



Configuration Management con Cfengine

Riccardo Bloise
Linux Day - Roma
24 Ottobre 2009

Per iniziare supponiamo di...

- Avere una larga rete di macchine eterogenee da amministrare.
- Voler implementare una o più configurazioni desiderate per tipologia di macchine e di voler installare software diverso in gruppi di macchine diverse.
- Avere la necessità di un tool che ci aiuti nel configurare e mantenere i server.

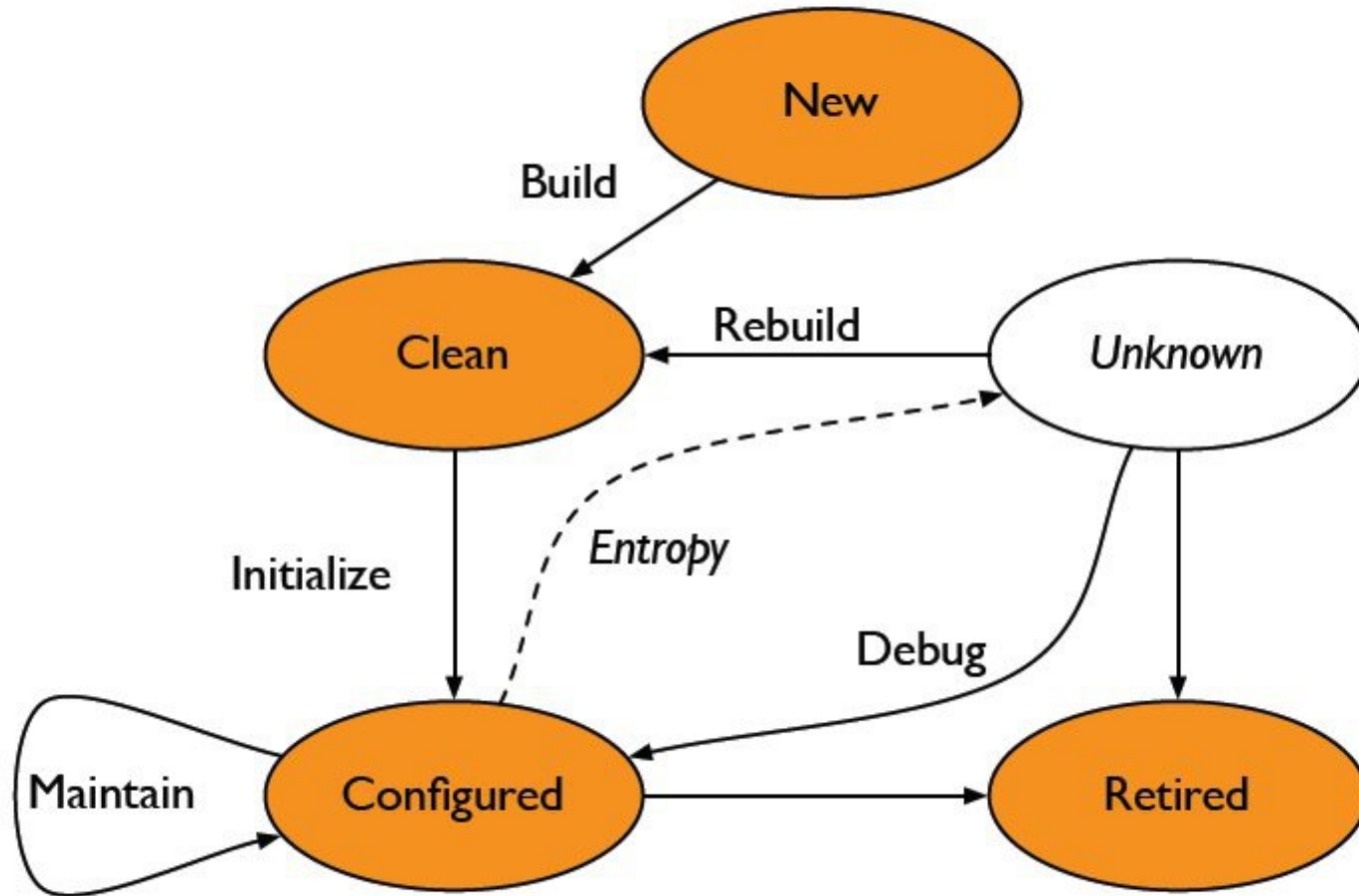
Configuration Management

- Un Configuration Management ha lo scopo di fornire soluzioni efficienti a problemi complessi.

