2019 Michigan IT Symposium

Automated Testing and Deployment of Infrastructure and Applications using Ansible and Molecule

2019 Michigan IT Symposium

Molecule

> whoami

- Been in Ann Arbor pretty much my entire life
- Studied Electrical Engineering and graduated in 2017, but found myself really enjoying my student jobs in IT
- Now a member of the DevOps team in HITS (*check out our poster!*)
- Kubernetes and Ansible (and Molecule) Contributor ™

www.github.com/mjlshen/talks

Do you believe in the need for configuration-as-code tools like Ansible?

Four Things We Care About*

1. Security
2. Availability
3. Resource Management
4. Service Discovery

*2017 Kelsey Hightower: Kubernetes Federation

Two Things Ansible Can Help With

1. Security
2. Availability
3. ~~Resource Management~~
4. ~~Service Discovery~~

*2017 Kelsey Hightower: Kubernetes Federation

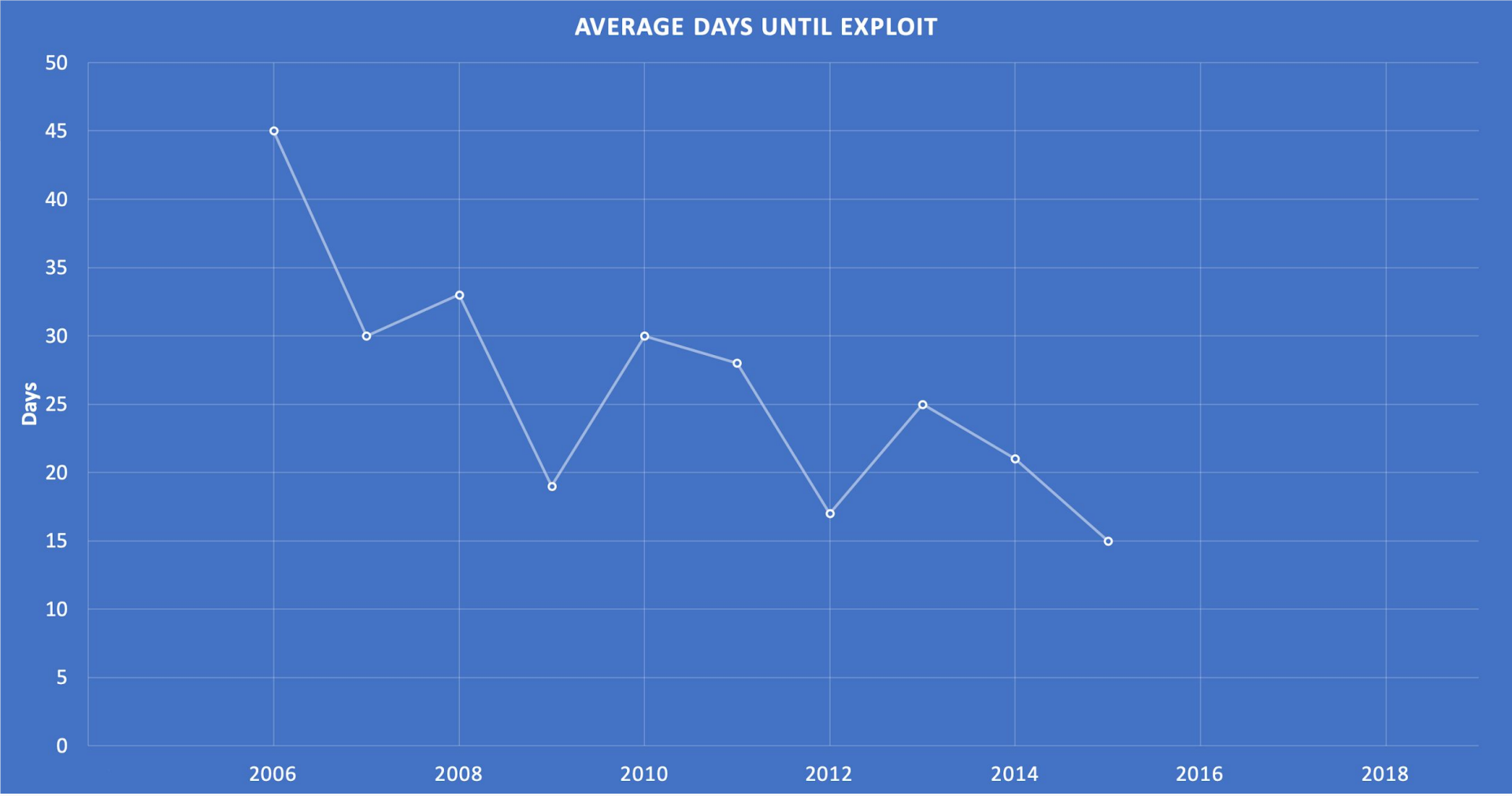
Four Things We Still Should Care About*

1. Security
2. Availability
3. Resource Management
4. Service Discovery

*2017 Kelsey Hightower: Kubernetes Federation

Threats to Our Configuration

- The Software Changes
- The Environment Changes



*2016 IBM X-Force/Gartner Research

Misconfigurations as a Security Risk

- Human, manual configuration continues to be a major risk in cybersecurity
- Ansible allows varying teams, such as security and operations teams, to use the same toolset
  - Consistent, repeatable, and secure environments can be deployed with collaboration using Ansible
- The cloud environment is even more complex

> molecule --help

“Molecule is designed to aid in the development and testing of Ansible roles.

