Production Database Setup Guide

Generated: November 19, 2025
Version: v1.0
Environment: Production

1. Overview

This guide provides step-by-step instructions for setting up the HRM Laravel Base application i production environment using PostgreSQL or MySQL database.

Prerequisites

- PostgreSQL 12+ or MySQL 8.0+
- PHP 8.4+ with required extensions
- Web server (Nginx/Apache) with SSL
- Sufficient disk space and memory
- Database backup strategy

2. Database Setup

2.1 PostgreSQL Setup (Recommended)

Database Creation

```
-- Create database
CREATE DATABASE hrm_production
    WITH ENCODING='UTF8'
    LC_COLLATE='en_US.UTF-8'
    LC_CTYPE='en_US.UTF-8'
    TEMPLATE=template0;


-- Create user
CREATE USER hrm_user WITH PASSWORD 'your_secure_password';


-- Grant privileges
GRANT ALL PRIVILEGES ON DATABASE hrm_production TO hrm_user;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO hrm_user;


-- Connect to database
\c hrm_production;


-- Enable required extensions
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
CREATE EXTENSION IF NOT EXISTS "pg_trgm";
```

Configuration Optimization

```
-- Memory settings for production
ALTER SYSTEM SET shared_buffers = '256MB';
ALTER SYSTEM SET effective_cache_size = '1GB';
ALTER SYSTEM SET maintenance_work_mem = '64MB';
ALTER SYSTEM SET checkpoint_completion_target = 0.9;


-- WAL settings
ALTER SYSTEM SET wal_buffers = '16MB';
ALTER SYSTEM SET wal_writer_delay = '200ms';
ALTER SYSTEM SET commit_delay = '1000ms';


-- Query optimization
ALTER SYSTEM SET random_page_cost = 1.1;
ALTER SYSTEM SET effective_io_concurrency = 200;
```

2.2 MySQL Alternative Setup

Database Creation

```
-- Create database
CREATE DATABASE hrm_production
    CHARACTER SET utf8mb4
    COLLATE utf8mb4_unicode_ci;


-- Create user
CREATE USER 'hrm_user'@'%' IDENTIFIED BY 'your_secure_password';


-- Grant privileges
GRANT ALL PRIVILEGES ON hrm_production. TO 'hrm_user'@'%';
FLUSH PRIVILEGES;
```

Configuration Optimization

```
# Memory settings
innodb_buffer_pool_size = 1G
innodb_log_file_size = 256M
innodb_flush_log_at_trx_commit = 2

## Query cache

query_cache_size = 64M
query_cache_type = 1

## Connection settings

max_connections = 200
max_connect_errors = 10000
```

---

3. Application Configuration

3.1 Environment Variables

Update your `.env` *file for production:*

```
# Database Configuration
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=hrm_production
DB_USERNAME=hrm_user
DB_PASSWORD=your_secure_password
```

For MySQL use:

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

Application Configuration

```
APP_ENV=production
APP_DEBUG=false
APP_URL=https://your-domain.com
APP_KEY=base64:your-32-character-key
APP_CIPHER=AES-256-CBC
```

Cache Configuration

```
CACHE_DRIVER=redis
CACHE_PREFIX=hrm_prod
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=your-redis-password
```

Session Configuration

```
SESSION_DRIVER=redis
SESSION_LIFETIME=120
SESSION_ENCRYPT=true
SESSION_PATH=hrm_prod_sessions
```

Queue Configuration

QUEUE_CONNECTION=redis
QUEUE_FAILED_DRIVER=database

Mail Configuration

MAIL_MAILER=smtp
MAIL_HOST=your-smtp-host
MAIL_PORT=587
MAIL_USERNAME=your-email@domain.com
MAIL_PASSWORD=your-email-password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=noreply@your-domain.com

