

Writing Ansible modules (do the harder stuff in Python)

Dec 14, 2023

Colin Vallance
Tim Way



CDW AMPLIFIED™
SERVICES

Intros



Colin Vallance

- Principal Technical Architect and programmability team lead at CDW
- Avid Python user since 2013 (at least) after having a fight with a bash script
- Personal life filled with cats, amateur radio, electronics projects, and lately a payphone



Tim Way

- Senior Consulting Engineer on CDW's Digital Velocity team
- Regular Open-source contributor (RedHat, Docker, and more..)
- Wide range of experience with software languages, networking, infrastructure, and Linux in general
- Loves Docker Podman



Sponsor Thanks



Code Reference

<https://github.com/cdwlabs/ChiPy-Decoder-Ring>



Ansible



What is Ansible

“Ansible® is an open source, command-line IT automation software application written in Python. It can configure systems, deploy software, and orchestrate advanced workflows to support application deployment, system updates, and more.”

- Uses SSH for transport*
- Human readable language (YAML)
- Keeps secrets safe with Ansible vault
- Hundreds of integrations (Infrastructure, Networks, Cloud, DevOps Tools, Security)

Multiple Flavours

Community Ansible

- ansible-core Python package
- Open Source
- CLI driven
- Runs on most* OSes with Python

Ansible Automation Platform

- Formerly Ansible Tower
- Subscription based product
- High availability
- Containerized execution environments
- GUI, REST API, and task engine

Ansible Terminology and Ecosystem

- Playbook
- Collections
- Roles
- Galaxy
- Plugins
 - Module

Playbook



- “Basic unit” of Ansible
- List of instructions to go and manage a piece of infrastructure
- List of plays, each play being able to target potentially different infrastructure and take separate actions
 - Frequently playbooks are a single play

Collections and Roles

Collection

- Distribution format for Ansible content
 - Can include playbooks, roles, modules, and plugins
- Provides a way to organize and share in a modular/versioned manner
- Makes installation available through a distribution server

Role

- A way of organizing tasks, handlers, variables, and other related content in a structured manner
- Reusable and modular unit of automation that can be shared
- Can be used to perform a specific function
 - configure a server, install a package, etc.
- More focused on a specific function than a collection

Galaxy

- Web-based repository for Ansible roles and collections
- Open source
- Namespaces often used by vendors
 - Microsoft, Amazon, Cisco, etc.
- Certified collections exist for AAP

```
[colin@22sMSTweb-01 ~]$ansible-galaxy --help
usage: ansible-galaxy [-h] [--version] [-v] TYPE ...

Perform various Role and Collection related operations.

positional arguments:
  TYPE
    collection  Manage an Ansible Galaxy collection.
    role        Manage an Ansible Galaxy role.

optional arguments:
  --version      show program's version number, config file location, configured module search path, module location,
                executable location and exit
  -h, --help     show this help message and exit
  -v, --verbose  Causes Ansible to print more debug messages. Adding multiple -v will increase the verbosity, the
                builtin plugins currently evaluate up to -vvvvvv. A reasonable level to start is -vvv, connection
                debugging might require -vvvv.
[colin@22sMSTweb-01 ~]$
```

Plugins

Ansible Included Examples

- Action
- Become
- Cache
- Callback
- Cliconf
- Connection
- Docs fragments
- Filter
- Httpapi
- Inventory
- Lookup
- Modules
- Module utilities
- Netconf
- Shell
- Strategy
- Terminal
- Test
- Vars

Modules

“Modules (also referred to as “task plugins” or “library plugins”) are discrete units of code that can be used from the command line or in a playbook task. Ansible executes each module, usually on the remote managed node, and collects return values. In Ansible 2.10 and later, most modules are hosted in collections.” - [Ansible Docs: Introduction to modules](#)

```
ansible webservers -m service -a "name=httpd state=started"
ansible webservers -m ping
ansible webservers -m command -a "/sbin/reboot -t now"
```

All modules are plugins but not all plugins are modules

Pause



Example Problem

The IT leadership at Drink-More-Water Ltd. have standardized on a naming convention to maintain consistency throughout their environment. As a DMW staff engineer you want to be able to use Ansible to automate deployments/changes/etc. but this crazy standard makes it really hard!

