

**MANUAL**

**INSTALACIÓN**

**PROYECTO**

**Proyecto PR101 - Fase 1**

DAW-IT00168-00-Proyecto PR101 - Fase 1

Autor: Andrés Cortés Escobedo

**INDICE**

[**Requisitos del sistema** 1](#_Toc66822583)

[**Servidor usado para estar en la nube** 1](#_Toc66822584)

[**Scripts que hay que ejecutar** 9](#_Toc66822585)

[**Instalar Apache, Php 7.4 y mysql** 12](#_Toc66822586)

[**PHP 8.0 en apache y PHP 7.4 en la línea de comandos por defecto.** 13](#_Toc66822587)

[**INSTALAR PHPMYADMIN** 14](#_Toc66822588)

[**Clave pública de nuestro usuario jdaw, para que se conecte a nuestro servidor.** 15](#_Toc66822589)

[**Referencias** 18](#_Toc66822590)

# **Requisitos del sistema**

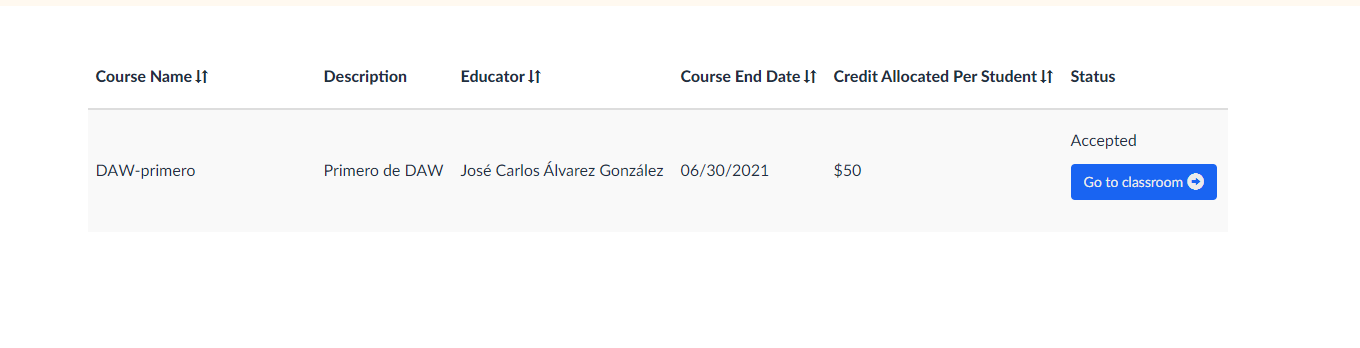
No necesitamos ningún requisito del sistema, solamente un dispositivo que tenga internet y se pueda conectar a la página web.

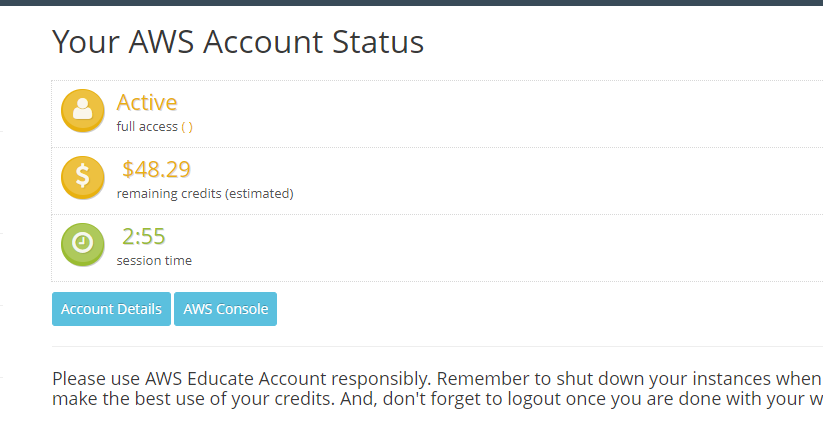
# **Servidor usado para estar en la nube**

Lo que he usado para instalar la aplicación web es un servidor de **‘Amazon Web Services’**, con una cuenta de ‘aws educate’ <https://www.awseducate.com/student/s/>, seleccionamos la Ubuntu server 20.04 LTS, que es de 64 bits.

