Developing Ansible modules

A module is a reusable, standalone script that Ansible runs on your behalf, either locally or remotely. Modules interact with your local machine, an API, or a remote system to perform specific tasks like changing a database password or spinning up a cloud instance. Each module can be used by the Ansible API, or by the :command:`ansible` or :command:`ansible-playbook` programs. A module provides a defined interface, accepts arguments, and returns information to Ansible by printing a JSON string to stdout before exiting.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 8); *backlink*
>
> Unknown interpreted text role "command".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 8); *backlink*
>
> Unknown interpreted text role "command".

If you need functionality that is not available in any of the thousands of Ansible modules found in collections, you can easily write your own custom module. When you write a module for local use, you can choose any programming language and follow your own rules. Use this topic to learn how to create an Ansible module in Python. After you create a module, you must add it locally to the appropriate directory so that Ansible can find and execute it. For details about adding a module locally, see :ref:`developing_locally`.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 10); *backlink*
>
> Unknown interpreted text role "ref".

Preparing an environment for developing Ansible modules

Installing prerequisites via apt (Ubuntu)

Due to dependencies (for example ansible -> paramiko -> pynacl -> libffi):

```
sudo apt update
sudo apt install build-essential libssl-dev libffi-dev python-dev
```

Installing prerequisites via yum (CentOS)

Due to dependencies (for example ansible -> paramiko -> pynacl -> libffi):

```
sudo yum check-update
sudo yum update
sudo yum group install "Development Tools"
sudo yum install python3-devel openssl-devel libffi libffi-devel
```

Creating a development environment (platform-independent steps)

1. Clone the Ansible repository: `$ git clone https://github.com/ansible/ansible.git`
2. Change directory into the repository root dir: `$ cd ansible`
3. Create a virtual environment: `$ python3 -m venv venv` (or for Python 2 `$ virtualenv venv`. Note, this requires you to install the virtualenv package: `$ pip install virtualenv`)
4. Activate the virtual environment: `$ . venv/bin/activate`
5. Install development requirements: `$ pip install -r requirements.txt`
6. Run the environment setup script for each new development shell process: `$ . hacking/env-setup`

> **Note**
>
> After the initial setup above, every time you are ready to start developing Ansible you should be able to just run the following from the root of the Ansible repo: `$ . venv/bin/activate && . hacking/env-setup`

Creating an info or a facts module

Ansible gathers information about the target machines using facts modules, and gathers information on other objects or files using info modules. If you find yourself trying to add `state: info` or `state: list` to an existing module, that is often a sign that a new dedicated `_facts` or `_info` module is needed.

In Ansible 2.8 and onwards, we have two type of information modules, they are `*_info` and `*_facts`.

If a module is named `<something>_facts`, it should be because its main purpose is returning `ansible_facts`. Do not name modules that do not do this with `_facts`. Only use `ansible_facts` for information that is specific to the host machine, for example network interfaces and their configuration, which operating system and which programs are installed.

Modules that query/return general information (and not `ansible_facts`) should be named `_info`. General information is non-host

specific information, for example information on online/cloud services (you can access different accounts for the same online service from the same host), or information on VMs and containers accessible from the machine, or information on individual files or programs.

Info and facts modules, are just like any other Ansible Module, with a few minor requirements:

1. They MUST be named `<something>_info` or `<something>_facts`, where <something> is singular.
2. Info `*_info` modules MUST return in the form of the :ref:`result dictionary<common_return_values>` so other modules can access them.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 80); *backlink*

   Unknown interpreted text role "ref".

3. Fact `*_facts` modules MUST return in the `ansible_facts` field of the :ref:`result dictionary<common_return_values>` so other modules can access them.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 81); *backlink*

   Unknown interpreted text role "ref".

4. They MUST support :ref:`check_mode <check_mode_dry>`.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 82); *backlink*

   Unknown interpreted text role "ref".

5. They MUST NOT make any changes to the system.
6. They MUST document the :ref:`return fields<return_block>` and :ref:`examples<examples_block>`.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 84); *backlink*

   Unknown interpreted text role "ref".

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 84); *backlink*

   Unknown interpreted text role "ref".

To create an info module:

1. Navigate to the correct directory for your new module: `$ cd lib/ansible/modules/`. If you are developing module using collection, `$ cd plugins/modules/` inside your collection development tree.
2. Create your new module file: `$ touch my_test_info.py`.
3. Paste the content below into your new info module file. It includes the :ref:`required Ansible format and documentation <developing_modules_documenting>`, a simple :ref:`argument spec for declaring the module options <argument_spec>`, and some example code.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 90); *backlink*

   Unknown interpreted text role "ref".

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 90); *backlink*

   Unknown interpreted text role "ref".

4. Modify and extend the code to do what you want your new info module to do. See the :ref:`programming tips <developing_modules_best_practices>` and :ref:`Python 3 compatibility <developing_python_3>` pages for pointers on writing clean and concise module code.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 91); *backlink*

   Unknown interpreted text role "ref".

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 91); *backlink*

   Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 93)

Unknown directive type "literalinclude".

```
.. literalinclude:: ../../../../examples/scripts/my_test_info.py
```

```
    :language: python
```

Use the same process to create a facts module.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 98)

Unknown directive type "literalinclude".

```
.. literalinclude:: ../../../../examples/scripts/my_test_facts.py
   :language: python
```

Creating a module

To create a module:

1. Navigate to the correct directory for your new module: `$ cd lib/ansible/modules/`. If you are developing a module in a :ref:`collection <developing_collections>`, `$ cd plugins/modules/` inside your collection development tree.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 106); *backlink*

   Unknown interpreted text role "ref".

