

## Puppet

Puppet es otra muy buena herramienta que nos ayuda en los despliegues automatizados.

El potencial de la herramienta radica en su uso cliente-servidor y su entendimiento y uso es más complicado que la herramienta antes analizada que fue ansible.

Para este laboratorio utilizamos dos ambientes el primero con un server puppet master que administra dos nodos y realiza una instalación sencilla del LAMP y el segundo ambiente que sería la extensión fue una instalación Standalone que realiza la instalación básica de Wordpress

### 1) Puppet Master

La autenticación en este caso se realiza por medio de certificados por lo cual es importante para la conexión correcta que la resolución de nombre se realice correctamente; para esto configuramos los dominios de las máquinas en el archivo host

```
192.168.1.8 puppet
127.0.1.1 web

jdominguex@jdominguex:~/puppet-standalone$ vim /etc/hosts
jdominguex@jdominguex:~/puppet-standalone$ ping web
PING web (192.168.1.17) 56(84) bytes of data.
64 bytes from web (192.168.1.17): icmp_seq=1 ttl=64 time=0.266 ms
64 bytes from web (192.168.1.17): icmp_seq=2 ttl=64 time=0.465 ms
64 bytes from web (192.168.1.17): icmp_seq=3 ttl=64 time=0.477 ms
```

Con esta configuración y puppet Agent instalado en los nodos la generación de certificados será correcta

- Para esta configuración sencilla de lamp utilizaremos un manifiesto sencillo con la configuración. (archivo completo en el adjunto)

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

```
# execute 'apt-get update'
exec { 'apt-update':
  command => '/usr/bin/apt-get update' # exec resource named 'apt-update'
                                     # command this resource will run
}

# install apache2 package
package { 'apache2':
  require => Exec['apt-update'], # require 'apt-update' before installing
  ensure => installed,
}

# ensure apache2 service is running
service { 'apache2':
  ensure => running,
}

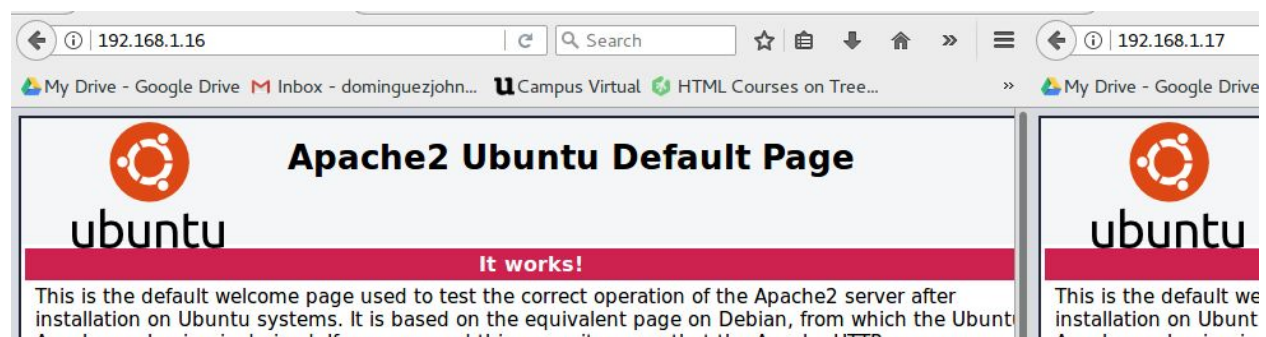
# install mysql-server package
package { 'mysql-server':
  require => Exec['apt-update'], # require 'apt-update' before installing
  ensure => installed,
}
```

El agente de puppet cada 30 minutos revisa los cambios realizados en los manifiestos y realiza los cambios automáticamente si de esta manera está configurado.

En esta ocasión realizaremos el proceso manual en uno de los servidores para revisar los resultados.

```
ubuntu@web:~$ sudo /opt/puppetlabs/bin/puppet agent --test
2017-05-30 01:12:12.813383 WARN puppetlabs.facter - locale environment variables
ere bad; continuing with LANG=C LC_ALL=C
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for web
Info: Applying configuration version '1496106737'
Notice: /Stage[main]/Main/Exec[apt-update]/returns: executed successfully
```

Luego de pasar los 30 minutos observamos la instalación en los servidores configurados



## 2) Instalación Standalone

La variación más importante y que se quería resaltar en esta instalación con el uso de los módulos que es un conjunto de manifiestos que realizan una acción ya establecida en esta ocasión la instalación de Wordpress.

```
vagrant@vagrant-ubuntu-trusty-64:~/modulos$ sudo puppet module list
/etc/puppet/modules
├── do-wordpress (v0.1.0)
├── puppet-staging (v2.2.0)
├── puppetlabs-apache (v1.11.0)
├── puppetlabs-concat (v2.2.1)
├── puppetlabs-mysql (v3.11.0)
└── puppetlabs-stdlib (v4.17.0)
vagrant@vagrant-ubuntu-trusty-64:~/modulos$
```

Tomando como base los módulos de Apache y mysql creamos e instalamos uno nuevo llamado do-wordpress el cual tendrá todos sus componentes en el archivo adjunto.

Ahora vamos a correr el módulo y observar los resultados

```
vagrant@vagrant-ubuntu-trusty-64:~/modulos$ sudo puppet apply /tmp/install-wp.pp
Warning: Could not retrieve fact fqdn
Warning: Config file /etc/puppet/hiera.yaml not found, using Hiera defaults
Notice: Compiled catalog for vagrant-ubuntu-trusty-64 in environment production in 2.40 seconds
Notice: MySQL Installation Complete
Notice: /Stage[main]/Wordpress/Notify[MySQL Installation Complete]/message: defined 'message' as 'MySQL Installation Complete'
Notice: Apache Installation Complete
Notice: /Stage[main]/Wordpress/Notify[Apache Installation Complete]/message: defined 'message' as 'Apache Installation Complete'
Notice: /Stage[main]/Wordpress::Wp/File[/tmp/latest.tar.gz]/ensure: defined content as '{md5}ea43ed518a64f398f1562889e38b26c8'
Notice: /Stage[main]/Wordpress::Wp/Exec[extract]/returns: executed successfully
Notice: /Stage[main]/Wordpress::Wp/Exec[copy]/returns: executed successfully
Notice: Wordpress Installation Complete
Notice: /Stage[main]/Wordpress/Notify[Wordpress Installation Complete]/message: defined 'message' as 'Wordpress Installation Complete'
Notice: Finished catalog run in 1.50 seconds
```

Como pudimos observar tiene una gran cantidad de plugins y herramientas que la hacen una herramienta muy potente para nuestra infraestructura, aunque su complejidad es un poco mas avanzada vale la pena invertir el tiempo en su conocimiento y comenzar a utilizarla.

Se utilizó como base los artículos de Digital Ocean:

<https://www.digitalocean.com/community/tutorials/how-to-create-a-puppet-module-to-automate-wordpress-installation-on-ubuntu-14-04>

<https://www.digitalocean.com/community/tutorials/how-to-install-puppet-4-in-a-master-agent-set-up-on-ubuntu-14-04>

<https://www.digitalocean.com/community/tutorials/how-to-install-puppet-4-on-ubuntu-16-04>