

# Desafio # 6

Fecha de entrega: 04/12/2024

## Objetivo:

Este desafío tiene como objetivo implementar un playbooks de Ansible y Roles para cada servicio.

Podremos desplegar un nuevo EC2 en AWS Academy de ser necesario para utilizar una familia de Ubuntu, **importante** usar la misma PEM para facilitar el acceso.

Utilizar el README de Sandbox como ayuda para acceder a la instancia de EC2 y configurar el *inventory.ini*

## Escenario:

Nuestro equipo de trabajo recientemente inició un nuevo sprint y nos asignaron una tarea muy importante para modularizar nuestro proyecto de *configuration management*, este sprint tiene como fecha de cierre el 06/12/2024 y debemos tener una versión funcional el día 04/12/2024 para la reunión de demos.

Este proyecto utiliza como herramienta de *configuration as a code* Ansible y permite instalar y desplegar nuestro sitio web de Wordpress en un host EC2 que es utilizaremos a demanda en un laboratorio de AWS Academy.

Como estamos trabajando en un entorno de Demo, necesitamos instalar PHP junto con sus componentes y MySQL dentro de una instancia EC2, no es necesario para este proyecto una base de datos RDS.

## Requisitos:

1. Utilizar un Bastion de AWS Academy
2. Identificar y crear posibles variables que se puedan reutilizar.

3. Crear un archivo playbook y roles específicos para PHP y MySQL
4. Testar y validar que todo funciona

Proyecto ejemplo: <https://medium.com/how-to-install-wordpress-on-ubuntu-a-step-by-step-guide>

## Entregables:

Los entregables establecidos para este proyecto con:

1. Código fuente del playbook de Ansible publicado en un repositorio de Github.
2. Guía detallada de cómo utilizar el Role publicado en el archivo README.md del repositorio.
3. Evidencia de las pruebas con resultado exitoso.

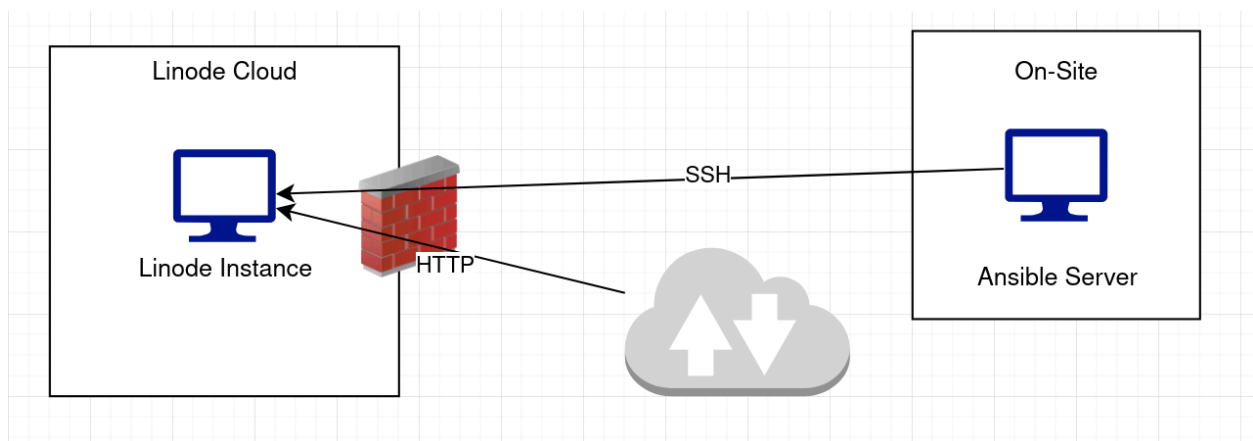
## Evaluación:

- Entrega en fecha.
- Redactar documentación legible y que sea comprendida por terceros.
- Añade material de soporte adicional.
  - Ejemplo: Diagrama de alto nivel.
- Cumple con las consignas solicitadas.
- Utilizar un archivo *playbook.yaml* y una carpeta role para separar cada servicio
- El entregable es funcional.
  - Ejemplo: repositorio con los archivos necesarios
  - Estructura de archivos (comando *tree* de Linux)

