

PlayTUBE



Github [msanchez3](#)

Index

PlayTUBE	1
Introducción	3
Objetivos del sistema	3
Alcance del sistema	3
Comparativa de productos.....	4
Planificación temporal y planificación de costos.	4
Plex.....	5
sshGuard.....	11
Que es sshGuard?.....	11
Instalación:	11
Umbral de la lista negra.....	12
Ejemplo de prohibición moderado.....	12
Prohibición agresiva	13
Instalación de apache.....	19
Copia de seguridad con Wordpress.....	24
¿Que es Docker Swarm?.....	25
Las ventajas que tendría al usar esta tecnología.....	25
Docker Swarm	26
Despliegue de mi proyecto con docker Swarn.	26
Prueba de proyecto con video	28
Conclusión personal.	30
Webgrafica	31

Introducción

Mi proyecto consta de las siguientes características, para empezar, tenemos una página **WordPress**, en la que se explica brevemente el objetivo del proyecto, después tenemos enlazada en la misma página, una barra de navegación con las siguientes secciones, **qBittorrent**, **plex**, **github** y una sección de login para poder logearnos.

En **qBittorrent** nos encontramos con el servicio encargado de descargar las películas a partir del archivo Torrent que descarguemos.

En la sección **Plex** nos dirigimos al reproductor de video que utilizaremos en nuestra máquina.

GitHub redirigirá a mi cuenta de **GitHub** donde está el proyecto y la sección de login para logar el usuario.

También tenemos montado un firewall llamado **sshGuard** que nos permitirá tener control sobre las cuentas que se conectan, etc..

sshGuard dispone de una whitelist pero igualmente nosotros crearemos un backup de emergencia en **wordpress** y por ultimo utilizaremos **Docker Swarm** para hacer un backup extra en caso de que la maquina caiga y que la pagina pueda seguir funcionando.

La mayor parte de esto está montada con **Docker** utilizando **docker-compose**.

Objetivos del sistema

El objetivo que quiero conseguir con este sistema, es crear un sistema de transmisión o “**Streaming**” el cual te permita ver series y otro tipo de videos, los cuales se puedan añadir y sea seguro en caso de parones a la vez que sea multiplataforma.

Alcance del sistema

Crear una alternativa a otros sistemas de reproducción gratuitos.

Comparativa de productos.

¿Qué son Jellyfin y Emby?

Jellyfin y Emby estos son dos servicios en parte gratuitos que suelen ser los más utilizados por la comunidad, para ver películas, series, etc.. son dos alternativas completamente válidas para crear tu propio programa para ver películas. También se pueden utilizar a parte de en local en red y también son multiplataforma. Emby se tiene que pagar para desbloquear ciertas funciones, pero Jellyfin es completamente gratuito.

Planificación temporal y planificación de costos.

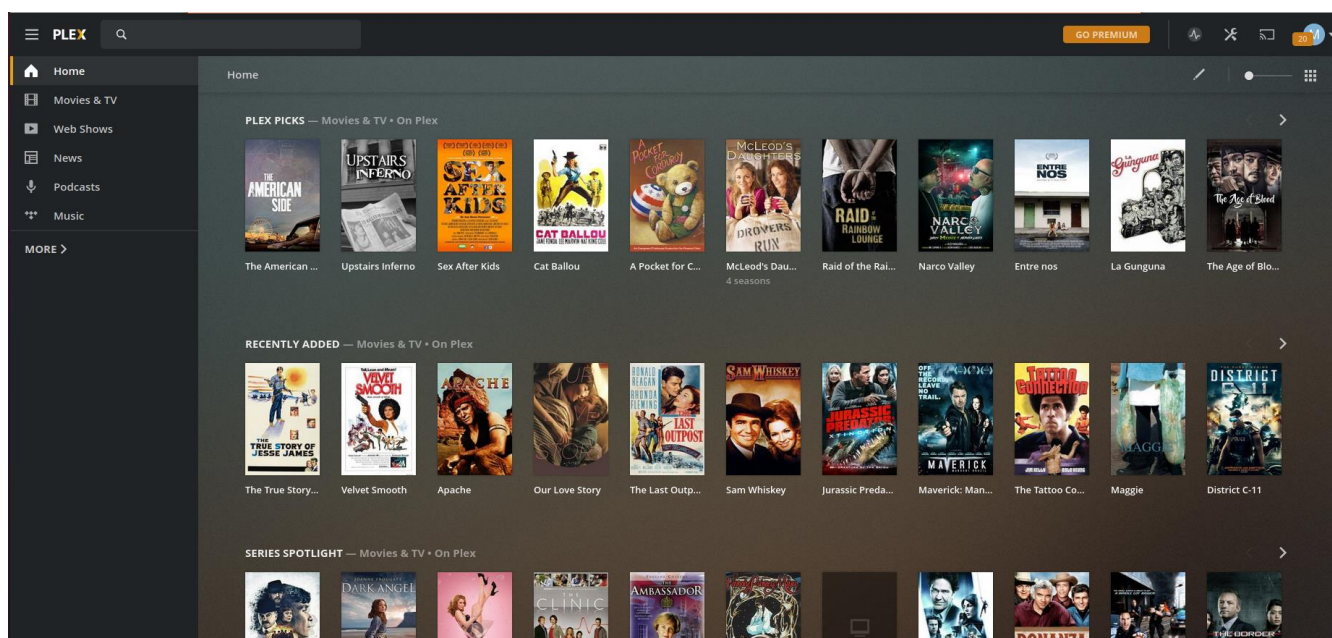
Los datos de la planificación temporal y los costes, se encuentran en el GitHub junto a otros datos que incumben al proyecto. Este es el [GitHub](#).

Plex

Plex es un servidor multimedia el cual guarda nuestras películas, series, música, libros...

Organizados y almacenarlos en un lugar específico, el cual podemos modificar.

Lo mejor de **Plex**, es la facilidad que tiene para entender la configuración que tiene, el moverse por los menús y el diseño que tiene, le permiten al usuario adaptarse rápidamente. También la seguridad que nos otorga el programa, que aun que no tengamos la versión premium sabemos con toda seguridad que nuestros datos estarán a salvo a la vez que nuestras películas y series.



Para lograr obtener este sistema de control de video y multimedia, a través de una interfaz capaz de llevar a cabo todo lo que este propósito conlleva, es necesario primeramente que esté conectada a un servidor, en el cual la aplicación **Plex** queda configurada para que todos estos beneficios como son recopilar videos, reproducir contenido multimedia, transmitir datos, tener una interfaz gráfica o GUI, utilizaremos **Docker-compose**, el cual nos permitirá tener un gran control y poner **Plex** en su mejor funcionamiento.

Instalación.

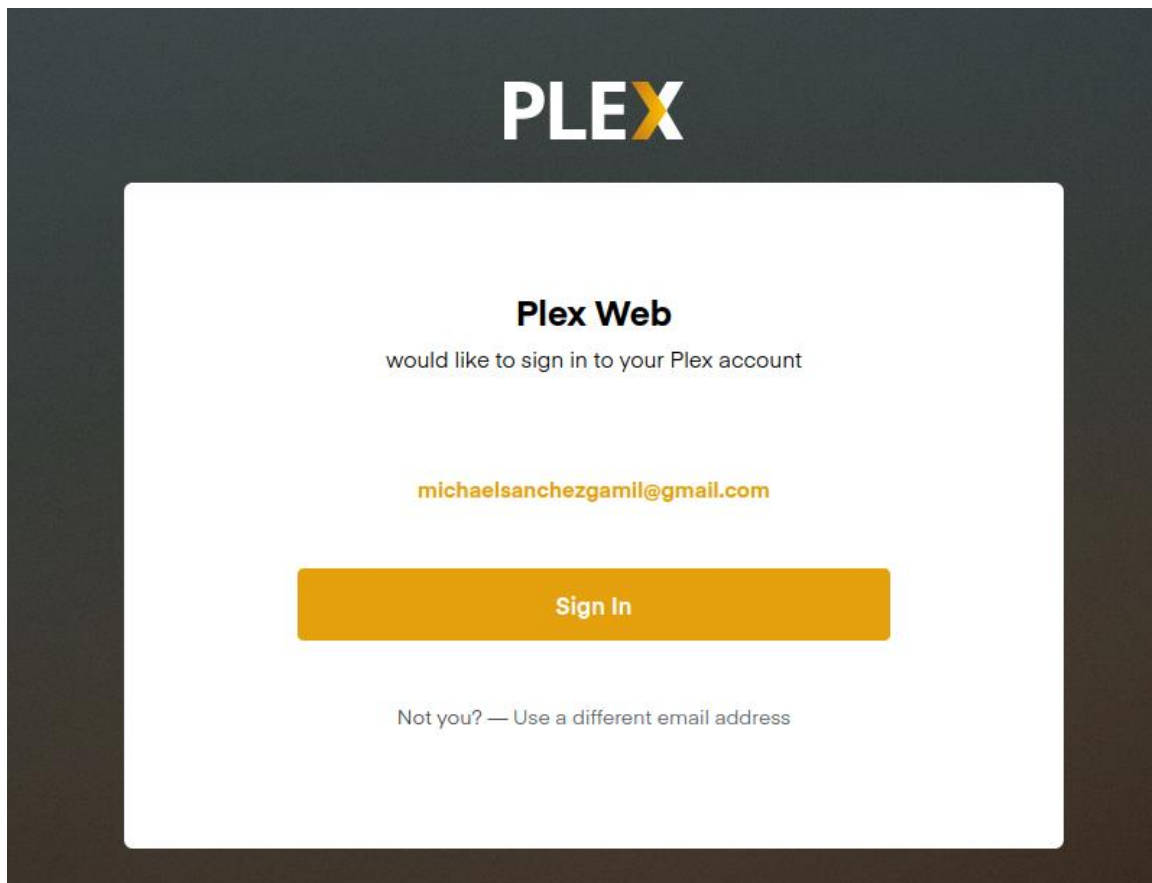
Para llevar a cabo la instalación de Plex he utilizado una imagen ya creada en el repositorio oficial del mismo, la cual dejare en la webgrafía.

