

Ansible Configuration



Managing Ansible Configuration Files

2

- The behavior of an Ansible installation can be customized by modifying settings in the **Ansible configuration file**.
- Ansible chooses its configuration file from one of several possible locations on the control node.
 - Using `/etc/ansible/ansible.cfg`
 - Using `~/.ansible.cfg`
 - Using `./ansible.cfg`
 - Environment Variable **ANSIBLE_CONFIG**

Using `/etc/ansible/ansible.cfg`

3

The ansible package provides a base configuration file located at **`/etc/ansible/ansible.cfg`**.

This file is used if no other configuration file is found.

Using ~/.ansible.cfg

4

Ansible looks for a **.ansible.cfg** file in the user's home directory. This configuration is used instead of the `/etc/ansible/ansible.cfg` if it exists and if there is no `ansible.cfg` file in the current working directory.

Using `./ansible.cfg`

5

If an `ansible.cfg` file exists in the directory in which the ansible command is executed, it is used instead of the global file or the user's personal file.

This allows administrators to create a directory structure where different environments or projects are stored in separate directories, with each directory containing a configuration file tailored with a unique set of settings.

ANSIBLE_CONFIG environment variable

6

- You can use different configuration files by placing them in different directories and then executing Ansible commands from the appropriate directory, but this method can be restrictive and hard to manage as the number of configuration files grows.
- A more flexible option is to define the location of the configuration file with the **ANSIBLE_CONFIG** environment variable.
- When this variable is defined, Ansible uses the configuration file that the variable specifies instead of any of the previously mentioned configuration files.

Configuration File Precedence

7

The search order for a configuration file is the reverse of the preceding list.

- 5 • Default from Ansible
- 4 • /etc/ansible/config
- 3 • ~/.ansible.cfg
- 2 • .ansible.cfg
- 1 • ANSIBLE_CONFIG

Configuration File Precedence

8

- Because of the multitude of locations in which Ansible configuration files can be placed, it can be confusing which configuration file is being used by Ansible.
- You can run the **ansible --version** command to clearly identify which version of Ansible is installed, and which configuration file is being used.

Managing Settings in the Configuration File

9

- The Ansible configuration file consists of several sections, with each section containing settings defined as **key-value** pairs.
- Section titles are enclosed in square brackets. For basic operation use the following two sections:
 - **[defaults]** sets defaults for Ansible operation
 - **[privilege_escalation]** configures how Ansible performs privilege escalation on managed hosts

Ansible Configuration

10

directive	description
Inventory	Specifies the path to the inventory file.
Remote User	The name of the user to log in as on the managed hosts. If not specified, the current user's name is used
Ask_pass	Whether or not to prompt for an SSH password. Can be false if using SSH public key authentication.
Become	Whether to automatically switch user on the managed host (typically to root) after connecting. This can also be specified by a play.
Become_method	How to switch user (typically sudo, which is the default, but su is an option).
Become_user	The user to switch to on the managed host (typically root, which is the default).
Become_ask_pass	Whether to prompt for a password for your become_method. Defaults to false.

Appendix



Configuring Connections

12

Ansible needs to know how to communicate with its managed hosts. One of the most common reasons to change the configuration file is to control which methods and users Ansible uses to administer managed hosts. Some of the information.

- The location of the inventory that lists the **managed hosts** and **host groups**
- Which connection protocol to use to communicate with the managed hosts (by default, SSH), and whether or not a nonstandard network port is needed to connect to the server
- Which remote user to use on the managed hosts; this could be root or it could be an unprivileged user
- If the remote user is unprivileged, Ansible needs to know if it should try to escalate privileges to root and how to do it (for example, by using sudo)
- Whether or not to prompt for an SSH password or sudo password to log in or gain privileges needed includes:

Inventory location

13

- In the **[defaults]** section, the inventory directive can point directly to a static inventory file, or to a directory containing multiple static inventory files and dynamic inventory scripts.

```
[defaults]  
inventory = ./inventory
```

Connection Settings

14

- Ansible connects to managed hosts using the **SSH protocol**. The most important parameters that control how Ansible connects to the managed hosts are set in the [defaults] section.
- Ansible attempts to connect to the managed host using the same user name as the local user running the Ansible commands.
- To specify a different remote user, set the **remote_user** parameter to that user name.

```
[defaults]  
inventory = ./inventory  
remote_user = root  
ask_pass = true
```

Connection Settings

15

- Using a Linux control node and **OpenSSH** on your managed hosts, if you can log in as the remote user with a password then you can probably set up SSH key-based authentication, which would allow you to set **ask_pass = false**.
- The first step is to make sure that the user on the control node has an SSH key pair configured in `~/.ssh`. You can run the **ssh-keygen** command to accomplish this.
- For a single existing managed host, you can install your public key on the managed host and use the **ssh-copy-id** command to populate your local `~/.ssh/known_hosts` file with its host key, as follows:

Escalating Privileges

16

- For **Security** and **Auditing reasons**, Ansible might need to connect to remote hosts as an **unprivileged user** before escalating privileges to get administrative access as root. This can be set up in the **[privilege_escalation]** section of the Ansible configuration file.
- To enable privilege escalation by default
 - set the directive **become =** in the configuration file.
- Even if this is set by default, there are various ways to override it when running ad hoc commands or Ansible Playbooks.

Escalating Privileges

17

- The following example **ansible.cfg** file assumes that you can connect to the managed hosts as **someuser** using **SSH key-based** authentication, and that someuser can use **sudo** to run commands as root without entering a password:

```
[defaults]
inventory = ./inventory
remote_user = someuser
ask_pass = true

[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false
```

Configuration File Comments

18

- There are two comment characters allowed by Ansible configuration files: the **hash** or **number sign** (#) and the **semicolon** (;).
- The number sign at the start of a line comments out the entire line.
- It must not be on the same line with a directive.
- The semicolon character comments out everything to the right of it on the line.
- It can be on the same line as a directive, as long as that directive is to its left.