Ansible

Simple Deployment and Configuration

PyCon Philippines
June 30 to July 1, 2012
Manila, Philippines

About Me

rodney
www.capsunlock.net
https://github.com/cocoy

unix friendly | sysadmin | biker

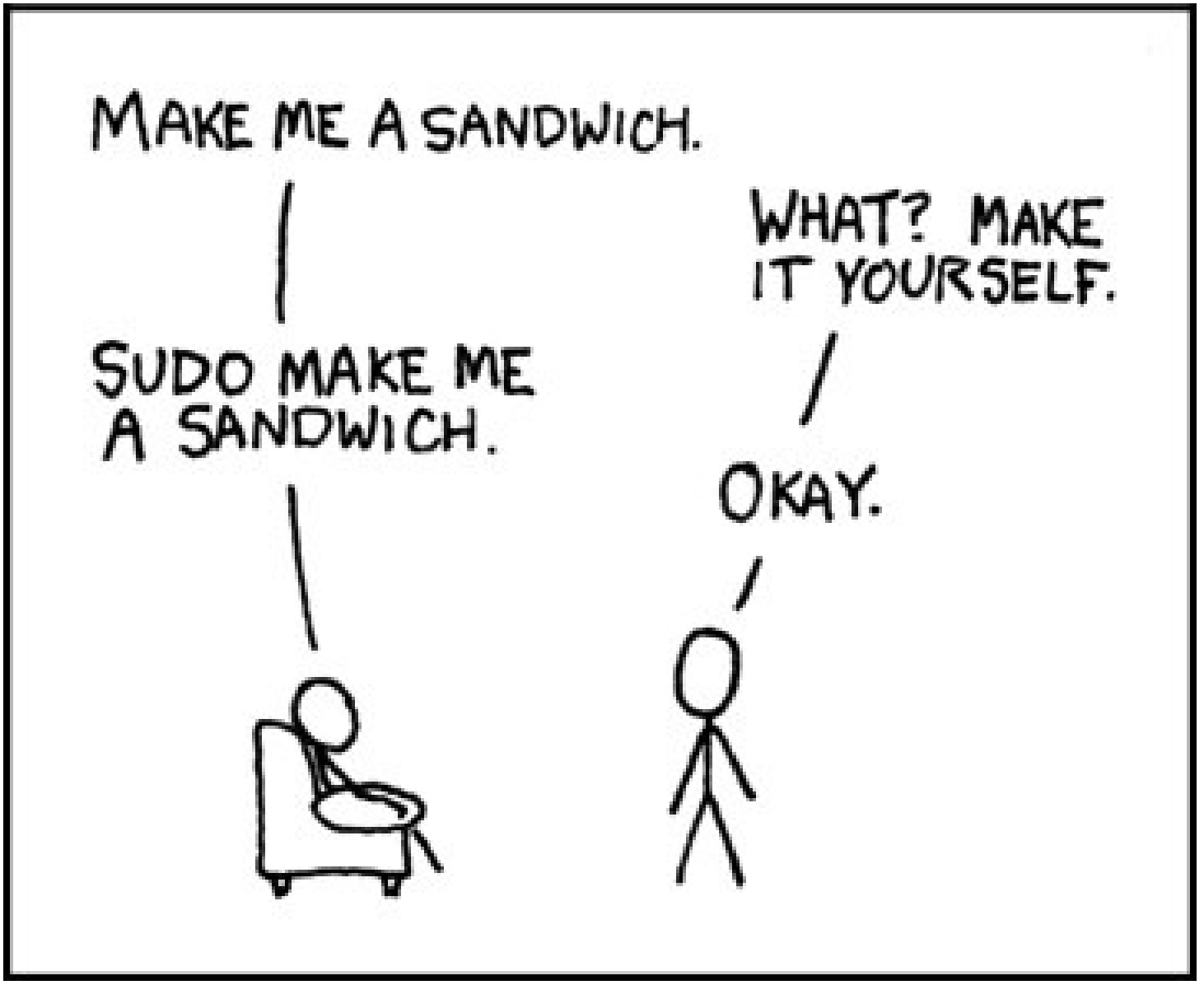


Photo by howard_ends

How to install and configure programs?

