

Documentación » Stacks de Docker

Stacks de Docker

Stacks de Docker

Anonymous · 25/09/2026

Stacks de Docker

Nuestros [stacks de Docker](#) permiten instalar en tu propio servidor cada plataforma que se integra con BillMySales, lista para producción. Son gratuitos y de código abierto (licencia MIT), y todos siguen las mismas convenciones: si sabes usar uno, sabes usarlos todos.

Esta guía explica lo que es común a todos los stacks. Lo propio de cada plataforma (variables obligatorias, recursos, actualizaciones y cómo restaurar un respaldo) está en la guía **Instalación con Docker** de cada origen de datos, y el detalle completo, en el README de su repositorio.

¿Prefieres que lo instalemos por ti?

Evaluamos y cotizamos la instalación en tu servidor, dejándola conectada con BillMySales. [Conversemos](#).

Qué incluye cada stack

- La plataforma, con su base de datos y los servicios que necesita (tareas programadas, colas, caché), cada uno en su propio contenedor.
- El servidor web Caddy, con HTTPS automático. Es el único servicio que publica puertos.
- Respaldos periódicos de la base de datos y de los archivos que no se pueden regenerar.
- Configuración segura por defecto y un límite de memoria en cada servicio.

Requisitos

- Un servidor con **Docker Engine 24 o superior** y **Docker Compose v2.24 o superior**, además de **Git** para descargar el stack.
- Los puertos **80 y 443** accesibles desde internet.
- Un **dominio** con su registro DNS apuntando a la IP del servidor.
- Un servicio de **correo SMTP**, para que la plataforma pueda enviar correos (confirmaciones, recuperación de contraseñas, notificaciones).

La memoria y el disco necesarios dependen de cada plataforma y están indicados en su guía.

Instalación

Descarga el stack de tu plataforma y crea el archivo de configuración a partir de la plantilla de producción. Por ejemplo, para WooCommerce:

```
git clone https://github.com/BillMySales/billmysales-docker-woocommerce.git
cd billmysales-docker-woocommerce
cp .env.prod.example .env
```

Abre el archivo `.env` con tu editor y completa las variables obligatorias: la URL pública de la plataforma, el dominio en `SITE_ADDRESS`, las contraseñas y claves secretas, y los datos de SMTP. Cada variable está documentada en el mismo archivo, y las obligatorias de cada plataforma están listadas en su guía.

Usa contraseñas y claves largas y aleatorias. Puedes generarlas con el comando que sugiere el `.env` de cada stack, normalmente:

```
openssl rand -hex 32
```

El formato hexadecimal evita caracteres como `/`, `+` o `=`, que pueden romper las URL de conexión a la base de datos.

Algunas plataformas necesitan claves con un formato especial; en ese caso, su guía indica cómo generarlas.

El cifrado del correo se define con `SMTP_SECURE`: `tls` (STARTTLS, normalmente en el puerto 587), `ssl` (SMTPS, puerto 465) o `none` (sin cifrado). El valor por defecto de cada stack está indicado en su `.env.prod.example`; ajústalo según tu proveedor de correo.

Levantar la plataforma

```
docker compose up -d
docker compose logs -f setup
```

El stack no arranca si falta una variable obligatoria, y te indica cuál. El servicio `setup` instala la plataforma la primera vez y en los siguientes arranques aplica las migraciones necesarias; se puede ejecutar cuantas veces quieras. Cuando sus registros terminan con `==> Done`, la plataforma está lista en tu dominio.

Para revisar que todo esté bien:

```
docker compose ps
```

Todos los servicios deben aparecer como `healthy`, y `setup` como `Exited (0)`.

HTTPS

Con tu dominio en `SITE_ADDRESS`, Caddy obtiene un certificado de Let's Encrypt y lo renueva automáticamente, sin configuración adicional. Si el servidor ya tiene otro proxy que se encarga del HTTPS, usa `SITE_ADDRESS=:80`.

Detrás de Traefik

Si ya usas Traefik en el servidor, el stack puede publicarse a través de él sin exponer puertos propios. En el `.env`, además de lo anterior, deja estas variables (con la variable de URL que corresponde a tu plataforma):

```
COMPOSE_FILE=compose.yaml:overrides/traefik.yaml
TRAEFIK_HOST=tienda.example.com
WP_URL=https://tienda.example.com
SITE_ADDRESS=:80
```

Por defecto el stack se conecta a la red de Docker `shared_network` y usa el endpoint `websecure` y el certresolver `letsencrypt`. Si tu Traefik usa otros nombres, ajusta `TRAEFIK_NETWORK`, `TRAEFIK_ENTRYPOINT` y `TRAEFIK_CERTRESOLVER`.

Respaldos

En producción los respaldos quedan activos desde el primer arranque: se guardan al iniciar y luego cada 24 horas, y se eliminan los que tienen más de 7 días. Puedes cambiarlo con `BACKUP_INTERVAL_HOURS` y `BACKUP_KEEP_DAYS`.

```
docker compose run --rm --no-deps backup now
docker compose run --rm --no-deps backup list
```

El primer comando crea un respaldo en el momento y el segundo lista los disponibles. Los pasos para restaurar uno dependen de cada plataforma (qué servicios detener y qué hacer después) y están en su guía.

Copia tus respaldos fuera del servidor

Los respaldos se guardan en el mismo servidor. Si el servidor falla, se pierden junto con la plataforma. Cópalos periódicamente a otro lugar.

Actualizaciones

Cada componente usa una versión fija, definida en una variable del `.env`. Según la plataforma, la aplicación se actualiza de una de estas dos formas:

- **Desde su panel de administración**, como en cualquier instalación (por ejemplo, WooCommerce o PrestaShop).
- **Cambiando la versión en el `.env`** y volviendo a levantar el stack; `setup` aplica las migraciones de la nueva versión.

La guía de cada plataforma indica cuál corresponde. La URL pública se toma siempre del `.env`: si cambias de dominio, actualízala junto con `SITE_ADDRESS` y ejecuta `docker compose up -d`.

Comandos útiles

Comando	Para qué sirve
<code>docker compose ps</code>	Ver el estado de los servicios.
<code>docker compose logs -f</code>	Ver los registros de todos los servicios.
<code>docker compose down</code>	Detener la plataforma sin borrar datos.

Nunca uses ``docker compose down -v`` en producción

La opción `-v` elimina los volúmenes: la base de datos, los archivos de la plataforma y los respaldos.

Seguridad

Los stacks vienen configurados de forma segura por defecto:

- Contraseñas y claves obligatorias, sin valores por defecto en producción.
- La base de datos no se expone fuera del servidor.
- Modo de depuración apagado y errores que nunca se muestran al visitante.
- Cabeceras de seguridad, versión de PHP oculta cuando aplica y rutas sensibles bloqueadas.

No incluyen un firewall de aplicaciones web, límite de intentos de inicio de sesión ni copias de respaldo fuera del servidor. Si los necesitas, agrégalos con las herramientas de tu plataforma o de tu proveedor.

La plantilla de desarrollo (`.env.dev.example`) usa contraseñas públicas: nunca la uses en un servidor.

Última actualización el 25/09/2026