Toda la configuración relacionada con el apartado Docker-compose esta explicada en el mismo (documento subido en el GitHub), para que cualquiera que quiera aplicar estos conocimientos sepa de que estamos hablando con facilidad.

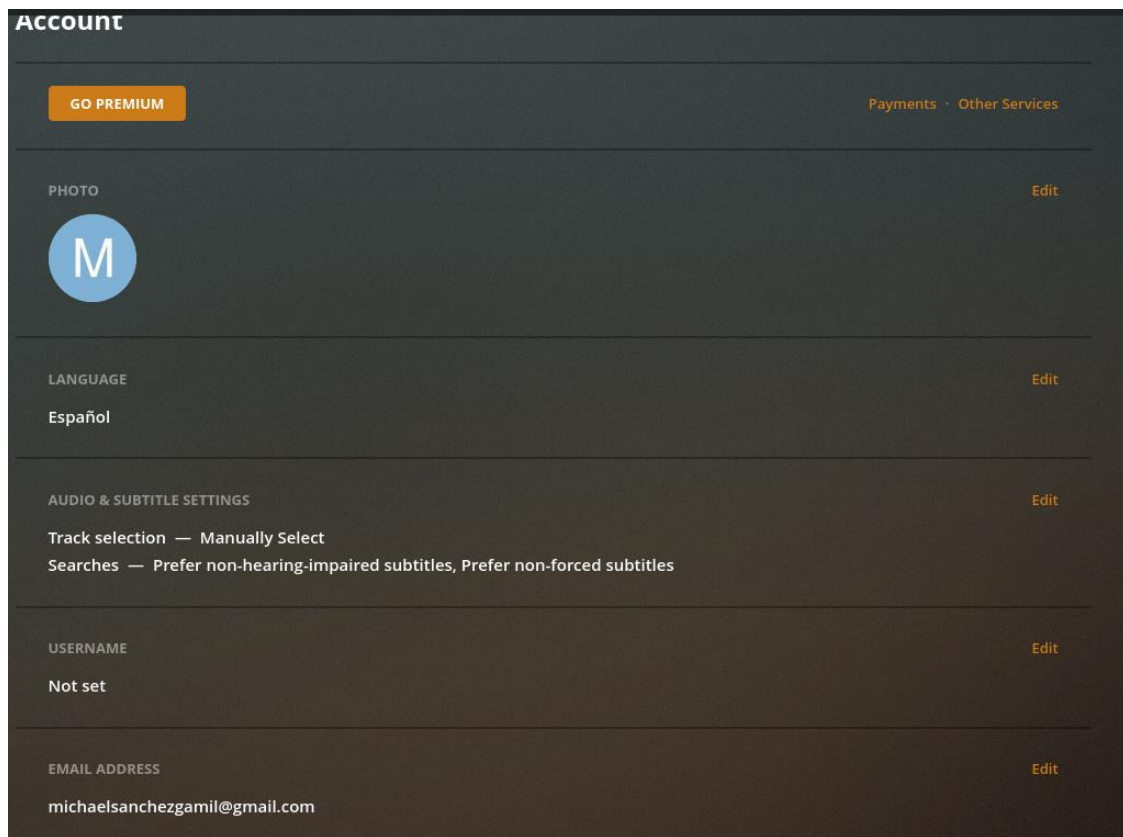
```
services:
  plex:
    image: linuxserver/plex
    container_name: plex
    network_mode: host
    environment:
      - PUID=1000
      - PGID=1000
      - VERSION=docker
      - UMASK_SET=022 #optional
    volumes:
      - /home/shared/plex/library:/config
      - /home/shared/plex/tvseries:/tv
      - /home/shared/plex/movies:/movies
    ports:
      - 32400:32400
    restart: unless-stopped
```

```
root@msanchez:/home/msanchez/Documentos/Docker# docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS        PORTS
fd519ab3807e   linuxserver/plex  "/init"                 3 days ago    Up 6 hours    
```

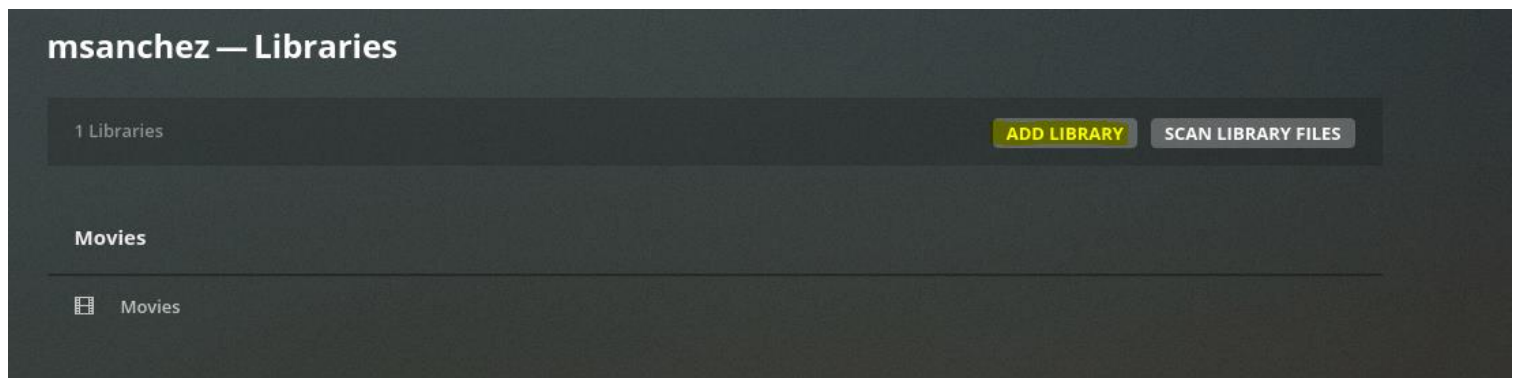
Para iniciar la GUI tenemos que acceder al navegador por la siguiente URL <http://127.0.0.1:32400>, tal y como se explica en la página oficial, aparte de que en la configuración hemos puesto que lea el puerto 32400, y para terminar con la configuración hacemos los siguientes pasos:



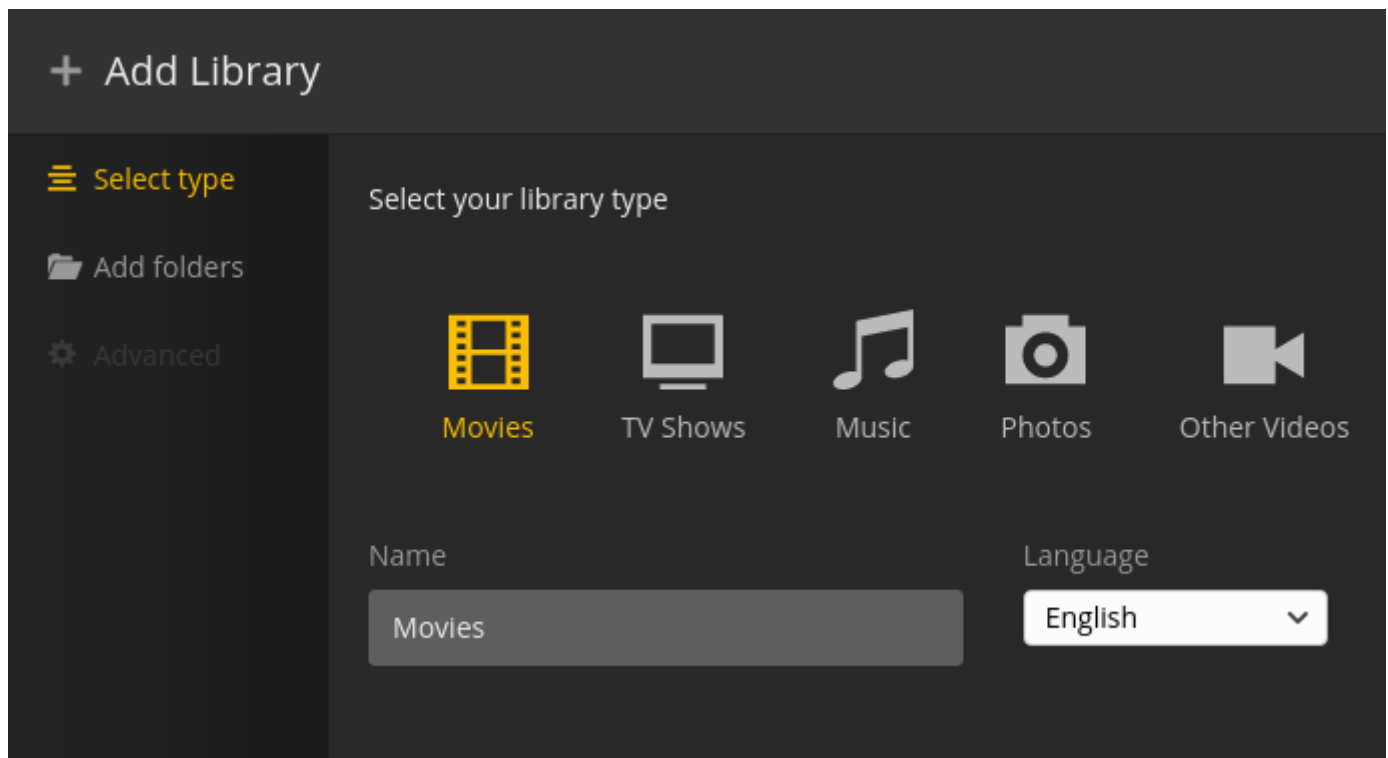
- Se inicia sesión creando una cuenta o con las cuentas de otras aplicaciones (Google, Facebook, etc..)



