



UCLM – E.S. de Informática

Ingeniería en Informática

GESTIÓN Y ADMINISTRACIÓN DE REDES

EVENTOS CON PUPPET



Autores:

Andrés Castellanos Cantos

Felipe Segovia Friginal

Contenido

1. Configuración Bot de Telegram	3
2. Posible solución utilizando Facter.....	3
3. Solución utilizando Hiera	3
3.1. Configuración de Hiera	3
3.2. Configuración de los nodos.....	4
3.3. Configuración del manifiesto y ejecución del evento.....	4
3.4. Descargar perfil y notificar evento al bot	5

1. Configuración Bot de Telegram

Para la configuración del bot de Telegram hemos utilizado el repositorio del profesor Tobías [ptobiasdiaz/tbnotify: Telegram Bot notifier \(github.com\)](https://github.com/ptobiasdiaz/tbnotify), donde se pueden seguir los pasos para arrancar el servidor y configurar el bot.

El bot que hemos utilizado es un bot que nos proporciona Telegram llamado Bot_Father que nos permite crear varios bots utilizando comandos para su configuración.

2. Posible solución utilizando Facter

Facter es una herramienta de Puppet, es un sistema de administración de configuración, para recopilar información sobre los nodos y guardarla como variables en el servidor Puppet. Estas variables se utilizan posteriormente para configurar y personalizar los nodos de manera automática.

Facter se ejecuta en cada nodo y recopila información sobre el hardware, el sistema operativo, la red y otros datos relevantes. Esta información se extrae del sistema en tiempo de ejecución y se almacena en variables de facter, que luego se envían al servidor Puppet.

La información recopilada por Facter puede ser muy variada, desde el nombre del nodo y la dirección IP, hasta el tipo de CPU, la cantidad de memoria RAM o el espacio en disco disponible. Facter también puede ser extendido para recopilar información adicional específica del entorno de cada nodo.

Hemos probado a instalar Facter en los nodos clientes y que recopilen información sobre los nodos para enviárselos al servidor para luego poder ejecutar el evento con la información recolectada, pero nos daba problemas al recolectar dicha información.

3. Solución utilizando Hiera

Hiera es un componente de Puppet que se utiliza para separar la configuración de los nodos del código de configuración. Proporciona una forma flexible de definir y gestionar datos específicos de los nodos, como variables, parámetros y configuraciones, de manera independiente al código de configuración de Puppet.

Pasos a seguir:

3.1. Configuración de Hiera

La configuración de Hiera la vamos a realizar en el archivo hiera.yaml del directorio :
/etc/puppetlabs/puppet

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
root@puppetmaster:/etc/puppetlabs/puppet# ls
auth.conf  devices  hiera.yaml  puppet.conf  reports  ssl
root@puppetmaster:/etc/puppetlabs/puppet# _
```

```

root@puppetmaster:/etc/puppetlabs/puppet# cat hiera.yaml
---
# Hiera 5 Global configuration file

version: 5

defaults:
  datadir: data
  data_hash: yaml_yaml

hierarchy:
  - name: "Variables globales"
    path: "node.yaml"
    lookup_options:
      strategy: "first"
#   data_hash: yaml_data

```

Esta es una posible configuración de Hiera donde lo más importante es definir el “datadir” donde se situará el archivo de configuración de los nodos y en “path” el archivo de configuración de los nodos. Además de otras configuraciones como el “data_hash” que es el tipo de archivo la estrategia de búsqueda de las variables de los nodos en “strategy”.

3.2. Configuración de los nodos

```

root@puppetmaster:/etc/puppetlabs/code/data# cat node.yaml
client1:
  ipaddress: 192.168.101
  roles:
    - webserver
    - database

client2:
  ipaddress: 192.168.1.22
  roles:
    - webserver

```

Ahora nos vamos al directorio y el archivo que hemos definido antes en la configuración de Hiera y vamos a definir las características de los nodos.

En nuestro caso vamos a configurar el nodo client2 que va a tener como ip “192.168.1.22” y el rol “webserver” ya que en él vamos a instalar el módulo ya explicado en la entrega anterior “lemp” que es un servidor web utilizando nginx y sql.

3.3. Configuración del manifiesto y ejecución del evento

Vamos al directorio donde tenemos definido el manifiesto que va a enviar al nodo

```

root@puppetmaster:/etc/puppetlabs/code/environments/production/manifests# ls
site.pp

```

```
node client2 {
  include lemp

  exec { 'tbnotify_send':
    command => '/bin/bash -c "tbnotify_send \"Configurados servicios en Nodo Cliente2 con ip: 192.168.1.22 con rol: webservice\""'
  }
}
```

Y vamos a definir el perfil del nodo client2, que es donde tenemos configurado el bot de Telegram, en el que vamos a incluir el módulo de lemp para que se ejecute el servicio web en el nodo y vamos a crear una función “tbnotify_send” de tipo exec que nos permitirá que cuando el nodo se descargue el perfil se ejecutara el comando que esté definido dentro de “command” en nuestro caso es el que aparece en la captura.

3.4. Descargar perfil y notificar evento al bot

Para traer el perfil definido en el manifiesto del PuppetMaster, nos situamos en la máquina del cliente y ejecutamos el siguiente comando:

```
felipe@felipe-VirtualBox:~$ sudo /opt/puppetlabs/bin/puppet agent --test
[sudo] contraseña para felipe:
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for client2
Info: Applying configuration version '1683731548'
Notice: /Stage[main]/Main/Node[client2]/Exec[tbnotify_send]/returns: executed successfully (corrective)
Notice: Applied catalog in 0.39 seconds
```

Y este sería el mensaje recibido por el bot de Telegram.