Filesystem Configuration

FILESYSTEM_DISK=local

Logging Configuration

LOG_CHANNEL=stack
LOG_LEVEL=warning
LOG_DEPRECATIONS_CHANNEL=null

3.2 Composer Dependencies

Ensure production dependencies are installed:

```
{
    "require": {
        "php": "^8.4",
        "laravel/framework": "^12.0",
        "ext-pgsql": "",
        "ext-pdo": "",
        "ext-bcmath": "",
        "ext-ctype": "",
        "ext-curl": "",
        "ext-dom": "",
        "ext-fileinfo": "",
        "ext-filter": "",
        "ext-gd": "",
        "ext-hash": "",
        "ext-iconv": "",
        "ext-json": "",
        "ext-libxml": "",
        "ext-mbstring": "",
        "ext-openssl": "",
        "ext-pcre": "",
        "ext-pdo": "",
        "ext-session": "",
        "ext-simplexml": "",
        "ext-spl": "",
        "ext-tokenizer": "",
        "ext-xml": ""
    }
}
```

---

4. Web Server Configuration

4.1 Nginx Configuration

```
server {
    listen 443 ssl http2;
    server_name your-domain.com;
    root /var/www/hrm-laravel-base/public;
    index index.php;

    # SSL Configuration
    ssl_certificate /etc/ssl/certs/your-domain.com.crt;
    ssl_certificate_key /etc/ssl/private/your-domain.com.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256;
    ssl_prefer_server_ciphers off;

    # Security Headers
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
    add_header X-XSS-Protection "1; mode=block";
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains";

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

    # Static Files
    location ~ \.(js|css|png|jpg|jpeg|gif|ico|svg)$ {
        expires 1y;
        add_header Cache-Control "public, immutable";
    }

    # PHP Processing
    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php8.4-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        include fastcgi_params;
    }

    # Gzip Compression
    gzip on;
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_types text/plain text/css application/json application/javascript text/xml application/xml text/xml applicati
}
```

4.2 Apache Alternative

```
:443>
    ServerName your-domain.com
    DocumentRoot /var/www/hrm-laravel-base/public
    DirectoryIndex index.php


    # SSL Configuration
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/your-domain.com.crt
    SSLCertificateKeyFile /etc/ssl/private/your-domain.com.key


    # Security Headers
    Header always set X-Frame-Options "SAMEORIGIN"
    Header always set X-Content-Type-Options "nosniff"
    Header always set X-XSS-Protection "1; mode=block"
    Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"


    # Laravel Configuration

        AllowOverride All
        Require all granted
        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule ^(.)$ index.php [QSA,L]
```

---

5. Deployment Steps

5.1 Code Deployment

```
# 1. Pull latest code
git pull origin main

## 2. Install dependencies

composer install --no-dev --optimize-autoloader

## 3. Clear caches

php artisan cache:clear
php artisan config:clear
php artisan route:clear
php artisan view:clear

## 4. Optimize for production

php artisan config:cache
php artisan route:cache
php artisan view:cache

## 5. Set permissions

chown -R www-data:www-data /var/www/hrm-laravel-base
chmod -R 755 /var/www/hrm-laravel-base/storage
chmod -R 755 /var/www/hrm-laravel-base/bootstrap/cache
```

5.2 Database Migration

```
# 1. Test database connection
php artisan tinker
>>> DB::connection()->getPdo();
```

2. Run migrations

```
php artisan migrate --force
```

3. Check migration status

```
php artisan migrate:status
```

4. Seed essential data

```
php artisan db:seed --class=ChartOfAccountsSeeder --force
php artisan db:seed --class=OrganizationSeeder --force
```

