

Asignatura	Datos del alumno	Fecha
Herramientas de Automatización de Despliegues	Apellidos: López Baya	
	Nombre: Christian	

## Laboratorio: Instalación completa de WordPress con Ansible

1.- Primeramente, se realizó la configuración de la máquina virtual en el vagrant file, donde entre las características principales de uso la distribución de "Ubuntu/bionic64" para la actividad, también se definió una ip estática como red privada y se hizo un port forwarding para poder acceder desde nuestro host al despliegue de Wordpress. Mas abajo también se definió la memoria requerida como cuantos CPU's usaremos para la actividad

```

Vagrantfile
You, 1 second ago | 1 author (You)
1  Vagrant.configure("2") do |config|
2      config.vm.box = "ubuntu/bionic64"
3      config.vm.network "private_network", ip: "192.168.33.40"
4      config.vm.network "forwarded_port", guest: 80, host: 9000
5      config.vm.provider "virtualbox" do |vbox|
6          vbox.memory = "2048"
7          vbox.cpus = "2"
8      end

```

2.- Mas abajo se empezó a provisionar usando Shell la maquina virtual donde únicamente se instaló ANSIBLE

```

config.vm.provision "shell", inline: <<-SHELL
    sudo apt-add-repository ppa:ansible/ansible -y
    sudo apt update
    sudo apt install ansible -y
    ansible --version
SHELL

```

Asignatura	Datos del alumno	Fecha
<b>Herramientas de Automatización de Despliegues</b>	Apellidos: López Baya	
	Nombre: Christian	

3.- Por ultimo en el VAGRANTFILE se provisiono ANSIBLE con el playbook definido previamente que se llama “playbook.yml”

```
config.vm.provision "ansible_local" do |ansible|
  ansible.playbook = "playbook.yml"
end
end
```

4.- Posteriormente se realizó la configuración del archivo “playbook.yml” que es el archivo de configuración donde se tiene la definición completa para el provisionamiento y despliegue de Wordpress.

Primeramente se definió el nombre de nuestro archivo, se definió que este aprovisionara a todos los host que apunten a en(En realidad solo aprovisionaremos localmente), también marco “become=true” para que todas las acciones las realice como superusuario y por ultimo se definió variables las cuales serán usadas a lo largo de la instalación

```
! playbook.yml
You, 30 minutes ago | 1 author (You)
1 - name: Instalar WordPress
2   hosts: all
3   become: yes
4   vars:
5     db_name: wordpress
6     db_user: wordpress
7     db_password: unir12345
8     wp_path: /var/www/html/wordpress
9     wp_title: 'Ansible con Wordpress - Christian Lopez Baya'
10    wp_admin_user: unir
11    wp_admin_password: unir12345
12    wp_admin_email: unir_christian@example.com
13
```

Asignatura	Datos del alumno	Fecha
<b>Herramientas de Automatización de Despliegues</b>	Apellidos: López Baya	
	Nombre: Christian	

Posteriormente se inicio con las tareas para la instalacion, iniciando con la actualizacion de paquetes nencesarios en nuestra maquina virtual, como tambien la instalacion de librerias necesarias para Wordpress como (Apache, PHP, MYSQL) y otras librerias necesarias como UNZIP que nos ayudara a descomprimir Wordpress al descargarlo

```
tasks:
  - name: Actualizar lista de paquetes
    apt:
      update_cache: yes

  - name: Instalar paquetes necesarios
    apt:
      name: '{{ item }}'
      state: present
    loop:
      - apache2
      - php
      - libapache2-mod-php
      - php-mysql
      - php-curl
      - php-gd
      - php-mbstring
      - php-xml
      - unzip
```

Posteriormente se continuo con las tareas de descarga de Wordpress, luego se descomprimió el mismo en la ruta por defecto del servidor APACHE2, se descargó también la Wordpress CLI para poder tomar acciones como la instalación de Wordpress, en adición a eso se configuro apache para que apunte directamente como inicio al “index.php” que esta dentro de la carpeta descomprimida Wordpress y por ultimo se reinicio el servicio de APACHE2

Asignatura	Datos del alumno	Fecha
Herramientas de Automatización de Despliegues	Apellidos: López Baya	
	Nombre: Christian	

```

- name: Descargar e instalar WordPress
  get_url:
    url: 'https://wordpress.org/latest.zip'
    dest: '/tmp/wordpress.zip'

- name: Descomprimir WordPress
  unarchive:
    src: '/tmp/wordpress.zip'
    dest: '/var/www/html/'
    remote_src: yes

- name: Descargar WP-CLI
  ansible.builtin.get_url:
    url: 'https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar'
    dest: '/usr/local/bin/wp'
    mode: '0755'

- name: Configurar Apache para apuntar a la nueva carpeta
  ansible.builtin.lineinfile:
    path: /etc/apache2/sites-available/000-default.conf
    regexp: "^\\s*DocumentRoot"
    line: '    DocumentRoot /var/www/html/wordpress'
    notify: restart apache2

