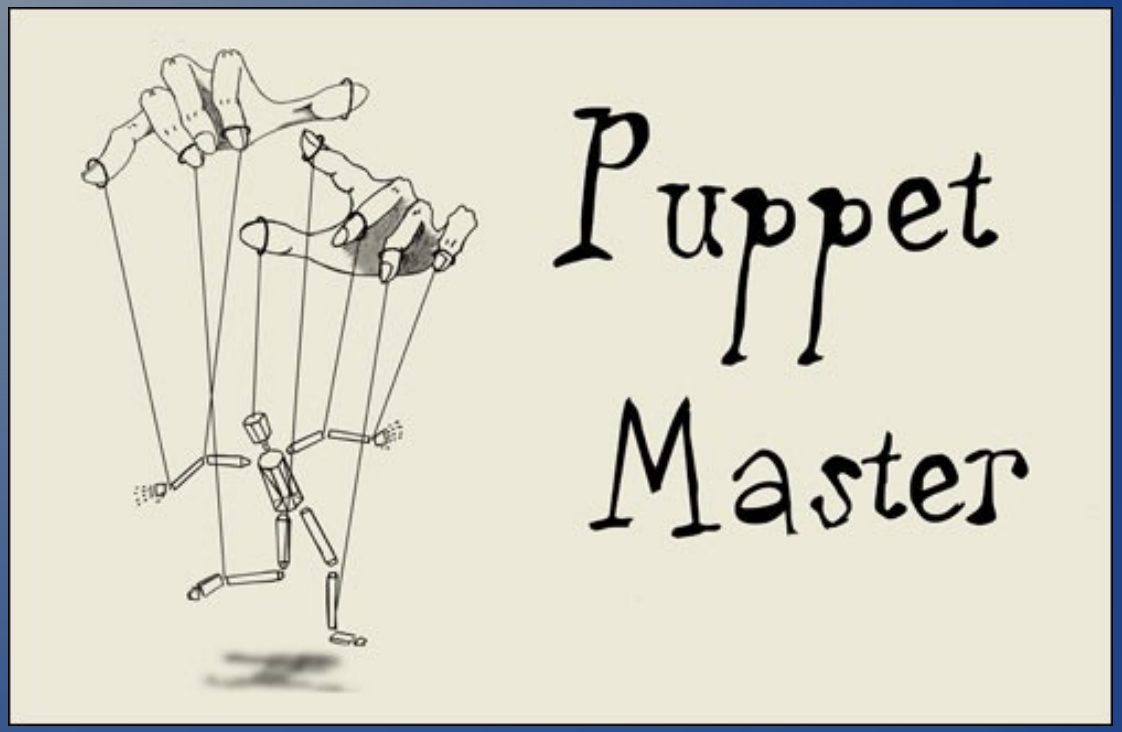
Orquestación de la Configuración con Puppet



Por: José Luis Jaime Gonzalez

Índice

- - Problemas comunes de un sys admin.
- - ¿Que es puppet?
- - ¿Como funciona?
- - Recursos
- - Creación de un modulo
- - Demo
- - Preguntas

Problemas comunes de un sys admin.

- Tareas repetitivas tiende a provocar errores.
- Poca portabilidad entre instalaciones (distintos sistemas operativos, distintas versiones etc.)
- Esfuerzo directamente proporcional al numero de servidores.
- ¿Que servicios están activos?, ¿Que usuarios están habilitados etc..?

¿Que es puppet?

- Puppet es una herramienta desarrollada por PuppetLabs para administrar la configuración de sistemas Linux/Unix y Windows.
- Basado en Ruby
- Utiliza un lenguaje declarativo

¿Que es un lenguaje declarativo?

- En puppet se describe el estado en el que quieres que este una maquina, y no como lograr ese estado.

