

**Universidad Peruana de Ciencias Aplicadas**

**Arquitectura de Computadoras y Sistemas Operativos**

**2021-1**

**Sección SI43**

**Grupo 8**

**Trabajo Final**

**Profesor:**

Salas Arbaiza, César Enrique

**ALUMNOS:**

U201719588 Jean Paul, Luyo Ballena

U201922128 Daniel Ulises, Barrionuevo Gutiérrez

# Introducción

En el presente proyecto se explicará cómo se llevó a cabo la solución para el caso planteado en el curso, el cual será realizado principalmente en la plataforma de Amazon Web Services educate.

El caso consiste en diseñar y dimensionar la infraestructura necesaria para poder crear un sitio web tolerante a fallos con su respectivo balanceador de carga y una base de datos exterior. Para corroborar el correcto funcionamiento de esta infraestructura se implementará un gestor de contenido (CMS), el cual será WordPress.

Este proyecto debe ser analizado técnica y económicamente; puesto que, se requiere que este diseño considere el mínimo costo posible para llevarlo a cabo.

Para el correcto desarrollo del proyecto, se realizó un plan de trabajo e investigación con el fin de recolectar toda la información necesaria sobre los elementos solicitados para la implementación de la solución del problema.

# Índice

[Introducción 0](#_Toc44500489)

[Índice 2](#_Toc44500490)

[Objetivo del Estudiante (Student Outcome) 3](#_Toc44500491)

[Análisis técnico y económico 4](#_Toc44500492)

[Implementación de la solución 5](#_Toc44500493)

[Presupuesto sustentado 26](#_Toc44500494)

[Conclusiones 28](#_Toc44500495)

[Anexos 29](#_Toc44500496)

[Bibliografía 31](#_Toc44500497)

# Objetivo del Estudiante (Student Outcome)

**ABET – EAC - Student Outcome 1:** La capacidad de identificar, formular y resolver problemas complejos de ingeniería aplicando los principios de ingeniería, ciencia y matemática.

**ABET – CAC - Student Outcome 1:** La capacidad de analizar un problema complejo aplicando los principios de computación, ciencia y matemática para identificar soluciones.

**ABET – CAC - Student Outcome 1:** La capacidad de analizar un problema complejo aplicando los principios de computación, ciencia y matemática para identificar soluciones.

# Análisis técnico y económico

**Máquina virtual:**

Para la implementación de la solución del proyecto se escogió en una VM de tamaño Estándar\_B1s, esto debido a que las máquinas virtuales que pertenecen a los B-Series son los ideales para proyectos que no necesiten una alta demanda de CPU constantemente, como servidores web o pequeñas bases de datos. Además, con la finalidad de considerar el menor costo posible, este tamaño cuesta $7.59 al mes (ver anexo 1).

Con respecto al hardware, la máquina virtual escogida es te tipo t2.micro, posee 1 vCPU (Procesador virtual), 1 GB de memoria RAM, con almacenamiento disponible solo para EBS.

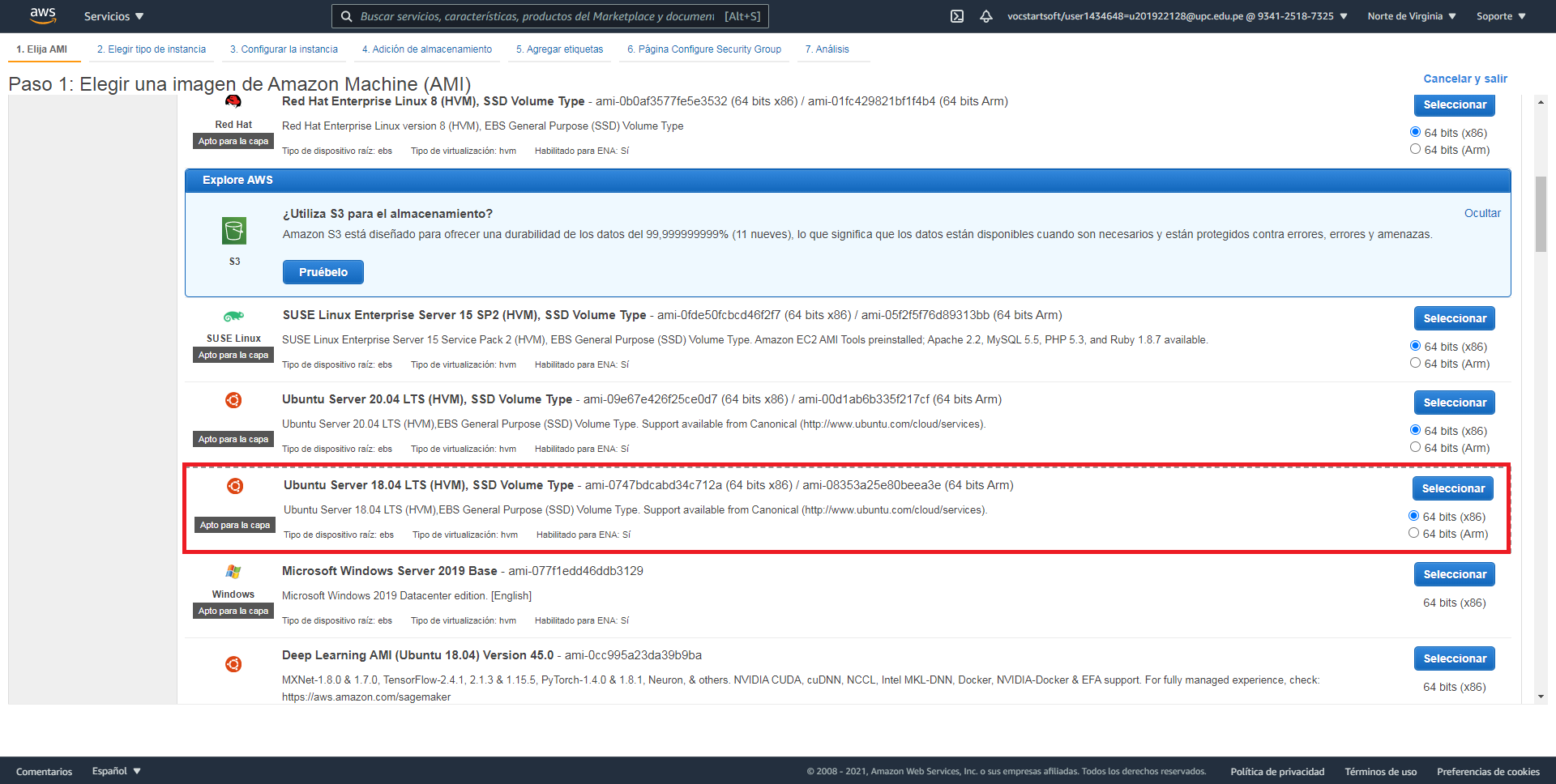
En la parte de sistema operativo, se escogió Linux el cual será trabajado por medio de Ubuntu Server. Esto debido a que la terminal de este sistema operativo posee una serie de características, como el hecho que poseen incorporados paquetes para la instalación de distintos recursos, los cuales facilitarán la correcta implementación de la solución.

