

# Migrating Roles to Roles in Collections on Galaxy

You can migrate any existing standalone role into a collection and host the collection on Galaxy. With Ansible collections, you can distribute many roles in a single cohesive unit of re-usable automation. Inside a collection, you can share custom plugins across all roles in the collection instead of duplicating them in each role's `:file:library/` directory.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 8); [backlink](#)**

Unknown interpreted text role "file".

You must migrate roles to collections if you want to distribute them as certified Ansible content.

## Note

If you want to import your collection to Galaxy, you need a [Galaxy namespace](#).

See `ref:developing_collections` for details on collections.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 16); [backlink](#)**

Unknown interpreted text role "ref".

- [Comparing standalone roles to collection roles](#)
- [Migrating a role to a collection](#)
- [Migrating a role that contains plugins to a collection](#)
- [Using `ansible\_legacy` to access local custom modules from collections-based roles](#)

## Comparing standalone roles to collection roles

`ref:Standalone roles <playbooks_reuse_roles>` have the following directory structure:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 26); [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]migrating\_roles.rst, line 28)**

Error in "code-block" directive: unknown option: "emphasize-lines".

```
.. code-block:: bash
   :emphasize-lines: 5,7,8
```

```
role/
  â"œâ"œâ"œ defaults
  â"œâ"œâ"œ files
  â"œâ"œâ"œ handlers
  â"œâ"œâ"œ library
  â"œâ"œâ"œ meta
  â"œâ"œâ"œ module_utils
  â"œâ"œâ"œ [*_plugins]
  â"œâ"œâ"œ tasks
  â"œâ"œâ"œ templates
  â"œâ"œâ"œ tests
  â"œâ"œâ"œ vars
```

The highlighted directories above will change when you migrate to a collection-based role. The collection directory structure includes a `:file:roles/` directory:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 45); [backlink](#)**

Unknown interpreted text role "file".

```
mynamespace/
  â"œâ"œâ"œ mycollection/
    â"œâ"œâ"œ docs/
    â"œâ"œâ"œ galaxy.yml
    â"œâ"œâ"œ plugins/
    â", â"œâ"œâ"œ modules/
    â", â", â"œâ"œâ"œ module1.py
    â", â"œâ"œâ"œ inventory/
    â", â"œâ"œâ"œ .../
    â"œâ"œâ"œ README.md
    â"œâ"œâ"œ roles/
    â", â"œâ"œâ"œ role1/
    â", â"œâ"œâ"œ role2/
    â", â"œâ"œâ"œ .../
    â"œâ"œâ"œ playbooks/
    â", â"œâ"œâ"œ files/
    â", â"œâ"œâ"œ vars/
    â", â"œâ"œâ"œ templates/
    â", â"œâ"œâ"œ tasks/
```

```
$ mkdir mynamespace/mycollection/roles/my role/
$ cp -r /path/to/standalone/role/mynamespace/my role/* mynamespace/mycollection/roles/my role/
```

4. Move any modules to the `file:plugins/modules/` directory.

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 138); backlink  
Unknown interpreted text role "file".
```

```
$ mv -r mynamespace/mycollection/roles/my_role/library/\* mynamespace/mycollection/plugins/modules/
```

5. Move any other plugins to the appropriate `file:plugins/PLUGINTYPE/` directory. See `ref:migrating_plugins_collection` for additional steps that may be required.

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 144); backlink  
Unknown interpreted text role "file".
```

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 144); backlink  
Unknown interpreted text role "ref".
```

6. Update `galaxy.yml` to include any role dependencies.
7. Update the collection `README.md` file to add links to any role `README.md` files.
8. Change any references to the role to use the `abbr:FQCN (Fully Qualified Collection Name)`.

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 150); backlink  
Unknown interpreted text role "abbr".
```

```
---  
- name: example role by FQCN  
  hosts: some_host_pattern  
  tasks:  
    - name: import FQCN role from a collection  
      import_role:  
        name: mynamespace.mycollection.my_role
```