  Ejemplo:

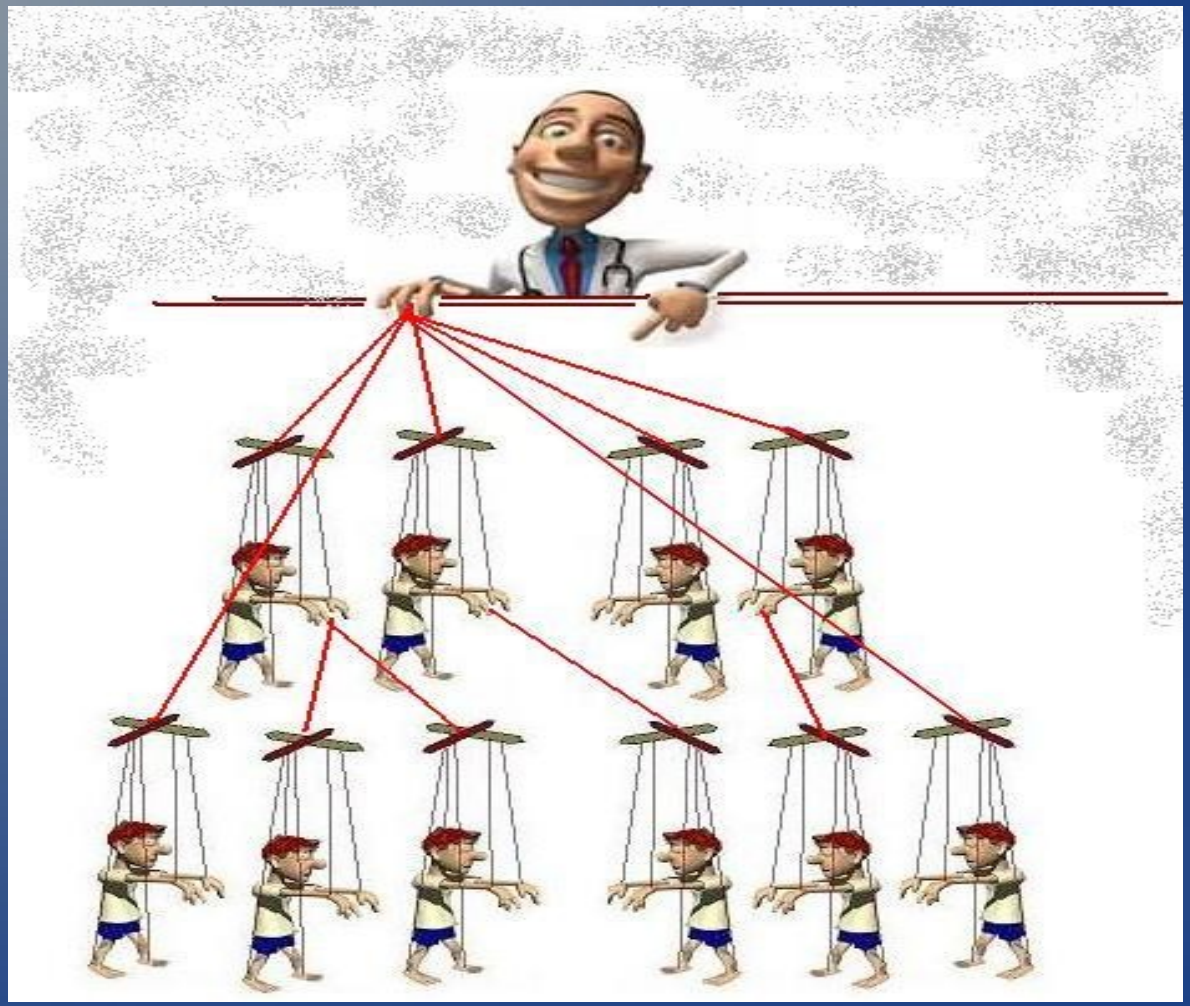
```
package {'apache2':
  ensure => installed,
}
```

```
service {'apache2':
    ensure => running,
}
```

