🔧 Railway Deployment Troubleshooting Guide

Last Updated: October 23, 2025
Issue: Railway is still using the FRONTEND Dockerfile instead of BACKEND Dockerfile
Status: GitHub is correct ✅ | Railway needs configuration update ⚠️

✅ Verification Completed

1. GitHub Repository State - CONFIRMED CORRECT ✅

We have verified that the GitHub repository is in the correct state:

- ✅ **Root railway.json REMOVED** - No longer causing conflicts
- ✅ **Latest commits pushed:**
- `7c4bf8d` - Fix: Remove root railway.json to fix backend deployment
- `ed30de7` - Add solution summary document
- ✅ **Backend configuration intact:**
- `backend/railway.json` exists with correct Dockerfile reference
- `backend/Dockerfile` is the correct Node.js backend (NOT nginx)
- `backend/railway.toml` exists

2. Current Problem Analysis

Railway is **STILL USING THE FRONTEND DOCKERFILE** as evidenced by these error logs:

```
[stage-1 1/3] FROM docker.io/library/nginx:alpine
[stage-1 2/3] COPY --from=builder /app/dist /usr/share/nginx/html
ERROR: "/app/dist": not found
```

This means Railway is:
- Using the root `Dockerfile` (frontend) instead of `backend/Dockerfile`
- Looking for nginx.conf at the root level
- Trying to build a React frontend instead of Node.js backend

---

🎯 Root Cause Analysis

Railway is failing because of **ONE OR MORE** of these reasons:

1. ⚠️ Root Directory NOT Set to “backend/”

Most Likely Issue

Railway needs to know to look in the `backend/` directory. If this isn’t set:
- Railway looks at the root directory

- Finds the root `Dockerfile` (frontend)
- Ignores `backend/railway.json`

2. 🔄 Railway Hasn't Auto-Deployed New Commits

Railway may not have triggered a new deployment after your fixes.

3. 💾 Railway Configuration Cache

Railway might be caching the old configuration even after commits.

4. 👁️ Viewing Old Deployment Logs

You might be looking at logs from a deployment BEFORE the fix.

📋 IMMEDIATE VERIFICATION CHECKLIST

Follow these steps in the Railway dashboard:

Step 1: Verify You're Looking at the Latest Deployment

1. Go to Railway dashboard → Your project → `workigom` service
2. Look at the **Deployments** tab
3. Check the **commit hash** of the latest deployment
4. ✅ **VERIFY:** Is it showing commit `7c4bf8d` or `ed30de7`?

If NO: Railway hasn't deployed the latest commits → Go to Step 2
If YES: Railway has the commits but wrong config → Go to Step 3

Step 2: Trigger a Manual Deployment

If Railway hasn't picked up the new commits:

1. Go to the **Deployments** tab
2. Click **"Deploy"** button (top right)
3. Select **"Redeploy from Latest Commit"**
4. Wait for deployment to start
5. Check the **Build Logs** for:
   `Using Detected Dockerfile`
6. **Look for:** Does it mention `nginx:alpine` or `node:20-alpine`?
   - `nginx:alpine` = ❌ WRONG (frontend Dockerfile)
   - `node:20-alpine` = ✅ CORRECT (backend Dockerfile)

Step 3: Verify Root Directory Setting ⚡ CRITICAL

This is the most important setting!

1. Go to your `workigom` service in Railway
2. Click on the **Settings** tab
3. Scroll down to find **"Root Directory"** or **"Source"** section

4. **Check the current value:**

❌ **If it shows:** (empty) or `/` or `frontend/`
- This is WRONG! Railway is looking at the root directory

✅ **Should be:** `backend` or `backend/`
- This tells Railway to work from the backend directory

1. **If it's wrong or empty, SET IT TO:** `backend`
2. Click **"Save"** or **"Update"**
3. This should **automatically trigger a new deployment**

Step 4: Verify Which Dockerfile Railway is Using

After setting Root Directory, check the build logs:

1. Go to **Deployments** → Latest deployment
2. Look at the **Build Logs**
3. Find this line:
   `Using Detected Dockerfile`
4. Below it, check what's being pulled:

✅ **CORRECT:**
```
[internal] load metadata for docker.io/library/node:20-alpine
  [builder 1/6] FROM docker.io/library/node:20-alpine
  COPY package*.json ./
  RUN npm ci
```

❌ **WRONG (what you're seeing now):**
```
[internal] load metadata for docker.io/library/nginx:alpine
  [stage-1 1/3] FROM docker.io/library/nginx:alpine
  COPY --from=builder /app/dist /usr/share/nginx/html
```

🚨 NUCLEAR OPTION: Fresh Start

If the above steps don't work, here's the "nuclear option" to force Railway to forget everything:

Option A: Disconnect and Reconnect Service (Recommended)

1. Go to Railway project → `workigom` service
2. Go to **Settings** tab
3. Scroll to **"Danger Zone"**
4. Click **"Remove Service"** (Don't worry, this won't delete your code!)
5. Confirm removal
6. Click **"+ New Service"**
7. Select **"GitHub Repo"**
8. Choose `volkanakbulut73/workigom`

9. **IMPORTANT:** When configuring:
   - Set **Root Directory** = `backend`
   - Set **Branch** = `master`
10. Configure environment variables (if any)
11. Deploy!