You can alternately use the `collections` keyword to simplify this:

```
---  
- name: example role by FQCN  
  hosts: some_host_pattern  
  collections:  
    - mynamespace.mycollection  
  tasks:  
    - name: import role from a collection  
      import_role:  
        name: my_role
```

## Migrating other role plugins to a collection

To migrate other role plugins to a collection:

1. Move each nonmodule plugins to the appropriate `file:plugins/PLUGINTYPE/` directory. The `file:mynamespace/mycollection/plugins/README.md` file explains the types of plugins that the collection can contain within optionally created subdirectories.

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 186); backlink  
Unknown interpreted text role "file".
```

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 186); backlink  
Unknown interpreted text role "file".
```

```
$ mv -r mynamespace/mycollection/roles/my_role/filter_plugins/\* mynamespace/mycollection/plugins/filter/
```

2. Update documentation to use the FQCN. Plugins that use `doc_fragments` need to use FQCN (for example, `mydocfrag` becomes `mynamespace.mycollection.mydocfrag`).
3. Update relative imports work in collections to start with a period. For example, `file:./filename` and `file:../asdfu/filestuff` works but `file:filename` in same directory must be updated to `file:./filename`.

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 194); backlink  
Unknown interpreted text role "file".
```

```
SystemMessage: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 194); backlink
```

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 194); [backlink](#)

Unknown interpreted text role "file".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 194); [backlink](#)

Unknown interpreted text role "file".

If you have a custom `module_utils` or import from `__init__.py`, you must also:

1. Change the Python namespace for custom `module_utils` to use the `abbr:'FQCN (Fully Qualified Collection Name)'` along with the `ansible_collections` convention. See [ref:'update\\_module\\_utils\\_role'](#).

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 199); [backlink](#)

Unknown interpreted text role "abbr".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 199); [backlink](#)

Unknown interpreted text role "ref".

2. Change how you import from `__init__.py`. See [ref:'update\\_init\\_role'](#).

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 201); [backlink](#)

Unknown interpreted text role "ref".

#### Updating `module_utils`

If any of your custom module utility, once you migrate to a collection you cannot address the module utility in the top level `ansible.module_utils` Python namespace. Ansible does not merge content from collections into the Ansible internal Python namespace. Update any Python import statements that refer to custom module utilities when you migrate your custom content to collections. See [ref:'module\\_utils in collections <collection\\_module\\_utils>'](#) for more details.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 209); [backlink](#)

Unknown interpreted text role "ref".

When coding with `module_utils` in a collection, the Python import statement needs to take into account the `abbr:'FQCN (Fully Qualified Collection Name)'` along with the `ansible_collections` convention. The resulting Python import looks similar to the following example:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 211); [backlink](#)

Unknown interpreted text role "abbr".

```
from ansible_collections.{namespace}.{collectionname}.plugins.module_utils.{util} import {something}
```

#### Note

You need to follow the same rules in changing paths and using namespaced names for subclassed plugins.

The following example code snippets show a Python and a PowerShell module using both default Ansible `module_utils` and those provided by a collection. In this example the namespace is `ansible_example` and the collection is `community`.

In the Python example the `module_utils` is `helper` and the `abbr:'FQCN (Fully Qualified Collection Name)'` is `ansible_example.community.plugins.module_utils.helper`:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 223); [backlink](#)

Unknown interpreted text role "abbr".

```
from ansible.module_utils.basic import AnsibleModule
from ansible.module_utils.common.text.converters import to_text
from ansible.module_utils.six.moves.urllib.parse import urlencode
from ansible.module_utils.six.moves.urllib.error import HTTPError
from ansible_collections.ansible_example.community.plugins.module_utils.helper import HelperRequest

argspec = dict(
    name=dict(required=True, type='str'),
    state=dict(choices=['present', 'absent'], required=True),
)
```

```

module = AnsibleModule(
    argument_spec=argspec,
    supports_check_mode=True
)

_request = HelperRequest(
    module,
    headers={"Content-Type": "application/json"},
    data=data
)

```

In the PowerShell example the `module_utils` is `hyperv` and the `abbr: FQCN (Fully Qualified Collection Name)` is `ansible_example.community.plugins.module_utils.hyperv`:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 249); [backlink](#)**  
Unknown interpreted text role "abbr".

