# DOCUMENTO TÉCNICO IWANNATRIP

### 

### 

[**DOCUMENTO TÉCNICO IWANNATRIP**](#_boudya1yyuz) **1**

[SISTEMA OPERATIVO](#_i2nge7wfc2wq) 3

[ENLACES ÚTILES](#_ajvb1hrimlqg) 3

[PROYECTOS IWANNATRIP](#_pol6vjonn5e) 3

[INSTALACIÓN STACK LAMP](#_pglakqm9lry) 3

[3.1 APACHE 2](#_ajjefkvpcmgl) 3

[3.2 MYSQL](#_6mk3a1pwhesn) 4

[3.3 PHP Y EXTENSIONES](#_nwdi65tdadag) 5

[3.4 PHPMYADMIN](#_fbyo8p60s2aa) 5

[3.4.1 Habilitando phpMyAdmin desde Apache](#_i5lvfi3wi9p9) 5

[3.4.2 Cambio de variables para mysql](#_q3zeq484r5a1) 6

[INSTALACIÓN SISTEMA DE BOOKING](#_rlq33h2soh82) 7

[PHPJABBERS-SCRIPTS](#_4x5c1rl4mvic) 7

[VIRTUAL HOST](#_ejzdtnd767es) 7

[SSL](#_tk7k4g63fkdt) 11

[CONFIGURAR EL SCHEDULING DE LARAVEL Y SUPERVISOR](#_7ocvx9magiqu) 12

[CONFIGURACIÓN DEL CORREO DE GMAIL](#_d3yvnn8zkcwc) 15

[INSTALACIÓN DE S3 EN EL PROYECTO PRINCIPAL](#_gbx7zw60r3th) 15

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

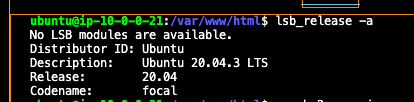
### 

### 

### 

### **SISTEMA OPERATIVO**

En la consola ejecutar el siguiente comando: **lsb\_release -a**

****

### **ENLACES ÚTILES**

Instalar LAMP - Opción 1

<https://www.tutsmake.com/install-lamp-linux-apache-mysql-php-on-ubuntu-20-04-aws-ec2/>

Instalar LAMP - Opción 2

<https://usefulangle.com/post/324/aws-ec2-install-linux-apache-mysql-php-phpmyadmin-lamp-stack-ubuntu-20-04>

Errores de Conexión SSH-EC2

<https://aws.amazon.com/es/premiumsupport/knowledge-center/ec2-linux-resolve-ssh-connection-errors/>

Disabled UFW en Ubuntu (si ya se activó y no se tiene acceso por SSH)

<https://dingyuliang.me/aws-disable-ufw-ec2-linux-instance/>

<https://robinding.medium.com/aws-disable-ufw-in-ec2-linux-instance-979c678475ea>

### **PROYECTOS IWANNATRIP**

* Iwannatrip: principal, directorio: **/var/www/html/Iwannatrip/public\_html**, versión de Laravel 5.1
* BookiW: script de php (se debe de instalar cuando ya este configurado el subdominio)
* DeliveryIWT: procesos internos, directorio: **/var/www/html/DeliveryIWT**, version de Laravel 5.8

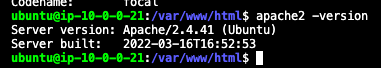
### **INSTALACIÓN STACK LAMP**

#### 3.1 **APACHE 2**

sudo apt update

sudo apt install apache2

apache2 -version



sudo service apache2 restart

#### 3.2 **MYSQL**

sudo apt-get install mysql-server

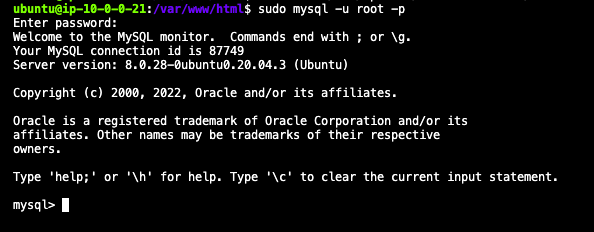
sudo mysql\_secure\_installation

Se le presentará una pantalla donde MySQL le pregunta si:

* Activate the VALIDATE PASSWORD PLUGIN? **Seleccionamos NO**
* Set the password for the root MySql account. Se ingresa el password de root personalizado, en este caso se uso **hWJ2CB6RVF@**
* Remove anonymous users? **Seleccionamos YES**
* Disallow root login remotely? **Seleccionamos NO**
* Remove test database and access to it? **Seleccionamos YES**
* Reload privilege tables now? **Seleccionamos YES**