Ad esempio:

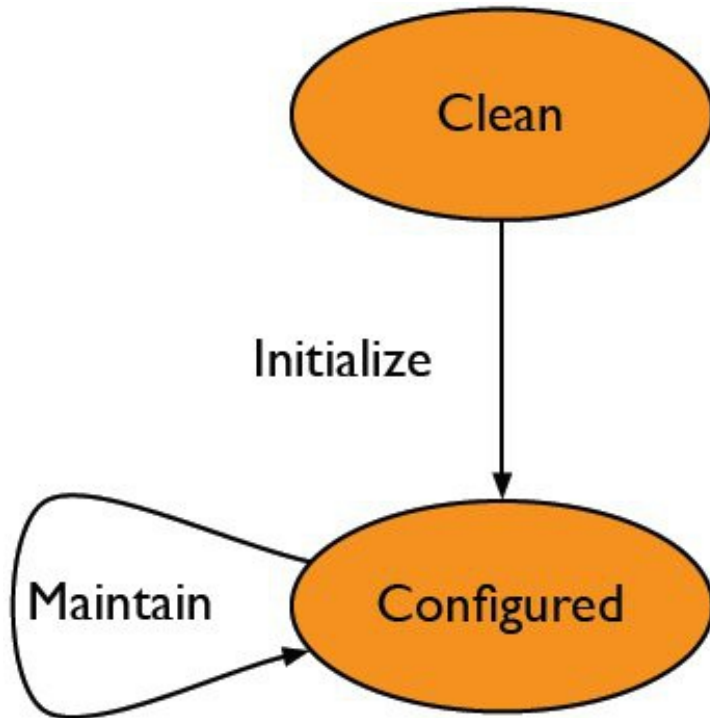
- Come faccio a gestire file di configurazione distribuiti?
- Come faccio a controllare le attività di manutenzione, come ad esempio backup e logging e come posso sincronizzare tali attività?
- Come faccio a garantire che i file di sistema importanti siano adeguatamente protetti contro gli accessi non autorizzati o modifiche accidentali ?

Ciclo di vita di un Host



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

Cos'è 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/>

Cosa posso fare con Cfengine...

- Configurare una larga rete di macchine direttamente da un server centrale.
- Installare automaticamente programmi tramite i corrispettivi gestori dei pacchetti e assicurare che siano della versione corretta per il sistema operativo in uso.
- Verificare che i processi previsti sono (o non sono) in esecuzione.
- Controllare l'utilizzo del disco e fornire un avviso quando i file-systems sono pieni; più in generale controllare varie anomalie con un apposito demone.
- Ricercare ed identificare le modifiche ai file per mantenere la sicurezza del sistema o per localizzare un errore umano.

Composizione di Cfengine

- 3 demoni.
- 6 file eseguibili.
- Vasta documentazione: pagine man, info ed ampia guida di riferimento.
- File di configurazione d'esempio.

Termini comunemente usati...

- **Host** – indica un server generico nella rete.
- **Classi** – gruppi di host che condividono una politica comune (`workernode_servers`, `db_servers`, `redhat_servers`, `gentoo_servers`).
- **Politica** – la descrizione di una configurazione.
- **Configurazione** – lo stato corrente dei file, dei processi e delle risorse di sistema su un host.

Componenti Essenziali

- **cfagent** – è l'interprete e l'agente di *Cfengine* eseguito in ogni macchina, interpreta la politica e la attua.
- **cfexecd** – è il demone che si occupa di schedulare l'esecuzione automatica di **cfagent** nei nodi, una sorta di wrapper che eventualmente può inviare email all'amministratore della rete.
- **cfserverd** – è il demone del server per l'esecuzione remota da parte dei client e per la copia dei file di configurazione.

Componenti aggiuntive

- **cfkey** – genera una coppia alla volta di chiavi pubbliche-private per host.
- **cfenvd** – si occupa di monitorare lo stato generale e di raccogliere le statistiche per il rilevamento di anomalie.
- **cfrun** – applicativo che consente l'esecuzione remota selettiva per gruppi di host opportunamente definiti.

