

Instalación con Docker

Instalar Medusa con Docker

Anonymous · 25/09/2026

Instalar Medusa con Docker

Sigue la [guía de los stacks de Docker](#) para instalar Medusa con nuestro [stack](#) y ten en cuenta lo propio de esta plataforma. El detalle completo está en el README del repositorio.

Recursos

El stack necesita 1 GB de RAM y alrededor de 1 GB de disco para las imágenes. En reposo usa unos 300 MiB en el servidor de Medusa, 220 MiB en el worker, 60 MiB en PostgreSQL, 10 MiB en Redis y 11 MiB en Caddy.

La imagen de Medusa se construye en tu servidor, por lo que el stack se levanta con `docker compose up -d --build` en vez de `docker compose up -d`. La primera construcción tarda unos 3 minutos y genera una imagen de unos 650 MB.

Variables obligatorias

Variable	Qué debe contener
MEDUSA_URL	URL pública de Medusa, por ejemplo <code>https://tienda.example.com</code> .
SITE_ADDRESS	El dominio, por ejemplo <code>tienda.example.com</code> .
DB_PASSWORD	Contraseña de la base de datos.
MEDUSA_JWT_SECRET y MEDUSA_COOKIE_SECRET	Claves secretas de Medusa.
MEDUSA_ADMIN_EMAIL y MEDUSA_ADMIN_PASSWORD	Correo y contraseña del administrador. Se crea si no existe un usuario con ese correo (si cambias el correo después, se agrega otro administrador), y la contraseña solo se aplica al crearlo.
SMTP_HOST	Recomendada: el stack arranca sin ella, pero la plataforma no podrá enviar correos. Servidor SMTP. Completa también SMTP_PORT, SMTP_SECURE, SMTP_USER, SMTP_PASSWORD y SMTP_FROM según tu proveedor. SMTP_SECURE es <code>tls</code> por defecto (STARTTLS obligatorio).

Configuración inicial de la tienda

La tienda se crea configurada para Chile: moneda `clp`, una región para el país `cl` con IVA de 19 % incluido en los precios, una bodega en Santiago con envío gratuito “Despacho” y el canal de venta por defecto con su clave pública de API. Si vendes en otro país, ajusta `MEDUSA_CURRENCY`, `MEDUSA_COUNTRY`, `MEDUSA_REGION_NAME`, las variables `MEDUSA_TAX_*`, `MEDUSA_PRICES_INCLUDE_TAX`, `MEDUSA_STOCK_LOCATION` y `MEDUSA_CITY` **antes** del primer arranque: estos valores solo se aplican en la instalación inicial y después se modifican desde el panel.

`MEDUSA_STORE_NAME` define el nombre de la tienda solo en la instalación inicial, pero los correos lo leen en cada arranque para su encabezado y sus asuntos. También en cada arranque, los correos formatean los montos con `MEDUSA_LOCALE` (por defecto, es-CL: \$19.980).

Acceso

Cuando `setup` termina, el panel de administración queda en `/app`, con el usuario `MEDUSA_ADMIN_EMAIL`, y la Store API en `/store`. La Store API se usa con la cabecera `x-publishable-api-key`, cuyo valor muestra `setup` en sus registros y también aparece en el panel, en Settings > Publishable API Keys.

Actualizaciones

Medusa se actualiza cambiando la versión en el `.env`. Respalda primero, cambia `MEDUSA_VERSION` y ejecuta `docker compose up -d --build`: la imagen se reconstruye con la nueva versión y `setup` aplica las migraciones antes de que arranque Medusa. Si una migración falla, se revierten todas las de esa ejecución.

Las migraciones siempre se aplican. Lo que se omite es la parte insegura de la sincronización de los enlaces entre módulos y los índices de búsqueda (eliminar tablas o índices de enlaces que ya no existen). Para aplicarla también, ejecuta una vez:

```
MEDUSA_MIGRATE_UNSAFE=true docker compose up -d
```

Restaurar un respaldo

Los respaldos incluyen la base de datos y los archivos subidos. Para restaurar uno:

```
docker compose stop medusa worker
docker compose run --rm --no-deps backup restore <fecha>
docker compose exec redis redis-cli FLUSHALL
docker compose up -d
```

El comando de Redis limpia la caché y las colas de los datos anteriores. La base de datos debe estar

en ejecución (`docker compose up -d db` si el stack está detenido). La restauración reemplaza la base de datos por la del respaldo, así que no queda nada de lo creado después.

Notas

- Medusa es *headless*: el stack incluye el panel y las APIs, pero **no una tienda pública** para tus clientes. La tienda es una aplicación aparte que usa la Store API; el punto de partida habitual es el [Next.js Starter Storefront](#) de Medusa.
- Si la tienda se ejecuta en el navegador desde otro dominio, agrégalo en `MEDUSA_STORE_CORS`.
- Los correos de recuperación de contraseña de los clientes enlazan a `MEDUSA_STOREFRONT_URL/reset-password`. Sin esa variable, no se envían.

Siguiente paso

Con Medusa funcionando, configúralo como origen de datos en BillMySales siguiendo la [configuración inicial](#).

Última actualización el 25/09/2026