MySLT Dashboard Hosting

🚀 MERN Dashboard Deployment - Complete Project Documentation

📋 Project Overview

Application: MySLT Monitoring Dashboard (MERN Stack)

Repository: https://github.com/Omindu1015/MYSLT-DASHBOARD

Deployment Date: December 13, 2025

Final Status: ✅ **Production Ready with HTTPS**

📁 Repository Structure (Monorepo Layout)

```
MYSLT-DASHBOARD/
├── client/          # React/Vite Frontend
├── Server/          # Node.js/Express Backend
├── package.json     # Root package.json
└── README.md        # Project documentation
```

🔄 Complete Deployment Timeline with Issues & Solutions

Phase 1: Environment Setup

Step 1: Create Rocky Linux VM in Proxmox

- **Action:** Created new VM with Rocky Linux 9.7
- **Configuration:** 2 CPU cores, 4GB RAM, 32GB storage
- **Network:** Bridged to vmbr0 with static IP `192.168.100.137`

Step 2: Initial SSH Configuration

- **Issue:** External SSH access failed with "Connection timed out"
- **Root Cause:** pfSense port forwarding misconfigured
- **Solution:**
  - ✅ Added port forward: WAN:9120 → 192.168.100.137:22
  - ✅ Fixed "No RDR (NOT)" checkbox (was incorrectly enabled)
  - ✅ Set correct destination to "WAN address" not specific IP
- **Result:** SSH access working via `ssh dpd@124.43.216.136 -p 9120`

Phase 2: System & Dependency Installation

Step 3: System Updates & Basic Tools

```
# Initial system update
sudo dnf update -y

# Install essential tools (git, curl, wget, vim)
sudo dnf install -y git curl wget vim

# Install development tools
sudo dnf groupinstall -y "Development Tools"
```

✅ **Status:** Completed without issues

Step 4: Node.js Installation

```
# Add Node.js 22.x repository
curl -fsSL <https://rpm.nodesource.com/setup_22.x> | sudo bash -
sudo dnf install -y nodejs
```

✅ **Verification:** Node.js v22.21.1 installed

Step 5: MongoDB Installation Challenges

First Attempt:

```
# MongoDB not in default repos
sudo dnf install -y mongodb mongodb-server
# ❌ ERROR: "No match for argument: mongodb"
```

Second Attempt (MongoDB 8.2):

```
# Added MongoDB 8.2 repository
sudo tee /etc/yum.repos.d/mongodb-org-8.2.repo << 'EOF'
[mongodb-org-8.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/9/mongodb-org/8.2/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://pgp.mongodb.com/server-8.2.asc
EOF

sudo dnf install -y mongodb-org
# ❌ GPG KEY ERROR: Status code: 404 for <https://pgp.mongodb.com/server-8.2.asc>
```

Third Attempt (MongoDB 8.0):

```
# Switched to MongoDB 8.0
sudo tee /etc/yum.repos.d/mongodb-org-8.0.repo << 'EOF'
[mongodb-org-8.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/9/mongodb-org/8.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://pgp.mongodb.com/server-8.0.asc
```

```
EOF

sudo rpm --import <https://pgp.mongodb.com/server-8.0.asc>
sudo dnf install -y mongodb-org

sudo systemctl start mongod
# ❌ CRITICAL ERROR: "Failed to start mongod.service: Unit mongod.service not found"
```

Root Cause Analysis:

- GPG key verification failed during installation
- Packages downloaded but not properly installed
- No MongoDB service file created

Verification:

```
rpm -qa | grep mongo
# ❌ OUTPUT: (empty) - No packages installed
```

Fourth Attempt (Successful Installation):

```
# Clean previous attempts
sudo dnf remove -y mongodb-org*
sudo rm -f /etc/yum.repos.d/mongodb*.repo

# Reinstall with proper GPG key
sudo tee /etc/yum.repos.d/mongodb-org-8.0.repo << 'EOF'
[mongodb-org-8.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/9/mongodb-org/8.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://pgp.mongodb.com/server-8.0.asc
EOF
```

```
sudo rpm --import <https://pgp.mongodb.com/server-8.0.asc>
sudo dnf install -y mongodb-org

sudo systemctl start mongod
# ❌ NEW ERROR: "Active: failed (Result: core-dump)"
```

AVX Instruction Set Issue:

