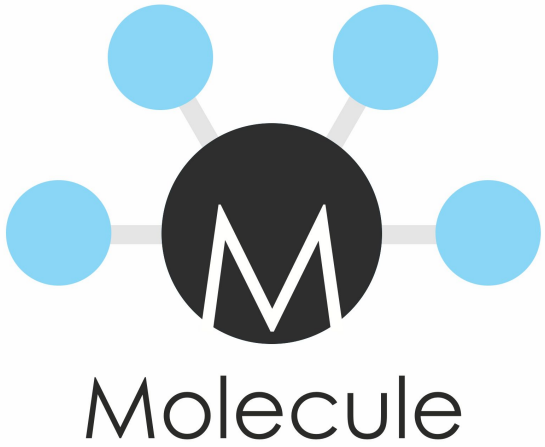
Testing Infrastructure with Molecule

Automated Testing of Ansible Infrastructure as Code

Todd Wardzinski
Senior Consultant

Qadeer Khan
Senior Consultant

Chris Kuperstein
Architect

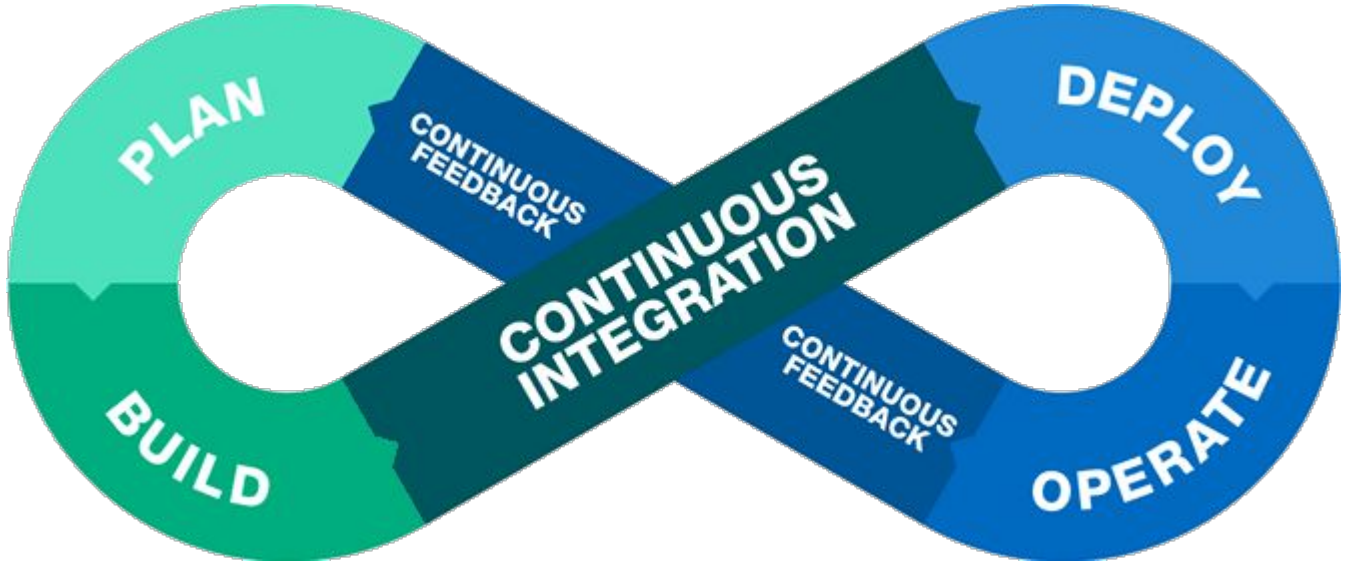


Molecule provides support for testing with multiple instances, operating systems and distributions, virtualization providers, test frameworks and testing scenarios.

Red Hat

The DevOps Story:

Development, operations... and everything in between.



Red Hat

What you need to know

- Ansible (good)
- Python (great)

- Docker (amazing) AND/OR
- Vagrant (good) AND/OR
- Your own virtual infrastructure (this works too!)

