Setting Up a LEMP Stack with WordPress on a Vagrant Debian 12 Virtual Machine and Accessing it Securely via Cloudflared Tunnel

Index

Requirements

- Purchase a **domain**.

- Create a **CloudFlared Tunnel**.

Modified the Vagrantfile of the TP01 to meet the following requirements:

- Install **LEMP stack** that consist of the components:
    - **Linux:** the OS that serves the environment for running the other components of the stack, in this case Debian 12.
    - **Nginx:** the web server that handles incoming HTTP requests and server content.
    - **MariaDB**: database server, used for storing and managing structured data in web applications. It is required to create a user and database for Wordpress other than the default.
    - **PHP:** is the programming language used for developing web applications.

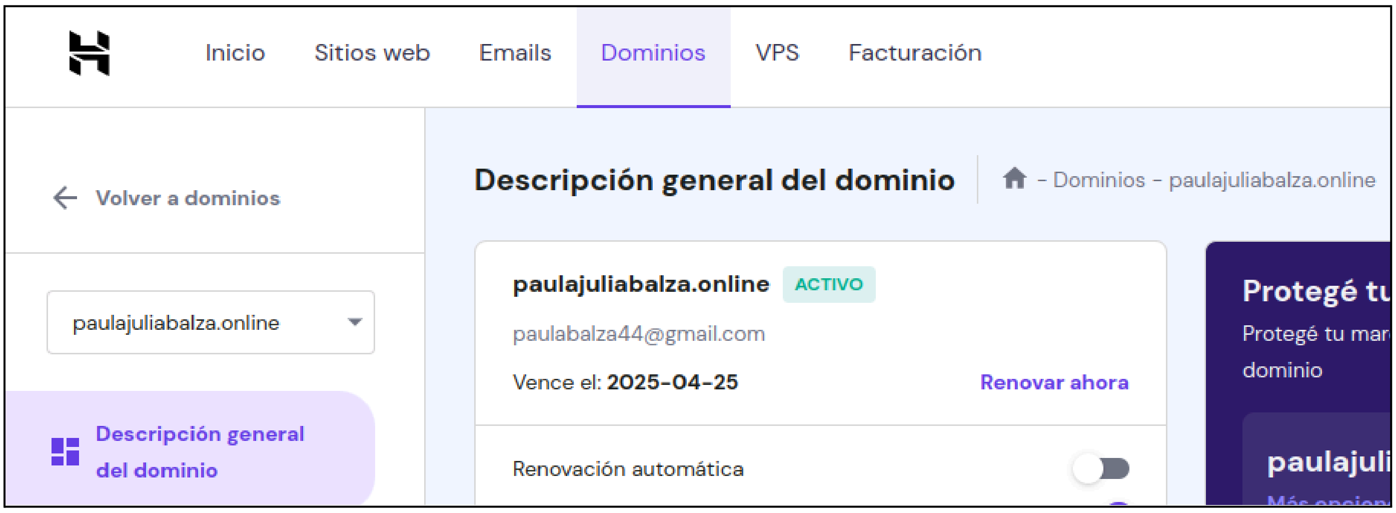
- Install and configure **WordPress**, a platform for building and managing websites.

- Finally, provide **access** to WordPress vía the CloudFlared Tunnel created.