Después de establecer la contraseña, puede comprobar si MySQL funciona correctamente iniciando sesión en la base de datos con el comando:

**sudo mysql -u root -p,** luegose ingresa el password de root, debe de salir la siguiente pantalla:



Como buena practica no se debe utilizar el usuario root para usarlo como credenciales en los scripts de PHP. Para esto crearemos un nuevo usuario y se le dan privilegios:

CREATE USER 'iw-user'@'localhost' IDENTIFIED BY 'G$RJDZ4PuLD3xLBH@iw';

GRANT ALL PRIVILEGES ON \*.\* TO 'iw-user'@'%' IDENTIFIED BY 'G$RJ79Z4PuLD3xLBH@iw';

FLUSH PRIVILEGES;

Salir de mysql (exit en la consola) y reiniciarlo **sudo systemctl restart mysql**.

#### 3.3 **PHP Y EXTENSIONES**

sudo apt-get install php libapache2-mod-php php-mysql php-curl php-gd php-json php-zip php-mbstring php-intl php-xml php-soap php-opcache php-cli php-fpm php-pdo php-pear php-bcmath

Se tiene que editar el archivo dir.conf del apache

**sudo nano /etc/apache2/mods-enabled/dir.conf**

Y dejarlo de la siguiente manera (index.php de primero)

**<IfModule mod\_dir.c>**

**DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm**

**</IfModule>**

Luego se tiene que reiniciar el apache: **sudo service apache2 restart**

#### 3.4 **PHPMYADMIN**

sudo apt-get install phpmyadmin

##### **3.4.1 Habilitando phpMyAdmin desde Apache**

Luego de terminar la instalación, phpMyAdmin no funcionará y debe incluir en el archivo de configuración de Apache '/etc/apache2/apache2.conf', lo editamos usando la sintaxis:

**sudo nano /etc/apache2/apache2.conf**

Y agregamos al final del archivo la siguiente línea de texto

**Include /etc/phpmyadmin/apache.conf**

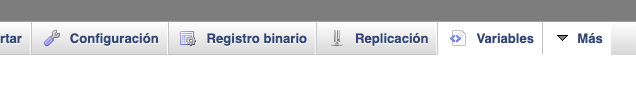
****

Y se reinicia el apache: **sudo service apache2 restart**. Una vez intslado y configurado podemos acceder con la siguiente URL:

<http://ec2-3-81-214-161.compute-1.amazonaws.com/phpmyadmin/index.php>

##### **3.4.2 Cambio de variables para mysql**

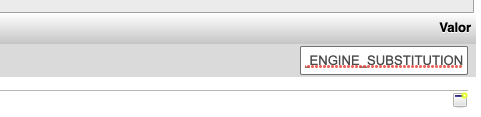
Dentro del phpmyadmin se tiene que cambiar unas variables para el correcto funcionamiento del proyecto. Sin seleccionar la BD, nos dirigimos a la pestaña Variables



Dentro de Variables, ubicamos la propiedad **sql\_mode y** le damos a editar

****

En el campo valor

****

Sustituir esto:

ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION

Por esto:

ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION

Por ultimo se tiene que setear en mysql el nivel del password de MYSQL en MEDIUM.

**sudo mysql -u root -p**

**Enter password: \*\*\*\*\*\*\*\*\*\***

**SHOW VARIABLES LIKE 'validate\_password%';**

Si esta el LOW, ejecutar el siguiente comando para cambiarlo a medium:

**SET GLOBAL validate\_password.policy=MEDIUM;**

### **INSTALACIÓN SISTEMA DE BOOKING**

**IMPORTANTE:** Instalarlo una vez que ya este configurado el virtual host y el ssl. Se debe de instalar con la URL final que se va a utlizar que debe de ser https. Esto evitara problemas de CORS al momento de usar el booking en el proyecto de iwannatrip.

#### PHPJABBERS-SCRIPTS

Para descargarse desde 0 el script del booking

* **URL:** https://www.phpjabbers.com/accounts/login
* **Email:** info@iwannatrip.com
* **Password:** AY6LN9QM
* **Licence:** 71014917085d7cccb7e31f4f64ec860e

Para ingresar al proceso de instalacion se usa la siguiente URL:

<https://bookiw.iwannatrip.com/>

Se le tienen que dar permisos a algunas carpetas en el directorio del proyecto:

**sudo chmod -R 777 app/config/config.inc.php app/web/upload app/web/backup app/web/invoices**

Seguir los pasos de instalación. Una vez terminado se tiene que actualizar el codigo con el codigo del repositorio.