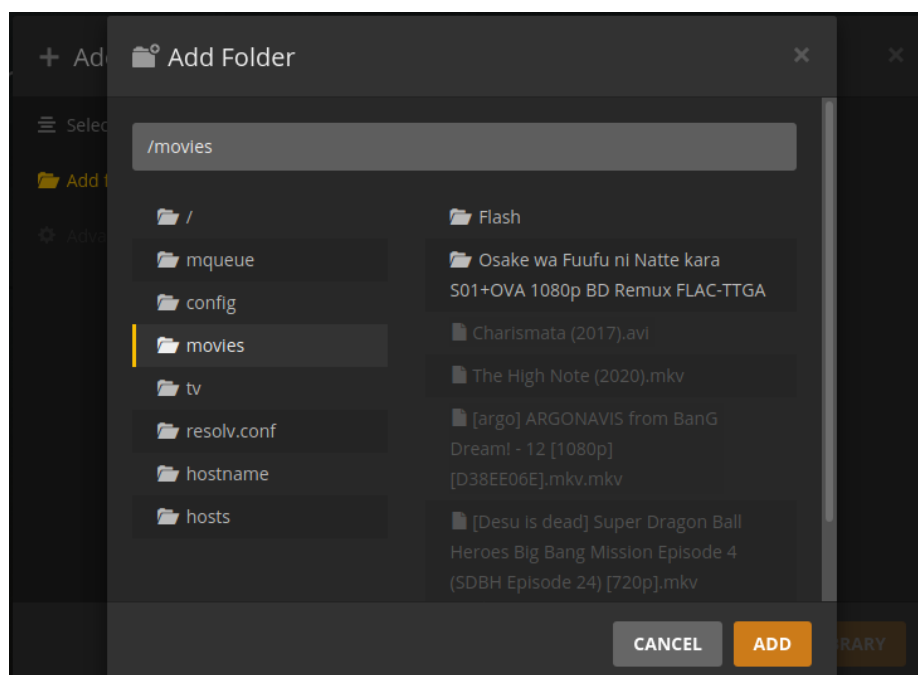
- Una vez iniciada la cuenta, ya se puede utilizar toda la aplicación, pero puedes seguir configurando tu cuenta.



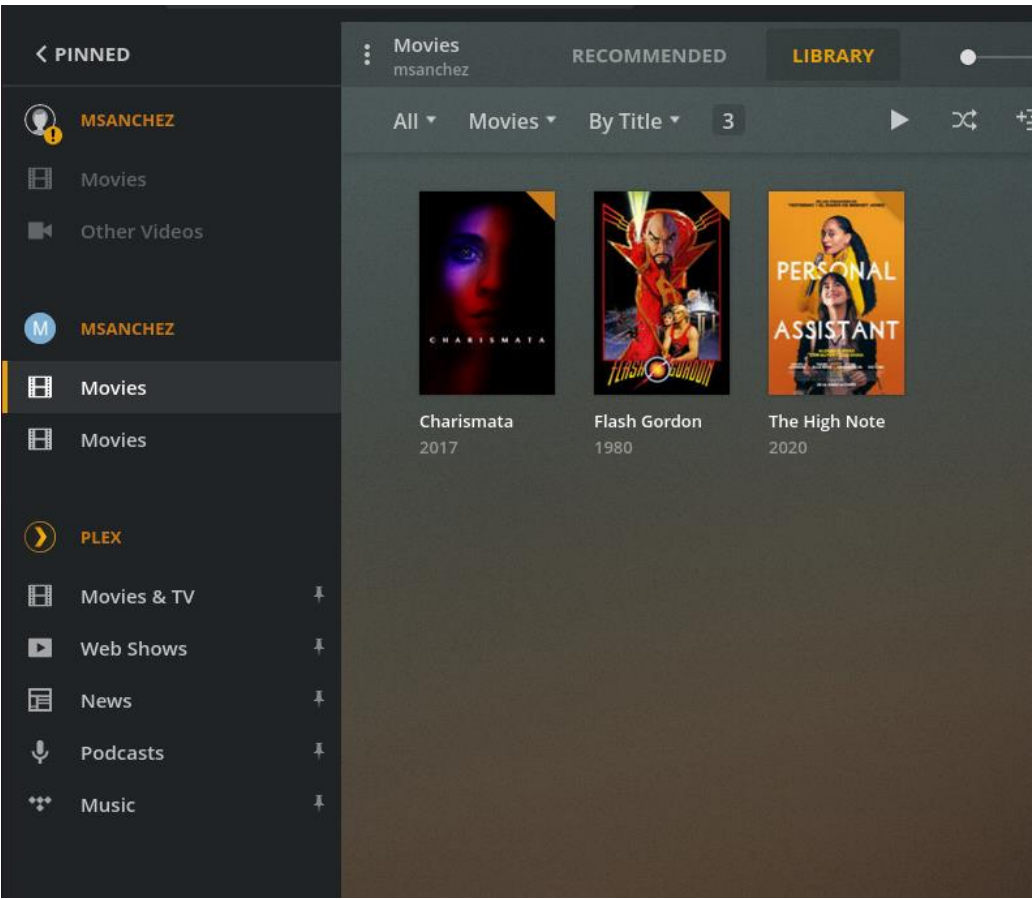
- Para añadir películas, hace falta crear una librería nueva.



- Seleccionamos el tipo de contenido que añadiremos a la librería y el nombre más el idioma.



- Una vez creada la librería seleccionamos la carpeta que contiene las películas, y listo.



sshGuard



Que es sshGuard?

- Es una herramienta que monitorea constantemente los “logs” del servicio “ssh” principalmente y se fija si hay intentos de “logue, ataques de fuerza bruta, etc..” y junto con el acceso que tiene a las “Iptable” bloquea el acceso para protegernos de un posible ataque.

Instalación:

1- asegurarnos de que nuestro sistema está actualizado.

```
apt-get update && apt-get upgrade -y
```

2- instalar.

```
apt-get install sshguard
```

```
root@msanchez:/home/msanchez/Documentos/Docker# sudo apt-get install -y sshguard
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
sshguard ya está en su versión más reciente (2.3.1-1ubuntu1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 114 no actualizados.
root@msanchez:/home/msanchez/Documentos/Docker# apt update
root@msanchez:/home/msanchez/Documentos/Docker# iptables -A INPUT -j sshguard
root@msanchez:/home/msanchez/Documentos/Docker# ip6tables -A INPUT -j sshguard
```

Creación de configuración para que sshGUARD funcione con los demás servicios:

La configuración del servicio **sshGuard** varía mucho dependiendo de los servicios que tengas en tu maquina y con los que quieres que interactúe.

Gracias a **sshGuard** podemos ver los registros de todo lo que a ocurrido en el transcurso del tiempo, entrando a este archivo o utilizando este comando:

`/usr/lib/systemd/scripts/sshguard-journalctl` y el servicio `systemd sshguard.service`. También como se ha comentado anteriormente se puede acceder desde este comando:

```
$ journalctl -afb -p info SYSLOG_FACILITY = 4 SYSLOG_FACILITY = 10
```

Dentro del siguiente directorio esta la configuración necesaria para que **sshGuard** se inicie `/etc/sshguard.conf` .

Dentro de la siguiente ruta, nos podemos encontrar un archivo en el que se pone ejemplo de la configuración, el cual se puede copiar y utilizar sin problemas `/usr/share/doc/sshguard/sshguard.conf.sample`o también se puede encontrar en [Bitbucket sshguard.conf.sample](#) .

