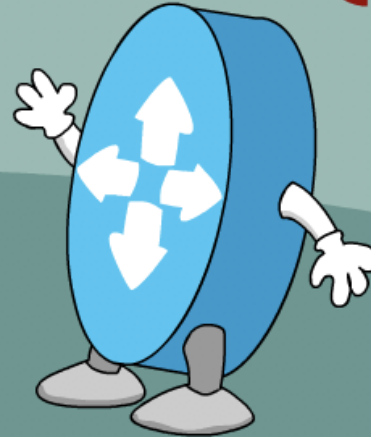




NETDEVOPS {LIVE!}



DEVNET

Better Ansible Network Automation with Roles and Custom Modules

John McDonough

NetDevOps Engineer, DevNet Sandbox

Twitter: @johnamcdonough

Season 2, Talk 7

<https://developer.cisco.com/netdevops/live>

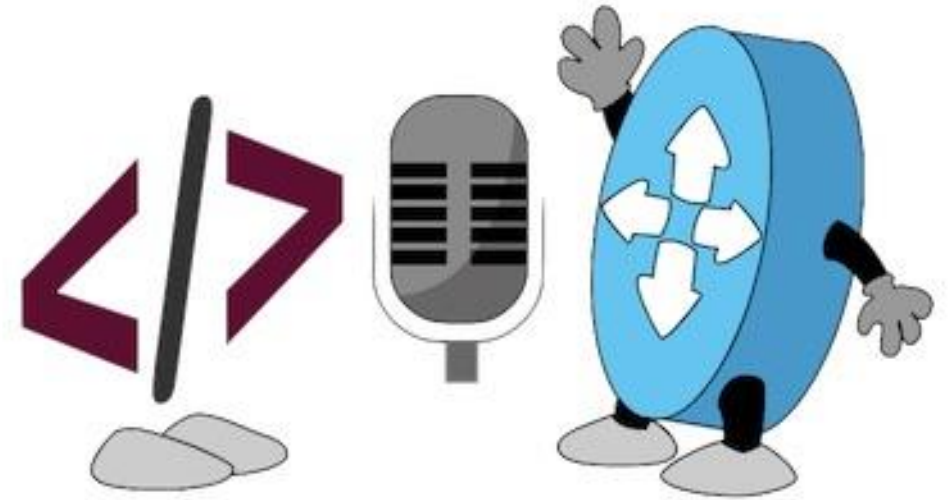


<http://cs.co/ndl>

Help us track NetDevOps Live Interest!

What are we going to talk about?

- Ansible Roles vs. Modules
 - What's the difference?
- Let's Make a Module
 - Typically Python
 - Self Documenting
 - Idempotent
- Let's Make a Role
 - Self-Contained / Portable
 - Loosely Coupled
 - Share with ansible-galaxy



Before we start...

- ✓ I have contributed to Ansible but there is much I don't know.
- ✓ Ansible is evolving, true today could be less true tomorrow.
- ✓ I always happy to know a better way.

*The name refers to the fictional
instantaneous hyperspace
communication system "ansible"*

As featured in Orson Scott Card's Ender's Game

Ansible Roles and Modules

What's the difference?

Roles – The Instructions

- Self-contained and portable
- Written in YAML
- Use for common configurations
- Enforcement of standards
- Called from a playbook

Modules – The Tools

- Do One Thing Well
- Python or PowerShell
- Uses APIs and/or CLI tools
- Abstract complexity from user
- Called by an Ansible task

Let's Make a Module

Let's Make a Module – Python

- Create and Activate a Python Virtual Environment
 - `python -m venv venv`
 - `source venv/bin/activate`
- Install Ansible
 - `pip install ansible`
- Install other requirements as needed
 - `pip install requests`
 - `pip install pylint`

```
NetDevOps Live:src $  
NetDevOps Live:src $ mkdir netdevops  
NetDevOps Live:src $ cd netdevops  
NetDevOps Live:netdevops $  
NetDevOps Live:netdevops $ python3 -m venv venv  
NetDevOps Live:netdevops $  
NetDevOps Live:netdevops $ source venv/bin/activate  
(venv) NetDevOps Live:netdevops $
```

```
(venv) NetDevOps Live:netdevops $  
(venv) NetDevOps Live:netdevops $ pip install ansible  
Collecting ansible  
  Using cached https://files.pythonhosted.org/packages  
0/ansible-2.7.10.tar.gz
```

```
(venv) NetDevOps Live:netdevops $  
(venv) NetDevOps Live:netdevops $ pip install requests  
Collecting requests  
  Using cached https://files.pythonhosted.org/packages  
b/requests-2.21.0-py2.py3-none-any.whl
```

https://docs.ansible.com/ansible/latest/dev_guide/developing_modules_general.html

