⚡ Railway Quick Fix - TL;DR

🔴 The Problem

Your deployment was failing with this error:

```
ERROR: "/nginx.conf": not found
```

Root Cause: Railway was using the wrong Dockerfile (frontend instead of backend).

✅ The Solution (3 Steps)

1️⃣ Set Root Directory

In Railway Dashboard:
1. Go to your backend service → **Settings**
2. Find **“Root Directory”**
3. Set to: `backend`
4. Save

2️⃣ Add Environment Variables

In Railway Dashboard → **Variables**:

```
DATABASE_URL=postgresql://user:password@host:port/database
JWT_SECRET=your-secret-here
```

Generate JWT_SECRET:

```
openssl rand -base64 32
```

3️⃣ Deploy

Railway will auto-deploy after setting root directory.

🎯 Verify Success

Check: `https://your-domain/api/health`

Should return:

```
{"status": "OK", "database": "connected"}
```

📋 Configuration Files

✅ `railway.toml` - Already Updated

Located: `/backend/railway.toml`

Key settings:
- Builder: Dockerfile
- Start command: `/app/start.sh`
- Health check: `/api/health`

✅ `Dockerfile` - Already Correct

Located: `/backend/Dockerfile`

Multi-stage build:
1. Build TypeScript → `/app/dist`
2. Production image with Node.js
3. Runs migrations + starts server

🚨 Common Mistakes to Avoid

❌ **Don't:** Deploy without setting root directory to `backend`
✅ **Do:** Set root directory to `backend` first

❌ **Don't:** Forget to set `DATABASE_URL`
✅ **Do:** Configure all required environment variables

❌ **Don't:** Use weak JWT secrets
✅ **Do:** Generate strong, random secrets

🔍 Quick Troubleshooting

Error	Solution
"nginx.conf not found"	Set root directory to `backend`
"Cannot connect to database"	Check `DATABASE_URL`
"dist not found"	Check build logs for TypeScript errors
"Port already in use"	Don't hardcode port, use `process.env.PORT`

📞 Need Help?

1. Check full guide: `RAILWAY_DEPLOYMENT_GUIDE.md`
2. View Railway logs: Dashboard → Deployments → View Logs

3. Test locally: `npm run build && npm start`

Updated: October 23, 2025