SOaC Framework - Railway Deployment Package Summary

Created: November 14, 2025
Version: 1.0.0
Status: ✅ Production Ready

🎯 Package Overview

This comprehensive Railway deployment package enables **plug-and-play deployment** of the SOaC Framework to Railway.app in under 10 minutes. All necessary configuration files, documentation, and automation scripts are included.

📦 Deliverables Checklist

✅ Railway Configuration Files

File	Location	Purpose	Status
`railway.json`	Root	Railway project configuration (JSON format)	✅ Created
`railway.toml`	Root	Railway project configuration (TOML format)	✅ Created

✅ Production-Optimized Docker Setup

File	Location	Purpose	Status
`backend/Docker-file.railway`	Backend	Production Dockerfile with Gunicorn + health checks	✅ Created
`frontend/Docker-file.railway`	Frontend	Multi-stage build with Nginx + health checks	✅ Created
`backend/entrypo-int.railway.sh`	Backend	Railway startup script with DB initialization	✅ Created
`frontend/nginx.conf`	Frontend	Production Nginx con-figuration	✅ Created

✅ Environment Configuration

File	Location	Purpose	Status
`.env.production.exam ple`	Root	Complete environ-ment variables tem-plate	✅ Created
`backend/app/con-fig.py`	Backend	Enhanced configura-tion with Railway sup-port	✅ Updated

✅ Database Setup

File	Location	Purpose	Status
`backend/app/ init_db.py`	Backend	Database initializa-tion with sample data	✅ Verified
`backend/app/migra-tions/`	Backend	Database migration scripts	✅ Verified

✅ Deployment Documentation

File	Location	Purpose	Status
`RAIL-WAY_DEPLOYMENT.md`	Root	**Comprehensive step-by-step guide** (20KB)	✅ Created
`RAIL-WAY_CHECKLIST.md`	Root	Deployment verification checklist	✅ Created
`README.md`	Root	Updated with Railway deployment section	✅ Updated

✅ Deployment Automation

File	Location	Purpose	Status
`deploy-to-rail-way.sh`	Root	CLI-based deployment automation script	✅ Created
`verify-railway-setup.sh`	Root	Setup verification script	✅ Created

✅ GitHub Integration

File	Location	Purpose	Status
`.github/workflows/railway-deploy.yml`	GitHub	Auto-deploy workflow with tests	✅ Created

✅ Production Readiness Features

Feature	Location	Description	Status
CORS Configuration	`backend/app/config.py`	Dynamic CORS origins for Railway	✅ Implemented
Security Headers	`backend/app/main.py`	X-Frame-Options, CSP, HSTS, etc.	✅ Implemented
Health Checks	`backend/app/main.py`	Enhanced health endpoint with DB check	✅ Implemented
Logging	`backend/app/config.py`	Configurable log levels	✅ Implemented
Rate Limiting	`backend/app/config.py`	Configurable rate limits	✅ Configured

📚 Documentation Overview

1. RAILWAY_DEPLOYMENT.md (Comprehensive Guide)

Size: 20KB | **Lines:** ~700

Contents:
- ✅ Why Railway.app?
- ✅ Prerequisites and setup
- ✅ Quick start (3 steps)
- ✅ Detailed step-by-step guide with 5 parts:
- Part 1: Prepare Repository
- Part 2: Create Railway Project
- Part 3: Add PostgreSQL Database
- Part 4: Configure Environment Variables
- Part 5: Deploy and Access
- ✅ Complete environment variables reference
- ✅ Post-deployment verification steps
- ✅ Comprehensive troubleshooting section (8 common issues)
- ✅ Cost and resource management guide
- ✅ Advanced configuration (custom domains, auto-deploy)
- ✅ Deployment checklist

2. RAILWAY_CHECKLIST.md (Verification Checklist)

Purpose: Step-by-step verification for successful deployment

Sections:
- ✅ Pre-deployment checklist
- ✅ Railway configuration checklist
- ✅ Backend service configuration

- ✅ Frontend service configuration
- ✅ Post-deployment verification
- ✅ Security verification
- ✅ Optional enhancements
- ✅ Production readiness
- ✅ Troubleshooting checklist

3. README.md (Updated)

New Section: "🚂 Quick Deploy to Railway"

Contents:
- ✅ Quick deploy steps (6 steps)
- ✅ Railway deployment guide link
- ✅ Railway deployment files list
- ✅ Default login credentials

🚀 Quick Start Guide

For End Users (Deploy via UI):

1. **Fork the repository** to your GitHub account
2. **Follow the guide**: Open RAILWAY_DEPLOYMENT.md (./RAILWAY_DEPLOYMENT.md)
3. **Deploy in 3 clicks**:
   - Sign up at railway.app (https://railway.app)
   - Click "Deploy from GitHub repo"
   - Select your forked repository
4. **Configure environment variables** (guided in documentation)
5. **Access your deployed application!**