- How does an engineer solve this issue?
- How do we all get a raise?
- How do we get this fictional IT director to do a stint in engineering so they understand our pain?

Python is (mostly) our answer

Naming Convention

- Must be NetBios compatible
 - 15 ASCII characters max
 - Invalid characters: /: *?"<>|
- Building ID
 - Must be zero padded
 - Official source of truth is a CSV in the company share
- Time Zone
 - The time zone, derived from the building address, must be included
- Entity
 - 3:7 characters are allowed
 - This is usually a device name
- Device function
 - Official functions: server, network, virtualized, app, or other
 - Must abbreviate function to the first letter
- Component
 - This can be whatever the user wants
 - Only allowed to use remaining characters to remain NetBios compatible

Building Data

Building Name, Building ID, Address

HQ, 1, "625 W Adams St, Chicago, IL 60661, United States"

Denver, 22, "206 E 13th Ave, Denver, CO 80203"

Grape City, 7, "3070 Limestone Way Suite C, Paso Robles, CA 93446"

Targhee, 15, "3300 Ski Hill Rd, Alta, WY 83414"

Vancouver, 53, "530 Robson St, Vancouver, BC V6B 2B7, Canada"

Wren, 82, "94 St Aldate's, Oxford OX1 1BT, United Kingdom"

Aran, 2, "2 Chome-5-8 Higashishinagawa, Shinagawa City, Tokyo 140-0002, Japan"

Greenhouse, 9, "2655 South Park Ave, Buffalo, NY 14218"

Denver Web Server

```
-----  
| Wow, that's a weird hostname... |  
=====
```

^ ^
-- --
(oo)\-----
(__)\)\ ^
 ||-----w |
 || ||

```
[colin@22sMSTweb-01 ~/projects/cool-python-stuff]$
```

{building id}{device function}{time zone}{entity}{component}

One Answer

Ansible logic *can* accomplish this task.

- Complicated loops and logic
- Non-intuitive
- Could still need custom code
- Difficult to test
- Logic or separate playbook needed in each space this is used



Our Answer: Use Python!



- Python logic is as flexible as you want
 - If you can dream it, you can do it
- Single module everywhere
- Robust testing
 - “Test what you own”

Solutions: Package vs. Module Utilities

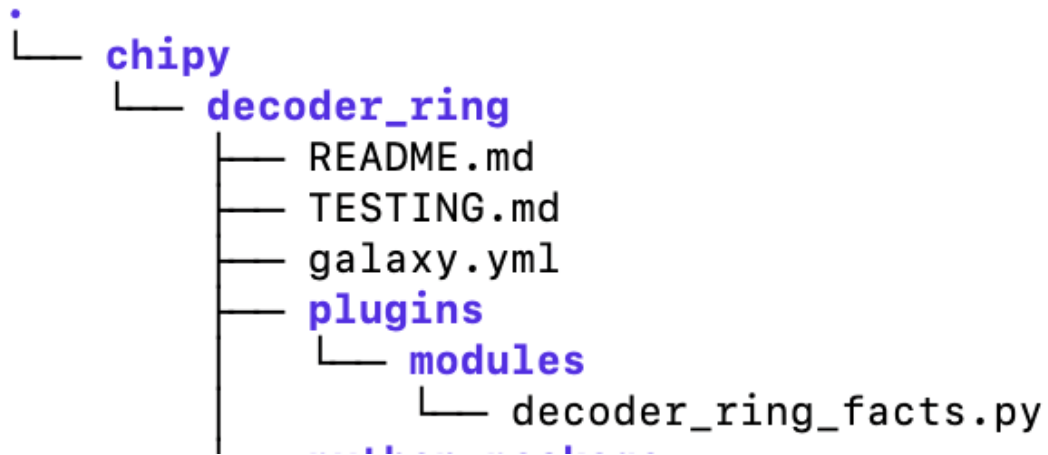
Collection Module

- Fits in to mature Python development
 - Tests
 - Packaging
 - Publishing
- Our solution example

