# Gestión de Configuraciones con *cfengine*

Carlos Vicente
Servicios de Redes
Universidad de Oregón

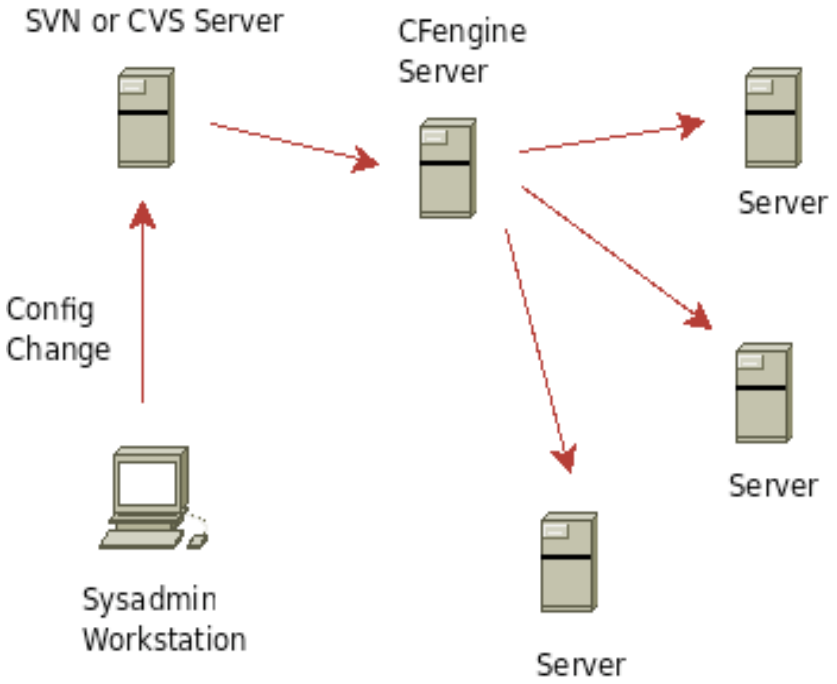
# Necesidad

- El estado de configuración de los sistemas tiende a la divergencia
- Los humanos tenemos no somos consistentes
  - Eliminación de las tareas repetitivas
- Acelerar el tiempo de puesta a punto de un sistema nuevo
- A medida que la red crece y el número de sistemas y servicios aumenta, la administración manual se vuelve una pesadilla

# Qué es *cfengine*

- cf = configuración, engine = motor
- Consiste en:
  - Un lenguaje de muy alto nivel para definir políticas de configuración
  - Varias herramientas:
    - cfagent – Un agente que ejecuta las políticas
    - cfexecd – Un *daemon* que gestiona el agente
    - cfservd – Un servidor de archivos y de acciones remotas
    - cfrun – Ejecuta agentes externos
- Es una *medicina contra la repetición*

# Cfengine + Control de Versiones



# Distribución del Paquete

```
bin/        Binarios
inputs/     Configuración
modules/    Extensiones
outputs/    Mensajes
ppkeys/     Claves privadas y públicas
state/      Información del entorno
```

# Sintaxis

- La configuración consta de: *acciones*, *condiciones* y *declaraciones:*

```
# comentarios
acción:
    clase1::
        declaración
        declaración
    clase2|clase3::
        declaración
```

# Condiciones

- Las condiciones en cfengine se componen de clases (classes) o grupos (groups), con operadores lógicos:
  - '.' o '&' equivale al 'AND' lógico
  - '|' equivale al 'OR' lógico
  - '!' equivale al 'NOT' lógico
- Los paréntesis sirven para modificar la precedencia
  - !Hr00.(parche_disponible|Tuesday)

  (Verdadero si NO es medianoche Y hay un parche o es martes)

# Clases predeterminadas

- Por defecto, cfengine define una serie de clases asociadas a la máquina donde se ejecuta
  - Verificar con `cfagent -pv`

```
10_1_1 10_1_1_1 ipv4_10 ipv4_10_1 ipv4_10_1_1
ipv4_10_1_1_1
192_168_1 192_168_1_1 ipv4_192 ipv4_192_168
ipv4_192_168_1 ipv4_192_168_1_1
net_iface_eth0 net_iface_eth1 net_iface_lo
host1 host1_example_com
```

# Clases definidas por el administrador

- Definir bajo la acción *classes*

```
classes:
  any::
    servidor_mx       = ( servidor1 servidor2 servidor3 )
    servidor_mail     = ( servidor4 servidor5 )
    cliente_cfengine = ( any )
```

# Acción *control*

- Especifica qué debe hacer el agente y cómo
  - Si no existe esta acción, nada ocurre!
- Define ciertas variables, configura valores por defecto y define el orden en que se han de ejecutar las acciones definidas en otras partes de la configuración
  - La variable de control más importante es *actionsequence*

# Acción *control*

```
control:

   site       = ( walc )
   domain     = ( localdomain )
   sysadm     = ( walcadmin@localdomain )
   smtpserver = ( mail.localdomain )

  SplayTime = ( 1 )
  schedule = ( Min00_05 Min15_20 Min30_35 Min45_50 )


   actionsequence =
      (
      links.some
      mountall
      links.others
      files
      )
```

