ansible

主讲：马永亮(马哥)
QQ群:169777636
客服QQ：2813150558, 1661815153
http://www.magedu.com
http://mageedu.blog.51cto.com

OS Provisioning

Red Hat Satellite

cobbler

Task Exec

virt-factory

func

ansible

OS Config

cfengine

puppet

chef

Deployment

capistrano/fabric
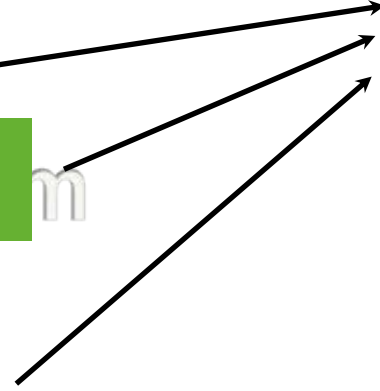
❖ **OS Provisioning**
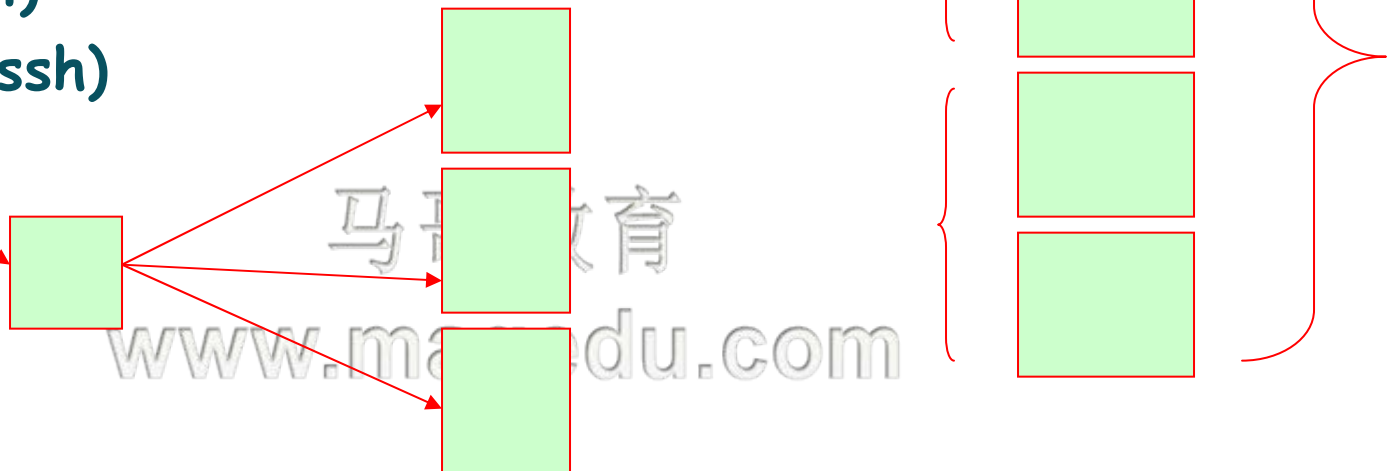 ➲ PXE, cobbler

❖ **OS config**
 ➲ cfengine, puppet, saltstack, chef

❖ **Deployment**
 ➲ func(ssl)
 ➲ fabric(ssh)
 ➲ ansible

❖ **A year old today.**

❖ **1100+ followers on github**

❖ **Top 10 New OSS Projects of 2012**

ROOKIE OF THE YEAR

BLACKDUCK

OPEN SOURCE

2012

马哥教育

www.magedu.com

❖ **Configuration (cfengine, Chef, Puppet)**

❖ **Deployment (Capistrano, Fabric)**

❖ **Ad-Hoc Tasks (Func)**

❖ **Multi-Tier Orchestration (Juju, sort of)**

- Minimal learning curve, auditability
- No bootstrapping
- No DAG ordering, Fails Fast
- No agents (other than sshd) - 0 resource consumption when not in use
- No server
- No additional PKI
- Modules in any language
- YAML, not code
- SSH by default
- Strong multi-tier solution

