TalentSync Deployment Guide

This guide provides step-by-Step instructions for deploying TalentSync to various hosting platforms.

Table of Contents

Free Hosting Options

Best Free Options (2025):

1. **Railway** - Most generous free tier, great performance
2. **Render** - Reliable, good for Node.js apps
3. **Heroku** - Limited free tier but stable
4. **Fly.io** - Good performance, Docker-based

Railway Deployment (Recommended)

Railway offers the best free tier for Node.js applications with automatic deployments.

Step 1: Prepare Your Code

1. **Create a GitHub repository:**

```bash
git init
git add .
git commit -m "Initial commit"
git remote add origin https://github.com/yourusername/talentsync.git
git push -u origin main
```

2. **Add a `railway.json` file:**

```json
{
  "$schema": "https://railway.app/railway.schema.json",
  "build": {
    "builder": "NIXPACKS"
  },
  "deploy": {
    "numReplicas": 1,
    "sleepApplication": false,
    "restartPolicyType": "ON_FAILURE"
  }
}
```

Step 2: Deploy to Railway

1. **Visit** railway.app
2. **Sign up** with GitHub
3. **Click "New Project"**
4. **Select "Deploy from GitHub repo"**
5. **Choose your TalentSync repository**

6. **Railway will automatically detect it's a Node.js app**

Step 3: Configure Environment Variables

1. **In Railway dashboard, go to Variables tab**
2. **Add these variables:**

```
JWT_SECRET=your-super-secret-jwt-key-change-this
PORT=3000
NODE_ENV=production
```

Step 4: Deploy

1. **Railway will automatically deploy**
2. **Get your app URL from the dashboard**
3. **Visit your live site!**

Railway Pros:

- ✅ $5 free credit monthly
- ✅ Automatic deployments from GitHub
- ✅ Custom domains supported
- ✅ Good performance
- ✅ Easy to use

Render Deployment

Render is another excellent free option with good Node.js support.

Step 1: Prepare Repository

Same as Railway - push your code to GitHub.

Step 2: Deploy to Render

1. **Visit** render.com
2. **Sign up with GitHub**
3. **Click "New +" → "Web Service"**
4. **Connect your GitHub repository**
5. **Configure the service:**
   - **Name:** talentsync
   - **Environment:** Node
   - **Build Command:** `npm install`
   - **Start Command:** `npm start`

Step 3: Environment Variables

Add in Render dashboard:

```
JWT_SECRET=your-super-secret-jwt-key
NODE_ENV=production
```

Step 4: Deploy

Render will build and deploy automatically.

Render Pros:

- ✅ Free tier available
- ✅ Automatic SSL certificates
- ✅ Custom domains

- ✅ Good documentation

Render Cons:

- ❌ Apps sleep after 15 minutes of inactivity
- ❌ Limited free tier hours

Heroku Deployment

Heroku has limited free options but is very reliable.

Step 1: Install Heroku CLI

```bash
# macOS
brew tap heroku/brew && brew install heroku

# Windows
# Download from heroku.com/cli

# Ubuntu/Debian
curl https://cli-assets.heroku.com/install.sh | sh
```

Step 2: Create Heroku App

```bash
heroku login
heroku create your-talentsync-app
```

Step 3: Configure Environment

```bash
heroku config:set JWT_SECRET=your-super-secret-jwt-key
heroku config:set NODE_ENV=production
```

Step 4: Deploy

```bash
git push heroku main
```

Heroku Pros:

- ✅ Very reliable
- ✅ Great documentation
- ✅ Add-ons ecosystem

Heroku Cons:

- ❌ Very limited free tier
- ❌ Apps sleep after 30 minutes
- ❌ Expensive paid plans

VPS Deployment (More Control)

For more control and better performance, use a VPS.

Recommended VPS Providers:

- **DigitalOcean:** $4-6/month droplets
- **Linode:** $5/month nanode

- **Vultr:** $2.50-6/month instances
- **Hetzner:** €3-5/month (Europe)

Step 1: Set Up Server (Ubuntu 22.04)

```bash
# Update system
sudo apt update && sudo apt upgrade -y

# Install Node.js 18
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs

# Install other dependencies
sudo apt install -y nginx git ufw
```

Step 2: Clone Your Project

```bash
cd /var/www
sudo git clone https://github.com/yourusername/talentsync.git
cd talentsync
sudo chown -R $USER:$USER /var/www/talentsync
npm install
```

Step 3: Configure Environment

```bash
```

```
cp .env.example .env
nano .env
```

Add your production settings:

```
env

PORT=3000
JWT_SECRET=your-super-secret-production-key
NODE_ENV=production
```