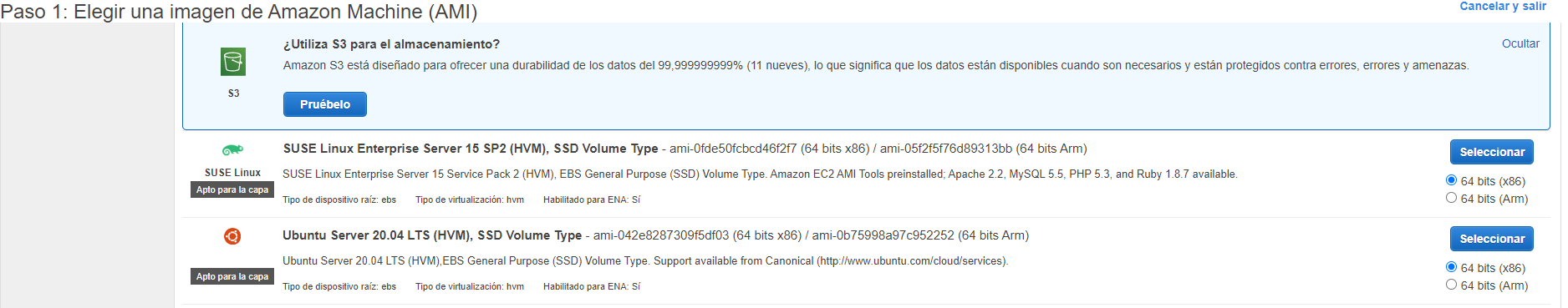
(esta parte corresponde mas bien a la parte del desarrollador/programador parte privada)

Tras iniciar sesión, en mi caso voy a la classroom que tengo y hacemos clic en ‘Go to classroom’

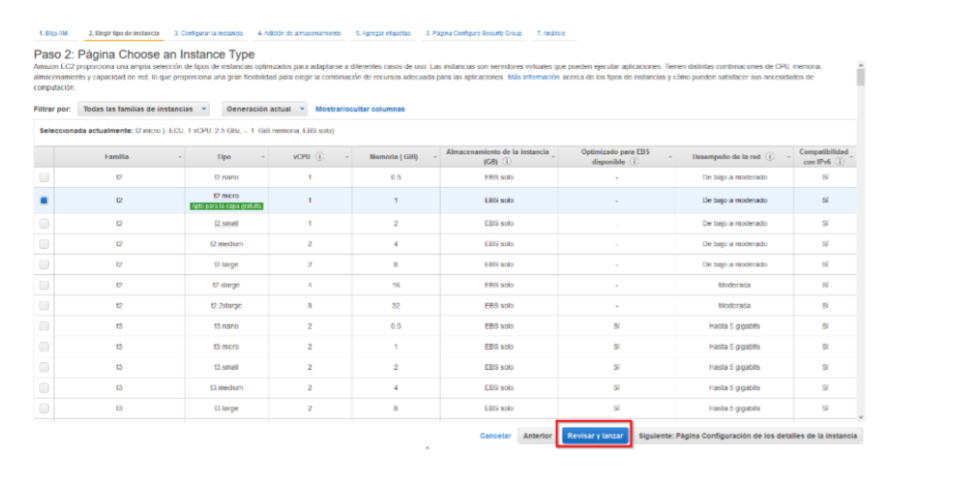




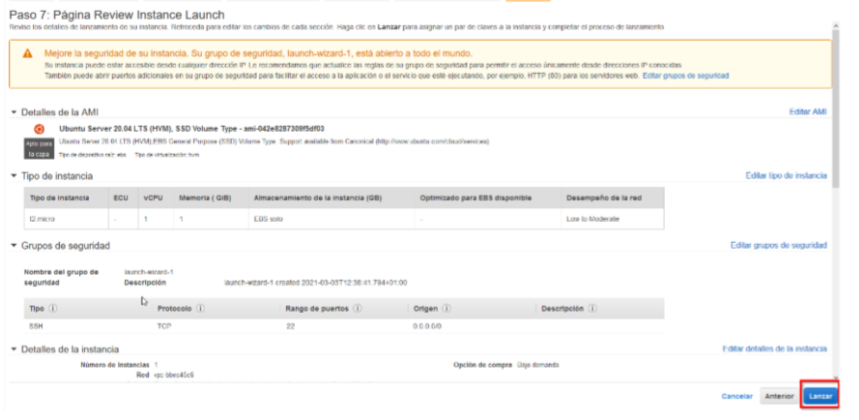
Le damos a ‘AWS Console ’

Hacemos clic en ‘Ejecute una máquina virtual ’ y Seleccionamos esta máquina.

Ahora toca seleccionar el tipo de instancia, una instancia en este caso es una maquina virtual alojada en este caso en AWS, contiene aplicaciones instaladas predeterminadas y configuraciones del sistema operativo.

