Ansible Basics

Oleg Fiksel

Security Consultant @ CSPI GmbH

oleg.fiksel@cspi.com | oleg@fiksel.info

OpenRheinRuhr 2015
AGENDA

INTRODUCTION
- Goals of this talk
- Configuration management

ANSIBLE
- Key Points
- Ad hoc Approach
- Playbook
- Run Playbook
- Idempotence
- Facts
- Handlers
- Best practices
- Summary

END
- Q & A
- Links
ABOUT ME

- Security Consultant at CSPI (former MODCOMP)
- Main topics
  - Automation
  - Virtualisation
  - Application Switching (load balancing)
  - Perl Coding
Founded in 1976 as MODCOMP Inc. Since 1985 in Germany.

Main scope: production of minicomputer for real-time environments.
Example: NASA Space Shuttle Program.

Development of real-time operating system Real/IX.
1990 - 1992 Cray and Bull equip their HPCs with Real/IX.
1996 purchased by CSPI.
Since 2015 re-branded as CSPI Germany.
About CSPI

- 3 locations world wide: US, DE, UK.
- CSPI Germany (Köln) ~90 employees.
  - 9 solution centers covering every aspect of IT-Security.
  - An opportunity to work on big infrastructures with cutting edge technology.
# GOALS OF THIS TALK

- This is not a comparison of configuration management systems.
- Basic theoretical understanding of configuration management.
- Introduction to ansible.
- Practical examples using ansible.
GOALS OF THIS TALK

- This is not a comparison of configuration management systems.
Goals of this talk

- This is not a comparison of configuration management systems.
- Basic theoretical understanding of configuration management.
GOALS OF THIS TALK

▶ This is not a comparison of configuration management systems.
▶ Basic theoretical understanding of configuration management.
▶ Introduction to ansible.
GOALS OF THIS TALK

- This is not a comparison of configuration management systems.
- Basic theoretical understanding of configuration management.
- Introduction to ansible.
- Practical examples using ansible.
**What is a goal of configuration management?**

Provide **easy, repeatable** and **scalable** provisioning and configuration management.
WHAT DOES THIS MEAN?
What does this mean?

- easy
WHAT DOES THIS MEAN?

- easy
  - configuration is consolidated versioned
WHAT DOES THIS MEAN?

▶ easy
  ▶ configuration is consolidated versioned
▶ repeatable
What does this mean?

- easy
  - configuration is consolidated versioned
- repeatable
  - provisioning produces every time the same result
**WHAT DOES THIS MEAN?**

- **easy**
  - configuration is consolidated versioned

- **repeatable**
  - provisioning produces every time the same result

- **scalable**
What does this mean?

- easy
  - configuration is consolidated versioned
- repeatable
  - provisioning produces every time the same result
- scalable
  - provisioning can be done to any number of machines
Ansible Key Points
Ansible Key Points

- Fresh (started February 2012)
Ansible Key Points

- Fresh (started February 2012)
- Simple
Ansible Key Points

- Fresh (started February 2012)
- Simple
  - YAML Syntax
Ansible Key Points

- Fresh (started February 2012)
- Simple
  - YAML Syntax
  - straight forward running scenario
Ansible Key Points

- Fresh (started February 2012)
- Simple
  - YAML Syntax
  - straight forward running scenario
- Agentless
ANSIBLE KEY POINTS

- Fresh (started February 2012)
- Simple
  - YAML Syntax
  - straightforward running scenario
- Agentless
  - dependencies for node
    - SSH + (Python >= 2.5 | Python < 2.5 + python-simplejson)
**USING ANSIBLE AS PSSH**

Ansible can be used as pssh.

```bash
ansible -i 10.0.0.1,10.0.0.2, all -m command -a '/bin/date'
```
 USING ANSIBLE AS PSSH

Ansible can be used as pssh.

```
ansible -i 10.0.0.1,10.0.0.2, all -m command -a '/bin/date'
```

Run `/bin/date` on machines 10.0.0.1 and 10.0.0.2.
## PLAYBOOK

Playbooks are YAML.

```yaml
---
# http://www.withoutthesarcasm.com/ubuntu-motd-landscape/
- hosts: all
  remote_user: root
  tasks:
    - name: remove landscape-client
      apt: name=landscape-client state=absent purge=yes
    - name: remove landscape-common
      apt: name=landscape-common state=absent purge=yes
```
How to run a Playbook?

ansible-playbook -i inventory_file playbook.yml

ansible-playbook -i hostname1,hostname2,192.168.0.10, playbook.yml
Idempotence

1Wikipedia Quote
Idempotence

“Idempotence is the property of certain operations in mathematics and computer science, that can be applied multiple times without changing the result.”

