

Guía Definitiva: Despliegue de Laravel 11 en Hostinger

Este Runbook detalla el proceso completo y verificado para desplegar una aplicación Laravel 11 en un entorno de hosting compartido de Hostinger, basado en lecciones aprendidas.

Fase 0: Preparación del Proyecto (Local)

Antes de siquiera pensar en el servidor, asegúrate de que tu proyecto local esté impecable.

1. Requisitos de Versión:

- Tu composer.json debe especificar php >= 8.2.
- Asegúrate de estar usando PHP 8.2 o superior en tu entorno local (XAMPP).

2. Configuración de Laravel 11:

- **Rutas API:** Confirma que el archivo routes/api.php existe. Si no, créalo vacío.
- **Carga de Rutas:** Verifica que bootstrap/app.php cargue explícitamente las rutas de API:
 - ->withRouting(
 - web: __DIR__.'/../routes/web.php',
 - api: __DIR__.'/../routes/api.php', // ¡Esencial!
 - commands: __DIR__.'/../routes/console.php',
 - health: '/up',
 -)
- **Middleware:** Registra cualquier alias de middleware (como role) en bootstrap/app.php dentro de ->withMiddleware(...):
 - \$middleware->alias([
 - 'role' => \App\Http\Middleware\EnsureRole::class,
 - // ...otros alias
 -]);

3. Archivo de Entorno Ejemplo (.env.example):

- Asegúrate de que contenga **todas** las variables de entorno necesarias para producción, pero con valores *dummy* o vacíos (¡sin secretos!).
- Incluye variables para la base de datos, URL de la aplicación, correo, y cualquier servicio externo.

4. Control de Versiones (Git):

- Asegúrate de que tu archivo .gitignore ignore vendor/, .env, storage/ (excepto storage/app/public), y bootstrap/cache/*.php.
- Haz commit de todos tus cambios y súbelos a tu repositorio remoto (GitHub, GitLab, etc.).

Fase 1: Preparación en Hostinger (hPanel)

Configura el entorno en el servidor.

1. Crear Base de Datos:

- Ve a Bases de datos -> Administración de Bases de Datos MySQL.
- Crea una base de datos y un usuario.
- **¡CRÍTICO!** Anota el **nombre completo** de la base de datos y el **usuario** (incluyendo el prefijo uXXXX_) y la contraseña.

2. Establecer Versión de PHP (Web):

- Ve a Avanzado -> Configuración de PHP.
- Selecciona **PHP 8.2 o PHP 8.3**. Esta es la versión que ejecutará tu sitio web.

3. Obtener Credenciales SSH:

- Ve a Avanzado -> Acceso SSH. Anota el Host, Puerto y Usuario.

Fase 2: Despliegue Inicial (SSH)

Conéctate al servidor y sube el código.

1. **Conexión SSH:** Conéctate usando tu cliente SSH.
2. **Navegación:** Ve al directorio raíz de tu dominio (ej. cd ~/domains/tu-dominio.com/public_html). **No crees subcarpetas.**
3. **Clonar Repositorio:** Clona tu proyecto directamente en la carpeta actual.
4. git clone URL_DE_TU_REPO .

Fase 3: Configuración del Entorno en Servidor (SSH)

Aquí aplicamos todas las lecciones aprendidas.

1. Identificar y Configurar PHP CLI:

- **Encuentra la ruta:**
- `ls -d /opt/alt/php*/usr/bin/php`

(Elige la ruta de 8.2 o 8.3, ej. `/opt/alt/php83/usr/bin/php`).

- **Crea un Alias (¡MUY RECOMENDADO!):**
- `echo "alias php='/opt/alt/php83/usr/bin/php'" >> ~/.bashrc && source ~/.bashrc`
- **Verifica:** `php -v` (debe mostrar 8.3.x).

2. Instalar Dependencias: Ahora puedes usar php directamente gracias al alias.

3. `php /usr/local/bin/composer install --no-dev --optimize-autoloader`

4. Configurar .env:

- Copia el ejemplo: `cp .env.example .env`
- Edita con nano `.env`:
 - `APP_ENV=production`
 - `APP_DEBUG=false`
 - `APP_URL=https://tu-dominio.com`
 - `DB_DATABASE=uXXXX_nombre_bd` (¡Nombre completo!)
 - `DB_USERNAME=uXXXX_usuario_bd` (¡Usuario completo!)
 - `DB_PASSWORD="la_contraseña_bd"`
 - Configura el resto (correo, etc.).
- Genera la clave: `php artisan key:generate`

5. Preparar Base de Datos:

- **Opción A (Recomendada si tienes .sql):** Sube tu archivo .sql y ejecútalo desde SSH o impórtalo desde phpMyAdmin.
- # (Si usas SSH)
- mysql -u uXXXX_usuario_bd -p uXXXX_nombre_bd < tu_archivo.sql

- **Opción B (Si todo está en migraciones Laravel):**
- php artisan migrate --force

- **Migraciones de Laravel (¡SIEMPRE NECESARIO!):** Ejecuta esto después de A o B para crear las tablas internas (sessions, tokens, etc.).
- php artisan migrate --force

- **Seeders (Si aplica):**
- php artisan db:seed --force

6. Configurar Enrutamiento (La Solución de los "Dos Recepcionistas"):

- **Crea .htaccess en public_html (Raíz del Dominio):**
- cat > .htaccess <<'HTEOF'
- RewriteEngine On
- RewriteCond %{REQUEST_URI} !^/public/
- RewriteRule ^(.*)\$ public/\$1 [L, QSA]
- HTEOF

- **Verifica/Ajusta .htaccess en public_html/public (El de Laravel):**
Asegúrate de que exista y que contenga la línea RewriteBase / justo después de RewriteEngine On.

7. Ajustar Permisos:

8. chmod -R 775 storage bootstrap/cache

9. **Optimizar la Aplicación:**

10. `php artisan optimize`

Fase 4: Verificación y Troubleshooting

1. **Prueba la URL Raíz:** Abre <https://tu-dominio.com> en tu navegador. Deberías ver tu aplicación Laravel (no un error 403 o 404).
2. **Prueba un Endpoint de API:** Usa Postman o curl para probar una ruta de tu API (ej. `/api/login`).
 - **Respuesta esperada para curl -I .../api/login:** 405 Method Not Allowed.
 - **Respuesta esperada para POST .../api/login:** 200 OK o 201 Created con datos JSON.
3. **Si obtienes Error 500:** Revisa `storage/logs/laravel.log` en el servidor.
4. **Si obtienes Error 404:**
 - Verifica ambos archivos `.htaccess`.
 - Limpia la caché de rutas: `php artisan route:clear` y luego `php artisan optimize`.
 - Verifica que la ruta existe con `php artisan route:list`.
 - **Considera desactivar temporalmente la CDN de Hostinger** (`hPanel -> Rendimiento -> CDN`) como último recurso de diagnóstico.
5. **Si obtienes Error 403:** Revisa los permisos de las carpetas y archivos (`chmod`), y asegúrate de que el `.htaccess` de la raíz esté correcto.

Fase 5: Despliegues de Actualización

Mucho más simple:

1. SSH a `public_html`.
2. `git pull origin main` (o la rama que uses).
3. `php /usr/local/bin/composer install --no-dev --optimize-autoloader`
4. `php artisan migrate --force` (si hay nuevas migraciones).
5. `php artisan db:seed --force` (si hay nuevos seeders).
6. `php artisan optimize`

¡Guarda esta guía! Te servirá como referencia infalible para futuros despliegues.