Focus: Orchestration
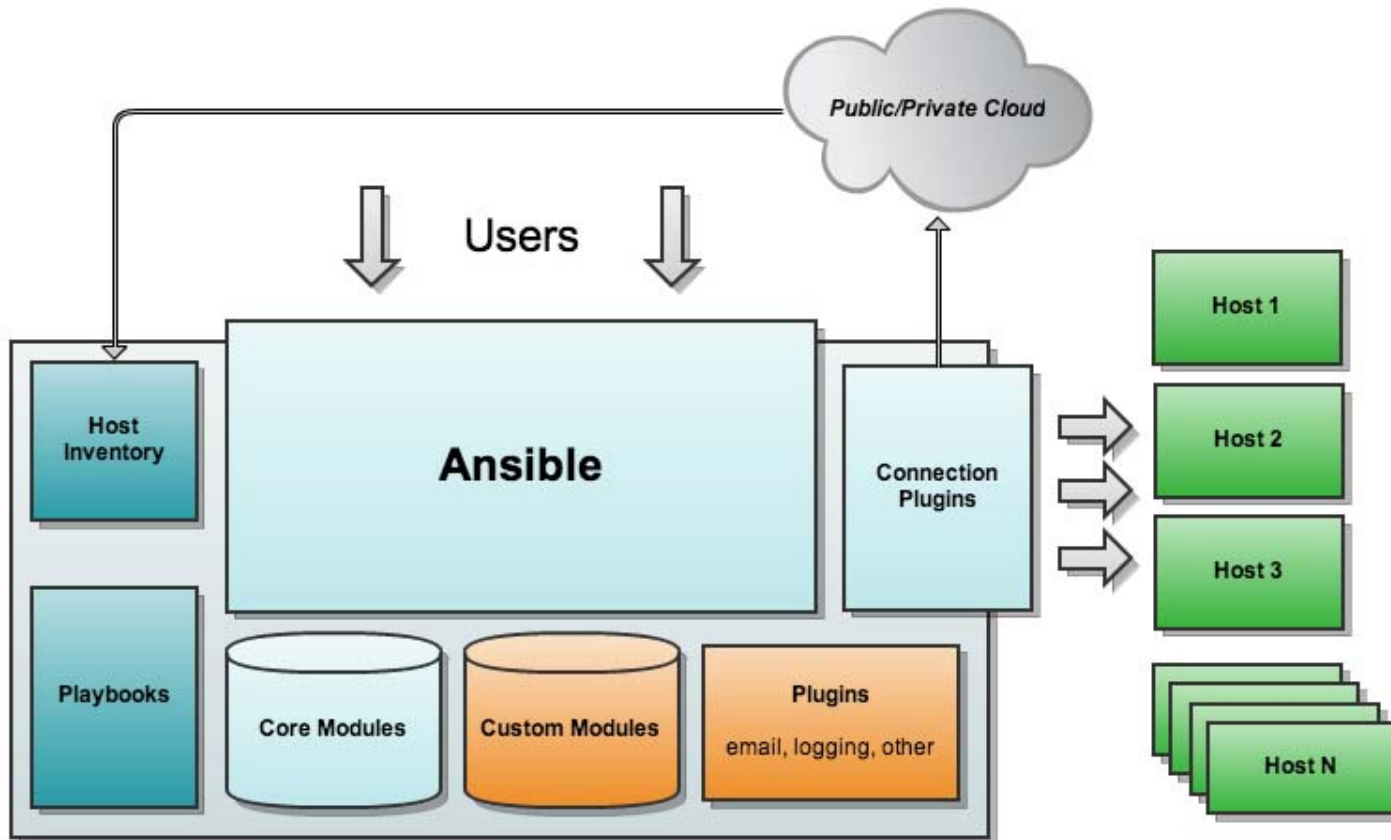
- ❖ **Rolling Updates**
- ❖ **Multi-tier: Web Tier, DB Tier, Content Tier**
- ❖ **Monitoring, Outage Windows**
- ❖ **Load Balancing, Pools**
- ❖ **Email, REST, Logging, Interaction, etc**

ansible playbooks

主讲：马永亮(马哥)
QQ群:169777636
客服QQ：2813150558，1661815153
http://www.magedu.com
http://mageedu.blog.51cto.com

