

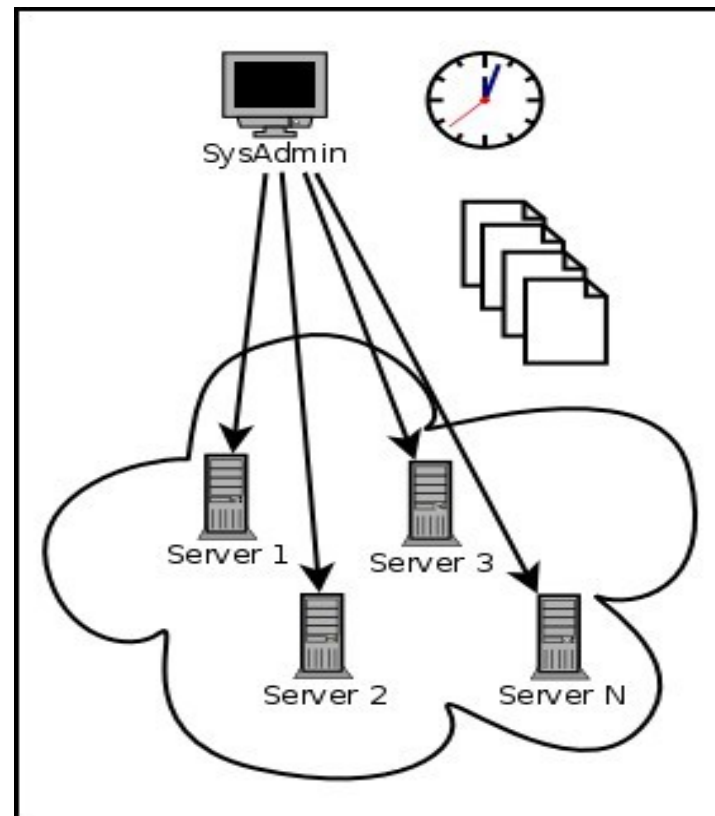
**Gestion centralizada equipos**

**ironotec**  
Internet y Sistemas sobre GNU/Linux

Mikel Jimenez Fernandez <mikel@irontec.com>  
Irontec - Internet y Sistemas sobre GNU/Linux

## Introducción

Cuando el numero de equipos a administrar crece, resulta una tarea muy tediosa el encargarse uno por uno de modificar y verificar configuraciones que sean comunes a todos.