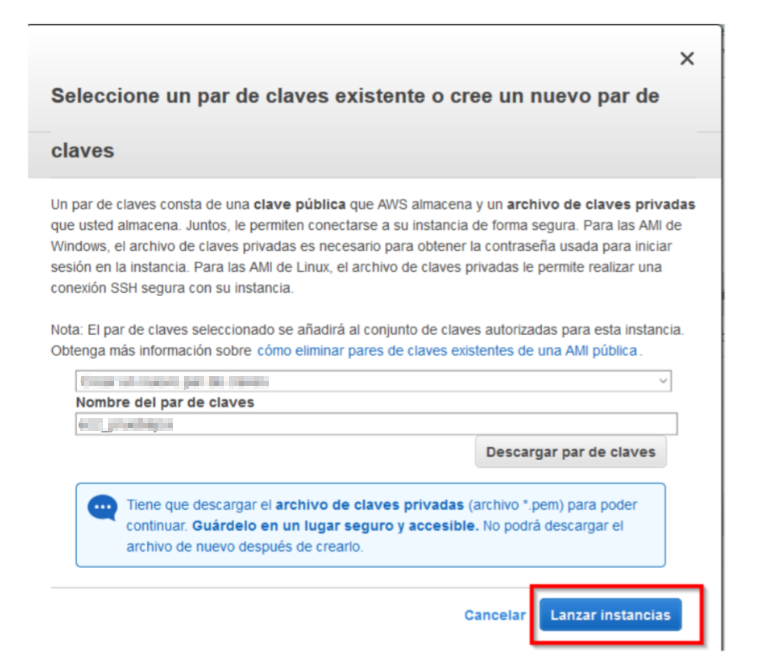


Seleccionamos esta instancia.

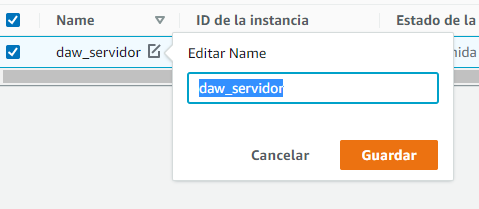


La lanzamos.

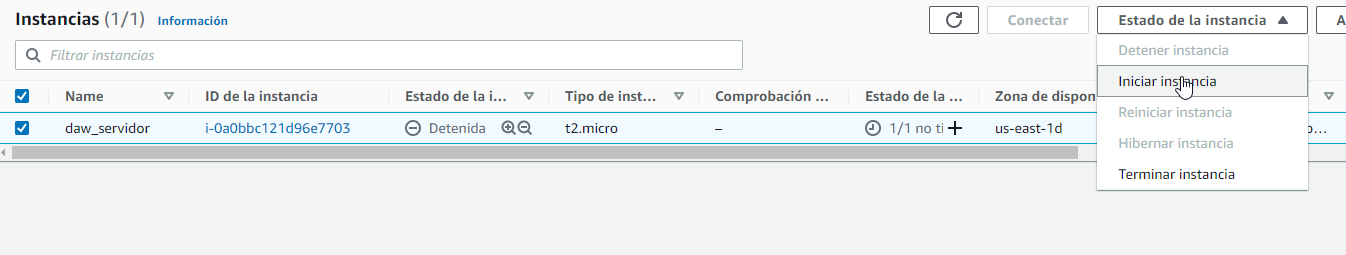
Con ello nos pedirá si queremos un par de claves de rsa, para que nos de la clave pública y con ello conseguir conectarnos a nuestra instancia.



Con esto ya si podemos lanzar nuestra instancia, después de esto hacemos clic en ver instancias.

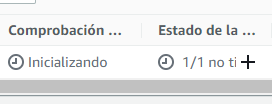


Nos saldrá nuestro panel y le damos a instancias, después ello veremos las instancias que tenemos en este caso solamente tenemos una, ponemos el ratón encima del nombre y se lo cambiamos para poder identificarla.

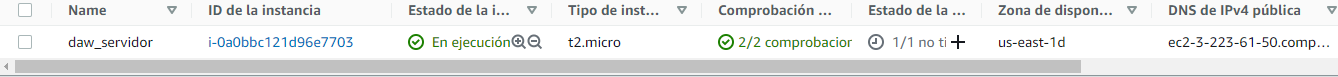


La seleccionamos y en estado de la instancia le damos a iniciar instancia esperamos unos minutos actualizamos página y ya veremos que esta iniciada.