```

#!powershell
#AnsibleRequires -CSharpUtil Ansible.Basic
#AnsibleRequires -PowerShell ansible_collections.ansible_example.community.plugins.module_utils.hyperv

$spec = @{
    name = @{ required = $true; type = "str" }
    state = @{ required = $true; choices = @("present", "absent") }
}
$module = [Ansible.Basic.AnsibleModule]::Create($args, $spec)

Invoke-HyperVFunction -Name $module.Params.name

$module.ExitJson()

```

### Importing from `__init__.py`

Because of the way that the CPython interpreter does imports, combined with the way the Ansible plugin loader works, if your custom embedded module or plugin requires importing something from an `:file: __init__.py` file, that also becomes part of your collection. You can either originate the content inside a standalone role or use the file name in the Python import statement. The following example is an `:file: __init__.py` file that is part of a callback plugin found inside a collection named `ansible_example.community`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 273); [backlink](#)**  
Unknown interpreted text role "file".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev\_guide\[ansible-devel] [docs] [docsite] [rst] [dev\_guide]migrating\_roles.rst, line 273); [backlink](#)**  
Unknown interpreted text role "file".

```

from ansible_collections.ansible_example.community.plugins.callback.__init__ import CustomBaseClass

```

### Example: Migrating a standalone role with plugins to a collection

In this example we have a standalone role called `my-standalone-role.webapp` to emulate a standalone role that contains dashes in the name (which is not valid in collections). This standalone role contains a custom module in the `library/` directory called `manage_webserver`.

```

my-standalone-role.webapp
├── defaults
├── files
├── handlers
├── library
├── meta
├── tasks
├── templates
├── tests
└── vars

```

1. Create a new collection, for example, `acme.webserver`:

```

$ ansible-galaxy collection init acme.webserver
- Collection acme.webserver was created successfully
$ tree acme -d 1
acme
├── webserver
│   ├── docs
│   ├── plugins
│   └── roles

```

2. Create the `webapp` role inside the collection and copy all contents from the standalone role:

```

$ mkdir acme/webserver/roles/webapp
$ cp my-standalone-role.webapp/* acme/webserver/roles/webapp/

```

3. Move the `manage_webserver` module to its new home in `acme/webserver/plugins/modules/`:

```

$ cp my-standalone-role.webapp/library/manage_webserver.py acme/webserver/plugins/modules/manage.py

```

#### Note

This example changed the original source file `manage_webserver.py` to the destination file `manage.py`. This is optional but the `abbr: FQCN (Fully Qualified Collection Name)` provides the `webserver` context as `acme.webserver.manage`.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 326); backlink
```

Unknown interpreted text role "abbr".

4. Change `manage_webserver` to `acme.webserver.manage` in `:file:tasks/` files in the role ( for example, `my-standalone-role.webapp/tasks/main.yml`) and any use of the original module name.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 328); backlink
```

Unknown interpreted text role "file".

#### Note

This name change is only required if you changed the original module name, but illustrates content referenced by `abbr:'FQCN (Fully Qualified Collection Name)'` can offer context and in turn can make module and plugin names shorter. If you anticipate using these modules independent of the role, keep the original naming conventions. Users can add the `ref:'collections keyword <collections_using_playbook>'` in their playbooks. Typically roles are an abstraction layer and users won't use components of the role independently.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 332); backlink
```

Unknown interpreted text role "abbr".

