Ansible

DevOps for people who hate DevOps

John Lynch
CTO Kaleo Software
@johnrlynch
www.rigelgroupllc.com

In the beginning...

- Developers wrote code
- Systems Administrators deployed code

...until one day...

- I know, I’ll write code to tell the computer how to set up **itself**!

Shell Scripts

```
#!/bin/sh
echo "fs.file-max=20000" | sudo tee -a /etc/sysctl.conf
sudo apt-get -y install build-essential postgresql-9.2
sudo apt-get -y install apache2
sudo a2enmod rewrite
sudo tee /etc/apache2/sites-available/mysite <<ENDOFFILE
  <VirtualHost *:80>
    Include /etc/apache2/sites-available/mysite-common
  </VirtualHost>
ENDOFFILE
```

Shell Scripts :(

- Not idempotent
- Not “robust”
- Everyone’s rolling their own

Shell Scripts

+

VC Cash

=

“Model-driven frameworks orchestration complex infrastructure API organic solution automation databag”

puppet labs

OPSCODE

I need servers to manage and deploy my servers.

WAT?

Stop the madness!

Ansible

Simple to start

Scales “up” when you need it to

Ansible

The “Ansible”

Dependencies

Python
Jinja2
PyYAML
Paramiko

SSH

Servers

Python

Ansible

```
$ brew install python

$ pip install jinja2
$ pip install PyYAML
$ pip install paramiko
$ pip install ansible
```

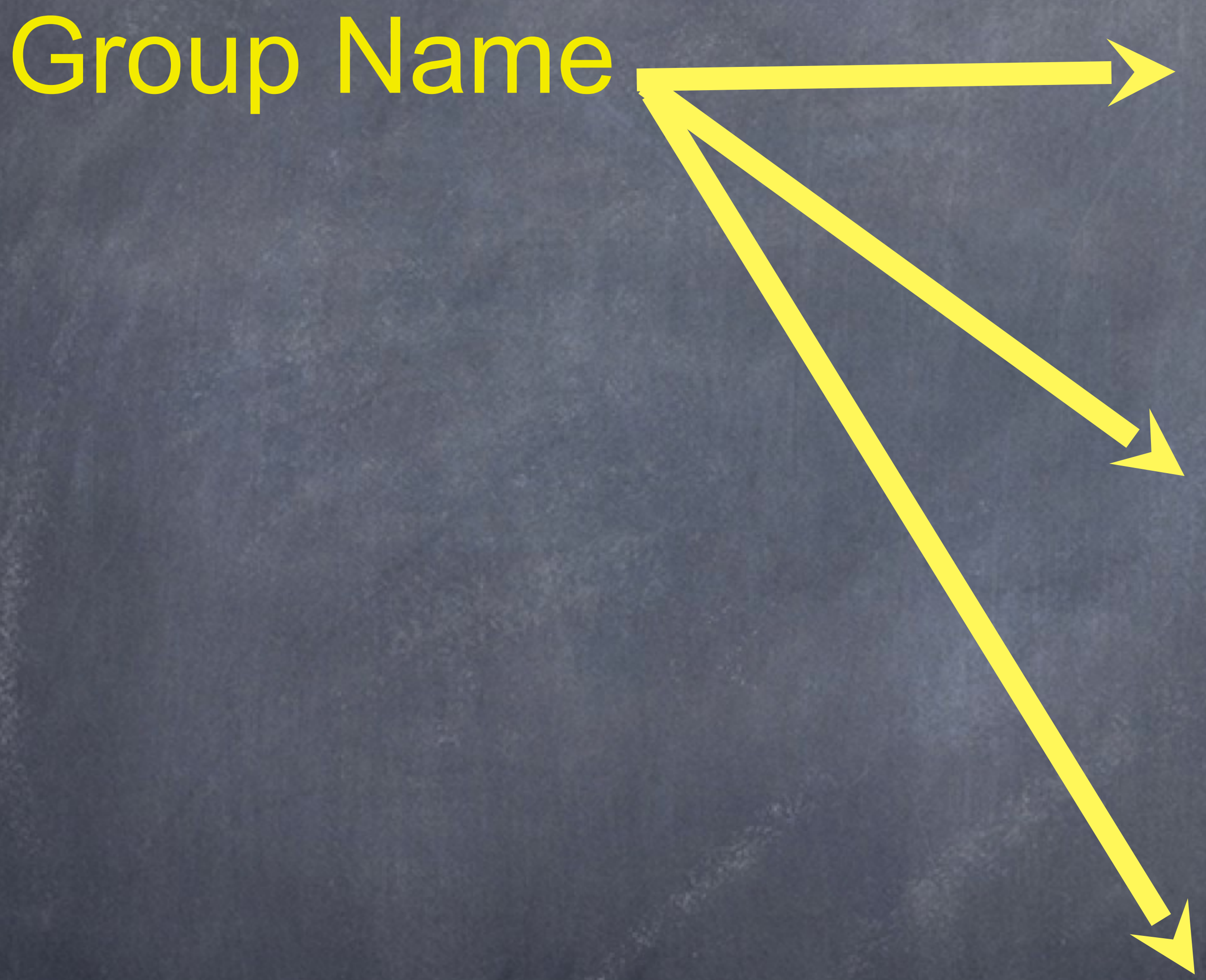
The Basics

- SSH Configuration

```
$ cat ~/.ssh/config

IdentityFile ~/.ssh/id_rsa
IdentityFile ~/.ssh/github_rsa
IdentityFile ~/.ssh/ec2_east
```