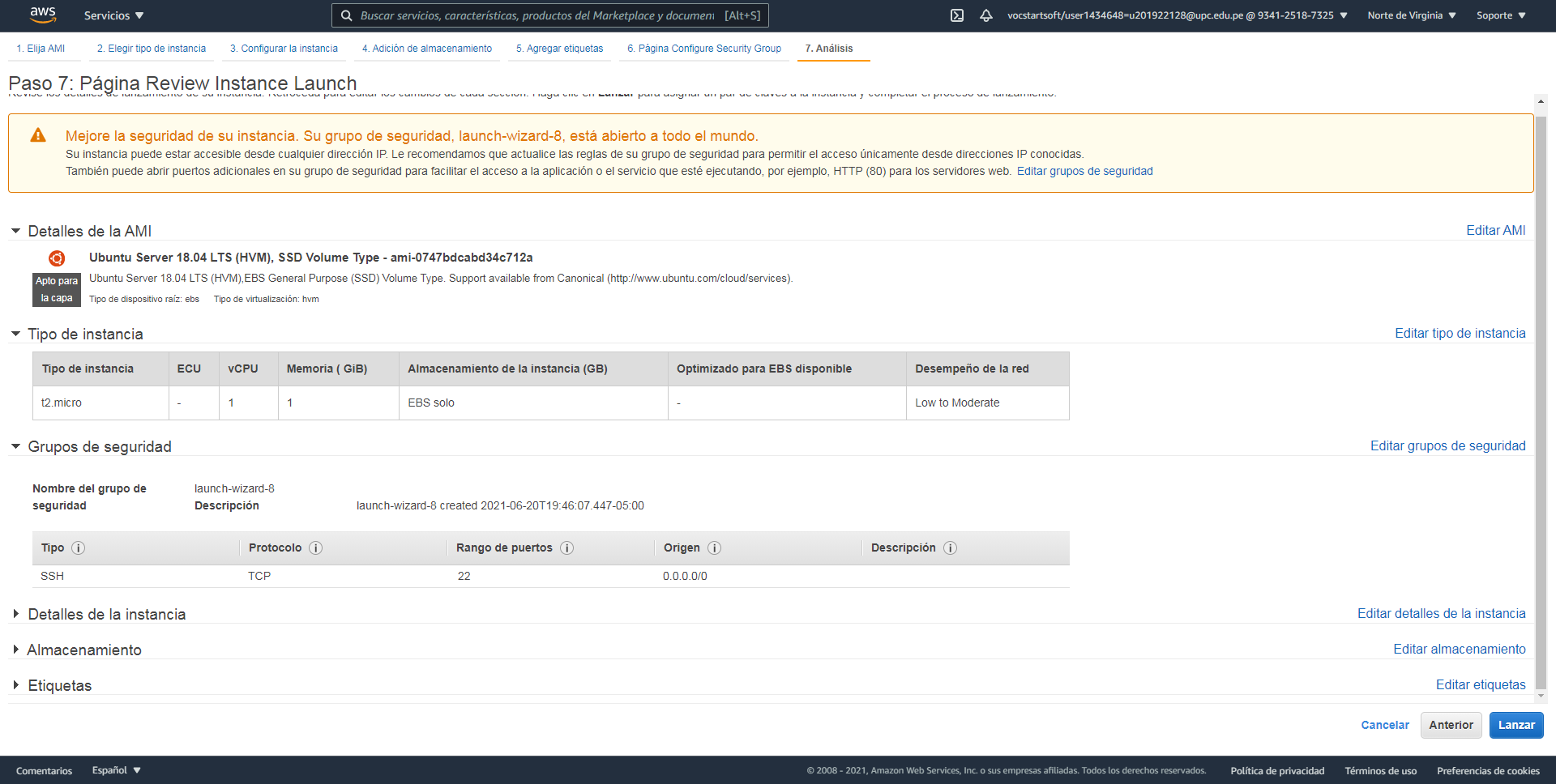
**Instancia de base de datos:**

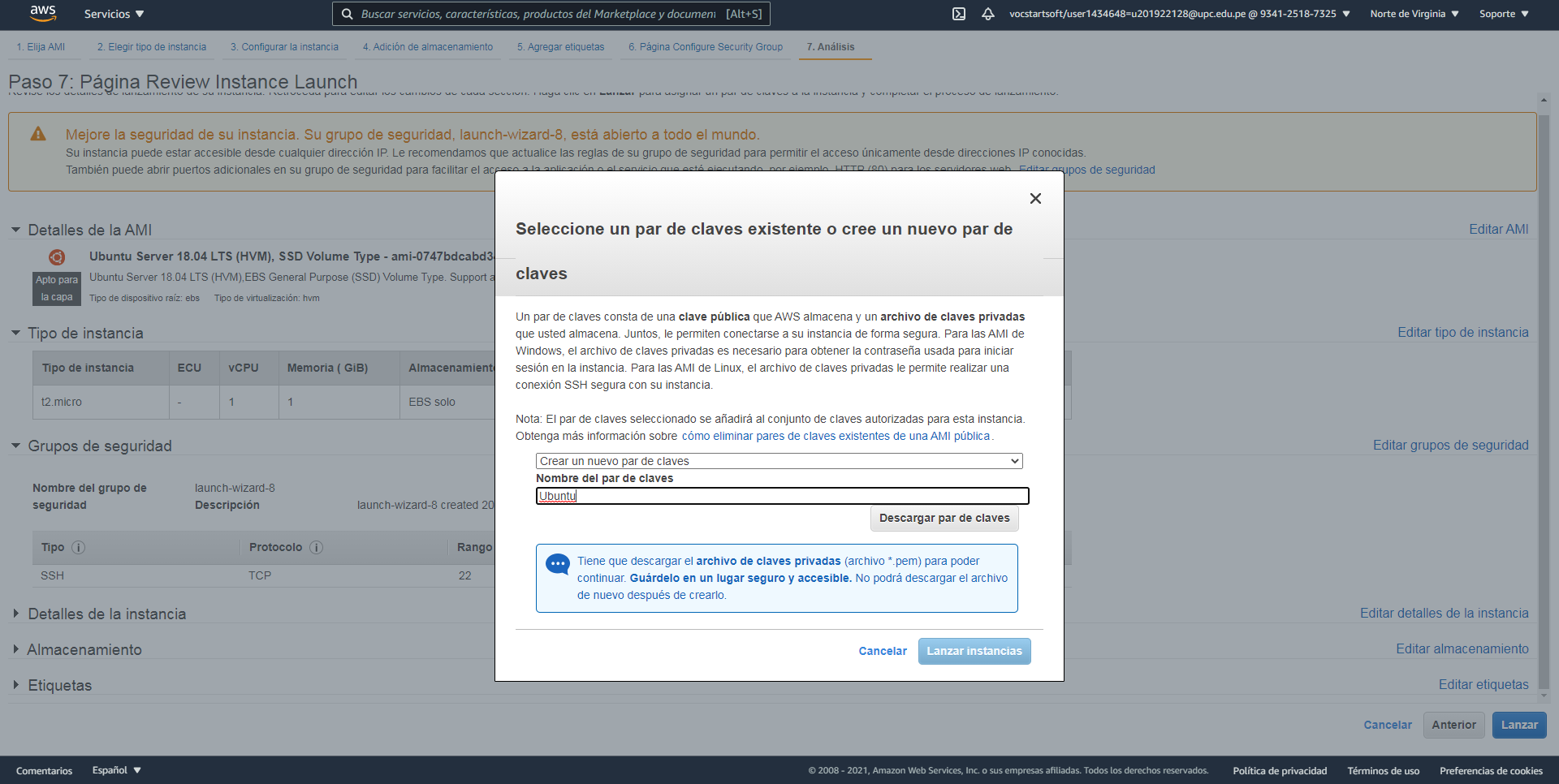
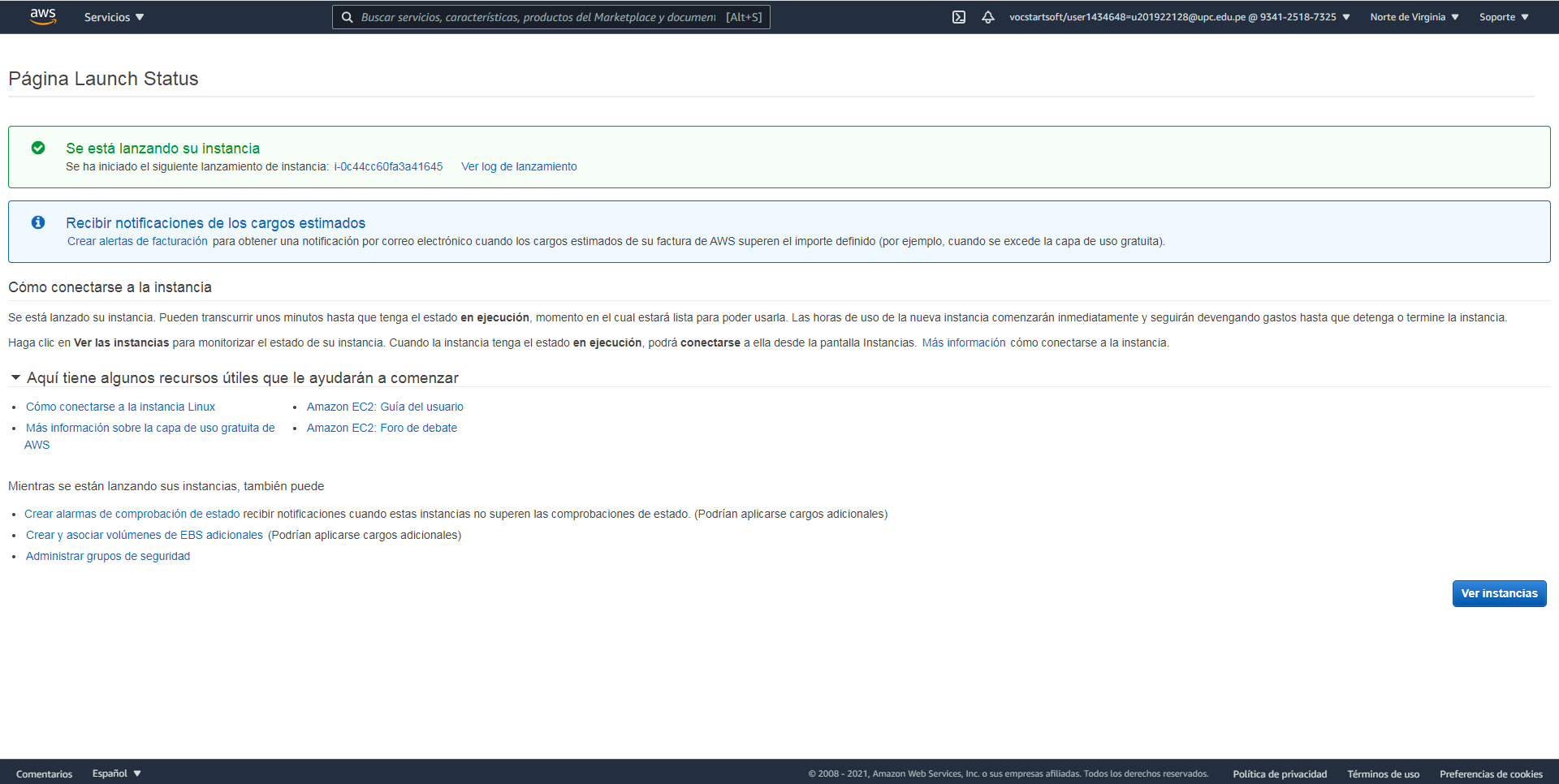
En lo que implica a la base de datos externa para el proyecto, se ha escogido MariaDB, el cual es un gestor de base de datos multiplataforma y gratuito. Esto último es importante, debido a que existen bases de datos que no lo son, como SQL Server (posee una versión gratuita, pero con limitaciones técnicas llamada SQL Express) y el hecho que no sea de pago satisface el requisito de emplear el menor costo posible.