“Bare” Python

- Fast
- Approachable
- Works around maturity restrictions

Anatomy of an Ansible Plugin



- Typically extends or uses an ansible-core class
- Plugins **can** be very short pieces of largely glue code against libraries (SDKs) or complex full implementations
- File naming of modules and integration tests are important

Module: import and setup

```
import dmw_decoder
from ansible.module_utils.basic import AnsibleModule
from ansible.module_utils.common.parameters import env_fallback

def run_module():
    # define available arguments/parameters a user can pass to the module
    module_args = dict(
        api_key=dict(
            type="str", fallback=(env_fallback, ["DMW_DECODER_API_KEY"]), required=True
        ),
        building_id=dict(type="str", required=True),
        component=dict(type="str", required=True),
        device_function=dict(type="str", required=True),
        entity=dict(type="str", required=True),
    )
```

Module: AnsibleModule instance

```
# the AnsibleModule object will be our abstraction working with Ansible
# this includes instantiation, a couple of common attr would be the
# args/params passed to the execution, as well as if the module
# supports check mode
module = AnsibleModule(argument_spec=module_args, supports_check_mode=True)

decoder = dmw_decoder.Decoder(api_key=module.params["api_key"])
dmw_decoder_hostname = decoder.create_netbios_compatible_name(
    module.params["building_id"],
    module.params["device_function"],
    module.params["entity"],
    module.params["component"],
)

result = dict(
    ansible_facts=dict(dmw_decoder_hostname=dmw_decoder_hostname),
    changed=False,
)
```


Module: return JSON

```
# if the user is working with this module in only check mode we do not
# want to make any changes to the environment, just return the current
# state with no modifications
# if module.check_mode:
#     module.exit_json(**result)

# in the event of a successful module execution, you will want to
# simple AnsibleModule.exit_json(), passing the key/value results
module.exit_json(**result)

def main():
    run_module()

if __name__ == "__main__":
    main()
```

Assumptions

- Ansible is installed
- Our package, dmw_decoder, is installed

Exploration



Potholes

Decisions were made

- CSV file for institutional data
 - Included in package
- Secrets handling
- Oddly written Python package
- Monorepo-like



"Other" Dependencies



- Ansible has a support matrix
 - Control Node
 - Managed Node
- AAP2 Requirements

Ansible Support Matrix

Version	Support	End Of Life	Control Node Python	Target Python / PowerShell
2.16	GA: 06 Nov 2023 Critical: 20 May 2024 Security: Nov 2024	May 2025	Python 3.10 - 3.12	Python 2.7 Python 3.6 - 3.12 Powershell 3 - 5.1
2.15	GA: 22 May 2023 Critical: 06 Nov 2023 Security: 20 May 2024	Nov 2024	Python 3.9 - 3.11	Python 2.7 Python 3.5 - 3.11 PowerShell 3 - 5.1
2.14	GA: 07 Nov 2022 Critical: 22 May 2023 Security: 06 Nov 2023	20 May 2024	Python 3.9 - 3.11	Python 2.7 Python 3.5 - 3.11 PowerShell 3 - 5.1

https://docs.ansible.com/ansible/latest/reference_appendices/release_and_maintenance.html#ansible-core-support-matrix

Know your install

Distro	Version	Python	Pkg Mgr	Pip
Debian	12	3.11.2	core 2.14.3	core 2.16.2
Ubuntu	Jammy	3.10.12	ansible 2.10.8	core 2.16.2
Alpine	3.19.0	3.11.6	core 2.16.1	core 2.16.2
Rocky	9	3.9.18	core 2.14.9	core 2.15.8
RHEL*	8	3.9.18	core 2.15.3	core 2.15.8

Resources

- Example GitHub Repo
 - <https://github.com/cdwlabs/ChiPy-Decoder-Ring/>
- Ansible Documentation
 - <https://docs.ansible.com/>

Contact

Tim

TimWay@cdw.com

- GitHub
 - @timway

Colin

Colin.Vallance@cdw.com

- GitHub
 - @crvatwork
 - @crvallance
- ChiPy Slack

Questions?



Thank You!