Photo by Rudolf_Chuba

Code Deployment to one or more servers

Photo by rumpleteaser

Review and Repeat Installation + Deploy

Photo by mikekline

SysAdmin Tools Available

Ad-Hoc Tasks (on the fly)	Deployment	Configuration Management
Func	Capistrano	CFEngine
pssh	Vlad	Puppet
clusterSSH	Fabric	Chef

Ansible

https://github.com/ansible/ansible

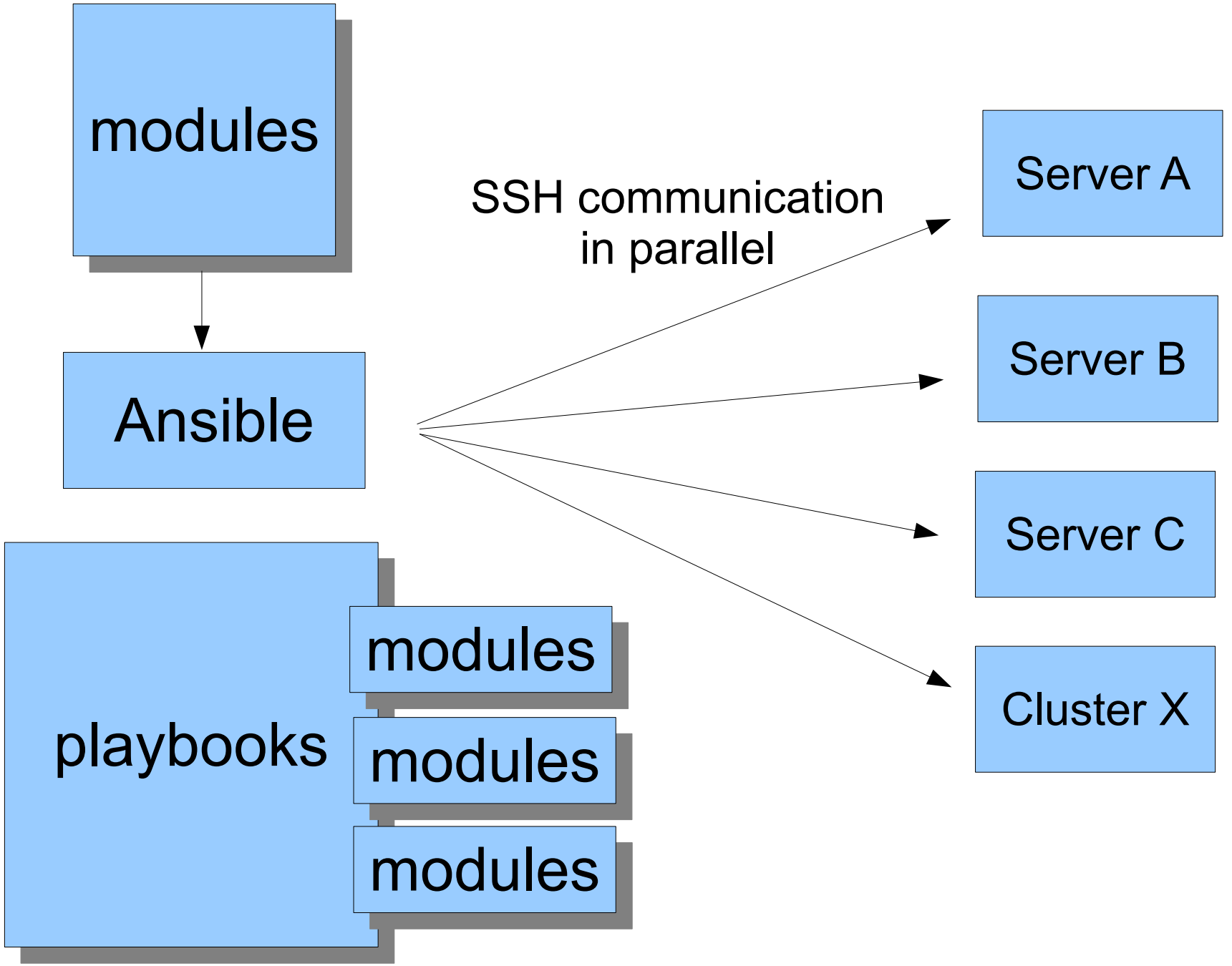
http://ansible.github.com/

Ansible is like a coffee.

