2019 Michigan IT Symposium

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

Jaime Magiera & Michael Shen

An Overview of Ansible

Jaime Magiera
LSA TS RCI Infrastructure Services

Michigan IT Symposium
SPARKING CONNECTIONS & IGNITING IDEAS ACROSS U-M

What is Ansible?

1. System for managing hosts en masse
2. Developed by Michael Dehaan in 2013
3. Grew out of the need for scalable management with easy-to-understand configuration
4. Initially supported by AnsibleWorks, Inc, which was purchased by Red Hat in 2015

Components of Ansible

Simple, but powerful

- Inventory: List of hosts in YAML or INI format
- Playbooks: Configuration files in YAML format
- Python interpreter
- Modules
- Roles: Collections of configuration files and other resources

Playbooks

1. YAML format
2. Can be written with any text editor. There are also several GUI tools.
3. References built-in Ansible or third-party modules
4. Playbooks are passed to the Ansible binary application or uploaded to a git repo for access from other services.

ansible-common / playbooks / **configure_webserver.yml**

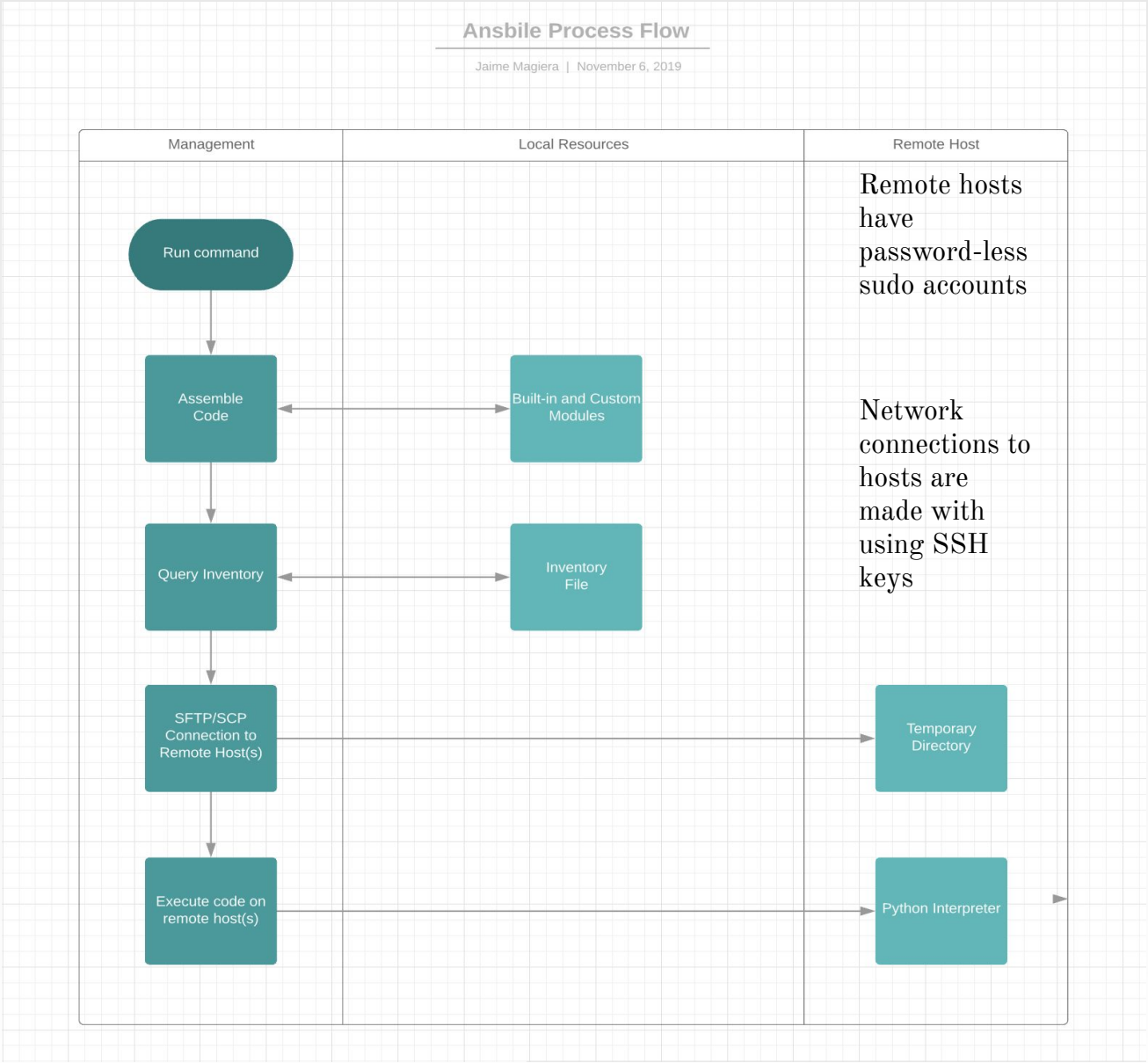
```
- hosts: "{{ target_server }}"
  become: true
  vars:
    configscriptpath: "/usr/local/lsa_inf/scripts/"
    vhost: "{{ vhost }}"
  tasks:
  - name: install the latest version of Apache
    yum:
      name: httpd
      state: latest
  - name: install the latest version of mod-ssl
    yum:
      name: mod_ssl
      state: latest
  - name: install the latest version of Mariadb
    yum:
      name: mariadb-server
      state: latest
  - name: create sites-enabled directory
    file:
      path: "/etc/httpd/sites-enabled"
      owner: apache
      group: apache
      state: directory
  - name: copy vhost file with owner and permissions
    copy:
      src: /usr/local/lsa_inf/data/spider_prod_data/lsa-museums.lsa.umich.edu.conf
      dest: /etc/httpd/sites-enabled/
      owner: apache
      group: apache
      mode: 0644
  - name: add include statement to apache.conf
    lineinfile:
```

