

# INSTALACIÓN WORDPRESS EN AWS



## **Sumario**

1. Introducción.....	3
2. Creacion de instancias.....	4
3. Descargar Apache y PHP.....	6
4. Crear Base de Datos.....	8
5. Elastic File System.....	13
6. Descarga de Wordpress.....	15

# 1. Introducción

Vamos a los servicios de VPC y creamos una VPC para las subredes.

The screenshot shows the 'Create VPC' wizard. On the left, under 'VPC settings', there are tabs for 'VPC only' (selected) and 'VPC and more'. Under 'Name tag auto-generation', 'Auto-generate' is checked and set to 'vpc-wordpress'. Under 'IPv4 CIDR block', the range is '10.0.0.0/16' with 65,536 IPs. Under 'IPv6 CIDR block', 'No IPv6 CIDR block' is selected. Under 'Tenancy', 'Default' is chosen. A note at the bottom says '► Encryption settings - optional'. On the right, the 'Preview' section shows a tree structure of the VPC. It starts with 'VPC Show details' (vpc-wordpress-vpc), which branches into 'Subnets (4)' (us-east-1a and us-east-1b). Each AZ has two subnets: 'A' (vpc-wordpress-s1) and 'B' (vpc-wordpress-s2).

Le pondremos 2 subredes publicas y 2 privadas. Ademas es recomendable ponerle un NAT gateway.

The screenshot shows the 'Create VPC' wizard. Under 'Number of Availability Zones (AZs)', '2' is selected. Under 'Number of public subnets', '2' is selected. Under 'Number of private subnets', '2' is selected. Under 'NAT gateways (\$)', 'Regional - new' is selected. A callout box at the bottom left provides information about the regional NAT gateway: 'Introducing regional NAT gateway' (AWS now offers a multi-AZ NAT Gateway, eliminating the need for separate NAT Gateways across availability zones). On the right, the 'Preview' section shows the VPC structure: 'VPC Show details' (vpc-wordpress-vpc) with 'Subnets (4)' (us-east-1a and us-east-1b) each containing two subnets (A and B).

## 2. Creacion de instancias

Vamos a ir a EC2 > Instancias. Aquí le daremos a Launch Instance

The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and a prominent orange 'Launch instances' button. Below the buttons is a search bar and a dropdown menu set to 'All states'. The main table lists two instances: 'SRI-SSL' (terminated, t2.micro) and 'dani\_server' (terminated, t3.micro). The 'Launch instances' button is highlighted with a red arrow.

A la instancia le pondremos un nombre y elegiremos el sistema operativo Debian.

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, the name 'servidorwordpress' is entered. In the 'Application and OS Images (Amazon Machine Image)' step, the 'AMI from catalog' tab is selected, showing 'Debian 13 (HVM), SSD Volume Type' as the chosen image. This image is highlighted with a red box. To the right, a sidebar displays summary information: 'Number of instances: 1', 'Software Image: Debian 13 (HVM)', 'Virtual server type: t3.micro', 'Firewall (security group): New security group', and 'Storage (volume): 1 volume(s) -'. A 'Cancel' button is also visible.

A parte, necesitaremos crear un grupo de seguridad con la siguiente configuración:

The screenshot shows two side-by-side configuration interfaces.

**Left Panel (Security Group Creation):**

- Firewall (security groups) | Info**: A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
- Create security group** (radio button selected).
- Select existing security group** (radio button available).
- Security group name - required**: `seguridadwordpress`.
- Description - required**: `launch-wizard-3 created 2025-12-18T13:05:44.824Z`.
- Inbound Security Group Rules**:
  - Security group rule 1 (TCP, 22, 212.104.181.12/32, http)**: Type: ssh, Protocol: TCP, Port range: 22. Source type: Custom, Source: 212.104.181.12/32.
  - Security group rule 2 (TCP, 80, 0.0.0.0/0, Apertura puerto 80)**: Type: HTTP, Protocol: TCP, Port range: 80. Source type: Anywhere, Source: 0.0.0.0/0.

**Right Panel (EC2 Instance Summary):**

- Summary**
- Number of instances**: 1
- Software Image (AMI)**: Debian 13 (HVM), SSD Volume Type, ami-0f9c27b471bdcc702
- Virtual server type (instance type)**: t2.micro
- Firewall (security group)**: New security group
- Storage (volumes)**: 1 volume(s) - 8 GiB
- Launch instance** (button highlighted with a red arrow pointing from the bottom right of the left panel).
- Preview code** (button below the launch button).

