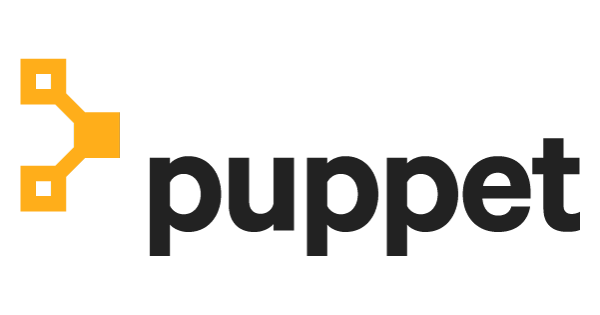


**UCLM – E.S. de Informática**

Ingeniería en Informática

GESTIÓN Y ADMINISTRACIÓN DE REDES

**GESTIÓN DE CONFIGURACIÓN CON PUPPET**



Autores:

Andrés Castellanos Cantos

Felipe Segovia Friginal

Contenido

[1. INTRODUCCIÓN 3](#_Toc129721330)

[**1.1 ¿QUE ES PUPPET?** 3](#_Toc129721331)

[**1.2. VENTAJAS E INCONVENIENTES DE PUPPET** 3](#_Toc129721332)

[**1.2.1. VENTAJAS** 3](#_Toc129721333)

[**1.2.2. INCONVENIENTES** 4](#_Toc129721334)

[**1.2.3. COMPARACIÓN CON OTROS SERVICIOS** 4](#_Toc129721335)

[**1.3. FORMAS DE ACCESO A PUPPET** 4](#_Toc129721336)

[**1.3.1. ACCESO A PUPPET CON ENDPOINT TCP** 4](#_Toc129721337)

[**1.3.2. ACCESO A PUPPET CON SOCKET UNIX** 5](#_Toc129721338)

[2. MONITORIZACIÓN DE PUPPET 5](#_Toc129721339)

[**2.1. MONITORIZACION DE PUPPET CON FACTER** 5](#_Toc129721340)

[**2.2. TIPOS DE EVENTOS O NOTIFICACIONES DE PUPPET** 6](#_Toc129721341)

[3. GESTIÓN DE PUPPET 7](#_Toc129721342)

[4. REDUNDANCIA Y BALANCEO DE CARGA EN PUPPET 7](#_Toc129721343)

[**4.1. CÓMO USAR EL BALANCEADOR DE CARGA** 8](#_Toc129721344)

[5. LENGUAJE EN PUPPET 8](#_Toc129721345)

[**5.1. MÓDULOS** 9](#_Toc129721346)

[**5.2. RECURSOS** 9](#_Toc129721347)

[**5.3. VARIABLES** 9](#_Toc129721348)

[**5.4. MANIFESTOS** 9](#_Toc129721349)

[6. COMPONENTES 10](#_Toc129721350)

[7. CONCLUSIÓN 11](#_Toc129721351)

[8. BIBLIOGRAFIA 11](#_Toc129721352)

# **1. INTRODUCCIÓN**

## **1.1 ¿QUE ES PUPPET?**

***Puppet*** es un sistema de gestión de configuraciones que nos permite automatizar configuraciones en nuestros servidores de forma fácil y dinámica en un solo fichero de configuración. Es una herramienta desarrollada por ***PuppetLabs***, está escrito en el lenguaje Ruby.

Con Puppet, se pueden definir y gestionar la configuración de los sistemas operativos, aplicaciones, servicios y otros componentes de infraestructura. Puppet puede ser utilizado en una variedad de sistemas operativos, incluyendo Linux, Unix y Windows, y es ampliamente utilizado en entornos empresariales y en la nube. Es una herramienta Open Source bajo la licencia Apache 2.0.

***Puppet Enterprise***, la solución empresarial de Puppet, se ofrece a través de una suscripción anual, cuyo precio se basa en el número de nodos que se van a gestionar. Los precios comienzan en alrededor de ***$10 USD por nodo por año*** y pueden disminuir a medida que se gestionan más nodos. Puppet también ofrece una ***versión gratuita y de código abierto llamada Puppet Community***, que se puede descargar y utilizar de forma gratuita.

Además de Puppet Enterprise, Puppet también ofrece otros productos y soluciones:

— ***Bolt***: una herramienta de automatización de scripts.

— ***Continuous Delivery for Puppet*** Enterprise: una solución de entrega continua.

— ***Puppet Remediate***: una solución de gestión de vulnerabilidades.