# Solución:

## Setup:

Usaré el siguiente esquema para la implementación:



1. Desde el servidor con Ansible voy a generar la clave SSH y copiarla a la instancia remota con:

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/ansible_key -N ""
```

2. y la copio al sistema remoto

```
ssh-copy-id -i ~/.ssh/ansible_key.pub root@172.233.27.216
```

3. Configuro el archivo de inventario para Ansible:

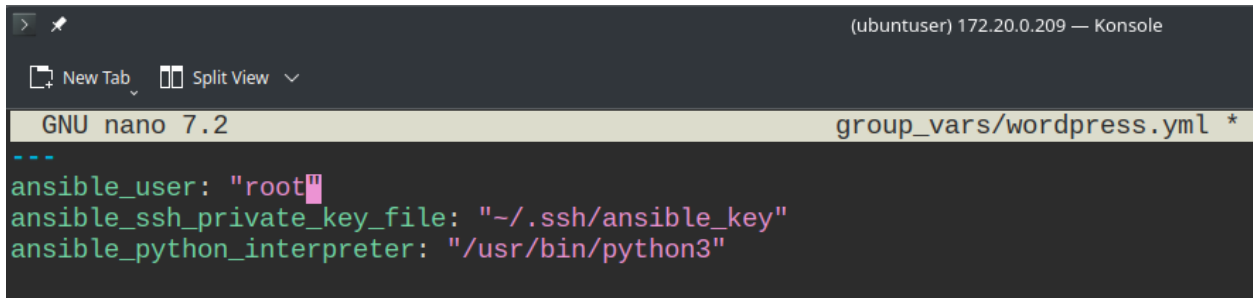
```
[wordpress]
172.233.27.216 ansible_user=root
ansible_ssh_private_key_file=~/.ssh/ansible_key
ansible_python_interpreter=/usr/bin/python3
```

4. Test de ping

```
ubuntuuser@ubuntu1ab:~/Desf6$ ansible -i inventory.ini wordpress -m ping
172.233.27.216 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
ubuntuuser@ubuntu1ab:~/Desf6$
```


# Variables:

Voy a mover las variables de usuario para el grupo de wordpress a un archivo de variables, dentro de un directorio group\_vars, archivo wordpress.yml (mismo nombre que el grupo en el inventario)



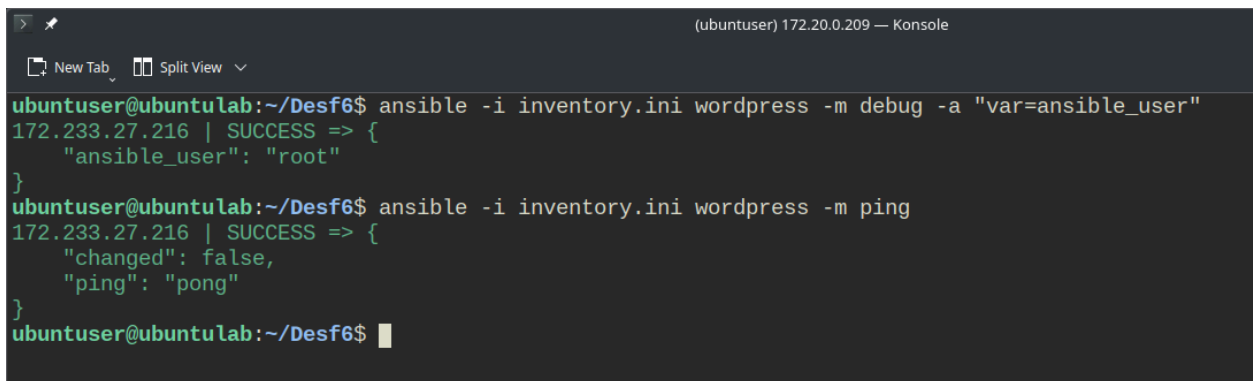
```
(ubuntuser) 172.20.0.209 — Konsole
GNU nano 7.2 group_vars/wordpress.yml *
---
ansible_user: "root"
ansible_ssh_private_key_file: "~/.ssh/ansible_key"
ansible_python_interpreter: "/usr/bin/python3"
```

Y en Inventory solo queda la IP de la instancia remota:



