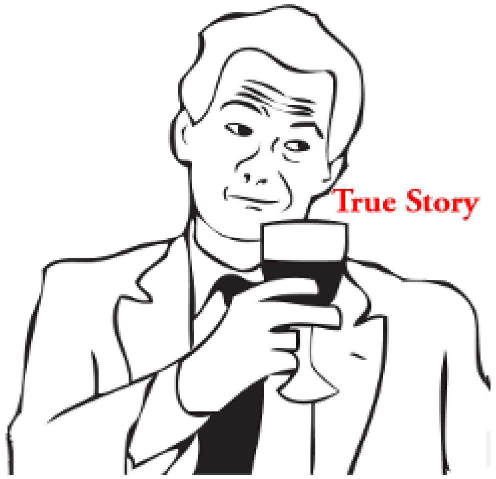
Infrastructure Configuration and Management with Ansible

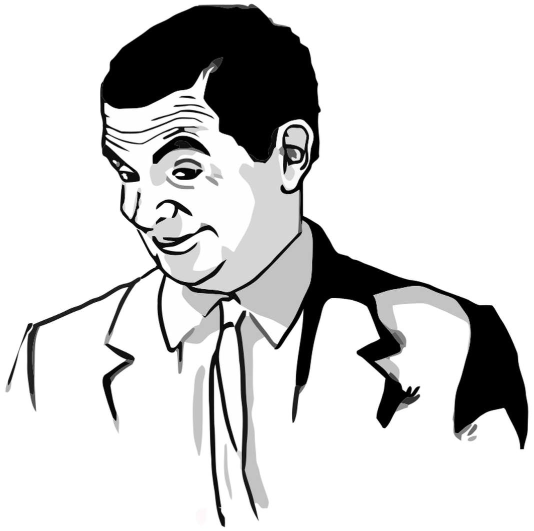
Kaklamanos Georgios

Scientific Computing Center
Aristotle University of Thessaloniki

Central Configuration and Management Tools

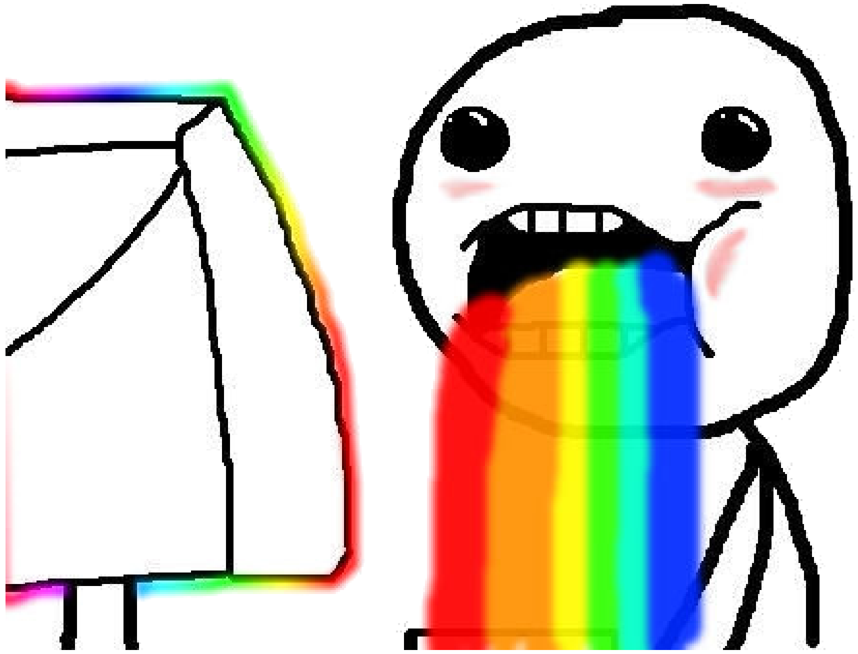
- What are they?
- Why do we need them?
- The three most popular:
  - Chef, Puppet, CFEngine
- The 2 general models:
  - Push vs Pull





Introduction to Ansible...

