Hospital IoT HUB System Deployment Briefing Document

Subject: Network and Security Requirements for IoT HUB System (iothub.com)

Summary of Recommended Specifications for a Reliable and Solid Setup

To ensure the smooth and reliable operation of the IoT Hub System, the following server specifications are recommended:

- vCPUs: 4–8 (8 vCPUs recommended for high reliability and solid performance)
- RAM: 16 GB
- Storage: 100–200 GB NVMe SSD
- Network: 1 Gbps bandwidth with low latency
- Operating System: Ubuntu Server (latest LTS)

Note: The server must be provisioned and prepared by the internal IT department.

Elixion Medical GmbH will be responsible for deploying the required services and the IoT Hub components on the prepared server.

After setup, the IT team must provide server access credentials (either root or a superuser account with sudo privileges), and VPN details, if necessary, to Elixion Medical GmbH for installation and configuration.

For the device connectivity requirements please see: 2.3 IoT Device Connectivity Requirements

---

1. System Overview

We are deploying a Docker-based IoT Hub Management System (iothub.com domain) requiring **SSL encryption** for GDPR compliance. The system consists of:

- **Ubuntu Server**: Hosts Docker containers (Traefik reverse proxy, MQTT broker, Gateway Service, FHIR server).
- **Windows Clients**: Nurse/doctor stations accessing via https://*.iothub.com.
- **External IoT Devices**: Connect to the MQTT broker for real-time data transmission.

---

2. Network Requirements

2.1 DNS Configuration

- **Action**: Route all *.iothub.com requests to the Ubuntu server.

```
Record Type: A
Name: *.iothub.com
Value: [Ubuntu Server IP] (e.g., 192.168.1.100)
TTL: 3600
```

Domain names using by IOT Hub System:
iothub.com
api.iothub.com
admin.iothub.com
keycloak.iothub.com
traefik.iothub.com
grafana.iothub.com
hapi-fhir.iothub.com

Recommendation: *Instead of using a different domain, which can introduce unnecessary complexity and configuration overhead, it is best to create a DNS record for* ***iothub.com*** *pointing to the Docker host's IP address. This approach allows you to obtain and use a single wildcard certificate (*.iothub.com) for all subdomains, greatly simplifying certificate management.*

- Hospital DNS must resolve iothub.com and *.iothub.com to the IoT Hub server IP.

2.2 Network Diagram

[External IoT Devices] → (Port 8883) → [Hospital Firewall] → [Ubuntu Server]
↓
[Windows Clients] ← (HTTPS) ← [Traefik] ← (Internal Services: Gateway:8181, HAPI-FHIR, etc.)

2.3 IoT Device Connectivity Requirements

To ensure proper connectivity between IoT devices and the IoT Hub system via MQTT, the following prerequisites must be fulfilled:

Wi-Fi Credentials:
The hospital IT department must provide the necessary Wi-Fi credentials / login methods and certificates that the IoT devices will connect to.
MQTT Connectivity:
Devices will connect to the MQTT broker hosted on the IoT Hub server using port 8883 (TLS).
Therefore, **the server must be accessible from the same network that the devices connect to**.

MAC Address Registration (if required):
If MAC address whitelisting is enforced on the hospital Wi-Fi, a list of device MAC addresses will be shared with the IT department ahead of time.

Network Access:
The IoT Hub server must be reachable from the device network. Please ensure proper routing and firewall configuration to allow inbound MQTT connections on port 8883 (MQTT over TLS).

---

3. Firewall Rules

Ensure the Ubuntu server is accessible with these rules:

Direction	Port	Protocol	Source	Purpose
Inbound	443	TCP	Internal Networks	HTTPS Traffic (Web Apps)
Inbound	8883	TCP	IoT Devices (External)	MQTT over TLS
Outbound	Any	TCP/UDP	N/A	Block except critical updates & installations
Inbound	2575	TCP	Internal Networks	HL7 Receiver
Outbound	257X	TCP	Internal Networks	HL7 Exporter

---

4. Security Configuration

4.1 SSL Certificate Authority (CA) Deployment

- **Purpose**: Avoid browser warnings for internal services.
- **Steps**:
    1. **Locate CA Certificate**: Provided as rootCA.pem (generated on Ubuntu server).
    2. **Deploy via Group Policy (GPO)**:
        - **Path**: Computer Configuration → Policies → Windows Settings → Security Settings → Public Key Policies → Trusted Root Certification Authorities
        - **Action**: Import rootCA.pem.
    3. **Verify Installation**:

```
certutil -verifystore Root | findstr "mkcert"
# Should return CA details
```

- **Fallback Script**: For non-GPO devices:

```
certutil -addstore -f Root "\\path\to\rootCA.pem"
```

4.2 MQTT Security

- **IoT Device Connections**:
    - Uses mqtt://[Docker Host]:1883 for observation data traffic. DNS definition is not necessary.