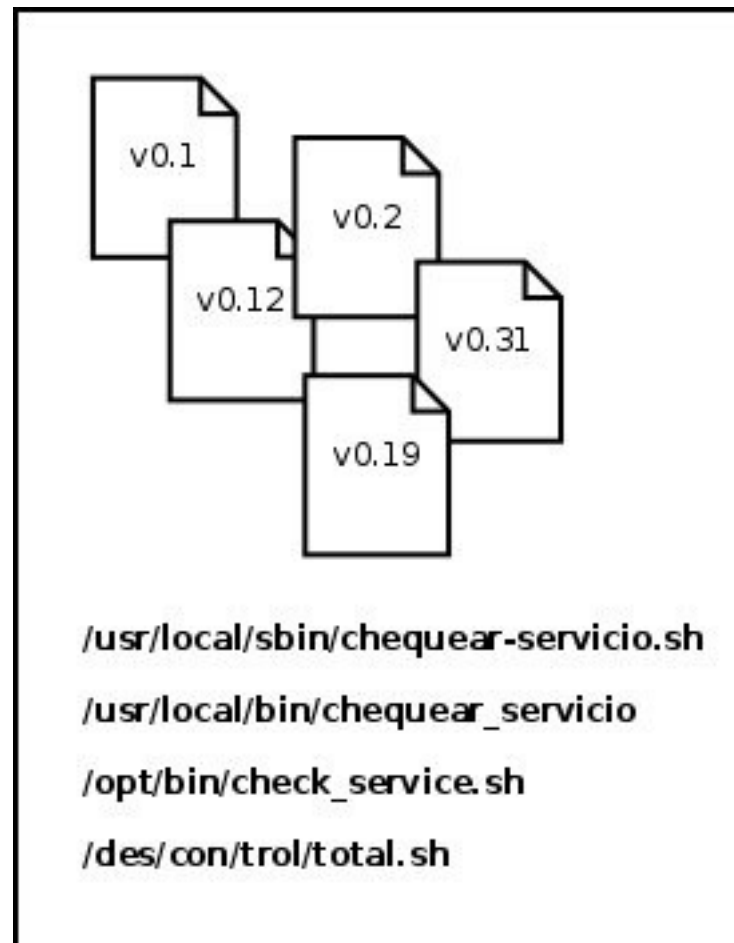
## Introducción

Es una tarea muy repetitiva y generalmente, siempre se nos “escapa” algo.

En caso de querer automatizar este proceso, es necesario ser muy meticuloso a la hora de programar el script encargado de hacerlo, y muchas veces nos resulta muy complicado,

## Introducción

ya que nos podemos encontrar con entornos dispares, tales como, diferente versión distribución, diferente distribución, rutas a los binarios/ficheros...

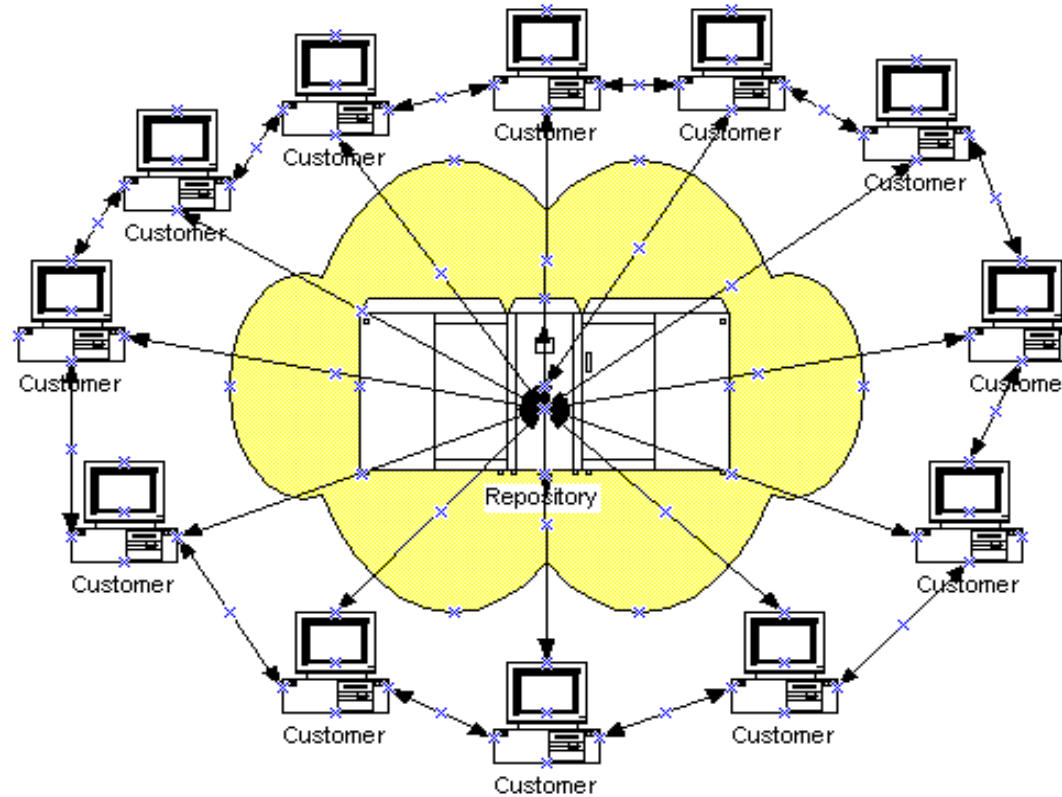


## Introducción

Es por ello que resulta muy interesante el contar con una herramienta que nos ayude en este proceso, tanto para el despliegue inicial, como para posteriores actualizaciones, revisiones...