❖ **Inventory**

❖ **Modules**

❖ **Ad Hoc Commands**

❖ **Playbooks**
- ➲ Tasks
- ➲ Variables
- ➲ Templates
- ➲ Handlers
- ➲ Roles

```
[webservers]
web1.magedu.com

[dbservers]
db1.magedu.com
db2.magedu.com
```

- ❖ **Cobbler**

- ❖ **OpenStack**

- ❖ **EC2 or Eucalyptus Cloud**

- ❖ **Etc**

❖ **ansible dbservers -m ping**

❖ **ansible webservers -m service -a "name=httpd state=restarted"**

- ❖ **can be written in any language as long as they output JSON**
- ❖ **take parameters and conditions to define desired state**
- ❖ **handles processing of system resources, services, packages, files, etc. in idempotent fashion**
- ❖ **"seek to avoid changes to the system unless a change needs to be made"**
- ❖ **ansible comes preloaded with a plethora of modules**
- ❖ **tons of community pull requests**

❖ **run a single, one-off command**

❖ **run on a full or partial inventory**

❖ **run on a single host**

❖ **no need to save for later**

```
$ ansible webservers –m command –a "dpkg-query –W openssl" –u joe –k
SSH password:
foo.example.com | success | rc=0 >>
openssl 1.0.1e-2+deb7u10

bar.example.com | success | rc=0 >>
openssl 1.0.1e-2+deb7u10
```

Playbooks

- ❖ **More powerful configuration management**
- ❖ **Kept in source control, developed, validated**
- ❖ **Declare configurations of more complex mutli-system enviornments**
- ❖ **Arrange and run tasks synchronously or asynchronously**

❖ **Contain one or more plays**

❖ **Written in YAML**

➲ **declarative config**

➲ **not code**

❖ **Executed in the order it is written (aka Imperative)**

```
---

- name: deploy web server
  user: foouser
  sudo: True
  hosts: all

  tasks:
  - name: install apache
    apt: pkg=apache2-mpm-prefork state=latest
```

```
---

- name: deploy web server
  user: foouser
  sudo: True
  hosts: all

  tasks:
  - name: install apache
    apt: pkg=apache2-mpm-prefork state=latest
```

```
---

- name: deploy web server
  user: foouser
  sudo: True
  hosts: all

  tasks:
  - name: install apache
    apt: pkg=apache2-mpm-prefork state=latest
```

← **Inventory**

```
---

- name: deploy web server
  user: foouser
  sudo: True
  hosts: all

  tasks:
  - name: install apache
    apt: pkg=apache2-mpm-prefork state=latest
```

```
---

- name: deploy web server
  user: foouser
  sudo: True
  hosts: all

  tasks:
  - name: install apache
    apt: pkg=apache2-mpm-prefork state=latest
```

Documentation

Arguments

Module

Output

```
$ ansible-playbook -i ~/etc/hosts main.yml

PLAY [deploy web server] **********************************************

GATHERING FACTS ******************************************************
ok: [ss-dfw-00]                    Desired state

TASK: [install apache] ***********************************************
changed: [ss-dfw-00]

PLAY RECAP ***********************************************************
ss-dfw-00                 : ok=2   changed=1   unreachable=0    failed=0


foouser@ss-dfw-00:~$ sudo netstat -putan | grep 80
tcp6       0      0 :::80                   :::*                    LISTEN
11306/apache2
```