Antes de que se inicie hará las compobraciones necesarias y cuando este 2/2 significa que ha hecho las comprobaciones y en el estado veremos iniciada.



Ahora mismo esta en estado de inicialización aún tenemos que esperar.



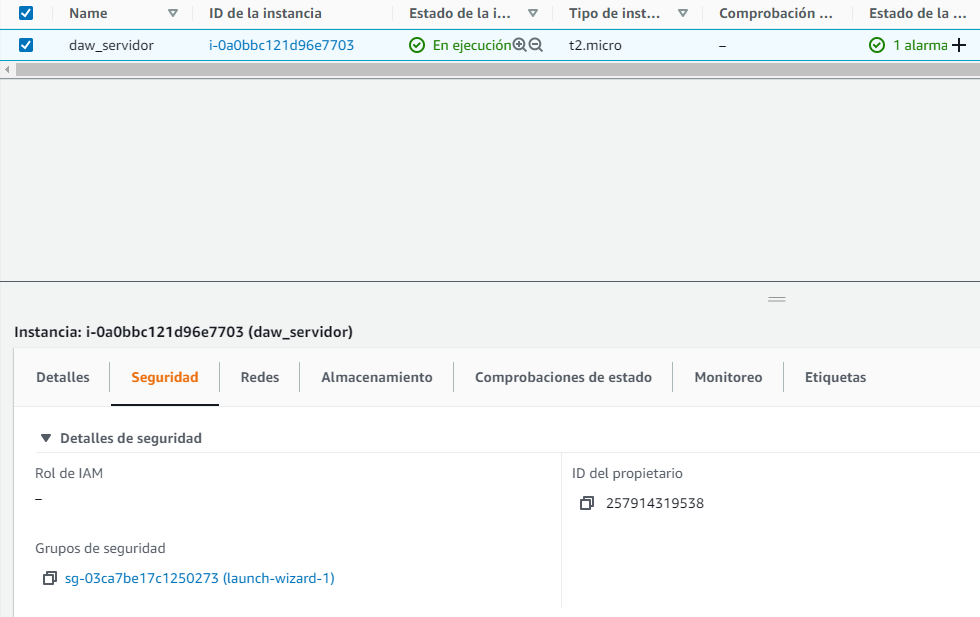
Como vemos las comprobaciones ya están hechas, y el estado de la instancia esta ya en ejecución, podemos ya entrar en nuestro servidor virtual.

**IMPORTANTE : Cuando hayamos terminado de usar la máquina y la queremos apagar para que no nos quite el total del saldo si no esta en uso, debemos darle a DETENER INSTANCIA, si le damos a terminar instancia será eliminada.**

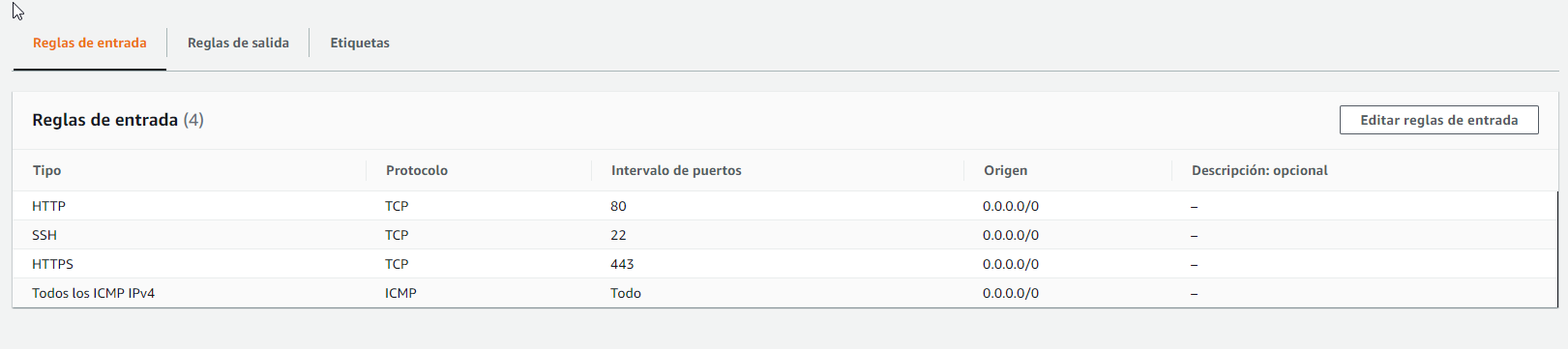
Si nuestro ordenador usa sistema operativo Windows podemos usar git bash ya que usamos Git si tuvieses Linux como sistema no tendrías que instarte ningún programa de terceros, bueno tendrías que instalarte git si no lo tuvieses también.