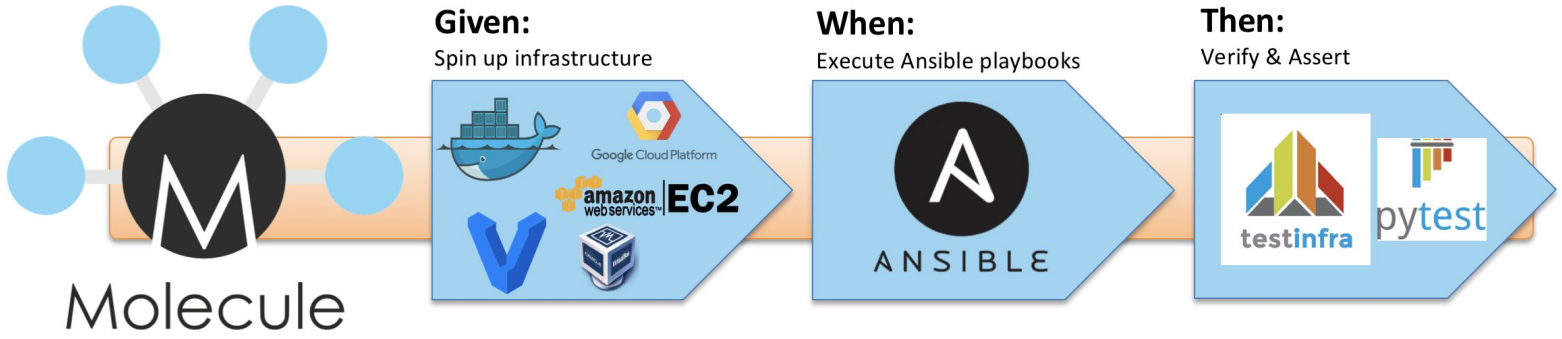
Under the Hood

--> **Test matrix**

└── default

- ├── **lint**
- ├── cleanup
- ├── destroy
- ├── dependency
- ├── syntax
- ├── **create**
- ├── prepare
- ├── **converge**
- ├── idempotence
- ├── side_effect
- ├── **verify**
- ├── cleanup
- └── **destroy**

- **Lint**: Check your code for best practices, conventions (yes/no vs. true/false), don't use command/shell modules, please be idempotent, etc. (**ansible-lint**)
- **Create**: Deploys container/vm to target infra
- **Converge**: Execute the role against the resource
- **Verify**: Validate and assert states (**TestInfra**)
- **Destroy**: Clean up leftover stuff after testing



Source:

IJonas Hecht https://blog.codecentric.de/en/2018/12/test-driven-infrastructure-ansible-molecule/

Molecule Usage https://molecule.readthedocs.io/en/latest/usage.html

Configuring Your Testing Environment

- **Configuration is Central.**
  - Utilizing molecule.yml we can define a multitude of items.
  - For instance, configuring two platforms to test against for a single playbook.
  - Other options include drivers, provisioners and verifiers.
- **Multiple Machines Configured At Once**
  - By configuring platforms we're more easily able to test against multiple platforms at once.
- **Test Runners**
  - Verifier has two options to test with including testinfra and ansible.
    - Testinfra tests are written utilizing Python
    - Ansible tests are written using standard YAML.

molecule > default > molecule.yml > ...

```yaml
---
dependency:
  name: galaxy
driver:
  name: podman
platforms:
  - name: centos7
    image: centos:7
    pre_build_image: true
    privileged: true
  - name: centos8
    image: centos:latest
    privileged: true
provisioner:
  name: ansible
verifier:
  name: testinfra
```

Assertions

- Provide the backbone to the tests, ensuring that functionally the playbooks do what they're supposed to do.
- Many expressive ways to write example tests.
  - For instance, the **host.package** class provides ***is_installed*** to test whether the package is present or not.
- Referencing documentation provides best source for constructing your tests.
  - https://testinfra.readthedocs.io/en/latest/

```
@pytest.mark.parametrize('pkg', [
    'httpd',
    'firewalld'
])
def test_pkg(host, pkg):
    package = host.package(pkg)
    assert package.is_installed
```

Quick Demo Time / Q&A

- Scenario #1
  - Playbook
    - A playbook needs to install packages httpd, firewalld.
    - A configuration file needs to be created in /tmp called demo.txt.
  - Molecule
    - We need to validate packages and files are created.
- Scenario #2
  - Playbook
    - We have an existing role that has been created.
      - Role installs nginx, wget.
  - Molecule
    - We need to apply molecule to this role to facilitate testing.

Thanks!

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat

References

- Getting Started with Ansible / Molecule
  - Getting Started Guide — Molecule 3.1.0a1.dev5 documentation
- TestInfra Documentation
  - Testinfra test your infrastructure — testinfra 5.3.2.dev0+gcd485d4.d20200903 documentation
- Molecule Tricks
-