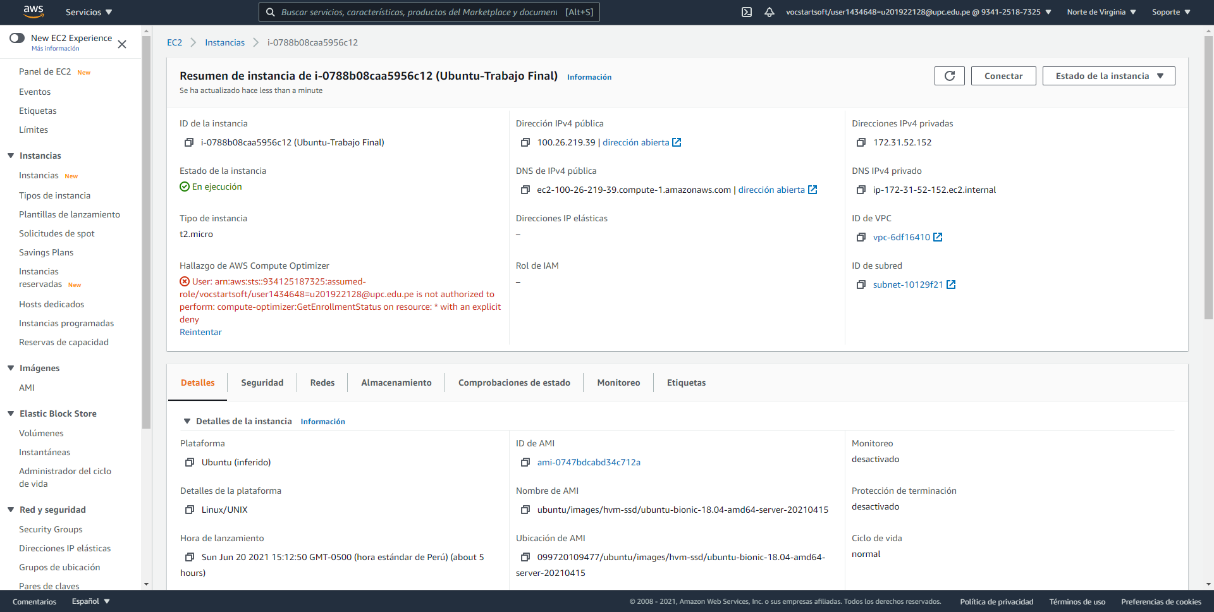
# Implementación de la solución

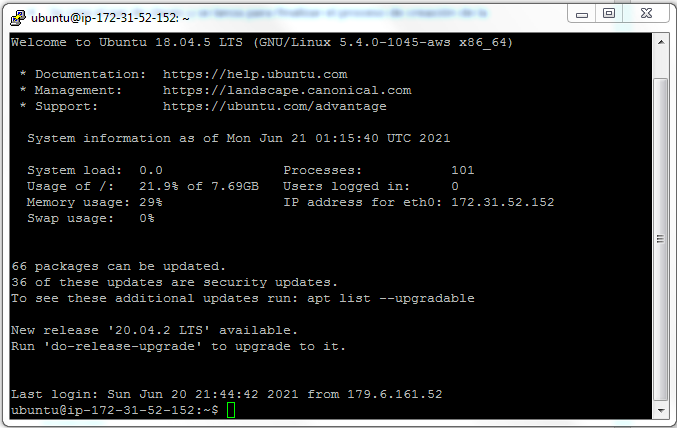
Lo primero para llevar a cabo la solución del caso planteado es habilitar una instancia de computo en Microsoft Azure for Students, para ello se siguió una serie de pasos, los cuales serán explicados detalladamente a continuación:

* Primero, se seleccionó la aplicación base de la máquina virtual, en este caso Ubuntu Server 18.04 LTS.
* Luego, se revisan los detalles:



* Se crea el par de claves y se lanza para finalizar el proceso de creación de la máquina virtual:
* La máquina virtual se habrá creado correctamente y se podrá visualizar detalles de esta:



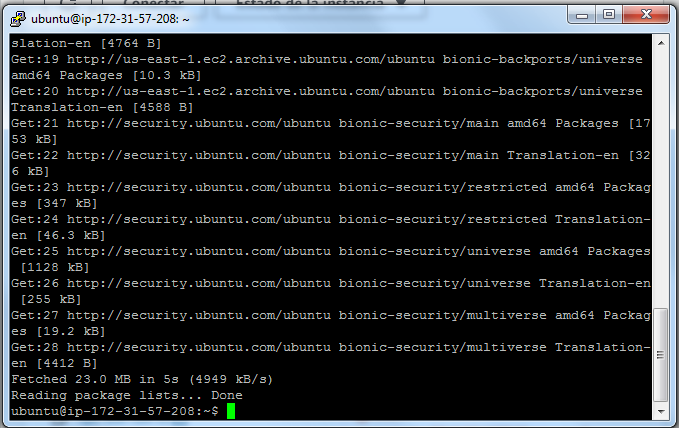
* Luego de esto y para verificar que la máquina virtual funcione correctamente, nos ayudaremos del programa PuTTY. Este es un cliente SSH con el que podremos conectarnos a servidores remotos; en nuestro caso, la máquina virtual recientemente creada.

Ahora, para continuar se ha decidido optar por la instalación de la pila LAMP, ya que es un conjunto de software de código abierto que permiten a un servidor alojar aplicaciones web. El acrónimo LAMP, deriva de Linux, Apache, MariaDB y PHP, los cuales instalaremos en nuestra máquina virtual en ese orden.

Primero, procederemos a instalar Apache en nuestra VM, ya que es un servidor de código abierto y está disponible en múltiples plataformas incluido Linux.

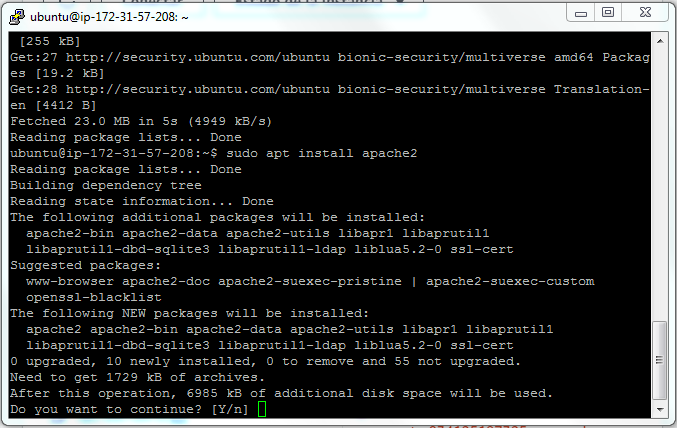
Para instalar Apache en nuestra VM primero tendremos que actualizar los paquetes locales, para ello ingresamos el siguiente comando:

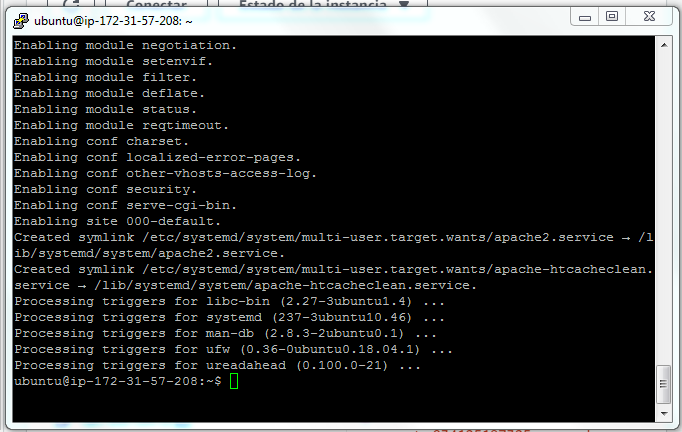
$ sudo apt-get update



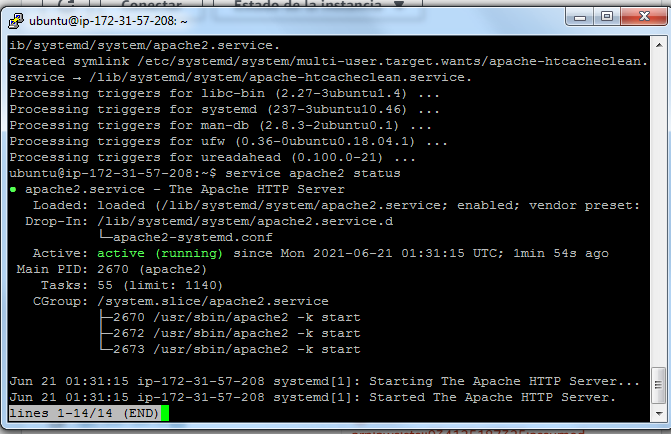
Una vez actualizado los paquetes correctamente, procederemos a instalar el paquete apache2 con el siguiente comando:

$ sudo apt install apache2

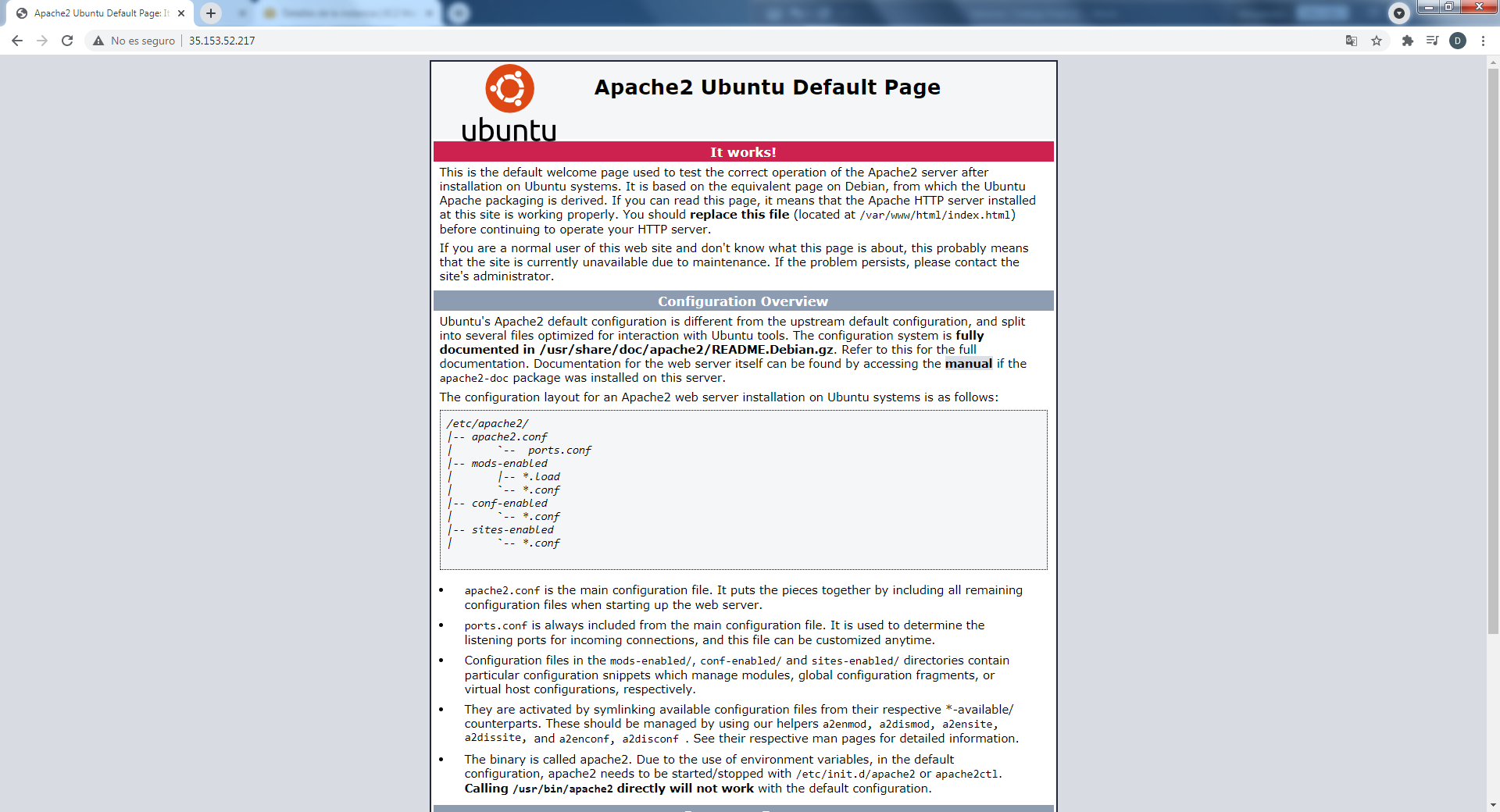
Nos preguntará si deseamos proceder con la instalación, ya que se utilizarán espacio en el disco; en este caso 1730 kB y luego de esto 6986 kB y escribimos “y” para confirmar:

Una vez que se haya finalizado la instalación de los paquetes y las dependencias requeridas, podemos comprobar si se ha instalado correctamente con el siguiente comando:

$ service apache2 status

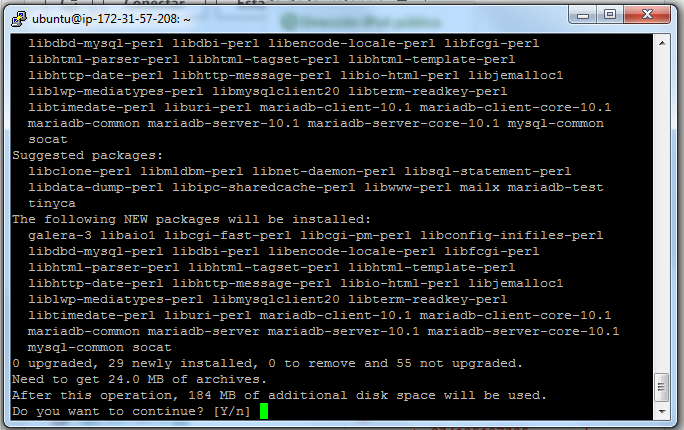


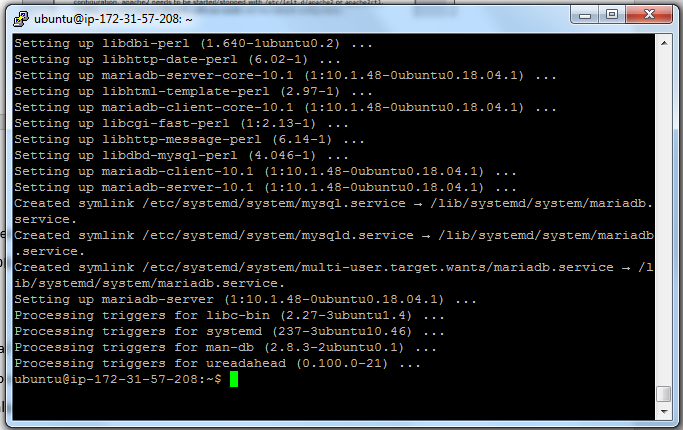
Otra opción para comprobar la correcta instalación es copiando la dirección de IP pública de nuestra VM en el navegador web:



El siguiente paso es instalar nuestra instancia de base de datos externa; en este caso MariaDB. Para esto ejecutamos el siguiente comando:

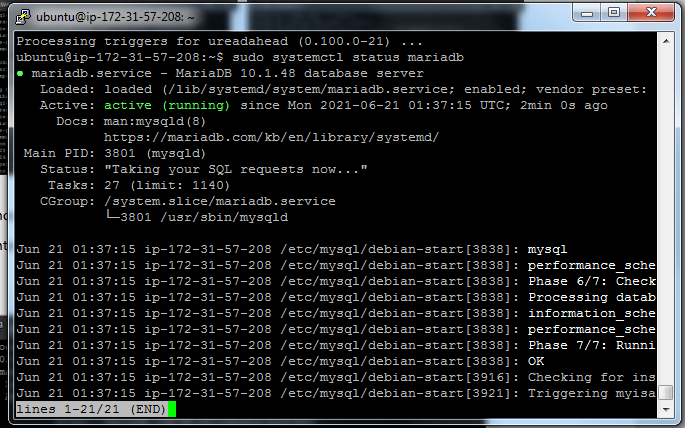
$ sudo apt install mariadb-server

De igual forma que la instalación del Apache, tenemos que confirmar que se utilizará espacio en el disco para la instalación, por lo cual presionaremos “y” para continuar con la instalación:



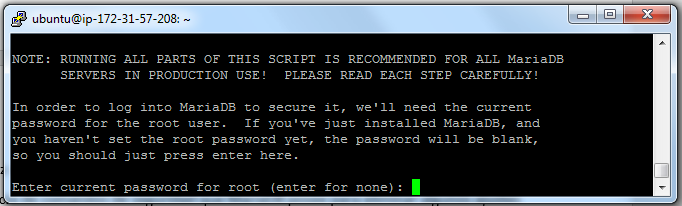
Podemos comprobar que la instalación se ha llevado a cabo correctamente con el siguiente comando:

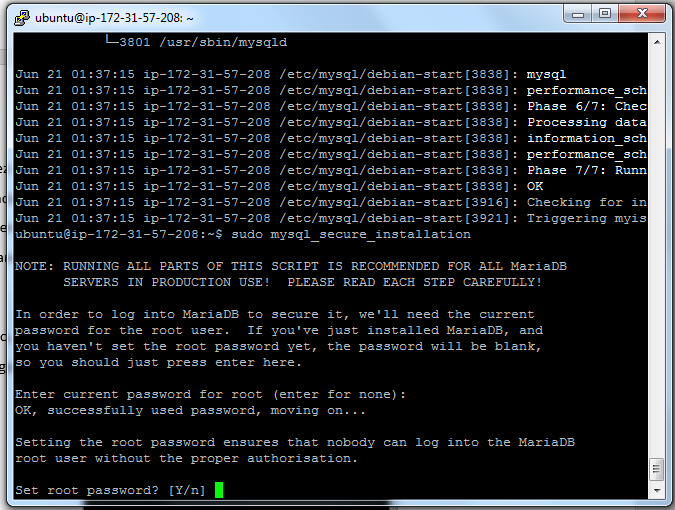
$ sudo systemctl status mariadb

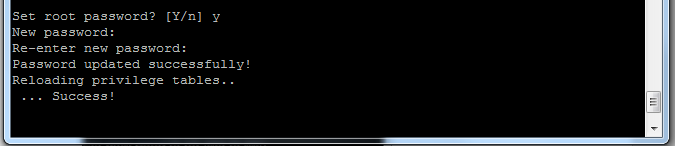


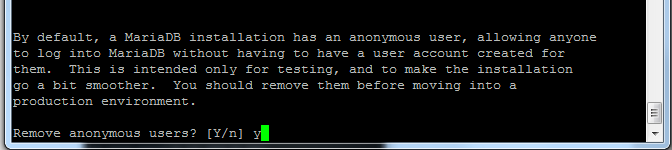
Una vez comprobado que la correcta instalación de MaríaDB, debemos ejecutar una secuencia de comandos de seguridad que MariaDB posee para eliminar algunos ajustes predeterminados y asegurar el acceso a la base de datos. Iniciamos esta secuencia ejecutando el siguiente comando:

$ sudo mysql\_secure\_installation

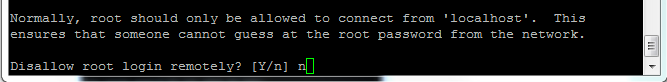
Primero, nos solicitará introducir la contraseña root de la actual base de datos; sin embargo, como acabamos de instalar MariaDB pulsamos “ENTER”

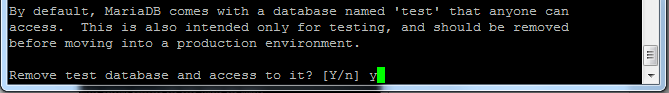
Luego, nos preguntará si queremos crear una nueva contraseña y presionamos “y” para aceptar. Ingresamos la nueva contraseña y la confirmamos.

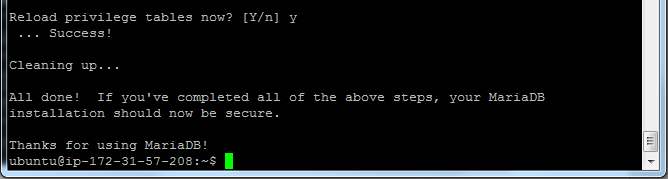


Posterior a esto, se nos preguntará si queremos quitar los usuarios anónimos, por seguridad aceptamos.

Se nos solicitará si queremos deshabilitar el inicio remoto para el usuario root, y pulsamos “no”, ya que necesitamos ingresar.



Luego, nos preguntará si queremos remover las bases de datos de prueba y aceptamos con la tecla “y”.

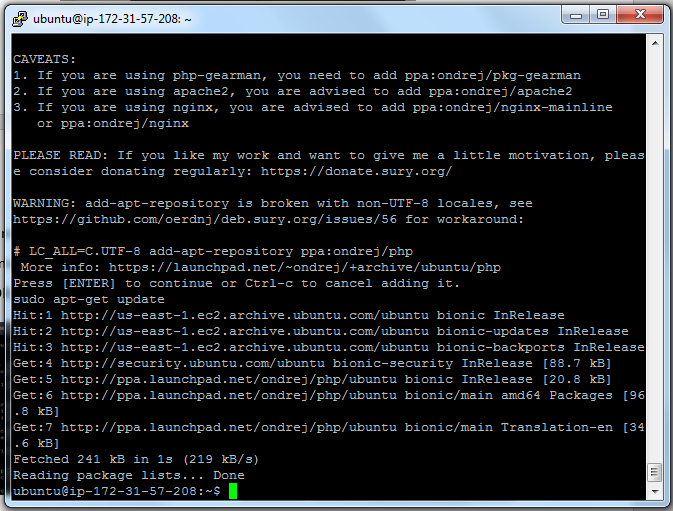
Por último, se nos consultará si queremos actualizar los privilegios de las tablas ahora y aceptamos y habremos finalizado con la secuencia de comandos de seguridad de MariaDB.

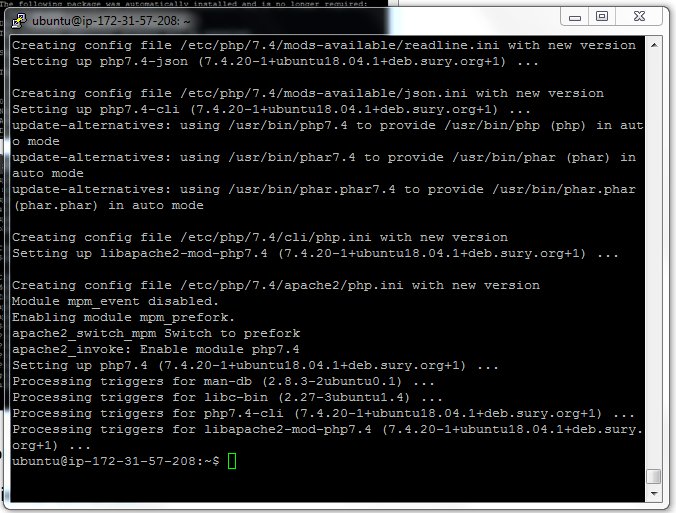
Ahora tenemos que instalar PHP para terminar la instalación del servidor web, pero antes de esto, realizamos la actualización de la última versión de PHP (7.4) con los siguientes comandos:

$ sudo apt -y install software-properties-common

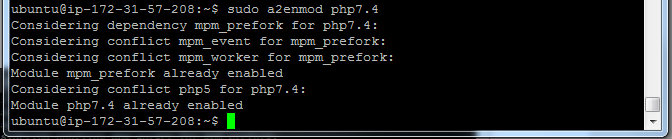
$ sudo add-apt-repository ppa:ondrej/php

$ sudo apt-get update



Ahora podremos realizar la instalación de PHP 7.4 en nuestra máquina virtual con el siguiente comando:

$ sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline

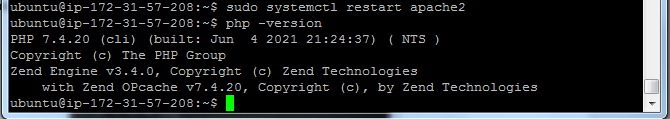
Activamos el módulo de Apache PHP 7.4 para luego no tener problemas. Esto lo realizamos con el siguiente comando:

$ sudo a2enmod php7.4

Reiniciamos el servidor Apache2 con el siguiente comando:

$ sudo systemctl restart apache2

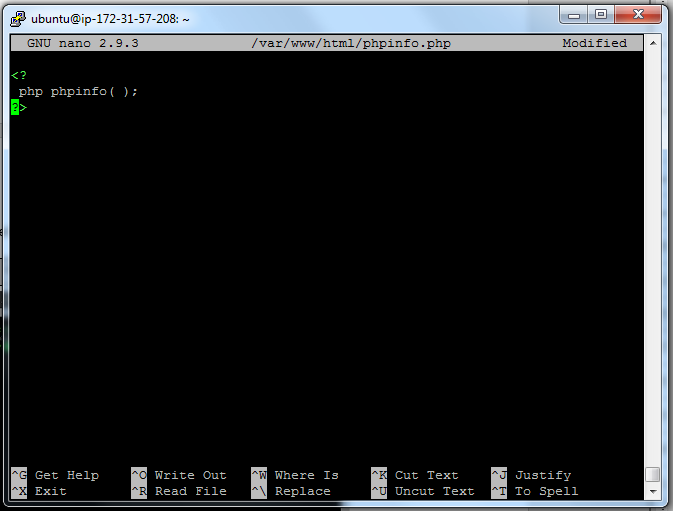
Luego confirmamos que la versión que hemos instalado de php escribiendo el siguiente comando:



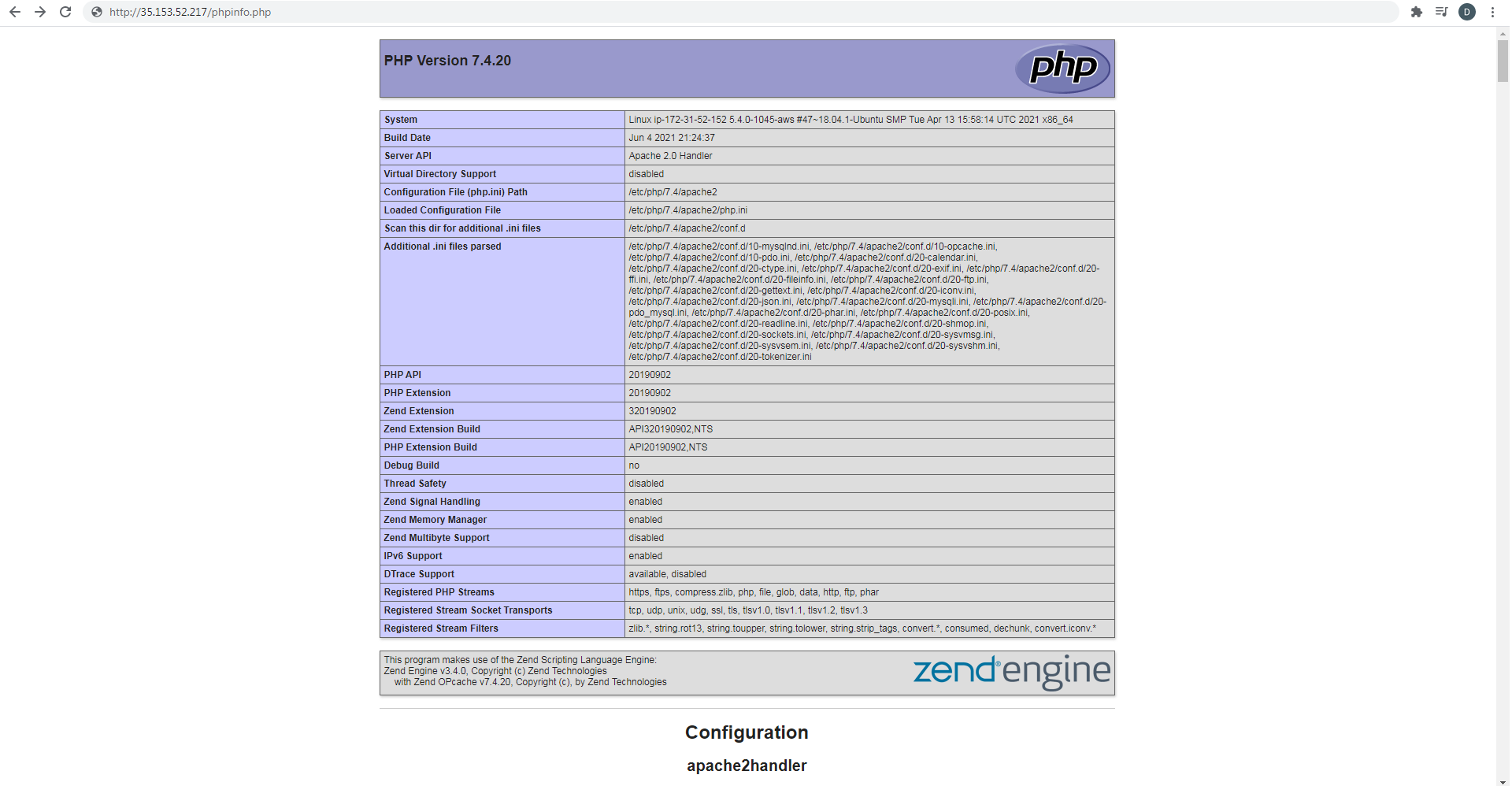
$ php -version

Para comprobar la configuración de PHP 7.4 con Apache 2, creamos un archivo “phpinfo.php” en el directorio root de Apache2 con los siguientes comandos:

$ sudo nano /var/www/html/phpinfo.php

 Se nos abrirá el editor de la terminal, donde escribimos lo siguiente:

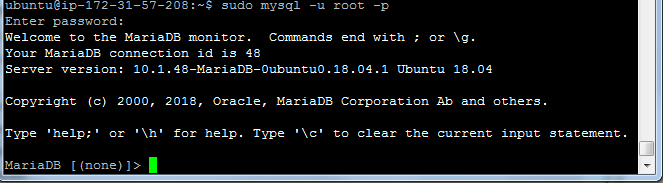
<? php phpinfo( ); ?>

Finalmente podremos ver la información del PHP instalado por medio nuestra misma IP en el navegador, agregándole “/phpinfo.php”, el cual es el nombre del archivo que acabamos de crear.

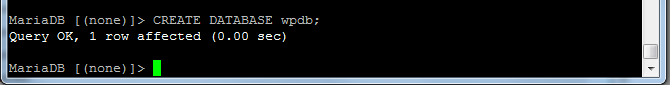
Ahora podremos comenzar a implementar el CMS, el cual en este caso es WordPress. Para esto, primero creamos la base de datos de WordPress por medio del servidor de base de datos ya instalado (MariaDB).

Colocamos el siguiente comando para ingresar a la base de datos:

$ sudo mysql -u root -p

Nos pedirá nuestra contraseña que asignamos al usuario root al momento de instalar nuestra base de datos, y luego de esto ingresamos.

Creamos una base de datos, la cual en este caso la llamaremos “wpdb” por medio del siguiente comando:



CREATE DATABASE wpdb;

Luego creamos un usuario y su contraseña para la base de datos, el cual en este caso llamaremos “wpuser”, con el siguiente comando:

CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'new\_password\_here';

Le asignamos el acceso total de la base de datos:

GRANT ALL ON wpdb.\* TO 'wpuser'@'localhost' IDENTIFIED BY 'user\_password\_here' WITH GRANT OPTION;

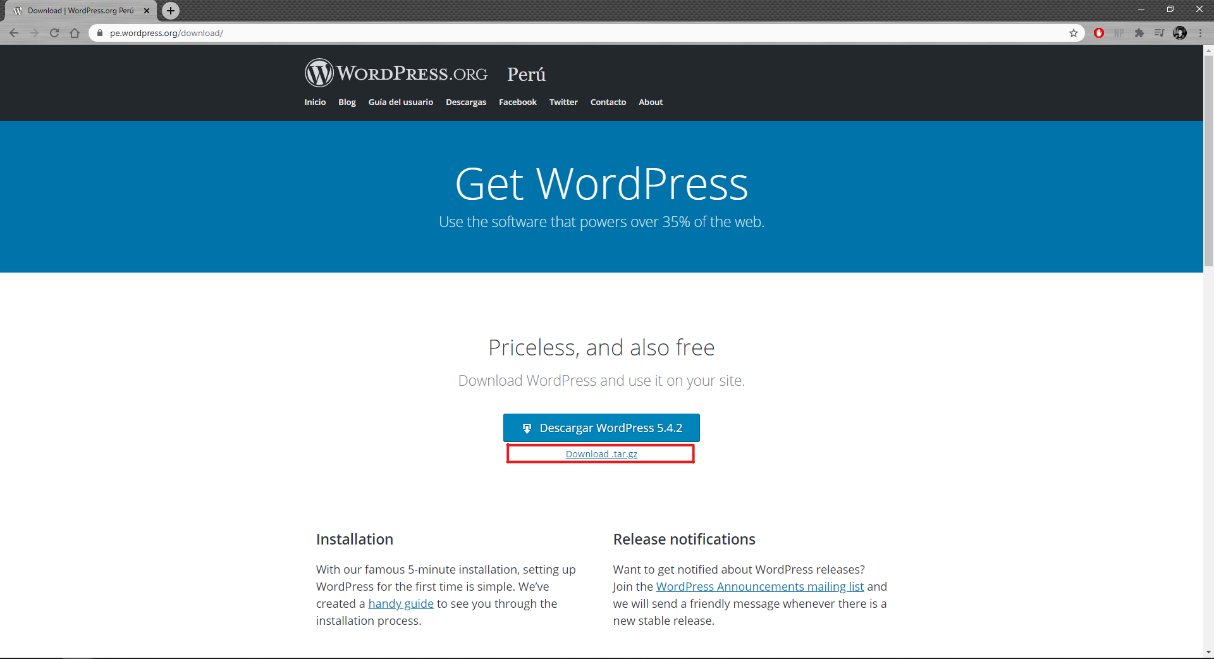
Guardamos lo cambios y salimos de la base de datos:

FLUSH PRIVILEGES;

EXIT;

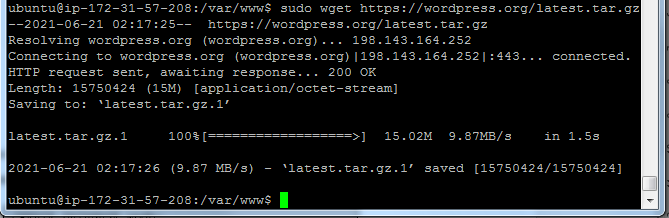
Luego, comenzaremos con la descarga de WordPress por medio de la terminal, para esto visitamos la página web de WordPress: <https://pe.wordpress.org/download/>

Aquí copiamos la dirección del enlace de descarga del archivo .tar.gz, el cual en este caso es: <https://wordpress.org/latest.tar.gz>

Primero nos dirigimos al directorio donde se encuentra nuestro servidor web, con el comando:

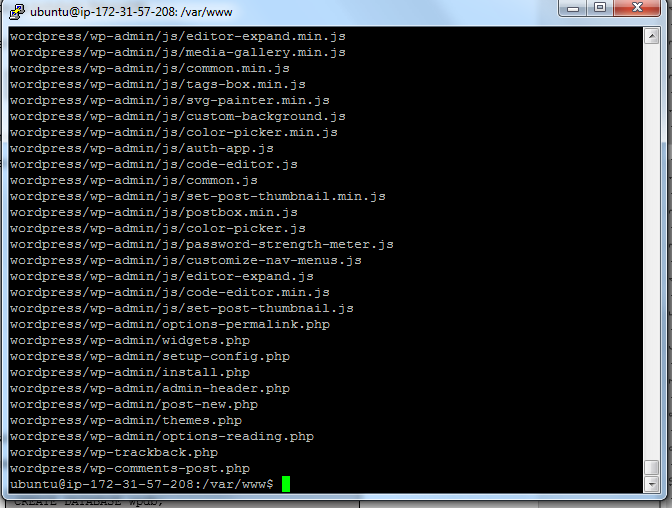
$ cd /var/www

Una vez estemos en este directorio descargamos el paquete de instalación de WordPress, para esto pegamos la dirección del enlace previamente copiada de la página web de WordPress, con el siguiente comando:



$ sudo wget https://wordpress.org/latest.tar.gz

Luego descomprimimos el directorio descargado con el siguiente comando:



$ sudo tar -zxvf latest.tar.gz

Una vez descomprimido nuestro archivo, vamos a cambiar de propietario a este directorio para que otros usuarios puedan modificarlo, esto lo realizamos con el comando:

$ sudo chown -R www-data /var/www/wordpress

Movemos la carpeta a un nuevo directorio root de WordPress:

$ sudo mv wordpress /var/www/html/wordpress

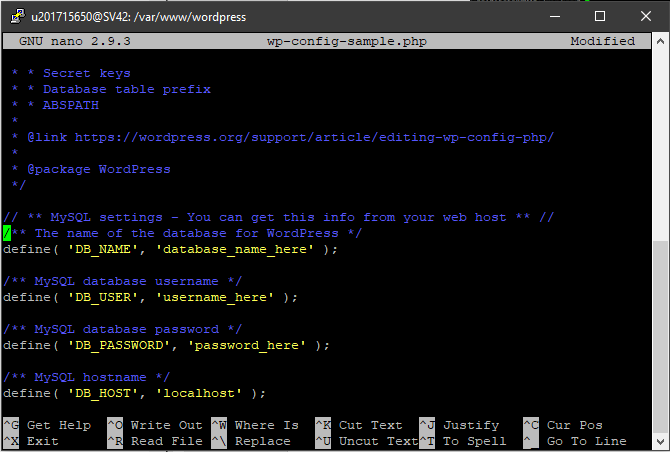
Ahora ejecutamos el siguiente comando para crear el archivo de WordPress “wp-config.php”:

$ sudo mv /var/www/html/wordpress/wp-config-sample.php /var/www/html/wordpress/wp-config.php

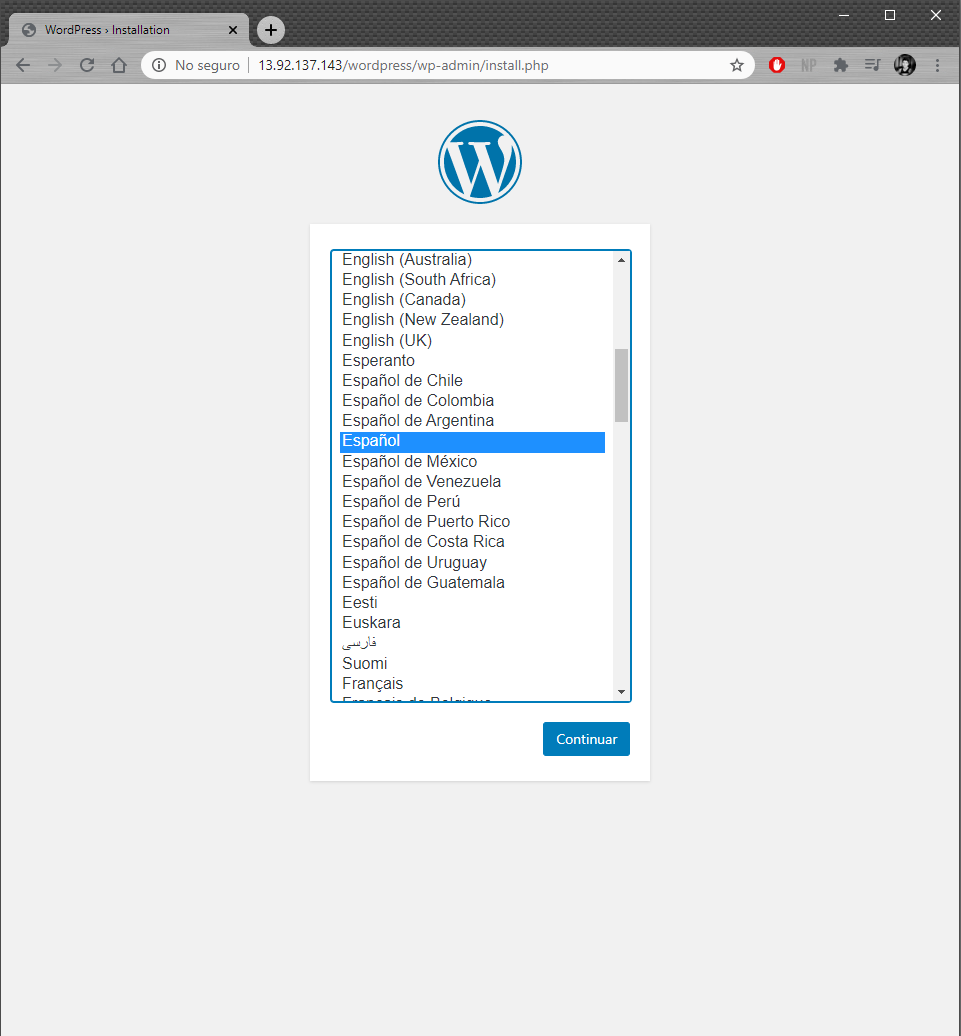
Ahora vamos a modificar el archivo de configuración del WordPress con el siguiente comando:

$ sudo nano /var/www/html/wordpress/wp-config.php

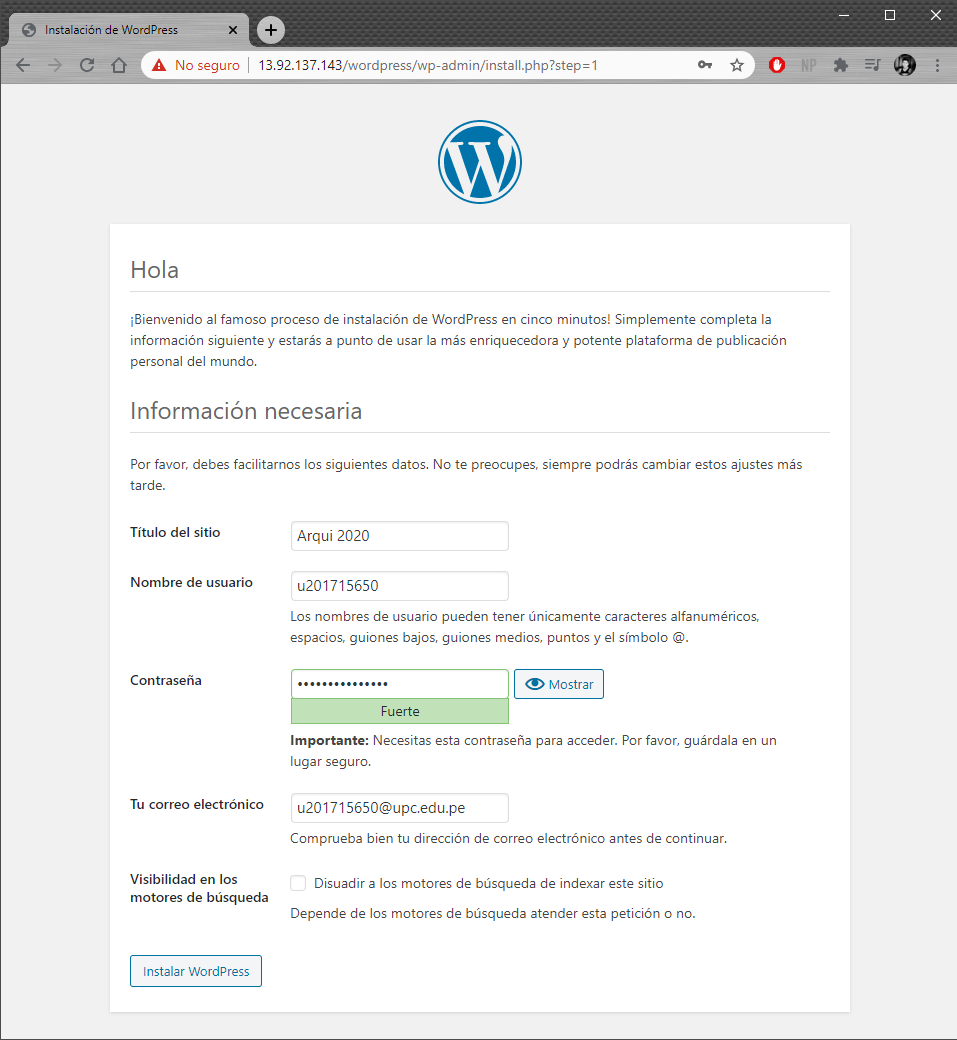
Aquí modificamos los campos ‘database\_name\_here, ‘username\_here’ y ‘password\_here’ con los datos de la base de datos creada anteriormente para WordPress en MariaDB.