```
(ubuntuser) 172.20.0.209 — Konsole
GNU nano 7.2 inventory.ini *
[wordpress]
172.233.27.216
```

Pruebo nuevamente ping y debug



```
(ubuntuser) 172.20.0.209 — Konsole
ubuntuser@ubuntu1ab:~/Desf6$ ansible -i inventory.ini wordpress -m debug -a "var=ansible_user"
172.233.27.216 | SUCCESS => {
  "ansible_user": "root"
}
ubuntuser@ubuntu1ab:~/Desf6$ ansible -i inventory.ini wordpress -m ping
172.233.27.216 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
ubuntuser@ubuntu1ab:~/Desf6$
```

# Playbooks:

Crearé la siguiente estructura de directorios para el playbook:

roles/php/ para la instalación de PHP y dependencias  
roles/mysql/ para la instalación de MySQL y dependencias  
playbook.yml, el playbook principal

```
> ✂
New Tab Split View
ubuntuser@ubuntulab:~/Desf6$ tree
.
├── group_vars
│   └── wordpress.yml
├── inventory.ini
├── playbook.yml
└── roles
    ├── mysql
    │   └── tasks
    │       └── main.yml
    └── php
        └── tasks
            └── main.yml
```

playbook.yml

```
> ✂ (ubuntuser)
New Tab Split View
ubuntuser@ubuntulab:~/Desf6$ cat playbook.yml
---
- name: Configurar servidor Linode para WordPress
  hosts: wordpress
  become: true
  roles:
    - php
    - mysql
```

roles/php/tasks/main.yml

```
ubuntuuser@ubuntu1ab:~/Desf6$ cat roles/php/tasks/main.yml
---
- name: Instalar PHP y dependencias
  apt:
    name:
      - php
      - php-mysql
      - php-xml
      - php-mbstring
      - php-curl
      - php-zip
      - php-gd
    state: present
    update_cache: yes
ubuntuuser@ubuntu1ab:~/Desf6$
```

roles/mysql/tasks/main.yml

```
ubuntuuser@ubuntu1ab:~/Desf6$ cat roles/mysql/tasks/main.yml
---
- name: Instalar MySQL Server
  apt:
    name: mysql-server
    state: present
    update_cache: yes

- name: Habilitar y arrancar MySQL
  service:
    name: mysql
    state: started
    enabled: yes
```

Ejecución del Playbook:

```
(ubuntuuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...
ubuntuuser@ubuntu1ab:~/Desf6$ ansible-playbook -i inventory.ini playbook.yml

PLAY [Configurar servidor Linode para WordPress] *****

TASK [Gathering Facts] *****
ok: [172.233.27.216]

TASK [php : Instalar PHP y dependencias] *****
changed: [172.233.27.216]

TASK [mysql : Instalar MySQL Server] *****
changed: [172.233.27.216]

TASK [mysql : Habilitar y arrancar MySQL] *****
ok: [172.233.27.216]

PLAY RECAP *****
172.233.27.216      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

