

Ansible

Systems configuration doesn't *have* to be complicated

Jan-Piet Mens

March 2013

@jpmens

**@jpmens: consultant,
author, architect, part-time
admin, small-scale fiddler,
loves LDAP, DNS,
plain text, and things
that **work**.**

once upon a **time**, we
had **shell** scripts and
SSH loops

then it got
complicated ...

this is what we
want:

Terminal

```
jpm@jmbp /etc/ansible $ echo kanga >> hosts
jpm@jmbp /etc/ansible $ ansible-playbook tmux.yml
```

```
PLAY [all] *****
```

```
TASK: [Install tmux package] *****
```

```
changed: [kanga]
```

```
ok: [piglet]
```

```
ok: [roo]
```

```
ok: [pooh]
```

```
ok: [eeyore]
```

```
ok: [tigr]
```

```
TASK: [Configure tmux] *****
```

```
changed: [kanga]
```

```
ok: [piglet]
```

```
ok: [roo]
```

```
ok: [pooh]
```

```
ok: [eeyore]
```

```
ok: [tigr]
```

```
PLAY RECAP *****
```

eeyore	: ok=2	changed=0	unreachable=0	failed=0
kanga	: ok=2	changed=2	unreachable=0	failed=0
piglet	: ok=2	changed=0	unreachable=0	failed=0
pooh	: ok=2	changed=0	unreachable=0	failed=0
roo	: ok=2	changed=0	unreachable=0	failed=0
tigr	: ok=2	changed=0	unreachable=0	failed=0

```
jpm@jmbp /etc/ansible $ █
```

No more daemons

No more agents

Not another PKI

Not another host

No more ports

No databases

Automation should **not**
require programming
experience; it **MUST**
[RFC 2119] **be easy**

We all have other stuff to do, don't we?