Inventory

1. Ini format
2. Multiple entry types
   a. <hostname> <ip>
   b. <ip>
   c. <fqdn>
3. Supports host groups using the [header] nomenclature
4. Multiple host files

```
[jaimelm@oatmeal ~]$ cat /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group
```

How does Ansible modify the remote host?


Ansbile Process Flow
Jaime Magiera | November 6, 2019
Management
Local Resources
Remote Host
Run command
Assemble Code
Built-in and Custom Modules
Query Inventory
Inventory File
SFTP/SCP Connection to Remote Host(s)
Temporary Directory
Execute code on remote host(s)
Python Interpreter
Remote hosts have password-less sudo accounts
Network connections to hosts are made with using SSH keys

Available Modules

There are modules for virtually every aspect of host management

Module Index

- All modules
- Cloud modules
- Clustering modules
- Commands modules
- Crypto modules
- Database modules
- Files modules
- Identity modules
- Inventory modules
- Messaging modules
- Monitoring modules
- Net Tools modules
- Network modules
- Notification modules
- Packaging modules
- Remote Management modules
- Source Control modules
- Storage modules
- System modules
- Utilities modules
- Web Infrastructure modules
- Windows modules

Let's explore Playbooks...

Click me.

Expanding Ansible's Power

Integration into larger systems

- Host Management
- Security Scanning
- Network Configuration
- Containers
- Cloud

... many more ...

AWX/Ansible Tower

What is AWX/Ansible Tower?

- System for managing and executing playbooks and roles across groups of hosts
- Provides inventory with manual and auto-generated host groups
- Allows for scheduling of Jobs
- Adding “Facts”
  AWX is the Open Source upstream of Ansible Tower

Let's explore AWX...

Click me.

Red Hat Satellite

What is Red Hat Satellite?