Let's Make a Module – Module Sections

- Modules have distinct sections
 - Sh-Bang
 - DOCUMENTATION
 - imports
 - main()
 - AnsibleModule Class
 - Argument Spec
 - Required If
 - Supports Check Mode
 - Determine State / Update if required
 - Exit – with result

```
#!/usr/bin/python

DOCUMENTATION = '''
---
module: module_name
author: author_name
short_description: short_description
description:
    - long_description
options:
    option_1:
        description:
            - This is option 1
        required: true
    option_2:
        description:
            - This is option 2
        required: false
state:
    description:
        - Whether the thing should exist or not.
    default: present
    choices: ['present', 'absent']
'''
```

https://docs.ansible.com/ansible/latest/dev_guide/developing_modules_general.html

Let's Make a Module – Development and Testing

- Local Module Testing
 - Ansible modules have an implied JSON contract
 - JSON IN → MODULE ← JSON OUT
- Playbook Module Testing
 - Run with a playbook
- Ensure your code looks good, You might share it.
 - pylint
 - pylint does not respect virtual envs
 - Use \$(which pylint) in virtual env

```
aci_modules ▸ library ▸ {} args.json ▸ ...
1  {
2  |   "ANSIBLE_MODULE_ARGS": {
3  |       "name": "TenantA",
4  |       "descr": "This is tenant A",
5  |       "state": "present"
6  |   }
7  }
```

```
(venv) NetDev0ps Live:library $ $(which pylint) my_aci_tenant.py
***** Module my_aci_tenant
my_aci_tenant.py:63:0: C0413: Import "import json" should be placed
my_aci_tenant.py:64:0: C0413: Import "import requests" should be pl
my_aci_tenant.py:65:0: C0413: Import "import urllib3" should be pla
my_aci_tenant.py:69:0: C0413: Import "from ansible.module_utils.bas
import-position)
my_aci_tenant.py:112:4: W0603: Using the global statement (global-s

-----
Your code has been rated at 9.38/10 (previous run: 9.38/10, +0.00)
```

https://docs.ansible.com/ansible/latest/dev_guide/developing_modules_general.html
https://docs.ansible.com/ansible/latest/dev_guide/testing_sanity.html

Let's Make a Role

Let's Make a Role – Initialization

- Ansible Role Tools
 - `ansible-galaxy init <role-name>`
- defaults – default variables for the role.
- files – files for the role.
- handlers – tasks run at play completion.
- meta – meta data for the role, if publishing on galaxy.ansible.com
- tasks – tasks executed by the role.
- vars – other variables for the role.
- templates – templates for the role.

```
(venv) NetDevOps Live:aci_modules $
(venv) NetDevOps Live:aci_modules $ ansible-galaxy init my_aci_role
- my_aci_role was created successfully
(venv) NetDevOps Live:aci_modules $ ls
LICENSE          ansible.cfg      host_vars       library         site.yml
README.md       group_vars      inventory       my_aci_role
(venv) NetDevOps Live:aci_modules $
```

```
(venv) NetDevOps Live:aci_modules $ tree my_aci_role/
my_aci_role/
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html

Let's Make a Role – Tasks and Variables

- Tasks
 - tasks/main.yml
 - Contains tasks for role
 - Can reference variables from role and from playbook
- Variables
 - default/main.yml
 - If defined nowhere else
 - lowest precedence
 - vars/main.yml
 - Contains variables for the role