compréhensible

welcome to
Ansible

push-based
pull possible

from **zero** to **prod** in
minutes

Python

2.6 + Paramiko, PyYAML, Jinja2 on **manager**

2.4 + simplejson on **nodes**

Can run in *virtualenv* and from *git* checkout

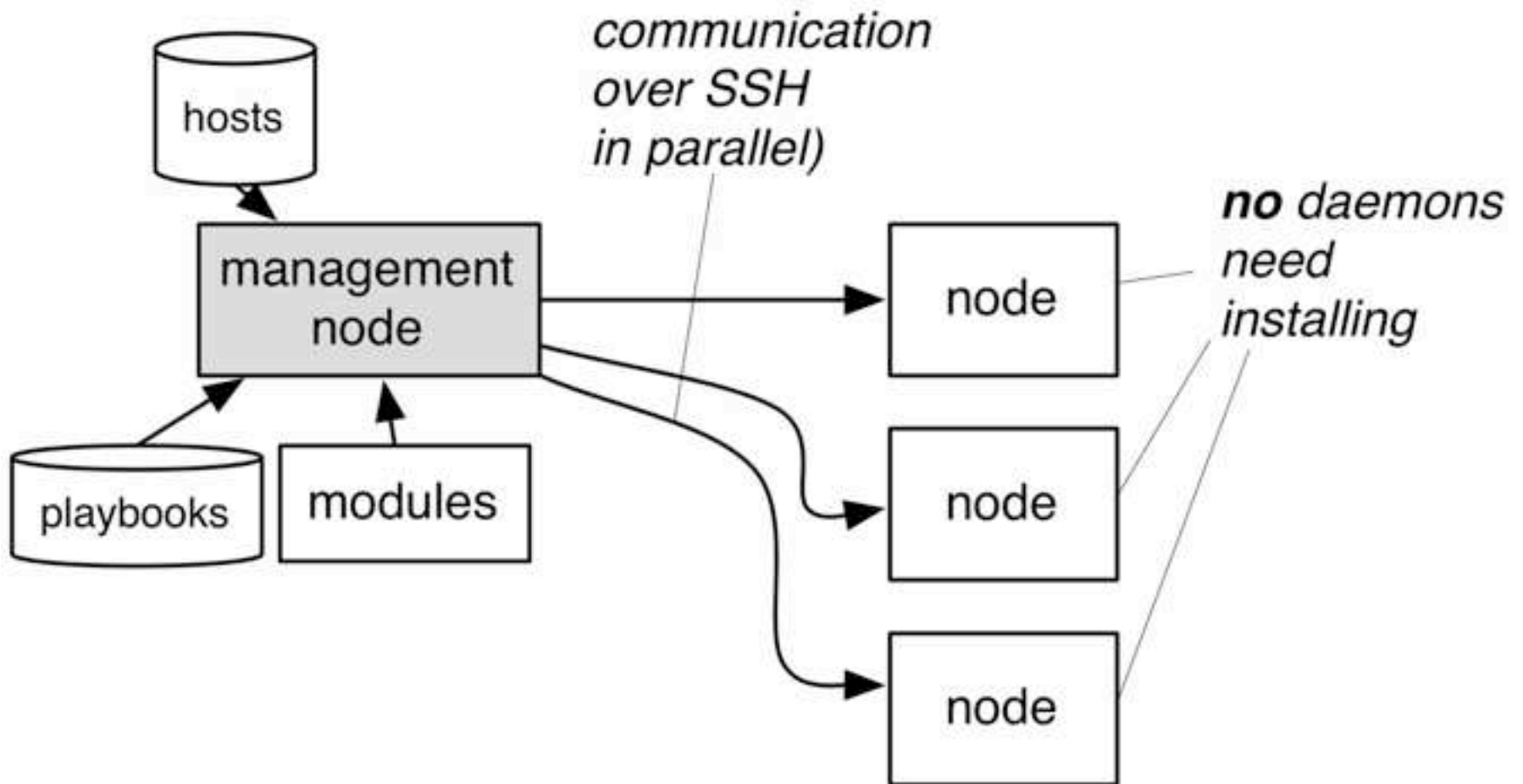
SSH

keys, Kerberos, passwords



doesn't need **root**
can **sudo**

Modus operandi



Do this **once, now**

ad-hoc

Install **packages**

yum, apt, and no, you don't want zypper do you?

Minimal **config** language

no XML, no Ruby, no ...

Inventory

```
$ cat ${ANSIBLE_HOSTS:-/etc/ansible/hosts}
```

```
[local]  
127.0.0.1
```

```
[webservers]  
www.example.com ntp=ntp1.pool.ntp.org  
web[10-23].example.com  
sushi ansible_ssh_host=127.0.0.1 ansible_ssh_port=222
```

```
[devservers]  
a1.ww.mens.de
```

executable hosts

- CMDB (LDAP, SQL, etc.)
- Cobbler
- EC2, OpenStack, etc.
- make your own: **JSON**

Target selection

webservers

all

ldap.example.com

webservers:!web20.example.com

*.example.com

192.168.6.*

ad-hoc copy

```
$ ansible devservers -m copy -a 'src=resolver.conf dest=/etc/resolver.conf'
a1.ww.mens.de | success >> {
    "changed": true,
    "dest": "/etc/resolver.conf",
    "group": "adm",
    "md5sum": "c6fce6e28c46be0512eaf3b7cfdb66d7",
    "mode": "0644",
    "owner": "jpm",
    "path": "resolver.conf",
    "src": "/home/jpm/.ansible/tmp/ansible-322091977449/resolver.conf",
    "state": "file"
}
```

facts

```
"ansible_architecture": "x86_64",  
  "ansible_default_ipv4": {  
    "address": "192.168.1.194",  
    "gateway": "192.168.1.1",  
    "interface": "eth0",  
    "macaddress": "22:54:00:02:8e:0f",  
  },  
  "ansible_distribution": "CentOS",  
  "ansible_distribution_version": "6.2",  
  "ansible_fqdn": "a1.ww.mens.de",  
  "ansible_hostname": "a1",  
  "ansible_processor_count": 1,  
  "ansible_product_name": "KVM",  
  "ansible_swapfree_mb": 989,
```