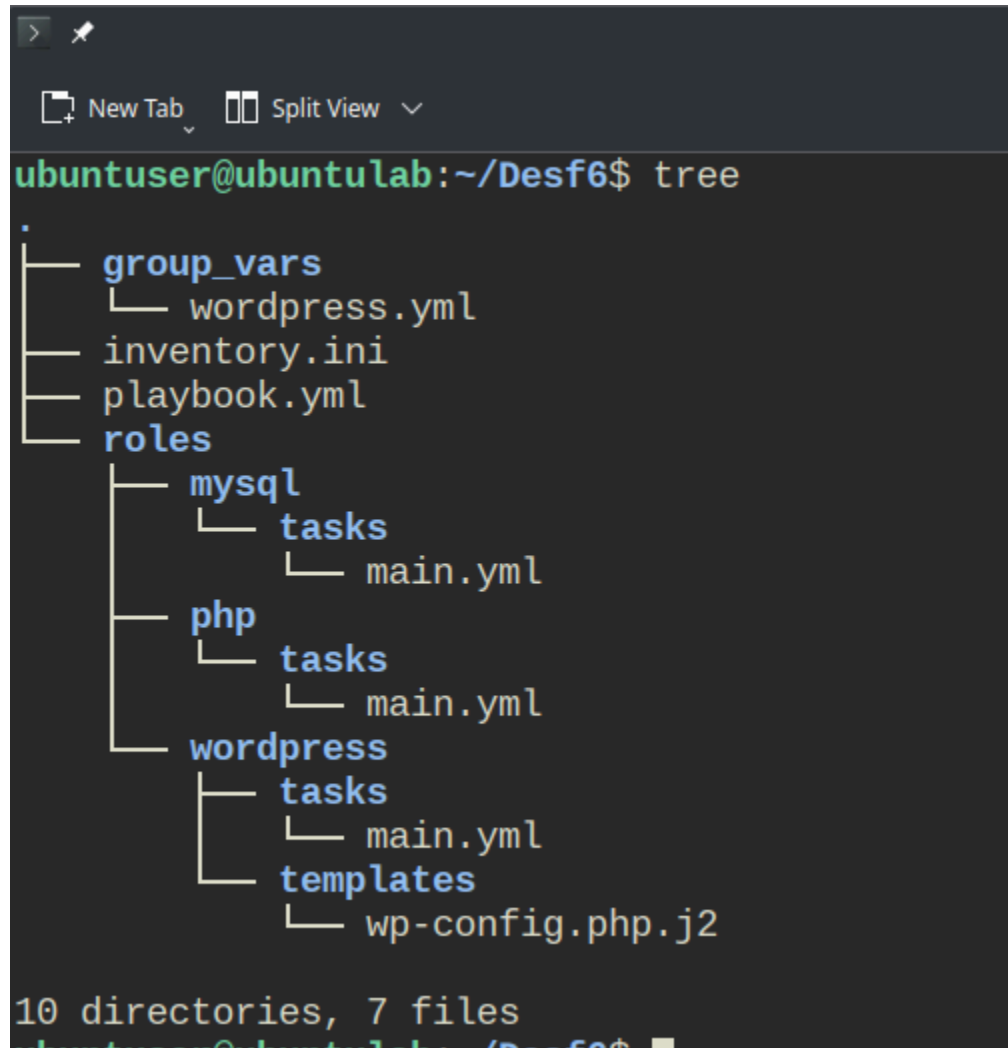
ubuntuuser@ubuntu1ab:~/Desf6$
```

# Configuración y Wordpress

*Nota: Luego de terminar la configuracion y, leyendo aqui*

*<https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-wordpress-with-lamp-on-ubuntu-18-04> me di cuenta de que hay muchas variables que se podrían “subir” al paso de definicion de variables. Esto es bueno porque hace el playbook más agnóstico y reutilizable pero no llegué a implementarlo.*

Creo un nuevo rol para la instalación de wordpress, la estructura de directorios queda así:

A terminal window with a dark background. The prompt is 'ubuntuser@ubuntulab:~/Desf6\$'. The command 'tree' has been executed, showing a directory tree. The tree starts with a root directory containing 'group\_vars', 'inventory.ini', 'playbook.yml', and 'roles'. 'roles' contains three sub-directories: 'mysql', 'php', and 'wordpress'. Each of these sub-directories contains 'tasks' and 'main.yml'. The 'wordpress' directory also contains a 'templates' sub-directory with a file 'wp-config.php.j2'. At the bottom, it says '10 directories, 7 files'.