Steps

Purchase a domain in Hostinger

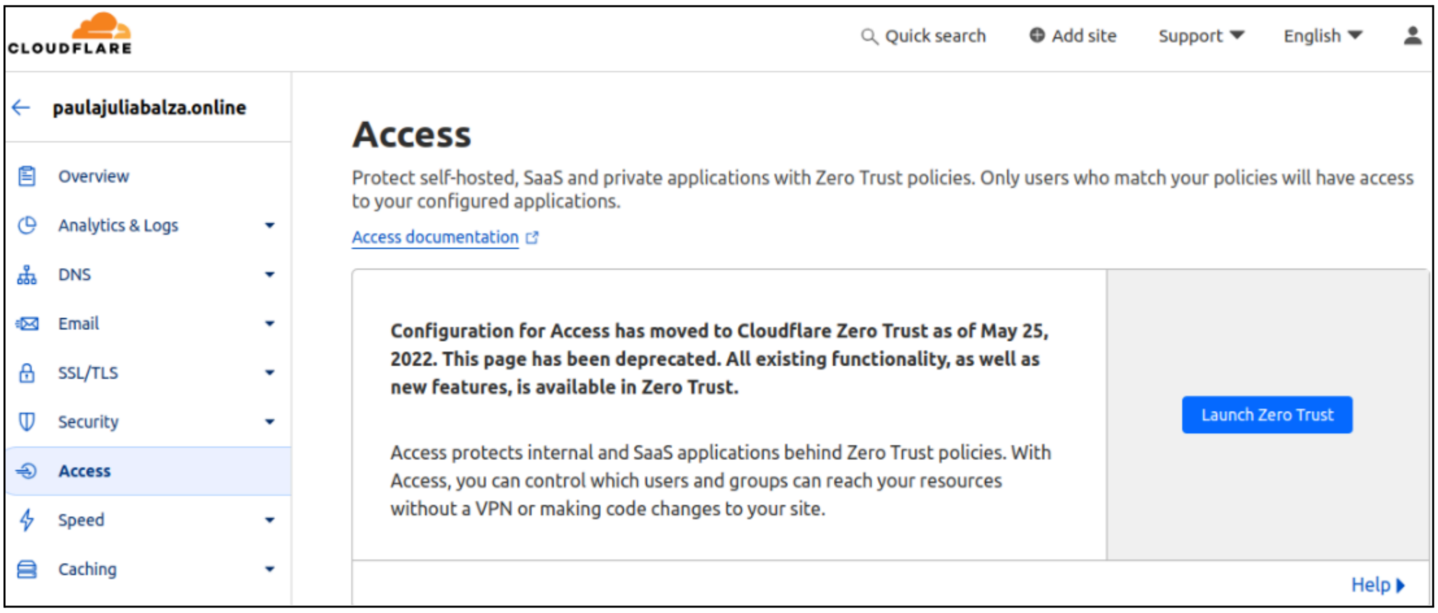
The domain paulajuliabalza.online was purchased on the Hostinger website (https://www.hostinger.com.ar/dominios).



paulajuliabalza.online ACTIVO
paulabalza44@gmail.com
Vence el: 2025-04-25 Renovar ahora
Renovación automática

Configure a Cloudflare Tunnel

After adding our website to Cloudflare (https://www.cloudflare.com/) in the Access section, we chose 'Launch Zero Trust' and began configuring the tunnel.