- **Error:** SIGILL (Illegal Instruction) crash
- **Root Cause:** MongoDB 8.0 requires AVX CPU instructions
- **VM Configuration:** Proxmox CPU type was incompatible

Final Solution:

1. **Shutdown VM** in Proxmox
2. **Changed CPU type** to `host` (passthrough)
3. **Restarted VM**
4. **Reinstalled MongoDB 8.2** (now works with AVX)

```
sudo systemctl start mongod
# ✅ SUCCESS: "Active: running"
```

Phase 3: Application Deployment

Step 6: Project Setup

```
# Create project directory
sudo mkdir -p /var/www
sudo chown -R dpd:dpd /var/www
cd /var/www

# Clone repository (SSH method used)
```

```
git clone git@github.com:0mindu1015/MYSLT-DASHBOARD.git
cd MYSLT-DASHBOARD
```

Step 7: Backend Deployment

```
cd Server
npm install

# Install missing dependency found during runtime
npm install net-snmp

# Configure environment
nano .env
# Updated to:
# MONGODB_URI=mongodb+srv://... (MongoDB Atlas cloud)
# MONGODB_LOCAL_URI=mongodb://localhost:27017/myslt_dashboard
# PORT=5001
# NODE_ENV=production

# Test backend
node src/server.js
# ✅ SUCCESS: Server running on port 5001, MongoDB connected
```

Step 8: PM2 Process Manager Setup

```
sudo npm install -g pm2
pm2 start src/server.js --name myslt-backend
pm2 save
pm2 startup
# ✅ Backend now auto-starts on boot
```

Step 9: Frontend Build

```
cd ../client
npm install
npm run build
# ✅ Build successful: 633kB JS bundle created
```

Phase 4: Web Server Configuration

Step 10: Nginx Installation & Initial Configuration

```
sudo dnf install -y nginx
sudo systemctl start nginx
sudo systemctl enable nginx
```

Step 11: Nginx Configuration Challenges

First Configuration Attempt:

```
server {
    listen 80;
    server_name 124.43.216.136;

    location / {
        root /var/www/MYSLT-DASHBOARD/client/dist;
        try_files $uri $uri/ /index.html;
    }

    location /api/ {
        proxy_pass <http://localhost:5001/>;
    }
}
```

❌ **Issue:** Default Rocky Linux test page shown instead of React app

Debugging Process:

```
# Checked active configuration
sudo nginx -T | grep "default_server"
# Found: Default server block in /etc/nginx/nginx.conf overriding our config

# Tested different port
sudo nano /etc/nginx/conf.d/myslt.conf
# Changed to listen 8080 instead of 80
```

❌ **New Issue:** 403 Forbidden and 502 Bad Gateway errors

Permission Issues Identified:

1. **403 Forbidden:** Nginx couldn't access frontend files
2. **502 Bad Gateway:** Nginx couldn't connect to backend on port 5001

Root Cause: SELinux security restrictions

Solutions Applied:

```
# Fix file permissions
sudo chmod -R 755 /var/www/MYSLT-DASHBOARD/client/dist

# Fix SELinux context for web files
sudo chcon -Rt httpd_sys_content_t /var/www/MYSLT-DASHBOARD/client/dist

# Allow Nginx network connections
sudo setsebool -P httpd_can_network_connect 1

# Restart Nginx
sudo systemctl restart nginx
```

✅ **Result:** Frontend and API working on port 8080

Step 12: Switch Back to Port 80

Challenge: Default Nginx config conflicting
Solution: Modified main `/etc/nginx/nginx.conf`:

```
# Changed default server from port 80 to 8080
server {
    listen       8080;       # Changed from 80
    listen       [::]:8080;  # Changed from ::80
    # ... rest of config
}
```

✅ **Result:** Application working on standard HTTP port 80

Phase 5: External Access & SSL

Step 13: pfSense Port Forwarding Setup

Initial Rules:

1. **Port 9120** → **22** (SSH management)
2. **Port 9121** → **80** (HTTP application)
3. **Port 80** → **80** (Temporary for Let's Encrypt)

Step 14: Domain Configuration

Domain Assigned: `dpdlab1.slt.lk` **DNS Configuration:** A record pointing to `124.43.216.136`

Step 15: SSL Certificate Challenges

First Certbot Attempt:

```
sudo dnf install -y epel-release
sudo dnf install -y certbot python3-certbot-nginx
sudo certbot --nginx -d dpdlab1.slt.lk
```