Step 4: Install PM2 (Process Manager)

```
bash

sudo npm install -g pm2
pm2 start server.js --name talentsync
pm2 startup
pm2 save
```

Step 5: Configure Nginx

```
bash

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

Add this configuration:

```
nginx
```

```
server {
    listen 80;
    server_name your-domain.com www.your-domain.com;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Enable the site:

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

Step 6: Set Up Firewall

```bash
sudo ufw allow OpenSSH
sudo ufw allow 'Nginx Full'
sudo ufw enable
```

Step 7: SSL Certificate (Optional but Recommended)

```bash
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d your-domain.com -d www.your-domain.com
```

Domain Setup

Free Domain Options:

- **Freenom:** .tk, .ml, .ga domains
- **DuckDNS:** Free subdomain service
- **No-IP:** Free DNS service

Paid Domain Options:

- **Namecheap:** $8-12/year
- **Google Domains:** $12/year
- **Cloudflare:** $8-10/year

Setting Up Domain:

1. **Point domain to your server:**
   - **For VPS:** Add A record pointing to your server IP
   - **For Platform deployments:** Add CNAME record pointing to platform URL
2. **Example DNS settings:**

```
Type: A
Name: @
Value: 143.198.123.456 (your server IP)

Type: A
Name: www
Value: 143.198.123.456
```

Data Management

Important Notes:

1. **Backup Strategy:**

```bash
# Create backup script
#!/bin/bash
DATE=$(date +%Y%m%d_%H%M%S)
tar -czf "backup_$DATE.tar.gz" data/ uploads/
```

2. **Data Persistence:**
   - Platform deployments may lose data on restarts
   - Consider upgrading to database for production

3. **File Storage:**
   - For production, consider cloud storage:
   - AWS S3, Google Cloud Storage, or Cloudinary

Database Upgrade (Recommended for Production):

1. **Add MongoDB Atlas (Free tier):**

```bash
npm install mongoose
```

2. **Or PostgreSQL:**

```bash
npm install pg
```

Monitoring and Maintenance

1. Application Monitoring:

```bash
# Check app status
pm2 status
pm2 logs talentsync

# Restart if needed
pm2 restart talentsync
```

2. Server Monitoring:

```bash
```

```
# Check disk space
df -h

# Check memory usage
free -h

# Check server uptime
uptime
```

3. Updates:

```
bash

# Update application
cd /var/www/talentsync
git pull origin main
npm install
pm2 restart talentsync
```

Troubleshooting

Common Issues:

1. **App won't start:**

```
bash

# Check logs
pm2 logs talentsync

# Check if port is available
sudo netstat -tlnp | grep :3000
```

2. **File upload issues:**

```bash
# Check permissions
ls -la uploads/
chmod 755 uploads/resumes/
```

3. **Database connection issues:**

```bash
# Check if data directory exists
ls -la data/
mkdir -p data uploads/resumes
```

4. **Domain not working:**
   - Check DNS propagation: whatsmydns.net
   - Verify Nginx configuration: `sudo nginx -t`
   - Check SSL certificate: `sudo certbot certificates`

Cost Breakdown

Free Hosting:

- **Railway:** $0 (with $5 monthly credit)
- **Render:** $0 (with limitations)
- **Domain:** $0-12/year

VPS Hosting:

- **DigitalOcean Droplet:** $4-6/month
- **Domain:** $8-12/year

- **Total:** ~$60-80/year

Recommended Setup for Beginners:

1. Start with **Railway** (free)
2. Use free domain or buy cheap one
3. Upgrade to VPS when you need more control

Recommended Setup for Production:

1. **DigitalOcean/Linode VPS** ($5/month)
2. **Custom domain** ($10/year)
3. **MongoDB Atlas** (free tier)
4. **Cloudflare** for CDN/security (free)

Final Checklist

Before going live:

- [ ] Change JWT_SECRET to secure random string
- [ ] Set NODE_ENV=production
- [ ] Test all functionality (register, login, job posting, applications)
- [ ] Set up SSL certificate
- [ ] Configure backups
- [ ] Test file uploads
- [ ] Verify email works (if implemented)
- [ ] Check mobile responsiveness
- [ ] Set up monitoring/alerts

Support

If you encounter issues:

1. **Check server logs** first
2. **Review this guide** for common solutions
3. **Check platform documentation**:
   - Railway: railway.app/docs
   - Render: render.com/docs
   - DigitalOcean: docs.digitalocean.com

Remember: Start simple with a free platform, then upgrade as your needs grow!