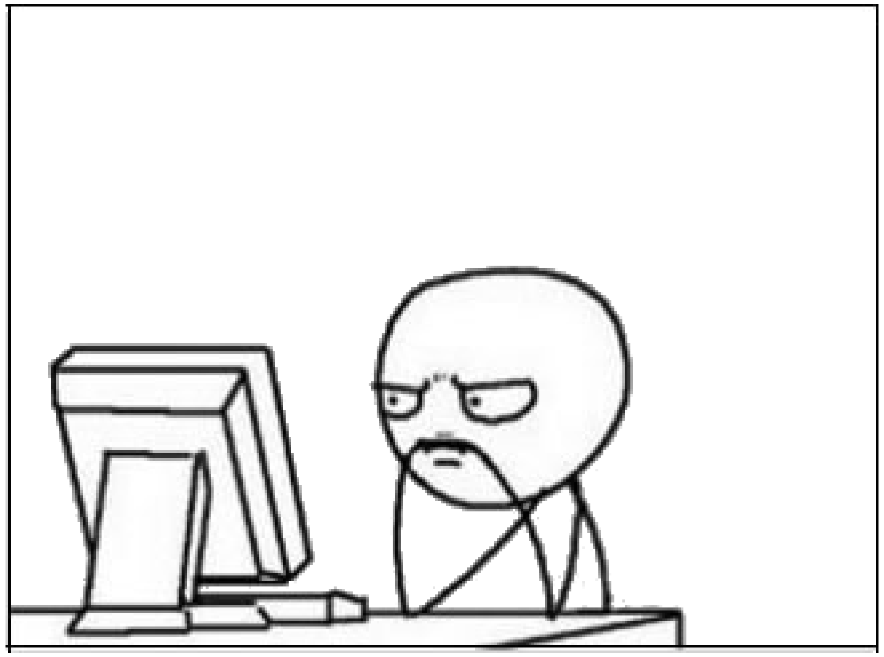
- Initial Release February 2012
- Written in Python
- Works over SSH



- Push modules to hosts from any machine. No daemon or server required.
- Supports both Ad Hoc Execution and Orchestration
- Modules can be written in any language that can return JSON (Python, Ruby, Bash, etc)

Ansible components

- Inventory
- Variables
- Ansible (command)
- Playbook
  - Tasks
  - Handlers
- Files / Templates



Inventory

```
# This is a comment line!
[web-servers]
apache01.example.org
apache02.example.org

[dns-servers]
dns01.example.org
dns02.example.org

[core-services]
web-servers
dns-servers

[centos5-servers]
apache01.example.org
dns01.example.org
```

```
[centos6-servers]
apache02.example.org
dns02.example.org

[local-services]
core-services
ntp01.example.org
...
```

Variables

- User defined
  - host_vars
  - group_vars
- Gathered by ansible
  - setup module
  - Facter available! (used by Puppet)
  - Ohai available! (used by Chef)

```
$ cat host_vars/apache01.example.org
---
extraconfiguration:
  - extra_1.conf
  - extra_2.conf

$ cat group_vars/all
ntp_server: ntp01.example.org

$ ansible apache01.example.com -m setup
apache01.example.org | success >> {
    "ansible_facts": {
        "ansible_architecture": "x86_64",
         "ansible_distribution": "CentOS",
         "facter_architecture": "x86_64",
         "facter_lsbdistid": "CentOS"
         ...
```

Ansible command examples

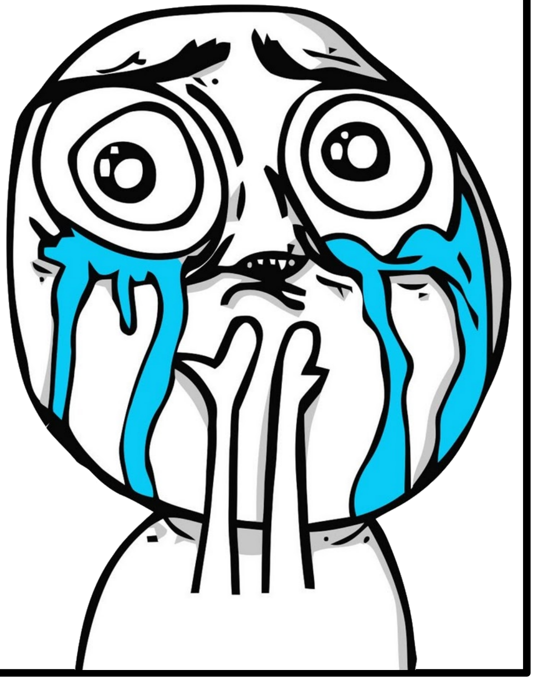
```
$ ansible <host-pattern> [-f forks] [-m module_name] [-a args]

$ ansible all -m shell -a “uptime”

$ ansible all -u ansible -s -m shell -a “yum -y update”

$ ansible web-servers -u ansible -s -m                    \
yum -a “name=httpd state=latest disablerepo=CentAlt”

$ ansible web-servers -u ansible -s -m copy -a            \
  “src=httpd.conf dest=/etc/httpd/conf/httpd.conf         \
  owner=root group=root mode=0644”

$ ansible web-servers -u ansible -s -m service -a         \
“name=httpd state=restarted”
```

