

UrbanFleet Deployment Guide

Complete Guide to Deploy Your Website

1. Prerequisites

Software Requirements

Software	Version	Purpose
Node.js	18.x or higher	JavaScript runtime
npm	9.x or higher	Package manager
PostgreSQL	14.x or higher	Database
Git	Latest	Version control

Server Requirements

- Minimum 1GB RAM
- 10GB storage space
- Linux-based OS (Ubuntu 20.04+ recommended)
- SSL certificate for HTTPS

Required Environment Variables

Variable	Description	Example
DATABASE_URL	PostgreSQL connection string	postgresql://user:pass@host:5432/dbname
NODE_ENV	Environment mode	production
PORT	Server port (optional)	5000

2. Project Structure

```
urbanfleet/
├── client/          # Frontend React application
|   ├── src/
|   |   ├── pages/    # Page components
|   |   └── components/
|   └── index.html
├── server/          # Backend Express server
|   ├── index.ts
|   ├── routes.ts
|   └── storage.ts
├── shared/           # Shared code (schema)
|   └── schema.ts
└── dist/             # Production build (after build)
    ├── public/        # Frontend static files
    └── index.cjs      # Server bundle
└── package.json
```

3. Option A: Deploy on Replit (Easiest)

Step 1: Click the "Publish" button in Replit (top right corner)

Step 2: Configure your deployment settings

Step 3: Your site will be live at a .replit.app domain

Step 4: (Optional) Add a custom domain in deployment settings

Benefits: Automatic SSL, built-in database, zero configuration, automatic updates

4. Option B: Deploy to VPS (DigitalOcean, AWS, etc.)

Step 1: Prepare Your Server

```
# Update system  
sudo apt update && sudo apt upgrade -y  
  
# Install Node.js 20  
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -  
sudo apt install -y nodejs  
  
# Install PostgreSQL  
sudo apt install -y postgresql postgresql-contrib  
  
# Verify installations  
node --version  
npm --version  
psql --version
```

Step 2: Setup PostgreSQL Database

```
# Access PostgreSQL  
sudo -u postgres psql  
  
# Create database and user  
CREATE DATABASE urbanfleet;  
CREATE USER urbanfleet_user WITH ENCRYPTED PASSWORD 'your_password';  
GRANT ALL PRIVILEGES ON DATABASE urbanfleet TO urbanfleet_user;  
\q
```

Step 3: Clone and Setup Project

```
# Clone from GitHub  
git clone https://github.com/YOUR_USERNAME/urbanfleet.git  
cd urbanfleet  
  
# Install dependencies  
npm install  
  
# Create environment file  
nano .env
```

Step 4: Configure Environment Variables

```
# Add to .env file:  
DATABASE_URL=postgresql://urbanfleet_user:your_password@localhost:5432/urbanfleet  
NODE_ENV=production  
PORT=5000
```

Step 5: Build and Deploy

```
# Build the application  
npm run build  
  
# Push database schema  
npm run db:push  
  
# Start the server  
node dist/index.cjs
```

5. Option C: Deploy with PM2 (Production)

Step 1: Install PM2

```
sudo npm install -g pm2
```

Step 2: Start Application with PM2

```
# Start the application  
pm2 start dist/index.cjs --name "urbanfleet"  
  
# Save PM2 configuration  
pm2 save  
  
# Setup PM2 to start on boot  
pm2 startup
```

Step 3: Useful PM2 Commands

```
pm2 status          # Check status  
pm2 logs urbanfleet # View logs  
pm2 restart urbanfleet # Restart app  
pm2 stop urbanfleet    # Stop app
```

6. Setup Nginx Reverse Proxy

Step 1: Install Nginx

```
sudo apt install -y nginx
```

Step 2: Configure Nginx

```
sudo nano /etc/nginx/sites-available/urbanfleet
```

Add Configuration:

```
server {  
    listen 80;  
    server_name yourdomain.com www.yourdomain.com;  
  
    location / {  
        proxy_pass http://localhost:5000;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
    }  
}
```

Step 3: Enable Site

```
sudo ln -s /etc/nginx/sites-available/urbanfleet /etc/nginx/sites-enabled/  
sudo nginx -t  
sudo systemctl restart nginx
```

7. Setup SSL Certificate (HTTPS)

```
# Install Certbot
sudo apt install -y certbot python3-certbot-nginx

# Get SSL certificate
sudo certbot --nginx -d yourdomain.com -d www.yourdomain.com

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

8. Deployment Checklist

Task	Status
Server provisioned with required specs	[]
Node.js 18+ installed	[]
PostgreSQL installed and configured	[]
Database created and user configured	[]
Project cloned from repository	[]
Dependencies installed (npm install)	[]
Environment variables configured	[]
Application built (npm run build)	[]
Database schema pushed (npm run db:push)	[]
Application started with PM2	[]
Nginx configured as reverse proxy	[]
SSL certificate installed	[]
Domain DNS configured	[]
Application tested and verified	[]

9. Troubleshooting

Common Issues

Database Connection Error:

Verify DATABASE_URL is correct and PostgreSQL is running:

```
sudo systemctl status postgresql
```

Port Already in Use:

Check what's using port 5000:

```
sudo lsof -i :5000
```

Permission Denied:

Ensure proper file permissions:

```
sudo chown -R $USER:$USER /path/to/urbanfleet
```

10. Support & Contact

For technical support or questions about deployment:

- Email: support@urbanfleet.ae
- Website: www.urbanfleet.ae

Document Version: 1.0

Last Updated: January 2026