1. Management of Linux hosts
    a. OS Installation
    b. Software Repository Subscriptions
    c. Software Installation
    d. Configuration Management
    e. Monitoring
2. Foreman is the upstream

Let's explore Red Hat Satellite...

Click me.

Integrating Ansible and Satellite

Some simple examples

- Automatically add hosts to AWX Inventory on build
- Use WebHooks for post-build playbook runs

Putting it all together.

Managing a Host with Ansible

1. Write or download Playbooks and Roles for the tasks you want done

2. Commit those Playbooks and Roles into a Repository

3. Install no-password sudo account on desired host (e.g. on build)

4. Run playbooks manually or automated on the desired hosts

Integrating Ansible and Satellite

1. Create an AWX/Tower account that has access to Satellite

2. Create a smart inventory that reads from Satellite

3. Add webhooks to Provisioning Templates that activate playbooks in AWX

Conclusion

Ansible has helped usher in the DevOps paradigm, where scalable host management is done with a combination of coding, code repositories, pushed tasks, and state monitoring.

Resources

- Ansible
- Molecule
- AWX
- Red Hat Satellite

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!*)
- Ansible and Molecule (and Kubernetes) Contributor™

www.github.com/mjlshen/talks mishen@umich.edu

> whoami

- In March I knew practically nothing about Ansible

6. Integrating Ansible with RHEL Builds

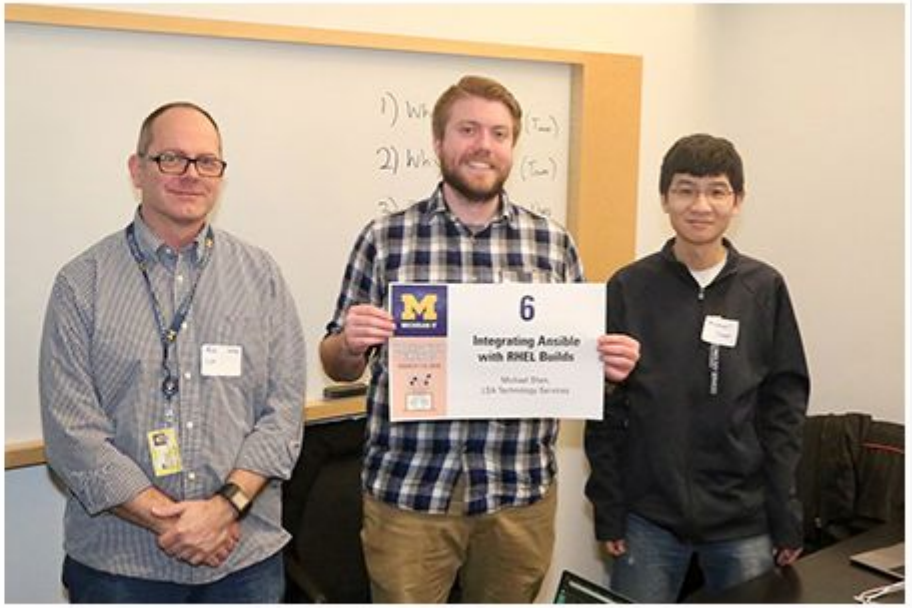
Submitted by: Michael Shen, *LSA Technology Services*

Description:

We're transitioning away from Satellite and Ansible is one potential option, this just represents an initial foray of sorts into what kind of work would need to be done for a full-transition.

