

How Puppet fits into your existing architecture

2011-06-28

San Francisco, CA

SFO DevOps

Garrett Honeycutt

Professional Services Consultant

garrett@puppetlabs.com

<http://linkedin.com/in/garretthoneycutt>



We are hiring

- Professional Services
- QA Engineer
- Core Developer

PuppetConf

9/19 – 23 in PDX

<http://puppetconf.com>

- Facebook
- Zynga
- DTO Solutions
- Google
- Eucalyptus

Provisioning

Goals:

- One button deploys
- Quick and easy re-provisioning
- No upgrades – just build new systems
 - solves issue of intermediate states

Provisioning

Start from a known base!

- Use the same base install for all nodes
- Smallest footprint of what it means to be a node on your network
- Allows for easier reddeploys on other systems (VMWare, bare metal, EC2, Rackspace, Vagrant, etc)

Provisioning

PXE

- Provisions VM's and Physical systems the same way

Cloudy API's

- May not be an option if you have physical hardware

Provisioning

Cobbler

- My favorite provisioning system for PXE
- Handles tftp/dhcp/dns/repo's
- Namely for RedHat-ish systems, also supports Solaris, Debian, and images (ie: memtest, windows, firmware upgrades, etc)
- <http://github.com/ghoneycutt/puppet-cobbler>

Provisioning

Puppet CloudPack

- Provision EC2 (others forthcoming) systems

Provisioning

Chicken and Egg with Software Repo's

- `--tags repo`
- Preferred over run stages for simplicity and portability in modules

Provisioning

Certificate management

- autosigning is your friend
- can also pre-generate certs
- `gencert.php` – uses reverse DNS

External Node Classifier

Puppet Dashboard

- source of truth for list of nodes
- Add/Remove hosts through API – ties into provisioning

Package Management

Run your own Software Repositories

- You control when package versions change
- Packages are not mysteriously missing
- Much faster provisioning

Package Management

Version control your repositories

- Does not mean you need to use a VCS
- `/data/repos/CentOS_5.5_Base` symlink to `/data/repos/CentOS_5.5_Base-2011062700`
- Use `hardlink(1)` to deal with duplicate files

Package Management

package {}

- ensure => present or absent
- no version #'s

Package Management

`no package { 'foo': ensure => latest }`

- not so homogeneous clusters while systems converge
- ideally upgrades happen with rebuilds
- upgrades are triggered en masse during a maintenance window

Account Management

Use a directory service

- LDAP
- Active Directory

Account Management

Role based access control

- Groups get access, NOT users
- Who is in what team can be delegated to HR/management

Account Management

`/etc/security/access.conf`

- controls groups that may access the system
- <http://github.com/ghoneycutt/puppet-pam>

Account Management

List users as virtual resources sorted by UID and realize as necessary

```
@common::mkuser { 'apachehup':  
  
    uid          => '32001',  
  
    gid          => '32001',  
  
    home         => '/home/apachehup',  
  
    managehome  => true,  
  
    comment     => 'Apache Restart User',  
  
    dotssh      => true,  
  
}
```

<http://github.com/ghoneycutt/puppet-generic>

Data storage

Data?

- information that your node serves or creates

Data storage

Keep data stored off node

- SAN / NAS / Cloudy store
- rebuilt machines reconnect to your data

Disposable Architecture

<http://www.linkedin.com/in/ericheydrick>

I just lost a system.. big deal.

Failure is going to happen, let it.

Disposable Architecture

Develop other metrics to determine system health

- not how many systems are alive
- response times
- % of anticipated capacity

Auto-scaling

Tying it together

- (de)provision based on metrics
 - capacity, response, etc

How Puppet fits into your existing architecture

2011-06-28

San Francisco, CA

SFO DevOps

Garrett Honeycutt

Professional Services Consultant

garrett@puppetlabs.com

<http://linkedin.com/in/garretthoneycutt>

