

# Railway Deployment Verification Checklist

---

Use this checklist to verify your Railway configuration step by step.

---

## Pre-Flight Checklist

---

### GitHub Repository Status



- ☐ **Railway.json removed from root?**
    - Check: No `railway.json` in root directory
    - ☒ Only `railway.json.frontend.backup` should exist
  - ☐ **Latest commits pushed?**
    - Commit `7c4bf8d` : "Fix: Remove root railway.json to fix backend deployment"
    - Commit `ed30de7` : "Add solution summary document"
    - Run: `git log --oneline -3` to verify
  - ☐ **Backend configuration intact?**
    - ☐ `backend/railway.json` exists
    - ☐ `backend/Dockerfile` exists (Node.js, NOT nginx)
    - ☐ `backend/railway.toml` exists
- 

## Railway Dashboard Verification

---

### Step 1: Service Configuration

**Location:** Railway Dashboard → Your Project → workigom service → Settings

- ☐ **Service is connected to correct GitHub repo?**
    - Expected: `volkanakbulut73/workigom`
    - Where to check: Settings → Source Repo
  - ☐ **Correct branch selected?**
    - Expected: `master`
    - Where to check: Settings → Source → Branch
  - ☐  **ROOT DIRECTORY SET TO backend ? ← MOST IMPORTANT!**
    - Expected: `backend` or `backend/`
    - Where to check: Settings → Source → Root Directory
    -  If empty or showing `/` → THIS IS THE PROBLEM!
-

## Step 2: Latest Deployment Status

**Location:** Railway Dashboard → Your Project → workigom service → Deployments

- [ ] **Latest deployment shows recent commit?**
    - Check commit hash: Should be `7c4bf8d` or later
    - Where to check: Deployments tab → Top deployment card
  - [ ] **Deployment is recent?**
    - Check timestamp: Should be within last hour (or since your fix)
    - If older: Railway hasn't deployed since you pushed the fix
- 

## Step 3: Build Log Verification

**Location:** Latest Deployment → Build Logs

### ✓ **CORRECT Indicators (what you WANT to see):**

- [ ] **Using Node.js base image?**

```
[internal] load metadata for docker.io/library/node:20-alpine
```
- [ ] **Backend build steps present?**

```
[builder 1/6] FROM docker.io/library/node:20-alpine
[builder 2/6] WORKDIR /app
[builder 3/6] COPY package*.json ./
```
- [ ] **Prisma generation step?**

```
RUN npm run prisma:generate
```
- [ ] **TypeScript compilation?**

```
RUN npm run build
```
- [ ] **No nginx references?**
  - Should NOT see: `nginx:alpine`
  - Should NOT see: `COPY nginx.conf`
  - Should NOT see: `/usr/share/nginx/html`

### ✗ **WRONG Indicators (what you're currently seeing):**

- [ ] **Using nginx base image?**

```
[internal] load metadata for docker.io/library/nginx:alpine
```

⚠ If you see this → Railway is using FRONTEND Dockerfile
- [ ] **Looking for /app/dist?**

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

⚠ If you see this → Railway is using FRONTEND Dockerfile
- [ ] **Looking for nginx.conf?**

```
COPY nginx.conf /etc/nginx/conf.d/default.conf
ERROR: "/nginx.conf": not found
```

⚠ If you see this → Railway is using FRONTEND Dockerfile

---

## Step 4: Deployment Success Indicators

**Location:** Latest Deployment → Deploy Logs

- [ ] **Build completed successfully?**
  - ✓ Build successful
- [ ] **Deployment started?**
  - ✓ Starting deployment
- [ ] **Health check passed?**
  - ✓ Health check passed at /api/health
- [ ] **Service is running?**
  - Status should show: “Active” or “Running”
  - Should have a public URL or internal URL



## Action Items Based on Checklist Results



### If Root Directory is NOT set to `backend` :

**Priority: URGENT - This is the main issue!**

1. Go to: Settings → Source
2. Find: “Root Directory” field
3. Set to: `backend`
4. Click: “Save” or “Update”
5. Wait: Railway will auto-deploy
6. Verify: Check build logs for `node:20-alpine`



### If Latest Commit is Old (not 7c4bf8d or ed30de7):

**Railway hasn’t deployed your fix yet!**

1. Go to: Deployments tab
2. Click: “Deploy” button (top right)
3. Select: “Redeploy from Latest Commit”
4. Wait: For deployment to start
5. Verify: Check commit hash matches latest