3-Minute Pitch

Skills needed: Developers, Desktop Support, Project Managers

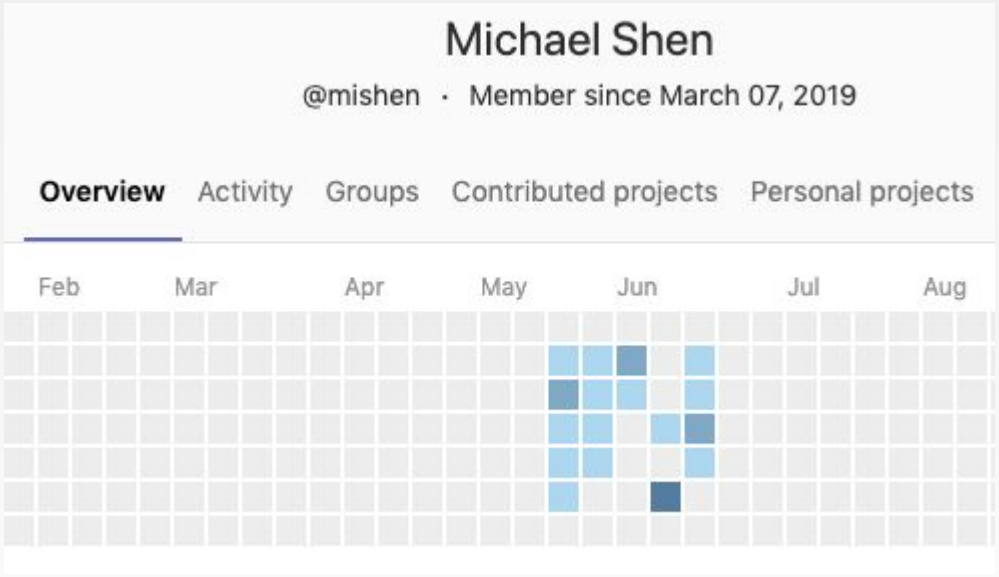


> whoami

- By May I learned enough to be convincing enough to attend the Red Hat Summit in Boston

> whoami

