

Deployment Issue Analysis & Solution Summary

Issue Diagnosis

What Went Wrong?

Based on the deployment logs you provided, I identified the root cause of your Railway deployment failure:

Error Message:

```
ERROR: failed to solve: failed to calculate checksum of ref: "/nginx.conf": not found
```

Root Cause Analysis:

1. Wrong Dockerfile Being Used:

- Railway was executing the **root-level Dockerfile** (frontend)
- Should have been using **backend/Dockerfile** (backend)

2. Why This Happened:

- The **"Root Directory"** in Railway was NOT set to `backend`
- Railway defaulted to the repository root
- Found the frontend Dockerfile instead of backend Dockerfile

3. Evidence from Logs:

```
COPY nginx.conf /etc/nginx/conf.d/default.conf
```






This line is from the frontend Dockerfile, not the backend one!

What I Fixed

1. Updated `railway.toml`

Location: `/home/ubuntu/workigom/backend/railway.toml`

Changes Made:

-  Added critical warning about setting Root Directory
-  Enhanced configuration with watch patterns
-  Updated start command to use the startup script
-  Added detailed comments for environment variables
-  Included examples for generating secrets

View the file: See below for full content

2. Verified Dockerfile

Location: `/home/ubuntu/workigom/backend/Dockerfile`

Verification Results:

- ☒ Multi-stage build correctly configured
- ☒ TypeScript compilation works (tested locally)
- ☒ Dist directory is created properly
- ☒ Prisma client generation included
- ☒ Production dependencies optimized
- ☒ Startup script handles migrations
- ☒ Health check configured

No changes needed - Dockerfile is already correct!

3. Created Deployment Guides

Created two comprehensive guides:

A. RAILWAY_DEPLOYMENT_GUIDE.md (Full Guide)

- Complete step-by-step deployment instructions
- Detailed troubleshooting section
- Environment variable configuration
- Security best practices
- File structure verification

B. RAILWAY_QUICK_FIX.md (Quick Reference)

- 3-step solution
- Quick troubleshooting table
- Common mistakes to avoid
- Essential commands



What You Need to Do Now

Critical Action Required: Set Root Directory

This is the **ONLY** thing preventing your deployment from working:

1. **Go to Railway Dashboard**
 - Navigate to your project
 - Click on your **backend service**
2. **Open Settings**
 - Find the **“Root Directory”** setting
 - Set it to: `backend`
 - Save the changes
3. **Railway will automatically redeploy**
 - Watch the deployment logs
 - Look for successful build messages

Environment Variables to Configure

In Railway Dashboard → Variables tab, add:

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

# Optional (will use defaults if not set)
NODE_ENV=production
PORT=3001
```

Generate JWT_SECRET:

```
openssl rand -base64 32
```



Verification Checklist

After deployment, verify these:

- ☐ Deployment completes without errors
- ☐ Health endpoint works: `https://your-domain/api/health`
- ☐ Response shows `"status": "OK"`
- ☐ Database connection is established
- ☐ No errors in Railway logs

Expected Response:

```
{
  "status": "OK",
  "timestamp": "2025-10-23T...",
  "uptime": 123.45,
  "database": "connected"
}
```

Updated Files

backend/railway.toml

```
# ⚠ CRITICAL: Railway Backend Configuration
#
# IMPORTANT: Railway Dashboard Settings Required:
# 1. Set "Root Directory" to: backend
# 2. Set "Builder" to: Dockerfile
#
# Without setting the Root Directory to "backend", Railway will use the
# wrong Dockerfile (frontend) and deployment will fail!

[build]
builder = "dockerfile"
dockerfilePath = "Dockerfile"
watchPatterns = ["src/**", "prisma/**", "package.json"]

[deploy]
startCommand = "/app/start.sh"
healthcheckPath = "/api/health"
healthcheckTimeout = 100
restartPolicyType = "ON_FAILURE"
restartPolicyMaxRetries = 10

# Environment Variables (Must be set in Railway Dashboard):
# Required:
# - DATABASE_URL: PostgreSQL connection string (example: postgresql://
# user:password@host:5432/database)
# - JWT_SECRET: Secret key for JWT tokens (generate with: openssl rand -base64 32)
#
# Optional (with defaults):
# - PORT: Server port (default: 3001)
# - NODE_ENV: Environment (default: production)
```

What You Learned

The Problem

Railway's root directory setting determines which directory it considers as the project root. Without setting it to `backend`, Railway looked at your repository root and found the wrong Dockerfile.

The Solution

By setting the root directory to `backend`, Railway will:

1. Use the correct Dockerfile (`backend/Dockerfile`)
2. Find the correct `package.json`
3. Build the TypeScript backend correctly
4. Create the `/app/dist` directory
5. Deploy successfully!

Future Deployments

For any monorepo setup where you have multiple services (frontend, backend, etc.), always remember to:

- Set the root directory correctly

- Use appropriate Dockerfiles for each service
 - Configure environment variables per service
-

Additional Resources Created

All files are in `/home/ubuntu/workigom/backend/` :

1. **RAILWAY_DEPLOYMENT_GUIDE.md** - Complete deployment guide
 2. **RAILWAY_QUICK_FIX.md** - Quick reference card
 3. **DEPLOYMENT_SUMMARY.md** - This file (summary of changes)
 4. **railway.toml** - Updated Railway configuration
-

If Deployment Still Fails

1. **Double-check Root Directory:**
 - Must be exactly: `backend` (lowercase, no trailing slash)
 2. **Check Environment Variables:**
 - `DATABASE_URL` must be valid PostgreSQL connection string
 - `JWT_SECRET` must be set
 3. **View Logs:**
 - Go to Railway Dashboard → Deployments
 - Click on the failed deployment
 - Read the build logs for specific errors
 4. **Common Issues:**
 - Database not accessible from Railway
 - Missing environment variables
 - TypeScript compilation errors
 - Port binding issues
-

Summary

Problem Identified: 

Root Cause Found: 

Solution Provided: 

Files Updated: 

Documentation Created: 

Local Build Tested: 

Next Step: Set Root Directory to `backend` in Railway Dashboard

Analysis Date: October 23, 2025

Status: Ready for Deployment

Confidence Level: High 

Your Dockerfile and railway.toml are correct. The only missing piece is the Root Directory setting in Railway!