Output

```
$ ansible-playbook -i ~/etc/hosts main.yml

PLAY [deploy web server] ********************************************************

GATHERING FACTS ****************************************************************
ok: [ss-dfw-00]


TASK: [install apache] *********************************************************
changed: [ss-dfw-00]


PLAY RECAP *********************************************************************
ss-dfw-00                    : ok=2    changed=1    unreachable=0    failed=0



foouser@ss-dfw-00:~$ sudo netstat -putan | grep 80
tcp6       0      0 :::80                    :::*                    LISTEN
11306/apache2
```

NOT Desired state

www.magedu.com

Idempotency

1 * N   0 + N

Output

```
$ ansible-playbook -i ~/etc/hosts main.yml

PLAY [deploy web server] **********************************************

GATHERING FACTS ******************************************************
ok: [ss-dfw-00]

TASK: [install apache] ***********************************************
ok: [ss-dfw-00]

PLAY RECAP ***********************************************************
ss-dfw-00                  : ok=2    changed=0    unreachable=0    failed=0
```

Idempotency

```
---
·
·
·
  tasks:
  - name: install apache on Debian based distros
    apt: pkg=apache2-mpm-prefork state=latest
    when: ansible_os_family=="Debian"

  - name: install apache on Red-Hat based distros
    yum: pkg=httpd state=latest
    when: ansible_os_family=="RedHat"
```

```
---
·
·
·
  tasks:
  - name: install apache on Debian based distros
    apt: pkg=apache2-mpm-prefork state=latest
    when: ansible_os_family=="Debian"


  - name: install apache on Red-Hat based distros
    yum: pkg=httpd state=latest
    when: ansible_os_family=="RedHat"
```

```
---
-
-
-
  tasks:
  - include: apache_debian.yml
    when: ansible_os_family=="Debian"

  - include: apache_redhat.yml
    when: ansible_os_family=="RedHat"
```

```
---

·

·

·
  tasks:
  - include: apache_debian.yml
    when: ansible_os_family=="Debian"


  - include: apache_redhat.yml
    when: ansible_os_family=="RedHat"
```

```
---
# apache_debian.yml

tasks:
- name: install apache on Debian based distros
  apt: pkg=apache2-mpm-prefork state=latest
```

```
---
# apache_redhat.yml

  tasks:
  - name: install apache on Red-Hat based distros
    yum: pkg=httpd state=latest
```

Let's deploy   LAMP with Ansible!

webservers   **dbservers**

❖ **Inventory file**
- ➲ [webservers]
- ➲ web0
- ➲ web1
- ➲ [dbservers]
- ➲ db0

马哥教育
www.magedu.com

```
---
# This playbook deploys the whole application stack in this
site.

- name: apply common configuration to all nodes
  hosts: all
  user: root

  roles:
    - common

- name: configure and deploy the webservers and application
code
  hosts: webservers
  user: root

  roles:
    - web

- name: deploy MySQL and configure the databases
  hosts: dbservers
  user: root

  roles:
    - db
```

lamp_simple

roles

common

db

web

```
---
# This playbook deploys the whole application stack in this
site.

- name: apply common configuration to all nodes
  hosts: all
  user: root

  roles:
    - common

- name: configure and deploy the webservers and application
code
  hosts: webservers
  user: root

  roles:
    - web

- name: deploy MySQL and configure the databases
  hosts: dbservers
  user: root

  roles:
    - db
```