```
ubuntuser@ubuntulab:~/Desf6$ tree
.
├── group_vars
│   └── wordpress.yml
├── inventory.ini
├── playbook.yml
└── roles
    ├── mysql
    │   ├── tasks
    │   │   └── main.yml
    ├── php
    │   ├── tasks
    │   │   └── main.yml
    └── wordpress
        ├── tasks
        │   └── main.yml
        └── templates
            └── wp-config.php.j2

10 directories, 7 files
ubuntuser@ubuntulab:~/Desf6$
```

Actualización de los roles:

**mysql/tasks/main.yml**

```
(ubuntuser) 172.20.0.209 — Konsole

New Tab Split View
GNU nano 7.2 mysql/tasks/main.yml *
--
- name: Instalar MySQL Server
  apt:
    name: mysql-server
    state: present
    update_cache: yes

- name: Habilitar y arrancar MySQL
  service:
    name: mysql
    state: started
    enabled: yes

- name: Crear base de datos de WordPress
  mysql_db:
    name: "wordpress_db"
    state: present

- name: Crear usuario MySQL para WordPress
  mysql_user:
    name: "wordpress_user"
    password: "wordpress_pass"
    priv: "wordpress_db.*:ALL"
    state: present

- name: arrancar MySQL
  service:
    name: mysql
    state: started
    enabled: yes
```

roles/wordpress/tasks/main.yml



```
(ubuntuser) 172.20.0.209 — Konsole
New Tab Split View
GNU nano 7.2 wordpress/tasks/main.yml *
---
- name: Descargar WordPress
  get_url:
    url: "https://wordpress.org/latest.tar.gz"
    dest: "/tmp/wordpress.tar.gz"

- name: Extraer WordPress
  unarchive:
    src: "/tmp/wordpress.tar.gz"
    dest: "/var/www/html/"
    remote_src: yes

- name: Configurar el archivo wp-config.php
  template:
    src: "wp-config.php.j2"
    dest: "/var/www/html/wordpress/wp-config.php"
  notify:
    - Reiniciar Apache
```

roles/wordpress/templates/wp-config.php.j2

```
(ubuntuser) 172.20.0.209 — Konsole
New Tab Split View
GNU nano 7.2 wordpress/templates/wp-config.php.j2
<?php
define( 'DB_NAME', 'wordpress_db' );
define( 'DB_USER', 'wordpress_user' );
define( 'DB_PASSWORD', 'wordpress_pass' );
define( 'DB_HOST', 'localhost' );
define( 'DB_CHARSET', 'utf8' );
define( 'DB_COLLATE', '' );

$table_prefix = 'wp_';

define( 'WP_DEBUG', false );

if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

require_once ABSPATH . 'wp-settings.php';
```

Y actualizo el playbook principal para incluir el rol de wordpress:

```
(ubuntuser) 172.20.0.209 — Konsole

GNU nano 7.2                                playbook.yml *
---
- name: Configurar servidor Linode para WordPress
  hosts: wordpress
  become: true
  roles:
    - php
    - mysql
    - wordpress
```

Ejecución del Playbook: (spoiler: no anduvo)

```
(ubuntuser) 172.20.0.209 — Konsole

ubuntuser@ubuntulab:~/Desf6$ ansible-playbook -i inventory.ini playbook.yml

PLAY [Configurar servidor Linode para WordPress] *****

TASK [Gathering Facts] *****
ok: [172.233.27.216]

TASK [php : Instalar PHP y dependencias] *****
ok: [172.233.27.216]

TASK [mysql : Instalar MySQL Server] *****
ok: [172.233.27.216]

TASK [mysql : Habilitar y arrancar MySQL] *****
ok: [172.233.27.216]

TASK [mysql : Crear base de datos de WordPress] *****
Fatal: [172.233.27.216]: FAILED! => {"changed": false, "msg": "A MySQL module is required: for Python 2.7 either PyMySQL, or MySQL-python, or for Python 3.X mysqlclient or PyMySQL. Consider setting ansible_python_interpreter to use the intended Python version."}

PLAY RECAP *****
172.233.27.216      : ok=4    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

ubuntuser@ubuntulab:~/Desf6$
```

# Troubleshooting

Ejecución del Playbook:

```
(ubuntuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...
ubuntuser@ubuntu1ab:~/Desf6$ ansible-playbook -i inventory.ini playbook.yml

PLAY [Configurar servidor Linode para WordPress] *****

TASK [Gathering Facts] *****
ok: [172.233.27.216]

TASK [php : Instalar PHP y dependencias] *****
ok: [172.233.27.216]

TASK [mysql : Instalar MySQL Server] *****
ok: [172.233.27.216]