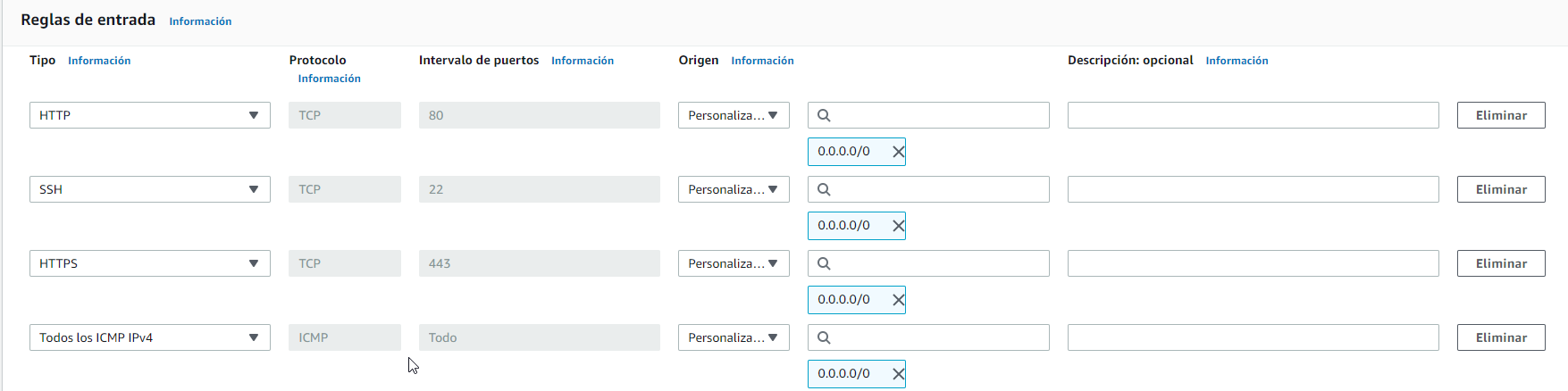
Antes de entrar mediante SSH, vamos hacer unas pequeñas configuraciones a nuestro servidor virtual.

Como nuestro servidor virtual de Ubuntu, su uso va a ser de servidor Web tendremos que darle unas reglas para que los usuarios puedan acceder mediante http y https, para conectarnos nosotros median ssh y poder hacernos ping para comprobar por ejemplo si tenemos conexión a internet.



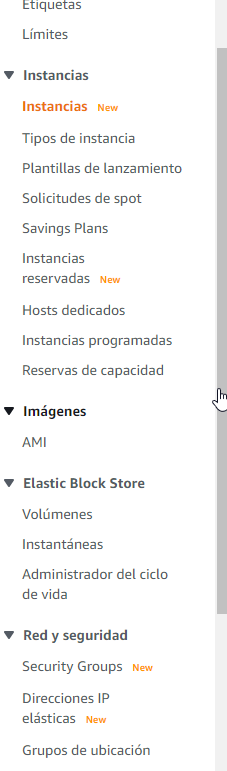
Donde estábamos actualmente en nuestras instancias si bajamos hacia abajo vemos que pone id y nuestra id mas el nombre del servidor entre paréntesis, tenemos que dale a la pestaña de seguridad y hacer clic en el grupo de seguridad, que estará de color azul.



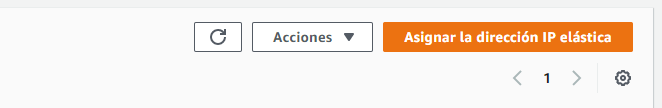
Aquí podemos ver que yo ya las reglas de entrada ya las tengo hechas, para hacerlas tenemos que hacer clic en ‘Editar reglas de entrada’

Tendriamos que seleccionar los protocolo que seria HTTP, SSH, HTTPS Y Todos los ICMP IPv4 protocolo en todos tcp el numero de puerto y el origen 0.0.0.0, para que las aceptemos todas y guardamos reglas.

Ahora solos nos queda darle una ip elástica , ya que no podemos darle una ip fija por que nos costaría dinero, con esto conseguimos darle una dirección ip pública a la que se pueda tener acceso desde intenet y con ello asociar nuestra dirección ip a esta, por lo que nunca cambiaría esta dirección ip.



Nos vamos a `Direcciones IP elásticas `.



Hacemos clic en `Asignar la dirección IP elástica`.