Los precios de estos productos también varían y se basan en factores como el número de usuarios y la cantidad de nodos que se van a gestionar. En general, los precios de Puppet pueden variar significativamente según las necesidades específicas de cada organización.

## **1.2. VENTAJAS E INCONVENIENTES DE PUPPET**

Puppet es una herramienta de gestión de configuración y automatización de infraestructuras de TI, y como tal, tiene una serie de ventajas e inconvenientes en comparación con otros productos similares. A continuación, se presentan algunas de las principales ventajas e inconvenientes de Puppet.

### **1.2.1. VENTAJAS**

— ***Declarativo***: Puppet permite a los administradores de sistemas describir la configuración de sus sistemas de manera declarativa, lo que significa que especifican cómo deberían ser los sistemas en lugar de como configurarlos. Esto hace que ***la gestión de la configuración sea más fácil y menos propensa a errores***.

— ***Escalabilidad***: Puppet es altamente escalable y puede ser utilizado para gestionar una gran cantidad de nodos en una infraestructura de TI, lo que lo hace ideal para entornos empresariales.

— ***Multiplataforma***: Puppet es compatible con una amplia variedad de sistemas operativos, incluyendo Linux, Unix y Windows.

### **1.2.2. INCONVENIENTES**

— **Curva de aprendizaje**: Puppet tiene una curva de aprendizaje empinada y puede llevar tiempo para que los usuarios aprendan a utilizarlo eficazmente.

— **Configuración inicial**: La configuración inicial de Puppet puede ser compleja y puede requerir cierta experiencia en la configuración de sistemas.

— **Costo**: Puppet Enterprise, la versión empresarial de Puppet, tiene un costo, lo que puede ser un inconveniente para algunas organizaciones.

— **Enfoque único**: Puppet tiene un enfoque único de la gestión de la configuración, lo que significa que puede no ser adecuado para todas las organizaciones y puede requerir la implementación de otras herramientas para la gestión de tareas específicas.

### **1.2.3. COMPARACIÓN CON OTROS SERVICIOS**

En comparación con Chef, Puppet tiene una sintaxis más clara y fácil de entender, lo que la hace más adecuada para usuarios nuevos o inexpertos. Sin embargo, Chef es más adecuado para entornos muy grandes y complejos.

En comparación con Ansible, Puppet es más adecuada para entornos con una infraestructura más compleja y heterogénea, aunque Ansible es más fácil de aprender y de usar para usuarios nuevos.

## **1.3. FORMAS DE ACCESO A PUPPET**

Existen varias formas de acceder al servicio Puppet, dependiendo de cómo esté configurado y utilizado en su entorno de TI. A continuación, se describen algunas de las formas más comunes de acceder al servicio Puppet:

— ***Consola web***: La consola web de Puppet proporciona una interfaz gráfica de usuario (GUI) para la gestión de la configuración y el monitoreo de los nodos. Los usuarios pueden acceder a la consola web a través de un navegador web utilizando la URL y las credenciales de inicio de sesión apropiadas.

— ***Línea de comandos***: Será la que utilicemos en la demo y que consiste en ser utilizado a través de la línea de comandos (CLI) utilizando el comando "puppet" y sus subcomandos. Esto permite a los usuarios realizar tareas de gestión de configuración y automatización de forma programática.

— ***API*** ***REST***: Puppet proporciona una API REST que permite a los usuarios interactuar con el servicio de Puppet a través de solicitudes HTTP. Esto puede ser útil para la integración con otras herramientas o sistemas.

— ***Herramientas de terceros***: Existen varias herramientas de terceros que pueden utilizarse para acceder y gestionar el servicio Puppet, como Foreman y Puppet Explorer.

### **1.3.1. ACCESO A PUPPET CON ENDPOINT TCP**

Se puede acceder a Puppet a través de endpoints TCP. Puppet utiliza una arquitectura cliente-servidor, donde el servidor de **Puppet se ejecuta en un nodo central** y **los clientes** (los nodos que se están gestionando) **se conectan al servidor para obtener la configuración y recibir actualizaciones**.

La comunicación entre el servidor de Puppet y los clientes se realiza a través de endpoints TCP ***utilizando el protocolo HTTPS***. El servidor de Puppet escucha en el ***puerto 8140*** por defecto, mientras que los clientes se conectan a través de este puerto para obtener su configuración y recibir actualizaciones.

Es importante tener en cuenta que la configuración de seguridad de Puppet puede ser bastante compleja y puede requerir la configuración de certificados y autenticación para garantizar la seguridad de la comunicación entre el servidor y los clientes.