TASK [mysql : Habilitar y arrancar MySQL] *****
ok: [172.233.27.216]

TASK [mysql : Crear base de datos de WordPress] *****
fatal: [172.233.27.216]: FAILED! => {"changed": false, "msg": "A MySQL module is required: for Python 2.7 either PyMySQL, or MySQL-python, or for Python 3.X mysqlclient or PyMySQL. Consider setting ansible_python_interpreter to use the intended Python version."}

PLAY RECAP *****
172.233.27.216 : ok=4 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0

ubuntuser@ubuntu1ab:~/Desf6$
```

```
TASK [mysql : Crear base de datos de WordPress]
*****
fatal: [172.233.27.216]: FAILED! => {"changed": false, "msg": "A MySQL module
is required: for Python 2.7 either PyMySQL, or MySQL-python, or for Python 3.X
mysqlclient or PyMySQL. Consider setting ansible_python_interpreter to use the
intended Python
version."}
```

Dado que la instancia remota viene con python3 instalo pip y PyMySQL

```
sudo apt-get install python3-pip
pip3 install PyMySQL
```

Lo agregué al playbook, junto con apt-update y apt-upgrade en el task de Wordpress

```
(ubuntuser) 172.20.0.209 — Konsole

GNU nano 7.2 main.yml *
---
- name: apt update y upgrade
  apt:
    update_cache: yes
    upgrade: dist
    autoclean: yes
    autoremove: yes

- name: Instalar pip3
  apt:
    name: python3-pip
    state: present

- name: Instalar PyMySQL con pip3
  pip:
    name: PyMySQL
    state: present
    extra_args: --break-system-packages # Investigar por que necesita este argumento

- name: Descargar WordPress
  get_url:
    url: "https://wordpress.org/latest.tar.gz"
    dest: "/tmp/wordpress.tar.gz"

- name: Extraer WordPress
  unarchive:
    src: "/tmp/wordpress.tar.gz"
    dest: "/var/www/html/"
    remote_src: yes

- name: Configurar el archivo wp-config.php
  template:
    src: "wp-config.php.j2"
    dest: "/var/www/html/wordpress/wp-config.php"
  notify:
    - Reiniciar Apache
```

## Ejecución del Playbook:

```
(ubuntuser) 172.20.0.209 — Konsole

ubuntuser@ubuntulab:~/Desf6$ ansible-playbook -i inventory.ini playbook.yml

PLAY [Configurar servidor Linode para WordPress] *****

TASK [Gathering Facts] *****
ok: [172.233.27.216]

TASK [php : Instalar PHP y dependencias] *****
ok: [172.233.27.216]

TASK [mysql : Instalar MySQL Server] *****
ok: [172.233.27.216]

TASK [mysql : Habilitar y arrancar MySQL] *****
ok: [172.233.27.216]

TASK [mysql : Crear base de datos de WordPress] *****
fatal: [172.233.27.216]: FAILED! => {"changed": false, "msg": "unable to find /root/.my.cnf. Exception message: (1698, \"Access denied for user 'root'@'localhost'\")"}

PLAY RECAP *****
172.233.27.216      : ok=4   changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

ubuntuser@ubuntulab:~/Desf6$
```

TASK [mysql : Crear base de datos de WordPress]

```
*****
fatal: [172.233.27.216]: FAILED! => {"changed": false, "msg": "unable to find
/root/.my.cnf. Exception message: (1698, \"Access denied for user
'root'@'localhost'\")\"}"}
```

MySQL está rechazando la conexión de root desde localhost, esto sucede en las versiones más actuales de MySQL.

*Workaround: Permitir el login con el user root a mysql. Nota: aunque necesito acceso a la instancia remota para hacer este cambio, en un ambiente productivo no usaría root para conectarme sino un usuario dedicado, lo que anularía este problema.*

```
(root) 172.233.14.112 — Konsole

New Tab Split View

root@localhost:~# sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '3NqSvQQeDJXf4Gu2';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
root@localhost:~#
```

**Ejecución del Playbook:**

```
(ubuntuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...

