same way as other pathstrings:

roles\_path **=** /opt/mysite/roles:/opt/othersite/roles

Roles will be first searched for in the playbook directory. Should a role not be found, it will indicate all the possible paths that were searched.

#### [squash\_actions](http://docs.ansible.com/ansible/latest/intro_configuration.html#id139)

New in version 2.0.

Ansible can optimise actions that call modules that support list parameters when using with\_ looping. Instead of calling the module once for each item, the module is called once with the full list.

The default value for this setting is only for certain package managers, but it can be used for any module:

squash\_actions **=** apk,apt,dnf,homebrew,package,pacman,pkgng,yum,zypper

Currently, this is only supported for modules that have a name parameter, and only when the item is the only thing being passed to the parameter.

#### [stdout\_callback](http://docs.ansible.com/ansible/latest/intro_configuration.html#id140)

New in version 2.0.

This setting allows you to override the default stdout callback for ansible-playbook:

stdout\_callback **=** skippy

#### [strategy\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#id141)

Strategy plugin allow users to change the way in which Ansible runs tasks on targeted hosts.

This is a developer-centric feature that allows low-level extensions around Ansible to be loaded from different locations:

strategy\_plugins **=** ~/.ansible/plugins/strategy\_plugins/:/usr/share/ansible\_plugins/strategy\_plugins

Most users will not need to use this feature. See [Developing Plugins](http://docs.ansible.com/ansible/latest/dev_guide/developing_plugins.html) for more details

#### [strategy](http://docs.ansible.com/ansible/latest/intro_configuration.html#id142)

Strategy allow to change the default strategy used by Ansible:

strategy **=** free

#### [sudo\_exe](http://docs.ansible.com/ansible/latest/intro_configuration.html#id143)

If using an alternative sudo implementation on remote machines, the path to sudo can be replaced here provided the sudo implementation is matching CLI flags with the standard sudo:

sudo\_exe **=** sudo

#### [sudo\_flags](http://docs.ansible.com/ansible/latest/intro_configuration.html#id144)

Additional flags to pass to sudo when engaging sudo support. The default is ‘-H -S -n’ which sets the HOME environment variable, prompts for passwords via STDIN, and avoids prompting the user for input of any kind. Note that ‘-n’ will conflict with using password-less sudo auth, such as pam\_ssh\_agent\_auth. In some situations you may wish to add or remove flags, but in general most users will not need to change this setting::

sudo\_flags**=**-H -S -n

#### [sudo\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#id145)

This is the default user to sudo to if --sudo-user is not specified or ‘sudo\_user’ is not specified in an Ansible playbook. The default is the most logical: ‘root’:

sudo\_user **=** root

#### [system\_warnings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id146)

New in version 1.6.

Allows disabling of warnings related to potential issues on the system running ansible itself (not on the managed hosts):

system\_warnings **=** True

These may include warnings about 3rd party packages or other conditions that should be resolved if possible.

#### [timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#id147)

This is the default SSH timeout to use on connection attempts:

timeout **=** 10

#### [transport](http://docs.ansible.com/ansible/latest/intro_configuration.html#id148)

This is the default transport to use if “-c <transport\_name>” is not specified to /usr/bin/ansible or /usr/bin/ansible-playbook. The default is ‘smart’, which will use ‘ssh’ (OpenSSH based) if the local operating system is new enough to support ControlPersist technology, and then will otherwise use ‘paramiko’. Other transport options include ‘local’, ‘chroot’, ‘jail’, and so on.

Users should usually leave this setting as ‘smart’ and let their playbooks choose an alternate setting when needed with the ‘connection:’ play parameter:

transport **=** paramiko

#### [vars\_plugins](http://docs.ansible.com/ansible/latest/intro_configuration.html#id149)

This is a developer-centric feature that allows low-level extensions around Ansible to be loaded from different locations:

vars\_plugins **=** ~/.ansible/plugins/vars\_plugins/:/usr/share/ansible\_plugins/vars\_plugins

Most users will not need to use this feature. See [Developing Plugins](http://docs.ansible.com/ansible/latest/dev_guide/developing_plugins.html) for more details

#### [vault\_password\_file](http://docs.ansible.com/ansible/latest/intro_configuration.html#id150)

New in version 1.7.

Configures the path to the Vault password file as an alternative to specifying --vault-password-file on the command line:

vault\_password\_file **=** /path/to/vault\_password\_file

As of 1.7 this file can also be a script. If you are using a script instead of a flat file, ensure that it is marked as executable, and that the password is printed to standard output. If your script needs to prompt for data, prompts can be sent to standard error.

### [Privilege Escalation Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id151)

Ansible can use existing privilege escalation systems to allow a user to execute tasks as another. As of 1.9 ‘become’ supersedes the old sudo/su, while still being backwards compatible. Settings live under the [privilege\_escalation] header.

