your management layer should be cattle too

`$ whoami`

Evgeni Golov

Senior Software Engineer at Red Hat

ex-Consultant at Red Hat

Debian Developer

❤️ FOSS ❤️

❤️ automation ❤️

“In the old way of doing things, we treat our servers like pets, for example Bob the mail server. If Bob goes down, it’s all hands on deck. […] In the new way, servers are numbered, like cattle in a herd. For example, www001 to www100. When one server goes down, it’s taken out back, shot, and replaced on the line.”

(Randy Bias, Bill Baker, ~2011)

idea

- everything should be repeatable, reproducible and replaceable
  - configuration management
  - immutable infrastructure
- generally applied to servers you have many of (“workers”)
- often ignored for systems that exist once (Foreman)

why change?

- You can deploy an identical testing environment
- Or one with minor differences (e.g. other networks)
- Lab environment on your laptop? Sure!
- Rebuild prod from scratch!

how change?

- Two step process:
  - Step 1: make Foreman installation automated
  - Step 2: make Foreman configuration automated
- Bonus: make all your efforts Open Source so others can benefit!
- We’ll use Ansible, but the concepts are applicable everywhere

Step 1: make Foreman installation automated

acquire a system to install on

- For lab-on-my-laptop:
  - Vagrant
  - Containers
- For test/prod:
  - oVirt/RHV
  - Containers

acquire a system to install on

- ideally your lab, your test and your prod use the same technology (container, virt, metal)
- for the demo in this talk we’ll use Vagrant (prod: RHV)
- there is currently no container for Katello, so a lot of deployments are classical VMs

install Foreman

- configure the needed repositories
- install the packages
- execute `foreman-installer`

install Foreman

- enter `theforeman.operations` collection
- goal: easy Foreman operations (installation, upgrade, etc) in VMs
- provided by the Foreman project and *used* by the Foreman project
- “successor” of the content you could find in `theforeman/forklift`, now suited for general consumption

install Foreman

```yaml
roles:
  - role: foreman_repositories
    vars:
      foreman_repositories_version: '2.3'
  - role: theforeman.operations.installer
    vars:
      installer_scenario: foreman
```

install Katello

```
roles:
  - role: foreman_repositories
    vars:
      foreman_repositories_version: '2.3'
  - role: katello_repositories
    vars:
      katello_repositories_version: '3.18'
  - role: theforeman.operations.installer
    vars:
      installer_scenario: katello
```

install more Plugins

```
roles:
  …
  - role: theforeman.operations.installer
    vars:
      installer_scenario: katello
      installer_options:
      - '--enable-foreman-plugin-ansible'
      - '--enable-foreman-proxy-plugin-ansible'
      - '--enable-foreman-plugin-remote-execution'
      - '--enable-foreman-proxy-plugin-remote-execution-ssh'
```

install Foreman

- at this point we have a Foreman (with plugins) running
- and can continue with adding things *inside* Foreman

Step 2: make Foreman configuration automated

structured data is key

- if we could describe everything inside Foreman in a structured way, we’d be done
- we can manage a lot with Ansible using the `theforeman.foreman` collection
- modules for managing individual entities inside Foreman
- roles to encapsulate workflows

structured data is key

```
- name: create domains
  theforeman.foreman.domain:
    name: "{{ item }}"
  loop:
    - example.com
    - example.org
```

structured data is key

`vars.yml`:

```
domains:
  - example.com
  - example.org
```

playbook:

```
- name: create domains
  theforeman.foreman.domain:
    name: "{{ item }}"
  loop: "{{ domains }}"
```

structured data is key

`vars.yml`:

```yaml
products:
- name: CentOS 7
  repositories:
    - name: CentOS 7 Base x86_64
      url: http://mirror.centos.org/centos/7/os/x86_64/
    - name: CentOS 7 Extras x86_64
      url: http://mirror.centos.org/centos/7/extras/x86_64/
    - name: CentOS 7 Updates x86_64
      url: http://mirror.centos.org/centos/7/updates/x86_64/
- name: Foreman Client
  repositories:
    - name: Foreman Client CentOS 7
      url: https://yum.theforeman.org/client/2.3/el7/x86_64/
```

structured data is key

playbook:

```
vars_files:
  - vars.yml
roles:
  - role: theforeman.foreman.repositories
```

data for a “content consumer”

- products/repositories (`t.f.repositories`)
- content views (no role yet)
- lifecycle environments (role in progress)
- activation keys (`t.f.activation_keys`)

actions for a “content consumer”

- repositories need to be synced
- content views need to be published (if used)
- modules to do this exist, but the *when* greatly varies based on environment

Step 3: maintenance

upgrading Foreman

- Foreman in a VM means upgrades at some point
- Switch repositories, update packages, run installer

cleaning Katello

- when you use Content Views, old (unused) versions of them accumulate

```
- role: theforeman.foreman.content_view_version_cleanup
  vars:
    content_view_version_cleanup_keep: 10
```

TBD

operations:

- finalize repository configuration
- proxy deployment (exists in forklift, needs porting/cleaning)

configuration:

- no feature parity with UI/CLI yet
    - especially for provisioning cases that differ per compute

DEMO

Links

- destructivebuilds repo for the demo
- forklift
- Foreman Operations Collection
- Foreman Ansible Collection

Thanks!

✉ evgeni@golov.de

🌐 die-welt.net

🐦 @zhenech

💬 @zhenech@chaos.social

🐙 @evgeni