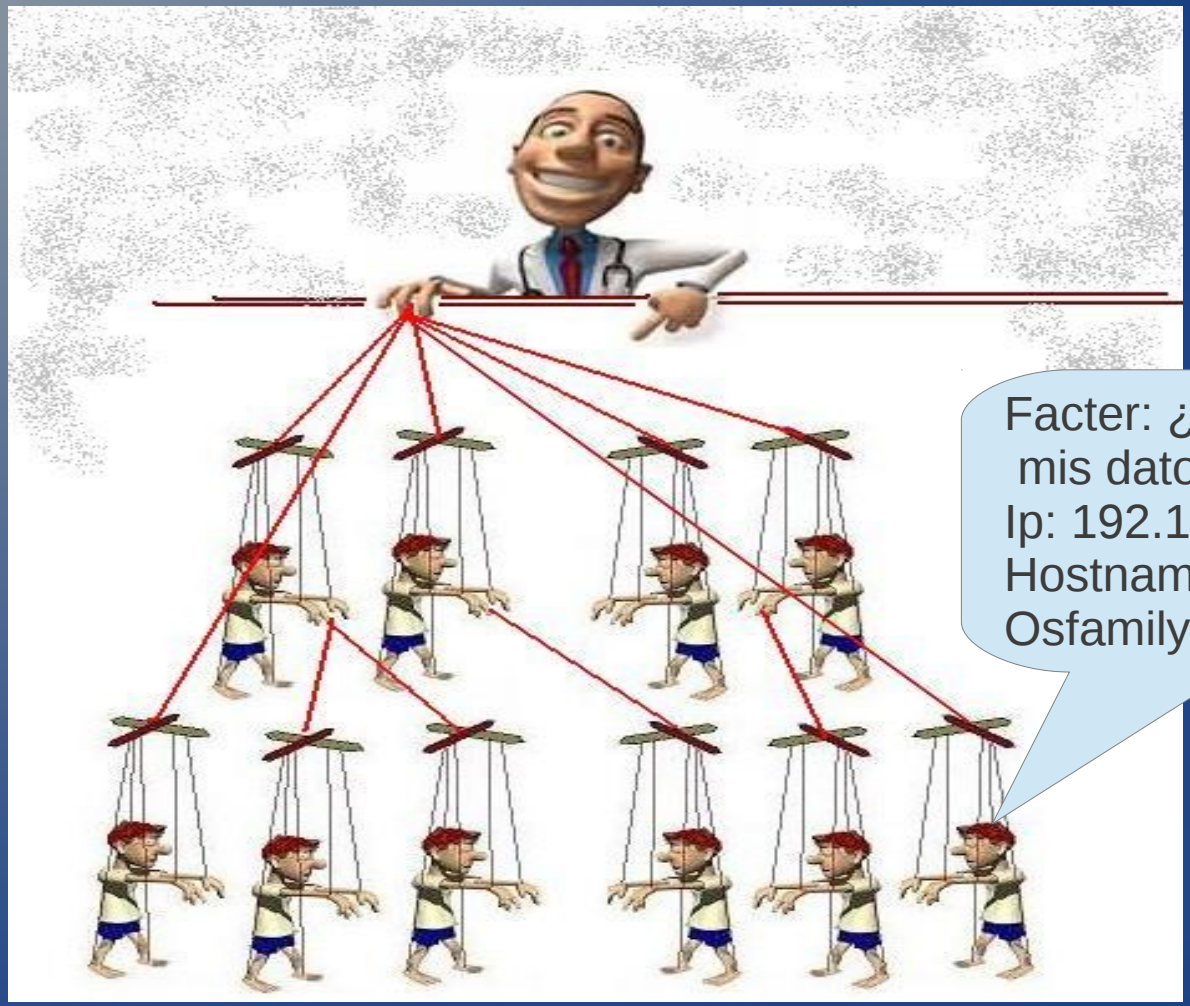
¿Como funciona?