La idea es gestionar todo desde un repositorio centralizado , desde el cual beberán los servidores que deseamos “sincronizar”.

# Introducción



Todo esto también se extrapola a la instalación y gestión de paquetes (que paquete, que versión...).

## Introducción

Esta estructura facilita muchísimo la gestión de los equipos, ya que solamente tendríamos un punto de acción, el cual se sincronizará con los nodos conectados al mismo.

## Herramientas gestión

A continuación se analizarán algunas herramientas destinadas para este fin.

Las aplicaciones que se analizarán serán:

- Spacewalk
- Capistrano
- Landscape
- Cfengine
- Puppet



Spacewalk es una solución libre de gestión de sistemas Linux/Solaris.

Spacewalk permite:

- Inventariado de sistemas (hardware y software)
- Instalación y actualización software en los equipos
- Monitorización
- Despliegue de configuraciones unificadas
- Todo gestionable desde interfaz gráfica

## Spacewalk

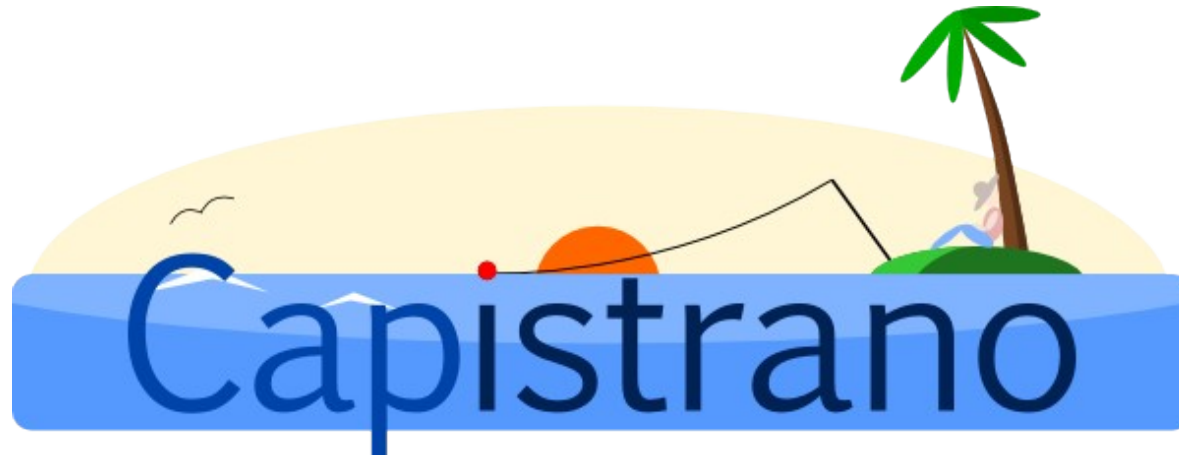
Spacewalk es el sistema en el que esta basado la “Red Hat Network Satellite”, el sistema de gestión de servidores centralizado que utiliza RedHat.

Spacewalk funciona es sistema RedHat y Fedora y en las distribuciones derivadas de estas, tales como Centos, Scientific Linux, etc...

El soporte para Debian esta siendo desarrollado actualmente, pero actualmente no es usable, ya que esta en una fase muy experimental.

Es por esto que Spacewalk no es compatible con sistemas Debian ni con sus derivados como Ubuntu.

## Capistrano



Capistrano es una herramienta de gestión de tareas automatizadas que se sirve de SSH para ejecutar comandos en maquinas remotas.

## Capistrano

Capistrano está escrito íntegramente en Ruby, y este mismo lenguaje es el que utiliza en su configuración.

