

Amazingly Awesome Ansible

@elsom25 / @Boltmade



Hi.



boltmade

Why Ansible?

Let's talk about deploying
“cloud” software

Deployments are complex

Deployments are transient

Deployments are essential

Ansible helps

Ansible gives you a common
language

Ansible gives you repeatability
(and speed)

Ansible gives you a deploy
manual

Ansible isn't perfect

You need to know *how* to deploy
software

Ansible is powerful

We're going to build a
deployment manual

An Ansible “playbook”

“Playbooks are Ansible’s configuration, deployment, and orchestration language. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process.”

–Ansible Docs

Our playbook will deploy a
simple Rails 5 app

A brief overview

Roles

Inventory

production

[db_servers]
159.203.29.210

[web_servers]
159.203.35.220
159.203.46.230

[production:children]
db_servers
web_servers

staging

[db_servers]
159.203.10.42

[web_servers]
159.203.10.42

[staging:children]
db_servers
web_servers

Variables

```
# playbooks/group_vars/all/main.yml
```

```
---
```

```
git_url: https://my-git-repo.com/
```

```
git_branch: dev
```

```
app_name: ansible-workshop-app
```

```
app_path: "/opt/{{ app_name }}"
```

```
# playbooks/group_vars/production/main.yml
```

```
---
```

```
git_branch: master
```

```
rails_env: production
```

Templates

```
# passenger.conf.j2
```

```
passenger_app_env      {{ rails_env }};  
passenger_default_user {{ passenger_user }};  
passenger_default_group www-data;
```

Becomes

```
# passenger.conf
```

```
passenger_app_env      production;  
passenger_default_user george;  
passenger_default_group www-data;
```

Galaxy Roles

Project Structure

```
production    # inventory file for production servers
staging       # inventory file for staging environment

group_vars/
  group1      # group variables for group1
  group2      # ""

host_vars/
  hostname1   # host variables for hostname1
  hostname2   # ""

site.yml      # master playbook
webservers.yml # playbook for webserver tier
dbservers.yml # playbook for dbserver tier
```

```
roles/
  common/      # this hierarchy represents a "role"
    tasks/    #
      main.yml # can be broken into more files
    handlers/  #
      main.yml # handlers file
    templates/ # files used with `template`
      ntp.conf.j2 # templates end in .j2
    files/     #
      bar.txt    # files used with `copy`
      foo.sh     # scripts used with `script`
    vars/      #
      main.yml   # role variables for common
    defaults/   #
      main.yml   # default low-priority variables
      meta/      #
      main.yml   # role dependencies

  webtier/     # like "common" but for webtier role
  monitoring/  # ""
  fooapp/      # ""
```


Let's dive in!



boltmade

Other cool things

Ansible + Vagrant

Ansible Vault

```
playbooks/  
  group_vars/  
    all/  
      main.yml  
      vaulted_app_user_password.yml  
  production/  
    main.yml  
    vaulted_db_password.yml  
    vaulted_secret_key_base.yml  
  ...
```

```
# playbooks/production/main.yml
```

```
---
```

- rails_env: production
- db_password: "{{ vaulted_db_password }}"
- secret_key_base: "{{ vaulted_secret_key_base.yml }}"

```
# playbooks/production/vaulted_db_password
```

```
# ansible-vault view secret_file
```

```
---
```

- vaulted_db_password: my-secret-password



@elsom25 / @Boltmade

https://docs.ansible.com/ansible/playbooks_best_practices.html

<http://hakunin.com/six-ansible-practices>