Despues de eso, la instancia ya estará correctamente creada, faltaría desde el cmd desde el PC, conectarnos a la instancia con SSH.

```
C:\Users\Usuario>cd .ssh
C:\Users\Usuario\.ssh>ssh -i "wordpress.pem" admin@ec2-52-91-244-202.compute-1.amazonaws.com
The authenticity of host 'ec2-52-91-244-202.compute-1.amazonaws.com (52.91.244.202)' can't be established.
ED25519 key fingerprint is SHA256:DeHgIcwNvXDT1Efom3Khw8ngHDzkh0RallcxYYEq9ZE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-91-244-202.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Linux ip-172-31-21-194 6.12.48+deb13-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.48-1 (2025-09-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
admin@ip-172-31-21-194:~$
```

### 3. Descargar Apache y PHP

Hacemos un update para recibir los paquetes que nos faltan.

```
admin@ip-172-31-21-194: ~
admin@ip-172-31-21-194:~$ admin@ip-172-31-21-194:~$ sudo apt update
```

Despues instalamos el servidor apache con este comando:

```
admin@ip-172-31-21-194: ~
admin@ip-172-31-21-194:~$ sudo apt install apache2
```

Despues lo iniciamos y comprobamos que esta “running” .

```
admin@ip-172-31-21-194: ~
admin@ip-172-31-21-194:~$ sudo systemctl start apache2
admin@ip-172-31-21-194:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
admin@ip-172-31-21-194:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-12-18 13:52:25 UTC; 39s ago
     Invocation-Id: bbed8e743824132d13b8af6b370053
       Docs: https://httpd.apache.org/docs/2.4/
      Main PID: 1392 (apache2)
        Tasks: 55 (limit: 1127)
       Memory: 5.4M (peak: 5.6M)
         CPU: 33ms
      CGroup: /system.slice/apache2.service
              └─1392 /usr/sbin/apache2 -k start
                  ├─1394 /usr/sbin/apache2 -k start
                  ├─1395 /usr/sbin/apache2 -k start

Dec 18 13:52:25 ip-172-31-21-194 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 18 13:52:25 ip-172-31-21-194 apachectl[1391]: AH00558: apache2: Could not reliably determine the server's f
Dec 18 13:52:25 ip-172-31-21-194 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-17/17 (END)
```

Si vamos al navegador y ponemos la dirección IP de la instancia, veremos que nuestro servidor apache está activo.

The configuration layout for an Apache2 web server installation on Debian 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.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`.

También es conveniente obtener PHP, para ello deberíamos hacer otro update y upgrade.

```
admin@ip-172-31-21-194: ~$ sudo apt -y update && sudo apt upgrade
```

Luego descargamos PHP con un modulo de apache.

```
admin@ip-172-31-21-194:~$ sudo apt install php libapache2-mod-php php-cli
Installing:
  libapache2-mod-php  php  php-cli
```

Es necesario descargar mysql como modulo de apache:

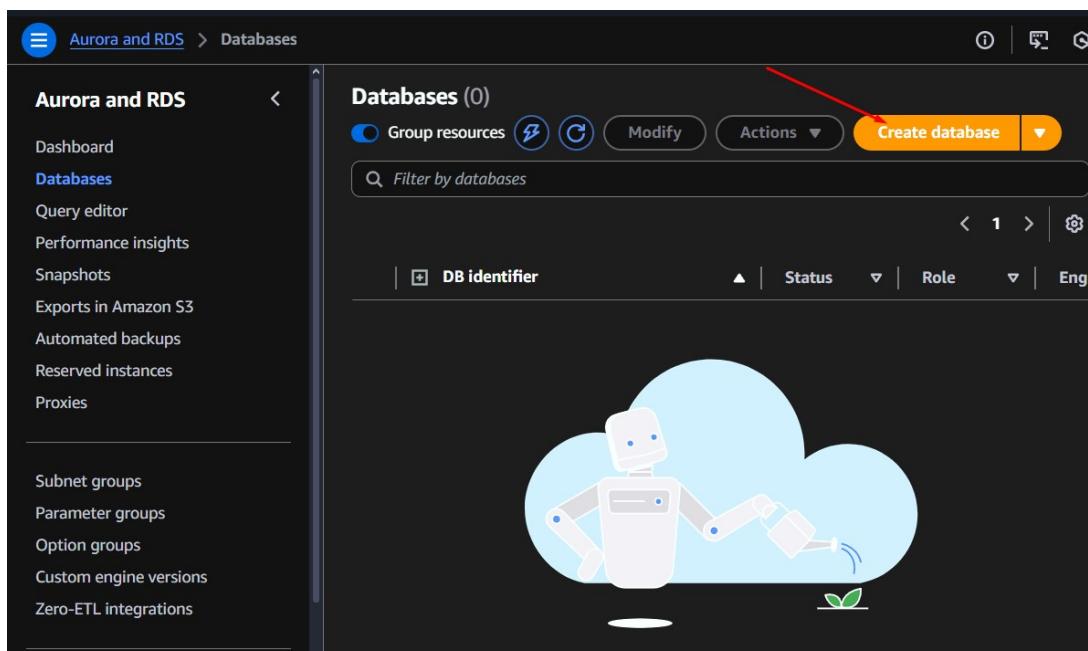
```
admin@ip-172-31-21-194:~$ sudo apt install php-mysql
```

