Ansible

swiss army knife
orchestration

Brian Coca
May 2013

My Ops evolution

manual updates + logbook

at + .bat scripts (yes, Windows)

{telnet, rsh, ssh} + for loops + command line

shells/perl/python custom scripts + cssh/pssh

CM and Provisioning Applications

What I DON'T want

agents/daemons (more software to manage)

ports (more firewall rules)

ssl/pki (cert/key management!!!)

any setup in general (I'm lazy that way)

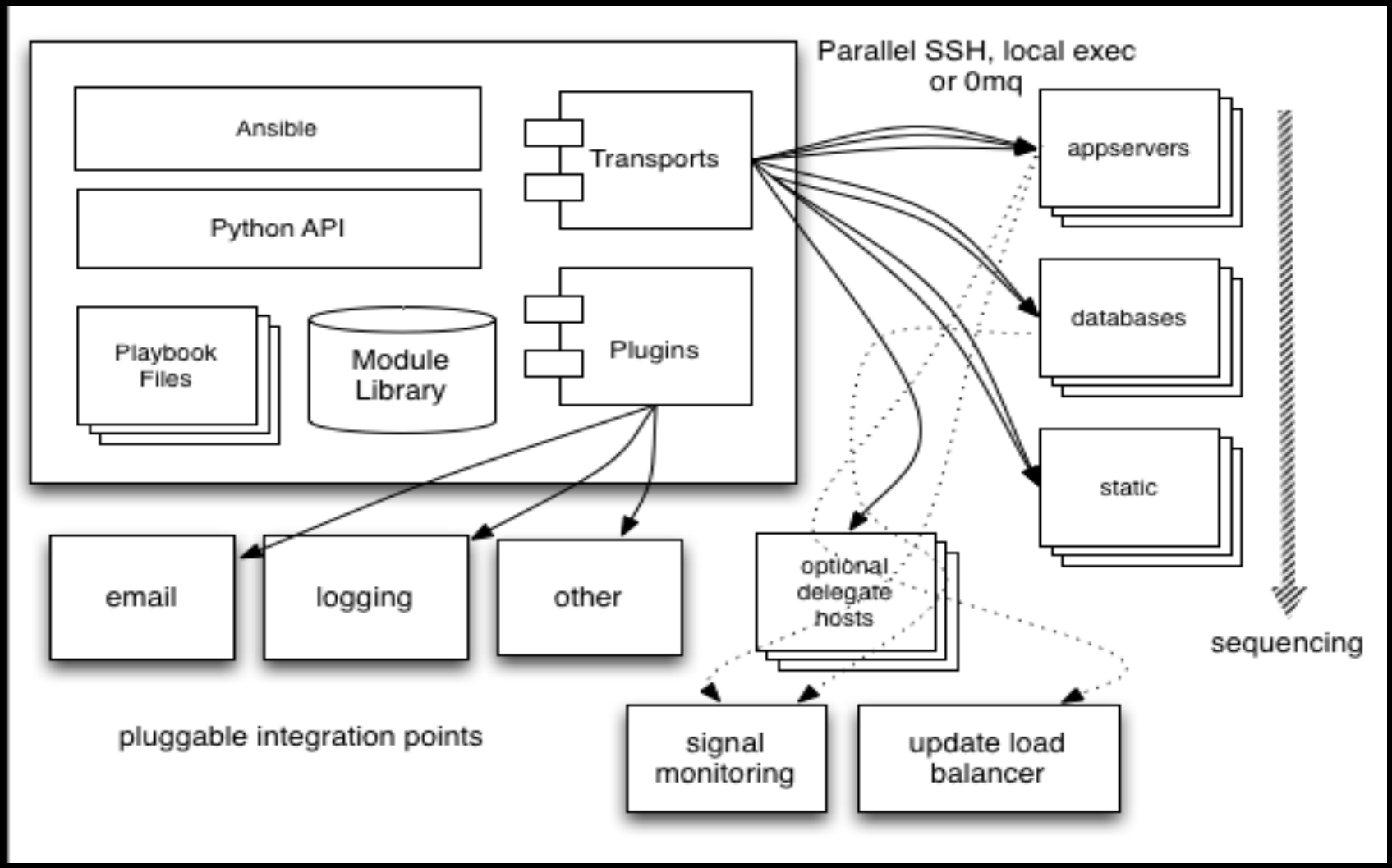
translations for provision, config and ad-hoc