En el siguiente apartado lo dejamos todo como esta y le damos a asignar, seleccionamos nuestra máquina virtual y ya tendríamos nuestra dirección IP elástica.

Nos vamos otra vez a Instancias, podemos ver que en el apartado IP elástica ya tenemos una dirección IP, que la podemos usar tanto para hacer las conexiones de ssh o http/s y ping con esa IP, ahora solo nos hace falta darle un DNS.

En mi caso como tengo cuenta Github students, tengo un dominio gratis en [.Tech](https://get.tech/).

Después de haberle asignado un DNS a nuestro servidor ahora toca entrar en el y hacer las siguientes configuraciones.

# **Scripts que hay que ejecutar**

Que se meta en el enlace de la página web (<http://proyectos-andres.tech/pr101>).

Si eres desarrollador/programador estos son los siguientes scripts que he hecho para la configuración de la máquina.

Tras habernos dado AWS el par de claves.

Nos habrá dado la clave pública de nuestro servidor para conectarnos a ella tenemos que ir a la carpeta de nuestro usuario, si estamos en git bash o en una consola Unix haciendo un cd ~, nos lleva directamente a él.

Creamos un directorio que se llame .ssh (con el punto al principio significa que esta en oculto, por lo que hay que hacer el comando ls -a para ver los directorios ocultos )

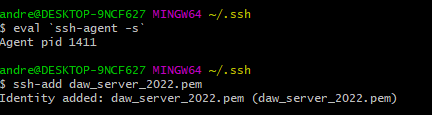
Nos copiamos la llave pública y la dejamos dentro de este directorio, guardar bien esta llave ya que es muy importante para entrar en el servidor.

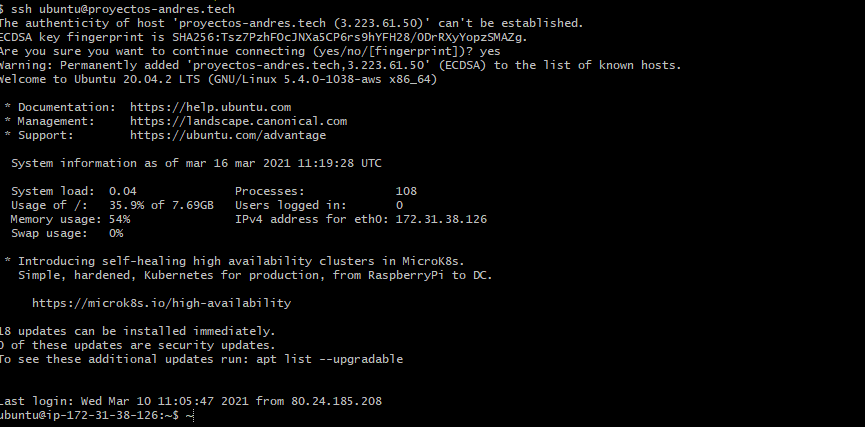
Para conectarnos mediante ssh :

Eval `ssh-agent -s` 🡪 para crear el proceso, luego nos dará el numero de que el proceso se ha creado.

Ssh-add clave publica 🡪 para añadirla a ese proceso .

Ssh usuario del servidor@ip/dns del servidor para conectarnos a él.





Tras habernos conectado actualizaremos nuestro repositorios y nuestra instancia :

Sudo apt-get update && sudo apt-get upgrade -y

Instatalamos los paquetes básicos en español :

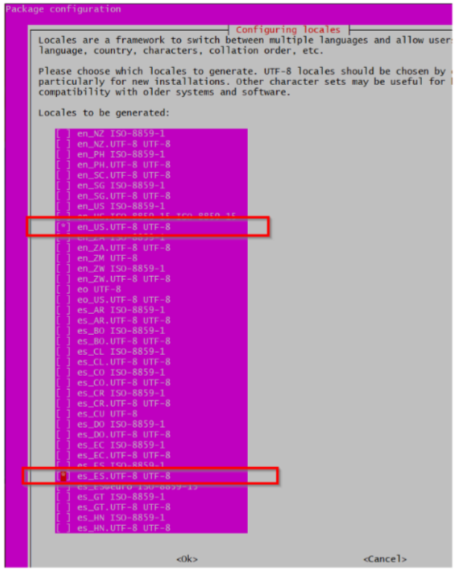
Sudo apt-get install language-pack-es -y

Configuramos nuestro locale para trabajar en español :

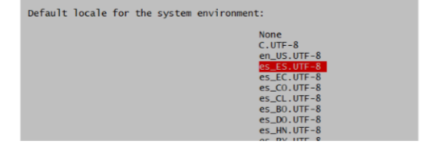
Sudo dpkg-reconfigure locales

Dejamos seleccionado el locale en\_US.UTF-8 UTF-8

Seleccionamos con el espacio el de Español España es\_ES.UTF-8 UTF\_8 :



Seleccionamos el locale por defecto es\_ES.UTF-8



Para comprobar que se está aplicando el locale que queremos, ejecutamos el comando locale

Veremos que no lo ha hecho aún para ello lo que debemos hacer es salir mediante ssh haciendo un exit

Volvemos a entrar y ya estará todo correctamente.

