



**RED HAT
TECH EXCHANGE**
Europe, Middle East & Africa

DISCOVER. DESIGN. DEPLOY.

Port Aventura, Spain September 25-29, 2017



ANSIBLE PATTERNS AND BEST PRACTICES

Fabio Alessandro Locati
Senior Consultant
25/09/2017

AGENDA

Today topics I'd like to cover

- Ansible Core
- Ansible Tower
- Ansible engagements

BEFORE WE START

PUBLIC SERVICE ANNOUNCEMENTS

Probably you already know:

- Ansible Tower is now Open Source (AWX project)
- Ansible Tower 3.2 has recently been released
- Ansible Core has recently been productized (Ansible Engine)

WHERE DO I COME FROM

- Consulting since 2004
- Consulting on and around Ansible since 2014
- Consulting for Red Hat since 2016

ANSIBLE CORE

GENERAL TIPS

- Respect the KISS principle
- Always name your tasks
- Generalize your playbooks
- Replace files vs add lines
- Idempotency

IDEMPOTENCY

Idempotency 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.

- Always make idempotent roles
- Distinguish between idempotent and not idempotent Playbooks
- Use names that remember the idempotency status of the playbook or role
 - Install X → Ensure X is present
 - Add line X to file Y → Ensure line X is present in file Y

NATIVE YAML STYLE

Ugly compressed syntax:

```
- name: Ensure mysql is installed
  yum: name=mysql state=present update_cache=True
```

Native syntax:

```
- name: Ensure mysql is installed
  yum:
    name: mysql
    state: present
    update_cache: True
```

VARIABLES

- Use human readable (meaningful) names
- Prefix variable names with role name
 - `http_port: 443`
 - `mysql_data_path: /var/mysql/`
- No bare variables
 - `with_items: the_list` # Not OK
 - `with_items: '{{ the_list }}'` # OK

PLAYBOOKS

- Create small (duplo) playbooks and then combine them for eventual big tasks
- Specific module >> command >> shell >> script >> raw
- If you are using commands (or worst) methods, consider writing a custom module
- Remove the **debug** tasks or set a high (2+) *verbosity* level (Ansible 2.1+)

TEMPLATES

- Call templates based on the output file name
- Begin all your templates files with:

```
{{ ansible_managed | comment }}
```

- Do not manage variables in templates
- Do not use complex/nested conditionals
- Do not use complex/nested iterations
- Avoid logic based on the hostname

ROLES

- Keep roles simple
- Prefer reusable roles
- Every role needs to have one (and only one) purpose
- Keep the roles in the *roles/* folder or use **ansible galaxy**
- Do not use git submodules for roles

DEPLOYMENT

- Use ansible-lint in the CI pipeline
- Eventually consider ansible-review
- Always peer review
- Use a single branch
- Execute all idempotent roles automatically and often (ie: every day)
- Investigate unexpected changes

ANSIBLE TOWER

ARCHITECTURE

- Do you really need HA Ansible Tower?
 - Do you really need HA PostgreSQL?
 - What HA means in your organization/company/bank/...?
-
- How many DataCenters do you have?
 - Where are they located?
 - What kind of connections do you have between them?

INVENTORIES

- Which is the golden source of information about your infrastructure?
- Is there one?
- Where those data should live (logically)?
- What is the level of consistency you require from your inventory?

FYI: inventories synchronization can be sloooooow (ie: 6hrs for a Satellite sync)

ANSIBLE ENGAGEMENTS

KINDS OF ENGAGEMENTS

- Discovery sessions
- Formal Training

- Ansible and automation workshops (2-5 days)
- Design and Implement automation (3-10 days)
- Support to write Playbooks/training on the job (2-10 days)

- Installing/Upgrading/Clustering Ansible Tower (1-2 days)
- Handholding/health checks/check-in (few days per month max)

RISKY ENGAGEMENTS

- “Please come and automate everything without our help”
- Write Playbooks for customer utilization
- Write Modules for customer utilization

THANKS!



redhat.®

RED HAT TRAINING: ANSIBLE

ADDRESSING THE SKILLS GAP IN YOUR CUSTOMER'S ORGANIZATION



OPTION 1: HIRE

Time consuming
Expensive
Difficult



OPTION 2: CONTRACT

Quick
Expensive
Short-term



OPTION 3: TRAIN

Flexible
Cost effective
Long-term

*Training is **cheaper** than other alternatives when your customer consider the hidden costs of hiring, high turnover, low productivity, and employee disengagement*

ANSIBLE TRAINING

Courses:

- Ansible Essentials: Simplicity in Automation Technical Overview (D0007) > **Free of charge!**
- Automation with Ansible I (D0407)
- Automation with Ansible II: Ansible Tower (D0409)

Exam:

- Red Hat Certificate of Expertise in Ansible Automation (EX407)

Red Hat Learning Subscription includes the 2 Ansible trainings and +50 additional Red Hat courses!



redhat.®