DAY 1

- Welcome and introduction
- Recap of Ansible basics & Ansible basic training
  - Short quiz
  - Variable precedence
  - Error Handling - "block, rescue", "handlers"
- Best practices and use of modules (introduction to some common modules)
  - shell vs command
  - Privilege Escalation
  - Asynchronous Actions & Polling
  - Delegation, Local Actions

DAY 2

- Best practices and use of modules (introduction to some common modules)
  - Prompts / Start with / Step
  - File based configuration
    - Manage configuration in files instead of command line parameters
  - Dynamic inventory
    - Generation of *inventory files*, their advantages and uses
- Templates / Jinja2
  - Syntax
  - Hands-on exercise on *templates*
  - Logging / Reporting (some ideas), incl. troubleshooting
- Credential management
  - Ansible Vault
  - Hashicorp Vault

DAY 3

- Ansible Galaxy / Collections
  - Building generic roles & collections
  - Hands-on exercise to create your own collection
  - Use meta/main.yml and meta/requirements.yml
  - optional: Dynamic groups
- Custom plugins
  - Create your own plugin (What types of plugins are there?)
  - Hands-on - Create your own filter
- Custom modules
  - How to create your own module
  - Best practices - *when* does it make sense to create a module?
  - (optional) Hands-on - create your own module
- Feedback and Conclusion

OPITZ CONSULTING

ANSIBLE COLLECTIONS MOLECULE

Ansible Advanced Training

01

SHARED ROLES

02

COLLECTIONS

03

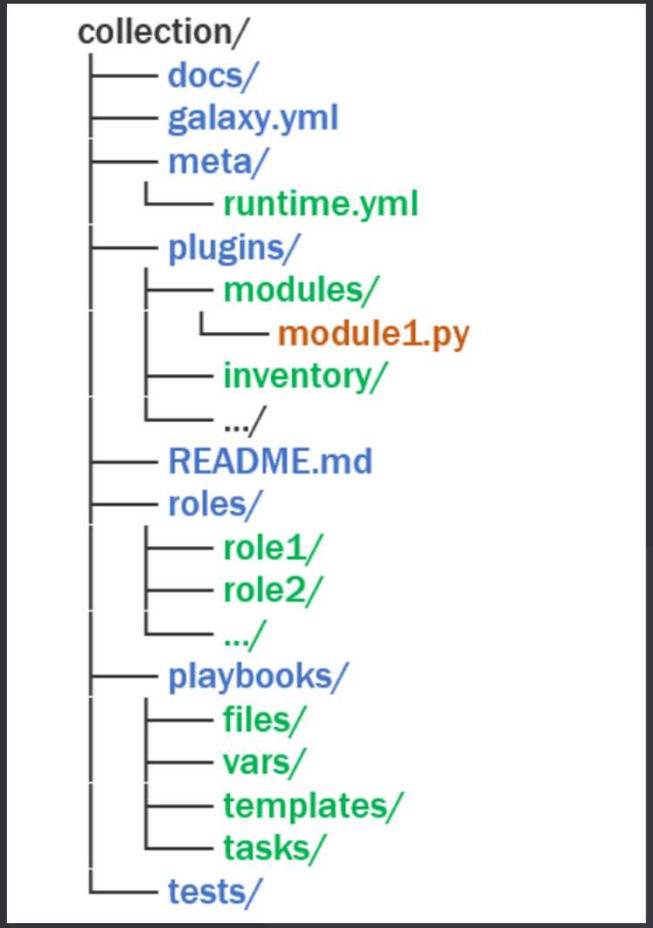
MOLECULE

02 COLLECTIONS

- What is a collection?
- Use of a public collection
- Build your own collection

OVERVIEW

- Collection of roles, modules, plugins and docs
- Follows the idea of a marketplace
  - Ansible Galaxy
- Focus on reusable roles
  - Share logic of your playbook roles
  - Share internal or external



CONFIGS / IMPORTANT KEYWORDS FOR COLLECTIONS

Key	Description
namespace	The namespace of the collection.
name	Name of the collection.
version	The version of the collection.
readme	A relative path to the readme file.
authors	A list of the collection's authors.
description	A summary of the collection.
license	A licence or a list of licences for collection content.
license file	A relative path to the licence file for the collection.
tags	A list of tags used for indexing/searching.
dependencies	A dictionary of data is required for the collection to be usable.
repository	A URL to the SCM repository.
documentation	A URL to any online documentation.
homepage	A URL to the homepage of the collection/project.
issues	A URL to the collection's issue tracker.

03 TESTING / MOLECULE

- Testing of Ansible Playbooks

WHY TESTING

- Test your playbook against multiple architectures / OS
  - Ubuntu, RHEL, SLES, ...
- Docker / Containers can help us to test playbooks
  - Locally
  - In a pipeline
- Test a playbook for idempotency
  - Can I run my Playbook twice?
- Use in combination with linting
  - You can integrate both in your pipeline

Console

```
# Run Docker test containers
docker run --name test1 -it geerlingguy/docker-ubuntu2204-ansible bash
docker run --platform linux/x86_64 --name test2 -it geerlingguy/docker-centos8-ansible bash
```

Inventory / hosts file

```
# Hosts / inventory
[containers]
test1 ansible_connection=docker
test2 ansible_connection=docker
```

Start / test your playbook

```
# Start / test your playbook against
ansible-playbook site.yml -v -i hosts
```

HOW MOLECULE WORKS

https://medium.com/opstree-technology/how-to-test-ansible-playbook-role-using-molecules-with-docker-b428a7f790d0

WHAT MOLECULE CAN HELP US

- Wrapper around your Playbooks
- Uses Ansible / Playbooks itself (python)
- Test of
  - Collection
  - Roles
- Testing
  - Initial Installation
  - Check idempotency
  - Side effect
    - Failover simulation, etc.
  - Validation / test
  - Cleanup of Testenvironment

- Driver (Provider)
  - Runtime environment for tests
  - Docker, Podman, KubeVirt, ...
- Scenario
  - e.g. VM Setup + Database installation
- Command
  - To start target environment
  - e.g. Docker command
- Verifier
  - Validation of the playbook run
- Phases of molecule
  - create (provisioning of environment)
  - converge (test of your playbook)
  - verify (check of your playbook run)

MOLECULE IN ACTION