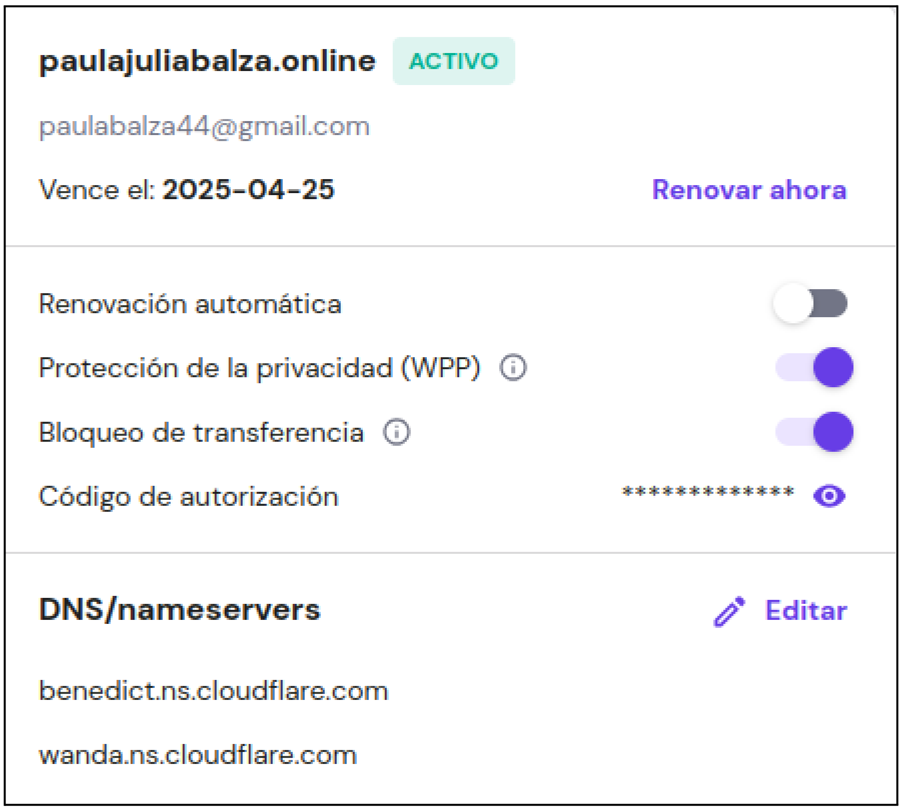
Access

Protect self-hosted, SaaS and private applications with Zero Trust policies. Only users who match your policies will have access to your configured applications.

Configuration for Access has moved to Cloudflare Zero Trust as of May 25, 2022. This page has been deprecated. All existing functionality, as well as new features, is available in Zero Trust.

Access protects internal and SaaS applications behind Zero Trust policies. With Access, you can control which users and groups can reach your resources without a VPN or making code changes to your site.

It is important to specify the DNS nameservers provided by Cloudflare in our Hostinger domain.



paulajuliabalza.online ACTIVO

paulabalza44@gmail.com

Vence el: **2025-04-25** Renovar ahora

Renovación automática

Protección de la privacidad (WPP)