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

Molecule encourages an approach that results in consistently developed roles that are well-written, easily understood and maintained.”

> molecule --help

"Molecule is designed to aid in the development and testing of Ansible roles.

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

Molecule encourages an approach that results in consistently developed roles that are well-written, easily understood and maintained."

> molecule --help

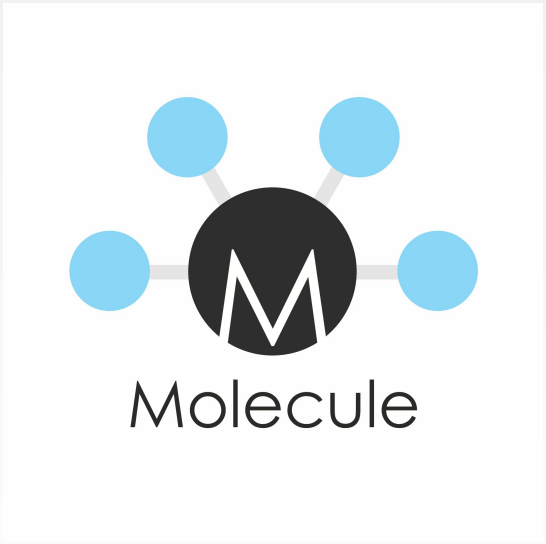
“Molecule is designed to aid in the development and testing of Ansible roles.

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

Molecule encourages an approach that results in consistently developed roles that are well-written, easily understood and maintained.”

Installation

- https://molecule.readthedocs.io/en/stable/installation.html
- For the tools you need to follow along here:
  - Python, Docker Desktop, Virtualbox, Vagrant
  - pip3 install --user ansible molecule ‘molecule[vagrant]’ ‘molecule[docker]’



> molecule init role --role-name it_symposium

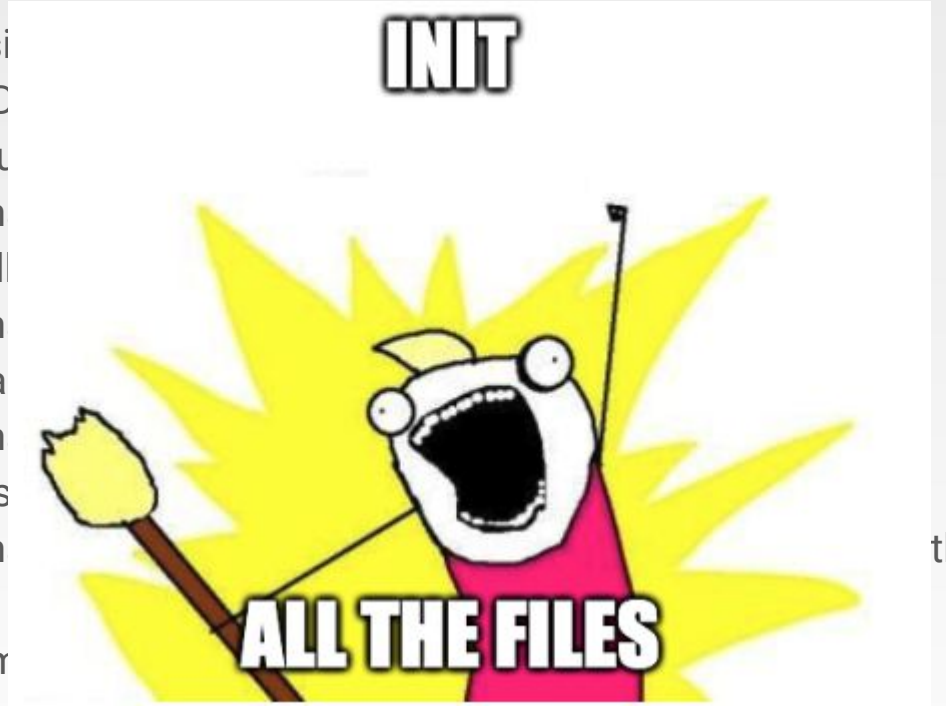
```
it_symposium
├── README.md
├── defaults
│   └── main.yml
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
└── vars
    └── main.yml
```

```
it_symposium
├── molecule
│   └── default
│       ├── Dockerfile.j2
│       ├── INSTALL.rst
│       ├── molecule.yml
│       ├── playbook.yml
│       └── tests
│           ├── __pycache__
│           │   └── test_default.cpython-37.pyc
│           └── test_default.py

9 directories, 12 files
```