```

Posterior a esa tarea se realizó la instalación de MySQL, la creación de la base de datos como la creación de Usuario el cual se conectará a Wordpress más adelante

```

- name: Install MySQL
  apt:
    name:
      - mysql-server
      - python-mysqldb
    state: present
    update_cache: yes

- name: Crear base de datos y usuario en MySQL
  mysql_db:
    name: '{{ db_name }}'
    state: present

- name: Crear usuario MySQL
  mysql_user:
    name: '{{ db_user }}'
    password: '{{ db_password }}'
    host: localhost
    priv: '*.*:ALL'
    notify: restart apache2

```

Asignatura	Datos del alumno	Fecha
<b>Herramientas de Automatización de Despliegues</b>	Apellidos: López Baya	
	Nombre: Christian	

Luego se realizo la configuracion del archivo “wp-config.php” donde se llamo al template con las configuraciones para la base de datos de wordpress, posteriormente se lo llevo al destino de la carpeta donde esta wordpress en nuestra VM. Posterior a ello se hizo la instalacion de wordpress con el CLI previamente instalado y usando las variables definidas al inicio. Por ultimo tenemos tambien definido un Handler para el reinicio de APACHE2 el cual se fue llamando en algunas tareas para poder aplicar las mismas

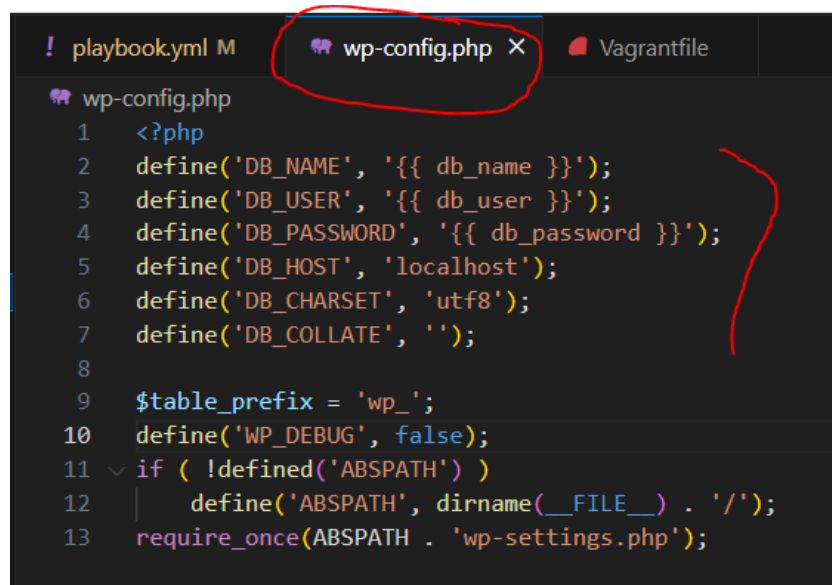
```

- name: Configurar wp-config.php
  template:
    src: wp-config.php
    dest: '/var/www/html/wordpress/wp-config.php'
    notify: restart apache2

- name: Instalar WordPress con WP-CLI
  ansible.builtin.command:
    cmd: 'wp core install --url=http://localhost:9000 --title="{{ wp_title }}" --admin_user="{{ wp_admin_user }}" --admin_password="{{ wp_admin_password }}" --skip-checks'
    become: yes

handlers:
  - name: restart apache2
    service:
      name: apache2
      state: restarted

```



```

! playbook.yml M wp-config.php X Vagrantfile

wp-config.php
1 <?php
2 define('DB_NAME', '{{ db_name }}');
3 define('DB_USER', '{{ db_user }}');
4 define('DB_PASSWORD', '{{ db_password }}');
5 define('DB_HOST', 'localhost');
6 define('DB_CHARSET', 'utf8');
7 define('DB_COLLATE', '');
8
9 $table_prefix = 'wp_';
10 define('WP_DEBUG', false);
11 if ( !defined('ABSPATH') )
12     define('ABSPATH', dirname(__FILE__) . '/');
13 require_once(ABSPATH . 'wp-settings.php');

```

Asignatura	Datos del alumno	Fecha
<b>Herramientas de Automatización de Despliegues</b>	Apellidos: López Baya	
	Nombre: Christian	

5.- Por último, al momento de realizar un “vagrant up” este instala todas las dependencias y podemos acceder mediante “localhost:9000” donde se puede visualizar el blog en Wordpress directamente levantado con ANSIBLE y Vagrant

```
TASK [Crear base de datos y usuario en MySQL] *****
changed: [default]

TASK [Crear usuario MySQL] *****
[WARNING]: Module did not set no_log for update_password
changed: [default]

TASK [Configurar wp-config.php] *****
changed: [default]

TASK [Instalar WordPress con WP-CLI] *****
changed: [default]

RUNNING HANDLER [restart apache2] *****
changed: [default]

PLAY RECAP *****
default                : ok=13   changed=12   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