Nota: Los comandos de systemd no son compatibles con sshGuard, todo se indica en el archivo anteriormente nombrado.

Umbral de la lista negra

Predeterminadamente un ataque de gran peligrosidad empieza en el intento de iniciar sesión en la 12 vez. Una vez sabemos eso bloqueamos permanentemente a un usuario que supere esa cifra, también podemos modificar este parámetro en la siguiente línea de los archivos de configuración para que nosotros decidamos cuantos “puntos” de peligro tiene que haber para que se perma-banne a alguien de la página [12 intentos = 120 puntos].

```
BLACKLIST_FILE = 200: /var/db/sshguard/blacklist.db
```

En `200`: este ejemplo, le dice a sshguard que prohíba permanentemente un host después de alcanzar un nivel de peligro de 200.

Finalmente [reiniciar](#) `sshguard.service`

Ejemplo de prohibición moderado

Aquí creo un ejemplo de lo que es una prohibición algo más restringida que la predeterminada:

- Monitorea [sshd](#) y [vsftpd](#) a través de registros del [systemd / Journal](#)
- Bloquea a los atacantes después de 2 intentos (cada uno con un costo de 10, explicando el `20`valor del `THRESHOLD`parámetro) durante 180 segundos con un tiempo de bloqueo posterior más largo por un factor de 1.5. Tenga en cuenta que este retraso multiplicativo 1.5 es interno y no está controlado en la configuración
- Los atacantes están en la lista negra permanente después de 10 intentos (10 intentos que tienen un costo de 10 cada uno, explicando el `100`valor en el `BLACKLIST_FILE`parámetro)
- Bloquea no solo la IP del atacante sino toda la subred IPv4 24 (notación [CIDR](#))

```
/etc/sshguard.conf
```

```
#
Ruta completa al backend ejecutable (requerido, no predeterminado) BACKEND = "/usr/lib/sshguard/sshg-fw-iptables"

# Comando del lector de registro (opcional, no predeterminado)
LOGREADER = "LANG = C /usr/bin/journalctl -afb -p info -nl -t sshd -t vsftpd -o cat "

# Cuántos intentos problemáticos desencadenan un bloque
THRESHOLD = 20
# Los bloques duran al menos 180 segundos
BLOCK_TIME = 180
# Los atacantes son recordados por hasta 3600 segundos
DETECTION_TIME = 3600

# Umbral de lista negra y nombre de archivo
BLACKLIST_FILE = 100: /var/db/sshguard/blacklist.db

# Tamaño de subred IPv6 para bloquear. El valor predeterminado es una sola dirección, notación CIDR. (opcional, predeterminado a 128)
IPv6_SUBNET = 64
# Tamaño de subred IPv4 para bloquear. El valor predeterminado es una sola dirección, notación CIDR. (opcional, predeterminado a 32)
IPv4_SUBNET = 24
```

Prohibición agresiva

En algunos sistemas los ataques son casi continuos, y la probabilidad de fallo de sesión es casi nula, así que siempre tenemos la opción de poner que cualquier error sea fatal, ya que en cualquier momento el administrador puede quitar el bloqueo, en algunos sistemas es mejor poner la prohibición agresiva.

```
THRESHOLD = 10
BLACKLIST_FILE = 10: /var/db/sshguard/blacklist.db
```

Finalmente [reiniciar](#) `sshguard.service`.

Además, siempre se pueden restringir los inicios de sesión a través de una sola conexión para evitar problemas, es posible que desee cambiar `/etc/ssh/sshd_config` definiendo:

```
MaxAuthTries 1
```

[Reinicie](#) `sshd.service` para que este cambio surta efecto.

Anotación:

El servicio sshGuard está instalado en el sistema para que proteja el sistema sin tener que iniciar ningún container. Eso lo hace más seguro ya que al estar en el sistema, se suma a los contenedores automáticamente.

qBittorrent



qBittorrent es uno de los más apreciados clientes de Bittorrent, ya que ofrece una interfaz sencilla y amigable para que todos los usuarios tengan facilidad al utilizarlo, está escrito en C++ y tiene licencia GPL.

Es mundialmente conocido y utilizado por la comunidad, qBittorrent ofrece cifrado, enlaces magnéticos, reenvío de puertos UPnP y NAT-PMP, aparte de otras muchas.

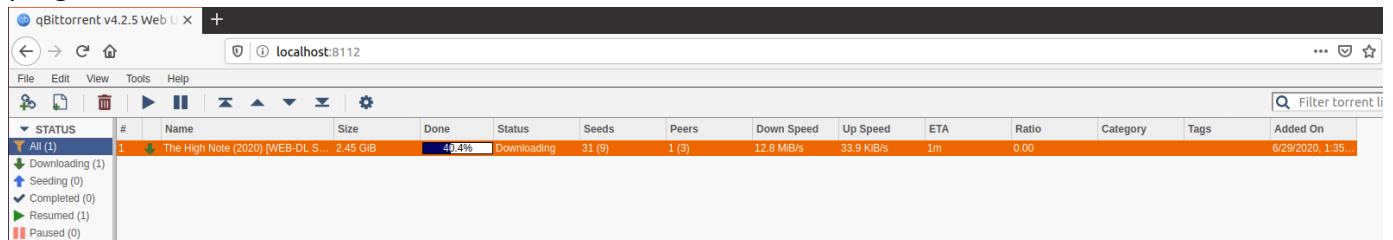
A continuación, explicare la instalación:

Instalación:

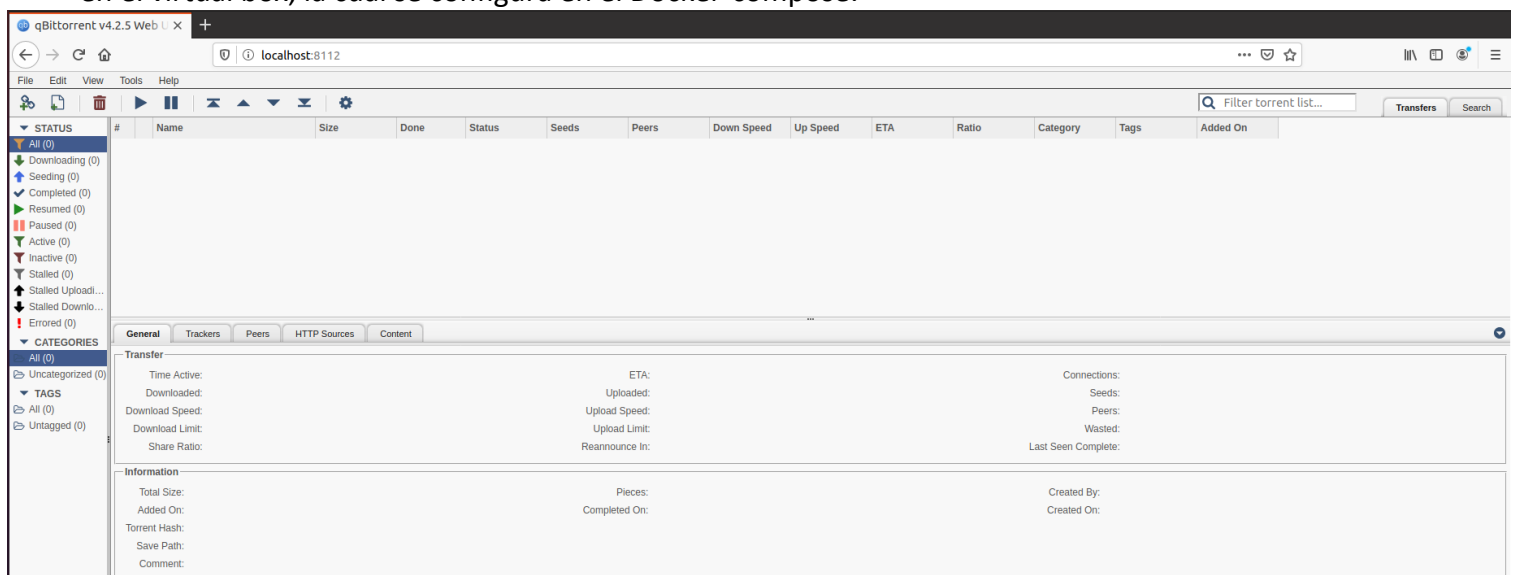
La instalación de qBittorrent es casi exactamente igual que la de **plex** si nos referimos al documento docker-compose, estas son las líneas de configuración del servicio qBittorrent.

```
qbittorrent:
  image: linuxserver/qbittorrent
  container_name: qbittorrent
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=Europe/London
    - UMASK_SET=022
    - WEBUI_PORT=8909
  volumes:
    - /home/share/plex/library:/config
    - /home/share/plex/tvseries:/tv
    - /home/share/plex/movies:/movies
  ports:
    - 6881:6881
    - 6881:6881/udp
    - 8909:8909
  restart: unless-stopped
```

Al instalar qBittorrent viene con una contraseña predeterminada que se cambia en los ajustes dentro del programa.



para poder acceder a la GUI del servicio, debemos tener la redirección de puertos **8112:8112** en el virtual box, la cual se configura en el Docker-compose.