#### [become](http://docs.ansible.com/ansible/latest/intro_configuration.html#id152)

The equivalent of adding sudo: or su: to a play or task, set to true/yes to activate privilege escalation. The default behavior is no:

become **=** True

#### [become\_method](http://docs.ansible.com/ansible/latest/intro_configuration.html#id153)

Set the privilege escalation method. The default is sudo, other options are su, pbrun, pfexec, doas, ksu:

become\_method **=** su

#### [become\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#id154)

The equivalent to ansible\_sudo\_user or ansible\_su\_user, allows to set the user you become through privilege escalation. The default is ‘root’:

become\_user **=** root

#### [become\_ask\_pass](http://docs.ansible.com/ansible/latest/intro_configuration.html#id155)

Ask for privilege escalation password, the default is False:

become\_ask\_pass **=** True

#### [become\_allow\_same\_user](http://docs.ansible.com/ansible/latest/intro_configuration.html#id156)

Most of the time, using sudo to run a command as the same user who is running sudo itself is unnecessary overhead, so Ansible does not allow it. However, depending on the sudo configuration, it may be necessary to run a command as the same user through sudo, such as to switch SELinux contexts. For this reason, you can set become\_allow\_same\_user to True and disable this optimization.

### [Paramiko Specific Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id157)

Paramiko is the default SSH connection implementation on Enterprise Linux 6 or earlier, and is not used by default on other platforms. Settings live under the [paramiko\_connection] header.

#### [record\_host\_keys](http://docs.ansible.com/ansible/latest/intro_configuration.html#id158)

The default setting of yes will record newly discovered and approved (if host key checking is enabled) hosts in the user’s hostfile. This setting may be inefficient for large numbers of hosts, and in those situations, using the ssh transport is definitely recommended instead. Setting it to False will improve performance and is recommended when host key checking is disabled:

record\_host\_keys **=** True

#### [proxy\_command](http://docs.ansible.com/ansible/latest/intro_configuration.html#id159)

New in version 2.1.

Use an OpenSSH like ProxyCommand for proxying all Paramiko SSH connections through a bastion or jump host. Requires a minimum of Paramiko version 1.9.0. On Enterprise Linux 6 this is provided by python-paramiko1.10 in the EPEL repository:

proxy\_command **=** ssh -W "%h:%p" bastion

### [OpenSSH Specific Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id160)

Under the [ssh\_connection] header, the following settings are tunable for SSH connections. OpenSSH is the default connection type for Ansible on OSes that are new enough to support ControlPersist. (This means basically all operating systems except Enterprise Linux 6 or earlier).

#### [ssh\_args](http://docs.ansible.com/ansible/latest/intro_configuration.html#id161)

If set, this will pass a specific set of options to Ansible rather than Ansible’s usual defaults:

ssh\_args **=** -o ControlMaster**=**auto -o ControlPersist**=**60s

In particular, users may wish to raise the ControlPersist time to encourage performance. A value of 30 minutes may be appropriate. If -o ControlPath is set in ssh\_args, the control\_path setting is not used.

#### [control\_path](http://docs.ansible.com/ansible/latest/intro_configuration.html#id162)

This is the location to save ControlPath sockets. This defaults to:

control\_path**=**%**(**directory**)**s/ansible-ssh-%%h-%%p-%%r

On some systems with very long hostnames or very long path names (caused by long user names or deeply nested home directories) this can exceed the character limit on file socket names (108 characters for most platforms). In that case, you may wish to shorten the string to something like the below:

control\_path **=** %**(**directory**)**s/%%h-%%r

Ansible 1.4 and later will instruct users to run with “-vvvv” in situations where it hits this problem and if so it is easy to tell there is too long of a Control Path filename. This may be frequently encountered on EC2. This setting is ignored if -o ControlPath is set in ssh\_args.

#### [control\_path\_dir](http://docs.ansible.com/ansible/latest/intro_configuration.html#id163)

New in version 2.3.

This is the base directory of the ControlPath sockets. It is the %(directory)s part of the control\_path option. This defaults to:

control\_path\_dir**=**~/.ansible/cp

#### [retries](http://docs.ansible.com/ansible/latest/intro_configuration.html#id164)

Adds the option to retry failed ssh executions if the failure is encountered in ssh itself, not the remote command. This can be helpful if there are transient network issues. Enabled by setting retries to an integer greater than 1. Defaults to:

retries **=** 0

#### [scp\_if\_ssh](http://docs.ansible.com/ansible/latest/intro_configuration.html#id165)

Occasionally users may be managing a remote system that doesn’t have SFTP enabled. If set to True, we can cause scp to be used to transfer remote files instead:

scp\_if\_ssh **=** False

There’s really no reason to change this unless problems are encountered, and then there’s also no real drawback to managing the switch. Most environments support SFTP by default and this doesn’t usually need to be changed.

#### [pipelining](http://docs.ansible.com/ansible/latest/intro_configuration.html#id166)

Enabling pipelining reduces the number of SSH operations required to execute a module on the remote server, by executing many ansible modules without actual file transfer. This can result in a very significant performance improvement when enabled, however when using “sudo:” operations you must first disable ‘requiretty’ in /etc/sudoers on all managed hosts.