```
aci_modules ▸ my_aci_role ▸ tasks ▸ ! main.yml ▸ YAML
12  ▫ - name: ENSURE_TENANT_VRF_EXISTS
13  ▫   aci_vrf:
14  ▫     <<: *login_info
15  ▫     state: "present"
16  ▫     validate_certs: False
17  ▫     tenant: "{{ tenant }}"
18  ▫     vrf: "{{ vrf }}"

aci_modules ▸ my_aci_role ▸ vars ▸ ! main.yml
1  ---
2  # vars file for my_aci_role
3  vrf: "prod_vrf"
4  bd: "prod_bd"
5  gateway: "10.1.100.1"
6  mask: "24"
```

https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html

Let's Make a Role – Call from Playbook

- Call the role from the playbook
- Make use of playbook variables
- Make use of loops
- Make use of YAML anchors and aliases

```
vars:  
  login_info: &login_info  
  hostname: "{{ hostname }}"  
  username: "{{ username }}"  
  password: "{{ password }}"
```

```
<<: *login_info
```

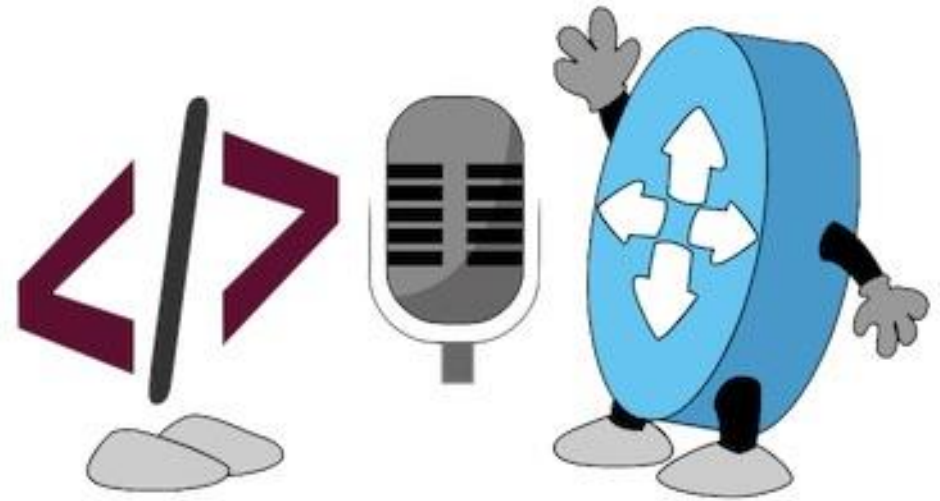
```
- name: Create/Update ACI Tenant  
  my_aci_tenant:  
    <<: *login_info  
    name: "{{ item.tenant }}"  
    descr: "{{ item.descr }}"  
    state: present  
    loop:  
      "{{ tenants }}"  
  
