

# ANSIBLE

## Introduction to Ansible Collections

Ganesh B. Nalawade  
PSE, Ansible Engineering  
Github/IRC: ganeshrn

Abhijeet Kasurde  
SSE, Ansible Engineering  
Github/IRC: akasurde

## Agenda

- What is a Collection?
- What problems collection solve?
- Deep dive with Collections
- Collection demo
- Ansible galaxy for Collections
- Reference for Collection

## About me - Abhijeet

- Free and Open Source software evangelist
- Senior Software Engineer at Ansible by Red Hat
- Works primarily as maintainer for Ansible VMware and cloud related modules
- Co-host for Ansible Pune meetup group

## About me - Ganesh

- Principal Software Engineer at Ansible by Red Hat
- Work primarily as upstream developer in Ansible Networking
- Worked extensively on Network management plane developing software for on/box automation and programmability infra.
- Co Organiser for Ansible Pune meetup group

- Bundle of Ansible content
  - Any kind of plugins
  - modules
  - module\_utils
  - roles
  - playbooks (TBD)
- With a namespace + name

- Difficult distribution of non-role content
- Plugin/role name collisions
- Difficult to do code-sharing for most plugins
- Vendor is the owner of content

## Role Structure

```
myorg/  
├── myrole  
│   ├── action_plugins  
│   │   └── my_action_plugin.py  
│   ├── connection_plugins  
│   │   └── my_connection_plugins.py  
│   ├── filter_plugins  
│   │   └── my_filter_plugin.py  
│   ├── library  
│   │   └── test_module.py  
│   └── module_utils  
│       └── test.py
```

## Collection Structure

```
myorg/  
├── mycollection  
│   ├── action  
│   │   └── my_action_plugin.py  
│   ├── connection  
│   │   └── my_connection_plugin.py  
│   ├── filter  
│   │   └── my_filter_plugin.py  
│   ├── module_utils  
│   │   └── test.py  
│   └── modules  
│       └── test_module.py
```

- Multiple roles with dependencies
  - Maintain version dependencies across roles and Ansible version
- Roles already distribute plugins but the problem is
  - Requires role to be “entered” to use plugin content
  - Plugins are not “first-class citizens” within as compared to shipping plugins in Ansible
  - Versioning/maintenance issues



- Collection may contain 1-N:
  - Roles/Modules/Plugins
  - Playbooks (TBD)
- Collection is the unit of distribution and versioning
- Installable:
  - globally
  - per-user
  - content-adjacent (ie per-project)

- Default paths (searched in this order)
  - (current playbooks)/collections
  - ~/.ansible/collections
  - /usr/share/ansible/collections
- ansible-galaxy can be used to create/install/distribute collections
  - Supported from Ansible 2.9 onwards

```
$ ansible-galaxy collection init my_namespace.my_collection
```

```
$ ansible-galaxy collection build
```

```
$ ansible-galaxy collection install my_namespace-my_collection-1.0.0.tar.gz -p ./collections
```

```
$ ansible-galaxy collection publish path/to/my_namespace-my_collection-1.0.0.tar.gz --api-key=SECRET
```

- Ansible has a global flat plugins namespace
- Role1 and Role2 both distribute my\_module
  - Which one will get used?
  - Can I use both of them in same run? (no)
  - If ansible core later includes a my\_module?
  - What if project has in library/?
  - What if the system has ib /usr/share/ansible?
  - Issue exists for roles and all modules/plugins

- Collections have a namespace and name
  - For distribution, namespace == Galaxy user/org
- Content is collections is accessed by namespace
  - mynamespace.mycollection.my\_module
  - f5.bigip.provisioning\_role
- Built-in collections
  - ansible.builtin
  - ansible.legacy

```
---
- hosts: somehosts
  tasks:
    - myns.mycollection.athing

    - ansible.builtin.ping  # use only the ping packaged in core

    - ansible.legacy.ping  # use core or library(etc)/ping.py
      when: thing | myns.mycollection.filter == 42

    - ping:  # still works, == ansible.legacy.ping:
```

That's awfully verbose

```
---  
- hosts: somehosts  
  collections:  
    - ansible.builtin  
    - myns.mycollection  
    - othersn.othercollection  
  tasks:  
    - mymodule: # first found in list of collections  
    - othersn.otehrcollection.mymodule # fully qualified is fine
```

- Ordered list of collections for unqualified name
- `ansible.legacy` auto-appended
  - if `ansible.builtin` or `ansible.legacy` isn't explicitly listed
  - eases adding collection search to existing plays
- settable on plays/blocks/tasks



- `module_utils` in roles only works for modules
  - inventory plugins can't use `module_utils` code
  - unless you're using ansible shipped code

- Entire collection is a Python namespace package
- All installed collection content is accessible
  - even across collections!
  - `ansible_collections` root Python namespaces
  - `ansible_collections.mynamespace.mycollection`

# DEMO

- [Collection support with ansible-galaxy \(2.9\)](#)

# Contributing to Ansible

- [Community groups](#)
- [IRC channels](#)
- Google Groups:  
<https://groups.google.com/forum/#!forum/ansible-project>  
<https://groups.google.com/forum/#!forum/ansible-devel>
- [Ansible galaxy](#)

# Thank You

GitHub / IRC: @ganeshrn

GitHub / IRC: @akasurde