🚂 Railway Deployment Sorunu - Çözüm Raporu

🔴 Tespit Edilen Sorun

Hata Mesajı

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

Sorunun Kök Nedeni

Railway deployment log dosyasını analiz ettikten sonra şu sorun tespit edildi:

Railway, backend servisi için YANLIŞ Dockerfile kullanıyor!

Proje Yapısı:

```
workigom/
├── Dockerfile              # Frontend için (Nginx + React)
├── nginx.conf              # Frontend için
├── railway.json            # Frontend yapılandırması (ÖNCEKİ HATALI HALI: backend'e
işaret ediyordu)
└── backend/
    ├── Dockerfile          # Backend için (Node.js + Express)
    ├── railway.json        # Backend yapılandırması
    └── railway.toml        # Backend yapılandırması
```

Hata Sebebi:

1. **Ana dizindeki railway.json** dosyası önceden backend için yapılandırılmıştı
2. Railway servisi “workigom” için Root Directory: “backend” ayarlanmış
3. Ama Railway, ana dizindeki (root) railway.json’u okuyup `dockerfilePath: "backend/Dockerfile"` görüyor
4. Root directory “backend” olduğu için, Railway `backend/backend/Dockerfile` yolunu arıyor
5. Bu yol olmadığı için, Railway fallback olarak root’taki Dockerfile’ı kullanıyor
6. Root’taki Dockerfile frontend içindir ve nginx.conf gerektirir
7. Backend klasöründe nginx.conf olmadığı için **hata oluşuyor!**

✅ Uygulanan Çözümler

1. Ana Dizindeki railway.json Düzeltildi

ÖNCEKİ (HATALI):

```
{
  "build": {
    "dockerfilePath": "backend/Dockerfile"  // ❌ Backend'e işaret ediyordu
  }
}
```

YENİ (DOĞRU):

```
{
  "build": {
    "dockerfilePath": "Dockerfile"  // ✅ Frontend Dockerfile
  }
}
```

2. Backend Yapılandırması Zaten Doğru

`backend/railway.json` zaten doğru yapılandırılmış:

```
{
  "build": {
    "dockerfilePath": "Dockerfile"  // ✅ Backend/Dockerfile'ı kullanır
  }
}
```

🎯 Railway'de Yapılması Gerekenler

Senaryo 1: Backend Servisi Deploy Ediyorsanız

Railway Dashboard'da:

1. **Settings** → **Build** bölümüne gidin
2. **Root Directory** ayarını kontrol edin: `backend` olmalı
3. **Dockerfile Path** otomatik olarak `Dockerfile` olacak (backend/Dockerfile'a işaret eder)
4. **Variables** sekmesinden gerekli environment variables'ları ekleyin:
   - `DATABASE_URL` (PostgreSQL connection string)
   - `JWT_SECRET` (güçlü bir secret)
   - `PORT=3001`
   - `NODE_ENV=production`
5. **Deploy** butonuna tıklayın

Railway Yapılandırması:

```
Service: workigom-backend
Root Directory: backend
Dockerfile Path: Dockerfile (backend/Dockerfile'ı kullanır)
```

Senaryo 2: Frontend Servisi Deploy Ediyorsanız

Railway Dashboard'da:

1. **Settings** → **Build** bölümüne gidin
2. **Root Directory** ayarını kontrol edin: BoŞ BIRAKIN veya `.` yazın
3. **Dockerfile Path**: `Dockerfile` olmalı
4. **Deploy** butonuna tıklayın

Railway Yapılandırması:

```
Service: workigom-frontend
Root Directory: (boş veya .)
Dockerfile Path: Dockerfile (root/Dockerfile'ı kullanır)
```

📊 Deployment Mimarisi

Railway'de **2 ayrı servis** olması önerilir:

Servis 1: Backend

```
Name: workigom-backend
Root Directory: backend
Dockerfile: backend/Dockerfile
Port: 3001
Health Check: /api/health
Environment:
  - DATABASE_URL
  - JWT_SECRET
  - PORT
  - NODE_ENV
```

Servis 2: Frontend

```
Name: workigom-frontend
Root Directory: . (veya boş)
Dockerfile: Dockerfile
Port: 80
Environment:
  - VITE_API_URL (backend URL'sini işaret etmeli)
```