Lo más relevante son las siguientes líneas:

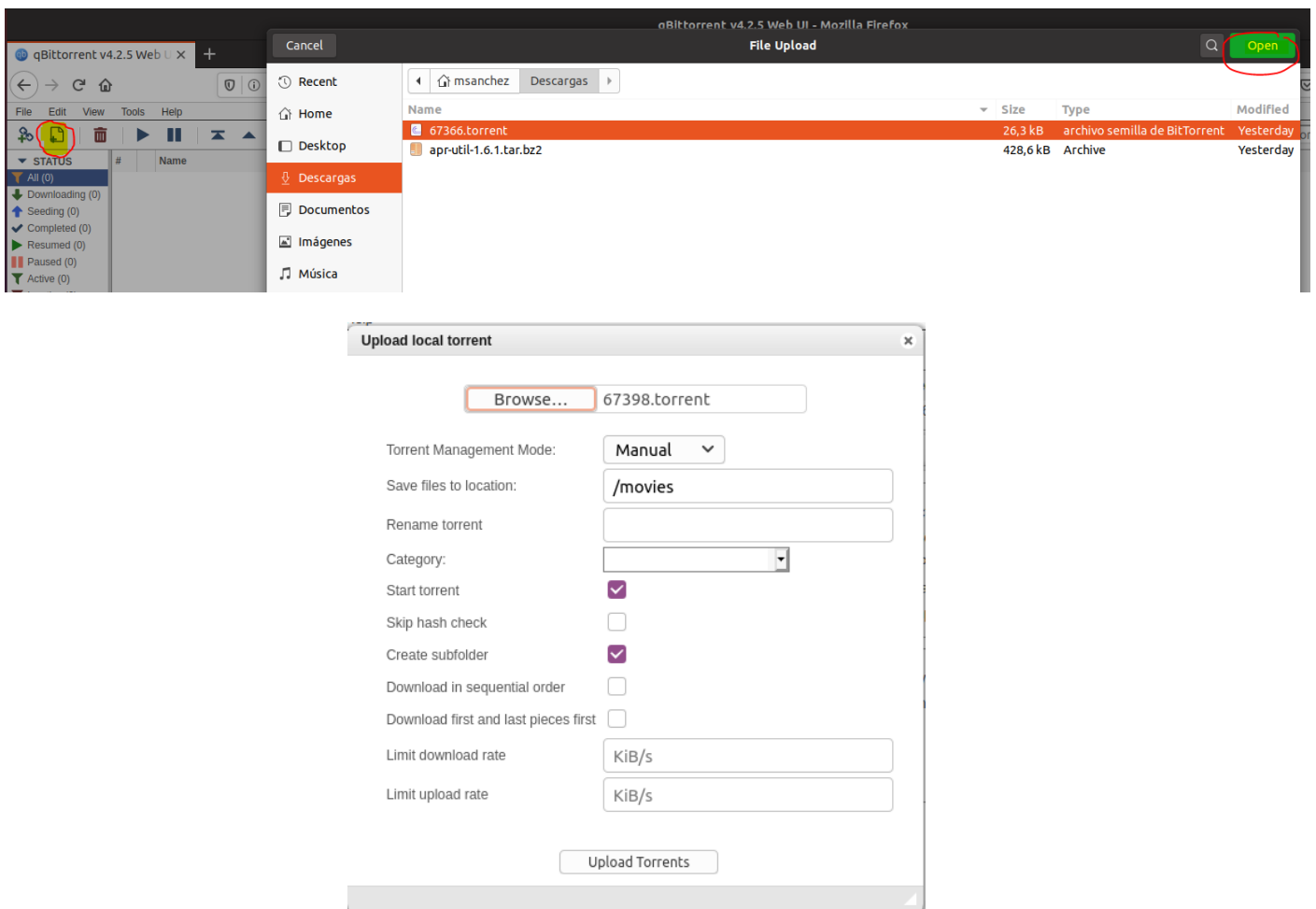
volumes:

- /home/share/plex/library:/config
- /home/share/plex/tvseries:/tv
- /home/share/plex/movies:/movies

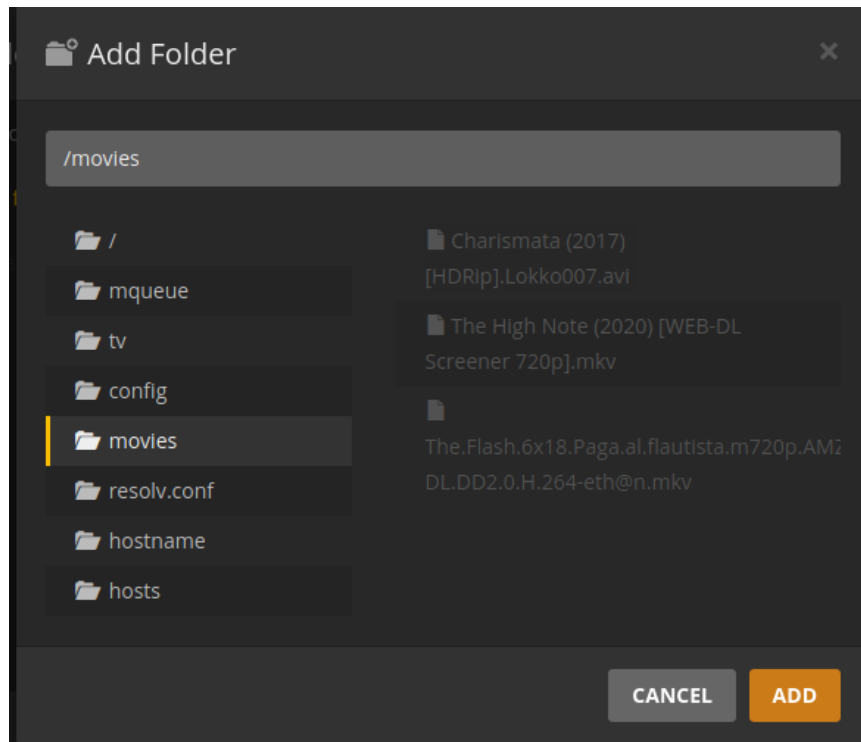
En estas líneas se marca que es un volumen compartido, lo que quiere decir es que cualquier cosa dentro de este contenedor se compartirá automáticamente con el que lo esté compartiendo también, esto quiere decir que cualquier película se añadirá automáticamente a esa carpeta (después de haber hecho una configuración previa), y que cualquier reproductor de video (en nuestro caso **plex**) será capaz de visualizarlas.

Pondré un ejemplo de funcionamiento:

-Tendremos que descargar la película/anime/serie que queremos en archivo Torrent, en este caso descarga una película (el nombre del archivo Torrent no tiene importancia)



una vez descargado está archivo, el cual en mi caso es una película la podremos ver desde el servidor de video.



Como he comentado anteriormente esto es posible gracias a que las carpetas están compartidas por un volumen el cual a través del mismo servidor y maquina host, pueden ver.

#	Name	Size	Done	Status	Seeds	Peers	Down Speed	Up Speed	ETA	Ratio	Category	Tags	Added On
1	Osake wa Fuufu ni Natte kara S01...	11.54 GiB	100.0%	Seeding	0 (6)	1 (12)	0 B/s	74.0 KiB/s	∞	0.68			6/30/2020, 5:55...
2	[Desu is dead] Super Dragon Ball ...	104.0 MiB	100.0%	Queued	0 (41)	0 (1)	0 B/s	0 B/s	∞	0.00			7/1/2020, 3:06...
3	[argo] ARGONAVIS from BanG Dr...	894.6 MiB	100.0%	Seeding	0 (5)	0 (24)	0 B/s	0 B/s	∞	0.63			7/1/2020, 2:44...
4	[New Wave] Kamen Rider Zero-O...	200.4 MiB	100.0%	Seeding	0 (5)	0 (3)	0 B/s	0 B/s	∞	1.75			6/30/2020, 6:29...

```
qbittorrent:
image: linuxserver/qbittorrent
container_name: qbittorrent
environment:
  - PUID=1000
  - PGID=1000
  - TZ=Europe/London
  - UMASK_SET=022
  - WEBUI_PORT=8909
volumes:
  - /home/shared/plex/library:/config
  - /home/shared/plex/tvseries:/tv
  - /home/shared/plex/movies:/movies
ports:
  - 6881:6881
  - 6881:6881/udp
  - 8909:8909
restart: unless-stopped
```

Instalación de apache

Para la instalación de wordpress primero se necesita tener un servidor web instalado.