2. Create your new module file: `$ touch my_test.py`.
3. Paste the content below into your new module file. It includes the :ref:`required Ansible format and documentation <developing_modules_documenting>`, a simple :ref:`argument spec for declaring the module options <argument_spec>`, and some example code.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 108); *backlink*

   Unknown interpreted text role "ref".

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 108); *backlink*

   Unknown interpreted text role "ref".

4. Modify and extend the code to do what you want your new module to do. See the :ref:`programming tips <developing_modules_best_practices>` and :ref:`Python 3 compatibility <developing_python_3>` pages for pointers on writing clean and concise module code.

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 109); *backlink*

   Unknown interpreted text role "ref".

   System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 109); *backlink*

   Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 111)

Unknown directive type "literalinclude".

```
.. literalinclude:: ../../../../examples/scripts/my_test.py
   :language: python
```

Verifying your module code

After you modify the sample code above to do what you want, you can try out your module. Our :ref:`debugging tips <debugging_modules>` will help if you run into bugs as you verify your module code.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst, line 117); *backlink*

Unknown interpreted text role "ref".

Verifying your module code locally

If your module does not need to target a remote host, you can quickly and easily exercise your code locally like this:

- Create an arguments file, a basic JSON config file that passes parameters to your module so that you can run it. Name the arguments file `/tmp/args.json` and add the following content:

```
{
    "ANSIBLE_MODULE_ARGS": {
        "name": "hello",
        "new": true
    }
}
```

- If you are using a virtual environment (which is highly recommended for development) activate it: `$ . venv/bin/activate`
- Set up the environment for development: `$ . hacking/env-setup`
- Run your test module locally and directly: `$ python -m ansible.modules.my_test /tmp/args.json`

This should return output like this:

```
{"changed": true, "state": {"original_message": "hello", "new_message": "goodbye"}, "invocation": {"module_args": {"name": "
```

Verifying your module code in a playbook

The next step in verifying your new module is to consume it with an Ansible playbook.

- Create a playbook in any directory: `$ touch testmod.yml`
- Add the following to the new playbook file:

```
- name: test my new module
  hosts: localhost
  tasks:
  - name: run the new module
    my_test:
      name: 'hello'
      new: true
    register: testout
  - name: dump test output
    debug:
      msg: '{{ testout }}'
```

- Run the playbook and analyze the output: `$ ansible-playbook ./testmod.yml`

Testing your newly-created module

The following two examples will get you started with testing your module code. Please review our :ref:`testing <developing_testing>` section for more detailed information, including instructions for :ref:`testing module documentation <testing_module_documentation>`, adding :ref:`integration tests <testing_integration>`, and more.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, line 177); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, line 177); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, line 177); *backlink*

Unknown interpreted text role "ref".

> **Note**
>
> Every new module and plugin should have integration tests, even if the tests cannot be run on Ansible CI infrastructure. In this case, the tests should be marked with the `unsupported` alias in aliases file.

Performing sanity tests

You can run through Ansible's sanity checks in a container:

`$ ansible-test sanity -v --docker --python 2.7 MODULE_NAME`

> **Note**
>
> Note that this example requires Docker to be installed and running. If you'd rather not use a container for this, you can choose to use `--venv` instead of `--docker`.

Adding unit tests

You can add unit tests for your module in `./test/units/modules`. You must first set up your testing environment. In this example, we're using Python 3.5.

- Install the requirements (outside of your virtual environment): `$ pip3 install -r ./test/lib/ansible_test/_data/requirements/units.txt`
- Run `. hacking/env-setup`
- To run all tests do the following: `$ ansible-test units --python 3.5`. If you are using a CI environment, these tests will run automatically.

> **Note**
>
> Ansible uses pytest for unit testing.

To run pytest against a single test module, you can run the following command. Ensure that you are providing the correct path of the test module:

`$ pytest -r a --cov=. --cov-report=html --fulltrace --color yes test/units/modules/.../test/my_test.py`

Contributing back to Ansible

If you would like to contribute to `ansible-core` by adding a new feature or fixing a bug, create a fork of the ansible/ansible repository and develop against a new feature branch using the `devel` branch as a starting point. When you have a good working code change, you can submit a pull request to the Ansible repository by selecting your feature branch as a source and the Ansible devel branch as a target.

If you want to contribute a module to an :ref:`Ansible collection <contributing_maintained_collections>`, review our :ref:`submission

checklist <developing_modules_checklist>`, :ref:`programming tips <developing_modules_best_practices>`, and :ref:`strategy for maintaining Python 2 and Python 3 compatibility <developing_python_3>`, as well as information about :ref:`testing <developing_testing>` before you open a pull request.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, **line 215**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, **line 215**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, **line 215**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, **line 215**); ***backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, **line 215**); ***backlink***
>
> Unknown interpreted text role "ref".

The :ref:`Community Guide <ansible_community_guide>` covers how to open a pull request and what happens next.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, **line 217**); ***backlink***
>
> Unknown interpreted text role "ref".

Communication and development support

Join the `#ansible-devel` chat channel (using Matrix at ansible.im or using IRC at irc.libera.chat) for discussions surrounding Ansible development.

For questions and discussions pertaining to using the Ansible product, join the `#ansible` channel.

To find other topic-specific chat channels, look at :ref:`Community Guide, Communicating <communication_irc>`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel][docs][docsite][rst][dev_guide]developing_modules_general.rst`, **line 227**); ***backlink***
>
> Unknown interpreted text role "ref".

Credit

Thank you to Thomas Stringer (@trstringer) for contributing source material for this topic.