Die ganze Welt mit Ansible automatisieren

International PHP Conference 2016

Tobias Schlitt

October 25, 2016

DevOps Borat
@DEVOPS_BORAT

Folgen

To make error is human. To propagate error to all server in automatic way is #devops.

About Me

Tobias Schlitt

https://qafoo.com

- Legacy code?
- Performance issues?
- Architecture challenges?
- Continuous Integration / Deployment?

Get in touch:

toby@qafoo.com

Why Configuration Management?

- Reproducible
- Documented (in code)
- Automated
- Reusable
- Testable

Necessary building block to benefit from the cloud

How do You do Config Management?

- Not at all
- Bash
- PHP
- **Ansible**

- Puppet
- Chef
- SaltStack

ANSIBLE

by Red Hat®

https://www.ansible.com/


Why just one tool?

Outline

Ansible Basics

Practical: Provisioning

Deployment

Ansible Basics

- Remote communication over SSH
- Push based by default
- Declarative DSL written in YAML
- No agents required - only Python 2.4+

Ansbile Basics

- Module
- Inventory
- Playbook
- Role

Ansible Basics: Modules

Modules are reusable, parameterizable functions

(currently >750 and counting . . .)

- apt. . .
- cron
- digital_ocean_. . .
- file
- . . .

Ansible Basics: Inventory

Inventory contains all managed hosts and information about them

Ansible Basics: Inventory

```
[all]
web1.foo ansible_ssh_host=33.44.55.66
web2.foo ansible_ssh_host=33.44.55.77
dbm.foo ansible_ssh_host=33.44.55.88
lb.foo ansible_ssh_host=33.44.55.99

[web]
web1.foo
web2.foo

[db]
dbm.foo

[lb]
lb.foo
```

Ansible Basics: Playbook

Combines multiple module calls into a sequence of tasks executed on host groups

Ansible Basics: Playbook

```
1 ---
2 - hosts: all
3   tasks:
4     - debug: msg="Hello World from {{ ansible_hostname }}!
        "
```

Ansible Basics: Executing a Playbook

```
$ ansible-playbook -i inventories/production provision.yml

PLAY [all] *********************************

TASK [setup] *******************************
[...]

TASK [debug] *******************************
ok: [dbm.foo] => { "msg": "Hello World from dbm.foo" }
ok: [web2.foo] => { "msg": "Hello World from web2.foo" }
ok: [web1.foo] => { "msg": "Hello World from web1.foo" }
ok: [lb.foo] => { "msg": "Hello World from lb.foo" }

PLAY RECAP *********************************
dbm.foo    : ok=2  changed=0  unreachable=0  failed=0
lb.foo     : ok=2  changed=0  unreachable=0  failed=0
web1.foo   : ok=2  changed=0  unreachable=0  failed=0
web2.foo   : ok=2  changed=0  unreachable=0  failed=0
```

Ansible Basics: Roles and Variables

Roles hide implementation details and improve reusability

Ansible Basics: Roles and Variables

```
1  - hosts: web
2    roles:
3      - role: logrotate
4        logrotate_files:
5          - name: application
6            pattern: "/var/log/app/*.log"
7            frequency: daily
8            rotate: 14
9
10 - hosts: db
11   roles:
12     - role: logrotate
13       logrotate_files:
14         - name: mysql
15           pattern: "/var/log/mysql/*.log"
16           frequency: weekly
17           rotate: 7
```

Ansible Basics: Roles and Variables

```
1  # roles/logrotate/tasks/main.yml
2  ---
3  - name: Write Logrotate
4    template: >
5      src=logrotate.j2
6      dest=/etc/logrotate.d/{{ item.name }}
7      owner=root
8      group=root
9      mode=0644
10   with_items: logrotate_files
```

Ansible Basics: Roles and Variables

```
# roles/logrotate/templates/logrotate.j2
{{ item.pattern }} {
    {{ item.frequency|default('daily') }}
    missingok
    copytruncate
    rotate {{ item.rotate|default('7') }}
    compress
    notifempty
}
```

Outline

Ansible Basics

Practical: Provisioning

Deployment

Provisioning

- Install required software
- Apply system global configuration
- Configure servers by type, e.g.
  - App / web
  - Database
  - Cache
  - Search
  - Queue
  - . . .

