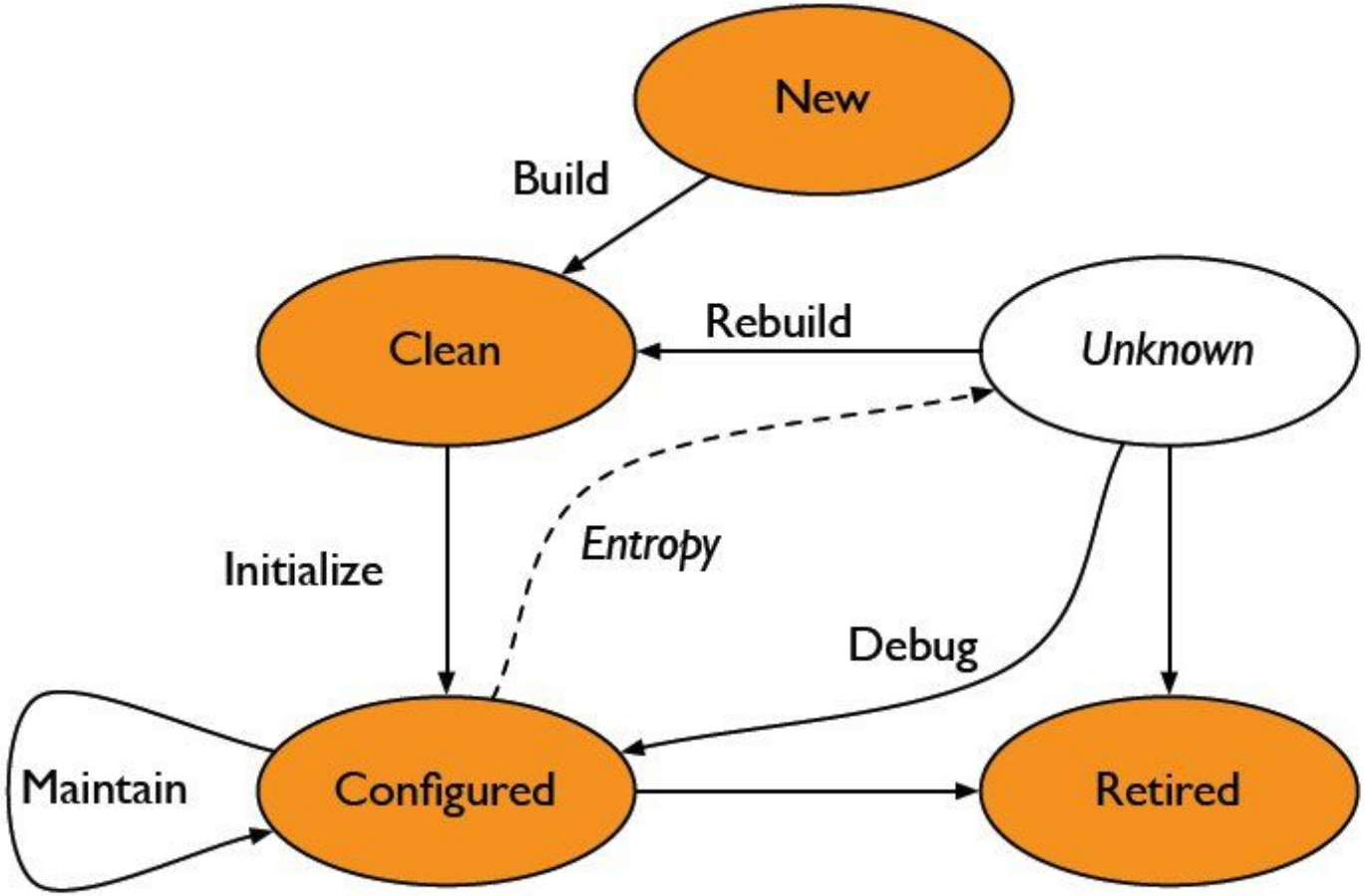
Configuration Management with Cfengine

Steven Kreuzer
NYC BSD Users Group
July 2008

Configuration Management

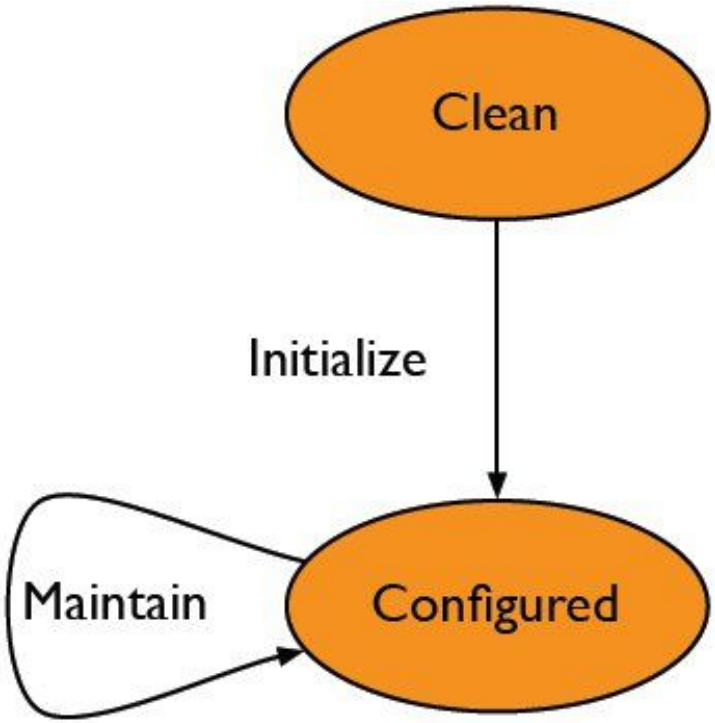
- Configuration management facilities provide efficient solutions to complex problems. For example:
  - How do I manage configuration files?
  - How do I know maintenance tasks, such as, backups are completed at the right times in the right places?
  - How do I ensure that important system files are properly protected against unauthorized access and modification?

Host life cycle



Adapted from The Practice of System and Network Administration, p. 5.

What is Cfengine?



"Cfengine … is an autonomous agent and a middle to high level policy language and agent for building expert systems to administrate and configure large computer networks."

http://www.cfengine.org/

What to do with a tool like this…

- Ensure that files altered by package managers are correctly tailored and adjusted to perform in your environment.
- Verify that processes are (or aren’t) running.
- Monitor disk usage and provide warning when file-systems are full
- Search and identify file changes to maintain system security or for locating human error.

Components

- cfagent - interprets policy and implements in a convergent manner
- cfexecd – is a scheduler and wrapper, sends you email.
- cfservd - server daemon for remote copy and execution