lamp_simple

roles

common

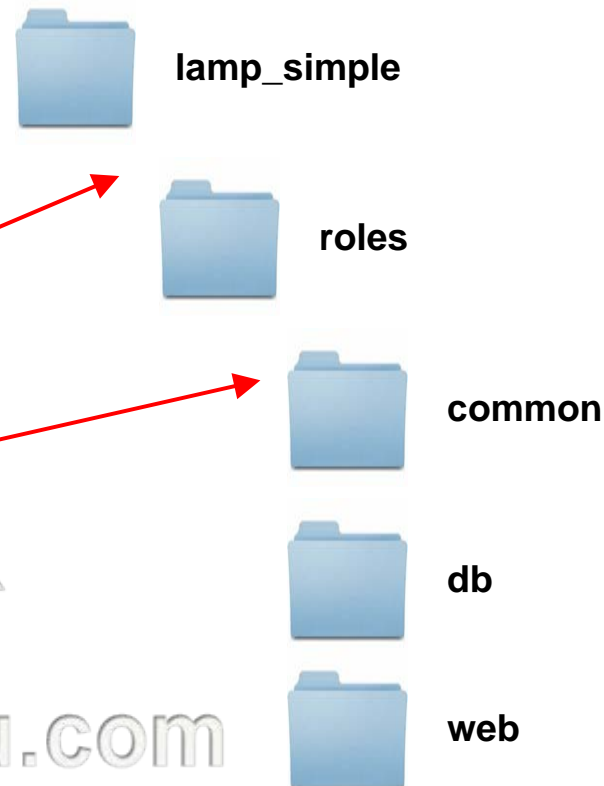db

web

```
---
# This playbook deploys the whole application stack in this
site.

- name: apply common configuration to all nodes
  hosts: all
  user: root

  roles:
    - common

- name: configure and deploy the webservers and application
code
  hosts: webservers
  user: root

  roles:
    - web

- name: deploy MySQL and configure the databases
  hosts: dbservers
  user: root

  roles:
    - db
```

lamp_simple

roles

common

db

web

Deploy LAMP

common

tasks

db

tasks

web

tasks

playbooks

Deploy LAMP

common

tasks

```
---
# This playbook contains common plays that will be run on all nodes.

- name: Install ntp
  yum: name=ntp state=present
  tags: ntp

- name: Configure ntp file
  template: src=ntp.conf.j2 dest=/etc/ntp.conf
  tags: ntp
  notify: restart ntp
…
```

马哥教育

www.magedu.com

db

tasks

```
---
# This playbook will install mysql
# and create db user and give permissions.

- name: Install Mysql package
  yum: name={{ item }} state=installed
  with_items:
   - mysql-server
   - MySQL-python
   - libselinux-python
   - libsemanage-python
…
```