Playbooks

```
- hosts: all                            # This will run on all hosts
  user: ansible                         # The users used for login
  sudo: True                            # If there is a need for sudo
  tasks:                                # The tasks to execute
   - include: tasks/vim/install.yml     # Install vim for convenience
   - include: tasks/emacs/install.yml   # and emacs to keep the flames on

- hosts: webservers                     # This will run only on web servers
  user: ansible
  sudo: True
   - include: tasks/httpd/install.yml   # Install apache
   - include: tasks/httpd/configure.yml # Copy the configuration
  handlers:
   - include: handlers/httpd.yml        # Apache handler
   - include: handlers/network.yml      # Network things
```

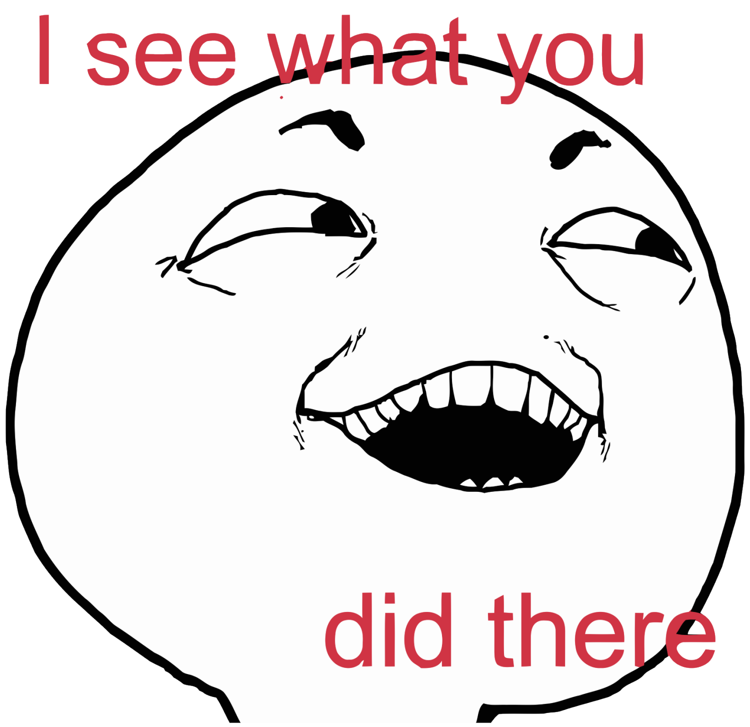
Tasks, actions and handlers

```
$ cat tasks/httpd/install.yml
#### Task to install Apache
- name: Install Apache
  action: yum name=httpd state=latest

$ cat tasks/httpd/config.yml
#### Task to copy Apache configuration
- name: Copy the base Apache configuration.
  action: template src=httpd.conf.j2 dest=/etc/httpd/conf/httpd.conf \
  owner=root group=root mode=0644”
  notify:
    - restart httpd

#### Copy some extra configuration to the conf.d folder.
- name: Copy extra configuration for apache.
  action: copy src=extra_1.conf dest=/etc/httpd/conf.d/extra_1.conf
  notify:
    - restart httpd
...
```



Tasks, actions and handlers

```
$ cat handlers/httpd.yml
#### Handler for various tasks of apache
- name: start httpd
  action: service name=httpd enabled=yes state=started
- name: restart httpd
  action: service name=httpd enabled=yes state=restarted

$ cat handlers/network.yml
#### Handler for various network tasks.
- name: restart network
  action: service name=network state=restarted
- name: eth1 up
  action: shell /sbin/ifconfig eth1 up
- name: restart iptables
  action: service name=iptables state=restarted
```

