



ansible-docsmith - ultimate tool to document ansible roles

Kirill Satarin

Red Hat, Principal Software Engineer

GitHub: @kksat

Presenting personal opinion

ANSIBLE



1. What is ansible role and important parts of it

2. How to document ansible role

`ansible-community/antsibull-docs`

`foundata/ansible-docsmith`

3. Using ansible-docsmith demo

4. Comparison



```
$ ansible-galaxy role init <role_name>
```

```
$ tree sample_role/
```

```
sample_role/
```

```
|— defaults
|   |— main.yml
|— meta
|   |— argument_specs.yml
|   |— main.yml
|— README.md
|— tasks
|   |— main.yml
|— vars
|   |— main.yml
```



```
---
dependencies: []
galaxy_info:
  role_name:
  author:
  description:
  license:
  min_ansible_version:
  platforms:
    - name:
      versions:
  galaxy_tags: [...]
```



- This is exactly what you see on galaxy.ansible.com for standalone roles and roles inside collections
- No recommended or required structure, compare to modules

Roles

Requirements

Role variables

Dependencies

Example playbook

License

Author Information

Modules

DOCUMENTATION:

Module:

Short_description:

Description:

Version_added:

Author:

Options:

...

Requirements:

...

EXAMPLES:

...

RETURN:

...



```
<entry-point-name>:
  short_description:
  description:
  version_added:
  author:
  options:
    <variable>:
      type: str
      default: <value>
      description: <text>
      choice: <list of possible values>
    <variable2>:
<entry-point-name>:
...

```

```
$ tree sample_role/  
sample_role/  
├── defaults/  
├── meta/  
├── tasks/  
│   ├── main.yml  
│   ├── <entry-point1>.yaml  
│   ├── <entry-point...>.yaml  
│   └── <entry-point-last>.yaml  
└── vars/
```

- **name:** Run tasks/other.yaml instead of 'main'
ansible.builtin.include_role:
 name: myrole
 tasks_from: <entry point name>
- **name:** Run tasks/other.yaml instead of 'main'
ansible.builtin.import_role:
 name: myrole
 tasks_from: <entry point name>



Several sources of role documentation - for modules there is only one

[Galaxy.ansible.com](https://galaxy.ansible.com)

- **readme.md** - managed manually

docs.ansible.com (or other docs website)

- rst file generated by ansible-doc from **meta/argument_specs.yml**

\$ ansible-doc

- text (or json) generated from meta/argument_specs.yml
- **ansible-doc -t role <namespace>.<collection_name>.<role_name>**

One documentation source to rule them all?



Builds role documentation from collection metadata and ansible-doc JSON, outputting Sphinx-ready RST for docsites

Generates collection docsites for role content via the collection/current workflows

Lints role docs: validates role schemas/markup and checks role cross-references within collections



```
$ pip install ansible-docsmith==2.0.0
```

```
$ ansible-docsmith generate
```

- Generates human-readable documentation directly into README.md

```
$ ansible-docsmith validate
```

- **Validates that argument_specs.yml and defaults/ agree (missing defaults, missing specs, unknown keys, mismatches)**
- Supports dry-run, verbose output, and template customization for README content



- Not all attributes (compared to modules) are present for roles
 - LICENSE, Author, requirements, dependencies
- Double maintenance and misalignment of documentation among `readme.md` and `argument_specs.yml`
- Several sources of role documentation (`readme.md` and `meta/argument_specs.yml`)



Collection structure

ansible-docsmith validate

ansible-docsmith generate

Custom jinja template to generate readme.md

```
__limitations__:
  options: {}
  short_description: Limitations
  description:
    - Role was tested only on single host.

__dependencies__:
  options: {}
  short_description: Dependencies
  description:
    - Role has no dependencies to other roles.
```

```
{% for entry_name, entry_spec in specs.items() %}
{% if entry_name[:2] == '__' %}
{% set section_title = entry_spec.short_description or entry_name %}
## {{ section_title }}<a id="{{ entry_name | replace('__', '') | replace('_', '-') }}"></a>

{{ entry_spec.description | format_description }}

{% endif %}
{% endfor %}
```



ansible-docsmith

Pros

- Great for role readme.md generation and validation
- Single source of information - argument_specs.yml
- Easy to customize with Jinja2 templates

Cons

- Does not support other content (only roles)
- Markdown output is only contains data in argument_specs.yml file

antsibull-docs (ansible-doc)

Pros

- Produces **full collection documentation**
- Good for collection docs web publishing

Cons

- Some documentation might be missing (for roles)



Thank you for your attention

GitHub: @kksat

<https://kksat.github.io/>

Ansible community forum:

<https://forum.ansible.com/>

ANSIBLE