### **VIRTUAL HOST**

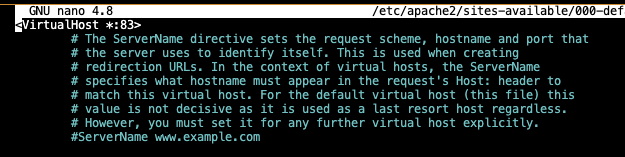
Se tienen 3 proyectos: BookiW, DeliveryIWT y Iwannatrip. Se tiene que crear un virtual host para cada proyecto.



Para esto se tiene que modificar un archivo y agregar 3 nuevos en el apache.

* Modificar archivo: **sudo nano /etc/apache2/sites-available/000-default.conf**

En este archivo se cambia el puerto 80 por el 83. Este puedo no tiene reglas en el grupo de seguridad. Solo se le crean reglas de seguridad a los puertos 80, 81 y 82.



* Crear archivo: **sudo nano /etc/apache2/sites-available/deliveryiwt.conf**

Se crea el archivo con la sigueinte configuración: para este proyecto se accede con la URL del EC2 [http://ec2-3-81-214-161.compute-1.amazonaws.com:82](http://ec2-3-81-214-161.compute-1.amazonaws.com:82/email), este proyecto no es necesario crear un subdominio en Route53 porque no se expone a los usuarios, es un proyecto para procesos internos.

<VirtualHost \*:82>

ServerName deliveryiw.iwannatrip.com

ServerAlias www.deliveryiw.iwannatrip.com

DocumentRoot /var/www/html/DeliveryIWT/public

<Directory /var/www/html/DeliveryIWT/public>

Options +FollowSymlinks

AllowOverride All

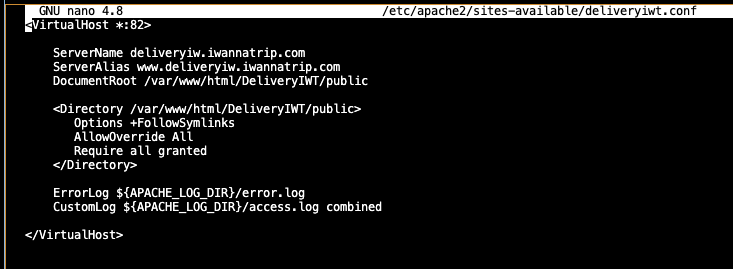
Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

****

* Crear archivo: **sudo nano /etc/apache2/sites-available/iwanatrip.conf**

Se crea el archivo con la sigueinte configuración: (el ServerName y el ServerAlias deben registrase como dominios en Route53). Este es la configuración del proyecto principal de Iwannatrip.

<VirtualHost \*:80>

ServerName iwannatrip.com

ServerAlias www.iwannatrip.com

DocumentRoot /var/www/html/Iwannatrip/public\_html/public

<Directory /var/www/html/Iwannatrip/public\_html/public>

Options +FollowSymlinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

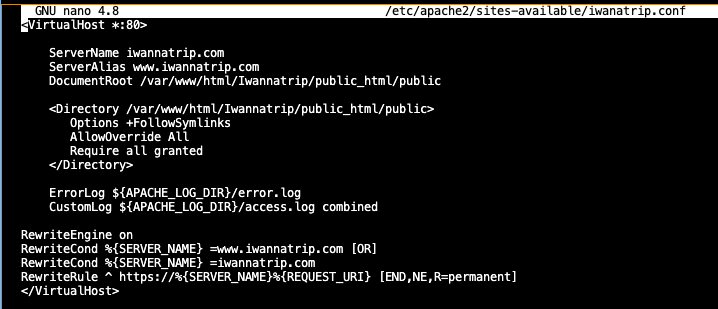
RewriteEngine on

RewriteCond %{SERVER\_NAME} =www.iwannatrip.com [OR]

RewriteCond %{SERVER\_NAME} =iwannatrip.com

RewriteRule ^ https://%{SERVER\_NAME}%{REQUEST\_URI} [END,NE,R=permanent]

</VirtualHost>

****

* Crear archivo: **sudo nano /etc/apache2/sites-available/bookiwt.conf**

Se crea el archivo con la sigueinte configuración: (el ServerName y el ServerAlias deben registrase como subdominios en Route53)