Photo by rwp-roger

It will get you started in Minutes

Photo by Linemar pennytoy


modules
Ansible
playbooks
modules
modules
modules
SSH communication
in parallel
Server A
Server B
Server C
Cluster X

Configure Ansible Machine

$ pip install Jinja2 PyYAML paramiko

Configure Ansible Machine

$ git clone https://github.com/ansible/ansible.git

$ cd ./ansible

$ source ./hacking/env-setup

Set ANSIBLE_HOSTS

```
$ echo "127.0.0.1" > ~/my_servers.txt

$ export ANSIBLE_HOSTS=~/my_servers.txt
```

Checking your setup:

Command:

$ ansible all -m ping --ask-pass -u ubuntu

Expected Output:

```
127.0.0.1 | success >> {
    "ping": "pong"
}
```

Adding Remote Systems

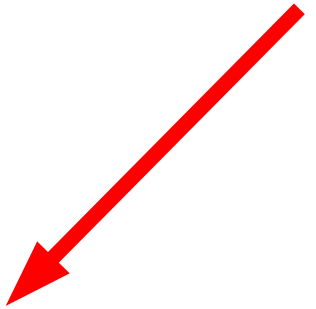
prompt> vi my_servers

```
[web-servers]
web1.mydomain.com
web2.mydomain.com

[db-master]
db1.mydomain.com

[db-slave]
slavedb.mydomain.com
```

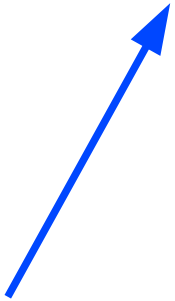
Running Ad-hoc Tasks

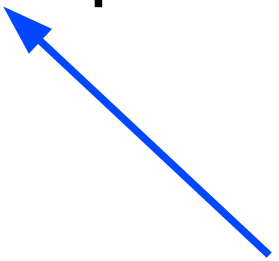


```
$ ansible web-servers  -m command -a “uptime” \
                       -u ubuntu -k
```

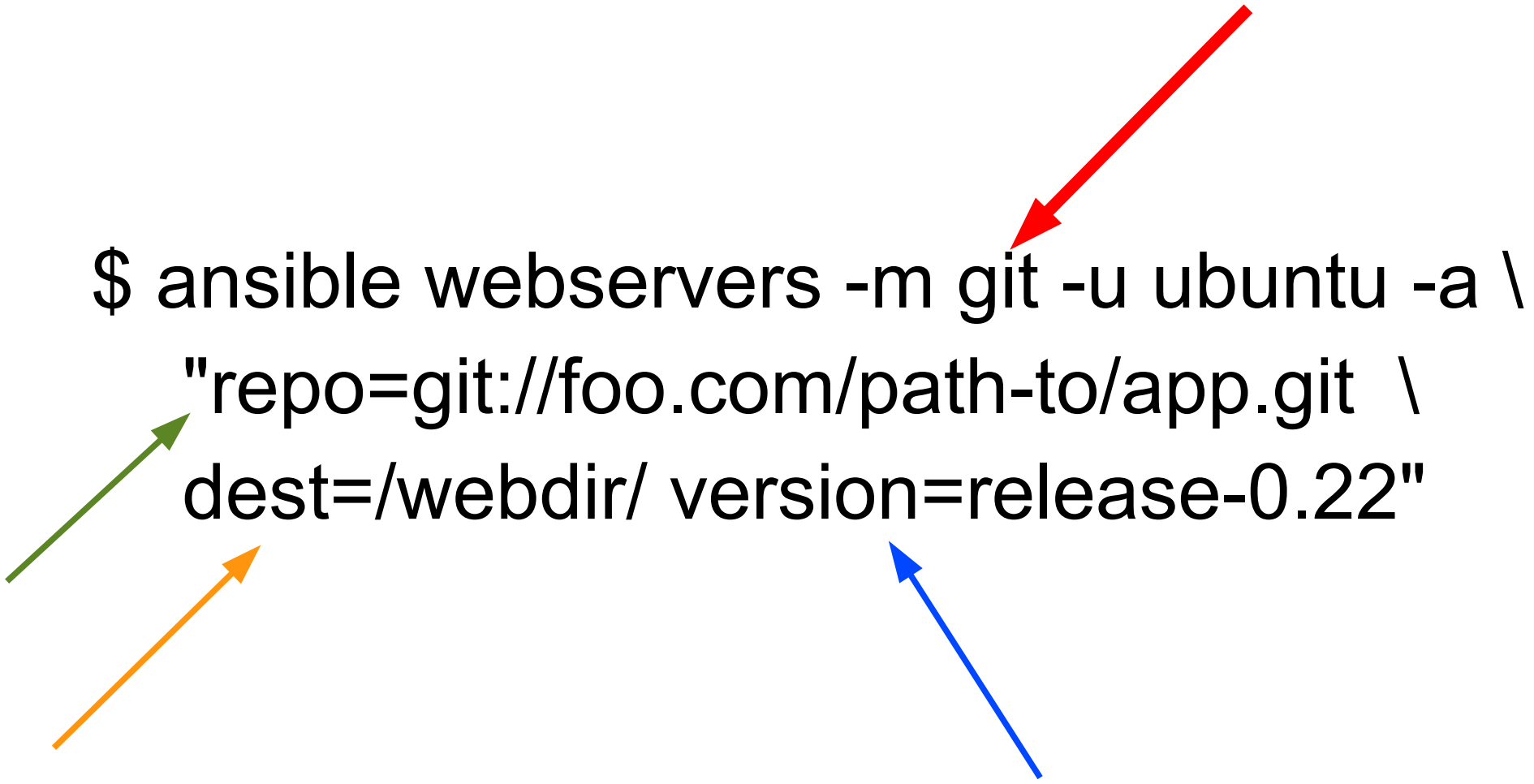
Running Ad-hoc Task
Deploy Example

```
$ ansible web-servers -u ubuntu -m copy \
        -a "src=/etc/hosts dest=/tmp/hosts"
```