Plus **ohai** and **facter** if installed on node

modules

apt, apt_repository, assemble, async_status, authorized_key, command, copy, cron, debug, easy_install, facter, fail, fetch, file, fireball, get_url, git, group, ini_file, lineinfile, mail, mount, mysql_db, mysql_user, nagios, ohai, pause, ping, pip, postgresql_db, postgresql_user, raw, seboolean, selinux, service, setup, shell, slurp, subversion, supervisorctl, template, user, virt, yum

Plus many more: **provisioning**, contrib, etc.

Playbooks

- YAML
- OS **configuration**
- APP **deployment**
- collections of **actions** using **modules**
- each group of actions is a **play**
- notification **handlers**

Install, configure **tmux**

- hosts: devservers
 - user: f2
 - sudo: True
 - vars:
 - editmode: vi
 - tasks:
 - name: Install tmux package
 - action: yum name=tmux state=installed
 - name: Configure tmux
 - action: template src=tmux.conf.j2 dest=/etc/tmux.conf
 - name: Tell master
 - action: shell echo "\${ansible_fqdn} done" >> /tmp/list
 - delegate_to: k4.ww.mens.de

variables

- From **inventory**
- In **plays**
- From **host_vars/** files
- From **group_vars/** files
- From **register**

editmode: emacs

admin: Jane Jolie

location: Bldg Z8/211

{{ templates }}

templates in Jinja2

```
# {{ ansible_managed }}
```

```
{# editmode is either "vi" or "emacs" #}
```

```
set -g prefix C-a
```

```
set -g status-utf8 on
```

```
setw -g mode-keys {{ editmode }}
```

```
# Ansible managed: tmux.conf.j2 modified on 2012-10-14 09:47:11 by jpm on hippo
```

```
set -g prefix C-a
```

```
set -g status-utf8 on
```

```
setw -g mode-keys vi
```

