# Railway Deployment Checklist

This checklist ensures you have everything ready for a successful Railway deployment.

## ✅ Pre-Deployment Checklist

### Repository Setup

- [ ] Repository forked or cloned to your GitHub account
- [ ] All Railway deployment files present:
- [ ] `railway.json`
- [ ] `railway.toml`
- [ ] `backend/Dockerfile.railway`
- [ ] `frontend/Dockerfile.railway`
- [ ] `backend/entrypoint.railway.sh`
- [ ] `frontend/nginx.conf`
- [ ] `.env.production.example`
- [ ] `RAILWAY_DEPLOYMENT.md`
- [ ] `deploy-to-railway.sh`
- [ ] `.github/workflows/railway-deploy.yml`

### Railway Account

- [ ] Railway account created at railway.app (https://railway.app)
- [ ] GitHub account connected to Railway
- [ ] Repository access granted to Railway

## ✅ Railway Configuration Checklist

### Project Setup

- [ ] New Railway project created
- [ ] GitHub repository connected
- [ ] PostgreSQL database added
- [ ] `DATABASE_URL` environment variable auto-generated

### Backend Service Configuration

- [ ] Service created from GitHub repo
- [ ] Root Directory set to: `backend`
- [ ] Dockerfile Path set to: `Dockerfile.railway`
- [ ] Environment variables configured:
- [ ] `SECRET_KEY` (64+ character random string)
- [ ] `ALGORITHM=HS256`
- [ ] `ACCESS_TOKEN_EXPIRE_MINUTES=1440`
- [ ] `ENVIRONMENT=production`
- [ ] `FRONTEND_URL` (will be set after frontend deployment)

- [ ] `MOCK_MODE=true` (or `false` for real devices)
- [ ] `ENABLE_BACKGROUND_COLLECTION=true`
- [ ] `EVENT_COLLECTION_INTERVAL=300`
- [ ] Backend service deployed successfully
- [ ] Backend domain generated
- [ ] Backend health check passing: `curl https://your-backend.railway.app/health`

### Frontend Service Configuration

- [ ] Service created from GitHub repo
- [ ] Root Directory set to: `frontend`
- [ ] Dockerfile Path set to: `Dockerfile.railway`
- [ ] Environment variables configured:
- [ ] `VITE_API_BASE_URL` (your backend Railway URL)
- [ ] Frontend service deployed successfully
- [ ] Frontend domain generated
- [ ] Frontend accessible in browser

### Cross-Service Configuration

- [ ] Backend `FRONTEND_URL` updated with actual frontend URL
- [ ] Frontend `VITE_API_BASE_URL` updated with actual backend URL
- [ ] Both services redeployed after URL updates

## ✅ Post-Deployment Verification

### Backend Verification

- [ ] Backend URL accessible: `https://your-backend.railway.app`
- [ ] Health endpoint returns 200: `curl https://your-backend.railway.app/health`
- [ ] API documentation accessible: `https://your-backend.railway.app/docs`
- [ ] Database connection successful (check health endpoint response)
- [ ] No errors in Railway logs

### Frontend Verification

- [ ] Frontend URL accessible: `https://your-frontend.railway.app`
- [ ] Login page loads correctly
- [ ] No console errors in browser
- [ ] API calls successful (check browser network tab)

### Functional Testing

- [ ] Login with admin credentials (`admin` / `admin123`)
- [ ] Dashboard loads with sample data
- [ ] Devices page shows sample devices
- [ ] Rules page displays detection rules
- [ ] Incidents page shows sample incidents
- [ ] Events page accessible (Phase 3A)
- [ ] Operational Models page accessible (Phase 3B)

### Security Verification

- [ ] Admin password changed from default
- [ ] `SECRET_KEY` is unique and strong (not default value)
- [ ] HTTPS working (Railway provides automatic SSL)
- [ ] CORS configured correctly (no console errors)
- [ ] Database credentials secure (managed by Railway)