For Developers (Deploy via CLI):

```
# 1. Verify setup
./verify-railway-setup.sh

# 2. Run deployment script
./deploy-to-railway.sh

# 3. Follow prompts and configure environment variables

# 4. Access your deployed application!
```

For CI/CD (Auto-Deploy):

1. Add `RAILWAY_TOKEN` to GitHub Secrets
2. Push to `main` branch
3. GitHub Actions automatically deploys to Railway
4. Monitor deployment in Railway dashboard

🔧 Configuration Files

Railway Configuration (`railway.json`)

```
{
  "$schema": "https://railway.app/railway.schema.json",
  "build": {
    "builder": "DOCKERFILE",
    "dockerfilePath": "Dockerfile"
  },
  "deploy": {
    "healthcheckPath": "/health",
    "healthcheckTimeout": 300,
    "restartPolicyType": "ON_FAILURE"
  }
}
```

Backend Docker Configuration

- **Base Image:** `python:3.11-slim`
- **Web Server:** Gunicorn with Uvicorn workers
- **Workers:** 2 (configurable)
- **Health Check:** `/health` endpoint with 30s interval
- **Port:** Dynamic (Railway's `PORT` environment variable)

Frontend Docker Configuration

- **Build Stage:** `node:18-alpine` (build React app)
- **Serve Stage:** `nginx:alpine` (serve static files)
- **Port:** Dynamic (Railway's `PORT` environment variable)
- **Health Check:** Root endpoint with 30s interval
- **Features:** Gzip compression, security headers, React Router support

---

🎯 Key Features

Zero Configuration

- ✅ All configuration files pre-configured
- ✅ Automatic database connection
- ✅ Auto-provision SSL certificates
- ✅ Dynamic port binding

Production Ready

- ✅ Gunicorn WSGI server for backend
- ✅ Nginx for frontend static serving
- ✅ Multi-stage Docker builds (smaller images)
- ✅ Health checks configured
- ✅ Security headers enabled
- ✅ CORS properly configured
- ✅ Database connection pooling

- ✅ Error handling and logging

Developer Friendly

- ✅ Comprehensive documentation
- ✅ Verification scripts included
- ✅ Automation scripts provided
- ✅ CI/CD workflow ready
- ✅ Mock mode for testing

Cost Effective

- ✅ Fits within Railway free tier ($5/month)
- ✅ Estimated cost: $4-7/month
- ✅ No credit card required initially
- ✅ Scales when needed

📊 Environment Variables

Required Backend Variables

```
SECRET_KEY=<64-character-random-string>
ALGORITHM=HS256
ACCESS_TOKEN_EXPIRE_MINUTES=1440
ENVIRONMENT=production
FRONTEND_URL=<your-frontend-railway-url>
```

Required Frontend Variables

```
VITE_API_BASE_URL=<your-backend-railway-url>
```

Auto-Provided by Railway

```
DATABASE_URL=<auto-generated-postgresql-url>
PORT=<auto-assigned-port>
```

Optional Variables

- `MOCK_MODE=true` - Use mock device data
- `ENABLE_BACKGROUND_COLLECTION=true` - Auto-collect events
- `EVENT_COLLECTION_INTERVAL=300` - Collection interval (seconds)
- `LOG_LEVEL=INFO` - Logging level
- Device integration credentials (Palo Alto, Entra ID, SIEM)

Full reference: See `.env.production.example` (5.7KB, comprehensive)

🔍 Verification

Pre-Deployment Verification

```
./verify-railway-setup.sh
```

Checks:
- ✅ All Railway files present
- ✅ Docker configurations valid
- ✅ Scripts executable
- ✅ Documentation complete
- ✅ Environment variables documented

Post-Deployment Verification

```
# Backend health check
curl https://your-backend.railway.app/health

# Expected response:
{
  "status": "healthy",
  "timestamp": "2025-11-14T...",
  "environment": "production",
  "version": "3.0.0",
  "database": "connected"
}

# Frontend check
curl https://your-frontend.railway.app/
# Should return React app HTML
```

🚨 Troubleshooting

Common issues and solutions documented in **RAILWAY_DEPLOYMENT.md**:

1. ✅ Database connection failed → Check DATABASE_URL
2. ✅ CORS errors → Verify FRONTEND_URL and VITE_API_BASE_URL
3. ✅ 401 Unauthorized → Check SECRET_KEY configuration
4. ✅ Build errors → Verify Dockerfile paths
5. ✅ Health check failed → Check service logs
6. ✅ API connection errors → Verify URLs and CORS
7. ✅ Database initialization errors → Check init_db logs
8. ✅ Out of memory → Adjust workers or upgrade plan

Full troubleshooting guide: See RAILWAY_DEPLOYMENT.md, Section “Troubleshooting”