web

tasks

```
---
# These tasks install http and the php modules.

- name: Install http and php etc
  yum: name={{ item }} state=present
  with_items:
   - httpd
   - php
   - php-mysql
   - …

- name: insert iptables rule for httpd
  lineinfile: dest=/etc/sysconfig/iptables create=yes state=present
regexp="{{ httpd_port }}" insertafter="^:OUTPUT "
              line="-A INPUT -p tcp  --dport {{ httpd_port }} -j
ACCEPT"
  notify: restart iptables
  …
```

```
site.yml                 # master playbook
webservers.yml           # playbook for webserver tier
dbservers.yml            # playbook for dbserver tier

roles/
    common/              # this hierarchy represents a "role"
        tasks/           #
            main.yml     #  <-- tasks file can include smaller files if warranted
        handlers/        #
            main.yml     #  <-- handlers file
        templates/       #  <-- files for use with the template resource
            ntp.conf.j2  #  <------- templates end in .j2
        files/           #
            bar.txt      #  <-- files for use with the copy resource
            foo.sh       #  <-- script files for use with the script resource
        vars/            #
            main.yml     #  <-- variables associated with this role

    webtier/             # same kind of structure as "common" was above, done for the webtier role
    monitoring/          # ""
    fooapp/              # ""
```
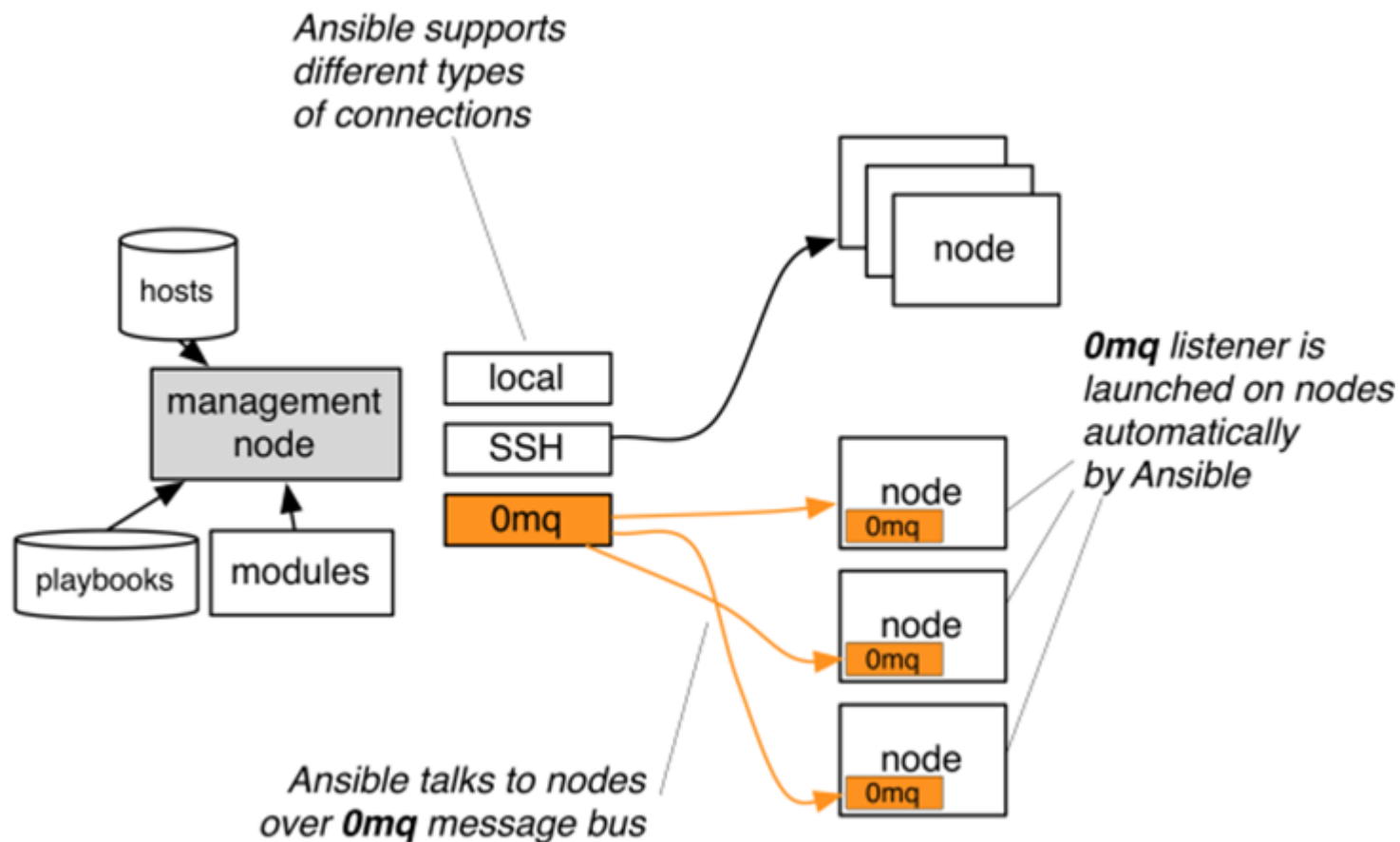
Ansible supports different types of connections

hosts

management node

playbooks   modules

local

SSH

0mq

node

0mq listener is launched on nodes automatically by Ansible

node
0mq

node
0mq

node
0mq

Ansible talks to nodes over **0mq** message bus

repository contains Ansible distribution/updates, configs, playbooks, scripts, templates, etc.

repository

transport via **HTTP(S)**, **git**, **SSH**, anything, basically

Provides Ansible "playbooks" and pushes them to repository

nodes obtain copy of recipes in repository and run **Ansible**

manager

node

node

node

nodes are basically "standalone": they can work without Ansible manager

❖ 博客：**http://mageedu.blog.51cto.com**
❖ 主页：**http://www.magedu.com**
❖ **QQ：1661815153，113228115**
❖ **QQ群：203585050，279599283**

马哥教育

Thank You!