### **1.3.2. ACCESO A PUPPET CON SOCKET UNIX**

Los sockets Unix son una forma común de comunicación interproceso en sistemas Unix/Linux, y pueden ser utilizados por Puppet para la comunicación entre el servidor y los clientes en entornos donde la comunicación a través de TCP no es posible o deseable.

Para utilizar sockets Unix en Puppet, **se debe especificar la ubicación del socket en la configuración de Puppet del servidor y de los clientes**. Por defecto, Puppet utiliza endpoints TCP, pero se puede configurar para utilizar sockets Unix en su lugar.

# **2. MONITORIZACIÓN DE PUPPET**

Hay varias formas de monitorizar la actividad de Puppet. A continuación se describen algunas de las formas más comunes de monitorizar la actividad de Puppet:

— ***Consola web***: La consola web de Puppet proporciona una interfaz gráfica de usuario (GUI) para la gestión de la configuración y el monitoreo de los nodos.

— ***Registro de eventos***: Puppet genera registros de eventos que contienen información detallada sobre las acciones que realiza en los nodos administrados. Estos registros de eventos pueden ser utilizados para monitorizar la actividad de Puppet y solucionar problemas de configuración.

— **Monitorización de recursos**: Puppet proporciona una herramienta llamada *Puppet* *Resource* que permite a los usuarios ver y gestionar los recursos configurados en los nodos.

— ***Monitoreo de métricas***: Puppet también proporciona una interfaz para monitorear las métricas de rendimiento de los nodos. Los usuarios pueden utilizar herramientas de monitoreo como Nagios o Grafana para visualizar y analizar estas métricas y tomar medidas preventivas para optimizar el rendimiento de Puppet.

## **2.1. MONITORIZACION DE PUPPET CON FACTER**

Facter es una herramienta que se utiliza en conjunto con Puppet para **obtener información del sistema operativo y otros datos de los nodos gestionados**. Facter proporciona una forma de acceder a los datos del sistema operativo y otros detalles importantes sobre los nodos de forma programática, lo que lo convierte en una herramienta valiosa para la monitorización de Puppet.

Para la monitorización de Puppet con Facter, se pueden **utilizar herramientas de visualización como Grafana o Kibana** para analizar los datos recopilados por Facter y mostrar información sobre el estado de los nodos y su rendimiento.

Es importante tener en cuenta que Facter debe ser configurado y utilizado de acuerdo con las mejores prácticas de seguridad y privacidad en su entorno, para garantizar que la información del sistema operativo y otros datos sensibles se manejen de manera segura y responsable.

## **2.2. TIPOS DE EVENTOS O NOTIFICACIONES DE PUPPET**

En Puppet, es posible capturar eventos y realizar acciones en función de ellos utilizando los denominados "**Event Handlers**". Un ***Event Handler*** es una sección de código en un manifiesto de Puppet que se ejecuta cuando se produce un evento específico.

Supongamos que queremos realizar una acción cada vez que un archivo de configuración de un servicio de configuración de un servicio es modificado. En este caso, podemos utilizar el evento "***File[<ruta\_del\_archivo>]/content***" para capturar ese evento.

Para ello, podemos crear un **Event Handler** en nuestro manifiesto de la siguiente manera:

event { **'config\_file\_change'**:

subscribe => **File**[**'/ruta/al/archivo/config'**],

refreshonly => **true**,

notify => **Exec**['mi\_accion']

}

exec { **'mi\_accion'**:

command => **'/ruta/a/mi\_script.sh'**,

refreshonly => **true**,

}

El recurso "***Exec['mi\_accion']***" es una acción que se ejecutará cuando se produzca el evento. En este caso, se trata de la ejecución de un script que realiza la acción deseada.

Con esta configuración, cada vez que se produzca un cambio en el contenido del archivo "**/ruta/al/archivo/config",** se ejecutará el script "***/ruta/a/mi\_script.sh***".

A continuación, se describen algunos de los tipos de eventos y notificaciones más comunes de Puppet:

— **Cambios en la configuración**: Puppet puede generar eventos y notificaciones cuando se realizan cambios en la configuración de los nodos gestionados. Esto puede incluir la instalación o actualización de paquetes, la creación o modificación de archivos de configuración y otros cambios en la configuración del sistema.

— **Fallos en la configuración**: Cuando Puppet encuentra problemas durante la aplicación de la configuración en un nodo, puede generar eventos y notificaciones para informar sobre el error y proporcionar información detallada sobre el problema.