❌ **Error:** "Certbot failed to authenticate some domains"
❌ **Detail:** Let's Encrypt couldn't reach server on port 80

Root Cause: pfSense only forwarding port 9121, not port 80

Temporary Solution:

1. Added **temporary port forward**: 80 → 80 in pfSense
2. Retried Certbot: ✅ **Certificate issued successfully**

New Issue: Certbot couldn't auto-configure Nginx

```
# Error: "Could not automatically find a matching server bloc
k for dpdlab1.slt.lk"
```

Manual Certificate Installation:

```
sudo certbot install --cert-name dpdlab1.slt.lk
# ✅ SUCCESS: Certificate deployed
```

Step 16: HTTPS Port Configuration

Decision Point: Standard port 443 vs custom port
Issue: Port 443 already used by pfSense web interface
Solution: Use custom port 9122 for HTTPS

Final pfSense Rules:

1. **9120** → **22** (SSH - Keep)
2. **9121** → **80** (HTTP redirect - Keep)
3. **9122** → **443** (HTTPS application - Keep)
4. **80** → **80** (Temporary - DELETE after SSL setup)

Step 17: Nginx SSL Configuration

Challenge: Certbot configured for standard ports, not custom port 9122
Solution: Manual Nginx configuration with port specification:

```
# HTTP redirect with custom HTTPS port
server {
    listen 80;
    server_name dpdlab1.slt.lk;
    return 301 https://dpdlab1.slt.lk:9122$request_uri;
}

# HTTPS server
server {
    listen 443 ssl;
    server_name dpdlab1.slt.lk;
    # ... SSL config from Certbot
    # ... Proxy and static file config
}
```

Phase 6: Final Testing & Cleanup

Step 18: Comprehensive Testing

```
# Test HTTP redirect
curl -I <http://dpdlab1.slt.lk:9121/>
# ✅ Returns: 301 redirect to HTTPS on port 9122

# Test HTTPS access
curl -k <https://dpdlab1.slt.lk:9122/>
# ✅ Returns: React application HTML

# Test API endpoint
curl -k <https://dpdlab1.slt.lk:9122/api/health>
# ✅ Returns: {"success":true,...}

# Test security (should fail)
curl -I <http://dpdlab1.slt.lk:9122/>
# ✅ Returns: 400 Bad Request (security working correctly)
```

Step 19: Cleanup

Removed: Temporary port 80 forward rule from pfSense
Kept:

1. Port 9120 → 22 (SSH management)
2. Port 9121 → 80 (HTTP→HTTPS redirect)
3. Port 9122 → 443 (HTTPS application)

🎯 Final Deployment Summary

Access URLs:

- 🌐 **Primary URL:** `https://dpdlab1.slt.lk:9122`
- 🔄 **Redirect URL:** `http://dpdlab1.slt.lk:9121` (auto-redirects to HTTPS)
- 🔧 **Management:** SSH via port 9120
- 📊 **API:** `https://dpdlab1.slt.lk:9122/api/...`

Technical Stack:

- **Frontend:** React/Vite served via Nginx
- **Backend:** Node.js/Express managed by PM2
- **Database:** MongoDB Atlas (cloud) with local fallback
- **Web Server:** Nginx with SSL reverse proxy
- **Security:** Let's Encrypt SSL, SELinux enforced
- **Networking:** pfSense with custom port forwarding

Key Issues Overcome:

1. ✅ MongoDB installation failures (GPG keys, AVX requirements)
2. ✅ Nginx permission issues (SELinux context)
3. ✅ Port conflicts and forwarding challenges
4. ✅ SSL certificate validation with custom ports
5. ✅ Service auto-start and process management

Lessons Learned:

1. Always check CPU compatibility for database versions
2. SELinux requires explicit permission configuration
3. Certbot requires standard port access for validation
4. Clear documentation of port mapping is essential
5. Testing both success and failure scenarios is crucial

📞 Support Information

- **SSH Access:** `ssh dpd@124.43.216.136 -p 9120`
- **Server IP:** 192.168.100.137 (internal), 124.43.216.136 (public)
- **Certificate Expiry:** March 13, 2026 (auto-renewal configured)
- **Backup Location:** `/var/www/MYSLT-DASHBOARD/`

Deployment Completed Successfully 🎉

All systems operational with production-grade security and reliability.