> molecule init role --role-name it_symposium

```
it_symposi
├── READ
├── defau
│   └── m
├── handl
│   └── m
├── meta
│   └── m
├── tasks
│   └── m
└── vars
    └── m
```



thon-37.pyc

9 directories, 12 files

> molecule init role --role-name it_symposium

```
it_symposium
├── README.md
├── defaults
│   └── main.yml
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
└── vars
    └── main.yml
```

```
it_symposium
├── molecule
│   └── default
│       ├── Dockerfile.j2
│       ├── INSTALL.rst
│       ├── molecule.yml
│       ├── playbook.yml
│       └── tests
│           ├── __pycache__
│           │   └── test_default.cpython-37.pyc
│           └── test_default.py
```

> molecule test

```
└── default
    ├── lint
    ├── dependency
    ├── cleanup
    ├── destroy
    ├── syntax
    ├── create
    ├── prepare
    ├── converge
    ├── idempotence
    ├── side_effect
    ├── verify
    ├── cleanup
    └── destroy
```

> molecule test

```
└── default
    ├── lint
    │   ├── dependency
    │   ├── cleanup
    │   ├── destroy
    │   └── syntax
    ├── create
    │   └── prepare
    ├── converge
    ├── idempotence
    │   ├── side_effect
    │   └── verify
    ├── cleanup
    └── destroy
```

Demo

molecule 101

> molecule test

> molecule lint

What did we get?

- Walk-through of a development workflow with Molecule
- Our configuration follows best-practice syntax guidelines
  - Ignored rules that don’t apply to us
- Proof that our Ansible role works without changing our test environment
- Proof that our Ansible role is safe to run over and over
  - Idempotence

Demo

Controlled Deployment and Updates of a Web Application - Sonatype Nexus

> molecule converge

> molecule login

> molecule destroy

What did we learn?

- Proof that our Ansible Role to initialize Sonatype Nexus works on CentOS 6 and CentOS 7
- Proof that our Ansible Role can upgrade versions of Sonatype Nexus
  - What happens when the next newest version is released?

What did we learn?

- Proof that our Ansible Role to initialize Sonatype Nexus works on CentOS 6 and CentOS 7
- Proof that our Ansible Role can upgrade versions of Sonatype Nexus
  - What happens when the next newest version is released?
  - What happens when a new team member is tasked with the upgrade?

Limitations of Molecule

- Specialized hardware can be difficult/impossible to emulate in Docker containers or VMs
  - Network Infrastructure
  - Drivers
- That does not stop me from using Ansible to manage these objects though!

Benefits of Molecule

- Easy onboarding to existing Ansible roles
  - https://gitlab.umich.edu/mishen/ansible-role-crashplan
- Integration into CI/CD (Continuous Integration/Continuous Delivery) pipelines

Benefits of Molecule

- Easy onboarding to existing Ansible roles
  - https://gitlab.umich.edu/mishen/ansible-role-crashplan
- Integration into CI/CD pipelines
  - The most important of all

Research* Says...

- While maturity models are very popular in the industry, maturity models are not the appropriate tool to use or mindset to have
  - Encourages vanity metrics tied to maturity models without relating it to customer outcomes
  - A “mature” state that means something different for each team
- What is important is enabling teams to make changes to their products or services without depending on other teams of systems
  - Loosely coupled architecture enables scaling
  - Simplifying complex, painful deployments - key contributor to burnout

*2018 Accelerate by Nicole Forsgren

Recommended Resources

- Books
  - Ansible for DevOps by Jeff Geerling
  - The Phoenix Project by Gene Kim (The Unicorn Project releasing soon)
  - Accelerate by Nicole Forsgren
  - Thinking in Systems by Donella Meadows
- Web Resources
  - Ansible Best Practices: https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html
  - Best Practices for Ansible Slide Deck: https://www.slideshare.net/GeorgeShuklin1/best-practices-for-ansible
  - Testing with Molecule: https://www.jeffgeerling.com/blog/2018/testing-your-ansible-roles-molecule

2019 Michigan IT Symposium

Questions?