<VirtualHost \*:81>

ServerName bookiw.iwannatrip.com

ServerAlias www.bookiw.iwannatrip.com

DocumentRoot /var/www/html/BookiW

<Directory /var/www/html/BookiW>

Options +FollowSymlinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

RewriteEngine on

RewriteCond %{SERVER\_NAME} =bookiw.iwannatrip.com [OR]

RewriteCond %{SERVER\_NAME} =www.bookiw.iwannatrip.com

RewriteRule ^ https://%{SERVER\_NAME}%{REQUEST\_URI} [END,NE,R=permanent]

</VirtualHost>



Una vez creados los VH se tienen que habilitar en el apache. Para esto se tienen que ejecutar los siguientes comandos:

**sudo a2enmod rewrite**

**sudo a2ensite deliveryiwt.conf**

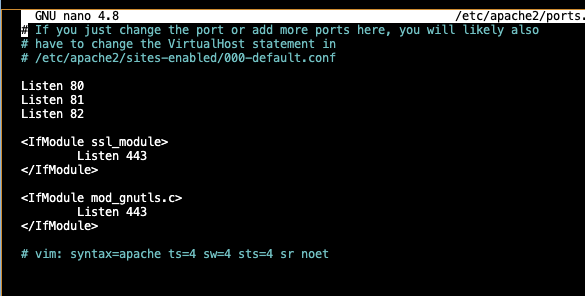
**sudo a2ensite iwanatrip.conf**

**sudo a2ensite bookiwt.conf**

**sudo systemctl reload apache2**

Se debe editar el archivo de puertos del apache para que el apache escuche en los puertos que se acaban de configurar en los VH. En este caso solo usamos el puerto 80,81 y 82

Para editar el archivo usamos: **sudo nano /etc/apache2/ports.conf**

****

Luego de agregar los puertos se debe reiniciar el apache.

**sudo systemctl reload apache2**

### **SSL**

Se tiene que habilitar el SSL en el apache, para esto ejecutamos los siguientes comandos:

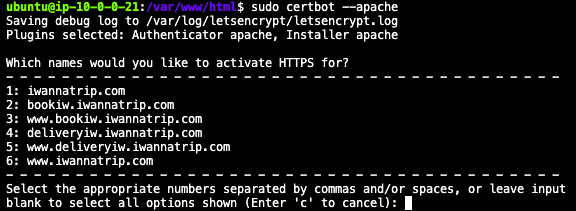
**sudo a2enmod ssl**

**sudo systemctl restart apache2**

Se tiene que instalar lo siguiente (<https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-20-04-es> para este link no se habilita en ufw en ubuntu, esa función la hace el grupo de seguridad de la instancia):

**sudo apt install certbot python3-certbot-apache**

Para crear los virtual host para que apunten al puerto 443 (SSL/TLS), se tiene que crear una regla en el grupo de seguridad para el puerto 443, luego se ejecuta el siguiente comando en la consola: **sudo certbot --apache**  este comando lista todos los names que estan disponibles pata habilitar el HTTPS



se seleccionan los que se requieren crear el certificado (si son varias se separan por comas 1,3,5). Si todo sale bien, se obtendrá un mensaje de la siguiente manera:

**- Congratulations! Your certificate and chain have been saved at:**

**/etc/letsencrypt/live/iwannatrip.com/fullchain.pem**

**Your key file has been saved at:**

**/etc/letsencrypt/live/iwannatrip.com/privkey.pem**

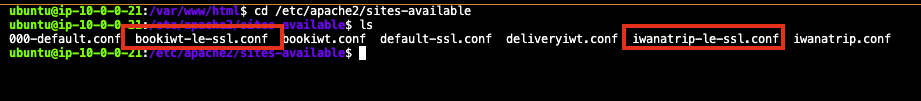
**Your cert will expire on 2022-07-04. To obtain a new or tweaked**

**version of this certificate in the future, simply run certbot again**

**with the "certonly" option. To non-interactively renew \*all\* of**

**your certificates, run "certbot renew"**

Esto creara un archivo(s) en la configuración del apache:

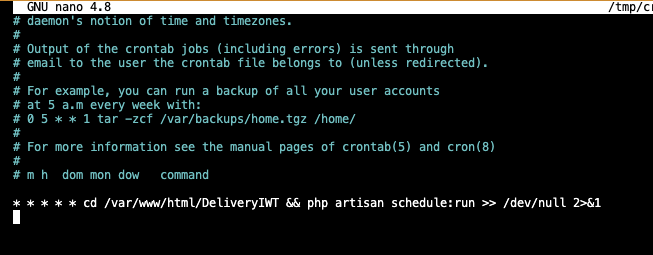


### **CONFIGURAR EL SCHEDULING DE LARAVEL Y SUPERVISOR**

Para que funcione el scheduler de laravel se tiene que agregar una línea en el crontab, para esto ir a la consola y ejecutar el comando **crontab -e**, si es la primera que se abre este archivo se pide que seleccione el editor de su preferencia (en este caso se seleccionó nano).

