ANSIBLE AUTOMATES

Ansible Collections,
Future of Ansible Ecosystem

Sean Cavanaugh
Technical Marketing Engineer @IPvSean
Ask the Expert
Agenda

● Ansible Project Changes
  ○ Why?
  ○ Step 1: Separate Ansible from the content
  ○ Step 2: Simplify Ansible Distribution Channels

● On-Prem Automation Hub
● Using an existing collection
● Building your own collection
Ansible Project scope has changed unprecedented growth!
Ansible Project Growth - Side Effects

**Customers**

- Support claim confusion, and who does what and where for issues/PRs?
- Need for stability (longer life cycle) for foundation components while...
- Need new module enablement / platform updates (shorter life cycle)

**Developers**

- 4,300+ open issues, 2,000 open pull requests
- “One size fits all” doesn’t work for all content sets
- Some things go in quickly, some things don’t
Step 1: Separate Ansible from the content

- Ansible Project
- Ansible Content
- Ansible Base
Content is now modular

Ansible Content

Ansible Base

Ansible Project 2.9
Introducing the Ansible Collection

Simplified and consistent content schema

- A standardized way to organize and package Ansible content
- Portable and flexible delivery
- Semantic versioning
Bug/Improvement for module introduced

Ansible Project 2.5

3-4 months

Bug/Improvement now in supported release

Ansible Project 2.6
After collections

Ansible Project 2.9

Bug/Improvement for module introduced

Content can be switched out immediately!

Ansible Project 2.10
What is in an Ansible Collection?

- Components are well defined, there is a standard for the directory structure

- Requires same standard of documentation that the Ansible Project does

- Scaffolding can be created with Ansible Galaxy command
Directory Structure

- **docs/**: local documentation for the collection
- **galaxy.yml**: source data for the MANIFEST.json that will be part of the collection package
- **playbooks/**: playbooks reside here
  - **tasks/**: this holds 'task list files' for include_tasks/import_tasks usage
- **plugins/**: all ansible plugins and modules go here, each in its own subdir
  - **modules/**: ansible modules
  - **lookups/**: lookup plugins
  - **filters/**: Jinja2 filter plugins
  - **connection/**: connection plugins required if not using default
- **roles/**: directory for ansible roles
- **tests/**: tests for the collection's content
### Step 2: Simplify Ansible Distribution Channels

<table>
<thead>
<tr>
<th>Ansible Galaxy</th>
<th>Ansible Automation Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td>galaxy.ansible.com</td>
<td>cloud.redhat.com</td>
</tr>
<tr>
<td>● Community supported</td>
<td>● Certified, jointly supported by Red Hat and Partner</td>
</tr>
<tr>
<td>● Extended to leverage Collections framework</td>
<td>● Access to advanced analytics</td>
</tr>
<tr>
<td>● “Latest and greatest”</td>
<td>● “Slow and steady”</td>
</tr>
</tbody>
</table>
Automation Hub: Certified Collections Today

arista.cvp
arubanetworks.aoscx
azure.azcollection
check_point.mgmt
cisco.aci
    .mso
cyberark.pas
dynatrace_innovationlab.dynatrace_collection
f5networks.f5_modules
fortinet.fortios
google.cloud
ibm.qradar

ibm.ibm_zos_core
    .specture_virtualize
nginxinc.nginx_controller
netapp.aws
    .azure
    .elements
    .ontap
purestorage.flasharray
    .flashblade
rubrikinc.cdm
sensu.sensu_go
splunk.enterprise_security
tirasa.syncope
On-Prem Automation Hub
(Coming Soon)
What about on-premises?

- **Control and Manage**
  Manage, audit, and test your organization’s view of available content

- **Local Secure Mirror**
  Sync single point to cloud.redhat.com to synchronize data
Air gapped?

- **Semantic Downloads**
  collections are versioned *tar.gz* files

- **Installable from Local File**
  Ansible can install *tar.gz* locally

Physical Secured Site
Using an existing Ansible Collection
Install the Ansible Collection:

```
ansible-galaxy collection install f5networks.f5_modules
```

This installs (by default) into:

```
~/.ansible/collections/ansible_collections
```

Now use the content! Use the fully qualified name:

```
namespace.collection.module
```

```
f5networks.f5_modules.bigip_device_info
```
What does this look like with F5 BIG-IP?

```yaml
---
- name: install and start apache
  hosts: f5
  become: yes

  tasks:
  - name: collect big-ip info
    f5networks.f5_modules.bigip_device_info:
      gather_subset:
        - system-info
      provider: "{{login_info}}"
      register: device_facts

  - name: display facts to terminal window
    debug:
      var: device_facts
```
What does this look like with CyberArk?

---

- **name**: `create session with cyberark vault`
  
  **hosts**: localhost

  **tasks**:
  
  - **name**: `Logon - use_shared_logon_authentication`
    
    **cyberark.pas.cyberark_authentication**:
    
    **api_base_url**: "https://www.my-server.com"
    
    **use_shared_logon_authentication**: True
What does this look like with F5 BIG-IP?

---

- **name**: install and start apache
  - **hosts**: f5
  - **become**: yes
  - **collections**:
    - f5networks.f5_modules

**tasks:**
- **name**: collect big-ip info
  - **bigip_device_info**:
    - **gather_subset**:
      - system-info
    - **provider**: "{{login_info}}"
  - **register**: device_facts

- **name**: display facts to terminal window
  - **debug**:
    - **var**: device_facts
Building a new Ansible Collection
Create a collection

ansible-galaxy collection init namespace.collection_name
Create a collection

ansible-galaxy collection init namespace.collection_name

[user@rhel ~] $ ansible-galaxy collection init test.hello_world
- Collection test.hello_world was created successfully
Create a collection

ansible-galaxy collection init namespace.collection_name

[user@rhel ~]$ ansible-galaxy collection init test.hello_world
- Collection test.hello_world was created successfully

[user@rhel8 ~]$ tree test
test
  └── hello_world
      ├── docs
      │   └── galaxy.yml
      ├── plugins
      │   └── README.md
      └── README.md
          roles

ANSIBLE AUTOMATES
Create a modules directory

$ cd test/hello_world
$ mkdir plugins/modules

Write your Python module code here

$ mv my_module.yml plugins/modules
Build and use

Build the Ansible Collection with the new module

$ cd /home/user/test/hello_world
$ ansible-galaxy collection build
Created collection for test.hello_world at /home/user/test/hello_world/test-hello_world-1.0.0.tar.gz

Install the local file collection

$ ansible-galaxy collection install test-hello_world-1.0.0.tar.gz
Process install dependency map
Starting collection install process
Installing 'test.hello_world:1.0.0' to '/home/student1/.ansible/collections/ansible_collections/test/hello_world'
---
- name: run simple playbook
  hosts: localhost
  gather_facts: false

  tasks:
  - name: try new module
test.hello_world.my_module:
    register: my_var

  - name: display output to terminal window
debug:
    var: my_var
Use your new content!
ANSIBLE AUTOMATES

THANK YOU