

# cPanel Shared Hosting Deployment Guide

## Table of Contents

Paths used in examples.....	1
Option A (preferred): Domain → `www/public`.....	1
Option B (cannot change Document Root): public_html bootstrap.....	1
1) Initial setup inside `www`.....	1
2) Bootstrap `public_html`.....	2
3) Expose storage files in public_html.....	3
4) Permissions.....	3
5) Optimize caches.....	3

This guide explains how to deploy and update the Laravel app on cPanel shared hosting. It covers both: (A) pointing the domain to the app's `public/` and (B) the public\_html bootstrap method when you cannot change the Document Root.

## Paths used in examples

- Repo/app root: `~/www` (your cloned project)
- Public web root: `~/public\_html` (unmodifiable document root)
- Replace `~` with your actual home if needed (e.g. `/home/USERNAME`).

## **Option A (preferred): Domain → `www/public`**

If your hosting allows changing the Document Root, set it to `www/public` in cPanel → Domains. Then skip to "Initial setup inside www" below and ignore the public\_html bootstrap steps.

## Option B (cannot change Document Root): public\_html bootstrap

When you cannot point the domain to `public/`, keep the app in `~/www` and serve it through `~/public\_html`.

### 1) Initial setup inside `www`

```
bash
# Clone the repository to ~/www (done once)
cd ~
rm -rf www
git clone <YOUR_REPO_URL> www

cd ~/www

# Install PHP dependencies (server must have Composer)
composer install --no-dev --prefer-dist --optimize-autoloader

# Create .env (or copy from template) and set values
cp .env.example .env 2>/dev/null || true

# Generate app key (if .env APP_KEY is empty)
php artisan key:generate

# Set important env in .env
# FILESYSTEM_DISK=public
# APP_URL=https://your-domain
# DB_* according to your database

# Run migrations (and seed if needed)
php artisan migrate --force
# php artisan db:seed --force # optional

# Create storage link (for local usage). We'll also wire public_html below.
php artisan storage:link || true
```

### 2) Bootstrap `public\_html`

Copy the Laravel public files (.htaccess, favicon, build assets) and create a bootstrap index which boots the app in `../www`.

```
bash
# Copy web server rewrite rules and static files
cp -f ~/www/public/.htaccess ~/public_html/.htaccess
cp -f ~/www/public/favicon.ico ~/public_html/ 2>/dev/null || true
cp -f ~/www/public/robots.txt ~/public_html/ 2>/dev/null || true

# Copy built assets (created locally via `npm run build`)
rm -rf ~/public_html/build
cp -r ~/www/public/build ~/public_html/
```

Create `~/public\_html/index.php` with:

```
<?php
define('LARAVEL_START', microtime(true));

require __DIR__.'/../www/vendor/autoload.php';
$app = require __DIR__.'/../www/bootstrap/app.php';

// Make Laravel treat public_html as the public path
$app->bind('path.public', function() {
    return __DIR__;
});

$kernel = $app->make(Illuminate\Contracts\Http\Kernel::class);
$response = $kernel->handle($request = Illuminate\Http\Request::capture());
$response->send();
$kernel->terminate($request, $response);
```

### 3) Expose storage files in public\_html

Preferred (if symlinks allowed):

```
bash
cd ~/public_html
ln -s ../www/storage/app/public storage
```

If symlinks are blocked, mirror files (repeat on updates):

```
bash
mkdir -p ~/public_html/storage
rsync -a ~/www/storage/app/public/ ~/public_html/storage/
```

### 4) Permissions

```
bash
cd ~/www
mkdir -p storage/app/public/company-logos
find storage -type d -exec chmod 755 {} \;
find storage -type f -exec chmod 644 {} \;
chmod 755 bootstrap/cache
```

### 5) Optimize caches

```
bash
cd ~/www
php artisan config:clear && php artisan config:cache
php artisan route:cache
```

```
php artisan view:clear  
php artisan optimize
```

### ## Updating after a `git pull`

Run these from `~/www` after you pull new changes:

```
bash  
cd ~/www  
git pull  
composer install --no-dev --prefer-dist --optimize-autoloader  
php artisan migrate --force  
php artisan config:cache  
php artisan route:cache  
php artisan view:clear  
php artisan optimize
```

If assets changed, rebuild locally and upload `public/build` to `~/public\_html/build`, or copy from `~/www/public/build` if already built:

```
bash  
rm -rf ~/public_html/build  
cp -r ~/www/public/build ~/public_html/
```

If you mirror storage (no symlink), re-sync:

```
bash  
rsync -a ~/www/storage/app/public/ ~/public_html/storage/
```

### ## .env essentials

```
APP_ENV=production  
APP_DEBUG=false  
APP_URL=https://your-domain  
FILESYSTEM_DISK=public  
  
DB_CONNECTION=mysql  
DB_HOST=localhost  
DB_PORT=3306  
DB_DATABASE=your_db  
DB_USERNAME=your_user  
DB_PASSWORD=your_password
```

### ## Troubleshooting

- Logo not showing

- Ensure `~/public\_html/storage` points to or mirrors `~/www/storage/app/public`.
- Check permissions: folders 755, files 644.
- Verify `APP\_URL`, `FILESYSTEM\_DISK=public`, then `php artisan config:clear` & `php artisan config:cache`.
- 500 error after deploy
  - Run caches again (config/route/view) and `composer install --no-dev --prefer-dist`.
  - Confirm PHP version  $\geq 8.1$ .
- Routes not working
  - Make sure `~/public\_html/.htaccess` is copied from `~/www/public/.htaccess` (Laravel rewrites).

### ## Quick checklist (fresh server)

- 1) Clone repo to `~/www` and run Composer
- 2) Create `.env`, set DB + APP\_URL + FILESYSTEM\_DISK
- 3) `php artisan key:generate`
- 4) `php artisan migrate --force`
- 5) Bootstrap `public\_html` (copy .htaccess, index.php, and build assets)
- 6) Storage symlink (or mirror) into `public\_html/storage`
- 7) Permissions: `storage/` + `bootstrap/cache`
- 8) Cache optimize and test site

This setup keeps your local and server repo roots identical (`www`), and uses a small bootstrap in `public\_html` when you cannot change the Document Root.