Contributing to the Ansible Documentation

Ansible has a lot of documentation and a small team of writers. Community support helps us keep up with new features, fixes, and changes.

Improving the documentation is an easy way to make your first contribution to the Ansible project. You do not have to be a programmer, since most of our documentation is written in YAML (module documentation) or reStructuredText (rST). Some collection-level documentation is written in a subset of Markdown. If you are using Ansible, you already use YAML in your playbooks. rST and Markdown are mostly just text. You do not even need git experience, if you use the `Edit on GitHub` option.

If you find a typo, a broken example, a missing topic, or any other error or omission on this documentation website, let us know. Here are some ways to support Ansible documentation:

- Editing docs directly on GitHub
- Reviewing open PRs and issues
- Opening a new issue and/or PR
- Verifying your documentation PR
    - Setting up your environment to build documentation locally
    - Testing the documentation locally
    - Building the documentation locally
        - Building a single rST page
        - Building all the rST pages
        - Building module docs and rST pages
        - Building rST files with `sphinx-build`
        - Running the final tests
- Joining the documentation working group

Editing docs directly on GitHub

For typos and other quick fixes, you can edit most of the documentation right from the site. Look at the top right corner of this page. That `Edit on GitHub` link is available on all the guide pages in the documentation. If you have a GitHub account, you can submit a quick and easy pull request this way.

> **Note**
>
> The source files for individual collection plugins exist in their respective repositories. Follow the link to the collection on Galaxy to find where the repository is located and any guidelines on how to contribute to that collection.

To submit a documentation PR from docs.ansible.com with `Edit on GitHub`:

1. Click on `Edit on GitHub`.
2. If you don't already have a fork of the ansible repo on your GitHub account, you'll be prompted to create one.
3. Fix the typo, update the example, or make whatever other change you have in mind.
4. Enter a commit message in the first rectangle under the heading `Propose file change` at the bottom of the GitHub page. The more specific, the better. For example, "fixes typo in my_module description". You can put more detail in the second rectangle if you like. Leave the `+label: docsite_pr` there.
5. Submit the suggested change by clicking on the green "Propose file change" button. GitHub will handle branching and committing for you, and open a page with the heading "Comparing Changes".
6. Click on `Create pull request` to open the PR template.
7. Fill out the PR template, including as much detail as appropriate for your change. You can change the title of your PR if you like (by default it's the same as your commit message). In the `Issue Type` section, delete all lines except the `Docs Pull Request` line.
8. Submit your change by clicking on `Create pull request` button.
9. Be patient while Ansibot, our automated script, adds labels, pings the docs maintainers, and kicks off a CI testing run.
10. Keep an eye on your PR - the docs team may ask you for changes.

Reviewing open PRs and issues

You can also contribute by reviewing open documentation issues and PRs. To add a helpful review, please:

- Include a comment - "looks good to me" only helps if we know why.
- For issues, reproduce the problem.
- For PRs, test the change.

Opening a new issue and/or PR

If the problem you have noticed is too complex to fix with the `Edit on GitHub` option, and no open issue or PR already documents the problem, please open an issue and/or a PR on the correct underlying repo - `ansible/ansible` for most pages that are not plugin or module documentation. If the documentation page has no `Edit on GitHub` option, check if the page is for a module within a collection. If so, follow the link to the collection on Galaxy and select the `repo` button in the upper right corner to find the source repository for that collection and module. The Collection README file should contain information on how to contribute to that collection, or report issues.

A great documentation GitHub issue or PR includes:

- a specific title
- a detailed description of the problem (even for a PR - it's hard to evaluate a suggested change unless we know what problem it's meant to solve)
- links to other information (related issues/PRs, external documentation, pages on docs.ansible.com, and so on)

Verifying your documentation PR

If you make multiple changes to the documentation on `ansible/ansible`, or add more than a line to it, before you open a pull request, please:

1. Check that your text follows our :ref:`style_guide`.

    > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\[ansible-devel][docs][docsite][rst][community]documentation_contributions.rst`, **line 64);** *backlink*
    >
    > Unknown interpreted text role "ref".

2. Test your changes for rST errors.
3. Build the page, and preferably the entire documentation site, locally.

    > **Note**
    >
    > The following sections apply to documentation sourced from the `ansible/ansible` repo and does not apply to documentation from an individual collection. See the collection README file for details on how to contribute to that collection.

Setting up your environment to build documentation locally

To build documentation locally, ensure you have a working :ref:`development environment <environment_setup>`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\[ansible-devel][docs][docsite][rst][community]documentation_contributions.rst`, **line 75);** *backlink*
>
> Unknown interpreted text role "ref".