- By June I learned enough to convince HITS to hire me



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

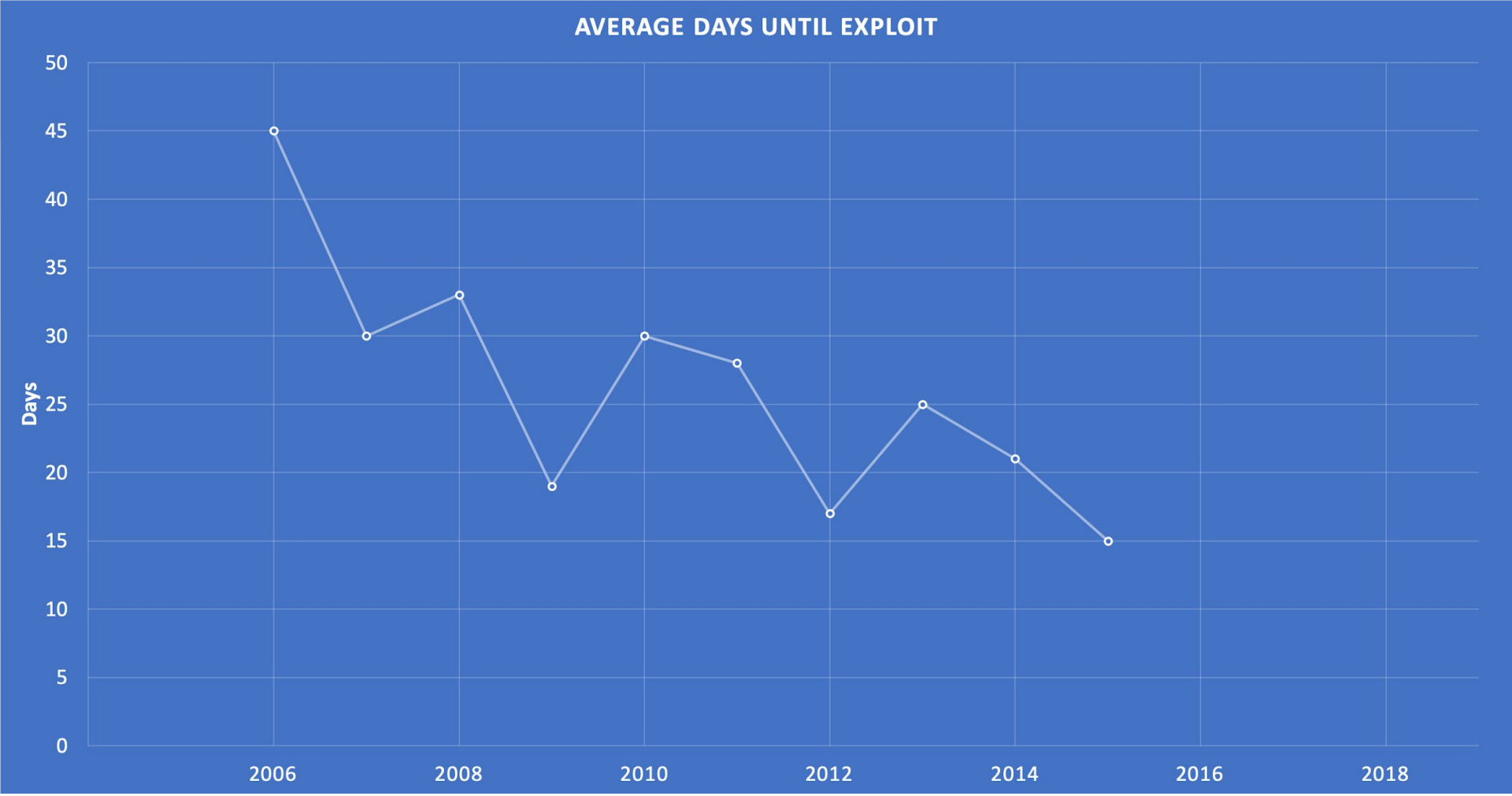
1. Security
2. Availability
3. ~~Resource Management~~
4. ~~Service Discovery~~