- Puppet tiene una estructura de Cliente-Servidor.
  - Puppetmaster Servidor
  - Puppet Clientes
- La comunicación va cifrada mediante SSL, certificados.
- Funcionamiento de tipo “pull”.

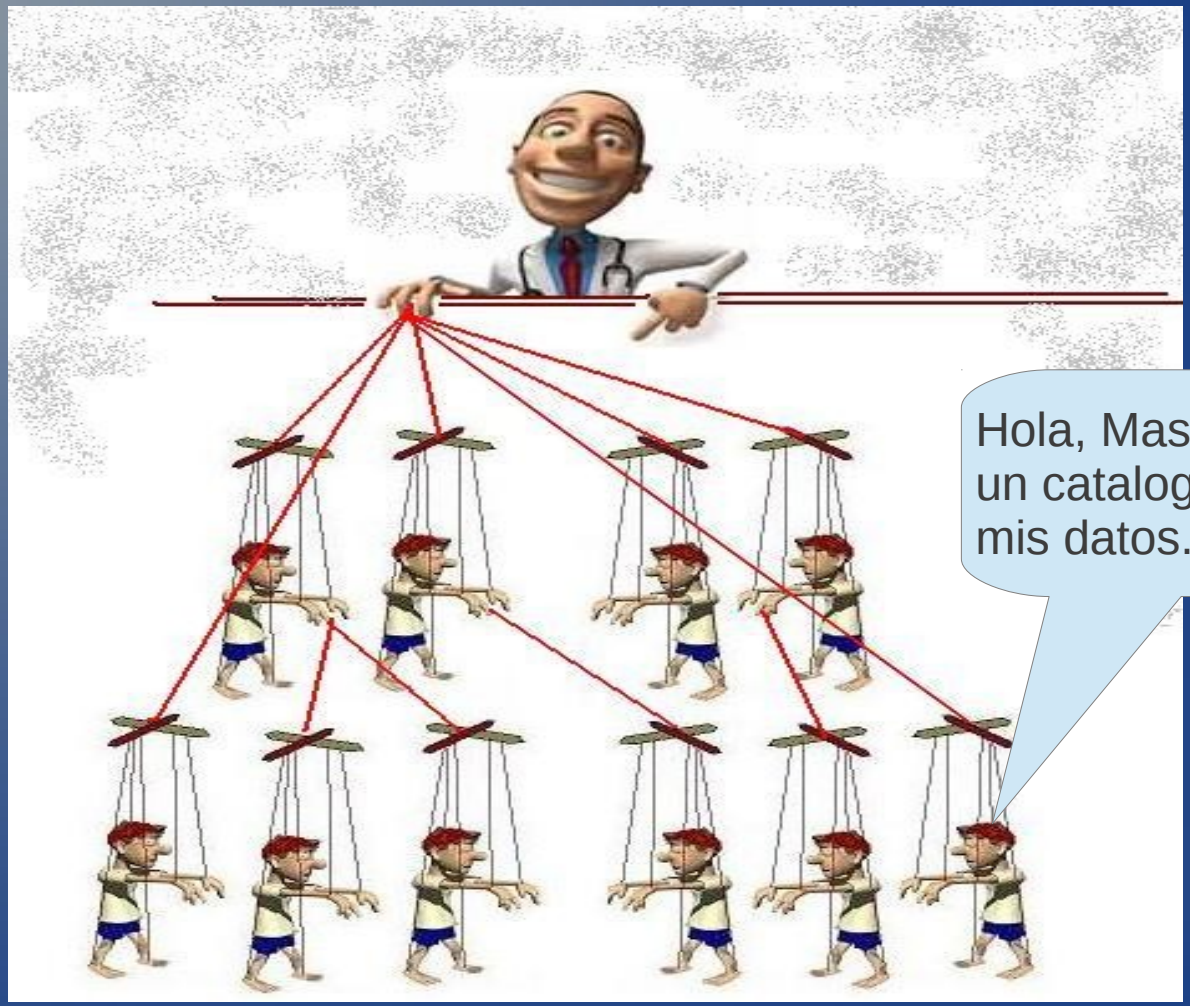
¿Como funciona?