4.3 Gateway Service

- **Port**: 8181 (internal use only, no external exposure). Processes IoT data, sends and receives messages from MQTT.
- **Port:** 443 for admin.iothub.com
- **Port:** 2575 for HL7 ORM messages import
- **Port:** 257x for HL7 ORU messages export

---

5. Compliance & Maintenance

5.1 GDPR Compliance

- All web traffic uses HTTPS (TLS 1.2+).

5.2 Certificate Renewal

- **Auto-Renewal**: Every 825 days (managed via Ubuntu server).
- **Manual Renewal**:

```
mkcert -force-renew "*.iothub.com"
docker restart traefik mosquitto
```

5.3 Monitoring

- **Logs**:
    - All services : Accessible via Grafana (https://grafana.iothub.com).

- **Alerts**: Configured for:
    - Unauthorized access attempts.
    - Certificate expiration (30-day advance notice).

---

6. IT Action Summary

1. **DNS**: Add wildcard record for *.iothub.com → Ubuntu IP.
2. **Firewall**: Open ports 443, 8883, 2575, 275x
3. **CA Deployment**: Distribute rootCA.pem via GPO/script.

---

7. Attachments

1. Network topology diagram.

---

8. Support Contacts

- **Deployment Issues**: Ulas Arican / +491747643106 / arican@elixionmedical.com

---

Thank you for your collaboration!

Below you can find “Potential Deployment Issues & Troubleshooting Guide”

Potential Deployment Issues & Troubleshooting Guide

1. Introduction

This document outlines common problems that might occur while preparing the IoT System (as described in the main deployment briefing) and provides guidance on how to investigate and resolve them. While the main briefing details the system's configuration, this guide focuses on **troubleshooting and problem-solving**.

2. Potential Issues

2.1 DNS and Wildcard Configuration

Symptoms

- Endpoints like https://*.iothub.com fail to resolve.
- Clients receive DNS errors (e.g., "DNS_PROBE_FINISHED_NXDOMAIN").

Possible Causes

1. Incorrect wildcard record (*.iothub.com) or missing DNS entry.
2. Propagation delays in DNS updates or local DNS cache issues.
3. Misconfigured DHCP settings if using Ubuntu Server as DNS.

Investigation

- Use nslookup api.iothub.com (Windows) or dig dashboard.iothub.com (Linux) to verify DNS resolution.
- Check DNS settings in your DNS management console or DHCP server logs.

Resolution

1. Correct the DNS record for *.iothub.com.
2. Flush DNS cache on client machines (ipconfig /flushdns on Windows).
3. If using the Ubuntu server as DNS, ensure dnsmasq or equivalent is configured correctly.

2.2 SSL Certificate Authority (CA) Distribution

Symptoms

- Browsers show “Not Secure” or “Certificate not trusted” warnings.
- MQTT over TLS fails with certificate validation errors.

Possible Causes

1. CA certificate (rootCA.pem) was not imported into the client’s trusted root store.
2. Using an outdated certificate or incorrect domain name (SAN mismatch).
3. Group Policy Object (GPO) not propagating properly.

Investigation

- Check the certificate validity and subject alternative names (SAN) via a browser or openssl x509 -in cert.pem -text -noout.
- Confirm that the CA certificate appears in the Windows “Trusted Root Certification Authorities”.
- Review Windows Event Logs for GPO application errors.

Resolution

1. Re-import or re-deploy rootCA.pem using the correct GPO path:
   - Computer Configuration → Policies → Windows Settings → Security Settings → Public Key Policies → Trusted Root Certification Authorities
2. For standalone devices, use the certutil -addstore -f Root "rootCA.pem" command.
3. Regenerate certificates with proper SAN entries if domain mismatches exist.

2.3 Firewall and Port Forwarding Issues

Symptoms

- Services (HTTP/HTTPS/MQTT) are unreachable.
- Connection timeouts when accessing iothub.com or MQTT broker.