- cfrun - trivial helper app that polls hosts and tells them to run cfagent

Additional components

- cfenvd - state monitor, collects statistics for anomaly detection
- cfkey - generates public-private key pairs (once) on a host

Commonly Used Terms…

- Host – Server of any kind
- Classes – Group of hosts sharing a common policy (www_servers, db_servers, freebsd7_servers, openbsd41_servers)
- Policy – The description of a configuration
- Configuration – The state of files, processes, system resources on a host

Getting started

- Installing
  - On FreeBSD (and possibly OpenBSD)
    - pkg_add –r cfengine
    - cd /usr/ports/sysutils/cfengine && make install
  - From Source
    - tar zxf cfegnine-${VERSION}.tar.gz
    - cd cfengine-${VERSION}
    - ./configure
    - make install

Getting it running on one host

- Things to think about
  - Writing a policy / configuration
  - Getting trusted communication working
- Autonomy: Always have a local copy of policy to minimize dependencies
  - Each host has /var/cfegine
  - bin, inputs, outputs, state
- Ultimately let Cfengine configure itself

Testing on a single host

$ vi /var/cfengine/inputs/cfagent.conf

```
control:
        actionsequence = ( shellcommands )

shellcommands:
        "/bin/echo Hello, World!"
```

$ /usr/local/sbin/cfagent –f ./cfagent.conf
cfengine:erdinger:/bin/echo Hello: Hello, World!

