



# Deploy Laravel su Netsons — Guida Completa



## 1. Pulizia del Progetto

Prepara l'ambiente di produzione eseguendo i seguenti comandi:

```
php artisan cache:clear
php artisan config:clear
php artisan route:clear
php artisan view:clear
npm run build
```



Il progetto è ora pronto per la distribuzione.

---



## 2. Compressione e Caricamento

- Comprimi l'intero progetto in un file .zip
  - Carica il file nella directory:  
home/utente/nome\_progetto
  - Estrai il contenuto nella stessa directory
- 



## 3. Organizzazione delle Cartelle

- Crea la directory:  
home/utente/public\_html/nome\_progetto
  - Sposta i file da:  
home/utente/nome\_progetto/public
- 👉 a 👉  
home/utente/public\_html/nome\_progetto

Nella cartella public originale, lascia solo:

```
build/
└─ manifest.json
```

---



## 4. Modifica del file [index.php](#)



Percorso:

home/utente/public\_html/nome\_progetto/[index.php](#)

Sostituisci il contenuto con:

```
<?php
use Illuminate\Foundation\Application;
use Illuminate\Http\Request;


define('LARAVEL_START', microtime(true));


if (file_exists($maintenance =
'/home/apghicha/nome_progetto/storage/framework/maintenance.php')) {
    require $maintenance;
}

require '/home/apghicha/nome_progetto/vendor/autoload.php';
```

```
$app = require_once '/home/apghicha/nome_progetto/bootstrap/app.php';

$app->handleRequest(Request::capture());
```

 Ricorda di sostituire **apghicha** e **nome\_progetto** con i tuoi dati reali.

 Aggiorna anche il file `.htaccess` in `public_html`:

```
RewriteCond %{REQUEST_URI} !^/nome_progetto(/|$)
```

---

## 5. Configurazione del Database

Modifica il file `.env` con i tuoi dati:

```
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=nome_database
DB_USERNAME=nome_utente
DB_PASSWORD=la_tua_password
```

---

## 6. Aggiornamento CSS/JS


Hai modificato solo CSS o JS?

Esegui:

```
npm run build
```


Poi carica i file da:

- `local/public/build/assets/` ➡ `public_html/nome_progetto/build/assets/`
- [manifest.json](#) ➡ `home/utente/nome_progetto/public/build/`


 In alternativa, puoi incollare direttamente le nuove righe CSS/JS nel file manager online.

---


## 7. Gestione Media senza Storage Link Laravel

 Crea la cartella:

```
resources/media/nome_cartella
```

 Configura config/[filesystems.php](#):

```
'images' => [
    'driver' => 'local',
    'root' => resource_path('media/nome_cartella'),
    'visibility' => 'public',
],
```

 Aggiungi in [web.php](#):

```
Route::get('/nome_cartella/{filename}', function ($filename) {
    $path = resource_path('media/destinations_images/' . $filename);

    if (!file_exists($path)) {
```

```
        abort(404);
    }

    return Response::file($path);
})->name('images');
```

✗ Commenta i symlinks in [filesystems.php](#):

```
'links' => [
    // public_path('storage') => storage_path('app/public'),
    // public_path('media') => storage_path('app/public/media'),
],
```

📷 Esempio in Blade:

```
destination }}">
```

📁 Assicurati che il nome della rotta corrisponda a quello definito.