Bloqueo de transferencia

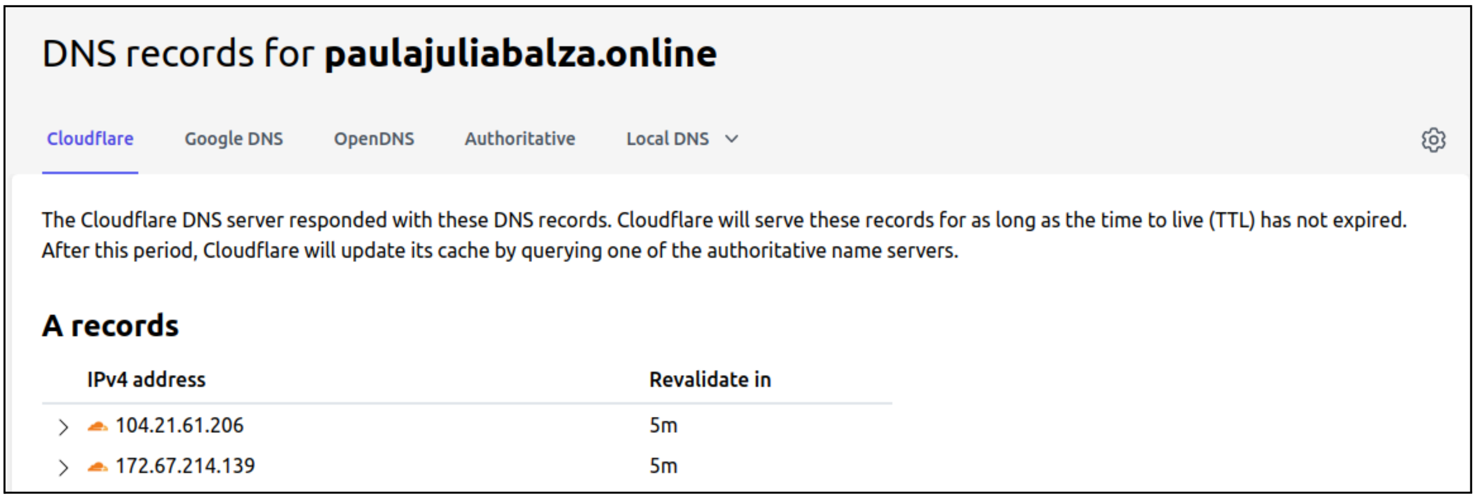
Código de autorización *************

DNS/nameservers Editar

benedict.ns.cloudflare.com

wanda.ns.cloudflare.com

You can check if it's working and if the DNS propagation was made correctly by querying the DNS nameservers of the domain with the following tool https://www.nslookup.io/.



DNS records for **paulajuliabalza.online**

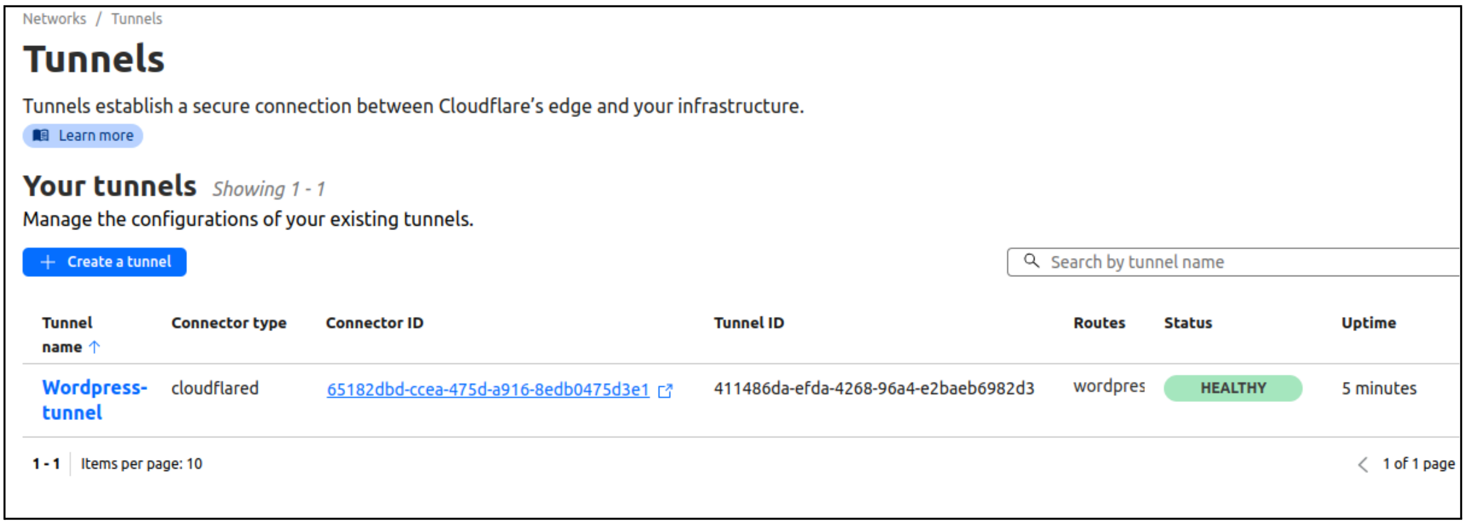
The Cloudflare DNS server responded with these DNS records. Cloudflare will serve these records for as long as the time to live (TTL) has not expired. After this period, Cloudflare will update its cache by querying one of the authoritative name servers.

A records

IPv4 address	Revalidate in
104.21.61.206	5m
172.67.214.139	5m

In brief steps, we navigate to Network > Tunnels > Add a Tunnel > Connector: CloudFlared > Specify a name for the tunnel > Environment: Debian 64 bit.

This last step will provide a list of commands and a Token that we need to run in the VM to establish the Tunnel connection from our VM to Cloudflare. After running the commands we can see the Tunnel up.