¿Como funciona?

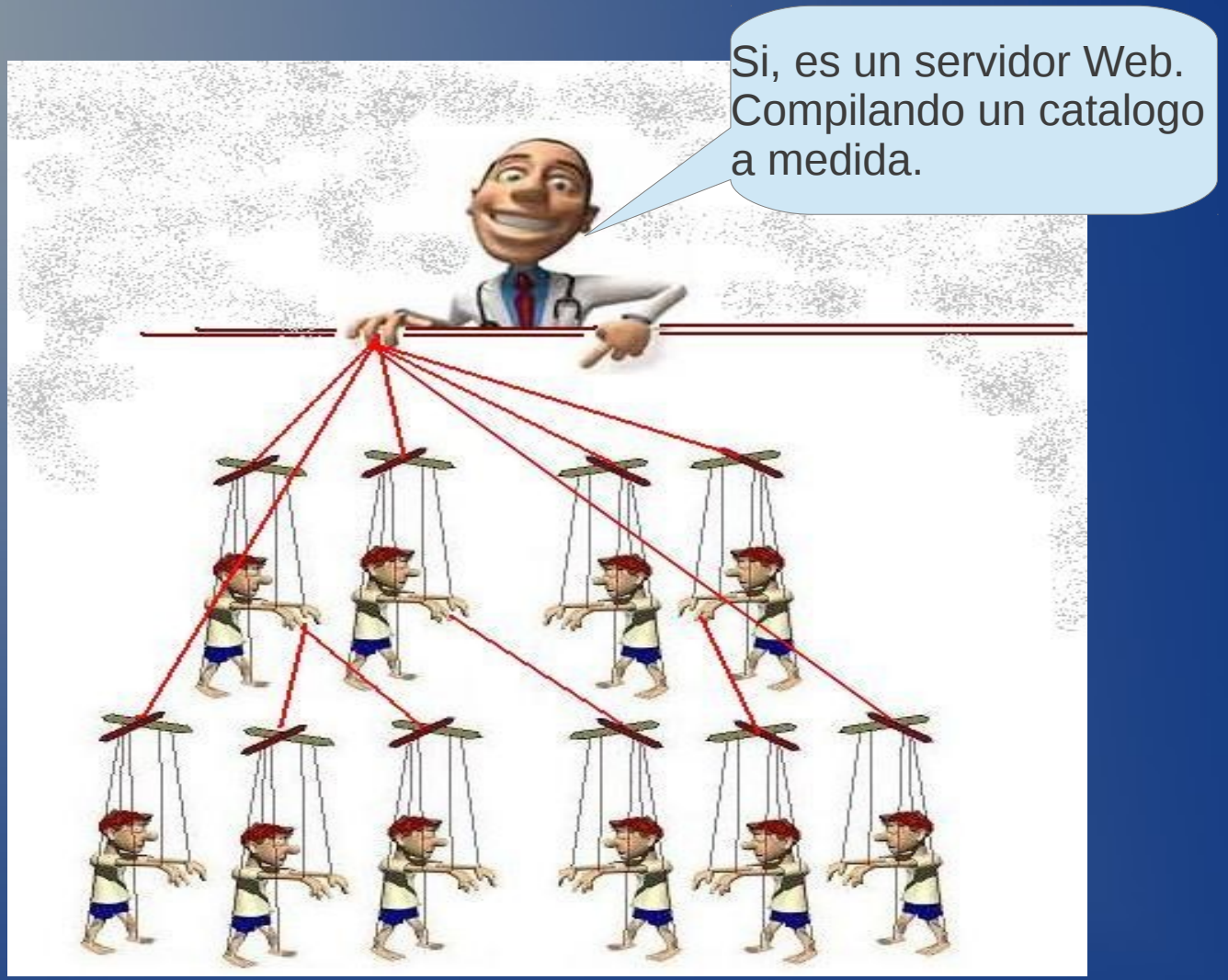


¿Como funciona?