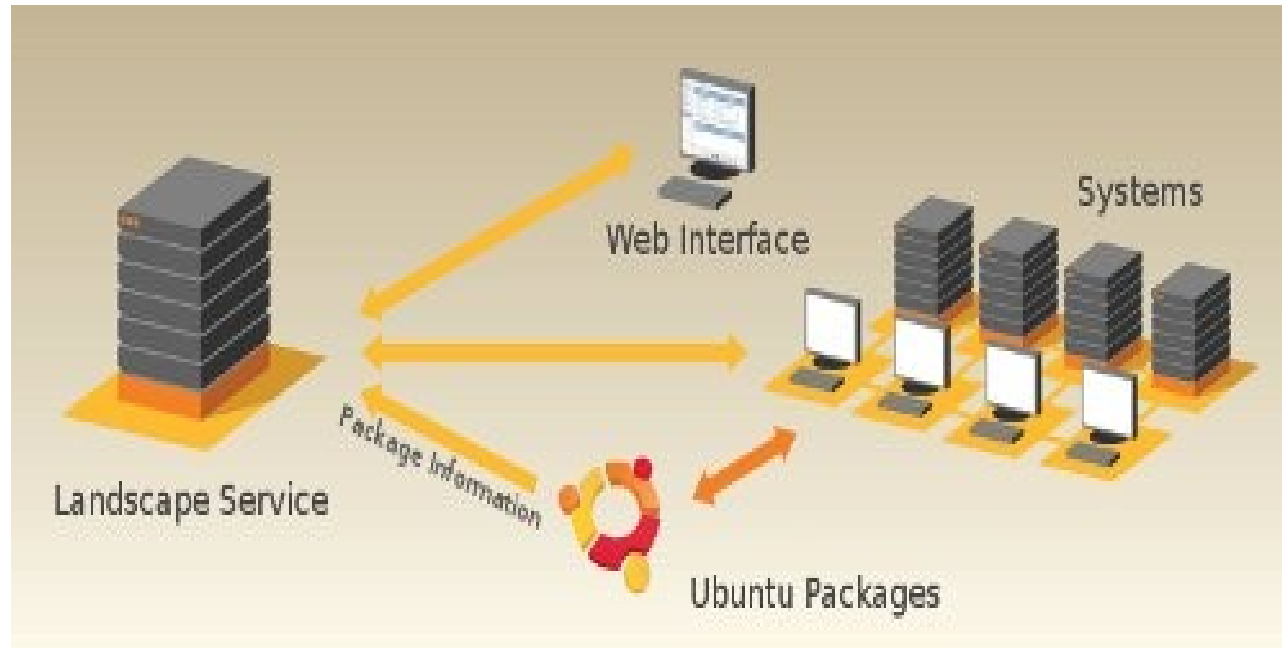
Su estructura es muy simple, ya que no se sirve de ningún demonio cliente o servidor.

Utiliza la filosofía “on demand”, esto es, la acción que deseemos realizar, se ejecutará en el instante que lancemos la orden.

Esto plantea un problema con los equipos que en el momento de ordenar la tarea estén apagados o sin conectividad, que estos equipos “no se enteraran” de la orden.

A nivel de escalabilidad se comporta bien en entornos pequeño-medios, pero cuando el volumen de los servidores crece, se hace mas difícil su mantenimiento.

## Landscape



Landscape es la herramienta desarrollada por Canonical, para la monitorización y administración centralizada de equipos Ubuntu.

<http://www.canonical.com/projects/landscape>

## Landscape

Landscape consta de una estructura cliente servidor, y toda la configuración y gestión de los equipos remotos se realiza mediante una interfaz web alojada en el servidor Landscape.

Al estar diseñada específicamente para Ubuntu, se integra a las mil maravillas con las peculiaridades del SO.

## Landscape

En cambio, la mayor pega de la implantación de esta solución, es su coste. (USD)

- 1 \$150
- 10 \$129
- 25 \$120
- 50 \$112
- 100 \$105
- 250 \$102
- 500 \$97,5

Para entornos educativos se aplicaría un descuento del 30%.

