



# Ansible

## A brief overview

Robin Long  
June 22, 2016



- How could we configure a server?
- Manual install? - Slow
- Shell scripts? - What is I run them a second time?
- Configuration Management? Many options here.
  - What do you mean I need a server to setup a server?

# Why Ansible?

- Lightweight.
- Simple.
- It is its own documentation.
- Forces better practises.

# Cutting the strings and setting yourself free.

---

- Why not puppet?
- Very complex, nested, unclear.
- Need a server to setup a server.
- Getting too big.
- RedHat bought ansible.

- Needs very few dependencies:
  - Python
  - Jinja2
  - PyYAML
  - Paramiko
  - laptop
- Ansible just uses SSH, no need for servers, certificates and client installs.

- Ansible uses a very simple hosts file
- default is /etc/ansible/hosts
- pass a -i <inventory file> to ansible

```
[storage-nodes]
stor[000:031].hec.lancs.ac.uk
```

```
[service-nodes]
fal-pygrid-15.lancs.ac.uk
fal-pygrid-30
```

```
[loki]
py-loki.lancs.ac.uk:222
```

```
[norse]
py-loki.lancs.ac.uk:222
py-eir.lancs.ac.uk
```

- We can use ansible on the command line to issue basic commands and tasks

```
$ ansible <host-pattern> [-f forks] [-m module_name] [-a args]
$ ansible storage-nodes -m yum -a 'name=httpd state=installed'
$ ansible storage-nodes -m service -a 'name=httpd state=running enabled=yes'
```

- ansible uses variables. System defaults called facts

```
$ ansible local -m setup
```

- use `-u <user>` if host and client differ.

- simple way to manage many machines.
- Declare configurations and orchestrate complex processes
- Uses YAML
- Contain many different plays - each play is a task (install and start running apache.)



```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
    remote_user: root

  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

  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

# Playbooks

## Splitting long lines

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
    remote_user: root

  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

  handlers:
  - name: restart apache
    service:
      name: httpd
      state: restarted
```

- execute by running

```
ansible-playbook playbook.yml -f 10
```

- we can include other playbooks in our main one with
  - `include: servers.yml`
- This allows more complex playbooks, and roles.
- Roles are much the same as playbooks, just with added search paths.

```
site.yml
storage-servers.yml
roles/
  common/
    files/
    templates/
    tasks/
    handlers/
    vars/
    defaults/
    meta/
storage-servers/
...
```

- storage-servers.yml would then be:

```
---
```

```
- hosts: webservers
  roles:
    - common
    - storage-servers
```

# Variables, Jinja2, Conditionals and Loops

- Ansible variables can be used in playbooks and templates.
- System created “facts” can be seen from running  
`ansible hostname -m setup`
- most useful with templates.
- variables called in playbooks and templates using  
`{{ variable_name }}`
- can filter variables using jinja2.

```
- name: Install cvmfs and dependencies.  
  yum: name={{ item }} state=present  
  with_items:  
    - fuse  
    - cvmfs  
    - cvmfs-init-scripts
```

```
# copy host cert and key to /etc/grid-security/  
- name: copy host cert and key to /etc/grid-security/  
  copy: src={{item.src}} dest={{item.dest}} mode={{item.mode}}  
  with_items:  
    - {src: "{{ host_cert }}", dest: /etc/grid-security/hostcert.pem, mode: "u=rw,g=r,o=r"}  
    - {src: "{{ host_key }}", dest: /etc/grid-security/hostkey.pem, mode: "u=r,g=,o="}
```



```
# Automatically created by ansible
# using the ansible-bdii-site role

SITEBDII ldap://{{ ansible_fqdn }}:2170/mds-vo-name=resource,o=grid
{%if SITEURLS is defined %}
{% for alias, url in SITEURLS.iteritems() %}
{{ alias }} ldap://{{ url }}:2170/mds-vo-name=resource,o=grid
{% endfor %}
{% endif %}
```

---

SITEURLS:

HEC: `carceri.hec.lancs.ac.uk`

DPM: `fal-pygrid-30.lancs.ac.uk`

```
# Automatically created by ansible  
# using the ansible-bdii-site role
```

```
SITEBDII ldap://py-fjalar.hec.lancs.ac.uk:2170/mds-vo-name=resource,o=grid  
DPM ldap://fal-pygrid-30.lancs.ac.uk:2170/mds-vo-name=resource,o=grid  
HEC ldap://carceri.hec.lancs.ac.uk:2170/mds-vo-name=resource,o=grid
```

```
group_vars/  
  all  
  bdii-site  
  cvmfs-client  
  loki  
  service-nodes  
  squid  
  storage-nodes  
  vac
```

```
---  
# Variables here are applicable to the bdii-site group  
SITE_NAME: UKI-NORTHGRID-LANCS-HEP  
SITE_COUNTRY: UK  
SITE_DESC: UKI-NORTHGRID-LANCS-HEP  
SITE_WEB: https://lancsgrid.wordpress.com  
SITE_LOC: Lancaster, UK  
SITE_LAT: 54.0105  
SITE_LONG: -2.784  
SITE_EMAIL: lcg-admin@lancs.ac.uk  
SITE_SECURITY_EMAIL: lcg-admin@lancs.ac.uk  
SITE_SUPPORT_EMAIL: lcg-admin@lancs.ac.uk
```

## OTHERINFO:

- GRID=EGEE
- GRID=GRIDPP
- GRID=WLCG
- GRID=NORTHGRID
- TIER=2

## SITEURLS:

```
HEC: carceri.hec.lancs.ac.uk  
DPM: fal-pygrid-30.lancs.ac.uk
```

- Time to share code?
- <https://github.com/lancsgrid/>
  - squid ( production )
  - bdii ( production )
  - cvmfs-client ( production )
  - argus ( in progress )

Robin Long  
June 22, 2016