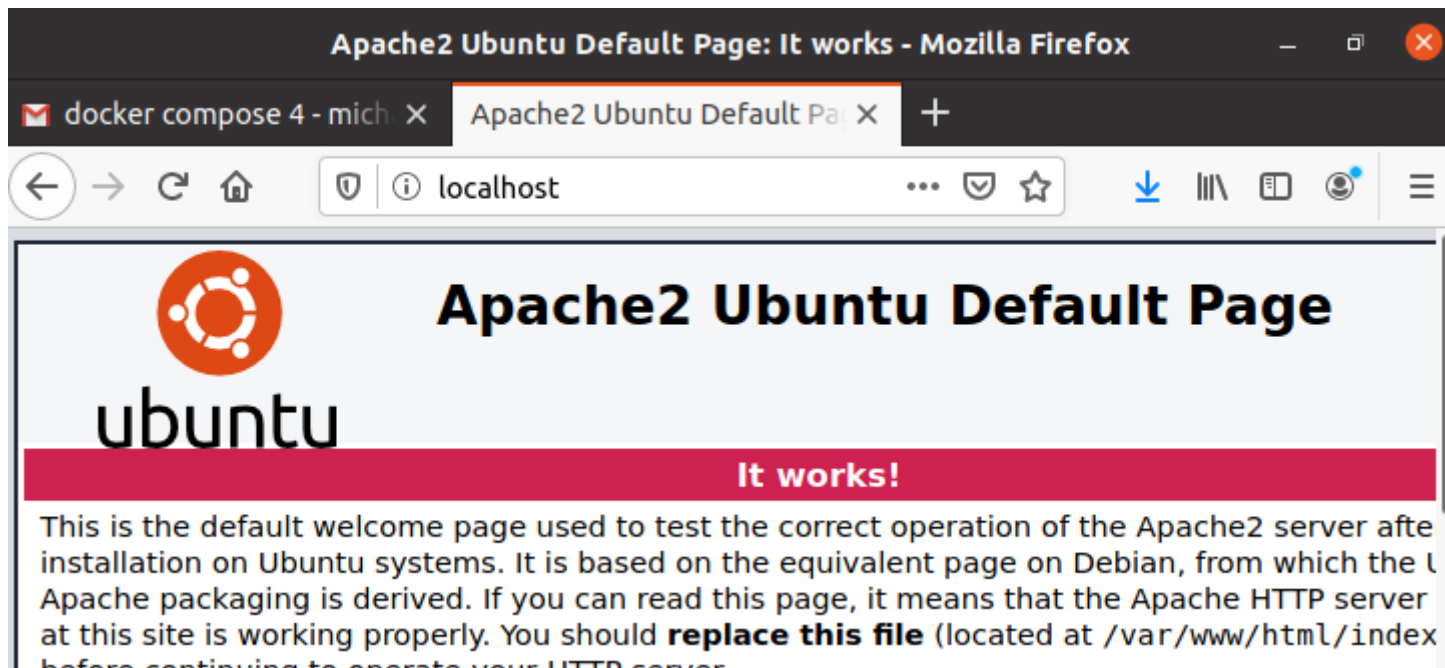
- Primero tenemos que tener actualizado nuestro sistema [apt-get update]
- Después instalamos apache2 [apt install apache2] (en todo este procedimiento tenemos que estar en sudo o ejecutarlo como tal)
- Seguidamente modificamos el firewall para que deje pasar al apache
- Utilizamos el comando [ufw app list] para ver la lista de aplicaciones disponibles a ser aceptadas o denegadas por el firewall.

```
root@msanchez:/home/msanchez/Documentos/httpd-2.4.43# ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

- Añadimos el siguiente comando [ufw allow 'Apache'] con este comando le indicamos que deje a Apache utilizar los puertos que necesite.
- Con el comando [ufw status], podemos comprobar que servicios están aceptados y si deniega o no algún parámetro.
- Por último, utilizamos systemctl status apache2 para ver si está activo y no tiene ningún fallo.

```
root@msanchez:/home/msanchez/Documentos/httpd-2.4.43# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2020-06-28 15:38:54 UTC; 41min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 61663 (apache2)
     Tasks: 55 (limit: 4615)
    Memory: 5.5M
    CGroup: /system.slice/apache2.service
            └─61663 /usr/sbin/apache2 -k start
              └─61665 /usr/sbin/apache2 -k start
                └─61666 /usr/sbin/apache2 -k start

jun 28 15:38:54 msanchez systemd[1]: Starting The Apache HTTP Server...
jun 28 15:38:54 msanchez apache2[61662]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 instead. See the error log for more details.
jun 28 15:38:54 msanchez systemd[1]: Started The Apache HTTP Server.
```



Proceso de ejecución de Wordpress.

Para poder utilizar el **Wordpress** y **MySQL** tenemos que crear en el **Docker-compose** los servicios correspondientes los cuales tendrán un volumen compartido.

```
root@msanchez: /home/msanchez/Documentos/Docker
GNU nano 4.8                                docker-compose.yml
db:
  image: mysql:5.7
  volumes:
    - db_data:/var/lib/mysql
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: msanchez
    MYSQL_DATABASE: DBplay
    MYSQL_USER: msanchez
    MYSQL_PASSWORD: msanchez

wordpress:
  depends_on:
    - db
  image: wordpress:latest
  ports:
    - "8080:80"
  restart: always
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: msanchez
    WORDPRESS_DB_PASSWORD: msanchez
    WORDPRESS_DB_NAME: DBplay
volumes:
  db_data: {}
```

Con el uso del comando **docker-compose up -d** hacemos que nos inicie los servicios mysql y wordpress

- Para poder ejecutar el .yml (archivo Docker-compose) tenemos que tener descargado he instalado tanto el Docker.io como el Docker-compose, en caso de intentar hacerlo te pide que lo instales.

```
root@msanchez:/home/msanchez/Documentos/Docker# docker-compose up -d
Command 'docker-compose' not found, but can be installed with:

snap install docker          # version 19.03.11, or
apt install docker-compose   # version 1.25.0-1

See 'snap info docker' for additional versions.

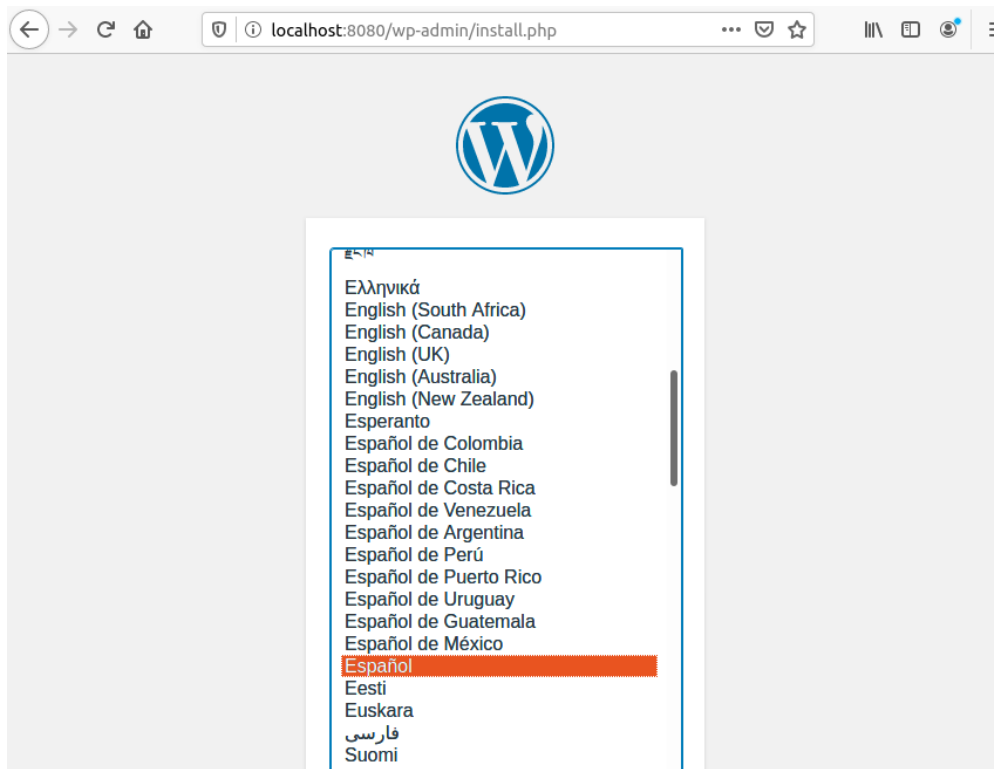
root@msanchez:/home/msanchez/Documentos/Docker# apt install docker-compose
```

```
root@msanchez:/home/msanchez/Documentos/Docker# docker-compose up -d
Creating network "docker_default" with the default driver
Creating volume "docker_db_data" with default driver
Pulling plex (linuxserver/plex)...
latest: Pulling from linuxserver/plex
bec29fbc7de: Pull complete
04255341a00e: Pull complete
a33bacabb01: Pull complete
67a79305ada5: Pull complete
ae6ab233447b: Pull complete
6f932896b93c: Pull complete
Digest: sha256:4db9358ead00bca481c9f0c26a1858e4f0e14f53a228f0a1a2ed67a900f967e8
Status: Downloaded newer image for linuxserver/plex:latest
Pulling qbittorrent (linuxserver/qbittorrent)...
latest: Pulling from linuxserver/qbittorrent
6b59f4d42254: Pull complete
875de62a65e8: Pull complete
7af6c5f56812: Pull complete
c7bd6e473c84: Pull complete
3a5530c35d7: Pull complete
```