## Landscape

<http://www.canonical.com/projects/landscape/landscape>

Entre otras muchas funcionalidades, permite:

- Instalación y gestión de paquetes
- Gestión centralizada configuraciones
- Repositorio paquetes centralizado
- Ejecución de scripts en maquinas remotas
- Monitorización sistemas
- Capacidad de administrar maquinas “off-line”
- Gestión usuarios, grupos, tareas programadas (cron) ...

Esto permite llegar a una granularidad de parametrizacion muy grande, sin necesidad de bajar a la linea de comandos.

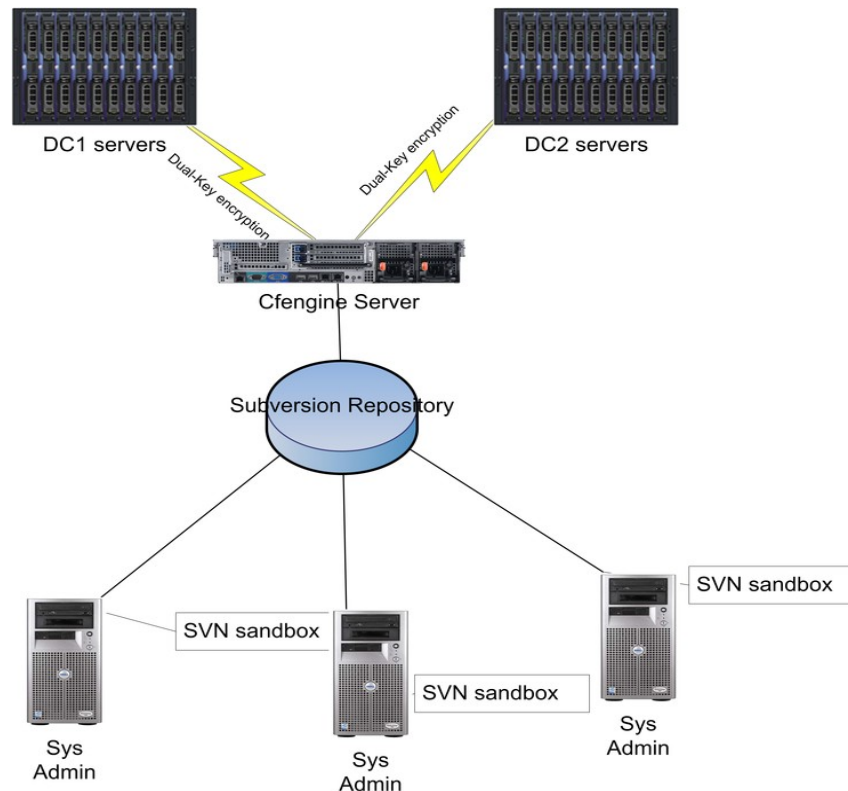


Cfengine es una herramienta de gestión centralizada de equipos.

Cf=Configuration Engine=Motor

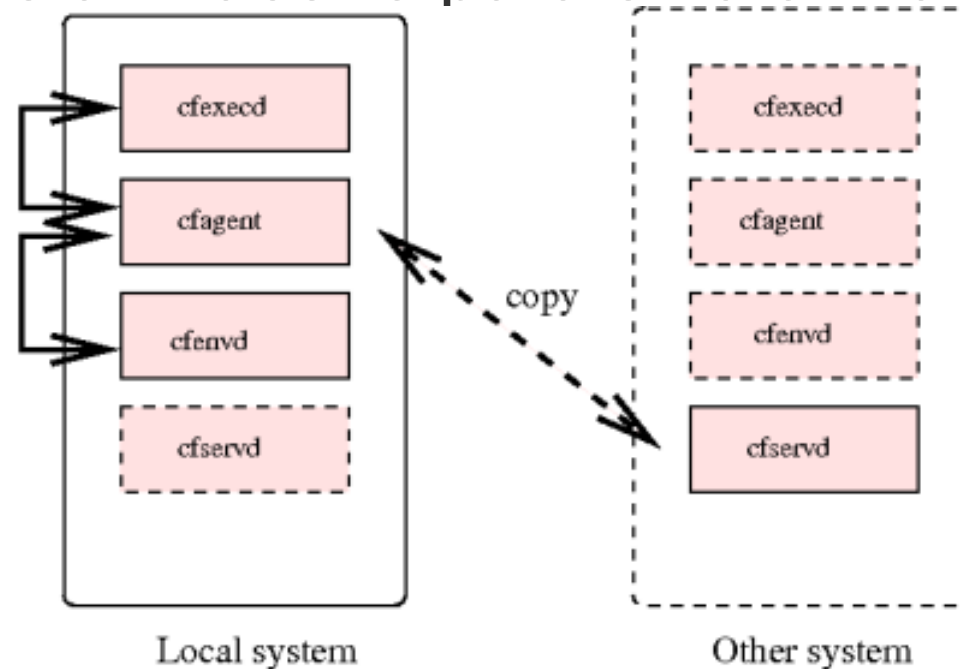
Utiliza una estructura cliente-servidor, esto es, los cambios hechos en el servidor o maestro, se replican a los “n” clientes o esclavos.