Enter Ansible

- Minimal setup (my definition: just needs Python*)
- Leverages existing auth{entication,orization}
  - SSH as default transport
  - Can use sudo or root keys
- Supports ad-hoc (1 liner reuse!)
- Can run from checkout (port/rpm/deb/pip/etc)
- Predictable, easily expandable, portable
- Aims to be idempotent

UNIX PHILOSOPHY

The diagram


Ansible
Python API
Playbook Files
Module Library
Transports
Plugins
Parallel SSH, local exec or 0mq
appservers
databases
static
email
logging
other
optional delegate hosts
sequencing
pluggable integration points
signal monitoring
update load balancer

Terminology

Inventory: list of hosts, groups and variables

Modules: actually do the work, any language

Plugins: callback, action and other hooks

Facts: data gathered from target hosts

Playbook: collections of plays

Play: loops over a list of tasks mapped to a list of hosts

Task: invokes a module to do the work (handlers are a type of task)

Inventory (hosts and groups)

Simple ini:

```
[web]
web0
web1.example.com ansible_ssh_port=1234
web[02:15]

[db]
mysql[1:3] mysql_port=6666
mysqlbackup ansible_ssh_host=10.10.1.23
```

Can be dynamic (+x):

- ec2.py
- cobbler
- just return json (easy to write your own)
- directory: will use all contained files

Modules

at first I knew all core, now ... :

add_host,debug,get_url,mount,postgresql_user,slurp,
apt,django_manage,git,mysql_db,rabbitmq_parameter,
subversion,apt_key,easy_install,group,mysql_user,
rabbitmq_plugin,supervisorctl,apt_repository,ec2,group_by,
nagios,rabbitmq_user,svr4pkg,assemble,ec2_facts,hg,ohai,
rabbitmq_vhost,sysctl,async_status,ec2_vol,ini_file,opkg,raw,
template,async_wrapper,facter,libr,pacman,script,uri,
authorized_key,fail,lineinfile,pause,seboolean,user,
cloudformation,fetch,lvol,ping,selinux,virt,command,file,
macports,pip,service,wait_for,copy,fireball,mail,pkgin,setup,
yum,cron,gem,mongodb_user,postgresql_db,shell,zfs

* The list was probably outdated at the time I updated it

Plugins

- callback/action
  can use data returned (log, db update)
  or as triggers (update dns)

- connection
  ssh alternatives (local, chroot, etc)

- inventory, lookup, vars, filter
  data from other sources or formats

Facts

On by default (gather_facts: False to disable)

Can use ohai/facter or custom fact modules

BSDs, Solaris, AIX lag behind, but not much
 * mdehaan: 'send me a patch :-)'

Usable in plays and modules (template)

Facts

```
#>ansible -m setup testbsd|head -n 10
testbsd | success >> {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "10.1.1.4"
        ],
        "ansible_all_ipv6_addresses": [],
        "ansible_architecture": "amd64",
        "ansible_machine": "amd64",
        "ansible_memfree_mb": 1161,
        "ansible_memtotal_mb": 15850,
```

Playbooks Plays and Tasks

nginx.yml

```
---
- hosts: web
  sudo: True
  vars:
       domain: example.com
  tasks:
    - name: Install nginx
      action: shell pkg_add -r nginx creates=/usr/local/bin/nginx

    - name: configure nginx
      template: src=templates/nginx.conf.j2 dest=/usr/local/etc/nginx.conf
      notify: nginx-restart

  handlers:
    - name: nginx-restart
      service: name=nginx state=restarted
```

```
#> ansible-playbook -K nginx.yml
```

