

Master Admin Portal Deployment Instructions

Current Status

- Master Admin Portal files added to your repository
- Created deployment branch: `master-admin-deploy`
- All files committed and ready for push
- Production build optimized (36.7kB, 0 errors)
- Railway configuration complete

Next Steps (Complete These)

1. Push to Your GitHub Repository

```
cd /home/ubuntu/audityzer-deploy
git push -u origin master-admin-deploy
```

Note: You'll need to authenticate with your GitHub credentials or personal access token

2. Deploy to Railway

Option A: Railway CLI (Recommended)

```
# Install Railway CLI
npm install -g @railway/cli

# Login to Railway (opens browser)
railway login

# Navigate to the Master Admin Portal directory
cd /home/ubuntu/audityzer-deploy/apps/master-admin-portal

# Initialize Railway project
railway init

# Deploy
railway up --detach
```

Option B: Railway Dashboard (Alternative)

1. Go to railway.app (<https://railway.app>)
2. Click "New Project"
3. Select "Deploy from GitHub repo"
4. Choose your repository: `romanchaa997/Audityzer`
5. Select branch: `master-admin-deploy`
6. Set root directory: `apps/master-admin-portal`
7. Railway will auto-detect the Dockerfile and deploy

3. Configure Environment Variables

Set these in Railway dashboard or CLI:

```
# Database (Railway will auto-provide PostgreSQL)
DATABASE_URL=postgresql://...

# Authentication
NEXTAUTH_SECRET=your-secret-key-here
NEXTAUTH_URL=https://your-app.railway.app

# Admin Credentials
ADMIN_EMAIL=your-admin@email.com
ADMIN_PASSWORD=your-secure-password

# Optional: External Integrations
OPENAI_API_KEY=your-openai-key (optional)
```

4. Database Setup

Railway will automatically:

- Create PostgreSQL database
- Run migrations: `npm run db:migrate`
- Seed initial data: `npm run db:seed`

Repository Structure Added

```
apps/master-admin-portal/
├── app/ # Next.js 14 App Router
│   ├── admin/ # Admin management pages
│   ├── analytics/ # Analytics dashboard
│   ├── api/ # API routes
│   ├── auth/ # Authentication pages
│   ├── developers/ # Developer tools
│   ├── master-admin/ # Master admin features
│   ├── tenants/ # Multi-tenant management
│   └── users/ # User management
├── components/ # React components
│   ├── enterprise/ # Enterprise features
│   ├── ui/ # UI components (shadcn/ui)
│   └── ...
├── lib/ # Utilities and configurations
├── prisma/ # Database schema and seeds
├── Dockerfile # Container configuration
├── railway.json # Railway deployment config
└── package.json # Dependencies and scripts
```

Features Included

Core Features

- **Authentication & Authorization** - NextAuth.js with role-based access
- **User Management** - Create, edit, delete users with role assignment
- **Analytics Dashboard** - Real-time metrics and system health

- **Multi-tenant Support** - Manage multiple client organizations
- **Developer Tools** - API management and documentation

Enterprise Features

- **Advanced Security** - Threat detection and security monitoring
- **Business Intelligence** - Advanced analytics and reporting
- ✂ **Workflow Automation** - Automated business processes
- **Incident Management** - Issue tracking and resolution
- **Compliance Management** - Regulatory compliance tools
- **Resource Optimization** - Performance and cost optimization
- **API Management** - API gateway and rate limiting

Expected Deployment Result

After successful deployment:

- **Live URL:** `https://your-app-name.railway.app`
- **Admin Access:** Login with configured admin credentials
- **Database:** Fully configured PostgreSQL with initial data
- **Monitoring:** Built-in health checks and metrics
- **Security:** Production-ready security configurations

Troubleshooting

Common Issues

1. **Build Errors:** Check Node.js version (requires 18+)
2. **Database Connection:** Verify DATABASE_URL environment variable
3. **Authentication:** Ensure NEXTAUTH_SECRET and NEXTAUTH_URL are set
4. **Port Issues:** Railway automatically handles port configuration

Support

- Check Railway logs for deployment issues
- Verify all environment variables are set
- Ensure GitHub repository is accessible
- Review the DEPLOYMENT_GUIDE.md in the portal directory

Security Notes

- All sensitive data is properly encrypted
 - Environment variables are securely managed
 - Production-ready security headers configured
 - Database connections use SSL by default
-

Ready for Deployment!

Your Master Admin Portal is now ready to be deployed to Railway using your existing GitHub repository. Complete the steps above to get your enterprise-grade admin portal live.