



Guía completa: Instalación de Chef + LAMP con Vagrant en Windows

◆ 1. Arquitectura del Proyecto

Vas a crear un entorno de pruebas local con 3 máquinas virtuales usando **Vagrant** + **VirtualBox** desde un sistema Windows:

Máquina	Rol	Hostname	Función principal
manager	Workstation	manager	Instala y gestiona Chef, ejecuta knife
agente1	Nodo	agente1	Ejecuta recetas Chef (LAMP aquí)
agente2	Nodo	agente2	Ejecuta recetas Chef (no usado en este ejemplo)

La ejecución con knife necesita un servidor de **Chef Infra Server (Enterprise)** con requerimientos de suscripciones y costo basado en número de agentes. SOLO USAREMOS LA PARTE COMUNITARIA.

◆ 2. Creación del Vagrantfile

Este archivo define tus 3 máquinas. Crea una carpeta vacía (por ejemplo chef - lab) y dentro un archivo llamado `Vagrantfile` con este contenido:

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/ubuntu-22.04"

  # Manager (Workstation)
  config.vm.define "manager" do |manager|
    manager.vm.hostname = "manager"
    manager.vm.network "private_network", ip: "192.168.33.10"
    manager.vm.provider "virtualbox" do |vb|
      vb.name = "manager"
      vb.memory = 1024
      vb.cpus = 1
    end
  end

  # Agente 1
  config.vm.define "agente1" do |agente1|
    agente1.vm.hostname = "agente1"
    agente1.vm.network "private_network", ip: "192.168.33.11"
    agente1.vm.provider "virtualbox" do |vb|
      vb.name = "agente1"
      vb.memory = 1024
      vb.cpus = 1
    end
  end

  # Agente 2
  config.vm.define "agente2" do |agente2|
    agente2.vm.hostname = "agente2"
```

```

    agente2.vm.network "private_network", ip: "192.168.33.12"
    agente2.vm.provider "virtualbox" do |vb|
      vb.name = "agente2"
      vb.memory = 1024
      vb.cpus = 1
    end
  end
end

```

Notas:

- Cada máquina tiene su IP fija.
- La box bento/ubuntu-22.04 es estable y compatible con Chef.
- Puedes ejecutar todas con: `vagrant up`

◆ **3. ¿Qué es Chef? ¿Cómo se compara con Ansible?**

Concepto	Chef	Ansible
Modelo	<i>Pull (cliente ejecuta recetas)</i>	Push (control central)
Lenguaje	Ruby DSL	YAML
Estado	Deseado	Deseado
Módulo central	Chef Server / Workstation	Controlador con SSH
Ejecución local	<code>chef-client</code>	<code>ansible-playbook</code>
◆ Receta en Chef = conjunto de instrucciones (como un <i>task</i> o <i>module</i> en Ansible).		
◆ Cookbook = colección de recetas organizadas con metadatos.		

◆ **4. Instalación de Chef Workstation en manager**

Entra a la máquina con:

```
vagrant ssh manager
```

Luego instala Chef Workstation:

```
curl -L https://omnitruck.chef.io/install.sh | sudo bash -s -- -P chef-workstation
```

Verifica la instalación:

```
chef -v
```

◆ 5. Preparar entorno Chef y registrar nodos

1. Inicializa tu entorno de trabajo:

```
mkdir ~/chef-repo
cd ~/chef-repo
chef generate repo .
```

2. Crea claves para autenticarte:

```
mkdir ~/.chef
cd ~/.chef
openssl genrsa -out chef_user.pem 2048
```

3. Crea un archivo de configuración knife.rb: (recordad que vi es nuestro amigo)

```
nano ~/.chef/knife.rb
```

Agrega lo siguiente:

```
current_dir = File.dirname(__FILE__)
log_level          :info
log_location       STDOUT
node_name          'chef_user'
client_key          "#{current_dir}/chef_user.pem"
chef_server_url     'http://localhost:8889'
cookbook_path       ["#{current_dir}/../cookbooks"]
```

◆ 6. Crear cookbook LAMP

En el directorio de tu repo:

```
cd ~/chef-repo/cookbooks
chef generate cookbook lamp_stack
cd lamp_stack
```

recipes/default.rb

```
# Instala Apache, MySQL y PHP
package 'apache2'
package 'mysql-server'
package 'php'
package 'libapache2-mod-php'
package 'php-mysql'

# Habilita e inicia Apache
service 'apache2' do
  action [:enable, :start]
end

# Crea un archivo index.php
```

```
file '/var/www/html/index.php' do
  content "<?php phpinfo(); ?>"
  mode '0644'
  owner 'www-data'
  group 'www-data'
end
```

metadata.rb

```
name 'lamp_stack'
maintainer 'Aula11Devops'
version '0.1.0'
```

Ejecutar cookbooks sin Chef Server (el escenario desarrollado)

♦ En tu máquina manager

1. Modifica el fichero /etc/hosts -- → 192.168.33.11 agente1:

♦ En tu máquina agente1

1. Entra a la máquina:

```
vagrant ssh agente1
```

2. Instala Chef (igual que hiciste en manager):

```
curl -L https://omnitruck.chef.io/install.sh | sudo bash -s -- -P chef
```

3. Crea esta estructura básica en agente1:

```
mkdir -p ~/chef-repo/cookbooks
cd ~/chef-repo/cookbooks
```

4. Copia el cookbook lamp_stack a ~/chef-repo/cookbooks/lamp_stack

- Puedes usar scp desde manager, o simplemente comprimirlo y moverlo usando carpeta compartida Vagrant.

```
vagrant@manager:~$ scp -r chef-repo/cookbooks/lamp_stack/
agente1:/home/vagrant/chef-repo/cookbooks/
```

6. Crea un archivo run_list.json:

```
vi run_list.json
```

Con este contenido:

```
{
  "run_list": [
    "recipe[lamp_stack]"
  ]
}
```

7. Ejecuta el cookbook en modo local: (no olvides realizar un apt update)

```
sudo chef-client -z -o 'recipe[lamp_stack]' -j run_list.json
```



◆ EXTRA. (a modo de ejemplo para la visualización de comandos si tuviéramos Chef server) Subir y ejecutar el cookbook

Desde el directorio raíz del repo y ejecuta:

```
knife bootstrap 192.168.33.11 -N agente1 --ssh-user vagrant --ssh-password vagrant --sudo --yes
```

- ◆ Esto instala `chef-client` en `agente1` y lo registra como nodo.

Luego, ejecuta:

```
knife cookbook upload lamp_stack
knife node run_list add agente1 'recipe[lamp_stack]'
knife ssh 'name:agente1' 'sudo chef-client' --manual-list --ssh-user vagrant --ssh-password vagrant
```

◆ 8. Validación

Abre tu navegador en Windows y accede a:

<http://192.168.33.11/index.php>

Deberías ver la página de `phpinfo()` 🎉



9. Recomendaciones finales

- Usa `chef-client` de forma periódica con cron si quieres aplicar cambios automáticos.
- Versiona tus cookbooks con Git.
- Chef es más potente para entornos grandes con infraestructura compleja. Para pequeños entornos, Ansible puede ser más ágil.