— **Estado de los nodos**: Puppet puede generar eventos y notificaciones para informar sobre el estado de los nodos gestionados. Esto puede incluir información sobre el estado de los servicios, el uso de recursos y otros detalles importantes sobre el estado del sistema.

— **Problemas de conectividad**: Puppet puede generar eventos y notificaciones cuando se producen problemas de conectividad entre los nodos y el servidor de Puppet. Esto puede incluir problemas de comunicación entre el servidor de Puppet y los agentes de los nodos gestionados.

— **Problemas de seguridad**: Puppet puede generar eventos y notificaciones para informar sobre los problemas de seguridad en el entorno. Esto puede incluir alertas sobre vulnerabilidades de seguridad conocidas o intentos de acceso no autorizados al sistema.

# **3. GESTIÓN DE PUPPET**

Para ***detener el servicio*** de Puppet, puede utilizar el siguiente comando en la línea de comandos de su sistema operativo:

**sudo service puppet stop**

Para ***reiniciar el servicio de Puppet***, puede utilizar el siguiente comando:

**sudo service puppet restart**

Para ***realizar un cambio en la configuración de Puppet***, primero debe editar el archivo de configuración de Puppet, que generalmente se encuentra en ***/etc/puppetlabs/puppet/puppet.conf***. Puede utilizar un editor de texto como nano o vim para editar el archivo. Por ejemplo:

**sudo nano /etc/puppetlabs/puppet/puppet.conf**

Realice los cambios necesarios en el archivo de configuración de Puppet y guárdelos. Luego, debe aplicar los cambios a Puppet para que se actualice con la nueva configuración. Para aplicar los cambios, debe ejecutar el siguiente comando en la línea de comandos:

**sudo puppet apply /etc/puppetlabs/puppet/manifests/site.pp**

Este comando aplica la configuración de Puppet que se encuentra en el archivo site.pp ubicado en el directorio ***/etc/puppetlabs/puppet/manifests***. Si tiene otros archivos de configuración de Puppet que desea aplicar, puede cambiar la ruta del archivo en el comando.

También puede utilizar el siguiente comando para aplicar la configuración de Puppet en todos los nodos gestionados por Puppet. Este comando fuerza la aplicación de la configuración de Puppet en todos los nodos gestionados por Puppet.

**sudo puppet agent -t**

# **4. REDUNDANCIA Y BALANCEO DE CARGA EN PUPPET**

Puppet es capaz de soportar la redundancia para garantizar la disponibilidad continua del servicio incluso si un nodo o servidor falla. Para lograr la redundancia, se pueden configurar varios servidores de Puppet en un clúster de alta disponibilidad.

La forma más común de lograr la redundancia de Puppet es mediante el **uso de un balanceador de carga que distribuya las solicitudes de los clientes a varios servidores de Puppet en el clúster**. Los clientes se conectan al balanceador de carga en lugar de a un servidor de Puppet específico, lo que permite que el tráfico se distribuya de manera uniforme entre los nodos.

Además, Puppet también admite la replicación de bases de datos y la sincronización de archivos entre los nodos de Puppet en el clúster para garantizar que la configuración y otros datos se mantengan consistentes en todos los nodos.

## **4.1. CÓMO USAR EL BALANCEADOR DE CARGA**

Para configurar los clientes de Puppet para conectarse al balanceador de carga en lugar de a un servidor de Puppet específico, siga los siguientes pasos:

1. Abra el archivo de configuración de Puppet en el cliente. El archivo se encuentra generalmente en la ruta **/etc/puppetlabs/puppet/puppet.conf**.

2. Busque la sección "[main]" en el archivo de configuración y agregue la siguiente línea:

**server = nombre\_o\_direccion\_IP\_del\_balanceador\_de\_carga**

Reemplace "*nombre\_o\_direccion\_IP\_del\_balanceador\_de\_carga*" con el nombre de host o la dirección IP del balanceador de carga que ha configurado.

3. Guarde el archivo de configuración y salga del editor.

4. Reinicie el servicio de Puppet en el cliente para que la nueva configuración tenga efecto. En sistemas Linux, puede hacerlo ejecutando el siguiente comando:

**systemctl restart puppet**

Una vez que haya configurado los clientes de Puppet para conectarse al balanceador de carga en lugar de a un servidor de Puppet específico, los clientes se conectarán al balanceador de carga, que distribuirá las solicitudes de los clientes entre los nodos del clúster de alta disponibilidad configurados.

# **5. LENGUAJE EN PUPPET**

El lenguaje de Puppet es una **sintaxis declarativa** que permite definir la configuración de un servidor o dispositivo de red, y Puppet se encarga de aplicar automáticamente esa configuración en todos los sistemas que se encuentren bajo su gestión.