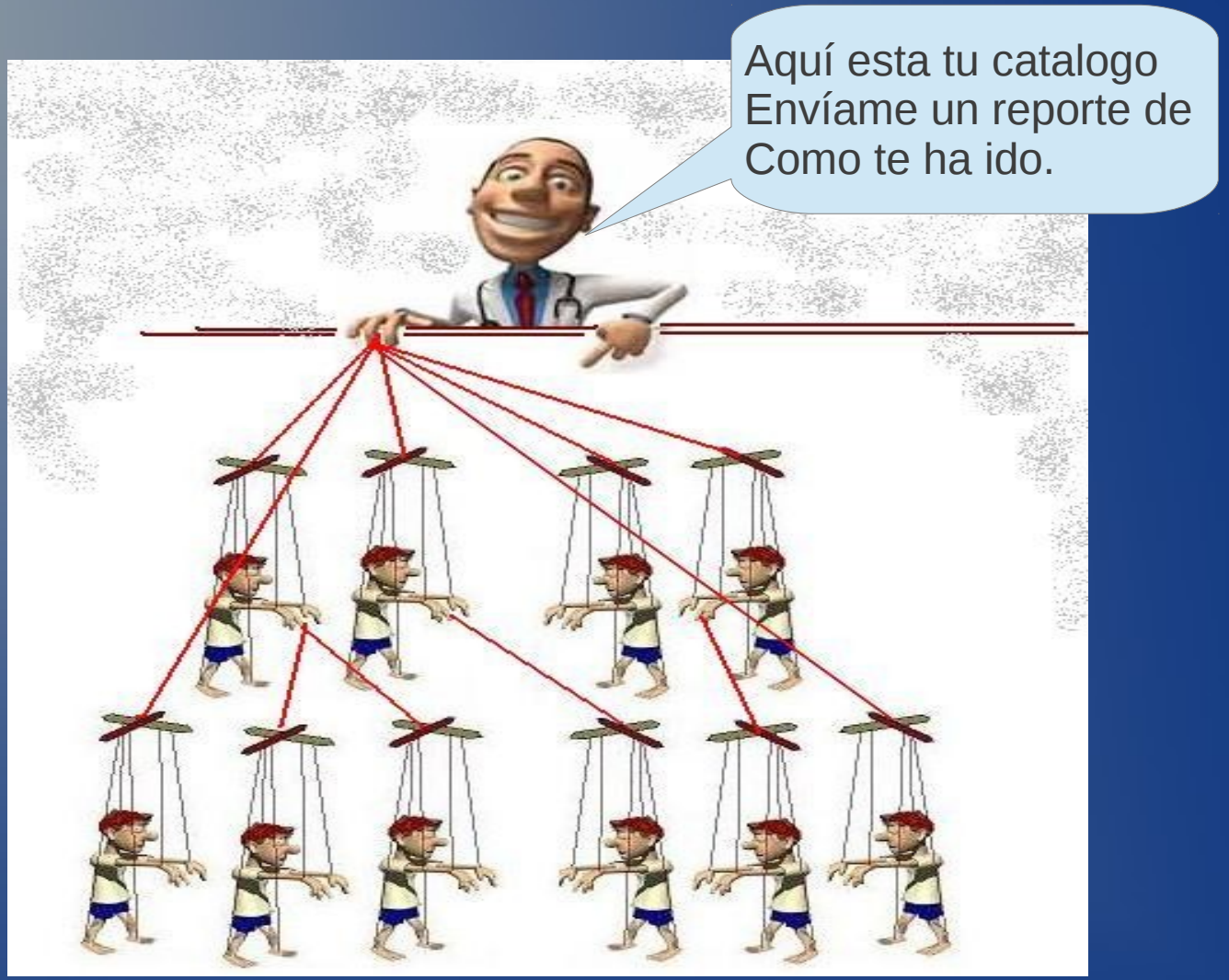


¿Como funciona?