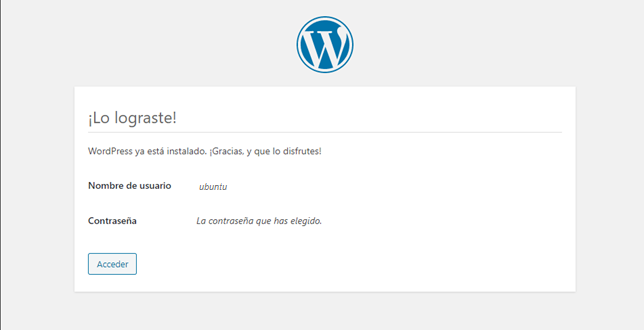
Accedemos por medio del navegador a nuestra dirección IP pública y al lado de esto escribimos “wordpress” y seleccionamos el idioma con el que trabajaremos en WordPress:



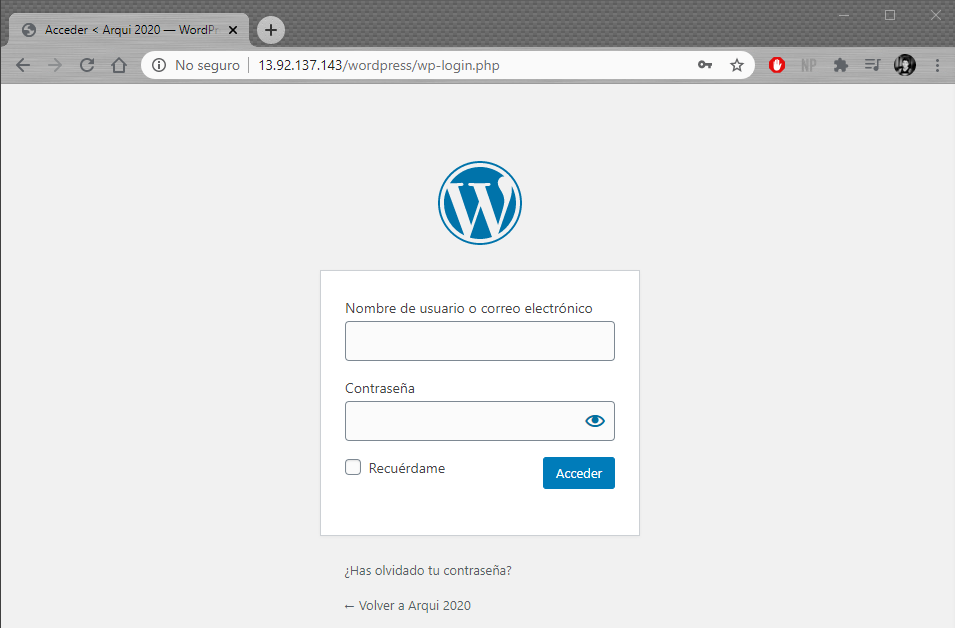
Luego ingresamos el título del sitio web, nombre de usuario, una contraseña, correo electrónico y procedemos a instalar el WordPress.



Nos mostrará un mensaje de confirmación que nos informará que la instalación se ha llevado a cabo de forma correcta:



Se nos pedirá nuestro usuario y contraseña que acabamos de registrar para acceder:



Finalmente, hemos concluido con la instalación de WordPress.

# Presupuesto sustentado

|  |  |  |
| --- | --- | --- |
| **Servicio** | **Software elegido** | **Precio** |
| Máquina virtual | B1S estándar | $ 0.0104 (por hora) |
| Almacenamiento | Premium SSD 30 GiB | $ 5.28 (por mes) |
| Base de datos | MariaDB | Gratuito |
| CMS | WordPress | Gratuito |

Máquina Virtual:

Para la máquina virtual se escogió una instancia B1S estándar, ya que posee 1 vCPU, 1GB RAM y almacenamiento temporal de 4GiB, el cual es el suficiente para este proyecto. Además, tiene un costo de $0.0104 por hora; por lo cual, el costo es el mínimo y va de acuerdo con el requisito de considerar el menor costo posible.

Almacenamiento:

En este caso se escogió el mínimo posible a la hora de la creación de la máquina virtual, el cual es un disco SSD Premium de 30GiB. Este disco al ser uno de estado sólido nos brinda una mayor rapidez en comparación al disco duro y es suficiente para la infraestructura creada. Este SSD cuesta $5.28 al mes. (Ver anexo 4)

Base de datos:

Se escogió la base de datos MariaDB ya que es un gestor de base de datos de licencia abierta; es decir no tiene costo alguno, y esto facilita el ir acorde con llevar el menor costo posible.

Gestor de contenido:

Para el CMS se empleó WordPress el cual es open Source; por lo cual no tendremos que pagar por descargar este software.

# Conclusiones

# Anexos

# Bibliografía

* Alfredo Carcamo (2020). *Instalación de WordPress en Azure con Ubuntu 18.04 LTS*. Recuperado de: <https://bit.ly/3fJfW4R> [Consulta: 21 de Junio del 2020].
* Computing for Geeks (2020). *How To Install PHP 7.4 on Ubuntu 20.04/18.04/16.04.* Recuperado de: <https://bit.ly/2YOPqQI> [Consulta: 21 de Junio del 2020].
* DesdeLinux (2020). *¿Cómo instalar WordPress en Ubuntu 18.04 y derivados?* Recuperado de: <https://bit.ly/3fEJ7pM> [Consulta: 21 de Junio del 2020].
* Dev/Central (2010). WILS: *How can a load balancer keep a single server site available*. Recuperado de: <https://bit.ly/2VDYNC2> [Consulta 27 de Junio del 2020].
* Digital Ocean (2018). *Cómo instalar en Ubuntu 18.04 la pila LAMP — Linux, Apache, MySQL y PHP.* Recuperado de: <https://do.co/3fLNAHj> [Consulta: 21 de Junio del 2020].
* Digital Ocean (2020). *Cómo instalar la pila Linux, Apache, MariaDB, PHP (LAMP) en Debian 9.* Recuperado de: <https://do.co/3fLNKhT> [Consulta: 21 de Junio del 2020].
* Guiadev (2018). *MySQL vs SQL Server.* Recuperado de: <https://bit.ly/3dmGsj4> [Consulta: 14 de Junio del 2020].
* Guiadev (2016). *MariaDB vs MySQL: ¿cuál debo elegir?* Recuperado de: <https://bit.ly/2YUe02R> [Consulta: 14 de Junio del 2020].
* Hostinger (2019). *Cómo instalar WordPress en Ubuntu 18.04 usando el Stack de LAMP.* Recuperado de: <https://bit.ly/3hQpD3m> [Consulta: 21 de Junio del 2020].
* IONOS (2020). *MariaDB vs. MySQL.* Recuperado de: <https://bit.ly/2Cts2Rw> [Consulta: 14 de Junio del 2020].
* Kelly D. (2016). *7 things that nobody told you about load balancers*. Recuperado de: <https://bit.ly/31BJXQy> [Consulta 27 de Junio del 2020].
* Microsoft (2020). *B-series burstable virtual machines sises.* Recuperado de: <https://bit.ly/2BAR3tn> [Consulta: 06 de Junio del 2020].
* Microsoft (2020). *Componentes de Azure Load Balancer*. Recuperado de: https://bit.ly/3dPHpAA [Consulta 27 de Junio del 2020].
* Microsoft (2020). *General purpose virtual machines sises.* Recuperado de: <https://bit.ly/2MCZtTu> [Consulta: 06 de Junio del 2020].
* Microsoft (2020*). Inicio rápido: Instalación de SQL Server y creación de una base de datos en Ubuntu*. Recuperado de: <https://bit.ly/37LNrB0> [Consulta: 21 de Junio del 2020].
* Microsoft (2019). *Tutorial: Equilibrio de la carga de tráfico interno con un equilibrador de carga básico en Azure Portal*. Recuperado de: <https://bit.ly/38iG5F9> [Consulta 27 de Junio del 2020].
* Microsoft (2019). *Tutorial: Instalación de un servidor web LAMP en una máquina virtual Linux en Azure.* Recuperado de: <https://bit.ly/2Yk5XNC> [Consulta: 21 de Junio del 2020].
* Microsoft Azure (2020). *Pricing Calculator*. Recuperado de: <https://bit.ly/2V57Bkh> [Consulta: 21 de Junio del 2020]
* Nagarajan M. (2019). *What is load balancer and how it works?* Recuperado de: <https://bit.ly/31z1fxv> [Consulta 27 de Junio del 2020].
* Platzi (2017). *Instalar LAMP Stack en Linux (Ubuntu 14.04).* Recuperado de: <https://bit.ly/3dloNbz> [Consulta: 21 de Junio del 2020].
* PuTTY (2020). *PuTTY FAQ.* Recuperado de: <https://bit.ly/3h06RpZ> [Consulta: 06 de Junio del 2020].
* Ubunlog (2020). *LAMP, instala Apache, MariaDB y PHP en Ubuntu 20.04.* Recuperado de: <https://bit.ly/37QghjE> [Consulta: 21 de Junio del 2020].
* Website for Students (2018). *Install WordPress on Ubuntu 16.04 | 17.10 | 18.04 with Apache2, MariaDB, PHP 7.2 and Let’s Encrypt SSL/TLS*. Recuperado de: <https://bit.ly/3138uxx> [Consulta: 21 de Junio del 2020].