Option B: Force New Build from Scratch

1. Go to **Settings** → **"Danger Zone"**
2. Find **"Clear Build Cache"** or similar option
3. Click it to clear all cached layers
4. Go to **Deployments**
5. Click **"Deploy"** → **"Redeploy from Latest Commit"**

🎯 Success Indicators

You'll know it's working when you see these in the logs:

✅ Correct Build Logs Should Show:

```
Using Detected Dockerfile
====

[internal] load metadata for docker.io/library/node:20-alpine
[builder 1/6] FROM docker.io/library/node:20-alpine

# Prisma-related steps
[builder] RUN npm run prisma:generate

# TypeScript compilation
[builder] RUN npm run build

# Final stage
FROM node:20-alpine
COPY package*.json ./
RUN npm ci --only=production
COPY --from=builder /app/dist ./dist
COPY --from=builder /app/prisma ./prisma
```

✅ Successful Deployment Should Show:

```
✓ Build successful
✓ Starting deployment
✓ Health check passed at /api/health
✓ Deployment successful
```

❌ WRONG - What You're Currently Seeing:

```
[internal] load metadata for docker.io/library/nginx:alpine
[stage-1 1/3] FROM docker.io/library/nginx:alpine
[stage-1 2/3] COPY --from=builder /app/dist /usr/share/nginx/html
ERROR: "/app/dist": not found
```

📊 Quick Diagnostic Commands

If you have Railway CLI installed:

```
# Check current deployment status
railway status

# View current service settings
railway service

# Trigger manual deployment
railway up

# Check environment variables
railway variables
```

🔍 Visual Checklist - What to Look For

In Railway Dashboard:

1. **Project Name:** Should show your project (e.g., “grand-vibrancy” or “profound-vitality”)
2. **Service Name:** Should show “workigom”
3. **GitHub Icon:** Should show `volkanakbulut73/workigom`
4. **Branch:** Should show “master” or “main”
5. **Root Directory:** Should show “backend” ⚠️ THIS IS KEY
6. **Latest Commit:** Should show `7c4bf8d` or `ed30de7`

In Build Logs:

1. **Dockerfile Detection:** Should say “Using Detected Dockerfile”
2. **Docker Image:** Should be `node:20-alpine` (NOT nginx:alpine)
3. **Copy Commands:** Should reference `dist/` and `prisma/` (NOT /app/dist)
4. **No nginx references:** Should NOT see “nginx.conf” anywhere

---

📞 Still Not Working?

If you’ve tried all the above and it’s still failing:

Double-Check These:

1. **Is your Railway project pointing to the correct GitHub repo?**
   - Repo: `volkanakbulut73/workigom`
   - Branch: `master`

2. **Is the Root Directory setting saved?**
   - Sometimes Railway doesn’t save the setting properly
   - Try setting it, saving, refreshing the page, and checking again

3. **Are you looking at the right service?**
   - If you have multiple services, make sure you're configuring the backend one

4. **Check Railway service settings match these:**
   `Root Directory: backend`
   `Branch: master`
   `Build Command: (empty - uses railway.json)`
   `Start Command: (empty - uses railway.json)`
   `Dockerfile Path: (empty - uses default Dockerfile in backend/)`

🎬 Next Steps

1. **FIRST:** Check Root Directory setting in Railway Settings → Source
2. **SECOND:** If wrong, set it to `backend` and save
3. **THIRD:** Wait for auto-deployment or trigger manual deployment
4. **FOURTH:** Check build logs for `node:20-alpine` (not nginx)
5. **FIFTH:** If still wrong, use the Nuclear Option

📝 Summary

The Fix is Simple:
1. Your code on GitHub is CORRECT ✅
2. Railway just needs the **Root Directory** setting updated to `backend`
3. This tells Railway to work from the backend folder
4. Railway will then find `backend/railway.json` and use the correct Dockerfile

Expected Outcome:
- Railway will build using `backend/Dockerfile` (Node.js backend)
- Will use `backend/railway.json` for configuration
- Will deploy your NestJS backend successfully
- Health check at `/api/health` will pass

🔗 Related Documentation

- RAILWAY_ROOT_DIRECTORY_FIX.md (./RAILWAY_ROOT_DIRECTORY_FIX.md) - Detailed fix explanation
- SOLUTION_SUMMARY.md (./SOLUTION_SUMMARY.md) - What we did to fix it
- backend/RAILWAY_DEPLOYMENT_GUIDE.md (./backend/RAILWAY_DEPLOYMENT_GUIDE.md) - Backend deployment guide

---

Need Help? Take screenshots of:
1. Railway Settings → Source/Root Directory section

2. Latest deployment build logs (first 50 lines)
3. Railway service overview showing commit hash

This will help diagnose the exact issue!