

all modules ansible documentation

Published: September 19, 2026 | Index ID: #-

All Modules Ansible Documentation: A Complete Guide

When you start working with Ansible, the first thing you'll notice is the sheer number of modules available. From simple file operations to complex cloud orchestration, Ansible's module ecosystem covers almost every IT task you can think of. But navigating that wealth of information can feel overwhelming—especially when you're trying to find the exact syntax, parameters, and examples you need to get a job done. That's where a clear, comprehensive view of **all modules Ansible documentation** comes into play.

This article is designed to give you a deep understanding of how Ansible's module documentation is structured, where to find it, and how to use it effectively. We'll walk through the official docs, the local `ansible-doc` tool, collections, and community resources. By the end, you'll be able to locate any module's documentation quickly, understand its arguments, and write more robust playbooks.

Table of Contents

- Introduction
- Module Basics
- Official Ansible Documentation
- Local Module Documentation with `ansible-doc`
- Understanding Collections and Their Docs
- Ansible Galaxy and Community Collections
- Tips for Efficient Documentation Use
- Advanced Documentation Features
- Common Issues and Troubleshooting
- Conclusion

Introduction

Modules are the workhorses of Ansible. Each module implements a specific action—creating a file, installing a package, or launching a VM. The documentation for these modules tells you:

- Which arguments are supported
- What the default values are
- How the module behaves in different modes (check, diff)
- Examples that illustrate typical use cases

Because Ansible is modular and extensible, the number of available modules keeps growing. The **all modules Ansible documentation** is therefore not a static list but a dynamic, living resource. Knowing how to navigate it is essential for anyone who wants to write reliable automation scripts.

Module Basics

What Is an Ansible Module?

In Ansible, a module is a reusable unit of code that performs a single task. Modules can be written in Python, Ruby,

or any language that can read JSON from stdin and write JSON to stdout. When a playbook runs, Ansible loads the appropriate module, passes the task parameters, and interprets the module's output.

Categories of Modules

Modules fall into several broad categories:

- Core modules—bundled with Ansible itself (e.g., `file`, `yum`, `service`).
- Community modules—published by the Ansible community and distributed via Ansible Galaxy.
- Collection modules—grouped under a namespace, often tied to a vendor or product (e.g.,

`amazon.aws.ec2_instance`).

Each category has its own documentation structure, but all follow the same general format: a module description, arguments, return values, and examples.

Official Ansible Documentation

Where to Find It

The primary source for module documentation is the official Ansible website. The URL structure is consistent:

1. Navigate to <https://docs.ansible.com>.
2. Choose the desired Ansible version (e.g., 2.12, 2.13, 2.14, or the latest release).
3. Click on the Modules link in the sidebar.
4. Browse or search for the specific module you need.

For example, the `file` module documentation is located at https://docs.ansible.com/ansible/latest/collections/ansible/builtin/file_module.html.

Structure of a Module Page

Each module page follows a predictable layout that makes it easy to find information quickly:

- Module name and description—a brief overview of what the module does.
- Module arguments—a table listing each argument, its type, whether it's required, default values, and a short description.
- Return values—information on what the module returns to Ansible.
- Examples—code snippets demonstrating common usage patterns.
- Notes—additional details such as compatibility, limitations, or best practices.

Using this structure, you can quickly assess whether a module meets your needs and how to configure it properly.

Search Functionality

The official docs provide a search bar at the top of the page. It's powerful because it searches across all modules and documentation sections. To find all modules related to a particular keyword, type the keyword and filter the results by "Modules."

For example, searching for "ec2" will return modules like `amazon.aws.ec2`, `amazon.aws.ec2_instance`, and `amazon.aws.ec2_vpc_subnet`.

Local Module Documentation with `ansible-doc`

Why Use `ansible-doc`?

While the online docs are comprehensive, you often need quick, offline access—especially in environments without

internet connectivity. The `ansible-doc` command provides a local, terminal-based view of module documentation.

Basic Usage

To display the documentation for a single module, run:

```
ansible-doc -s file
```

The `-s` flag shows a short, one-page summary. If you want the full documentation, omit the flag:

```
ansible-doc file
```

This will print a detailed view, including arguments, return values, and examples.

Searching for Modules

You can search for modules by name prefix or pattern. For instance:

```
ansible-doc -l | grep -i ec2
```

The `-l` flag lists all available modules, and `grep` filters the list. This is handy when you're not sure of the exact module name.

Custom Module Paths

When you develop your own modules or use third party collections, Ansible needs to know where to look. Set the `ANSIBLE_LIBRARY` environment variable or use the `--module-path` option:

```
export ANSIBLE_LIBRARY=/opt/custom_modules  
ansible-doc -l
```

or

```
ansible-doc -l --module-path /opt/custom_modules
```

This ensures `ansible-doc` includes modules from custom directories.

Understanding Collections and Their Docs

What Are Collections?

Collections bundle modules, plugins, roles, and documentation under a single namespace. They're the recommended way to distribute Ansible content. Collections are organized like this:

- `ansible.builtin`—default core modules.
- `community.general`—community-maintained modules.
- Vendor-specific collections, e.g., `amazon.aws`, `google.cloud`, `redhat.rhel_system_roles`.

Documentation for Collection Modules

When you install a collection via `ansible-galaxy collection install`, Ansible automatically downloads the module documentation. These docs are stored in the collection's docs directory and are accessible both online and via `ansible-doc`.

To view a collection module's docs locally, use:

```
ansible-doc -s amazon.aws.ec2_instance
```

Or to search within a collection:

```
ansible-doc -l | grep amazon.aws
```

Versioning and Compatibility

Collections often have version numbers (e.g., `amazon.aws: 2.3.0`). The docs for each version are preserved, allowing you to refer to older module syntax if you're maintaining legacy playbooks. When browsing the official site, you can switch the documentation version at the top of the page to see the corresponding module docs.

Ansible Galaxy and Community Collections

What Is Ansible Galaxy?

Ansible Galaxy is the official hub for sharing collections, roles, and playbooks. It's also a great resource for discovering module documentation that isn't part of the core distribution.

Searching for Modules on Galaxy

On the Galaxy website, use the search bar and filter by **Collections**. For example, searching for “ec2” will list collections like `amazon.aws` and `community.aws`. Clicking on a collection's page provides:

- Overview and description.
- Installation instructions.
- Links to the collection's GitHub repository.
- Documentation URLs.

Installing and Using Galaxy Collections

To install a collection:

```
ansible-galaxy collection install amazon.aws
```

After installation, you can reference modules in playbooks using the full namespace:

- name:

Baca Juga (Artikel Terkait)

- [alko case study solutions maktabore](http://www.aussizzlegal.com/alko-case-study-solutions-maktabore.pdf)

URL: <http://www.aussizzlegal.com/alko-case-study-solutions-maktabore.pdf>

- [alkaloids alkaloids plants tarek ismail kakhia](http://www.aussizzlegal.com/alkaloids-alkaloids-plants-tarek-ismail-kakhia.pdf)

URL: <http://www.aussizzlegal.com/alkaloids-alkaloids-plants-tarek-ismail-kakhia.pdf>

- [alkalize or die pdf](http://www.aussizzlegal.com/alkalize-or-die-pdf.pdf)

URL: <http://www.aussizzlegal.com/alkalize-or-die-pdf.pdf>

- [alkaline material safety data sheet batteries](http://www.aussizzlegal.com/alkaline-material-safety-data-sheet-batteries.pdf)

URL: <http://www.aussizzlegal.com/alkaline-material-safety-data-sheet-batteries.pdf>

Tags: [#modules](#) [#ansible](#) [#documentation](#)