Networks / Tunnels

Tunnels

Tunnels establish a secure connection between Cloudflare's edge and your infrastructure.

Learn more

Your tunnels *Showing 1 - 1*

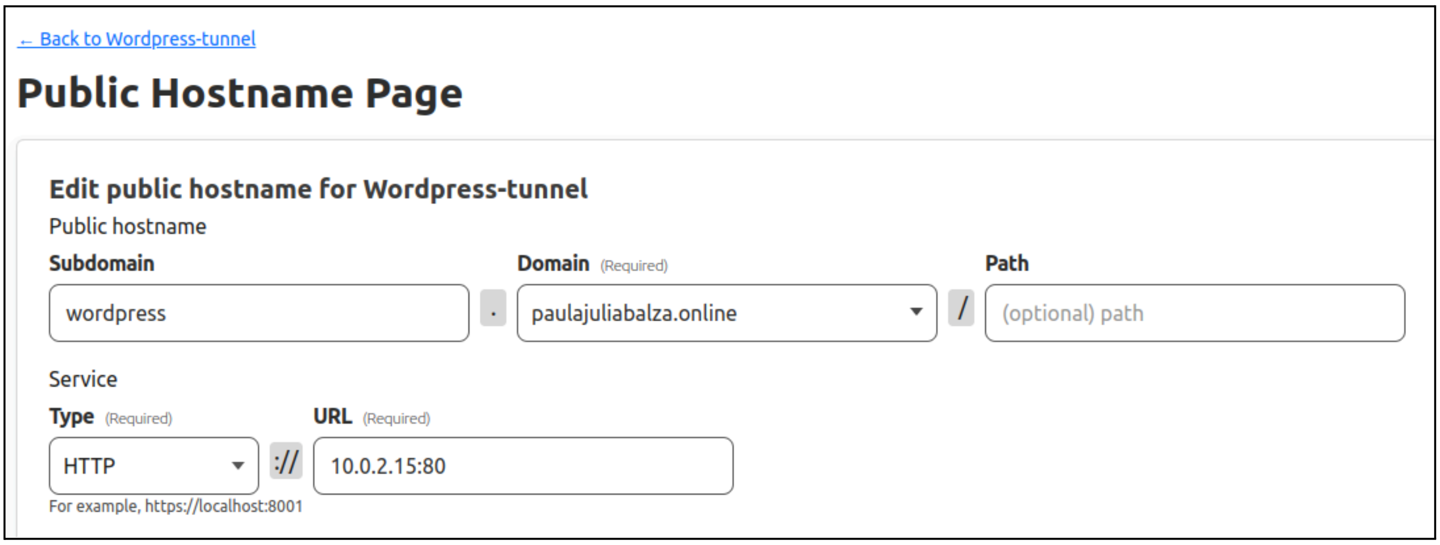
Manage the configurations of your existing tunnels.

Tunnel name	Connector type	Connector ID	Tunnel ID	Routes	Status	Uptime
Wordpress-tunnel	cloudflared	65182dbd-ccea-475d-a916-8edb0475d3e1	411486da-efda-4268-96a4-e2baeb6982d3	wordpres	HEALTHY	5 minutes

1 - 1 Items per page: 10 — 1 of 1 page

The list of commands is specified in a bash script called **tunnel_CloudFlare.sh** to automate the process with vagrant up.

After this is necessary to specify the domain that is going to be accessible via the tunnel, for this we specify the IP of our Debian 12 VM (you can obtain this by running the command ip a in the VM).



← Back to Wordpress-tunnel

Public Hostname Page

Edit public hostname for Wordpress-tunnel

Public hostname

Subdomain	Domain (Required)	Path
wordpress	paulajuliabalza.online	(optional) path

Service

Type (Required)	URL (Required)
HTTP	10.0.2.15:80

For example, https://localhost:8001

Automate the installation of a LEMP stack with Wordpress on a Vagrant Debian VM

For this we use a bash script called **provision_LEMP_Wordpress.sh** that is provisioned when we run the command vagrant up, this and the **tunnel_CloudFlare.sh** script are specified in the section provision of the Vagrantfile.