### If Build Logs Show `nginx:alpine`:

**Railway is still using the wrong Dockerfile!**

This means Root Directory is NOT set correctly:

1. Double-check Settings → Source → Root Directory
2. Must be set to `backend`

3. If already set, try removing and re-adding it
4. Save and wait for auto-deployment

---

### **If Nothing Above Works:**

**Use the Nuclear Option:**

#### **Option A: Disconnect & Reconnect Service**

1. Settings → Danger Zone → “Remove Service”
2. Confirm removal (code is safe in GitHub)
3. “+ New Service” → “GitHub Repo”
4. Select: `volkanakbulut73/workigom`
5. **CRITICAL:** Set Root Directory = `backend` during setup
6. Set Branch = `master`
7. Add environment variables
8. Deploy!

#### **Option B: Clear Build Cache**

1. Settings → Danger Zone
2. Find: “Clear Build Cache” option
3. Click to clear cache
4. Deployments → “Deploy” → “Redeploy from Latest Commit”



## **Quick Reference: Expected Settings**

### **Railway Service Configuration**

```
Service Name: workigom
GitHub Repo: volkanakbulut73/workigom
Branch: master
Root Directory: backend      ← CRITICAL!
Build Command: (empty)      ← Uses railway.json
Start Command: (empty)      ← Uses railway.json
Dockerfile Path: (empty)     ← Uses default Dockerfile in backend/
```

## Expected Build Output

```
# Step 1: Should see Node.js
FROM docker.io/library/node:20-alpine

# Step 2: Should see backend dependencies
COPY package*.json ./
RUN npm ci

# Step 3: Should see Prisma
COPY prisma ./prisma/
RUN npm run prisma:generate

# Step 4: Should see TypeScript build
RUN npm run build

# Step 5: Should see production setup
FROM node:20-alpine
COPY --from=builder /app/dist ./dist
COPY --from=builder /app/prisma ./prisma
```

---

## Success Criteria

You can consider the deployment successful when ALL of these are true:

- ☒ Root Directory is set to `backend` in Railway Settings
- ☒ Latest deployment shows commit `7c4bf8d` or later
- ☒ Build logs show `node:20-alpine` (NOT `nginx:alpine`)
- ☒ Build logs include Prisma generation step
- ☒ Build completes successfully without errors
- ☒ Deployment status shows “Active” or “Running”
- ☒ Health check passes at `/api/health`
- ☒ No references to `nginx` or `/app/dist` in logs

---

## Screenshots to Take (for debugging)

If you need help, take screenshots of:

1. **Settings → Source section** (showing Root Directory)
  2. **Deployments tab** (showing latest commit hash)
  3. **Build logs** (first 30-50 lines showing Dockerfile detection)
  4. **Deploy logs** (showing health check status)
  5. **Service overview** (showing current status)
-

## Timeline Expectations

---

After setting Root Directory to `backend` :

- **0-2 minutes:** Railway triggers auto-deployment
- **2-5 minutes:** Build phase (installing dependencies, Prisma, TypeScript)
- **5-7 minutes:** Deploy phase (starting service, health check)
- **7+ minutes:** Service is live and healthy

If it takes longer, check for:

- Build errors in logs
  - Environment variable issues
  - Database connection problems
- 

## Quick Links

---

### Railway Documentation

- [Root Directory Configuration](https://docs.railway.app/deploy/deployments#root-directory) (<https://docs.railway.app/deploy/deployments#root-directory>)
- [Dockerfile Deployments](https://docs.railway.app/deploy/dockerfiles) (<https://docs.railway.app/deploy/dockerfiles>)

### Your Documentation

- [RAILWAY\\_TROUBLESHOOTING\\_GUIDE.md](#) (`./RAILWAY_TROUBLESHOOTING_GUIDE.md`)
  - [RAILWAY\\_ROOT\\_DIRECTORY\\_FIX.md](#) (`./RAILWAY_ROOT_DIRECTORY_FIX.md`)
  - [SOLUTION\\_SUMMARY.md](#) (`./SOLUTION_SUMMARY.md`)
- 

## Still Stuck?

---

If you've gone through this entire checklist and it's still not working:

1. Take screenshots of the items listed above
  2. Check Railway community forums
  3. Verify your Railway plan has no deployment limits
  4. Check Railway status page for any outages
- 

**Remember:** The most common issue is the Root Directory setting!

Make sure it's set to `backend` in Settings → Source → Root Directory.

Everything else on GitHub is already correct! 