Para configurar nuestra zona horaria :

Sudo dpkg-reconfigure tzdata

Escogemos Europa y en zona horaria Madrid.

Y si ejecutamos date

Vemos que estaría la hora correcta

## **Instalar Apache, Php 7.4 y mysql**

Ejecutamos el siguiente comando :

Sudo apt-get install lamp-server^

Si ejecutamos

Php -v 🡪 veremos la versión instalada de php

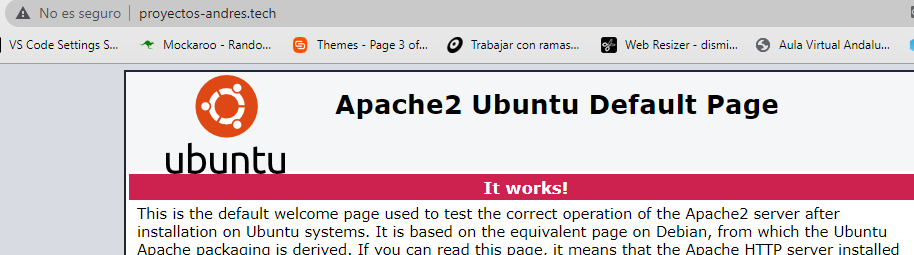
Mysql --version 🡪 veremos la versión de mysql instalada

Apache2ctl -v 🡪 veremos la versión de apache

Reiniciamos Apache2

Systemctl restart apache2

Comprobamos si funciona Apache metiéndonos en nuestra dirección IP /DNS, en nuestro navegador web.



Funciona.

Para agregar php en apache tenemos que hacer lo siguiente

Sudo a2enmod php7.4

Tenemos que añadir el modulo de php en este caso como hemos instalado la versión 7.4 por ello ponemos php7.4.

Luego dentro de ‘var/www/html/’ nos creamos un archivo php y escrimibimos la función de php

Phpinfo();

Y veremos que funciona.

## **PHP 8.0 en apache y PHP 7.4 en la línea de comandos por defecto.**

Instalamos php 8.0

Instalamos las utilidades para poder instalar un repositorio con firma :

Sudo apt install ca-certificates apt-transport-https software-properties-common

Agregamos el repositorio ondrej/php :

Sudo add-apt-repository ppa:ondrej/php

Instalamos php 8.0

Sudo apt-get install php8.0 libapache2-mod-php8.0 php8.0-mysql

Añadimos a apache el modulop php8.0

Sudo adismod php7.4

Sudo a2enmod php8.0

Reiniciamos Apache2

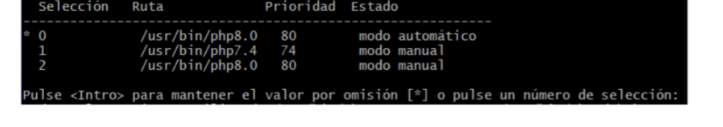
Systemctl restart apache2

Comprobamos como esta el servidor de apache

Sudo systemctl status apache2

Ahora toca configurar php7.4 en línea de comandos

Sudo update-alternatives –config php



Seleccionamos en este caso la 1 ya que es la versión 7.4 para línea de comandos.

Si hacemos un php -v podemos ver que en la línea de comandos esta instalada la versión 7.4

## **INSTALAR PHPMYADMIN**

Instalamos phpmyadmin mas sus dependencias y otros paquetes necesarios

sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl -y

Seleccionamos apache2, le damos contraseña a phpmyadmin y estaría listo.

Ahora entrar en sql y añadir usuarios en este caso vamos añadirle una contraseña a Jose Carlos.



Ejecutamos el siguiente código :

Sudo mysql

Ya estaríamos dentro de mysql

Ahora creamos el usuario.



Usuario 🡪 jdaw

Contraseña 🡪 dawAlzago@1

Le damos privilegios para todo



Ahora el usuario jdaw puede entrar en phpmyadmin y hacer lo que quiera ya que tiene todos los permisos.

