Make your environment sane with Ansible Automation Fabio Alessandro Locati

Initial situation

- 300 GlassFish installations
- A good mix of versions 4.0, 4.1, 4.1.1
- Same(ish) application running on it
- 250 running on 25 EC2 running CentOS 6.x (10 instances/server)
- 50 running on 50 bare metal systems running CentOS 5.x (1 instance/server)
- 300 instances of MySQL running
- Thousands of scripts around (5 per instance), theoretically all copies of the same base scripts

About me - Fabio Alessandro Locati

- 15+ years in ICT, majority in infrastructure consulting
- 7+ years using Ansible
- 150+ contributions in github.com/ansible/ansible
- Author of 4 books, 3 of which on Ansible:
  - Learning Ansible
  - Learning Ansible 2.7
  - Practical Ansible
- Now working for Red Hat as Senior Solution Architect supporting Global System Integrators (GSI) partners in EMEA

The research

Automation system that is:

- Simple
- Can coexist with legacy processes
- Does not change the security model
- Is self-documenting(ish)
- Idempotent

Idempotence

Idempotence: is the property of certain operations in mathematics and computer science, that can be applied multiple times without changing the result beyond the initial application.

Idempotent examples:

- $X = 100$ (always 100)
- $X = X^0$ (always 1)
- echo ”TEST” $>$ /root/example

Non-idempotent examples:

- $X = X * 2$
- echo ”TEST” » /root/example

Idempotence - tricky/edge cases

- yum update
- yum install ...
- wget ...
- echo "$x" > /root/test

Ansible

- Agent-less
- Connects to managed machines via SSH
- Does not care about the state of the rest of the system
- Applies changes in a sequential way
- It has a very gentle learning curve
- Playbooks can be easily read by non-technical people (i.e.: auditors)
- It is very simple setup
- It is a swiss-knife tool (configuration, deployment, orchestration)

Initial setup

- Create SSH keys
- Distribute SSH keys
- Create git repository
- Create inventory

How to select processes to automate

- Non critical operations
- Very well understanded operations
- Easy to test

Deploying new application servers

- Install Java
- Create the glassfish user
- Install unzip
- Download Payara
- Unarchive Payara
- Set Payara file ownership
- Create systemd unit

Examples

```
---
 - name: Ensure we have Java installed
   yum:
     name: java-1.8.0-openjdk
     state: present

 - name: Ensure that the glassfish user exists
   user:
     name: glassfish
     state: present

 - name: Ensure we have unzip installed
   yum:
     name: unzip
     state: present
```