- name: Add Networking  
  include_role:  
    name: my_aci_role  
  vars:  
    tenant: "{{ item.tenant }}"  
  loop:  
    "{{ tenants }}"
```

https://docs.ansible.com/ansible/latest/user_guide/playbooks.html

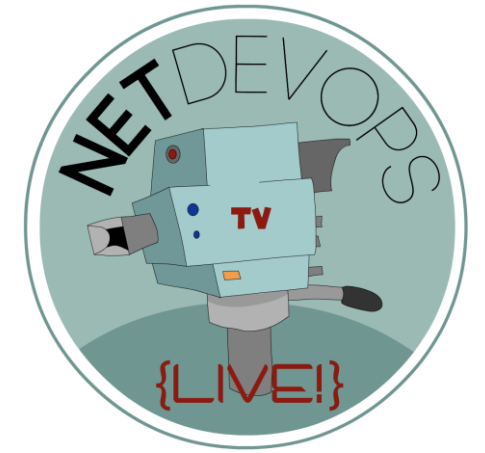
Summing up

What did we talk about?

- The Difference between Ansible Modules and Roles
 - Tasks oriented
 - Playbook oriented
- How to Create an Ansible Module
 - With Python
 - Other languages can be used
- How to Create an Ansible Role
 - For reuse
 - Consistency in Deployments



Webinar Resource List



- Learning Labs
 - Introduction to Ansible
<https://developer.cisco.com/learning/modules/sdx-ansible-intro>
 - ACI and Ansible
<https://developer.cisco.com/learning/modules/ansible-aci-intro>
- DevNet Sandboxes
 - ACI Always On <https://snadboxapicdc.cisco.com>
- Code Samples
 - Ansible on DevNet's Code Exchange
<https://developer.cisco.com/codeexchange/explore/#search=ansible>
 - Webinar Code
https://developer.cisco.com/codeexchange/github/repo/movinalot/aci_modules

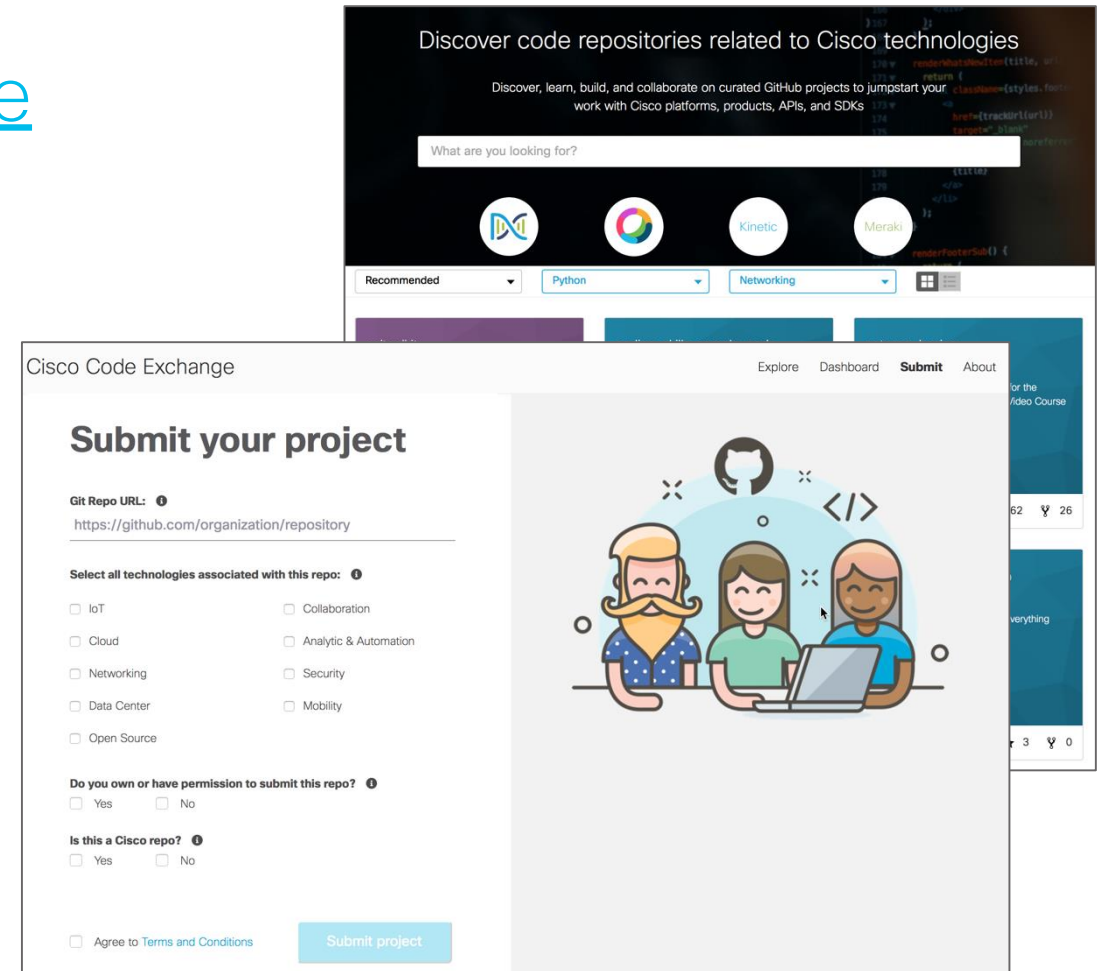
NetDevOps Live! Code Exchange Challenge

developer.cisco.com/codeexchange

**Take what you've seen today
and apply to an active project.
Then submit it to Code
Exchange!**

*Example: Create an Ansible Module for a
playbook task that currently uses the shell or
command modules.*

*Example: Create an Ansible Role for a task or
set of tasks common to several playbooks.*