Puppet utiliza una **estructura de bloques y recursos para definir la configuración** deseada. Los recursos representan elementos de configuración específicos, como paquetes de software, archivos de configuración, servicios, usuarios, grupos, etc. Los bloques permiten agrupar recursos y aplicarlos a diferentes sistemas de manera coherente.

**Puppet también utiliza variables y plantillas para permitir una mayor flexibilidad y adaptabilidad en la definición de configuraciones.** Las variables se utilizan para definir valores que pueden ser reutilizados en diferentes partes del código de Puppet, mientras que las plantillas permiten la creación de archivos de configuración dinámicos que pueden variar según la configuración de cada servidor.

## **5.1. MÓDULOS**

Son paquetes de código reutilizable que contienen clases, recursos y otros componentes necesarios para configurar un sistema específico. Los módulos se organizan en directorios y se pueden compartir y reutilizar entre diferentes sistemas y equipos.

**mynginx/**

**├── files/**

**│ ├── nginx.conf**

**│ └── site.conf**

**├── manifests/**

**│ └── init.pp**

## **5.2. RECURSOS**

Los recursos son la unidad básica de configuración en Puppet. Representan elementos de configuración específicos, como archivos, paquetes de software, usuarios, grupos, servicios, etc.

Package { ‘vim’:

ensure => installed,

}

Ejemplo de un recurso para instalar la herramienta “vim” en un servidor:

## **5.3. VARIABLES**

Las variables se utilizan para definir valores que pueden ser reutilizados en diferentes partes del código de Puppet. Las variables se definen con un nombre y un valor y se pueden utilizar en recursos, clases y otros componentes de Puppet.

$user\_name = 'Fernando'

user { $user\_name:

ensure => present,

}

## **5.4. MANIFESTOS**

Los manifestos son archivos que contienen código Puppet y que se utilizan para definir la configuración deseada de un sistema. Los manifestos pueden contener clases, recursos, variables, plantillas y otros componentes de Puppet.

Ejemplo de un manifest para instalar y configurar el servidor web "nginx" en un sistema utilizando todo lo anterior:

class nginx {

package { 'nginx':

ensure => installed,

}

service { 'nginx':

ensure => running,

enable => true,

}

file { '/etc/nginx/conf.d/my\_site.conf':

ensure => file,

content => template('nginx/site.conf.erb'),

}

}

# **6. COMPONENTES**

**—El Puppet Agent:** es el componente del software de gestión de configuración que se instala en los sistemas que se van a configurar y que se comunica con el servidor de Puppet para recibir y aplicar la configuración definida en los manifestos. Se ejecuta en segundo plano en los sistemas y realiza comprobaciones periódicas para ver si hay actualizaciones de la configuración.

**— *Puppet Master:***es el servidor centralizado que aloja y distribuye la configuración a los agentes de Puppet. El servidor de Puppet se encarga de compilar y entregar los manifestos de Puppet a los agentes.

**— *PuppetDB*:** donde se guardará la información de los nodos que se van a configurar y estadísticas y reportes del estado de los nodos. Ejecuciones que han fallado, porque, configuraciones.

**— *Facter:*** es una herramienta instalada en los clientes que recoge parámetros del sistema, como CPU, memoria, dirección IP, llamados ‘*facts’*. Usados para determinar que configuración se va a utilizar en cada nodo.

**— *Hiera:***es una base de datos clave valor que se configura en ficheros *.yml* que se utiliza principalmente para almacenar información sensible como *contraseñas*, claves, certificados ocultando dicha información.

# **7. CONCLUSIÓN**

Puppet es una gran elección como herramienta de automatización de configuración y administración de sistemas que permite a los usuarios definir y gestionar configuraciones de forma fácil y eficiente. Es altamente escalable y puede gestionar miles de servidores y dispositivos de manera simultánea, lo que lo hace adecuado para empresas de todos los tamaños.

# **8. BIBLIOGRAFIA**

<https://www.puppet.com/docs/puppet/>

[Cómo configurar Puppet Master y el Agente en CentOS 8 – HowtoForge](https://howtoforge.es/como-configurar-puppet-master-y-el-agente-en-centos-8/)

[(121) Paradigma Digital - Puppet como gestor de configuraciones – YouTube](https://www.youtube.com/watch?v=zKqoKHUxIPc&t=2378s)

[(121) Instalación y primeras configuraciones de Puppet - YouTube](https://www.youtube.com/watch?v=dZRzqzdlCI8&t=549s)