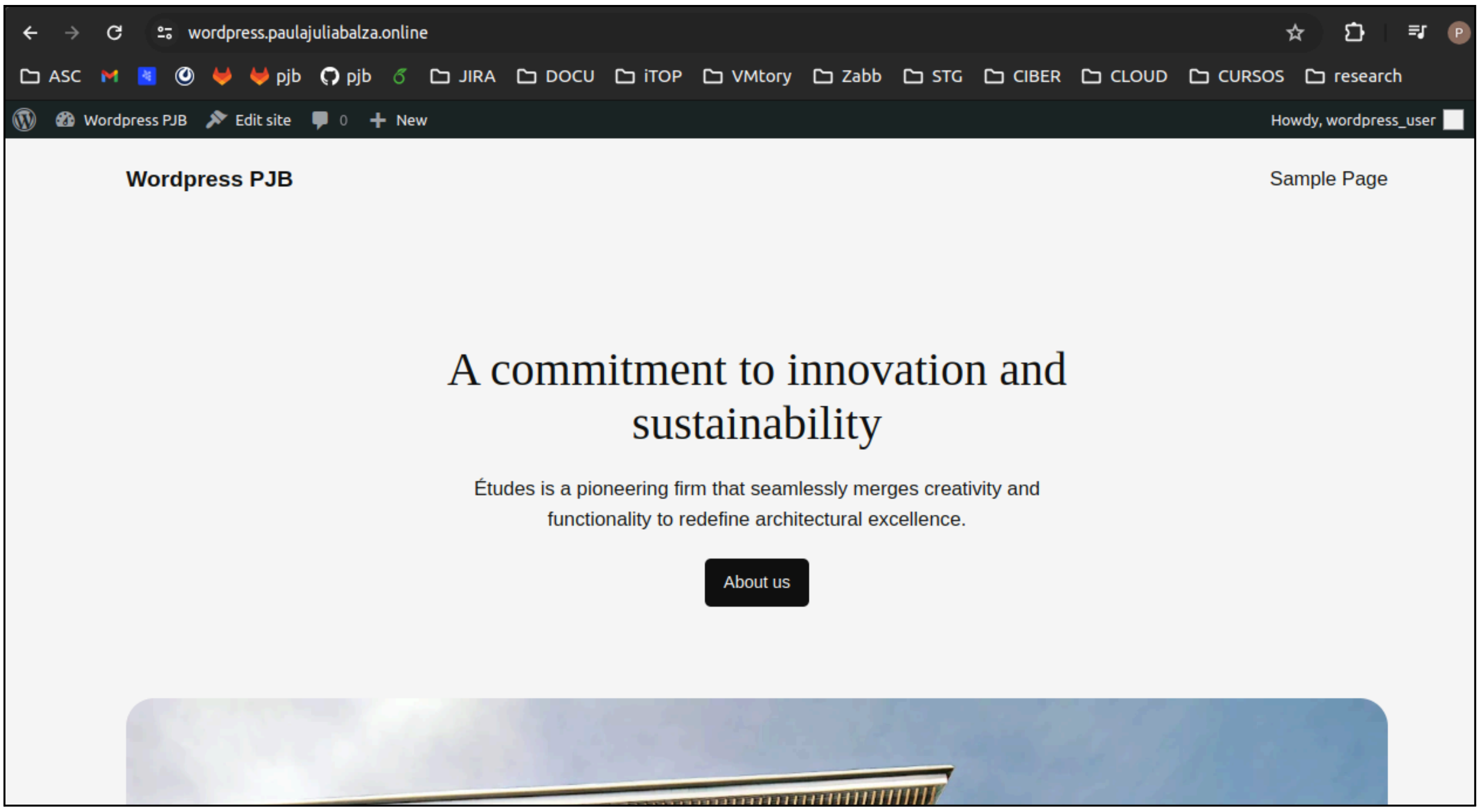
```
Vagrant.configure("2") do |config|
  config.vm.box = "debian/bookworm64"

  # Provider-specific configuration in this Case VirtualBox
  config.vm.provider "virtualbox" do |vb|
      # Display the VirtualBox GUI when booting the machine
      # vb.gui = true
      # Customize the amount of memory on the VM:
      vb.memory = "2048" # 2GB de RAM
      # Specify the name of the virtual machine.
      vb.name = "vm-Atenas"
  #config.vm.network "forwarded_port", guest: 80, host: 8080

  end

  # Enable provisioning with a shell script for Tunnel CloudFlare
  config.vm.provision "shell" do |s|
    s.path = "tunnel_CloudFlare.sh"
  end

  # Enable provisioning with a shell script for LEMP and Wordpress
  config.vm.provision "shell" do |s|
    s.path = "provision_LEMP_Wordpress.sh"
  end

end
```