5.3 Queue Setup

```
# 1. Install Redis
sudo apt-get install redis-server
```

2. Configure Redis

```
sudo systemctl enable redis-server
sudo systemctl start redis-server
```

3. Test queue connection

```
php artisan queue:failed
php artisan queue:monitor
```

---

6. Performance Optimization

6.1 PHP Configuration

```
# Production php.ini settings
memory_limit = 256M
max_execution_time = 300
max_input_vars = 3000
upload_max_filesize = 20M
post_max_size = 20M
max_file_uploads = 20

## OPcache settings

opcache.enable=1
opcache.memory_consumption=128
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=4000
opcache.revalidate_freq=60
opcache.validate_timestamps=1
opcache.save_comments=1
opcache.enable_file_override=1
```

6.2 Laravel Optimization

```
# Optimize autoloader
composer dump-autoload --optimize

## Cache configuration for production

php artisan optimize

## Precompile views

php artisan view:cache
```

---

7. Security Configuration

7.1 File Permissions

```
# Secure sensitive files
chmod 600 .env
chmod 600 config/app.php
```

Secure storage and cache

```
chmod -R 755 storage/
chmod -R 755 bootstrap/cache/
```

Set proper ownership

```
chown -R www-data:www-data storage/
chown -R www-data:www-data bootstrap/cache/
```

7.2 Firewall Configuration

```
# Allow only necessary ports
sudo ufw allow 22/tcp    # SSH
sudo ufw allow 80/tcp    # HTTP
sudo ufw allow 443/tcp   # HTTPS
sudo ufw deny 3306/tcp  # MySQL (if not needed externally)
sudo ufw deny 5432/tcp  # PostgreSQL (if not needed externally)
sudo ufw enable
```

7.3 SSL Certificate

```
# Let's Encrypt certificate
sudo certbot --nginx -d your-domain.com
```

Auto-renewal

```
sudo crontab -e
"0 12   /usr/bin/certbot renew --quiet && systemctl reload nginx"
```

---

8. Monitoring & Logging

8.1 Application Monitoring

```
# Enable Laravel Telescope (optional)
composer require laravel/telescope
php artisan vendor:publish --provider="Laravel\Telescope\TelescopeServiceProvider"

# Configure logging

LOG_CHANNEL=daily
LOG_LEVEL=warning
```

8.2 Database Monitoring

```
# PostgreSQL monitoring
sudo -u postgres psql -c "ALTER SYSTEM SET log_min_duration_statement = 1000;"
sudo -u postgres psql -c "ALTER SYSTEM SET log_statement = 'all';"

# MySQL slow query log

# Add to my.cnf:

slow_query_log = /var/log/mysql/slow.log
long_query_time = 2
```

8.3 Server Monitoring

```
# Install monitoring tools
sudo apt-get install htop iotop nethogs

# Monitor system resources

htop
iotop
nethogs
```

9. Backup Strategy

9.1 Database Backups

```
#!/bin/bash

# PostgreSQL backup script

BACKUP_DIR="/var/backups/hrm"
DATE=$(date +%Y%m%d_%H%M%S)

# Create backup directory

mkdir -p $BACKUP_DIR

# Full backup

pg_dump -h localhost -U hrm_user -d hrm_production \
    --format=custom \
    --file=$BACKUP_DIR/hrm_full_$DATE.sql

# Compressed backup

pg_dump -h localhost -U hrm_user -d hrm_production \
    --format=custom \
    --compress=9 \
    --file=$BACKUP_DIR/hrm_compressed_$DATE.sqlc

# Clean old backups (keep 7 days)

find $BACKUP_DIR -name "*.sql" -mtime +7 -delete
```

9.2 Application Backups

```
#!/bin/bash

## Application backup script

APP_DIR="/var/www/hrm-laravel-base"
BACKUP_DIR="/var/backups/hrm/app"
DATE=$(date +%Y%m%d_%H%M%S)

## Create backup

tar -czf $BACKUP_DIR/hrm_app_$DATE.tar.gz \
    --exclude='node_modules' \
    --exclude='.git' \
    --exclude='storage/logs' \
    --exclude='storage/framework/cache' \
    $APP_DIR
```