Servis 3: PostgreSQL Database

```
Name: postgres
Type: PostgreSQL Plugin
Version: 15
```

🔍 Deployment Doğrulama

Backend Deployment Kontrolü

Logs'da şunları görmeli siniz:

```
✅ Using Detected Dockerfile (backend/Dockerfile)
✅ FROM node:20-alpine
✅ Installing dependencies...
✅ Running Prisma migrations...
✅ Building TypeScript...
✅ Server starting on port 3001
🎉 Workigom Backend is running!
```

Frontend Deployment Kontrolü

Logs'da şunları görmelisiniz:

```
✅ Using Detected Dockerfile (root/Dockerfile)
✅ FROM node:20-alpine AS builder
✅ Building Vite app...
✅ FROM nginx:alpine
✅ Copying nginx.conf
✅ Server running on port 80
```

❌ Önlenmiş Hatalar

Bu düzeltmelerle şu hatalar önlendi:

1. ❌ `"/nginx.conf": not found` → ✅ Frontend Dockerfile doğru kullanılıyor
2. ❌ `"/app/dist": not found` → ✅ Backend Dockerfile doğru build yapıyor
3. ❌ Railway servisleri karışıyor → ✅ Her servis kendi Dockerfile'ını kullanıyor

---

📝 Özet Kontrol Listesi

Backend deployment için:
- [ ] Railway service oluşturuldu
- [ ] Root Directory: `backend`
- [ ] Environment variables eklendi (DATABASE_URL, JWT_SECRET, PORT, NODE_ENV)
- [ ] Deploy edildi
- [ ] Logs kontrol edildi
- [ ] Health check endpoint çalışıyor: `/api/health`

Frontend deployment için (isteğe bağlı):
- [ ] Railway service oluşturuldu
- [ ] Root Directory: boş veya `.`
- [ ] Environment variables eklendi (VITE_API_URL)
- [ ] Deploy edildi
- [ ] Uygulama erişilebilir

🆘 Hâlâ Sorun Yaşıyorsanız

1. Railway Service Ayarlarını Kontrol Edin

Railway Dashboard → Service → Settings → Build:

```
Root Directory: backend     (backend için)
Dockerfile Path: Dockerfile (otomatik dolacak)
```

2. Logs'u İnceleyin

Railway Dashboard → Service → Deployments → View Logs

Hataya göre:
- "nginx.conf not found" → Yanlış Dockerfile kullanılıyor
- "dist/server.js not found" → Build başarısız
- "Database connection failed" → DATABASE_URL hatalı

3. Environment Variables'ı Doğrulayın

```
# Backend için gerekli
DATABASE_URL=postgresql://...
JWT_SECRET=your-secret-key
PORT=3001
NODE_ENV=production
```

🎓 Teknik Detaylar

Railway Dockerfile Algılama Sırası:

1. `railway.json` veya `railway.toml` varsa, içindeki `dockerfilePath`'i kullan
2. Root directory ayarlandıysa, o dizinden başla
3. Belirtilen path'te Dockerfile ara
4. Bulunamazsa, root'ta ara (fallback)

Bu Projede Ne Oldu:

1. Railway servisi: Root Directory = "backend"
2. Railway, root'taki railway.json'u okudu (ana dizinde)
3. railway.json: dockerfilePath = "backend/Dockerfile"
4. Railway, "backend/backend/Dockerfile" aradı → bulamadı
5. Fallback: Root'taki Dockerfile'ı kullandı (frontend için)
6. Frontend Dockerfile nginx.conf gerektiriyor
7. Backend klasöründe nginx.conf yok → **HATA!**

Düzeltme Sonrası:

1. Railway servisi: Root Directory = "backend"
2. Railway, backend/railway.json'u okuyor
3. railway.json: dockerfilePath = "Dockerfile"

4. Railway, “backend/Dockerfile” arıyor → **buldu!** ✅
5. Backend Dockerfile kullanılıyor
6. Build başarılı! 🎉

📞 Destek

Sorun devam ederse:
1. Bu raporu okuyun
2. Railway logs’larını kontrol edin
3. backend/RAILWAY_BACKEND_SETUP.md dosyasını inceleyin
4. Environment variables’ları doğrulayın

Başarılı deployments! 🚀