[phpmyadmin](http://proyectos-andres.tech/phpmyadmin/)

## **Clave pública de nuestro usuario jdaw, para que se conecte a nuestro servidor.**

Tras habernos dado la clave públic nuestro usuario jdaw, toca añadirla al servidor a su usario, para ello usamos el comando scp que es un comando ssh para subir o bajar archivos, lo tenemos que hacer con nuestro usuario y subirla.

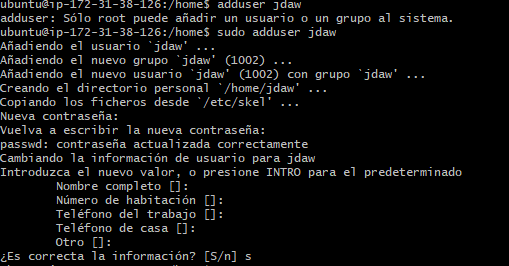


En nuestro servidor lo guardamos en el directorio temporal.

Nos conectamos mediante ssh y vamos hacia ese directorio



Vemos que ahí se encuentra ahora creamos un usario



Nombre del usuario en el servidor 🡪 jdaw

Contraseña del usuario en el servidor 🡪 daw

Ahora lo añadimos al grupo sudo para que tenga permisos de administrador, ya que Jose Carlos es un administrador bueno.



Ahora en el directorio de home de jdaw, creamos un directorio llamado .ssh y dentro de el guardamos la clave publica y le cambiamos el nombre a authorized\_keys





Usuario y contraseña en apache para que pueda entrar en la parte privada de nuestra aplicación web.

Nos tenemos que al directorio donde queramos que solo accedan por usuario y contraseña en mi caso sería dentro del directorio pr101/parte\_privada/ que se encuentra en var/www/html/ para ello ejecutamos lo siguiente :

Sudo htpasswd -c /var/www/html/pr101/parte\_privada/.htpasswd jdaw

Contraseña 🡪 daw

Si queremos añadir otro usuario seria el mismo comando sin ‘-c ’ ya que nos borraría el archivo y jdaw desaparece.

Sudo htpasswd /var/www/html/pr101/parte\_privada/.htpasswd otro usuario

Para que esto surja efecto tenemos entrar en la configuración de apache y crearnos un virtualhost :

sudo nano /etc/apache2/sites-enabled/000-default.conf

Escribimos lo siguiente, <directory aquí va donde queremos que cuando un usuario entre le pida usuario y contraseña>

<VirtualHost \*:80>

<Directory "var/www/html/pr101/parte\_privada">

AuthType Basic

AuthName "Restricted Content"

AuthUserFile /var/www/html/pr101/parte\_privada/.htpasswd

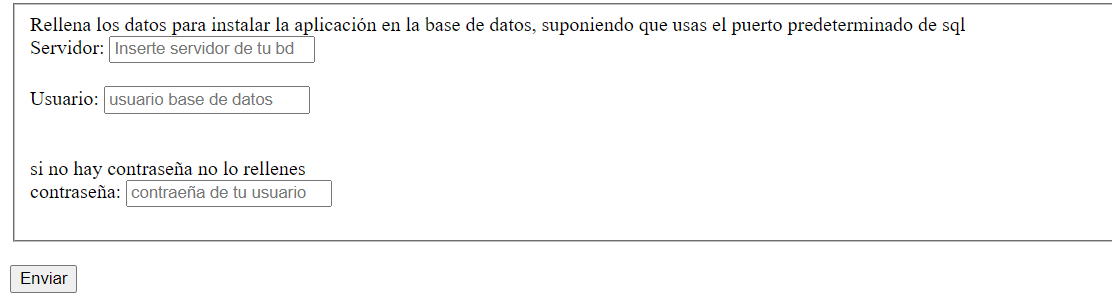
Require valid-user

</Directory>

</VirtualHost>

Reiniciamos apache y listo.

Luego nos vamos al siguiente [enlace](http://proyectos-andres.tech/pr101/conf/instalar.php) y nos llevará para instalar nuestra base de datos.



Servidor 🡪 localhost

Usuario 🡪 root

Contraeña 🡪 no añadir nada

Se instalará y listo.

Previamente se me ha olvidado comentarlo pero habría que poner al usuario root en sql y dejarlo sin contraseña.

# **Referencias**

* Alumno Andrés Cortés de 080 formación
* Documento “IT00179- Configurar AWS EC2 ubuntu 20.04” autor José Carlos Álvarez