Una vez que se abra el archivo se agrega la siguiente linea al final del archivo.

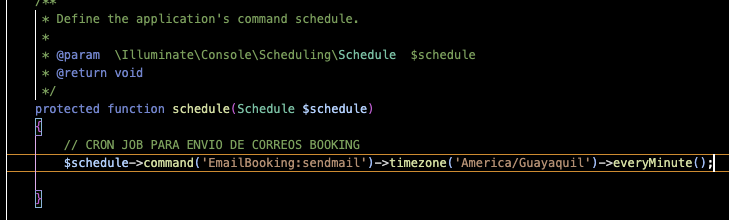
**\* \* \* \* \* cd /var/www/html/DeliveryIWT && php artisan schedule:run >> /dev/null 2>&1**



Para guadar el archivo **CTRL + O**, y para salir del edior **CTRL + X**.

Esto hace que se ejecuten los comandos creados dentro de

**/var/www/html/DeliveryIWT/app/Console/Kernel.php**

****

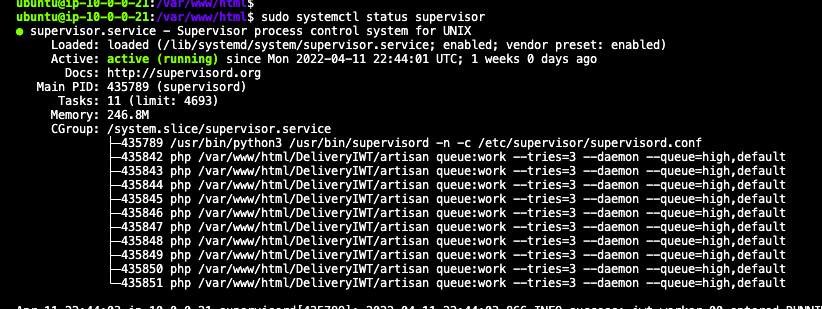
Este cron inserta registros en la tabla jobs (colas en laravel). Para procesar las colas se necesita instalar supervisor. <https://www.atlantic.net/vps-hosting/how-to-install-and-configure-supervisor-on-ubuntu-20-04/>

sudo apt-get install supervisor -y

supervisord -v



Para consultar el status de supervisor se usa el comando **sudo systemctl status supervisor**. Tiene que estan en **active**.

****

Se tiene que crear el archivo de configuración de supervisor.

Se cres primero la carpeta: **sudo mkdir /etc/supervisor/conf.d/**

Se crea el archivo: **sudo nano /etc/supervisor/conf.d/iwt-worker.conf**

Dentro del archivo de configuración, colocar lo siguiente:

[program:iwt-worker]

process\_name=%(program\_name)s\_%(process\_num)02d

command=php /var/www/html/DeliveryIWT/artisan queue:work --tries=3 --daemon --queue=high,default

autostart=true

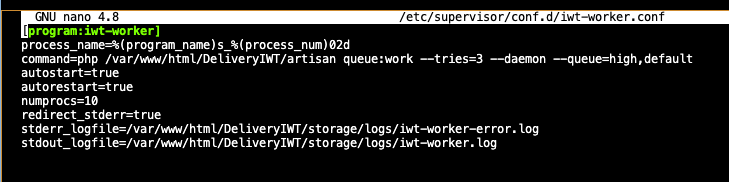
autorestart=true

numprocs=10

redirect\_stderr=true

stderr\_logfile=/var/www/html/DeliveryIWT/storage/logs/iwt-worker-error.log

stdout\_logfile=/var/www/html/DeliveryIWT/storage/logs/iwt-worker.log



Para guardar el archivo **CTRL + O**, y para salir del editor **CTRL + X**.

