

SOC-PORTAL APPLICATION MIGRATION GUIDE

Prepared by: Sabbir Hossain Borno

Date: 30-10-2025

1. Current Environment Overview

- **Application:** SOC-PORTAL
 - **Framework:** Next.js 15.5.4
 - **Node.js Version:** (Check node -v)
 - **Package Manager:** npm
 - **Process Manager:** PM2
 - **Database:** PostgreSQL 13.16
 - **DB Name:** SOC_PORTAL_DB
 - **DB User:** ngd_soc
 - **Web Server:** Nginx (port 8080)
 - **Reverse Proxy:** Node.js 127.0.0.1:5001
 - **Directory Structure:**
 - /home/soc_portal/
 - └─ app
 - └─ lib
 - └─ public
 - └─ node_modules
 - └─ logs
 - └─ pm2_logs
 - └─ ecosystem.config.js
 - └─ package.json
 - └─ package-lock.json
-

2. Pre-Migration Checklist (New Server)

1. OS and Updates

- CentOS 9 or Ubuntu 22.04, fully updated: **dnf update -y or apt update && apt upgrade -y**

2. Install Required Packages

3. **dnf install -y gcc make wget curl git nginx postgresql postgresql-server postgresql-contrib**

4. Install Node.js and npm

5. `curl -fsSL https://rpm.nodesource.com/setup_20.x | bash -`

6. **dnf install -y nodejs**

7. `node -v`

8. `npm -v`

9. Install PM2 Globally

10. **npm install -g pm2**

11. Database Setup

12. `postgresql-setup initdb` # CentOS

13. `systemctl enable --now postgresql`

14. `sudo -i -u postgres psql`

15. `CREATE USER ngd_soc WITH PASSWORD 'yourpassword';`

16. `CREATE DATABASE SOC_PORTAL_DB OWNER ngd_soc;`

17. `GRANT ALL PRIVILEGES ON DATABASE SOC_PORTAL_DB TO ngd_soc;`

18. Nginx Setup

- Copy `soc-portal.conf` to `/etc/nginx/conf.d/`
- Test: `nginx -t`
- Reload: `systemctl reload nginx`

19. Firewall

20. `firewall-cmd --add-port=8080/tcp --permanent`

21. firewall-cmd --add-port=5001/tcp --permanent

22. firewall-cmd --reload

23. Clone Application Repository

24. git clone <repo-url> /home/soc_portal

25. cd /home/soc_portal

26. npm install

3. Installed Node.js Libraries

@emnapi/core@1.4.5

lucide-react@0.536.0

@emnapi/runtime@1.4.5

luxon@3.7.1

@emnapi/wasi-threads@1.0.4

moment-timezone@0.6.0

@eslint/eslintrc@3.3.1

next@15.5.4

@napi-rs/wasm-runtime@0.2.12

nodemailer@7.0.9

@tailwindcss/postcss@4.1.11

pg@8.16.3

@tybys/wasm-util@0.10.0

react-chartjs-2@5.3.0

axios@1.12.2

react-datepicker@8.8.0

bcryptjs@3.0.2

react-dom@19.1.0

chart.js@4.5.0

react-hot-toast@2.5.2

chartjs-plugin-datalabels@2.2.0

react-icons@5.5.0

eslint-config-next@15.4.5

react-select@5.10.2

eslint@9.32.0

react-toastify@11.0.5

exceljs@4.4.0

react@19.1.0

framer-motion@12.23.12

tailwindcss@4.1.11

html-to-image@1.11.13

uuid@11.1.0

html2canvas@1.4.1

winston-daily-rotate-file@5.0.0

js-cookie@3.0.5

winston@3.17.0

lodash@4.17.21

4. PM2 Setup

- `cp ecosystem.config.js /home/soc_portal/`
- `pm2 start ecosystem.config.js --env production`
- `pm2 save`
- `pm2 startup`
- `pm2 list`
- `pm2 logs SOC-PORTAL`

5. Nginx Configuration

- Reverse proxy to Node.js: 127.0.0.1:5001
- Headers and WebSocket configuration copied from `soc-portal.conf`
- Test and reload:

`nginx -t`

`systemctl reload nginx`

6. Database Migration

- Backup current DB:

`pg_dump -U ngd_soc -d SOC_PORTAL_DB -h localhost > SOC_PORTAL_DB_backup.sql`

- Restore on new server:

`psql -U ngd_soc -d SOC_PORTAL_DB -h localhost < SOC_PORTAL_DB_backup.sql`

7. Environment Variables

- Set in `.env` or `ecosystem.config.js`:
- `NODE_ENV=production`
- `PORT=5001`
- `HOST=0.0.0.0`
- `NEXTAUTH_URL=http://<new-server-ip>:8080`
- `NEXTAUTH_SECRET=<new-secret-key>`

8. Post-Migration Verification

- Check PM2 logs for errors: pm2 logs SOC-PORTAL
 - Open application in browser: http://<new-server-ip>:8080
 - Verify database tables and content
 - Test uploads, email notifications, WebSocket connections
-

9. Optional Enhancements

- Enable SSL/TLS with Let's Encrypt
- Configure automated backups for PostgreSQL and /home/soc_portal/public/storage
- Set up monitoring with PM2 or external tools