Quick setup for multiple hosts

- Decide policy: cfagent.conf
- Distribute policy: cfservd.conf
- Setup clients to install themselves: update.conf
- Suppose 192.168.1.0/24 network

cfservd.conf

```
control:
    domain = ( lab.exit2shell.com )
    MaxConnections = ( 50 )
    AllowConnectionsFrom = ( 192.168.1.0/24 )
    TrustKeysFrom = ( 192.168.1.0/24 )

admit:
    /var/cfengine/inputs 192.168.*
    /var/cfengine/ppkeys/localhost.pub 192.168.*
```

cfagent.conf

```
control:
    domain = ( lab.exit2shell.com )
    schedule = ( Min10_15 Min30_35 Min50_55 )
    ChecksumUpdates = ( on )

import:
    any::
            cf.groups
            cf.site
    freebsd::
            cf.freebsd
```

update.conf

```
control:
     actionsequence  = ( copy tidy )
     domain          = ( lab.exit2shell.com )
     policyhost      = ( erdinger )
     master_cfinput  = ( /var/cfengine/inputs )
     workdir         = ( /var/cfengine )
     SplayTime = ( 10 ) # minutes
copy:
     $(master_cfinput) dest=$(workdir)/inputs
                  r=inf mode=700 type=checksum
                  include=cf.* include=*.conf
                  exclude=*.lst exclude=*.bak exclude=.* exclude=*~ exclude=#*
               server=$(policyhost)
                  trustkey=true
tidy:
     $(workdir)/outputs pattern=* age=7
```

cf.groups

groups:

web_servers = (www0 www1 www2)

db_servers = (db0 db1 db2)

cf.site (part 1)

```
control:
    actionsequence = ( files tidy editfiles )
    editfilesize = ( 0 )
    any::
            tmpdir = ( /tmp )
    freebsd|openbsd::
            shadowfile = ( /etc/master.passwd )
            shadowpermissions = ( 600 ) filegroup = ( wheel )
            crondir = ( /var/cron/tabs )
    linux::
            shadowfile = ( /etc/shadow )
            shadowpermissions = ( 400 ) filegroup = ( root )
            crondir = ( /var/spool/cron )
```

cf.site (part 2)

```
files:
   any::
         ${shadowfile}
               mode=$(shadowpermissions)
               owner=root group=$(filegroup)
               action=fixall
         /etc/passwd
               mode=644 owner=root
               group=$(filegroup) action=fixall
```

cf.site (part 3)

```
tidy:
    any::
            $(tmpdir) pattern=* age=7
                    recurse=inf rmdirs=sub
            /var/tmp pattern=* age=7
                    recurse=inf rmdirs=sub
editfiles:
    any::
    { /etc/services
            AppendIfNoSuchLine "cfengine        5308/tcp“
            AppendIfNoSuchLine "cfengine        5308/udp“
    }
```

cf.freebsd (part 1)

```
control:
    ActionSequence = ( packages editfiles)
    DefaultPkgMgr = ( freebsd)
    FreeBSDInstallCommand =
            ( "/usr/sbin/pkg_add
    ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/
    packages-7-stable/All/%s" )
    FreeBSDRemoveCommand =
    ("/usr/sbin/pkg_delete %s" )
```

cf.freebsd (part 2)

```
packages:
    freebsd.any::
                pdksh-5.2.14p2_2.tbz action=install
                sudo-1.6.9.15_1.tbz action=install
                vim-lite-7.1.293_1.tbz action=install
    freebsd.web_servers::
                apache-2.0.63.tbz action=install
                memcached-1.2.5.tbz action=install
                p5-DBD-Pg-2.6.4.tbz action=install
                mod_perl2-2.0.3_3,3.tbz action=install
                varnish-1.1.2.tbz action=install
    freebsd.db_servers::
                postgresql-client-8.3.1.tbz action=install
                postgresql-server-8.3.1.tbz action=install
```

cf.freebsd (part 3)

```
editfiles:
    freebsd.any::
                { /etc/rc.conf
                        Backup "false"
                        AppendIfNoSuchLine "sshd_enable=\"YES\""“
                }

    freebsd.web_servers::
                { /etc/rc.conf
                        Backup "false“
                        AppendIfNoSuchLine "apache2_enable=\"YES\“”
                }
```

Special Thanks

- Mark Burgess
  - Wrote Cfengine
  - Borrowed heavily from his talks
    - http://www.cs.virginia.edu/sigbed/archives/2006-04/Marc.pdf
    - http://www.cfengine.org/AutonomicCfengine.pdf
- Jeremy Mates
  - Borrowed some slides from his talk
    - http://sial.org/talks/kickstart-cfengine/

Questions