

# Heroku Deployment Guide

---

## Prerequisites

1. Install Heroku CLI: `brew install heroku/brew/heroku`
2. Login to Heroku: `heroku login`

## Deployment Steps

### 1. Create Heroku App

```
heroku create your-travel-planner-app
```

### 2. Add PostgreSQL Addon

```
heroku addons:create heroku-postgresql:essential-0
```

### 3. Set Environment Variables

```
# For production configuration
heroku config:set RAILS_ENV=production
heroku config:set RAILS_SERVE_STATIC_FILES=true
heroku config:set RAILS_LOG_TO_STDOUT=true

# Add API keys when ready
heroku config:set OPENAI_API_KEY=your_openai_key
heroku config:set GOOGLE_MAPS_API_KEY=your_google_maps_key
heroku config:set TRIPADVISOR_API_KEY=your_tripadvisor_key
```

### 4. Deploy

```
git add .
git commit -m "Initial deployment setup"
git push heroku main
```

### 5. Run Database Migrations

```
heroku run rails db:migrate
```

## 6. (Optional) Seed Database

```
heroku run rails db:seed
```

## Database Configuration

Your app is already configured to:

- Use SQLite in development/test (easier local setup)
- Use PostgreSQL in production (Heroku compatible)
- Read DATABASE\_URL environment variable automatically in production

## Local PostgreSQL Testing

If you want to test with PostgreSQL locally:

```
# Switch to PostgreSQL config
ruby bin/setup_database postgresql
cp config/database.yml.pg config/database.yml
rails db:create db:migrate

# Switch back to SQLite
cp config/database.yml.backup config/database.yml
```

## Notes

- The `pg` gem is only loaded in production environment
- SQLite gem is only loaded in development/test environments
- This setup provides the best of both worlds: easy development + production readiness