

Guía Práctica DevOps: Puppet + Vagrant para desplegar un LAMP en agente1


¿Qué aprenderás?

- Levantar un entorno virtual con Vagrant y VirtualBox en Windows
 - Configurar un Puppet Server y dos Agentes
 - Crear un módulo Puppet desde cero para instalar un servidor LAMP
 - Comparar con la estructura equivalente en Ansible
-

1 Vagrantfile con 3 máquinas (manager, agente1, agente2)

```
# Vagrantfile
Vagrant.configure("2") do |config|
  nodes = [
    { :hostname => "manager", :ip => "192.168.33.10", :mem => 2048 },
    { :hostname => "agente1", :ip => "192.168.33.11", :mem => 1024 },
    { :hostname => "agente2", :ip => "192.168.33.12", :mem => 1024 }
  ]

  nodes.each do |node|
    config.vm.define node[:hostname] do |node_config|
      node_config.vm.box = "bento/ubuntu-22.04"
      node_config.vm.hostname = node[:hostname]
      node_config.vm.network "private_network", ip: node[:ip]
      node_config.vm.provider "virtualbox" do |vb|
        vb.memory = node[:mem]
      end
      node_config.vm.provision "shell", inline: <<-SHELL
        apt-get update
        SHELL
      end
    end
  end
end
```

 Ejecuta `vagrant up` para levantar las máquinas. Usa `vagrant ssh manager` para conectarte al nodo correspondiente.

2 Instalación de Puppet Server en manager

```
# En manager:
sudo apt update
sudo apt install -y wget
wget https://apt.puppet.com/puppet7-release-jammy.deb
sudo dpkg -i puppet7-release-jammy.deb
sudo apt update
sudo apt install -y puppetserver
```

Configura memoria del servidor Puppet:

Edita el archivo:

```
sudo nano /etc/default/puppetserver
```

Cambia:

```
JAVA_ARGS="-Xms512m -Xmx512m"
```

Configura hostname en /etc/hosts

Agrega en cada VM: (agregar resolución de puppet para evitar modificar el nombre del server predeterminado → permisos con sudo en /etc/puppetlabs/puppetserver)

```
192.168.33.10 manager
192.168.33.10 puppet
192.168.33.11 agente1
192.168.33.12 agente2
```

Inicia el servidor Puppet:

```
sudo systemctl start puppetserver
sudo systemctl enable puppetserver
```


Instalación del Agente Puppet en agente1 y agente2

```
# En cada agente:
wget https://apt.puppet.com/puppet7-release-jammy.deb
sudo dpkg -i puppet7-release-jammy.deb
sudo apt update
sudo apt install -y puppet-agent
```

```
# Configura el agente:
sudo nano /etc/puppetlabs/puppet/puppet.conf
```

Agrega:

```
[main]
server = manager
```

 Asegúrate de que los nombres resuelvan correctamente en /etc/hosts.

Solicita el certificado:

```
sudo /opt/puppetlabs/bin/puppet agent --test
```

En el manager, firma los certificados:

```
sudo /opt/puppetlabs/bin/puppetserver ca list
sudo /opt/puppetlabs/bin/puppetserver ca sign --all
```

4 Verifica la comunicación

Desde los agentes:

```
sudo /opt/puppetlabs/bin/puppet agent --test
```

5 ¿Qué es un módulo, receta y manifest en Puppet?

- **Módulo:** Estructura que organiza recursos Puppet (como roles en Ansible).
 - **Receta:** Archivo `.pp` donde se definen tareas (como los tasks en Ansible).
 - **Manifest:** El código que Puppet ejecuta, ubicado por defecto en `manifests`.
-

6 Crear un módulo llamado `lamp_stack`

```
# En manager:
cd /etc/puppetlabs/code/environments/production/modules
sudo mkdir -p lamp_stack/{manifests,files,templates}
```

Explicación:

- `manifests/`: Aquí va el archivo `init.pp` con la receta principal.
- `files/`: (opcional) Archivos que puedes copiar al host desde Puppet.
- `templates/`: (opcional) Archivos `.erb` si quieres configurar plantillas.

Edita el archivo:

```
lamp_stack/manifests/init.pp
```

```
# lamp_stack/manifests/init.pp

class lamp_stack {
  package { ['apache2', 'mysql-server', 'php', 'libapache2-mod-php']:
    ensure => installed,
  }

  service { 'apache2':
    ensure => running,
    enable => true,
    require => Package['apache2'],
  }
}
```

```
}  
  
service { 'mysql':  
  ensure => running,  
  enable => true,  
  require => Package['mysql-server'],  
}  
}
```

7 Asignar el módulo a los agentes

Edita /etc/puppetlabs/code/environments/production/manifests/site.pp

```
# site.pp  
node 'agente1' {  
  include lamp_stack  
}  
  
node 'agente2' {  
  include lamp_stack  
}
```

8 Aplicar la configuración

MODO 1:

1. Instalar en manager: `sudo apt install puppet-bolt`

Crear fichero de inventario para bolt (nodos.yml)

targets:

- name: agente1

uri: 192.168.33.11

config:

transport: ssh

ssh:

user: vagrant

password: vagrant

host-key-check: false

- name: agente2

uri: 192.168.33.12

config:

transport: ssh

ssh:

user: vagrant

password: vagrant

host-key-check: false

2. Ejecutar puppet agent de manera remota:
bolt command run 'sudo /opt/puppetlabs/bin/puppet agent --test' --targets all --inventoryfile ./nodos.yml

MODO 2

En agente1:

```
sudo /opt/puppetlabs/bin/puppet agent --test
```

✅ Esto instalará y levantará Apache, MySQL y PHP en agente1.

9 Equivalente en Ansible (comparación)

lamp.yml - Ejemplo Ansible Playbook

```
- hosts: agente1, agente2
  become: true
  tasks:
    - name: Instalar paquetes LAMP
      apt:
        name: ['apache2', 'mysql-server', 'php', 'libapache2-mod-php']
        state: present
        update_cache: yes

    - name: Iniciar apache
      service:
        name: apache2
        state: started
        enabled: true

    - name: Iniciar mysql
      service:
        name: mysql
        state: started
        enabled: true
```

vs Comparación clave:

Puppet

Ansible

Manifests (.pp) Playbooks (.yml)

Puppet	Ansible
Módulos	Roles
Site.pp	Inventory + Main playbook
Declarativo	Procedimental



Conclusión

Este laboratorio te da un entorno real para practicar Puppet como alternativa a Ansible. Te muestra el flujo completo: **desde la infraestructura hasta la aplicación**, con ejemplos comparables entre herramientas. Ideal para proyectos reales o exámenes de DevOps.