Luego para aplicar los cambios, reiniciamos el server de apache y comprobamos si PHP ha sido instalado correctamente con “php -v”.

```
admin@ip-172-31-21-194:~$ sudo systemctl restart apache2
admin@ip-172-31-21-194:~$ php -v
PHP 8.4.11 (cli) (built: Aug  3 2025 07:32:21) (NTS)
Copyright (c) The PHP Group
Built by Debian
Zend Engine v4.4.11, Copyright (c) Zend Technologies
    with Zend OPcache v8.4.11, Copyright (c), by Zend Technologies
admin@ip-172-31-21-194:~$
```

## 4. Crear Base de Datos

En el servicio de RDS, le damos a Databases > Create Database.



Elegiremos una BD de MySQL.

**Choose a database creation method**

**Full configuration**  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

**Easy create**  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

**Engine type** [Info](#)

**Aurora (MySQL Compatible)** 

**Aurora (PostgreSQL Compatible)** 

**MySQL** 

**PostgreSQL** 

**MariaDB** 

**Oracle** 

**Microsoft SQL Server** 

**IBM Db2** 

Seleccionaremos el Template “Sandbox” y que solo cree una “DB instance deployment”.

**Templates**

Choose a sample template to meet your use case.

**Production**  
Use defaults for high availability and fast, consistent performance.

**Dev/Test**  
This instance is intended for development use outside of a production environment.

**Sandbox**  
To develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

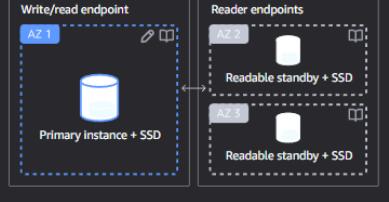
**Availability and durability**

**Deployment options** [Info](#)

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

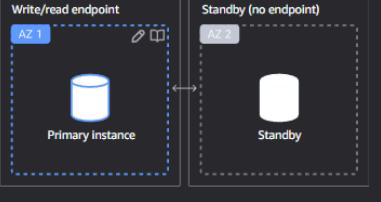
**Multi-AZ DB cluster deployment (3 instances)**  
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:

- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency



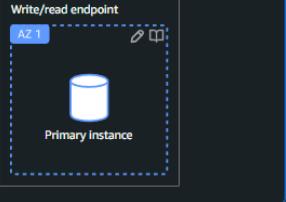
**Multi-AZ DB instance deployment (2 instances)**  
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:

- 99.95% uptime
- Redundancy across Availability Zones



**Single-AZ DB instance deployment (1 instance)**  
Creates a single DB instance without standby instances.  
This setup provides:

- 99.5% uptime
- No data redundancy



Le escribiremos un nombre, el usuario admin y una contraseña que queramos.

**Settings**

**DB instance identifier** [Info](#)  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**Credentials Settings**

**Master username** [Info](#)  
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.

**Managed in AWS Secrets Manager - most secure**  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

**Self managed**  
Create your own password or have RDS create a password that you manage.

**Auto generate password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** | [Info](#)

**Password strength** Strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

**Confirm master password** | [Info](#)

En la configuración de la instancia le ponemos que tenga un “db.t3.micro” con un SSD de 200 GB.