Installing Tools

roles/tools/tasks/main.yml

```
1  ---
2  - name: "Install Tools"
3    apt: pkg={{ item }} state=present update_cache=yes
4    with_items:
5      - pdftk
6      - curl
7      - tree
8      - libcurl4-openssl-dev
9      - libpcre3-dev
10     - gcc
11     - g++
12     - make
13     - cmake
```

Configure Web Server

roles/nginx/tasks/main.yml

```
1 ---
2 - name: Delete possible default Nginx vhosts
3   file: path={{ nginx_vhost_directory }}/{{ item }} state=
        absent
4   with_items:
5     - default
6     - www.domain.de.conf.example
7   notify: Restart Nginx
```

Ansible Loops

```
with_...
```

Restarting the Web Server

roles/nginx/**handlers**/main.yml

```
1  ---
2  - name: Restart Nginx
3    service: name=nginx state=restarted
```

Handlers

- Only triggered on “change”
- Only run once per playbook

Configuring VHosts

roles/nginx/tasks/main.yml

```
1
2  - name: Generate Vhosts
3    template:
4      src=vhost.conf
5      dest=/etc/nginx/{{ nginx_vhost_directory_name }}/{{
           item.subdomain | default("default") }}.conf
6      owner={{ nginx_user }}
7      group={{ nginx_user }}
8      mode=0644
9    with_items: "{{ vhosts }}"
10   notify: Restart Nginx
```

VHosts Template

roles/nginx/templates/vhost.conf

```
server {
    listen       *:80{% if item.default %} default_server{%
        endif %};
    server_name {{ item.subdomain|default("default") }};

    set $index "{{ item.script_name }}";
    set $project_root "/home/{{ project_user_dir }}/project
        /components";

    index $index;
    root "$project_root/{{ item.component }}/{{ item.dir }}
        ";
}
```

Default Variables

roles/nginx/defaults/main.yml

```
1 ---
2 nginx_user: "www-data"
3 nginx_vhost_directory_name: "sites-enabled"
```

Role Directory Structure

- <role>/tasks/main.yml
- <role>/handlers/main.yml
- <role>/{vars|defaults}/main.yml
- <role>/meta/main.yml

Outline

Ansible Basics

Practical: Provisioning

Deployment

User Input

Parameter variables

```
1  pre_tasks:
2    - name: "check for release_tag variable"
3      fail: msg="Call ansible-playbook with -erelease_tag=<
           tag>"
4      when: "release_tag is not defined"
5
6    - debug: msg="Deploying version {{ release_tag }}"
```

Require user input

```
1  vars_prompt:
2      - name: "release_tag"
3        label: "The project release tag to deploy"
4        private: no
```

Checkout / Upload

```
1  - name: Create and Update Bare checkout
2    git: >
3      dest="/opt/project/{{ release_tag }}"
4      repo="ssh://git@github.com/company/project.git"
5      bare=yes
6      force=yes
7      update=yes
8      version="{{ release_tag }}"
```

Care for security!

Adjust Environment

```
1  - name: "Update parameters.yml"
2    lineinfile: >
3      dest=/opt/project/{{ release_tag }}/app/config/
           parameters.yml
4      regexp="^    {{ item.name }}:"
5      line="    {{ item.name }}: {{ item.value }}"
6    with_items:
7      - { name: 'release_version', value: '{{ release_tag }}'
         }
```

Symlink Persistent Directories

```
1  - name: "Symlink Persistent Directories"
2    sudo: yes
3    file: >
4      path={{ item.dest }}
5      state=link
6      src={{ item.src }}
7      force=yes
8    with_items:
9      - { src: '/mnt/nfs/project/data', 'dest': '/opt/project
         /{{ release_tag }}/data' }
10     - { src: '/opt/project/img', 'dest': '/opt/project/{{
         release_tag }}/assets' }
```

Application Installation

```
1  - name: "Clear Cache"
2    sudo: yes
3    sudo_user: "project"
4    command: >
5      php app/console cache:clear --env=prod
6      chdir=/opt/project/{{ release_tag }}
7  - name: "Update database"
8    sudo: yes
9    sudo_user: "project"
10   command: >
11     ant db:initialize
12     chdir=/opt/project/{{ release_tag }}
13   when: "'migrator' in group_names"
```

Summary

- Pragmatic tool for config management
- Easy to use and adjust to various environments
- Huge collection of modules supported
- Outlook:
  - Ansible Tower
  - Ansible Vault

Contact: toby@qafoo.com

Qafoo
passion for software quality

THANK YOU

Rent a quality expert
qafoo.com