

# Deployment Guide - Edens Gröna Garden Service

---

This guide provides detailed instructions for deploying the Laravel application to production.

## Pre-Deployment Checklist

---

### 1. Server Requirements

- ☐ PHP 8.2 or higher installed
- ☐ PostgreSQL 14+ installed and running
- ☐ Composer installed
- ☐ Web server (Apache/Nginx) configured
- ☐ SSL certificate for HTTPS
- ☐ Sufficient disk space (min 1GB free)
- ☐ Required PHP extensions enabled

### 2. Security Preparation

- ☐ Change default admin password
- ☐ Generate new APP\_KEY
- ☐ Set strong database password
- ☐ Configure firewall rules
- ☐ Enable HTTPS only
- ☐ Disable directory listing
- ☐ Set proper file permissions

## Deployment Methods

---

### Method 1: Manual Deployment via FTP/SFTP

#### Step 1: Upload Files

```
# On your local machine, create a zip
zip -r laravel_garden.zip laravel_garden_service/ -x "*/vendor/*" -x "*/node_modules/*" -x "*/.git/*"

# Upload via SFTP
sftp user@your-server.com
put laravel_garden.zip
```

## Step 2: Extract and Setup on Server

```
# SSH into server
ssh user@your-server.com

# Extract files
unzip laravel_garden.zip
cd laravel_garden_service

# Install dependencies
composer install --no-dev --optimize-autoloader

# Set permissions
chmod -R 755 .
chmod -R 775 storage bootstrap/cache
chown -R www-data:www-data storage bootstrap/cache
```

## Step 3: Configure Environment

```
# Copy and edit environment file
cp .env.example .env
nano .env
```

Update these values:

```
APP_ENV=production
APP_DEBUG=false
APP_URL=https://yourdomain.com

DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=your_database
DB_USERNAME=your_username
DB_PASSWORD=your_secure_password
```

## Step 4: Database Setup

```
# Generate application key
php artisan key:generate

# Run migrations
php artisan migrate --force

# Seed database
php artisan db:seed --force

# Create storage link
php artisan storage:link
```

## Step 5: Optimize Application

```
# Cache everything
php artisan config:cache
php artisan route:cache
php artisan view:cache
php artisan event:cache

# Clear old cache
php artisan cache:clear
```

## Method 2: Using Git

### Step 1: Initialize Git Repository

```
# On your local machine
cd laravel_garden_service
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/yourusername/your-repo.git
git push -u origin main
```

### Step 2: Deploy to Server

```
# SSH into server
ssh user@your-server.com

# Clone repository
cd /var/www
git clone https://github.com/yourusername/your-repo.git laravel_garden
cd laravel_garden

# Install dependencies
composer install --no-dev --optimize-autoloader

# Setup environment
cp .env.example .env
nano .env

# Generate key and migrate
php artisan key:generate
php artisan migrate --force
php artisan db:seed --force
php artisan storage:link

# Optimize
php artisan optimize

# Set permissions
chmod -R 755 .
chmod -R 775 storage bootstrap/cache
chown -R www-data:www-data storage bootstrap/cache
```

### Step 3: Setup Deployment Script

Create `deploy.sh` :

```
#!/bin/bash

echo "Pulling latest changes..."
git pull origin main

echo "Installing dependencies..."
composer install --no-dev --optimize-autoloader

echo "Running migrations..."
php artisan migrate --force

echo "Optimizing application..."
php artisan optimize
php artisan config:cache
php artisan route:cache
php artisan view:cache

echo "Setting permissions..."
chmod -R 775 storage bootstrap/cache

echo "Deployment complete!"
```

Make it executable:

```
chmod +x deploy.sh
```

Deploy updates:

```
./deploy.sh
```

### Method 3: Using Laravel Forge (Recommended)

1. **Sign up for Laravel Forge** (<https://forge.laravel.com>)
2. **Connect your server** (DigitalOcean, AWS, Linode, etc.)
3. **Create new site** with your domain
4. **Connect Git repository**
5. **Configure environment** via Forge dashboard
6. **Enable quick deploy** for automatic deployments
7. **Setup deployment script:**

```
cd /home/forge/yourdomain.com
git pull origin main
$FORGE_COMPOSER install --no-dev --no-interaction --prefer-dist --optimize-autoloader

( flock -w 10 9 || exit 1
  echo 'Restarting FPM...'; sudo -S service $FORGE_PHP_FPM reload ) 9>/tmp/fpmlock

if [ -f artisan ]; then
  $FORGE_PHP artisan migrate --force
  $FORGE_PHP artisan optimize
fi
```

# Web Server Configuration

## Apache Configuration

```
<VirtualHost *:80>
    ServerName edensgrona.se
    ServerAlias www.edensgrona.se
    Redirect permanent / https://edensgrona.se/
</VirtualHost>

<VirtualHost *:443>
    ServerName edensgrona.se
    ServerAlias www.edensgrona.se
    DocumentRoot /var/www/laravel_garden/public

    <Directory /var/www/laravel_garden/public>
        Options -Indexes +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    # SSL Configuration
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/your-cert.crt
    SSLCertificateKeyFile /etc/ssl/private/your-key.key
    SSLCertificateChainFile /etc/ssl/certs/your-chain.crt

    ErrorLog ${APACHE_LOG_DIR}/edensgrona-error.log
    CustomLog ${APACHE_LOG_DIR}/edensgrona-access.log combined
</VirtualHost>
```

Enable modules and restart:

```
sudo a2enmod rewrite ssl
sudo systemctl restart apache2
```