**Instance configuration**

The DB instance configuration options below are limited to those supported by the engine that you selected above.

**DB instance class** | [Info](#)

**Hide filters**

**Show instance classes that support Amazon RDS Optimized Writes** [Info](#)  
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

**Include previous generation classes**

**Standard classes (includes m classes)**

**Memory optimized classes (includes r and x classes)**

**Burstable classes (includes t classes)**

**db.t3.micro**  
2 vCPUs 1 GiB RAM EBS Bandwidth: Up to 2,085 Mbps Network: Up to 5 Gbps

**Storage**

**Storage type** [Info](#)  
Provisioned IOPS SSD (io2) storage volumes are now available.

**General Purpose SSD (gp2)**  
Baseline performance determined by volume size

**Allocated storage** [Info](#)  
 GiB  
Allocated storage value must be 20 GiB to 6,144 GiB

**Additional storage configuration**

No hace falta que se conecte a ningun recurso EC2.

The screenshot shows the 'Connectivity' section of the AWS RDS configuration interface. It includes fields for 'Compute resource' (set to 'Don't connect to an EC2 compute resource'), 'Virtual private cloud (VPC)' (set to 'Default VPC (vpc-05a4f460ff5424ba)'), and 'DB subnet group' (set to 'default'). A note states: 'After a database is created, you can't change its VPC.'

Pondremos que no tenga acceso publico y tambi n elegiremos un nombre para ponerle al nuevo grupo de seguridad.

The screenshot shows the security configuration section. Under 'Public access', 'No' is selected. Under 'VPC security group (firewall)', 'Create new' is selected. A new security group named 'seguridadadbwordpress' is being created. Under 'Availability Zone', 'No preference' is chosen. Under 'RDS Proxy', 'Create an RDS Proxy' is selected. Under 'Certificate authority - optional', 'rds-ca-rsa2048-g1 (default)' is chosen, with an expiry date of May 26, 2061. A note says: 'If you don't select a certificate authority, RDS chooses one for you.'

Como podemos comprobar, la base de datos ya esta creada, falta ponerle una conexión EC2, entonces le damos clic derecho > Set up EC2 connection.

The screenshot shows the AWS RDS 'Databases' page. A green header bar at the top indicates that a database has been successfully created. Below this, a table lists the database 'bdwordpress'. The 'Actions' column for this database has a context menu open, with the option 'Set up EC2 connection' highlighted by a red arrow. Other options in the menu include 'Stop temporarily', 'Reboot', 'Delete', 'Set up Lambda connection', 'Migrate data from EC2 database', 'Create read replica', 'Create Aurora read replica', 'Create blue/green deployment', and 'Promote'.

De instancia le pondremos la que habiamos creado al principio.

The screenshot shows the 'Set up EC2 connection' wizard. The first step, 'Select EC2 instance', is displayed. The 'Database' dropdown is set to 'bdwordpress'. The 'EC2 instance' dropdown contains one item: 'i-00eeae9db5d609493 servidorwordpress us-east-1a'. At the bottom right of the wizard, there are 'Cancel' and 'Continue' buttons.

## 5. Elastic File System

Vamos a los servicios de EFS y creamos un nuevo sistema de archivos.

The screenshot shows the 'File systems' page in the Amazon EFS console. On the left, there's a sidebar with links for 'File systems', 'Access points', 'AWS Backup', 'AWS DataSync', 'AWS Transfer', and 'Documentation'. The main area has a header 'File systems (0)' with a search bar and buttons for 'View details', 'Delete', and 'Create file system'. Below the header, there's a table with columns: Name, File system ID, Encrypted, Total size, Size in Standard, Size in IA, and Size in Archive. A message 'No resources' is displayed. At the bottom right of the table area, there's a prominent orange 'Create file system' button, which is highlighted with a red arrow.

Le ponemos un nombre personalizado y elegiremos la VPC que habíamos creado antes.