9.3 Automated Backups

```
# Add to crontab
crontab -e

## Database backup at 2 AM daily

0 2   /path/to/backup_script.sh

## Application backup at 3 AM daily

0 3   /path/to/app_backup_script.sh

## Weekly full backup on Sunday at 1 AM

0 1   0 /path/to/full_backup_script.sh
```

---

10. Testing & Validation

10.1 Pre-Deployment Checklist

```
# Database connection test
php artisan db:show

## Environment check

php artisan env:current

## Route cache test

php artisan route:list

## Configuration check

php artisan config:show
```

10.2 Post-Deployment Validation

```
# Test critical functionality
curl -k https://your-domain.com/login
curl -k https://your-domain.com/api/health

## Check application logs

tail -f storage/logs/laravel.log

## Monitor error rates

grep -i "error\|exception" storage/logs/laravel.log | wc -l
```

---

11. Troubleshooting

11.1 Common Issues

Database Connection Issues

- Check database server status
- Verify credentials in `.env`
- Test network connectivity
- Check firewall rules

Permission Issues

- Verify file ownership
- Check directory permissions
- Ensure SELinux context

Performance Issues

- Monitor memory usage
- Check slow queries
- Verify cache configuration

11.2 Debug Commands

```
# Clear all caches
php artisan cache:clear && php artisan config:clear && php artisan route:clear && php artisan view:clear

# Check migration status
php artisan migrate:status

# Test database connection
php artisan tinker
>>> DB::connection()->getDatabaseName()
```

12. Maintenance Procedures

12.1 Zero-Downtime Deployment

```
# 1. Deploy to new directory
cp -r /var/www/hrm-laravel-base /var/www/hrm-laravel-base.new

## 2. Update symlinks

ln -sfn /var/www/hrm-laravel-base.new /var/www/hrm-laravel-base

## 3. Run migrations

php artisan migrate --force

## 4. Clear caches

php artisan cache:clear

## 5. Test application

curl -k https://your-domain.com/health

## 6. Update symlink to new version

rm /var/www/hrm-laravel-base
ln -s /var/www/hrm-laravel-base.new /var/www/hrm-laravel-base
```

12.2 Maintenance Mode

```
# Enable maintenance mode
php artisan down --message="System maintenance in progress"

## Deploy updates

## ... deployment steps ...

## Disable maintenance mode

php artisan up
```

---

13. Production Checklist

13.1 Pre-Launch Checklist

- [ ] Database server configured and optimized
- [ ] Application dependencies installed
- [ ] Environment variables configured
- [ ] SSL certificate installed and valid
- [ ] Web server configured and optimized
- [ ] File permissions set correctly
- [ ] Firewall rules configured
- [ ] Backup strategy implemented
- [ ] Monitoring tools configured
- [ ] Application tested thoroughly
- [ ] Performance benchmarks completed
- [ ] Security audit completed

13.2 Post-Launch Checklist

- [ ] Application responding to HTTPS requests
- [ ] Database connections working properly
- [ ] Caches functioning correctly
- [ ] Queues processing jobs
- [ ] Scheduled tasks running
- [ ] Backups executing successfully
- [ ] Monitoring tools collecting data
- [ ] Error rates within acceptable limits
- [ ] Performance metrics meeting targets

---

14. Support & Recovery

14.1 Emergency Procedures

```
# Quick rollback to previous version
git checkout previous-stable-tag
composer install --no-dev
php artisan migrate --force
php artisan cache:clear
```

Emergency database restore

```
pg_restore -h localhost -U hrm_user -d hrm_production \
    --clean --if-exists \
    /var/backups/hrm/hrm_emergency.sql
```

14.2 Contact Information

- **Primary Support**: support@your-domain.com
- **Emergency Contact**: +1-555-XXX-XXXX
- **Documentation**: https://docs.your-domain.com
- **Status Page**: https://status.your-domain.com

This production setup guide provides comprehensive instructions for deploying the HRM Laravel B application in a production environment with proper security, performance, and monito considerations.