generate /etc/hosts

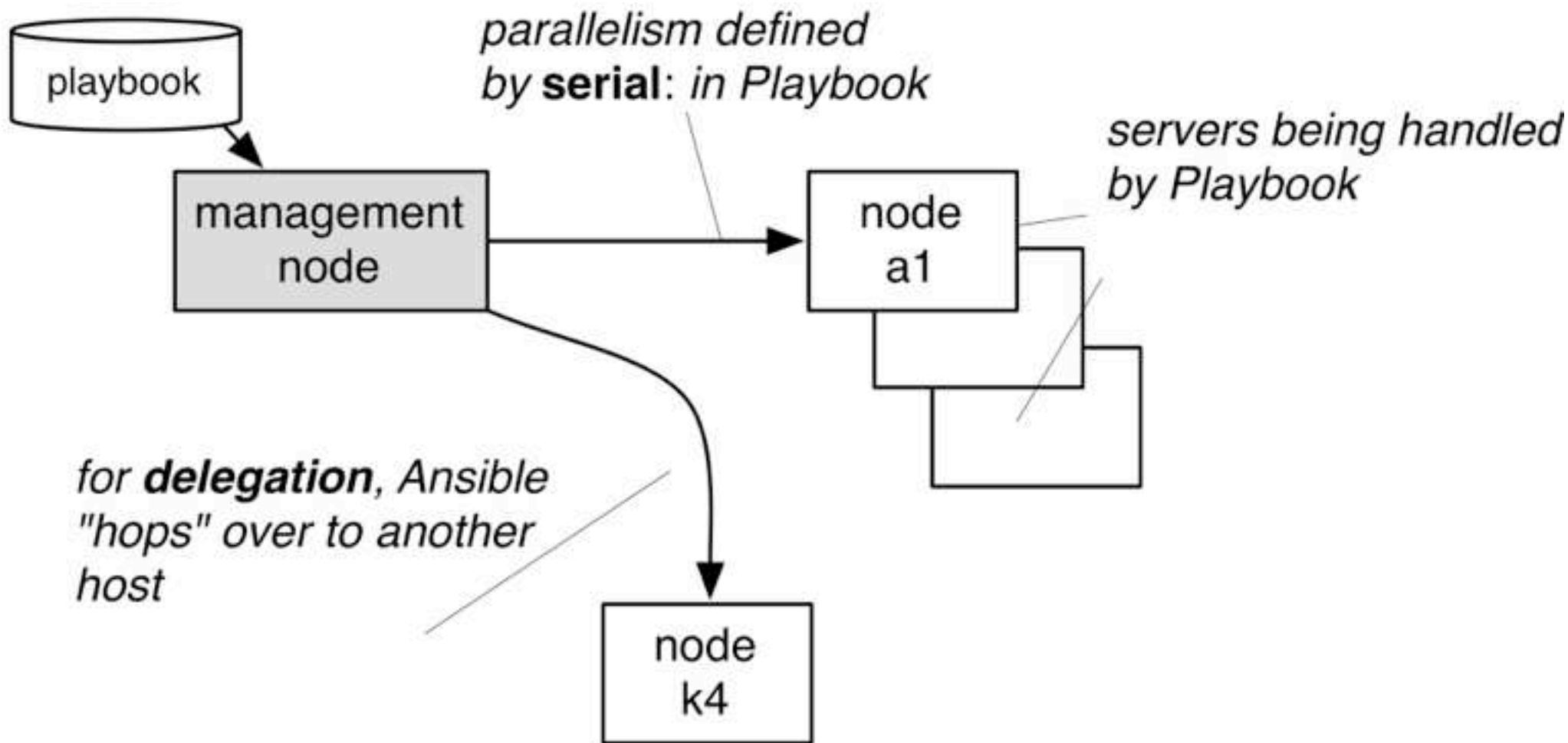
```
{% for k,v in hostvars.iteritems() -%}  
  {{ v['ansible_eth0']['ipv4']['address'] }} {{ k }} \br/>  {{ v['ansible_hostname'] }}  
{% endfor %}
```

```
192.168.1.218 k4.ww.mens.de k4  
192.168.1.194 a1.ww.mens.de a1  
...
```