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 332); backlink
```

Unknown interpreted text role "ref".

### Example: Supporting standalone roles and migrated collection roles in a downstream RPM

A standalone role can co-exist with its collection role counterpart (for example, as part of a support lifecycle of a product). This should only be done for a transition period, but these two can exist in downstream in packages such as RPMs. For example, the RHEL system roles could coexist with an [example of a RHEL system roles collection](#) and provide existing backwards compatibility with the downstream RPM.

This section walks through an example creating this coexistence in a downstream RPM and requires Ansible 2.9.0 or later.

To deliver a role as both a standalone role and a collection role:

1. Place the collection in `:file:/usr/share/ansible/collections/ansible_collections/`.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 344); backlink
```

Unknown interpreted text role "file".

2. Copy the contents of the role inside the collection into a directory named after the standalone role and place the standalone role in `:file:/usr/share/ansible/roles/`.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 345); backlink
```

Unknown interpreted text role "file".

All previously bundled modules and plugins used in the standalone role are now referenced by `abbr:'FQCN (Fully Qualified Collection Name)'` so even though they are no longer embedded, they can be found from the collection contents. This is an example of how the content inside the collection is a unique entity and does not have to be bound to a role or otherwise. You could alternately create two separate collections: one for the modules and plugins and another for the standalone role to migrate to. The role must use the modules and plugins as `abbr:'FQCN (Fully Qualified Collection Name)'`.

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 347); backlink
```

Unknown interpreted text role "abbr".

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\dev_guide\[ansible-devel] [docs] [docsite] [rst] [dev_guide]migrating_roles.rst, line 347); backlink
```

Unknown interpreted text role "abbr".

The following is an example RPM spec file that accomplishes this using this example content:

```
Name: acme-ansible-content
Summary: Ansible Collection for deploying and configuring ACME webapp
Version: 1.0.0
Release: 1%{?dist}
License: GPLv3+
Source0: acme-webserver-1.0.0.tar.gz
```

```

Url: https://github.com/acme/webserver-ansible-collection
BuildArch: noarch

%global roleprefix my-standalone-role.
%global collection_namespace acme
%global collection_name webserver

%global collection_dir %{_datadir}/ansible/collections/ansible_collections/%{collection_namespace}/%{collection_name}

%description
Ansible Collection and standalone role (for backward compatibility and migration) to deploy, configure, and manage the AC

%prep
%setup -qc

%build

%install

mkdir -p %{buildroot}/%{collection_dir}
cp -r ./* %{buildroot}/%{collection_dir}/

mkdir -p %{buildroot}/%{ datadir}/ansible/roles
for role in %{buildroot}/%{collection_dir}/roles/*
do
    cp -PR ${role} %{buildroot}/%{_datadir}/ansible/roles/%{roleprefix}${basename ${role}}

    mkdir -p %{buildroot}/%{ pkgdocdir}/${basename ${role}}
    for docfile in README.md COPYING LICENSE
    do
        if [ -f ${role}/${docfile} ]
        then
            cp -p ${role}/${docfile} %{buildroot}/%{_pkgdocdir}/${basename ${role}}/${docfile}
        fi
    done
done

%files
%dir %{ datadir}/ansible
%dir %{_datadir}/ansible/roles
%dir %{_datadir}/ansible/collections
%dir %{ datadir}/ansible/collections/ansible_collections
%{ datadir}/ansible/roles/
%doc %{ pkgdocdir}/*/README.md
%doc %{_datadir}/ansible/roles/%{roleprefix}*/README.md
%{collection_dir}
%doc %{collection_dir}/roles/*/README.md
%license %{ pkgdocdir}/*/COPYING
%license %{_pkgdocdir}/*/LICENSE

```

## Using ansible.legacy to access local custom modules from collections-based roles

Some roles within a collection use `ref: local custom modules <developing_locally>` that are not part of the collection itself. If there is a conflict between the custom module short name and the collection module name, you need to specify which module your tasks call. You can update the tasks to change `local_module_name` to `ansible.legacy.local_module_name` to ensure you are using the custom 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]migrating\_roles.rst, line 416); [backlink](#)**

Unknown interpreted text role "ref".