## Nginx Configuration

```
server {
    listen 80;
    listen [::]:80;
    server_name edensgrona.se www.edensgrona.se;
    return 301 https://edensgrona.se$request_uri;
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name edensgrona.se www.edensgrona.se;
    root /var/www/laravel_garden/public;

    index index.php;

    charset utf-8;

    # SSL Configuration
    ssl_certificate /etc/ssl/certs/your-cert.crt;
    ssl_certificate_key /etc/ssl/private/your-key.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    # Security Headers
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
    add_header X-XSS-Protection "1; mode=block";

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt { access_log off; log_not_found off; }

    error_page 404 /index.php;

    location ~ /\.php$ {
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        include fastcgi_params;
        fastcgi_hide_header X-Powered-By;
    }

    location ~ /\.(!well-known).* {
        deny all;
    }
}
```

Test and reload:

```
sudo nginx -t
sudo systemctl reload nginx
```

## SSL Certificate Setup

### Using Let's Encrypt (Free)

```
# Install Certbot
sudo apt update
sudo apt install certbot python3-certbot-apache # For Apache
# OR
sudo apt install certbot python3-certbot-nginx # For Nginx

# Obtain certificate
sudo certbot --apache -d edensgrona.se -d www.edensgrona.se
# OR
sudo certbot --nginx -d edensgrona.se -d www.edensgrona.se

# Auto-renewal (already configured by Certbot)
sudo certbot renew --dry-run
```

## Database Backup

### Automated Backup Script

Create `/home/backup/db-backup.sh` :

```
#!/bin/bash

# Configuration
DB_NAME="laravel_garden"
DB_USER="postgres"
BACKUP_DIR="/home/backup/database"
DATE=$(date +"%Y%m%d_%H%M%S")
BACKUP_FILE="$BACKUP_DIR/backup_$DATE.sql"

# Create backup directory if not exists
mkdir -p $BACKUP_DIR

# Perform backup
pg_dump -U $DB_USER $DB_NAME > $BACKUP_FILE

# Compress backup
gzip $BACKUP_FILE

# Delete backups older than 30 days
find $BACKUP_DIR -name "backup_*.sql.gz" -mtime +30 -delete

echo "Backup completed: $BACKUP_FILE.gz"
```

Make executable:

```
chmod +x /home/backup/db-backup.sh
```

## Setup Cron Job

```
crontab -e

# Add this line for daily backup at 2 AM
0 2 * * * /home/backup/db-backup.sh
```

## Monitoring and Maintenance

### Setup Log Rotation

Create `/etc/logrotate.d/laravel` :

```
/var/www/laravel_garden/storage/logs/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 0640 www-data www-data
    sharedscripts
}
```

### Health Check Script

Create `/home/scripts/health-check.sh` :

```
#!/bin/bash

SITE_URL="https://edensgrona.se"
STATUS=$(curl -s -o /dev/null -w "%{http_code}" $SITE_URL)

if [ $STATUS -eq 200 ]; then
    echo "Site is up (HTTP $STATUS)"
else
    echo "Site is down (HTTP $STATUS)"
    # Send alert email
    echo "Site is down!" | mail -s "Site Down Alert" admin@edensgrona.se
fi
```

## Performance Optimization

```
# Enable OPcache
sudo nano /etc/php/8.2/fpm/php.ini

# Add/uncomment these lines:
opcache.enable=1
opcache.memory_consumption=256
opcache.max_accelerated_files=20000
opcache.validate_timestamps=0

# Restart PHP-FPM
sudo systemctl restart php8.2-fpm
```



## Troubleshooting Production Issues

---

### Check Application Logs

```
tail -f storage/logs/laravel.log
```

### Check Web Server Logs

```
# Apache
tail -f /var/log/apache2/error.log

# Nginx
tail -f /var/log/nginx/error.log
```

### Check PHP-FPM Logs

```
tail -f /var/log/php8.2-fpm.log
```

### Clear All Caches

```
php artisan cache:clear
php artisan config:clear
php artisan route:clear
php artisan view:clear
composer dump-autoload
```

### Fix Permissions

```
cd /var/www/laravel_garden
sudo chown -R www-data:www-data .
sudo chmod -R 755 .
sudo chmod -R 775 storage bootstrap/cache
```

## Rollback Procedure

---

If something goes wrong:

```
# 1. Revert to previous Git commit
git log --oneline # Find previous commit hash
git reset --hard <commit-hash>

# 2. Restore database backup
gunzip /home/backup/database/backup_YYYYMMDD_HHMMSS.sql.gz
psql -U postgres laravel_garden < /home/backup/database/backup_YYYYMMDD_HHMMSS.sql

# 3. Clear caches and optimize
php artisan optimize:clear
php artisan optimize

# 4. Restart services
sudo systemctl restart php8.2-fpm
sudo systemctl restart apache2 # or nginx
```

## Post-Deployment Checklist

---

- ☐ Site loads correctly (check HTTPS)
- ☐ Admin panel accessible at /admin
- ☐ All pages working (Home, About, Contact)
- ☐ Forms submitting correctly
- ☐ Images displaying properly
- ☐ Social links working
- ☐ Mobile responsiveness verified
- ☐ SSL certificate valid
- ☐ Backup script running
- ☐ Health monitoring active
- ☐ Email notifications working

## Support

---

For deployment support:

- Email: [info@edensgrona.se](mailto:info@edensgrona.se)
- Phone: 076-049 28 28

---

**Last Updated:** January 2026

**Version:** 1.0.0