Running Ad-hoc Task
Deploy Example



```
$ ansible webservers -m git -u ubuntu -a \
    "repo=git://foo.com/path-to/app.git  \
    dest=/webdir/ version=release-0.22"
```

Using Playbooks

Creating Targets:

```
---
- hosts: web-servers
  user: ubuntu
  sudo: True
```

Using Playbooks

Defining Variables:

```
vars :
    config_msg: “Hello World”

vars_files:
   - /vars/external_vars.yml
```

Using Playbooks

Creating Tasks:

```
tasks:
    - name: update apt repo
      action: command apt-get update
```

Using Playbooks

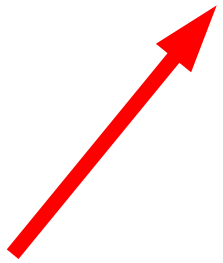
Creating Tasks for DEPLOYMENT

```
tasks:
    - name: deploy app
      action: copy  "src=/app/src.tgz dest=/var/appdir/"
```

Using Playbooks

Defining Handlers

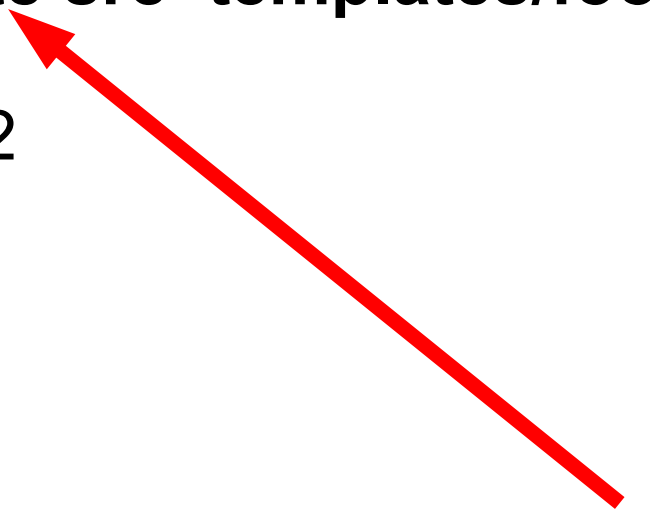
```
handlers:
    - name: reload-apache2
      action: service name=apache2 state=reloaded
```



Using Playbooks

Defining Tasks for Templates

```
tasks:
    - name: write sample template
      action: template src=templates/foo.j2 dest=/tmp/foo.txt
      notify:
      - reload-apache2
```

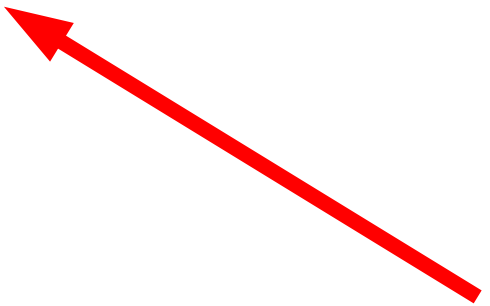


Templates

templates/foo.j2

```
# This is a very simple Jinja2 template representing
#an imaginary configuration file  for an imaginary app.

message = {{ config_msg }}
```



```
---
- hosts: web-servers
  user: ubuntu
  sudo: True

  vars:
     config_msg: "Hello World"

  tasks:
     - name: update apt repo
       action: command /usr/bin/apt-get update

     - name: install apache2
       action: apt pkg=apache2 ensure=installed

     - name: service apache2
       action: service name=apache2 state=started

     - name: write sample template
       action: template src='templates/test.j2' dest='/var/www/test.txt'
       notify:
       - reload apache2

  handlers:
     - name: reload-apache2
       action: service name=apache2 state=reloaded
```

Running Playbooks

$ ansible-playbook ubuntu-apache2.yml

Ansible

Infrastructure as Data

rodney
www.capsunlock.net
https://github.com/cocoy
@imcocoy