Playbooks Plays and Tasks (again)

Organize it your way!

```
users/
users/app.yml
users/admins.yml
users/departed.yml
users/devs.ym
users/user_actions.yml
users/files
users/files/keys
users/files/keys/user1.keys
users/files/keys/user2.keys
users/var_files
users/var_files/users.yml
users/var_files/app.yml
```

admins.yml

```
- name: Ensure ops users exist everywhere with wheel
  hosts:  all
  gather_facts: False
  sudo: True
  vars_files : [ var_files/users.yml ]
  vars:
    - active: [ bcoca, eoot, dog ]
    - user_groups: '${admin_group}'

  tasks:
    - include: user_actions.yml
       with_items: $active
```

Plays and Tasks (user_actions.yml)

```
- name: Ensure user exists
  user:  >
    name=${item}
    uid=${users.${item}.id}
    password="${users.$item.pass}"
    comment="${users.$item.name}"
    groups="${user_groups}"
    shell=${ansible_bash_interpreter}
    append=true
    state=present

 - name: Ensure .ssh exists
   file: dest=/home/${item}/.ssh state=directory owner=${item} group=${item} mode=0700

 - name: Setup ssh keys
   authorized_key: key="$FILE(files/keys/${item}.keys)" user=${item}

 - name: Log it now!
   local_action: shell logger "Verified user: ${item} at $ansible_hostname"
```

Variable files (users.yml)

```
users:
  alan:
     id: 1069
     name: Alan Parsons
     pass: '$1$UxoGyasdfX$kOj0MgisMK.eYtgRuP7R0'
  bcoca:
     id: 1056
     name: Brian Coca
     pass: '$2a$04$ra3LpTMihRH5v6/YASDF/33Fyhe8SWk/RZuJiN5wK'
  brian:
     id: 1096
     name: Brian May
     pass: '$2a$04$Sx.lL2ejqe6bxOeNSCAh/f8nd1.q9rO/ER2gW'
```

ad-hoc

#>ansible webs -m shell -a "awk '{print \$9}' /var/log/nginx/access.log|sort |uniq -c |sort -k1,1nr 2>/dev/null|column -t"

```
web1 | success | rc=0 >>
204417    200
48108     304
8550      302
6541      301
1696      404
269       206
web2 | success | rc=0 >>
205807    200
43762     304
```

And many other features (I've ignored/glossed over)

- delegate_to, tags, roles
- local, pull and fireball modes, ansible-doc
- fork and serial
- conditionals (only_if, when_+, notify)
- cowsay, jinja in playbooks
- --check, --diff, --limit, --list-+, --step, ...
- $PIPE, $FILE, $PASSWORD, ...
- 3rd party modules (http://ansible.cc/docs/contrib.html)
- and others I forgot or don't know
- new features daily, releases ~ 2 months

FreeBSD bootstrap

```
- hosts: all
  gather_facts: false
 vars:
   - ansible_python_interpreter: /usr/local/bin/python
  tasks:
     - name: check for python
       raw: which python
       register: pypath
       ignore_errors: True

    # Ansible targets require python >=2.4 + json, 2.6  and above include json
     - name: install python
       raw: pkg_add -r python27
       when_failed: $pypath

     - name: now start working with ansible!
       template: src=templates/sudoers dest=/usr/local/etc/sudoers validate='visudo -c %s'
```

My (BSD) TODO list:

- ~~Service Management~~
- ~~Network facts~~
- Jails: facts and ~~management~~ (service module)
- ZFS: facts and ~~management~~
- General facts improvement (distro, release, product, osfamily, etc)
- Package management {Free, ~~Open~~, ~~Net~~}BSD
- Improve FreeBSD port

More info and credits

http://ansible.cc
http://groups.google.com/group/ansible-project
#Ansible irc.freenode.net
http://ansible.cc/docs/contrib.html (3rd party)