ubuntuser@ubuntu1ab:~/Desf6$ ansible-playbook -i inventory.ini playbook.yml

PLAY [Configurar servidor Linode para WordPress] *****

TASK [Gathering Facts] *****
ok: [172.233.14.112]

TASK [php : Instalar PHP y dependencias] *****
ok: [172.233.14.112]

TASK [mysql : Instalar MySQL Server] *****
ok: [172.233.14.112]

TASK [mysql : Habilitar y arrancar MySQL] *****
ok: [172.233.14.112]

TASK [mysql : Crear base de datos de WordPress] *****
changed: [172.233.14.112]

TASK [mysql : Crear usuario MySQL para WordPress] *****
[WARNING]: Option column_case_sensitive is not provided. The default is now false, so the column's name will be upcased.
The default will be changed to true in community.mysql 4.0.0.
changed: [172.233.14.112]

TASK [mysql : arrancar MySQL] *****
ok: [172.233.14.112]

TASK [wordpress : Descargar WordPress] *****
ok: [172.233.14.112]

TASK [wordpress : Extraer WordPress] *****
ok: [172.233.14.112]

TASK [wordpress : Configurar el archivo wp-config.php] *****
ok: [172.233.14.112]

PLAY RECAP *****
172.233.14.112 : ok=10 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

ubuntuser@ubuntu1ab:~/Desf6$
```

## Prueba de conexión:



## Apache2 Default Page

# Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

Al conectarme recibí el default page de Apache. Revisando el directorio Sites Available, note que el Documentroot seguía apuntando a `/var/www/html`.

```

root@localhost:/etc/apache2/sites-available# cat 000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

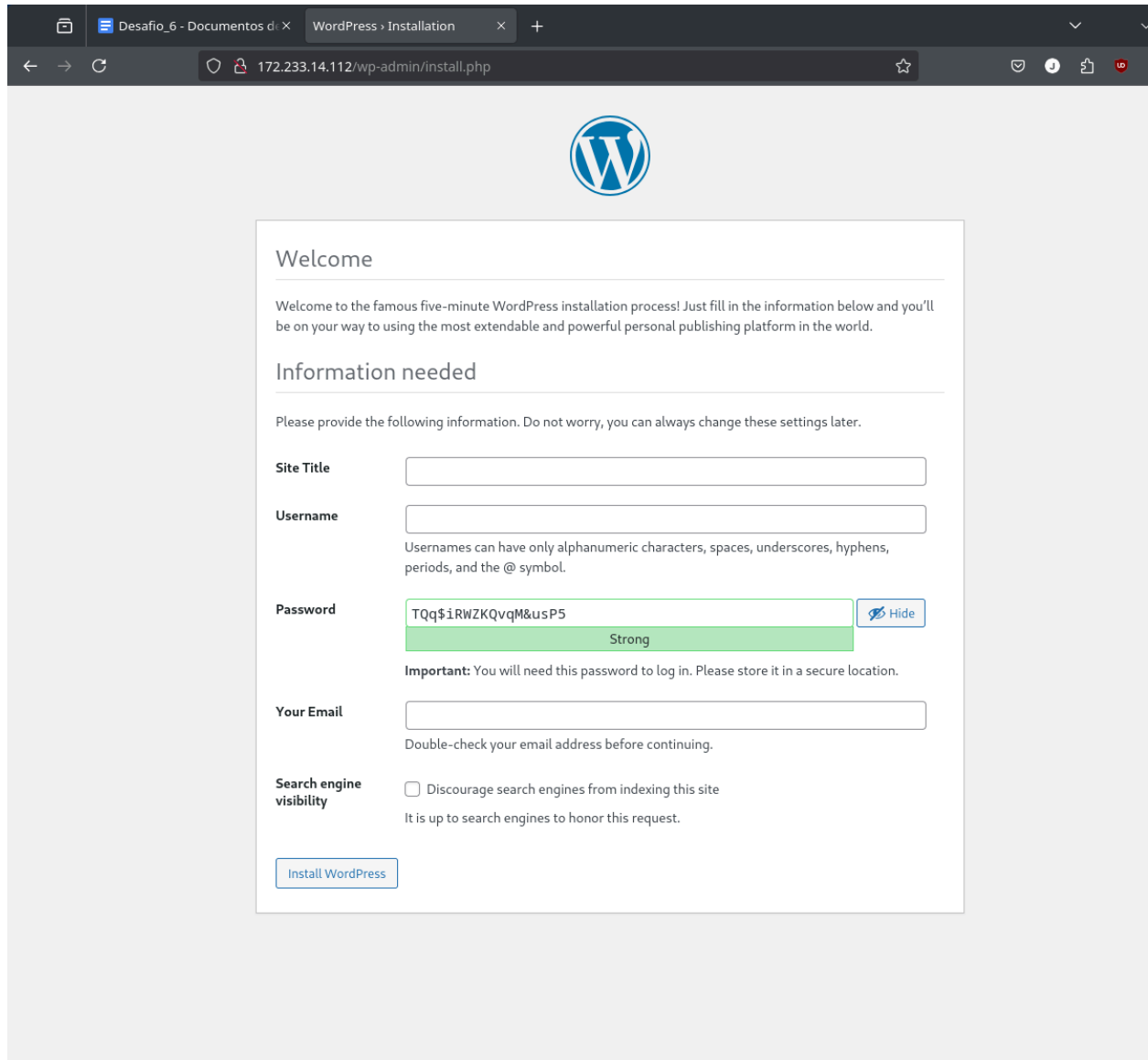
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

