KINTO QA
Management System

On-Premise Deployment Guide

Version 1.0 | November 2025

Table of Contents

1. System Overview

The KINTO QA Management System is a comprehensive manufacturing quality assurance platform designed for on-premise deployment. This system provides:

- Email/password authentication with role-based access control
- Dynamic role management (Admin, Operator, Reviewer, Manager)
- Checklist builder and execution
- Preventive maintenance tracking
- Spare parts management
- Purchase order generation
- PDF export capabilities
- Mobile-responsive Progressive Web App interface

2. System Requirements

Minimum Hardware Requirements:

- Operating System: Ubuntu 20.04 LTS or higher
- CPU: 2 cores minimum (4 cores recommended)
- RAM: 4 GB minimum (8 GB recommended)
- Storage: 50 GB minimum (100 GB recommended)
- Network: Static IP address for production access

Software Requirements:

- Node.js 20.x LTS
- PostgreSQL 14 or higher
- Nginx (reverse proxy)
- PM2 (process manager)
- Git (for code deployment)

3. Server Setup

Step 1: Update System

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Node.js 20 LTS

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
```

Step 3: Install PostgreSQL

```
sudo apt install postgresql postgresql-contrib -y
```

Step 4: Install Nginx

```
sudo apt install nginx -y
```

Step 5: Install PM2

```
sudo npm install -g pm2
```

Step 6: Install Build Tools

```
sudo apt install build-essential -y
```

4. Database Installation

Step 1: Create Database and User

Switch to postgres user and create database:

```
sudo -u postgres psql
```

Execute the following SQL commands:

```
CREATE DATABASE kinto_qa;
CREATE USER kinto_admin WITH ENCRYPTED PASSWORD 'YourStrongPassword123!';
GRANT ALL PRIVILEGES ON DATABASE kinto_qa TO kinto_admin;
\q
```

Step 2: Run Database Initialization Scripts

Navigate to the database folder and execute scripts in order:

```
cd /path/to/kinto-qa/database
psql -h localhost -U kinto_admin -d kinto_qa -f 01_init_schema.sql
psql -h localhost -U kinto_admin -d kinto_qa -f 02_seed_roles.sql
psql -h localhost -U kinto_admin -d kinto_qa -f 03_seed_admin_user.sql
```

Step 3: Verify Database Installation

```
psql -h localhost -U kinto_admin -d kinto_qa -c "SELECT * FROM roles;"
```

Step 4: Configure PostgreSQL for Production

Edit PostgreSQL configuration:

```
sudo nano /etc/postgresql/14/main/postgresql.conf
```

Update these settings:

```
shared_buffers = 256MB
effective_cache_size = 1GB
maintenance_work_mem = 64MB
work_mem = 16MB
max_connections = 100
```

Restart PostgreSQL:

```
sudo systemctl restart postgresql
sudo systemctl enable postgresql
```

5. Application Deployment

Step 1: Create Application Directory

```
sudo mkdir -p /var/www/kinto-qa
sudo chown -R $USER:$USER /var/www/kinto-qa
```

Step 2: Copy Application Files

Upload your application files to /var/www/kinto-qa using SCP, SFTP, or Git.

Step 3: Install Dependencies

```
cd /var/www/kinto-qa
npm install --production
```

Step 4: Build Frontend

```
npm run build
```

Step 5: Create Environment File

Create .env file in /var/www/kinto-qa:

```
NODE_ENV=production
PORT=5000

# Database Configuration
DATABASE_URL=postgresql://kinto_admin:YourPassword@localhost:5432/kinto_qa
PGHOST=localhost
PGPORT=5432
PGDATABASE=kinto_qa
PGUSER=kinto_admin
PGPASSWORD=YourPassword

# Session Secret (32+ character random string)
SESSION_SECRET=your_long_random_secret
```

Step 6: Start Application with PM2

```
pm2 start server/index.js --name kinto-qa
pm2 save
pm2 startup
```

6. Nginx Configuration

Step 1: Create Nginx Configuration File

Create /etc/nginx/sites-available/kinto-qa and configure as reverse proxy to port 5000.

Step 2: Enable Site

```
sudo ln -s /etc/nginx/sites-available/kinto-qa /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
sudo systemctl enable nginx
```

7. SSL Certificate Setup (Optional but Recommended)

Using Let's Encrypt for Free SSL:

```
sudo apt install certbot python3-certbot-nginx -y
sudo certbot --nginx -d yourdomain.com
```

8. Security Configuration

Firewall Setup:

```
sudo ufw enable
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw status
```

Security Checklist:

& Change default admin password immediately
& Use strong database passwords
& Enable HTTPS/SSL certificates
& Configure firewall rules
& Keep system and packages updated
& Monitor application logs regularly
& Implement automated backups

9. Backup Strategy

Database Backup:

```
pg_dump -U kinto_admin kinto_qa > backup.sql
```

Automated Daily Backups:

Create backup script and add to crontab:

```
sudo crontab -e
```

Add daily backup at 2 AM:

```
0 2 * * * /usr/local/bin/backup-kinto-qa.sh
```

10. Monitoring & Maintenance

Essential Monitoring Commands:

```
pm2 logs kinto-qa           # View application logs
pm2 monit                   # Monitor processes
sudo systemctl status nginx
sudo systemctl status postgresql
df -h                       # Check disk usage
free -h                     # Check memory usage
```

Regular Maintenance Tasks:

- Monitor disk space and clean up old logs
- Review application logs for errors
- Verify automated backups are running
- Update system packages monthly
- Review and rotate SSL certificates
- Check database performance metrics

11. Troubleshooting

Application Won't Start:

Check PM2 logs for errors:

```
pm2 logs kinto-qa
```

Database Connection Issues:

Test database connection:

```
psql -h localhost -U kinto_admin -d kinto_qa
```

Nginx Errors:

```
sudo nginx -t
sudo tail -f /var/log/nginx/error.log
```

Appendix A: Default Credentials

Default Admin Account

Username:

admin

Password:

Admin@123

Email:

admin@kinto.com

& þ IMPORTANT: Change this password immediately after first login!

Database Credentials

Database:

kinto_qa

User:

kinto_admin

Password:

(Set during installation)

Appendix B: Quick Command Reference

PM2 Commands:

```
pm2 start server/index.js --name kinto-qa    # Start
pm2 restart kinto-qa                         # Restart
pm2 stop kinto-qa                            # Stop
pm2 logs kinto-qa                            # Logs
pm2 monit                                    # Monitor
```

Nginx Commands:

```
sudo nginx -t                                # Test config
sudo systemctl restart nginx                 # Restart
sudo systemctl status nginx                  # Status
```

PostgreSQL Commands:

```
sudo systemctl status postgresql             # Status
sudo -u postgres psql                        # Access
pg_dump -U kinto_admin kinto_qa > backup.sql # Backup
```

End of Document

For technical support, refer to project documentation in /database/README.md
or contact your system administrator.