Per iniziare

■ Installazione

□ Su Scientific Linux (e Red Hat Linux):

- `yum install cfengine`

□ Dai sorgenti:

- `tar zxf cfengine- $\{VERSION\}$.tar.gz`

- `cd cfengine- $\{VERSION\}$`

- `./configure`

- `make && make install`

Primi passi da eseguire

- Cose da fare:
 - Scrivere una politica di configurazione.
 - Ottenere la conferma di sicurezza scambiando le chiavi e sincronizzando la parte server con i client (in caso di singolo host è tutto in locale).
- Autonomia: avere sempre una copia della politica per ridurre al minimo i rischi di malfunzionamenti.
 - In ogni host dopo l'installazione si trova `/var/cfengine/inputs`
 - In ogni host abbiamo anche `/var/cfengine/bin`, `../outputs`, `../state`
- Infine possiamo iniziare ad usare Cfengine

Test su un singolo host

```
# nano /var/cfengine/inputs/cfagent.conf
```

```
control:  
    actionsequence = ( shellcommands )  
  
shellcommands:  
    "/bin/echo Hello World!"
```

```
# /usr/sbin/cfagent -f /var/cfengine/inputs/cfagent.conf
```

```
cfengine:excalibur:/bin/echo Hello: Hello World!
```

Installazione per più host

- Decide la politica: `cfagent.conf`
- Distribuisce la politica: `cfserverd.conf`
- Aggiornamento dei nodi: `update.conf`
- Supponiamo di avere una rete:
192.168.1.0/24

File: cfservd.conf

- Il file di configurazione del server, definisce chi può avere accesso e a cosa:

control:

```
domain = ( lab.lugroma.org )  
AllowUsers = ( root )  
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.*
```

File: update.conf

- Si occupa di recuperare gli ultimi file di configurazione dal server, tutti gli host della rete hanno bisogno di questo file per la sincronizzazione di *Cfengine*:

control:

```
actionsequence = ( copy tidy )
domain         = ( lab.lugroma.org )
policyhost     = ( server_master )
master_cfinput = ( /var/cfengine/inputs )
SplayTime     = ( 10 ) # in minuti
```

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)
```

tidy:

```
$(workdir)/outputs pattern=* age=7
```

File: cfagent.conf (1° parte)

- E' l'ultimo file di configurazione che consente d'impostare una configurazione avanzata ed una politica di gestione della rete. Esso è inizialmente creato sul solo server MASTER e verrà poi distribuito ed aggiornato su tutta la rete:

control:

```
actionsequence = ( files directories tidy processes packages shellcommands )
domain          = ( roma1.infn.it )
timezone        = ( CET EST EDT )
access          = ( root )
master_cfinput  = ( /var/local/cfengine/files )
```

ScientificLinux|redhat::

```
DefaultPkgMgr   = ( rpm )
RPMInstallCommand = ( "/usr/bin/yum install -y %s" )
```

Gentoo::

```
PortageInstallCommand = ( "/usr/bin/emerge --nocolor %s" )
```

groups:

```
ScientificLinux = ( host1 host2 host3 )
Gentoo          = ( host4 host5 host6 )
```

File: cfagent.conf (2° parte)

files:

```
/etc/passwd mode=644 owner=root action=fixall  
/etc/shadow mode=600 owner=root action=fixall  
/etc/group mode=644 owner=root action=fixall
```

directories:

```
/tmp mode=1777 owner=root group=root
```

tidy:

```
/tmp r=inf age=7 rmdirs=sub
```

processes:

```
"cfenvd" restart "/etc/init.d/cfenvd start >/dev/null"  
"cfexecd" restart "/etc/init.d/cfexecd start >/dev/null"  
"cfservd" restart "/etc/init.d/cfservd start >/dev/null"
```

packages:

```
ScientificLinux|redhat::  
                figlet action=install  
Gentoo::  
                figlet action=intall
```

shellcommands:

```
"/bin/touch /var/cfengine/inputs/ciao_mondo" umask=022
```

Tipica sequenza di eventi

- **cfexecd** in esecuzione come demone, fa eseguire **cfagent** (~ una volta all'ora).
- **cfagent** legge *update.conf* e contatta **cfservd** in locale se è in esecuzione oppure contatta 'policyhost' per recuperare le ultime configurazioni.
- **cfagent** legge *cfagent.conf*, determinando le classi di macchine definite e comincia a svolgere le varie operazioni previste.

Risorse in rete

- <http://www.cfengine.org>
- <http://www.cfwiki.org>
- <http://www.gnu.org/software/cfengine>
- Mailing list: help-cfengine@gnu.org

Ringraziamenti

■ Mark Burgess

- Ideatore e programmatore di Cfengine
- Per aver sfruttato l'abbondante documentazione pubblicata:
 - <http://www.cs.virginia.edu/sigbed/archives/2006-04/Marc.pdf>
 - <http://www.cfengine.org/AutonomicCfengine.pdf>

■ Jeremy Mates

- Per l'ottima documentazione messa a disposizione:
 - <http://sial.org/talks/kickstart-cfengine/>



Domande