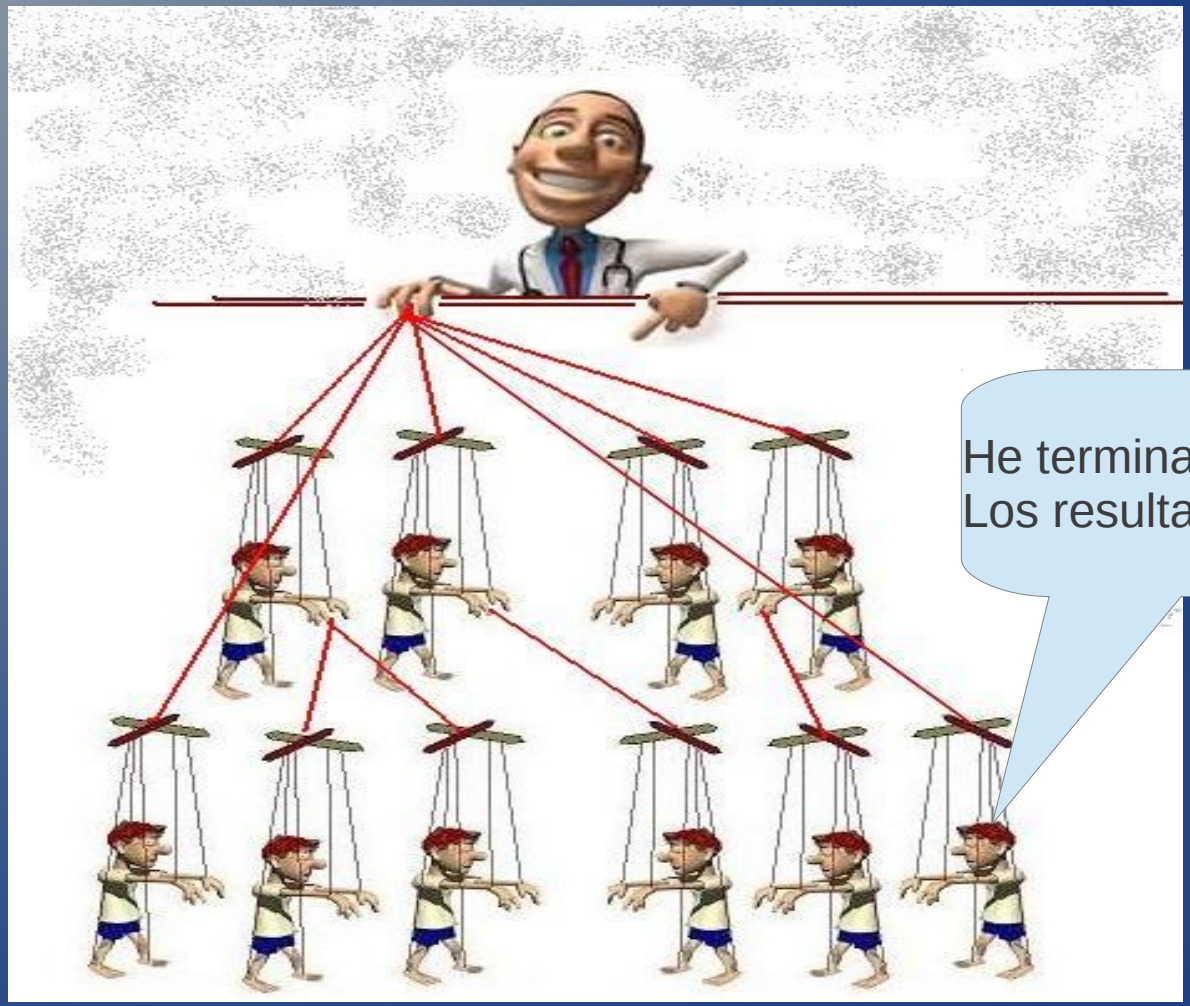
¿Como funciona?