Se le tiene que dar permisos de ejecución al archivo:

**sudo chmod +x /etc/supervisor/conf.d/iwt-worker.conf**

Luego ejecutar los siguientes comando en este orden:

**sudo supervisorctl reread**

Se obtendrá una respuesta como esta:

>> iwt-worker: changed

**sudo supervisorctl update**

Se obtendrá una respuesta como esta:

>> iwt-worker: stopped

>> iwt-worker: updated process group

**sudo supervisorctl start iwt-worker:\***

Es recomendable cada vez que se haga un cambio en el coigo de **DeliveryIWT** reiniciar las colas de laravel, limpiar la caché de laravel y reiniciar el supervisor y el apache

* Reiniciar colas y borrar cache

**cd /var/www/html/DeliveryIWT**

**php artisan cache:clear && php artisan route:clear && php artisan config:clear && php artisan view:clear**

**php artisan queue:restart**

* Reiniciar supervisor y el apache

**sudo service supervisor restart && sudo systemctl restart apache2**

### **CONFIGURACIÓN DEL CORREO DE GMAIL**

Para los proyectos de laravel (Iwannatrip y DeliveryIWT), esta es la configuración del correo que actualmente (archivo .env del root de los proyectos) y está funcionando actualmente (19/04/2022).

**MAIL\_DRIVER=smtp**

**MAIL\_HOST=smtp.gmail.com**

**MAIL\_PORT=587**

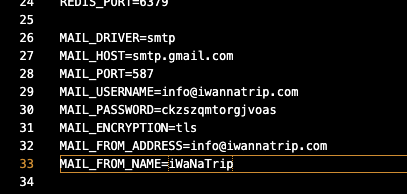
**MAIL\_USERNAME=info@iwannatrip.com**

**MAIL\_PASSWORD=ckzszqmtorgjvoas**

**MAIL\_ENCRYPTION=tls**

**MAIL\_FROM\_ADDRESS=info@iwannatrip.com**

**MAIL\_FROM\_NAME=iWaNaTrip**



En gmail se tiene que habilitar la autenticación de dos pasos y crear una contraseña de app.

Si se realiza un cambio en el archivo .env, se tiene que ejecutar los siguientes comandos para limpiar la caché y que tengan efecto los nuevos cambios:

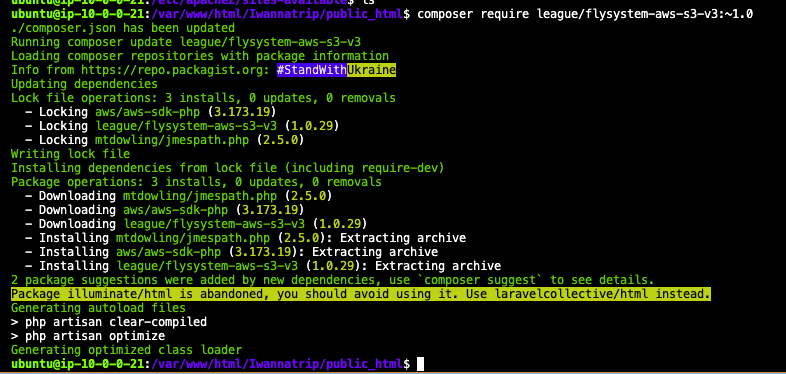
**php artisan config:cache && php artisan config:clear**

### **INSTALACIÓN DE S3 EN EL PROYECTO PRINCIPAL**

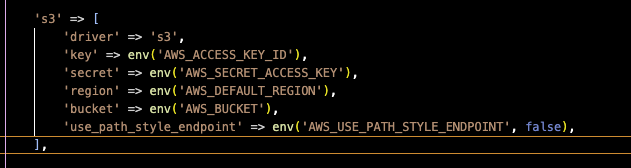
En el root del proyecto principal (**/var/www/html/Iwannatrip/public\_html**) ejecutar el siguiente comando:

**composer require league/flysystem-aws-s3-v3:~1.0**

**composer require --with-all-dependencies league/flysystem-aws-s3-v3 "^1.0"**

****

En el archivo **/var/www/html/Iwannatrip/public\_html/config/filesystems.php** se setean las variables de configuración de AWS.



Además, en el archivo .env se setean las siguientes variables

***AWS\_ACCESS\_KEY\_ID=xxxzzz***

***AWS\_SECRET\_ACCESS\_KEY=xxxyyy***

***AWS\_DEFAULT\_REGION=us-east-2***

***AWS\_BUCKET=my-awesome-bucket***

***AWS\_USE\_PATH\_STYLE\_ENDPOINT=false***