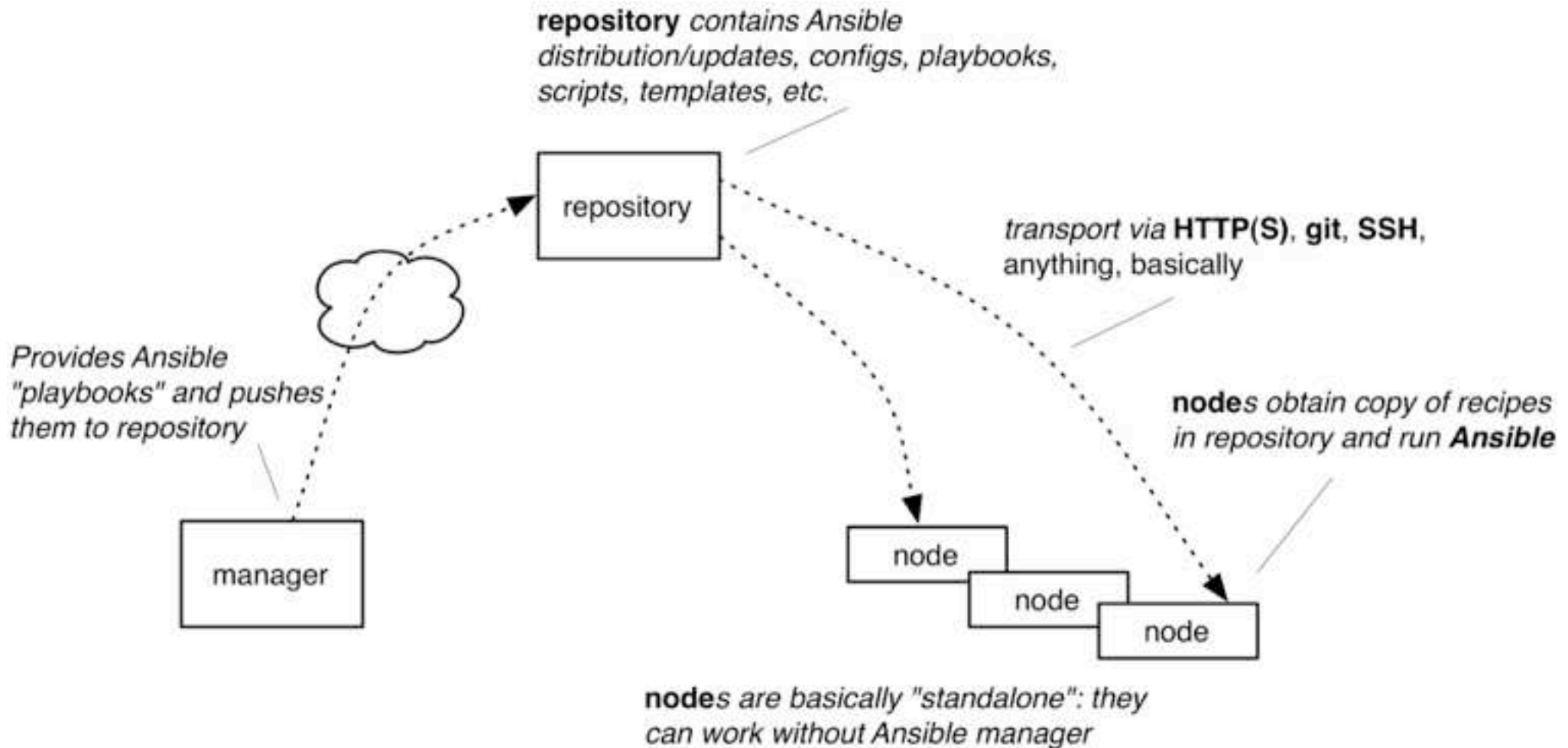
\$LOOKUP

- files
- pipe
- Redis
- DNS TXT
- ...