# Actionsequence

- Acciones permitidas en esta variable:

```
actionsequence =
   (
    mountall                # mount filesystems in fstab
    mountinfo               # scan mounted filesystems
    checktimezone           # check timezone
    netconfig               # check net interface config
    resolve                 # check resolver setup
    unmount                 # unmount any filesystems
    packages                # check for required packages
    shellcommands           # execute shell commands
    editfiles               # edit files
    addmounts               # add new filesystems to system
    directories             # make any directories
    links                   # check and maintain links (single and child)
    mailcheck               # check mailserver
    mountall                # (again)
    required                # check required filesystems
    tidy                    # tidy files
    disable                 # disable files
    files                   # check file permissions
    copy                    # make a copy/image of a master file
    processes               # signal / check processes
    module:name             # execute a user-defined module
)
```

# Acciones

- *copy*:
  - Copia archivos de un directorio a otro, o del servidor al cliente. Manipula permisos, chequea integridad, etc.
  - El servidor debe estar ejecutando 'cfservd'
  - El cliente debe ser admitido en la configuración del servidor
  - Cliente y servidor deben tener la misma hora
  - Cliente y servidor deben tener la clave pública del otro

```
copy:

    any::

    $(cfmaster) dest=$(cfworkdir)
       r=inf mode=o-rw type=checksum server=$(policyhost)
       trustkey=true exclude=*~ exclude=#*
```

# Acciones

- *editfiles*
  - Provee múltiples comandos para editar archivos:
    - AppendIfNoSuchLine, LocateLineMatching, RepaceWith, etc.
  - Ejemplo:

```
editfiles:
  redhat::
    { /etc/sysconfig/rhn/up2date
      LocateLineMatching '^forceInstall=[1-9]'
      ReplaceLineWith 'forceInstall=0'
      DefineClasses 'up2date_forceinstall' }
```

# Acciones

- *processes*
  - Manipula procesos.  Puede enviar señales (kill, hup, etc), reiniciar, definir clases, etc.

```
processes:

    ntp_conf_modified::
        "ntpd" signal=kill restart "/etc/init.d/ntpd restart"
```

# Acciones

- *tidy*:  Borra archivos del sistema que no son necesarios, por ejemplo archivos temporales, etc.
- *disable*: Renombra archivos que no son necesarios, pero que no se quiere borrar necesariamente.
- *directories*: Crea directorios
- *links*:  Crea enlaces
- *shellcommands*: Permite ejecutar scripts
- *packages*: Instala software empaquetado por la distribución (se puede definir el empaquetador)

# Instalación (en breve)

- Definir una máquina que funcionará como servidor (policyhost)
  - Poner todas las configuraciones en un directorio maestro
    - Ej. /var/lib/cfengine2/master/
  - Configurar el servidor en /var/lib/cfengine2/inputs/cfservd.conf
  - Ejecutar *cfservd*

# Configuración del Agente

- En Ubuntu: /var/lib/cfengine2/inputs
- Dos etapas:
  - *update.conf*: Copia todos los demás archivos de configuración desde el servidor
    - Configuración muy simple y estática
  - *cfagent.conf*: Controla toda la funcionalidad del agente cfengine
    - Se puede separar en varios archivos, usando *import*

# Instalación (en breve)

- Crear un *update.conf* básico que copie las configuraciones en /var/lib/cfengine2/inputs
- Crear un *cfagent.conf* definiendo las políticas
- Instalar cfengine en las máquinas cliente
  - Copiar la clave pública del cliente en el servidor
    - /var/lib/cfengine2/ppkeys (localhost.pub → root-192.168.1.10.pub)
  - Copiar la clave pública del servidor en el cliente (o utilizar trustkey=true)
  - Copiar el archivo *update.conf* en /var/lib/cfengine2/inputs
  - Ejecutar *cfexecd (*/etc/init.d/cfexecd start)

# Pruebas

- Ejecutar cfservd en modo debug

```
cfservd -d2
```

- Ejecutar cfagent directamente

```
cfagent -Kqv
```

- Definir una clase de prueba que restrinja el dominio de acción

```
testhosts = ( host1 )

testhosts::
  <acciones nuevas>
```

# Más pruebas

- Probar las configuraciones de servicios antes de instalarlos
  - Indispensable para servicios al público
    - (Apache, Sendmail, Bind, etc)
  - Se logra haciendo las copias en dos etapas
    - Etapa1:
      - Copiar del servidor al cliente en un directorio temporal
      - Ejecutar un script que chequee el archivo de configuración
    - Etapa2:
      - Si la prueba es satisfactoria, copiar el archivo en el lugar definitivo
      - Reiniciar el servicio

# Control de versiones

- El directorio maestro en el servidor debería mantenerse bajo un sistema de control de versiones (ej. Subversion)
  - Sólo deben guardarse los cambios una vez se hayan probado

# Referencias


- http://www.cfengine.org/docs/cfengine-Reference.html
- http://www.cfengine.org/docs/cfengine-Tutorial.html
- *Automate System Configurations and Changes with cfengine*. John Borwick. http://www.samag.com/documents/s=9936/sam0601e/
- *Introducing Cfengine.* Luke Kanies. http://www.onlamp.com/pub/a/onlamp/2004/04/15/cfengine.html
- Otras herramientas
  - *Puppet*: http://reductivelabs.com/trac/puppet