To work with documentation on your local machine, you need to have python-3.5 or greater and install the Ansible dependencies and documentation dependencies, which are listed in two :file:`requirements.txt` files to make installation easier:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\[ansible-devel][docs][docsite][rst][community]documentation_contributions.rst`, **line 77);** *backlink*
>
> Unknown interpreted text role "file".

```
pip install --user -r requirements.txt
pip install --user -r docs/docsite/requirements.txt
```

The :file:`docs/docsite/requirements.txt` file allows a wide range of versions and may install new releases of required packages. New releases of these packages may cause problems with the Ansible docs build. If you want to install tested versions of these dependencies, use :file:`docs/docsite/known_good_reqs.txt` instead:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\[ansible-devel][docs][docsite][rst][community]documentation_contributions.rst`, **line 87);** *backlink*
>
> Unknown interpreted text role "file".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\[ansible-devel][docs][docsite][rst][community]documentation_contributions.rst`, **line 87);** *backlink*
>
> Unknown interpreted text role "file".

```
pip install --user -r requirements.txt
pip install --user -r docs/docsite/known_good_reqs.txt
```

You can drop `--user` if you have set up a virtual environment (venv/virtenv).

> **Note**
>
> You may need to install these general pre-requisites separately on some systems: - `gcc` - `libyaml` - `make` - `pyparsing` - `six` On macOS with Xcode, you may need to install `six` and `pyparsing` with

`--ignore-installed` to get versions that work with `sphinx`.

> **Note**
>
> After checking out `ansible/ansible`, make sure the `docs/docsite/rst` directory has strict enough permissions. It should only be writable by the owner's account. If your default `umask` is not 022, you can use `chmod go-w docs/docsite/rst` to set the permissions correctly in your new branch. Optionally, you can set your `umask` to 022 to make all newly created files on your system (including those created by `git clone`) have the correct permissions.

Testing the documentation locally

To test an individual file for rST errors:

```
rstcheck changed_file.rst
```

Building the documentation locally

Building the documentation is the best way to check for errors and review your changes. Once *rstcheck* runs with no errors, navigate to `ansible/docs/docsite` and then build the page(s) you want to review.

> **Note**
>
> If building on macOS with Python 3.8 or later, you must use Sphinx >= 2.2.2. See #6803 for details.

Building a single rST page

To build a single rST file with the make utility:

```
make htmlsingle rst=path/to/your_file.rst
```

For example:

```
make htmlsingle rst=community/documentation_contributions.rst
```

This process compiles all the links but provides minimal log output. If you're writing a new page or want more detailed log output, refer to the instructions on :ref:`build_with_sphinx-build`

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\[ansible-devel][docs][docsite][rst][community]documentation_contributions.rst`, line 147); *backlink***
>
> Unknown interpreted text role "ref".

> **Note**
>
> `make htmlsingle` adds `rst/` to the beginning of the path you provide in `rst=`, so you can't type the filename with autocomplete. Here are the error messages you will see if you get this wrong:
>
> - If you run `make htmlsingle` from the `docs/docsite/rst/` directory: `make: *** No rule to make target 'htmlsingle'. Stop.`
> - If you run `make htmlsingle` from the `docs/docsite/` directory with the full path to your rST document: `sphinx-build: error: cannot find files ['rst/rst/community/documentation_contributions.rst'].`

Building all the rST pages

To build all the rST files without any module documentation:

```
MODULES=none make webdocs
```

Building module docs and rST pages

To build documentation for a few modules included in `ansible/ansible` plus all the rST files, use a comma-separated list:

```
MODULES=one_module,another_module make webdocs
```

To build all the module documentation plus all the rST files:

```
make webdocs
```

Building rST files with `sphinx-build`

Advanced users can build one or more rST files with the sphinx utility directly. `sphinx-build` returns misleading `undefined label` warnings if you only build a single page, because it does not create internal links. However, `sphinx-build` returns more extensive syntax feedback, including warnings about indentation errors and `x-string without end-string` warnings. This can be useful, especially if you're creating a new page from scratch. To build a page or pages with `sphinx-build`:

```
sphinx-build [options] sourcedir outdir [filenames...]
```

You can specify filenames, or â€"a for all files, or omit both to compile only new/changed files.

For example:

```
sphinx-build -b html -c rst/ rst/dev_guide/ _build/html/dev_guide/ rst/dev_guide/developing_modules_documenti
```

Running the final tests

When you submit a documentation pull request, automated tests are run. Those same tests can be run locally. To do so, navigate to the repository's top directory and run:

```
make clean &&
bin/ansible-test sanity --test docs-build &&
bin/ansible-test sanity --test rstcheck
```

Unfortunately, leftover rST-files from previous document-generating can occasionally confuse these tests. It is therefore safest to run them on a clean copy of the repository, which is the purpose of make clean. If you type these three lines one at a time and manually check the success of each, you do not need the &&.

Joining the documentation working group

The Documentation Working Group (DaWGs) meets weekly on Tuesdays in the Docs chat (using Matrix or using IRC at irc.libera.chat). For more information, including links to our agenda and a calendar invite, please visit the working group page in the community repo.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\community\[ansible-devel][docs][docsite][rst][community]documentation_contributions.rst`, line 218)

Unknown directive type "seealso".

```
.. seealso::
   :ref:`More about testing module documentation <testing_module_documentation>`

   :ref:`More about documenting modules <module_documenting>`
```