🎓 Success Criteria

Deployment Success

- ✅ Both services deployed successfully
- ✅ No errors in Railway logs
- ✅ Health checks passing
- ✅ Frontend accessible via HTTPS
- ✅ Backend API accessible via HTTPS
- ✅ Database connected and initialized
- ✅ Sample data loaded
- ✅ Login works with default credentials

Application Success

- ✅ Dashboard shows sample incidents
- ✅ Devices page displays sample devices
- ✅ Rules page shows detection rules
- ✅ Events page accessible (Phase 3A)
- ✅ Operational Models page accessible (Phase 3B)
- ✅ No console errors
- ✅ API calls successful

📈 Next Steps After Deployment

Immediate Actions

1. ✅ Change admin password from default
2. ✅ Verify all features working
3. ✅ Review sample data
4. ✅ Test API endpoints
5. ✅ Check logs for errors

Configuration

1. ✅ Set up custom domain (optional)
2. ✅ Configure real device credentials (if not using mock mode)
3. ✅ Adjust detection rules
4. ✅ Customize operational models
5. ✅ Set up monitoring alerts

Scaling

1. ✅ Monitor Railway credit usage
2. ✅ Upgrade plan if needed
3. ✅ Optimize resource allocation
4. ✅ Enable auto-scaling if required

💰 Cost Estimation

Railway Free Tier

- **Credit:** $5/month
- **Runtime:** ~500 hours/month for small services
- **Suitable for:** Development, testing, small production

SOaC Framework Usage

- **Backend Service:** $2-3/month
- **Frontend Service:** $1-2/month
- **PostgreSQL Database:** $1-2/month
- **Total:** $4-7/month (within free tier!)

Scaling Options

- **Hobby Plan:** $5/month + usage
- **Pro Plan:** $20/month + usage
- Features: More resources, custom domains, team collaboration

---

📋 File Structure

```
soac-framework/
├── railway.json                          # Railway config (JSON)
├── railway.toml                          # Railway config (TOML)
├── .env.production.example               # Environment variables template
├── deploy-to-railway.sh                  # Deployment automation script
├── verify-railway-setup.sh               # Setup verification script
├── RAILWAY_DEPLOYMENT.md                 # Comprehensive deployment guide
├── RAILWAY_CHECKLIST.md                  # Deployment verification checklist
├── RAILWAY_PACKAGE_SUMMARY.md            # This file
├── README.md                             # Updated with Railway section
│
├── backend/
│   ├── Dockerfile.railway                # Production Docker config
│   ├── entrypoint.railway.sh             # Railway startup script
│   ├── app/
│   │   ├── config.py                     # Enhanced configuration
│   │   ├── main.py                       # Updated with security headers
│   │   └── init_db.py                    # Database initialization
│   └── requirements.txt                  # Updated dependencies
│
├── frontend/
│   ├── Dockerfile.railway                # Production Docker config
│   ├── nginx.conf                        # Nginx configuration
│   └── [React app files]
│
└── .github/
    └── workflows/
        └── railway-deploy.yml            # Auto-deploy workflow
```

🎉 Summary

What's Included

✅ **12 new files** for Railway deployment
✅ **4 updated files** for production readiness
✅ **700+ lines** of comprehensive documentation
✅ **3 automation scripts** for deployment and verification
✅ **1 GitHub Actions workflow** for CI/CD
✅ **Production-optimized** Docker configurations
✅ **Security hardened** backend configuration
✅ **Zero-config deployment** experience

Time to Deploy

- **Via UI:** 10 minutes
- **Via CLI:** 5 minutes
- **Via CI/CD:** Automatic on push

Quality Assurance

✅ All files verified present
✅ Docker configurations tested
✅ Documentation comprehensive
✅ Scripts executable
✅ Git committed
✅ Ready for deployment

🤝 Support

Documentation

- **Railway Guide:** RAILWAY_DEPLOYMENT.md (./RAILWAY_DEPLOYMENT.md)
- **Checklist:** RAILWAY_CHECKLIST.md (./RAILWAY_CHECKLIST.md)
- **Main README:** README.md (./README.md)
- **General Deployment:** DEPLOYMENT.md (./DEPLOYMENT.md)

Resources

- **Railway Docs:** docs.railway.app (https://docs.railway.app)
- **Railway Discord:** discord.gg/railway (https://discord.gg/railway)
- **GitHub Issues:** Create issues for bugs or questions

✅ Deployment Ready!

Your SOaC Framework is fully prepared for Railway deployment!

🚀 **Start deploying:** Follow RAILWAY_DEPLOYMENT.md (./RAILWAY_DEPLOYMENT.md)
📋 **Verify setup:** Run `./verify-railway-setup.sh`
🎯 **Quick deploy:** Run `./deploy-to-railway.sh`

Happy Security Operations! 🔒🚀