¿Como funciona?



¿Como funciona?



¿Como funciona?

- ¿Que es un catalogo?
  - Un catalogo es un conjunto de recursos que definen el estado en el que tiene que estar un sistema.
- ¿Que es un recurso?
  - Un recurso puede ser la instalación de un paquete, un servicio, un fichero etc..

Recursos

- Paquetes, usuarios:

```
package { 'apache package':
    ensure => installed,
    name => "apache2",
  }
user {'www-data':
  ensure => present,
}
```

Recursos

- Ficheros:

```
file {'index.html apache':
    ensure => file,
    owner => www-data,
    group => www-data,
    mode => 0640,
    source => “puppet:///modules/apache/index.html”
    path => "/var/www/index.html",
}
```

Recursos

- Servicios, Comandos genericos:

```
service {'apache service':
    ensure => running/stopped,
    name => "apache2",
    }
 exec { "Extract tar.":
    path => "/bin:/usr/bin",
    unless => "find /tmp/prueba",
   command => "tar -xzf /tmp/prueba.tar.gz",
 }
```

Creación de un modulo.

- Estructura Módulos
  - /etc/puppet/modules/
    - Apache
      - Files : Ficheros estáticos
      - Templates : Ficheros dinámicos
      - Tests : Nos sirve para testear el modulo.
      - Manifests: init.pp, declaración de los recursos..

Creación de un modulo.

- Estructura Nodos
  - /etc/puppet/manifests/
    - nodes.pp : Fichero donde se definen los nodos

      node 'prueba.example.com' {

        include apache_simple

      }
    - site.pp : En este fichero importamos nodes.pp

      import “nodes.pp”

Creación de un modulo.

- Problema de hacer los módulos de esta manera:
  - Claridad. 10 paquetes, 12 ficheros, 5 servicios.
  - Flexibilidad, ¿puedo reutilizar el modulo con distinta configuración?
  - Compatibilidad, ¿es mi modulo compatible con varios sistemas?

Creación de un modulo.

- Claridad

  -Dividir el modulo en subclases

  - Init.pp
    - install.pp : Instalación de paquetes
    - config.pp : Ficheros de configuración
    - service.pp: Servicios

Creación de un modulo.

- Compatibilidad, Flexibilidad
  - Uso de variables:
    - Directorio index.html => /var/www/index.html
    - Directorio index.html => ${directorio_index}
  - Externalizar las variables:
    - Uso de HIERA

Creación de un modulo.

- ¿Que es Hiera?
  - Es un componente que viene de forma nativa a partir de la versión 3 de puppet.
  - Nos proporciona la posibilidad de externalizar las variables de los módulos.
  - Nos proporciona una estructura jerárquica.
  - Podemos usar ficheros .yaml, json, mysql etc.

Creación de un modulo.

Ejemplo: SNMP

- Centos => net-snmp
- Debian => snmp

:hierarchy:

- %{::osfamily}
- common

```
$snmp_package = hiera('snmp_package')
package {'snmp package':
        ensure => installed,
        name => ${snmp::snmp_package}
}
```

- hieradata/
  - Debian.yaml : snmp_package: 'snmp'
  - Centos.yaml: snmp_package: 'net-snmp'

Demo

- Modulo Lamp
  - Apache
  - Php
  - Mysql
- Modulo Tinyrss sobre Lamp
- Modulo Haproxy
- Scripts Automatización haproxy
- Puppet Dashboard

Preguntas

- Aclaración de Conceptos, dudas, consultas etc.

FIN