delegation



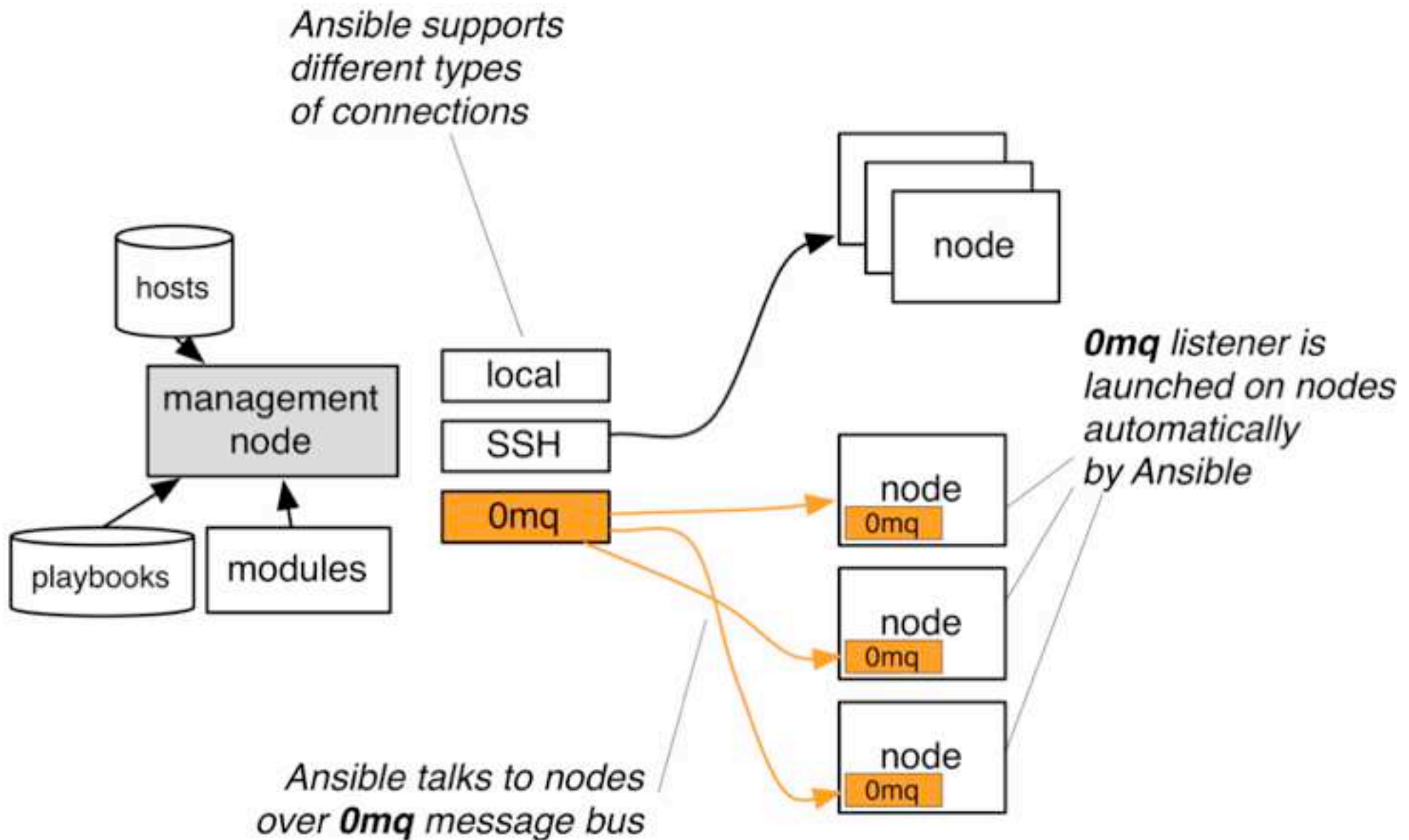
pull mode



fast, faster, **fireball**



fireball operation



ready, steady, fire!

```
# Initialize fireball
```

- hosts: nameservers
- gather_facts: false
- connection: ssh
- user: f2
- sudo: yes
- tasks:
 - action: fireball

```
# fireball now!
```

- hosts: nameservers
- connection: fireball
- tasks:
 - action: copy src=resolver.cf dest=/etc/resolv.conf
 - action: template src=bind.in dest=/etc/named.conf mode=0600

API: task execution

```
#!/usr/bin/env python
```

```
import ansible.runner  
import sys
```

```
res = ansible.runner.Runner(  
    pattern='a1*',  
    module_name='command',  
    module_args='/usr/bin/uptime'  
)  
.run()  
print res
```

```
{'dark': {}, 'contacted': {'a1.ww.mens.de': {u'changed': True, u'end': u'2012-10-22  
09:07:18.327568', u'stdout': u'09:07:18 up 100 days, 2:13, 3 users, load average:  
0.00, 0.00, 0.00', u'cmd': [u'/usr/bin/uptime'], u'rc': 0, u'start': u'2012-10-22  
09:07:18.323645', u'stderr': u'', u'delta': u'0:00:00.003923', 'invocation':  
{'module_name': u'command', 'module_args': u'/usr/bin/uptime'}}}}
```

Extensible

- Callbacks (Python)
- Action plugins (Python)
- Data sources (Python)
- Inventory sources (any language)

More **time** for **stuff**
that **matters**

ansible.cc

Join the **party!**