The screenshot shows the 'Create file system' dialog box. At the top, it says 'Create file system' and provides instructions: 'Create a file system with the recommended settings shown below by choosing Create file system. To view all settings or to customize your file system, choose Customize. [Learn more](#)'.  
**Name - optional**: Name your file system. The input field contains 'almacenwordpress'. Below it, a note says 'Name can include letters, numbers, and +-=\_.:/ symbols, up to 256 characters.'  
**Virtual Private Cloud (VPC)**: Choose the VPC where you want EC2 instances to connect to your file system. The dropdown menu shows 'vpc-033381c64faebf811' and 'vpc-wordpress-vpc'.  
**Recommended settings**: This section lists recommended settings for the file system:

Setting	Value	Editable after creation
Throughput mode <a href="#">Learn more</a>	Elastic	Yes
Transition into Infrequent Access (IA)	30 day(s) since last access	Yes
Transition into Archive	90 day(s) since last access	Yes
Transition into Standard	None	Yes
Automatic backups	Enabled	Yes
Encryption	Enabled	No

At the bottom of the dialog are buttons for 'Cancel', 'Customize' (which is outlined in blue), and 'Create file system' (which is orange).

Despues iremos a las Inbound Rules de la instancia y crearemos una regla nueva que permita a la instancia del servidor entrar.

The screenshot shows the 'Edit inbound rules' page. It lists three existing rules:

- sgr-0a954e1bb57f19571: Type HTTP, Protocol TCP, Port range 80, Source 0.0.0.0/0, Description Apertura puerto 80.
- sgr-0a0770194f60f3bb2: Type SSH, Protocol TCP, Port range 22, Source 212.104.181.12/32, Description http.
- : Type NFS, Protocol TCP, Port range 2049, Source 52.91.244.202/32, Description regla instancia.

At the bottom, there is a warning message: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." Below the message are 'Cancel', 'Preview changes', and 'Save rules' buttons.

Luego le daremos a que se conecte por IP.

The screenshot shows the 'Attach' dialog box. It has two main options:

- Mount via DNS
- Mount via IP

Below the options, there are sections for 'Using the EFS mount helper:' and 'Using the NFS client:', each containing a code snippet. At the bottom, there is a 'See our user guide for more information.' link and a 'Close' button.

Luego ponemos el siguiente comando para montar el recurso.

```
admin@ip-172-31-21-194: $ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0a1f5e749519a9206.efs.us-east-1.amazonaws.com:/ efs
```

## 6. Descarga de Wordpress

Luego nos vamos al directorio de html y usamos el comando wget para instalar wordpress.

```
Administrator Select admin@ip-172-31-21-194: /var/www/html
admin@ip-172-31-21-194:~$ cd /var/www/html
admin@ip-172-31-21-194:/var/www/html$ wget http://wordpress.org/latest.tar.gz
```

Luego descomprimimos lo descargado.

```
Administrator Select admin@ip-172-31-21-194:~$ tar -xf latest.tar.gz
admin@ip-172-31-21-194:~$
```

Luego instalamos el mysql client.

```
Administrator Select admin@ip-172-31-21-194:~$ sudo apt install default-mysql-client
Installing:
 default-mysql-client

Installing dependencies:
 libconfig-inifiles-perl  libdbi-perl  libncurses6      libterm-readkey-perl  mariadb-client-compat  mariadb-common
 libdbd-mariadb-perl     libmariadb3  libpcre2-posix3  mariadb-client       mariadb-client-core  mysql-common

Suggested packages:
 libclone-perl  libmldb-perl  libnet-daemon-perl  libsql-statement-perl

Summary:
 Upgrading: 0, Installing: 13, Removing: 0, Not Upgrading: 0
 Download size: 5527 kB
 Space needed: 86.0 MB / 6458 MB available

Continue? [Y/n] Y
```

Despues tenemos que conectarnos la base de datos para configurarla de manera que podamos usar wordpress.

```
Administrator Select admin@ip-172-31-21-194: /var/www/html
admin@ip-172-31-21-194:/var/www/html$ sudo mysql -u admin -h bdwordpress.ca2v85hycvw.us-east-1.rds.amazonaws.com -p
```

Finalmente, en el navegador podremos entrar en wordpress y configurar desde la interfaz web.



A continuación debes introducir los detalles de conexión de tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web.

Nombre de la base de datos	<input type="text" value="wordpress"/>	El nombre de la base de datos que quieras usar con WordPress.
Nombre de usuario	<input type="text" value="danibenitez"/>	El nombre de usuario de tu base de datos.
Contraseña	<input type="text" value="wordpressASIR "/>	La contraseña de tu base de datos.
Servidor de la base de datos	<input type="text" value="localhost"/>	Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.
Prefijo de tabla	<input type="text" value="wp_"/>	Si quieras ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.