Cfengine esta principalmente diseñado para la administración y gestión de ficheros de configuración, pero también puede gestionar puntos de montaje, trabajos cron, paquetes instalados...



Al igual que Landscape, soporta la gestión de equipos desconectados.

El maestro ordenara un cambio y si un cliente afectado por ese cambio estuviese desconectado, al arrancar observaría que su numero de revisión de configuración es inferior. Entonces solicitaría los cambios al maestro para sincronizar su estado.



El proyecto Cfengine lleva activo desde 1993 y es muy conocido para estas funciones. Cfengine utiliza un lenguaje de alto nivel para definir las acciones a realizar.

La configuracion se realiza integramente desde consola y mediante ficheros de configuracion.

A pesar de su potencia tiene desventajas como:

- Sintaxis compleja
  - Para hacer poco, hay que escribir mucho
- No reutilización de código
- No utiliza capa de abstracción de recursos
  - Importante en entornos con varias distribuciones
- Complejidad de interpretación

Cfengine es un sistema profesional de gestión centralizada de equipos, muy potente, del que se echan de menos en ocasiones mayor flexibilidad y facilidad.

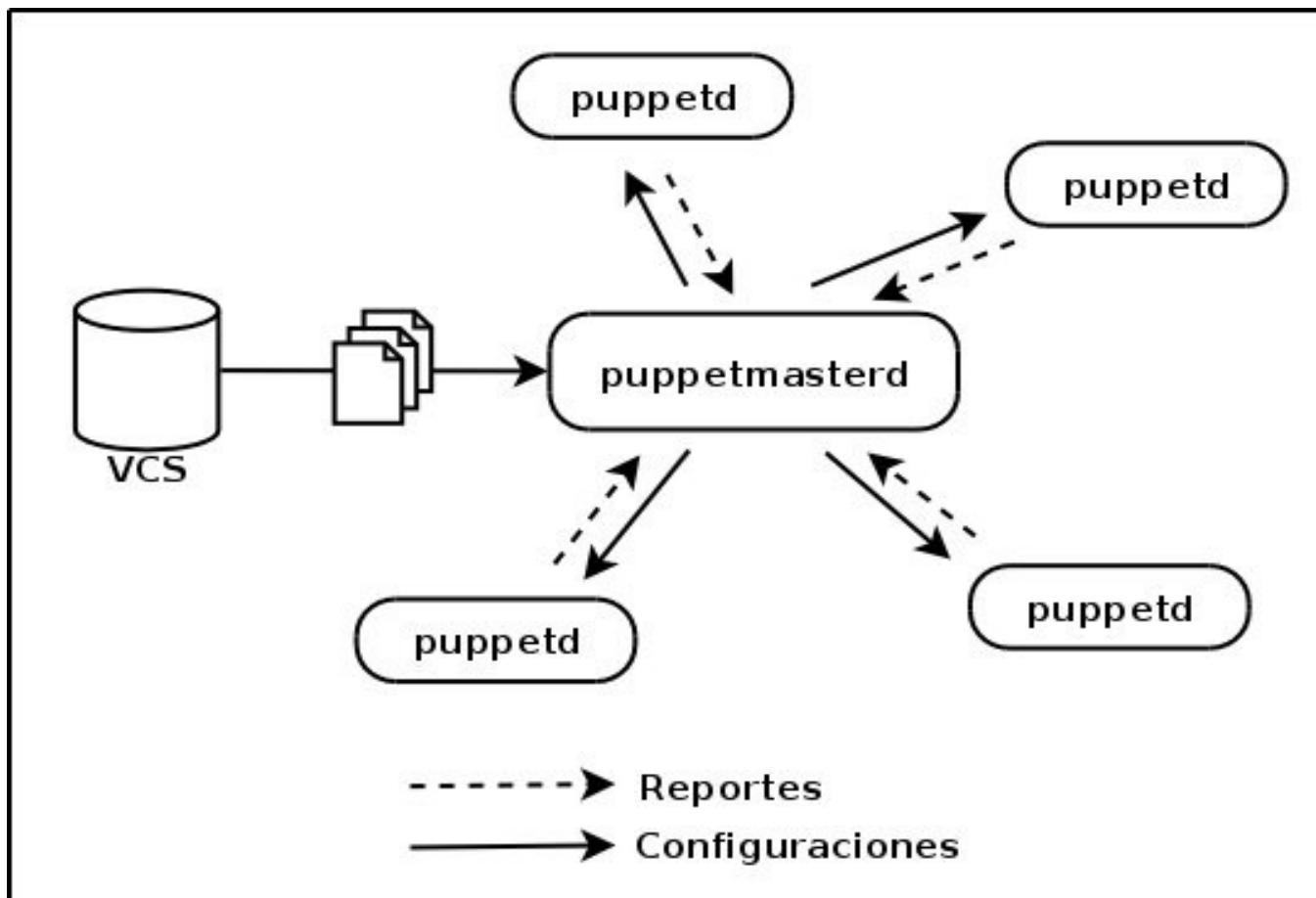
También cabe decir que la curva de aprendizaje es lenta.

**Puppet**

Puppet es un sistema muy muy parecido a Cfengine, de hecho nace con muchas referencias a Cfengine pero intentando “arreglar” las “limitaciones” de este, intentando que sea mas “human readable” y utilizando un lenguaje de mas alto nivel que Cfengine.