Luego de ajustarlo a `/var/www/html/wordpress` se pudo acceder a la pagina de configuracion:



The screenshot shows the WordPress installation interface in a web browser. The browser's address bar displays the URL `172.233.14.112/wp-admin/install.php`. The page features the WordPress logo at the top center. Below the logo, the 'Welcome' section provides a brief introduction to the five-minute installation process. The 'Information needed' section prompts the user to provide site details. It includes input fields for 'Site Title', 'Username', 'Password', and 'Your Email'. The 'Password' field is pre-filled with 'TQq\$iRWZKQvqM&usP5' and is marked as 'Strong'. A 'Hide' button is next to the password field. Below the password field, an 'Important' note states that the password is needed for login and should be stored securely. The 'Your Email' field has a note to double-check the address. A checkbox for 'Search engine visibility' is present, with a note that it is up to search engines to honor the request. At the bottom of the form is an 'Install WordPress' button.

WordPress

## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Do not worry, you can always change these settings later.

**Site Title**

**Username**

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

**Password**  [Hide](#)

Strong

**Important:** You will need this password to log in. Please store it in a secure location.

**Your Email**

Double-check your email address before continuing.

**Search engine visibility** ☐ Discourage search engines from indexing this site  
It is up to search engines to honor this request.

[Install WordPress](#)

Para corregir esto agregue un task en el rol de wordpress forzar la actualización del archivo:



```
(ubuntuser) 172.20.0.209 — Konsole
New Tab Split View
GNU nano 7.2 main.yml *
--
- name: apt update y upgrade
  apt:
    update_cache: yes
    upgrade: dist
    autoclean: yes
    autoremove: yes

- name: Instalar pip3
  apt:
    name: python3-pip
    state: present

- name: Instalar PyMySQL con pip3
  pip:
    name: PyMySQL
    state: present
    extra_args: --break-system-packages # Investigar por que necesita este argumento

- name: Descargar WordPress
  get_url:
    url: "https://wordpress.org/latest.tar.gz"
    dest: "/tmp/wordpress.tar.gz"

- name: Extraer WordPress
  unarchive:
    src: "/tmp/wordpress.tar.gz"
    dest: "/var/www/html/"
    remote_src: yes

- name: Configurar el archivo wp-config.php
  template:
    src: "wp-config.php.j2"
    dest: "/var/www/html/wordpress/wp-config.php"
  notify:
    - Reiniciar Apache

- name: Configurar Apache para servir WordPress
  lineinfile:
    path: /etc/apache2/sites-available/000-default.conf
    regexp: '^DocumentRoot /var/www/html'
    line: 'DocumentRoot /var/www/html/wordpress'
  notify:
    - Reiniciar Apache
```