## ✅ Optional Enhancements

### Custom Domain

- [ ] Custom domain added in Railway settings
- [ ] DNS records configured
- [ ] SSL certificate provisioned
- [ ] Application accessible via custom domain

### GitHub Auto-Deploy

- [ ] `.github/workflows/railway-deploy.yml` present
- [ ] `RAILWAY_TOKEN` added to GitHub Secrets
- [ ] Test deployment by pushing to main branch
- [ ] Workflow runs successfully

### Monitoring

- [ ] Railway logs reviewed for errors
- [ ] Railway metrics dashboard checked
- [ ] Health check endpoint monitored
- [ ] Alert notifications configured (optional)

### Device Integration (If Not Using Mock Mode)

- [ ] `MOCK_MODE=false` set
- [ ] Palo Alto NGFW credentials configured
- [ ] Entra ID tenant credentials configured
- [ ] SIEM credentials configured
- [ ] Device connections tested via UI
- [ ] Event collection working

## ✅ Production Readiness

### Performance

- [ ] Backend worker count appropriate (2 workers in `entrypoint.railway.sh`)
- [ ] Database connection pool configured
- [ ] Frontend build optimized (production mode)
- [ ] Static assets cached properly

### Monitoring & Logging

- [ ] Railway logs reviewed regularly
- [ ] Error patterns identified and resolved

- [ ] Performance metrics monitored
- [ ] Database performance acceptable

### Backup & Recovery

- [ ] Database backup strategy planned
- [ ] Railway automatic backups enabled (if on paid plan)
- [ ] Export important data regularly
- [ ] Recovery procedure documented

### Scaling

- [ ] Current resource usage monitored
- [ ] Scaling plan defined if needed
- [ ] Railway plan upgraded if free tier insufficient

## 🚨 Troubleshooting Checklist

If deployment fails, check:
- [ ] All environment variables set correctly
- [ ] No typos in service configuration
- [ ] Dockerfile paths correct (relative to root directory)
- [ ] Database URL present and correct
- [ ] CORS origins match actual URLs
- [ ] No port conflicts or hardcoded localhost URLs
- [ ] Railway logs for specific error messages
- [ ] PostgreSQL service running and healthy

## 📊 Resource Usage Monitoring

Monitor these to stay within Railway free tier:
- [ ] Monthly credit usage (Railway dashboard)
- [ ] Service CPU usage
- [ ] Service memory usage
- [ ] Database storage usage
- [ ] Network bandwidth usage

Free tier includes $5 monthly credit, typically covering:
- 2 small services (backend + frontend)
- 1 PostgreSQL database
- ~500 hours runtime
- Perfect for testing and development!

## 📚 Documentation References

- [ ] Read RAILWAY_DEPLOYMENT.md (./RAILWAY_DEPLOYMENT.md) completely
- [ ] Reviewed README.md (./README.md) Railway section
- [ ] Checked .env.production.example (./.env.production.example) for all variables
- [ ] Consulted Railway Documentation (https://docs.railway.app/)

# ✅ Deployment Complete!

Once all items are checked:
- ✅ Your SOaC Framework is successfully deployed to Railway!
- ✅ Application accessible via HTTPS
- ✅ Database initialized with sample data
- ✅ Ready for testing and development

**Next Steps:**
1. Explore the application features
2. Configure real device integrations (optional)
3. Customize detection rules
4. Monitor security incidents
5. Adjust operational models as needed

**Happy Security Operations!** 🔒🚀

---

# Need Help?

- **Railway Issues**: Check Railway Docs (https://docs.railway.app/) or Railway Discord (https://discord.gg/railway)
- **SOaC Framework Issues**: See RAILWAY_DEPLOYMENT.md (./RAILWAY_DEPLOYMENT.md) troubleshooting section
- **General Questions**: Review README.md (./README.md) and DEPLOYMENT.md (./DEPLOYMENT.md)

---