\(^1\)Wikipedia Quote
IDEMPOTENCE

"Idempotence is the property of certain operations in mathematics and computer science, that can be applied multiple times without changing the result." ¹

Simple: Goal of ansible playbook is to define the desired state and not script you way to this state.

¹Wikipedia Quote
EXAMPLE 1

1 ansible -i test-node, all -m shell \
2 -a ech"192.168.0.1 test-node" >> /etc/hosts'
**Example 2**

```bash
ansible --inventory test-node, all -m lineinfile \n   -a "dest=/etc/hosts line="192.168.0.1 test-node""
```
Example 3

```yaml
---
- hosts: all
  tasks:
    - name: clean up /etc/hosts
      lineinfile: dest=/etc/hosts regexp=192\.[0-9]{1,3}\.0 state=absent
    - name: add new /etc/hosts entry
      lineinfile: dest=/etc/hosts line="192.168.0.1 test-node"
```
Facts

Facts are fetched from a host and exported as variables, which can be used in playbooks.
Facts

Facts are fetched from a host and exported as variables, which can be used in playbooks.

See all facts for a host:

1. `ansible hostname -m setup`
2. `ansible -i hostname, all -m setup`
EXAMPLE

```yaml
---
- hosts: all
  tasks:
    - name: "shutdown CentOS 6 and 7 systems"
      command: /sbin/shutdown -t now
      when: ansible_distribution == "CentOS" and
           (ansible_distribution_major_version == "6"
            or
            ansible_distribution_major_version == "7")
```
TURN OFF GATHERING FACTS IN A PLAYBOOK

```
1   ---
2   - hosts: all
3       gather_facts: no
4   tasks:
5     - name: clean up /etc/hosts
6       lineinfile: dest=/etc/hosts regexp=192\..0 state=absent
7     - name: add new /etc/hosts entry
8       lineinfile: dest=/etc/hosts line="192.168.0.1 test-node"
```
Handlers only run after all of the tasks are run, and they only run once, even if they are notified multiple times. They always run in the order that they appear in the playbook, not the notification order.
EXAMPLE

---
- hosts: webservers
  handlers:
    - name: restart apache
      service: name=httpd state=reloaded
  tasks:
    - name: ensure apache is at the latest version
      yum: name=httpd state=latest
    - name: write the apache config file
      template: src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at boot)
      service: name=httpd state=started enabled=yes
**BEST PRACTICES**

1. stage          # inventory file for stage environment
2. production     # inventory file for production environment
3.
4. group_vars/    # assign variables to particular server groups
  5. group1
6. host_vars/     # systems specific variables
  7. hostname1
8.
9. site.yml       # master playbook
10. webservers.yml # playbook for webserver tier
11.
12. roles/        # this hierarchy represents a "role"
   13. common/
      14. tasks/
         15. main.yml          # tasks file can include smaller files if warranted
         16. handlers/
            17. main.yml         # handlers file
         18. templates/
            19. ntp.conf.j2     # files for use with the template resource
                           # --------- templates end in .j2
         20. files/
            21. foo.sh          # script files for use with the script resource
         22. vars/
            23. main.yml        # variables associated with this role
         24. defaults/
            25. main.yml        # default lower priority variables for this role
         26. meta/
            27. main.yml        # role dependencies
         28.
         29. monitoring/     # same kind of structure as "common" role
SUMMARY

▶ Try ansible (ad hoc approach)

▶ Read ansible documentation

▶ Read other Playbooks

▶ Think on Playbook Idempotence

▶ Split big Playbooks into Roles
Q & A
Thanks!

Oleg Fiksel

oleg.fiksel@cspi.com  |  oleg@fiksel.info
LINKS

- MODCOMP/CSPI
  - MODCOMP History
  - MODCOMP on Wikipedia
- Ansible
  - Ansible docs
  - Ansible - managed node requirements