

*NOTA: Se utilizó Debian lenny para hacer las correspondientes pruebas.*

## ¿Qué es?

Puppet es una herramienta para las tareas de administración automática de sistemas. Te permite automáticamente tener el control de una gran cantidad de máquinas enviándole órdenes a través del protocolo XML-RPC para que realicen determinada tarea.

## ¿Cómo funciona?

Funciona bajo la arquitectura cliente servidor. Por una lado tenemos el servidor, donde indicamos los elementos que queramos que posea el cliente. El servidor es un paquete llamado puppetmaster y su configuración se encuentra bajo el directorio */etc/puppet*. También tenemos al cliente, el cual recibe las ordenes y peticiones del servidor. Por ejemplo, si deseamos instalar un paquete, o ejecutar algún comando en determinadas computadoras configuraríamos el servidor con dicha orden y automáticamente en un intervalo de tiempo (30 minutos por defecto) recibirán y ejecutarán dichas órdenes.

## Instalación y uso

Comenzaremos por configurar el cliente. Para eso lo instalamos:

```
aptitude install puppet
```

Tanto el cliente como el servidor tienen sus archivos de configuración */etc/puppet*. EL único archivo que debemos editar por parte del cliente es *puppet.conf*.

```
[main]
server = puppetmaster0.ipostel.sl ## FQDN del servidor puppetmaster
logdir=/var/log/puppet
vardir=/var/lib/puppet
ssldir=/var/lib/puppet/ssl
rundir=/var/run/puppet
factpath=$vardir/lib/facter
report=true
factsync=false

[puppetmasterd]
certname=puppetmaster0
templatedir=/var/lib/puppet/templates
```

Una vez configurado esto, sólo faltaría el intercambio de llaves entre el cliente y el servidor, es una especie de "autenticación" entre ambos que explicaremos como realizar próximamente. Hasta entonces el cliente no podrá comunicarse con el servidor para recibir las peticiones. Por parte del cliente eso es todo lo que debemos configurar.

En cuanto a la instalación del servidor:

```
aptitude install puppetmaster
```

## Puppet

Aquí entran en juego dos archivos importantes: el primero es el del a configuración del servidor y el segundo es la configuración que tendrán los futuros clientes que se conecten al servidor.

Para la configuración del servidor debemos editar el archivo `/etc/puppet/fileserver.conf` y asegurarnos que tenemos declarado cual segmento de red puede conectarse y solicitar el servicio de nuestro servidor puppetmaster. El archivo de configuración debe quedar así:

```
[files]
  path /etc/puppet/files
  allow 172.26.16.0/24    Dirección de Red
```

El otro archivo que debemos crear es el `site.pp`, el cual debe ir dentro del directorio `/etc/puppet/manifests` (directorio ya creado por puppetmaster). Dicho archivo contendrá todas las especificaciones de cada cliente, a los cuales se hace referencia por su hostname o si quieres hacer referencia a todas, donde seleccionas el nodo *default*. Un ejemplo del archivo *site.pp*:

```
class clase_prueba {
  package { "vrms": ensure => installed }
  exec { "/usr/bin/aptitude update": }
  package { "nmap": ensure => installed }
  package { "less": ensure => installed }
  file { ["/tmp/archivo-prueba":
    ensure => present,
    mode   => 644,
    owner  => root,
    group  => root
  ]
}

node puppet {
  include clase_prueba
}

node belphegor {
  include clase_prueba
}

node default {
  include clase_prueba
}
```

También se puede usar el atributo: `import "classes/*.pp"` para mantener un orden en el sistema y solo hacer referencia a través de, por ejemplo el nodo default.

Una vez hecho la configuración inicial, y cualquier otra configuración que hagamos posteriormente: no olvidemos reiniciar el servicio puppetmaster luego de realizar los cambios.

```
/etc/init.d/puppetmaster restart
```

Ahora sólo falta decirle al cliente que nuestro servidor puppet está configurado y listo para atender peticiones. Los clientes consultan al servidor cuando reinicias el `puppet client` o cuando transcurre determinado intervalo de tiempo (30 minutos por defecto). Si no queremos esperar el tiempo determinado y deseamos que el cliente tome inmediatamente las órdenes del servidor debemos reiniciarlo:

```
/etc/init.d/puppet restart
```

## Puppet

El cliente debe mostrar la siguiente configuración:

```
[main] logdir=/var/log/puppet vardir=/var/lib/puppet ssldir=/var/lib/puppet/ssl rundir=/var/run/puppet  
factpath=$vardir/lib/facter pluginsync=true server=ip del servidor [puppetmasterd]  
templatedir=/var/lib/puppet/templates
```

Se nos presentará un problema que mencionamos anteriormente: El cliente y el servidor no se autentican y por ende no logran comunicarse. Cuando el cliente intenta realizar algún tipo de comunicación con el servidor éste las ignorará hasta que no confirmemos y creemos una llave pública para el cliente.

Primero debemos asegurarnos que el cliente logró establecer comunicación con el servidor; si lo hizo podremos ver en el servidor dicho intento de comunicación:

```
puppetca --list
```

Y nos aparecerá nuestro respectivo host: el cliente puppet.

Una vez logrado corroborar que nuestro cliente si logra "ver" al servidor puppetmaster debemos de firmar su llave pública. Para eso:

```
puppetca --sign HOST-CLIENTE-QUE-SE-VE-CON-PUPPETCA--LIST
```

otra opción seria agregar la opción en el archivo de configuración del servidor la opción autosign=true

Con esto debe bastar para la comunicación entre ambos, podríamos ver en el cliente puppet si los cambios fueron realizados correctamente o si el cliente hizo lo que el servidor le ordenó.

[Volver](#)