

# FFG 2016 - Ansible

## Serverless Configuration Management

Christoph Berkemeier

2016-02-26

# Table of Contents

- 1 Automation
- 2 Ansible
- 3 Ansible Galaxy
- 4 Ansible Modules
- 5 Ansible Inventory
- 6 Jinja2
- 7 Running Ansible
- 8 Ansible Debugging
- 9 Outlook
- 10 Conclusion

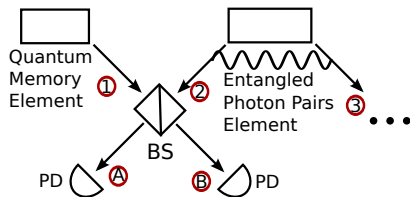


Figure: Quantum Repeater

# Configuration Management - Why?

Get Rid of Repetive Tasks

Faster Redeployments

Test Enviroments

Documentation

Standard Components - Standard Configurations

Abstraction Layer Above Shell Level

# Ansible

Written in Python

Works with Unix, Windows or Raw

Uses SSH, PowerShell Remoting or other Protocols

For Ad Hoc, Push or Pull actions

The Default is Push without a Running Server

<http://www.ansible.com/>

# Ansible Basic Concepts

## Hierarchy:

- Plays according to Playbooks
  - Groups of Servers
  - Roles
    - Tasks

## Operation:

- Serialized run of tasks
- Up to 5 parallel runs by default
- `ansible-playbook -f 5`

# Ansible Basic Code

```
# ls -l
```

```
- docker.yml
```

```
d group_vars
```

```
d host_vars
```

```
- site.yml
```

```
- hosts
```

```
d roles
```

```
# cat hosts
```

```
..
```

```
[docker]
```

```
docker01
```

```
docker02
```

```
# cat site.yml
```

```
..
```

```
- include:
```

```
docker.yml
```

```
#cat
```

```
docker.yml
```

```
- name:
```

```
docker specific
```

```
hosts: docker
```

```
remote_user:
```

```
root
```

```
roles:
```

```
- optuser
```

```
- docker
```

# Ansible Role Code

```
# ls -l
roles/docker
d defaults    # cat roles/docker/tasks/main.yml
d files      - include: mount.yml
d handlers   tags: docker
d meta       ..
-           - include: Debian.yml
README.md   when: ansible_os_family == 'Debian'
d tasks     tags: docker
d templates
d vars
```

# Ansible Loop Code

```
- name: Install Latest Kernel and Docker
apt: name= item state=installed
notify: reboot server
when: ansible_distribution == 'Ubuntu'
with_items:
- linux-generic-lts-wily
- docker-engine
```



Written in Ansible Code

User Generated Content → Lots of Roles

Hosted on Github

Easy Download with the ansible-galaxy Tool

# Ansible Modules

More than 400 Modules

Maintained by the Project itself and shipped with Ansible

There are Modules to Manage Software like Git, Open vSwitch, to run Puppet or Shell Scripts. "Core Modules" <—> "Extra Modules"

## Categories:

Cloud

Clustering

Commands

Database

Files

Inventory

Messaging

Monitoring

Network

Notification

Packaging

Source

Control

System

Utilities

Web

Windows

# Ansible Inventory Variables

```
ansible -i hosts -m setup -u $USERNAME -vvvv $IP
```

```
"ansible_lsb": {  
  "codename": "trusty",  
  "description": "Ubuntu [..]",  
  "id": "Ubuntu",  
  "major_release": "14",  
  "release": "14.04"  
},  
"factor_lsb": {  
  "codename": "trusty",  
  "description": "[..]",  
  "id": "Ubuntu",  
  "major_release": "14",  
  "release": "14.04"  
},  
"ohai_lsb": {  
  "codename": "trusty",  
  "description": "[...]",  
  "id": "Ubuntu",  
  "release": "14.04"  
},
```

Template engine similar to the Django template engine

Used in Flask, a micro web application framework

Python

# Jinja2 Syntax

General:

`{% ... %}` for Statements

`# ... ##` for Line Statements

`{# ... #}` for Comments not included in the output

`{{ ... }}` for Expressions to print to the output

More here: <http://jinja.pocoo.org/docs/dev/templates/>

# Jinja2 Control Loops

Control Loops:

```
{% if apache24 %} {% elif %} {% else %} {% endif %}
```

```
{% for host in list %} {% endfor %}
```

More here: <http://jinja.pocoo.org/docs/dev/templates/>

## Jinja2 Filter and Tests

Tests:

```
defined: {% if install_apache24 is defined %}
```

```
even: {% if hostid is even %}
```

Filter:

```
{{ install_apache24 | default(false) }}
```

```
{{ hostid | select("odd") }}
```

More here: <http://jinja.pocoo.org/docs/dev/templates/>

Write your own custom Python functions as filters

Ansible may Login as any User

When Ansible needs Privileges -> It becomes Superuser

Supported Methods: sudo, su, pbrun, pfexec, doas



## Prompts

```
---  
- hosts: all  
  
vars_prompt:  
  - name: "number"  
    prompt: "Your number?"  
  - name: "password"  
    prompt: "Your password?"
```

## Vault

```
ansible-vault create vaultfile.yml  
Other actions are  
- decrypt  
- edit  
- encrypt  
- rekey  
- view  
ansible-playbook --ask-vault-pass
```

# One Server and another one

## Delegation

tasks:

- name: Deploy Apache
  
- name: Add Server to Loadbalancer  
    delegate\_to: 192.168.1.1

## Run Once

- command: /opt/upgrade\_db.py  
    run\_once: true

## Delegated facts

(New in Version 2.0)

- hosts: app\_servers

tasks:

- name: facts from db servers  
    setup:  
        delegate\_to: "{{item}}"  
        delegate\_facts: True  
    with\_items:  
        "{{groups['dbservers']}}"

## Rolling Update Batch

### Size

```
- name: Webservers
  hosts: webservers
  serial: 3 <-> "30%"
```

## Maximum Failure

### Percentage

```
- hosts: webservers
  max_fail_percentage: 30
  serial: 10
```

## Ignoring Failed

### Commands

```
- name: this is no failure
  command: /bin/false
  ignore_errors: yes
```

## Aborting the play

```
- hosts: somehosts
  any_errors_fatal: true
  roles:
    - myrole
```

# Ansible Debugging

- name: ASensibleName

- tag: AContextTag

Verbose: `ansible-playbook -vvv`

Check: `ansible-playbook --check`

Diff: `ansible-playbook --diff`

Start and Step: `ansible-playbook --step`

Limit: `ansible-playbook --limit $HOSTGROUP or $HOSTS`

Lots of Ansible Code:

DebOps - <http://github.com/debops>

Network Automation:

NAPALM - <https://github.com/napalm-automation>

Imperative Automation:

Fabric - <http://www.fabfile.org>

Ansible + Fabric + ezjail:

BSDploy - <https://github.com/playground/bsdploy>

# Conclusion

No need for a server  $\Rightarrow$  low bootstrap

Powerful templating with jinja2

Good documentation on <http://docs.ansible.com/>

Mostly useful error messages

Enterprise web GUI Ansible Tower

Encrypted variables with Ansible vault

```
ansible-playbook -i hosts site.yml
```

```
-limit myServer -tags docker -check -diff
```

```
http://docs.ansible.com/ansible/playbooks\_best\_practices.html
```