By default, this option is disabled to preserve compatibility with sudoers configurations that have requiretty (the default on many distros), but is highly recommended if you can enable it, eliminating the need for [Accelerated Mode](http://docs.ansible.com/ansible/latest/playbooks_acceleration.html):

pipelining **=** False

#### [ssh\_executable](http://docs.ansible.com/ansible/latest/intro_configuration.html#id167)

New in version 2.2.

This is the location of the ssh binary. It defaults to ssh which will use the first ssh binary available in $PATH. This config can also be overridden with ansible\_ssh\_executable inventory variable:

ssh\_executable**=**"/usr/local/bin/ssh"

This option is usually not required, it might be useful when access to system ssh is restricted, or when using ssh wrappers to connect to remote hosts.

### [Accelerated Mode Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id168)

Under the [accelerate] header, the following settings are tunable for [Accelerated Mode](http://docs.ansible.com/ansible/latest/playbooks_acceleration.html). Acceleration is a useful performance feature to use if you cannot enable [pipelining](http://docs.ansible.com/ansible/latest/intro_configuration.html#pipelining) in your environment, but is probably not needed if you can.

#### [accelerate\_port](http://docs.ansible.com/ansible/latest/intro_configuration.html#id169)

New in version 1.3.

This is the port to use for accelerated mode:

accelerate\_port **=** 5099

#### [accelerate\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#id170)

New in version 1.4.

This setting controls the timeout for receiving data from a client. If no data is received during this time, the socket connection will be closed. A keepalive packet is sent back to the controller every 15 seconds, so this timeout should not be set lower than 15 (by default, the timeout is 30 seconds):

accelerate\_timeout **=** 30

#### [accelerate\_connect\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#id171)

New in version 1.4.

This setting controls the timeout for the socket connect call, and should be kept relatively low. The connection to the accelerate\_port will be attempted 3 times before Ansible will fall back to ssh or paramiko (depending on your default connection setting) to try and start the accelerate daemon remotely. The default setting is 1.0 seconds:

accelerate\_connect\_timeout **=** 1.0

Note, this value can be set to less than one second, however it is probably not a good idea to do so unless you’re on a very fast and reliable LAN. If you’re connecting to systems over the internet, it may be necessary to increase this timeout.

#### [accelerate\_daemon\_timeout](http://docs.ansible.com/ansible/latest/intro_configuration.html#id172)

New in version 1.6.

This setting controls the timeout for the accelerated daemon, as measured in minutes. The default daemon timeout is 30 minutes:

accelerate\_daemon\_timeout **=** 30

Note, prior to 1.6, the timeout was hard-coded from the time of the daemon’s launch. For version 1.6+, the timeout is now based on the last activity to the daemon and is configurable via this option.

#### [accelerate\_multi\_key](http://docs.ansible.com/ansible/latest/intro_configuration.html#id173)

New in version 1.6.

If enabled, this setting allows multiple private keys to be uploaded to the daemon. Any clients connecting to the daemon must also enable this option:

accelerate\_multi\_key **=** yes

New clients first connect to the target node over SSH to upload the key, which is done via a local socket file, so they must have the same access as the user that launched the daemon originally.

### [Selinux Specific Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id174)

These are settings that control SELinux interactions.

#### [special\_context\_filesystems](http://docs.ansible.com/ansible/latest/intro_configuration.html#id175)

New in version 1.9.

This is a list of file systems that require special treatment when dealing with security context. The normal behaviour is for operations to copy the existing context or use the user default, this changes it to use a file system dependent context. The default list is: nfs,vboxsf,fuse,ramfs:

special\_context\_filesystems **=** nfs,vboxsf,fuse,ramfs,myspecialfs

#### [libvirt\_lxc\_noseclabel](http://docs.ansible.com/ansible/latest/intro_configuration.html#id176)

New in version 2.1.

This setting causes libvirt to connect to lxc containers by passing –noseclabel to virsh. This is necessary when running on systems which do not have SELinux. The default behavior is no:

libvirt\_lxc\_noseclabel **=** True

#### [show\_custom\_stats](http://docs.ansible.com/ansible/latest/intro_configuration.html#id177)

New in version 2.3.

If enabled, this setting will display custom stats (set via set\_stats plugin) when using the default callback.

### [Galaxy Settings](http://docs.ansible.com/ansible/latest/intro_configuration.html#id178)

The following options can be set in the [galaxy] section of ansible.cfg:

#### [server](http://docs.ansible.com/ansible/latest/intro_configuration.html#id179)

Override the default Galaxy server value of [https://galaxy.ansible.com](https://galaxy.ansible.com/). Useful if you have a hosted version of the Galaxy web app or want to point to the testing site [https://galaxy-qa.ansible.com](https://galaxy-qa.ansible.com/). It does not work against private, hosted repos, which Galaxy can use for fetching and installing roles.

#### [ignore\_certs](http://docs.ansible.com/ansible/latest/intro_configuration.html#id180)

If set to yes, ansible-galaxy will not validate TLS certificates. This can be useful for testing against a server with a self-signed certificate.