~/ansible_hosts



```
[local]
127.0.0.1

[web-group]
www.bsg.mil
www2.bsg.mil

[db-group]
10.0.1.123
```

~/ansible_hosts

Path to python

```
[local]
127.0.0.1
ansible_python_interpreter=/usr/local/Cellar/python/2.7.3/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python
```

Ad-Hoc Commands

```
$ ansible all -m ping
$ ansible web-group -m ping
$ ansible all -a "/bin/echo hi"
```

```
$ ansible all -a "/sbin/reboot" -f 10
$ ansible all -m user -a "name=joe password=wat"
$ ansible all -m yum -a "name=nginx state=installed"
$ ansible all -m service -a "name=nginx state=started"
```

Ad-Hoc Commands

```
$ ansible local -m setup
127.0.0.1 | success >> {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "192.168.2.112"
        ],
        "ansible_architecture": "x86_64",
        "ansible_distribution": "MacOSX",
        "ansible_distribution_version": "10.8.4",
        "ansible_domain": "local",
        "ansible_fqdn": "Johns-MacBook-Pro.local",
        "ansible_hostname": "Johns-MacBook-Pro",
        "ansible_kernel": "12.4.0",
        "ansible_machine": "x86_64",
        "ansible_memfree_mb": 4035,
        "ansible_memtotal_mb": 16384,
        "ansible_model": "MacBookPro10,1",
        "ansible_os_family": "Darwin",
        "ansible_osrevision": "199506",
        "ansible_osversion": "12E55",
        "ansible_pkg_mgr": "unknown",
        "ansible_processor": "Intel(R) Core(TM) i7-3740QM CPU @ 2.70GHz",
        "ansible_processor_cores": "4",
        "ansible_python_version": "2.7.3",
        "ansible_system": "Darwin",
        "ansible_user_id": "john"
    },
    "changed": false
}
```

Playbooks

- YAML Files
- Declaratively define your configuration
- Can contain many “Plays” targeting different Groups

my-playbook.yml

```
---
- hosts: webservers
  user: root
  vars:
    http_port: 80
    max_clients: 200
  tasks:
    - name: ensure apache is at the latest version
      action: yum pkg=httpd state=latest
    - name: write the apache config file
      action: template src=httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running
      action: service name=httpd state=started
  handlers:
    - name: restart apache
      action: service name=httpd state=restarted
```

my-playbook.yml

```yaml
---
- hosts: webservers
  user: root
  vars_files:
    - settings.yml
  tasks:
    - name: ensure apache is at the latest version
      action: yum pkg=httpd state=latest
    - name: write the apache config file
      action: template src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running
      action: service name=httpd state=started
  handlers:
    - name: restart apache
      action: service name=httpd state=restarted
```

my-playbook.yml

```
---
- hosts: webservers
  user: root
  vars_files:
    - settings.yml
  tasks:
    - name: ensure apache is at the latest version
      action: yum pkg=httpd state=latest
    - name: recursively copy files from local to target
      local_action: command rsync -a /path/to/files
{{ inventory_hostname }}:/path/to/target/
    - name: ensure apache is running
      action: service name=httpd state=started
  handlers:
    - name: restart apache
      action: service name=httpd state=restarted
```

Running Playbooks

```
$ ansible-playbook mysite.yml -f 10
$ ansible-playbook mysite.yml --list-hosts
```

Launch Instance

```
$ ec2-run-instances ami-bfd3a3d6 \
                    -g Web \
                    -k ansible-ec2-us-east \
                    --instance-type m1.small

$ tee ~/.ssh/config <<ENDOFFILE
Host ec2-204-236-240-204.compute-1.amazonaws.com
IdentityFile ~/.ssh/ansible-ec2-us-east.pem
ENDOFFILE

$ tee ~/ansible_hosts <<ENDOFFILE
[ec2]
ec2-204-236-240-204.compute-1.amazonaws.com
ENDOFFILE

$ ansible ec2 -u ubuntu -m ping
```

Demo

Ansible 201

Conditional Task Execution

Lookup vars in external files

Fireball Mode

Ansible 201

Ansible Modules for:

EC2, Rackspace, Linode, OpenStack, Digital Ocean

Route53, S3, RDS

MySQL, Postgres, Riak, Mongo

Airbrake, Monit, Nagios, NewRelic, Pingdom

Netscaler, BigIP, Arista

FlowDock, HipChat, IRC, Jaber, Email

Thank You!