```
Digest: sha256:56cc33a35f71454a2f7a8b09a40b3184d99127be3c668527b8f95419934566eb
Status: Downloaded newer image for linuxserver/qbittorrent:latest
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
8559a31e96f4: Pull complete
d51ce1c2e575: Pull complete
c2344adc4858: Pull complete
fcf3ceff18fc: Pull complete
16da0c38dc5b: Pull complete
b905d1797e97: Pull complete
4b50d1c6b05c: Pull complete
d85174a87144: Pull complete
a4ad33703fa8: Pull complete
f7a5433ce20d: Pull complete
3dcd2a278b4a: Pull complete
Digest: sha256:32f9d9a069f7a735e28fd44ea944d53c61f990ba71460c5c183e610854ca4854
Status: Downloaded newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
8559a31e96f4: Already exists
e0276193a084: Pull complete
eb2d00c10344: Pull complete
f54006e0dc29: Pull complete
e0d3d1244592: Pull complete
3a60f364b0c5: Pull complete
3a3000000000: Pull complete
```

```
Digest: sha256:ca4e9b3078ce3497cfbadea6df299a6c46fceddc7eac04958207ef3e4d2edaa8
Status: Downloaded newer image for wordpress:latest
Creating plex ... done
Creating docker_db_1 ... done
Creating qbittorrent ... done
Creating docker_wordpress_1 ... done
```

Para acceder a la **Wordpress**, es tan simple como el haber hecho correctamente la instalación con el Docker-compose, si se ha hecho correctamente lo único que deberemos de hacer es acceder a <http://localhost:8080> que es el puerto que hemos añadido para wordpress y ya tendremos nuestros pasos de instalación a continuación.



Para tener nuestro **Wordpress** operativo, únicamente hace falta seguir los pasos de instalación que tiene el mismo instalador, e ir añadiendo los datos que queremos que tenga nuestra página web.

Información necesaria

Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Nombre de usuario

Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña

[Ocultar](#)

Fuerte

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Tu correo electrónico

Comprueba bien tu dirección de correo electrónico antes de continuar.

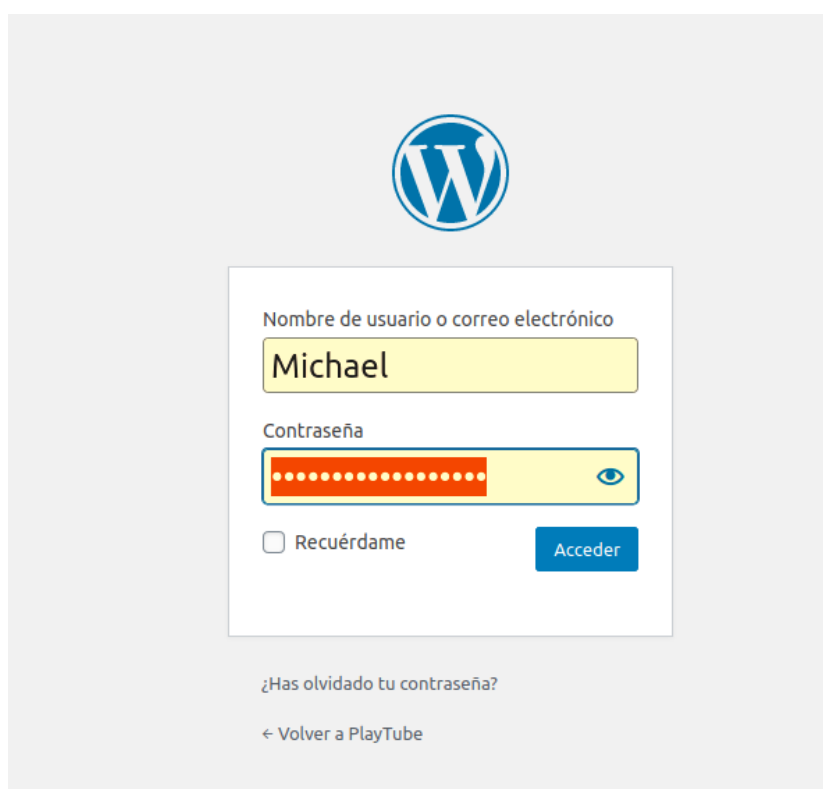
Visibilidad en los motores de búsqueda

☐ Disuadir a los motores de búsqueda de indexar este sitio

Depende de los motores de búsqueda atender esta petición o no.

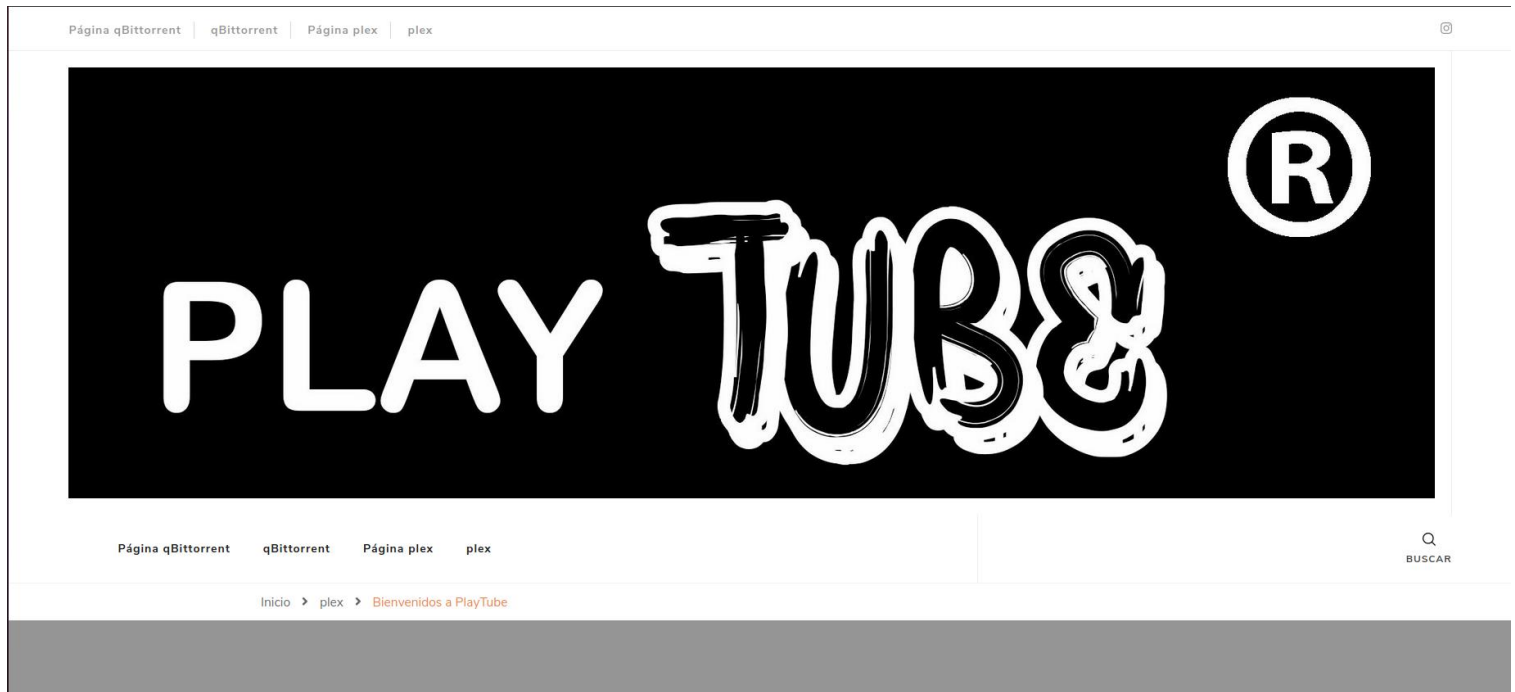
[Instalar WordPress](#)

Para finalizar, entraremos con nuestro usuario y contraseña de administrado añadida anteriormente:



The image shows the WordPress login interface. At the top center is the WordPress logo. Below it is a white login box with a yellow border. Inside the box, the first field is labeled 'Nombre de usuario o correo electrónico' and contains the text 'Michael'. The second field is labeled 'Contraseña' and contains a series of dots, with a small eye icon to its right. Below the password field is a checkbox labeled 'Recuérdame'. To the right of the checkbox is a blue button labeled 'Acceder'. Below the login box, there is a link that says '¿Has olvidado tu contraseña?' and a link that says '← Volver a PlayTube'.

Y una vez en la página, se verá nuestra página **wordpress** (en esta imagen ya se ve mi página con más consistencia, la principal estará vacía)



Copia de seguridad con Wordpress.

Para poder hacer el backup de **wordpress** requerimos de software exterior al que viene predeterminadamente en **wordpress** en el momento de la instalación.

Aquí dejo un [enlace](#) al video en el que explico cómo se hace el backup.

Docker-Swarm



¿Que es Docker Swarm?

Este es una herramienta que nos permite a los desarrolladores crear contenedores he implementarlos en modo swarm. Un clúster Swarm consiste en hacer un Docker el cual implemente su tecnología Engine Implmentation entrando en múltiples nodos. Los nodos de administración realizan la orquestación y la administración del clúster. Los nodos de trabajo reciben y ejecutan tareas desde los nodos de administración.

Esta tecnología era desconocida para mi hasta este año, en el que curse el modulo de M11 (**Seguretat i Alta Disponibilitat**)

Al implementarlo en mi sistema, me he dado cuenta que es una tecnología muy útil y que trabaja al nivel de un RAID tipo 5, el cual se encarga de proteger la información en caso de que alguno de los discos caiga

Las ventajas que tendría al usar esta tecnología.

- Se tiene un mayor control sobre el trabajo que se reparte entre todos los servicios, haciendo así que se pueda derivar una carga a otro swarm y así aumentar la eficiencia del trabajo (trabajo en equipo seria la equivalencia)

- Las aplicaciones se pueden implementar como servicios (o "microservicios") en un clúster Swarm. Las aplicaciones de varios contenedores pueden especificarse utilizando archivos YAML. Docker Compose puede implementar la aplicación. Las tareas se pueden distribuir a través de centros de datos usando etiquetas.

Docker Swarm

¿Qué características tiene Docker Swarm y cómo trabaja?

Nodo: Es una instancia de un Swarm (cluster).

Swarm: Es el cluster de los nodos en los que está instalado Docker Engine. En modo Swarm se despliegan servicios en lugar de contenedores.

Nodo Manager: Es el que gestiona el cluster y programa los servicios en los distintos nodos. Puede existir más de uno y pueden correr imágenes en él también.

Nodo Worker: Serán los nodos donde corren las tareas de los servicios.

Servicio: Es la propia imagen, el contenedor. Se pueden crear varias réplicas de estos servicios.

Tarea: Es una unidad atómica de un servicio que corre en un nodo, es decir una réplica de un servicio. Se ejecutan independientemente una de la otra.

Despliegue de mi proyecto con docker Swarm.

Docker init para especificar la maquina manager.