Slides/info I sto... er borrowed:

https://speakerdeck.com/mpdehaan/ansible
http://www.slideshare.net/gnosek/ansible
http://jpmens.net/2012/06/06/configuration-management-with-ansible/
https://speakerdeck.com/jpmens/ansible-an-introduction
http://www.feyrer.de/NetBSD/pkgsrcCon2013/

Other examples:

bin/departed

```
#!/usr/bin/env ansible-playbook -K
---
- name: Ensure only valid users
  hosts: all
  gather_facts: False
  sudo: True
  vars:
     departed: [ alan, bcoca, isaac, mathew, mike, venizio, willy ]
  tasks:
    - name: Delete departed user
      user: name=${item} state=absent remove=yes
      with_items: $departed
```

Interactive release (release.yml):

```
---
- hosts: localhost
  sudo: False
  gather_facts: False
  vars_prompt:
    - name: "branch"
      prompt: "Enter branch"
      private: False
  tasks:
    - pause:  prompt="Hit 'Enter' key to start deployment"
    - debug:  msg="deploying $repo/$branch to $target"
    - include: mainapp/prep.yml
    - include: secondaryapp/prep.yml

- hosts: $target
  sudo: true
  gather_facts: true
  vars:
    monitor: mynagiosserver
  tasks:

    - include: mainapp/deploy.yml
    - include: services/nuke.yml
    - include: mainapp/update.yml
    - include: dbhost/update.yml
      only_if: is_set($dbhost)
    - include: services/unnuke.yml
```

Interactive Release (services/nuke.yml):

```
- pause: "are you sure you want to stop all services?"
# Take out of rotation and stop services
 - name: shush nagios
   nagios: action=silence host=$inventory_hostname
   delegate_to: ${monitor}
   tags:  [   mon ]

 - name: nginx graceful stop
   action: shell /usr/local/etc/rc.d/nginx gracefulstop
   async: 60
   register: graceful
   ignore_errors: true
   tags:  [ stop ]

 - name: nginx forceful
   action: service name=nginx state=stopped
   when_integer: ${graceful.rc} != 0
   tags:  [ stop ]

 - name: stop svscan
   action: service name=svscan state=stopped ....
```

netbsd + mysql + ec2

```
---
- hosts:
  - security_group_ec2-dbservers
  sudo: yes
  gather_facts: false

  tasks:
   - name: Install mysql
     action: pkgin name=mysql-server-5.1.65 state=present

   - name: Install MySQL rc.d script
     template: src=/usr/pkg/share/examples/rc.d/mysqld dest=/etc/rc.d/mysqld mode=0755

   - name: Start MySQL service
     service: name=mysqld enabled=yes state=started

   - name: Install python-mysqldb (for mysql_user module)
     action: pkgin name=py27-mysqldb state=present

   - name: Setup DB
     mysql_db: db=webapp state=present

   - name: Add db-user
     mysql_user: name=webapp password=webapp state=present priv='webapp.*:INSERT,UPDATE,DROP,CREATE,
ALTER,LOCK TABLES,SELECT'

   - name: Copy over DB template
     copy: src=db/dump-names.sql dest=/tmp/dump-names.sql

   - name: Import DB data
     mysql_db: db=webapp state=import target=/tmp/dump-names.sql  login_user=webapp login_password=webapp
```

netbsd + mysql + ec2 output

```
% env ANSIBLE_HOSTS=./ec2.py ansible-playbook config-ec2-dbserver.yml

PLAY [security_group_ec2-dbservers] ********************

TASK: [Install mysql] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

TASK: [Install MySQL rc.d script] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

TASK: [Start MySQL service] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

TASK: [Install python-mysqldb (for mysql_user module)] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

TASK: [Setup DB] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

TASK: [Add db-user] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

TASK: [Copy over DB template] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

TASK: [Import DB data] **********************
changed: [ec2-anon.compute-1.amazonaws.com]

PLAY RECAP **********************
ec2-anon.compute-1.amazonaws.com : ok=8    changed=8    unreachable=0    failed=0
```