Configuration Management

It's all about power and control … and scale and accountability.

TGIF: January 30, 2009

What is configuration management?

- Revision control – cvs, rcs, subversion
- Build automation – jumpstart, kickstart
- File management and installation – cpp, rdist, radmind, tripwire
- Package management and installation – pkgadd, rpm, yum, up2date, opium
- Metrics - cricket
- Bug/problem tracking – nagios, symon

CM Tools:

- Wikipedia has 13 open source options
  - http://en.wikipedia.org/wiki/Comparison_of_open_source_configuration_management_software
  - Most are linux, solaris, and aix compatible
- Differences
  - Scope/scale
  - Client – server model
  - State/service based
  - Language
  - Monitoring capability

CUIT and CFengine

- Existing tools insufficient: cpp, opium
  - Manual processes
  - Lack of scale - /opt and certain files only
  - Lack of consistency (e.g. perms on /etc/httpd on mainwebservers/ sundial)
  - Many root accounts
  - Minimal oversight of changes - rcs
- AcIS Linux Project (sandbox)
  - Erik Dykema – 2004
  - No process for CM determination
- June 2006 (production)
  - Cyrus – rapid deployment of 30+ servers
  - Erik's departure
- ATG: radmind

What is Cfengine?

- Cfengine is a policy-based configuration management system written by Mark Burgess at Oslo University College. **Its primary function is to provide automated configuration and maintenance of computers, from a policy specification.**
- One of the main innovations of cfengine is the idea that changes in computer configuration should be carried out in a convergent manner. This means that each change operation made by the agent should have the character of a fixed point. **Rather than describing the steps needed to make a change, cfengine describes the final state in which one wants to end up.** The agent then ensures that the necessary steps are taken to end up in this "policy compliant state". Thus, cfengine can be run again and again, whatever the initial state of a system, and it will end up with a predictable result.
- Cfengine is used in both large and small companies, as well as in many universities and governmental institutions. Sites as large as 30,000 machines are reported, while sites of several thousand hosts running under cfengine are common.

http://en.wikipedia.org/wiki/Cfengine

Client – Server Model

- Masters
  - Master repository cfmaster:/etc/cfengine
  - Copy of /src/systemfiles/cfengine
- Clients
  - Connect to masters on schedule
  - Copy from master repository to local:/var/cfengine/inputs
  - Execute configs

Configuration Files

- *.conf for internal cfengine processes (main.conf, groups.conf)
  - import
  - schedules
  - trust relations
- *.cf for changes to be made on hosts
  - main.conf controls imports and order of actions
    - Most actionsequences are in main.conf for efficiency
- Classes - # cfagent -pv
  - internally defined “hard” classes : solaris, linux, architecture
  - clusters from hostdata.pl are generated into groups.conf
    - Make sure that both cluster and cf_cluster exist
  - defined within run
    - Defined if designated package installed or file/directory exists

actionsequences

$ grep actionseq main.conf

actionsequence = (disable directories processes copy shellcommands links editfiles files tidy packages)

- disable – removes/renames/deletes files
- directories – create directory with perms/owners
- processes – check for processes, possibly kill/restart
- copy – copy files usually from local repo to prod location
- shellcommands – quoted commands
- links – creates symlinks
- editfiles – edit production files
- files – checks and resets perms/ownership of files/directories
- tidy – clean up
- packages – install RH or Solaris packages – not homegrown

Building and maintaining a host

1. Register new host with mac address/configure solaris jumpstart
2. Using console, initiate kickstart (rhel) or jumpstart (solaris)
   - Installs OS
   - Installs base set of applications
   - Copies some default config files
   - Patches (solaris)
3. Log in and run cfagent
   - Cluster specific configuration
   - Package installation (Solaris and RH)
4. Configure kerberos and ssh keys
5. Run opium (Solaris only)
6. Install cpp files

Machine installed – configuration complete

- Run cfagent according to schedule (twice hourly)
- Cfagent talks to RedHat and gets updates most weekdays
- Opium run on Thursdays

Policy issues:

- What is a good strategy for editing CFEngine managed files?
  - It seems risky to edit files in the source tree, and have them get pushed immediately out to hosts. If you save a file in some invalid state, there's a chance it can get pushed out to some hosts.
- How to accomplish testing of CFEngine managed files?
  - Suppose you need to update config files for some application. How do you go about performaing a test before checking the updated files into the source tree where CFEngine will distribute them. Or is that not the best way…
- Tidying (deleting non-cfengine managed files)
  - Suggested strategy for this? Seems cleaner to have non-managed files get deleted, but I typically don't bother with that (and sometimes rely on having the ability to leave copies of files around).

More CFEngine resources

- https://www1.columbia.edu/sec/acis/sy/systems-manual/cfengine/cfengine_basics.html
- http://www.cfengine.org/
- Cfengine 2 reference: http://www.cfengine.org/docs/cfengine-Reference.html
- Cfwiki: http://www.cfwiki.org/cfwiki/index.php/Main_Page
- Sage Booklet: http://www.sage.org/pubs/16_cfengine/
- Watson-Wilson Cookbook: http://watson-wilson.ca/blog/cfcookbook.html

Future Projects

- Move cfengine configs to subversion or something like it
  - Prevent unexpected changes from propagating
  - Limit approval control to more knowledgeable admins
    - Create additional level of bureaucracy
    - Slow down change implementation
  - Requirements:
    - Unix sys admin time
    - Increased sys admin knowledge

Configuration Management (re)Evaluation

- Existing tools
  - Hostmonger
  - cpp
  - Opium
  - Cfengine 2
- Other options:
  - Cfengine3
  - Puppet
  - Bcfg2
  - Radmind
  - ??
- Trends
  - Scale
  - Fewer root accounts
  - Fewer people with access
  - Tighter management
- Challenges
  - cultural – people want root
  - sysadmin
    - lack of time
    - limited experience

Questions? Comments?

No complaints, please.