Ansible ecosystem

Julien Girardin

March 1, 2017

ANSIBLE

Syllabus

1. Ansible package
   - ansible cli
   - ansible-pull
   - ansible-galaxy
   - ansible-vault
2. git
3. jenkins

- **ansible** Run a single command on distant hosts.
  Usefull tasks: *ping*, *command*, *service*, *file*, *copy*
- **ansible-playbook** Run playbook file on distant hosts
  The real power of ansible !
- **ansible-pull** Run a playbook against the local machine
- **ansible-galaxy** Install roles from remote location.
  sources: *github*, *http*, *tar*
- **ansible-vault** Encrypt source file. Commit password encrypted by a symmetric master key
- ansible-container - *beta* - Build docker container

ansible cli

```
$ export ANSIBLE_HOSTS=/etc/ansible/ec2.py

$ export EC2_INI_PATH=/etc/ansible/ec2.ini

$ ansible -m ping tag_ansible_slaves
10.1.2.193 | success >> {
    "changed": false,
    "ping": "pong"
}
10.1.2.136 | success >> {
    "changed": false,
    "ping": "pong"
}
10.1.2.137 | success >> {
    "changed": false,
    "ping": "pong"
}
```

ansible-pull

```
#!/bin/bash
# Get this script as the AWS user-data for a fresh CentOS AMI
# It will be run on startup, and logs to /var/log/cloud-init.log
cat <<EOF > /etc/apt/sources.list.d/backport.list
deb http://ftp.debian.org/debian jessie-backports main
EOF
apt-get update
apt-get -y install ansible -t jessie-backports
apt-get -y install git
mkdir -p /home/debian/.ssh
cat <<EOF > /home/debian/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
EOF
chmod go-rw /home/debian/.ssh/id_rsa
cat <<EOF > /etc/boto.cfg
[Credentials]
aws_access_key_id = ABC123
aws_secret_access_key = ABC123ABC123
EOF
mkdir -p /usr/local/mbst
ansible-pull -C master \
    -d /usr/local/mbst/ansible \
    -U git@gitlab.com:account/repo.git \
    --key-file /home/centos/.ssh/id_rsa \
    --accept-host-key --full
```

ansible-galaxy

without galaxy

```
site.yml
subsite1.yml
subsite2.yml
groups_vars/
    prod/
        system.yml
        password.yml
    test/
        system.yml
        password.yml
host_vars/
roles/
    roles1/
    roles2/
    roles3/
    roles4/
```

VS

with galaxy

```
site.yml
subsite1.yml
subsite2.yml
groups_vars/
    prod/
        system.yml
        password.yml
    test/
        system.yml
        password.yml
host_vars/
requirement.yml
roles/
```

```
$ ansible-galaxy install
↪  -r requirements.yml
↪  -p roles
```

ansible-vault

```
$ ansible-vault create group_vars/prod/password.yml
Vault password:
# $EDITOR open
$ git add group_vars/prod/password.yml
$ git commit -m 'Add prod password into ansible-vault'

$ ansible-playbook site.yml --ask-vault-pass
Vault password:
```

Syllabus

1. Ansible package
2. git
3. jenkins

infrastructure as code

Freeze everything in git: reproductabilty is magic

Two philosophies

Big repo: Google

Everything in one repo

pros:

- Clear commit, only change text
- Easier tooling (dev: git grep, CI)

cons:

- Working with contractor

Multiple small repositories

pros:

- Sharing with other
- Abandon legacy

cons:

- Complex tooling
- Manage pointers on other project

```
code/
    lead -> git submodule
    techno -> git submodule
group_vars/
host_vars/
roles/
    common/
    mongodb -> git submodule
    lead -> symlink in code/lead/roles/ansible_lead
playbook.yml
```

Syllabus

1. Ansible package
2. git
3. jenkins

jenkins

- https://wiki.jenkins-ci.org/display/JENKINS/Ansible+Plugin
- https://wiki.jenkins-ci.org/display/JENKINS/SSH+Agent+Plugin
- https://wiki.jenkins-ci.org/display/JENKINS/Promoted+Builds+Plugin

behave ansible

https://github.com/Zempashi/behave-ansible