*2017 Kelsey Hightower: Kubernetes Federation

* https://xkcd.com/2232/

Threats to Our Configuration

- The Software Changes
- The Environment Changes



*2016 IBM X-Force/Gartner Research

Misconfigurations as a Security Risk

- Manual human configuration continues to be a major cybersecurity risk
- Ansible allows varying teams to use the same toolset
  - Consistent, repeatable, and secure environments can be collaboratively deployed and verified using Ansible

Misconfigurations as a Security Risk

- Manual human configuration continues to be a major cybersecurity risk
- Ansible allows varying teams to use the same toolset
  - Consistent, repeatable, and secure environments can be collaboratively deployed and verified 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.”

Demo

Deployment of a Web Application - Sonatype Nexus

> molecule converge

> molecule login

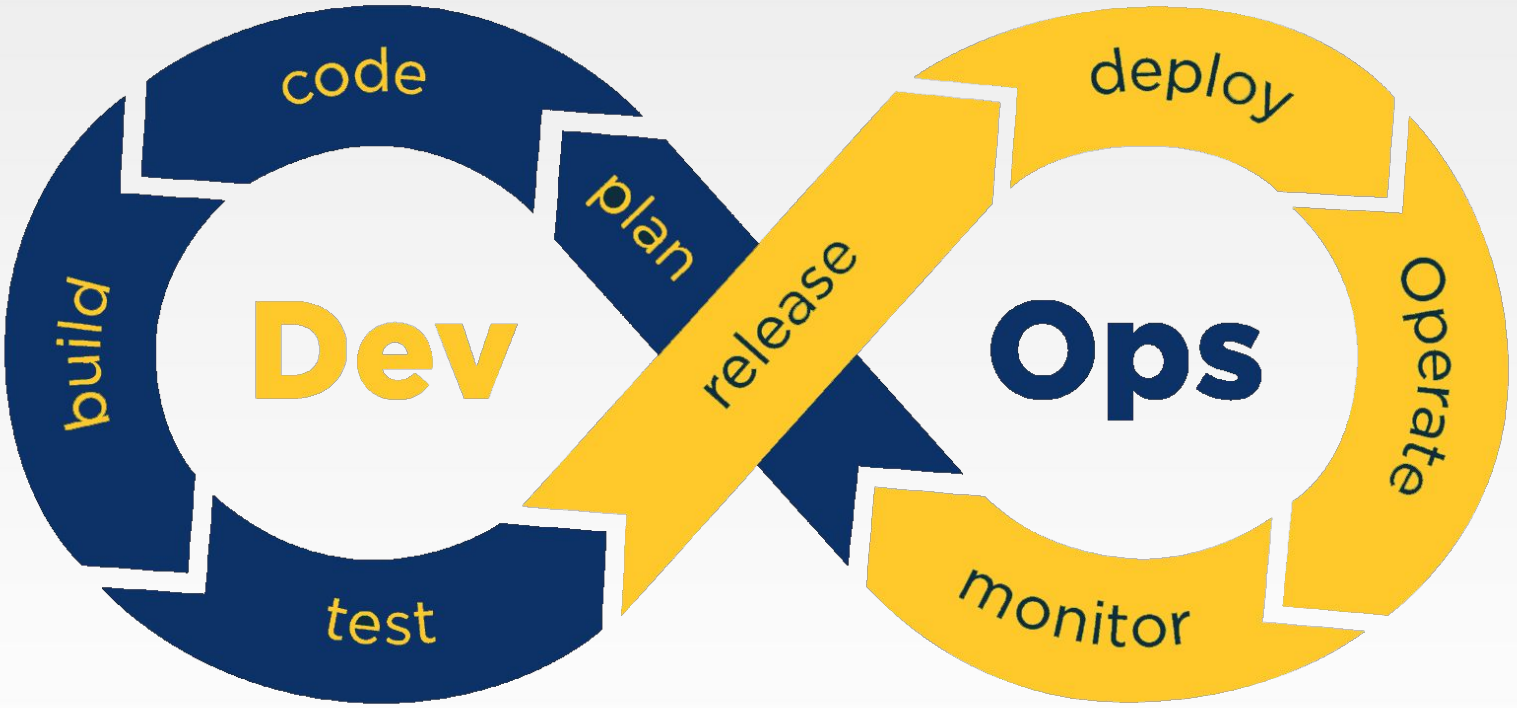
```
driver:
 name: vagrant
 provider:
   name: virtualbox
platforms:
 - name: centos7
   box: centos/7
   instance_raw_config_args:
     - 'vm.network "forwarded_port", host_ip: "127.0.0.1",
host: 8080, guest: 8081'
   memory: 4096
   cpus: 4
```

```
driver:
 name: docker
platforms:
 - name: centos7
   image: "geerlingguy/docker-centos7-ansible:latest"
   command: "/lib/systemd/systemd"
   volumes:
     - /sys/fs/cgroup:/sys/fs/cgroup:ro
   privileged: true
   pre_build_image: true
# ...Continued
```

```
driver:
 name: docker
platforms:
# ...from previous slide
 - name: ubuntu1804
   image: "geerlingguy/docker-ubuntu1804-ansible:latest"
   command: "/lib/systemd/systemd"
   volumes:
     - /sys/fs/cgroup:/sys/fs/cgroup:ro
   privileged: true
   pre_build_image: true
```

What did we get?

- A local development environment!



Demo

Controlled Update of a Web Application - Sonatype Nexus

> molecule converge

> molecule login

> molecule destroy

What do we now have?

- Repeatable, verifiable proof that our Ansible Role to initialize Sonatype Nexus works on CentOS 7.6

- Repeatable, verifiable proof that our Ansible Role can upgrade versions of Sonatype Nexus from 3.17.0 to 3.19.1

What happens when the next Nexus or OS update is released?

What happens when a new team member is tasked with the upgrade?