```
root@msanchez:/home/msanchez# docker swarm init --advertise-addr 192.168.0.27
Swarm initialized: current node (9l69bszc76ovho9w8ydxozf8f) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-51zd8its26u79qrozgon4km28m5hb70vgy23m3lyxw3e7x0rjq-ee8tr14wiyhrbasf3k2i96os5 192.168.0.27:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

Lo siguiente será copiar el **token** en una máquina que será un worker

PlayTUBE

```
root@msanchez1-VirtualBox:/home/msanchez1# docker swarm join --token SWMTKN-1-51zd8its26u79qrozgon4km28m5hb70vgy23m3lyxw3e7x0rjq-ee8tr14wiyhrbasf3k2i96os5 192.168.0.27:2377
This node joined a swarm as a worker.
root@msanchez1-VirtualBox:/home/msanchez1#
```

verificamos en el manager

Lo siguiente será hacer el deploy en el manager para que se despliegue en los workers

```
root@msanchez:/home/msanchez/Documentos/Docker# docker stack deploy -c docker-compose.yml playtube
Ignoring unsupported options: network_mode, restart
Ignoring deprecated options:
container_name: Setting the container name is not supported.
Creating network playtube_default
Creating service playtube_plex
Creating service playtube_qbittorrent
Creating service playtube_db
Creating service playtube_wordpress
root@msanchez:/home/msanchez/Documentos/Docker# docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
9169bszc76ovho9w8ydxozf8f *	msanchez	Ready	Active	Leader	19.03.8
c17s6rzqwx1fadcm1ajx19899	msanchez1-VirtualBox	Ready	Active		19.03.6

comprobación.

```
root@msanchez:/home/msanchez/Documentos/Docker# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
ad0789a0610d	linuxserver/qbittorrent:latest	"/init"	10 minutes ago
Up 10 minutes	6881/tcp, 8080/tcp, 6881/udp		
f88eefa2ee18	linuxserver/plex:latest	"/init"	10 minutes ago
Up 10 minutes	1900/udp, 3005/tcp, 8324/tcp, 5353/udp, 32410/udp, 32400/tcp, 32412-32414/udp, 32469/tcp		
147a51b10954	mysql:5.7	"docker-entrypoint.s..."	11 minutes ago
Up 11 minutes	3306/tcp, 33060/tcp		
fd519ab3807e	linuxserver/plex	"/init"	7 days ago
Up 21 minutes	plex		
4af1c2eb4895	linuxserver/qbittorrent	"/init"	7 days ago
Up 21 minutes	0.0.0.0:6881->6881/tcp, 0.0.0.0:8909->8909/tcp, 0.0.0.0:6881->6881/udp, 8080/tcp		
f342d33260b4	wordpress:latest	"docker-entrypoint.s..."	9 days ago
Up 21 minutes	0.0.0.0:8080->80/tcp		
9943fbf06de3	mysql:5.7	"docker-entrypoint.s..."	9 days ago
Up 21 minutes	3306/tcp, 33060/tcp		
	docker_wordpress_1		
	docker_db_1		

```
root@msanchez:/home/msanchez/Documentos/Docker#
```

Prueba de proyecto con video

Video de prueba.

<https://youtu.be/5sSwKcyHEt4>

¿Por qué utilizo Docker?

1. Estandarización y productividad

Los contenedores Docker garantizan consistencia a lo largo del proceso de desarrollo y despliegue de una aplicación, estandarizando el entorno en el que se ejecuta. Una de las principales ventajas de una arquitectura basada en Docker es precisamente la estandarización. Docker permite desarrollos, compilaciones, pruebas y entornos de producciones repetibles y replicables y sin necesidad de compartir enormes archivos.

2. Eficiencia de imágenes de contenedor

Docker te permite crear una imagen de contenedor y usar esa misma imagen a lo largo de todo el proceso de despliegue. Una gran ventaja de esto es la capacidad de separar pasos no dependientes del proceso y ejecutarlos en paralelo. El tiempo que va desde la compilación a la producción se puede acelerar bastante.

3. Simplicidad y configuraciones más rápidas

Una de las ventajas principales de Docker es la forma en la que simplifica las cosas. Los desarrolladores pueden coger su propia configuración, transformarlo en código y desplegarlo sin problemas. Como Docker se puede usar en una amplia variedad de entornos, los requisitos de la infraestructura ya no quedan vinculados al entorno de la aplicación.

4. Despliegue y escalabilidad rápidos

Docker consigue reducir el despliegue a cuestión de segundos. Esto se debe al hecho de que crea un contenedor para cada proceso y no arranca un sistema operativo. Los datos se pueden crear y eliminar sin temer que el coste de tenerlo que arrancarlo todo otra vez sea mayor de lo que se pueda permitir.

5. Pruebas continuas

Docker garantiza entornos consistentes desde la fase de desarrollo hasta la fase de producción. Los contenedores Docker se configuran para que conserven todas las configuraciones y dependencias internamente. Así, se puede usar el mismo contenedor desde desarrollo a producción con la seguridad de que no hay discrepancias ni intervención manual.

5. Seguridad

Una última ventaja en este apartado de usar Docker es la de la seguridad. Desde el punto de vista de la seguridad Docker garantiza que las aplicaciones que se están ejecutando en los contenedores estén totalmente segregadas y aisladas las unas de las otras, como ya hemos comentado, otorgando un control total, sobre la gestión y el tráfico de flujos. Ningún contenedor Docker puede entrar a ver los procesos que se están ejecutando dentro de otro contenedor. Desde un punto de vista de arquitectura del software, cada contenedor obtiene su propio conjunto de recursos que van desde las pilas de procesamiento hasta las de red.

Conclusión personal.

La verdad es que creo que me ha quedado un sistema muy pulido que se podría utilizar más concretamente para el ámbito personal, considero que no hace falta mencionar que siempre están las opciones vía web para hacer lo que mi pagina se encarga de hacer, aun que siempre es una opción viable para la gente que le guste tener control sobre lo que tiene para ver y hacer.

Plex es un servicio que no esta mal del todo, pero personalmente no lo utilizaría ya que los servicios que ofrece a la gente son “limitados” y prefiero buscar otra alternativa que, aunque me lleve más tiempo encontrar cientos de artículos tenerlos más al alcance en cualquier momento.

¿Que cambiaría?

El hecho de que para subir una película se necesite tener un Torrent para subirlas al mismo, creo que una opción de subida manual ayudaría a los usuarios más novatos y lo utilizarían más personas por ello, aparte de que estaría muy bien que te detectara automáticamente de que es un archivo (película, serie, etc..).

Webgrafica

<https://hub.docker.com/r/linuxserver/qbittorrent/>

<https://www.adslzone.net/reportajes/descargas/descargar-torrent-qbittorrent/>

<https://github.com/qbittorrent/qBittorrent/issues/7954>

<https://hub.docker.com/r/linuxserver/plex/>

<https://forums.plex.tv/t/problemas-con-plex-media-server/546040>

https://www.youtube.com/channel/UCrBzBOMcUVV8ryyAU_c6P5g

<https://www.youtube.com/watch?v=Ceqb53EXANK>

<https://en.wikipedia.org/wiki/QBittorrent>

<https://bitbucket.org/sshguard/sshguard/src/master/examples/sshguard.conf.sample>

[Mucha de la información saca en este proyecto es aprendizaje de mi curso ASIX]