The image shows two overlapping screenshots of the Cisco Code Exchange website. The top screenshot displays a search interface with the heading "Discover code repositories related to Cisco technologies". It includes a search bar with the placeholder text "What are you looking for?", several technology icons (including Docker, Kubernetes, Kinetic, and Meraki), and filter buttons for "Recommended", "Python", and "Networking". The bottom screenshot shows the "Submit your project" form. The form includes a "Git Repo URL" field with the example "https://github.com/organization/repository". Below this is a section titled "Select all technologies associated with this repo:" with checkboxes for IoT, Cloud, Networking, Data Center, Open Source, Collaboration, Analytic & Automation, Security, and Mobility. There are also checkboxes for "Do you own or have permission to submit this repo?" and "Is this a Cisco repo?". At the bottom of the form, there is a checkbox for "Agree to Terms and Conditions" and a blue "Submit project" button. To the right of the form is an illustration of three people (two men and one woman) working together at a laptop, with a GitHub logo and code symbols above them.

Looking for more about NetDevOps?

- NetDevOps on DevNet developer.cisco.com/netdevops
- NetDevOps Live! developer.cisco.com/netdevops/live
- NetDevOps Blogs blogs.cisco.com/tag/netdevops
- Network Programmability Basics Video Course developer.cisco.com/video/net-prog-basics/



Got more questions? Stay in touch!

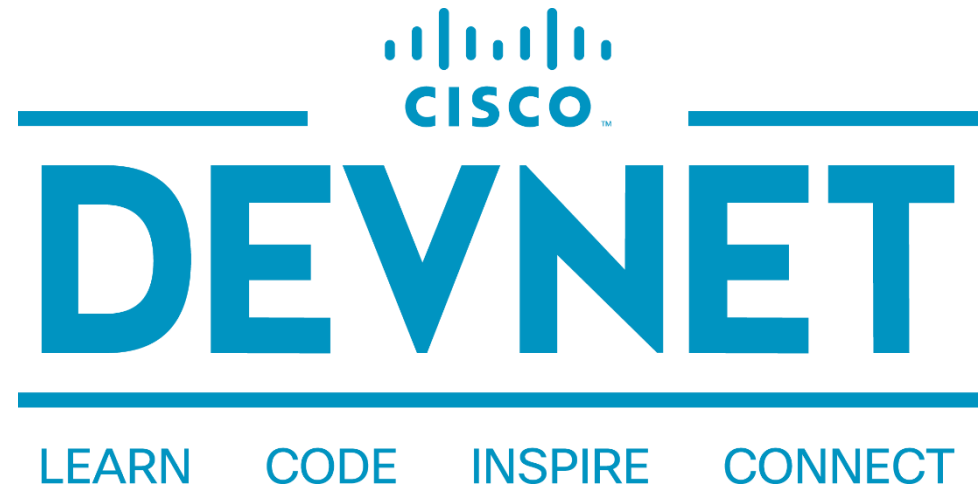


John McDonough

 jomcdono@cisco.com

 [@johnamcdonough](https://twitter.com/johnamcdonough)

 <http://github.com/movinalot>



developer.cisco.com

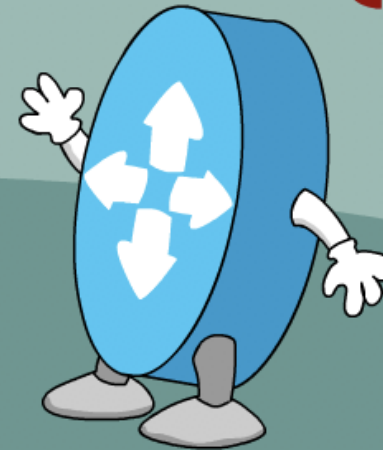
 [@CiscoDevNet](https://twitter.com/CiscoDevNet)

 facebook.com/ciscocodevnet/

 <http://github.com/CiscoDevNet>



NETDEVOPS {LIVE!}



DEVNET

<https://developer.cisco.com/netdevops/live>

@netdevopslive 