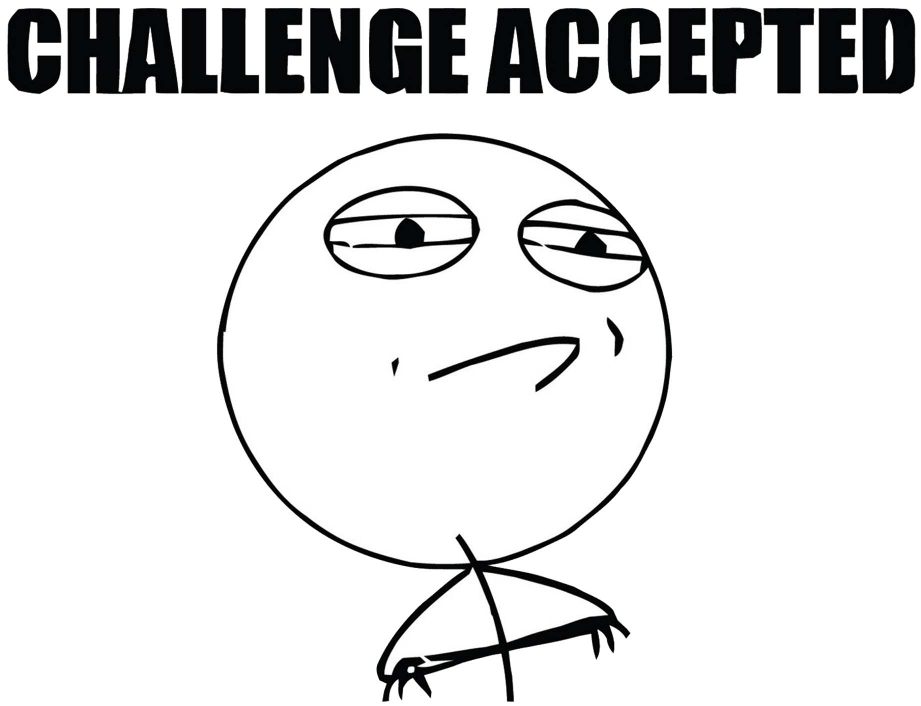


Files and Templates

- Files are just regular files from your local host...

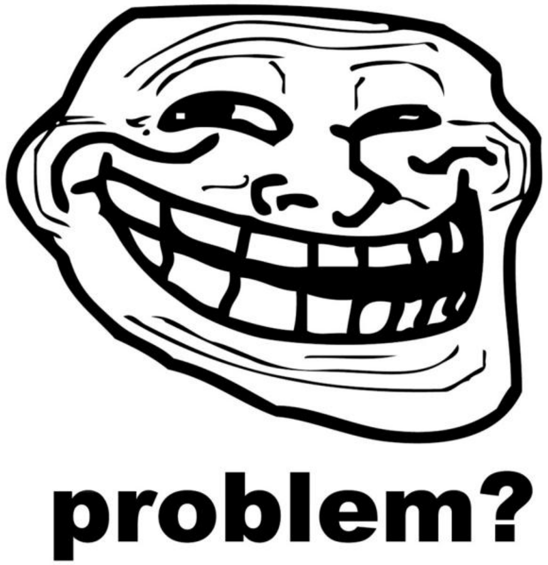
- Templates are written in Jinja2 and are populated using the variables from host_vars, group_vars or ansible_facts.

```
{% if extra_configuration %}

{% for config in extra_configuration %}

include conf.d/{{ config }}

{% endfor %}

{% endif %}
```



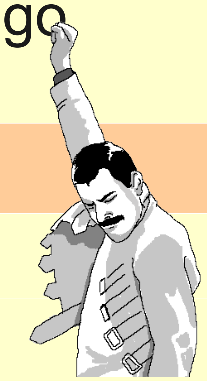
Comparison to others

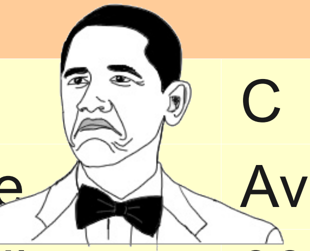
- Still a new project
- Lacks some of the older project features
  - dry run is still new
  - no rollback (yet)
  - optional support for pull mode
- Simple SSH, no daemons or clients
- Simple and easy to understand!
- Human Readable!

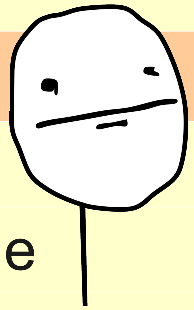


Comparison to others (by www.ohloh.net)

	Ansible	Puppet	Chef	CFEngine
Project Activity	Very High	Very High	Very High	Very High
Project Licence	GPL-3.0+	Apache-2.0, GPL-2.0+	Apache 2.0	GPL-3.0
Estimated Cost	$421,296.00	$4,744,764.00	$1,195,657.00	$1,308,710.00
All Time Statistics				
Contributors	231 developers	273 developers	277 developers	61 developers
Commits	4,940 commits	22,499 commits	7,453 commits	6,786 commits
Initial Commit	about 1 year ago	about 8 year ago	about 5 years ago	over 5 years ago
Most Recent Commit	1 day ago	about 16 hours ago	1 day ago	about 20 hours ago
Code Analysis				
Mostly Written In	Python	Ruby	Ruby	C
Comments	High	Low	Average	Average
Lines of Code	32,955 lines	323,329 lines	87,629 lines	96,230 lines







Questions?

I DONT ALWAYS ASK QUESTIONS
BUT WHEN I DO IT'S ABOUT ANSIBLE
memegenerator.net

Thank you!