# Puppet

Utiliza una estructura clientes servidor, y la sincronización del cliente y el servidor, se realiza mediante XMLRPC sobre un canal seguro SSL.



## Puppet

Puppet funciona sobre RedHat, Fedora, Gentoo, FreeBSD, Mac OS X...

Puppet permite al igual que Cfengine:

- Gestión de configuraciones/ficheros
- Gestión usuarios, grupos
- Gestión paquetes
- Scripts, Cron...
- Integración subversion
- ...

## Puppet

Uno de los puntos mas fuertes de Puppet, es la abstracción de recursos, esto es, si deseamos gestionar los usuarios del sistema, nos serviremos del recurso “users”, sin preocuparnos del comando en concreto a utilizar.

De igual modo, si deseamos instalar un paquete en un sistema Debian, utilizaremos el recurso “package”. Lo mismo haremos para un sistema basado en Fedora, y la abstracción de Puppet se encargará de usar “apt” o “yum”, según la distribución detectada.

Otro punto a tener en cuenta es el metalenguaje utilizado en los ficheros de configuracion de Puppet, frente a Cfengine.

## Puppet

Como hemos comentado anteriormente, Puppet utiliza un lenguaje de mayor alto nivel, enfocado a la gestión de *recursos*.

Estos son los recursos que se pueden administrar mediante Puppet

- User
- File
- Package
- Service
- Cron
- .....

Dentro de cada recurso, podríamos especificar los parametros de cada uno:

## En caso del User

- Home
- Uid
- Gid
- Shell
- ...