Demo

molecule 101

> molecule test

> molecule lint

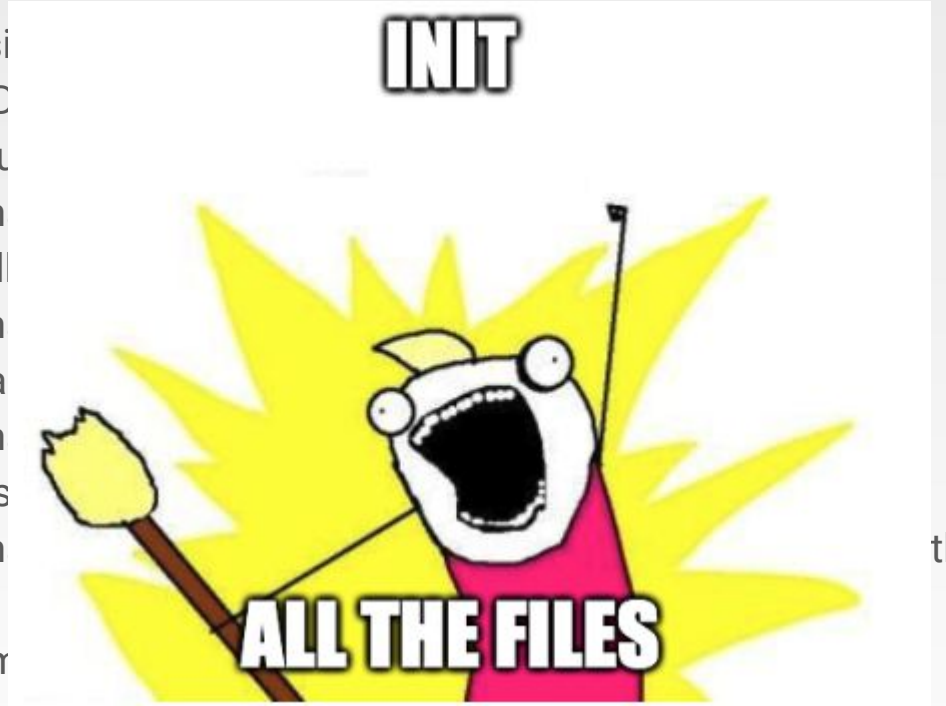
> molecule init role --role-name it_symposium

```
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
```

> cat molecule/default/molecule.yml

```
driver:
 name: docker
platforms:
 - name: instance
   box: centos:7
provisioner:
 - name: ansible
   lint:
     name: ansible-lint
```

> cat molecule/default/playbook.yml

```
- name: Converge
  hosts: all
  roles:
    - role: it_symposium
```

> 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
```

What did we get?

- Walk-through of an Ansible role development workflow with Molecule
- Our configuration follows best-practice syntax guidelines
  - Ignored rules that we don’t care about
- Proof that our Ansible role is safe to run over and over
  - Idempotence

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 or 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

Thank You

- Kyle Banas, Laura Fink, Linda Randolph, Joel VanLaven, Don Winsor, and Liz Zaenger (EECS DCO)
- Jim Deneen, Greyson LaHousse, Sean Quinn, and Jarrod Sandel (HITS)
- Rita Barvinok, Bill Custer, Steven Flack, John Gallias, Rob Heller, Steve Moser, Matt Rexer, Dan St. Pierre, and Vitaliy Pover (LSA TS - RSN)
- Brandon Case, Laura Green, Linda Hudson, Duane Lute, Stephen McClatchey, Chuck Schwartz, and Marlin Whittaker (LSA TS - Randall)
- Cathy Curley, Jaime Magiera, Jesse Reets (LSA TS)
- Shanelle Bolyut, Ross Bryan, Carol Lively, John Walsh, and JL Wilson (HITS DevOps)

2019 Michigan IT Symposium

Questions?

Jaime Magiera and Michael Shen
jaimelm@umich.edu mishen@umich.edu

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]’