Next we have the provision script shell **provision_LEMP_Wordpress.sh** with comments that detailed what is done in each step.

```
#!/bin/bash

# 1-Definition of variables
DBNAME=wordpress
DBUSER=wordpress_user
DBPASSWORD=pa55wordwordpre55

# 2-Update and Upgrade
sudo apt update -y && sudo apt upgrade -y

# 3-Install Nginx WebServer
sudo apt install nginx -y

# 4-Install MariaDb
sudo apt install mariadb-server -y

# 5-Install PHP
sudo apt install php-mysql php-fpm -y

# 6-Download latest Version of Wordpress
sudo wget -O - https://wordpress.org/latest.tar.gz | sudo tar -xzvf - -C /var/www/
sudo chown -R www-data:www-data /var/www/wordpress

# 7-Configure Nginx for Wordpress
echo 'upstream wp-php-handler {
    server unix:/var/run/php/php8.2-fpm.sock;
}
server {
    listen 80;
    server_name wordpress.paulajuliabalza.online;
    root /var/www/wordpress/;
    index index.php;
    location / {
        try_files $uri $uri/ /index.php?$args;
    }
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass wp-php-handler;
    }
}' | sudo tee /etc/nginx/sites-available/wordpress.conf > /dev/null

# 8-Remove nginx default file
sudo rm /etc/nginx/sites-available/default
sudo rm /etc/nginx/sites-enabled/default

# 9-Create symbolic link
sudo ln -s /etc/nginx/sites-available/wordpress.conf /etc/nginx/sites-enabled/

# 10-Reload Configuration
sudo systemctl reload nginx.service

# 11-Configuration Mariadb
sudo mariadb << EOF
CREATE DATABASE $DBNAME DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
GRANT ALL ON wordpress.* TO '$DBUSER'@'localhost' IDENTIFIED BY '$DBPASSWORD';
FLUSH PRIVILEGES;
EOF

# 12-Configure Wordpress user and DB
sudo cp /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php
sudo sed -i "s/database_name_here/$DBNAME/g" /var/www/wordpress/wp-config.php
sudo sed -i "s/username_here/$DBUSER/g" /var/www/wordpress/wp-config.php
sudo sed -i "s/password_here/$DBPASSWORD/g" /var/www/wordpress/wp-config.php

# 13-Reload Configuration
sudo systemctl reload nginx.service
```

Final result

Finally, by just running **vagrant up** we can access Wordpress running in our Debian 12 VM vía the purchased domain wordpress.paulajuliabalza.online, this shows the power of automation.



Wordpress PJB

Sample Page

A commitment to innovation and sustainability

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

References

https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lemp-on-debian-9
https://netshopisp.medium.com/how-to-install-wordpress-with-lemp-stack-on-debian-12-server-070b9ec91bdb