## File

- User
- Group
- Mode
- Source
- ....

## Paquete

- Ensure (absent, purged, present/installed)
- status
- ...

## Service

- Ensure (running/stopped)
- Enable (true/false)
- ...

## Cron

- Day
- Hour, minute, second
- Command...

## Puppet

Puppet utiliza la filosofía de “que estado quiero obtener”, abstrayéndose del “modo” de hacerlo, a diferencia de Cfengine, que trabaja a un nivel mas bajo.

Cabe decir que Puppet también permite bajar a bajo nivel, permitiéndonos un nivel de especificación muy grande de las configuraciones de nuestros servidores.

A continuacion mostramos algún ejemplo de configuración:

En este caso definimos una clase llamada “fichero” que setea el usuario y grupo de los ficheros “/etc/passwd” y “/etc/shadow” a root:root con permisos 644 para “/etc/passwd” y 400 a “/etc/shadow”.

```
class fichero {  
  file {  
    "/etc/passwd":  
      owner => "root",  
      group => "root",  
      mode => 644;  
    "/etc/shadow":  
      owner => "root",  
      group => "root",  
      mode => 440;  
  }  
}
```

En este caso definimos una clase llamada “norootssh” para no permitir al usuario root acceder por ssh.

```
class norootssh {
  package { "openssh-server":
    ensure => present
  }
  service { "ssh":
    require => Package[openssh-server],
    ensure => running
  }
  exec { "perl -pi -e 's/^PermitRootLogin yes$/PermitRootLogin no/' /etc/ssh/sshd_config":
    path => "/bin:/usr/bin:/sbin:/usr/sbin",
    unless => "grep 'PermitRootLogin no' /etc/ssh/sshd_config",
    notify => Service[ssh]
  }
}
```

Asignamos la clase “fichero” al host “wwwserver” y la clase “norootssh” al host “mailserver”.

```
node wwwserver {  
    include fichero  
}  
  
node mailserver {  
    include norootssh  
}
```

## Conclusion

Después de analizar las diferentes aplicaciones, se sacan las siguientes conclusiones:

- En el caso en el que el presupuesto no sería problema, la mejor opción sin duda sería Landscape, ya que esta especialmente diseñado para Ubuntu, y se tendría soporte oficial por parte de Canonical. A parte de la getsion centralizada de paqueteria, configuraciones... consta de una potente herramienta de monitorización e inventariado de equipos.
- Spacewalk se descarta por la poca madurez del proyecto en entornos Debian. Quizas estará listo en un futuro, pero según las conversaciones con el desarrollador, aun queda mucho para que se un sistema usable en producción.

## Conclusion

- En cuanto a Capistrano, se echan de menos muchas funcionalidades que el resto de sistemas analizados si ofrecen y es por ello que se descarta.
- Los dos candidatos principales son Puppet y Cfengine.
- Puppet ofrece mayor flexibilidad sobre Cfengine
  - Mayor facilidad de administración
  - Ficheros de configuración menos extensos (reutilizacion de codigo, templates)
  - Una comunidad muy activa
  - Plugins
- Las dos herramientas son de entorno consola, pero existe algun visor web para Puppet (PuppetView, PuppetShow)

**Conclusion**

Desde Irontec recomendamos Puppet como mejor opción, dadas las características (extensión, número de equipos...) al que está dirigido este análisis

**TABLA RESUMEN**

	<b>Admin Grafica</b>	<b>Debian</b>	<b>Rollback</b>	<b>Inventar io</b>	<b>S.L.</b>	<b>Cliente/ servidor</b>	<b>Complejida d</b>
<b>Capistra no</b>	No	Si	Si	No	SI	No	Media
<b>Landsca pe</b>	SI	SI	No	Si	Si/No	Si	Baja
<b>Spacewa lk</b>	Si	No	No	Si	Si	Si	Baja
<b>CfEngine</b>	No	Si	No	No	Si	Si	Muy alta
<b>Puppet</b>	No	Si	No	No	Si	Si	Alta