Possible Causes

1. Firewall rules not matching the intended source/destination or protocol/port settings.

2. NAT configuration blocking external IoT device traffic to the MQTT broker.

3. Overly restrictive outbound rules preventing Docker containers from updating or communicating.

Investigation

- Test port reachability (e.g., telnet iothub.com 443 or nc -vz iothub.com 1883).
- Check firewall logs and NAT rules for dropped traffic.
- Examine Docker container logs for connection failures.

Resolution

1. Update firewall rules to allow inbound ports (443, 8883, 2575) from the correct source IPs.
2. Ensure NAT is configured so external devices can reach 8883 on the Ubuntu server.
3. Permit necessary outbound traffic for system updates or container image pulls.

2.4 Docker & Container Configuration

Symptoms

- Containers fail to start or remain in a crash loop.
- Services respond on unexpected ports or addresses.

Possible Causes

1. Incorrect Docker Compose file referencing wrong ports or volumes.
2. Conflict between container ports and host ports.
3. Container logs indicate missing dependencies or environment variables.

Investigation

- Use docker ps -a and docker logs [container_name] to review container status.
- Check environment variables in .env files or Docker Compose configuration.
- Inspect port mappings in docker-compose.yml for conflicts.

Resolution

1. Correct any port collisions by adjusting host/container mappings (e.g., - "443:443").

2. Verify that each container's environment variables (e.g., MQTT_HOST, EXS_PORT) match the actual configuration.

3. Restart containers after changes: docker-compose down && docker-compose up -d.

2.5 Gateway & FHIR Integration

Symptoms

- Missing or incomplete data in the FHIR server.
- Internal REST calls to Gateway Service (http://exs:8181) fail.

Possible Causes

1. Gateway Service misconfiguration (wrong FHIR endpoint or missing authentication).
2. FHIR server not started or container is down.
3. Network segmentation blocking port 8181.

Investigation

- Review Gateway service logs to see if connections to FHIR are successful.
- Check the Docker service logs for the FHIR container.
- Confirm firewall rules or internal network routes allow port 8181 traffic.

Resolution

1. Update the Gateway configuration with the correct FHIR base URL and credentials.
2. Ensure the FHIR container is running (docker-compose ps).
3. Allow internal traffic on port 8181 and test connectivity (e.g., curl http://exs:8181/debug).

2.6 GPO Deployment & Windows Client Issues

Symptoms

- Windows clients do not receive the new CA certificate or do not auto-configure proxy settings for the new domain.
- GPO changes do not apply or appear delayed.

Possible Causes

1. The GPO is not linked to the correct Organizational Unit (OU).
2. Clients are not connected to the domain or are in the wrong OU.
3. GPO replication issues within Active Directory.

Investigation

- Run gpupdate /force on the client to manually trigger GPO updates.
- Check gpresult /r on Windows clients to verify which GPOs are applied.
- Inspect Active Directory logs for replication errors.

Resolution

1. Ensure the GPO containing the CA certificate is linked to the correct OU.
2. Wait for or force replication across all domain controllers.
3. Verify that the client machines are in the correct OU and domain.

3. General Troubleshooting & Resolution Process

1. **Identify** the Problem: Gather logs from containers, firewall, DNS, and operating systems.
2. **Analyze** the Logs: Look for error messages, timestamps, and relevant stack traces.
3. **Hypothesize** the Root Cause: Based on symptoms and log data, determine likely issues (configuration, firewall, DNS, etc.).
4. **Test & Validate**: Make incremental configuration changes, then retest to confirm whether the issue is resolved.
5. **Document** the Fix: Note what was changed, how, and why it resolved the issue to maintain an audit trail.

4. Summary

By anticipating the most common deployment and configuration problems—ranging from DNS resolution to SSL certificates, firewall settings, Docker container issues, and GPO distribution—teams can more quickly identify, troubleshoot, and resolve problems. Adopting a structured troubleshooting process ensures minimal downtime and preserves GDPR compliance throughout the IoT system's lifecycle.

Key Takeaways:

- Always verify DNS and certificate configurations first.
- Tighten firewall rules but keep essential ports open for critical services.
- Monitor logs in real-time (Traefik, Mosquitto, FHIR, Gateway etc.) to detect potential issues early.
- Maintain documentation of changes for compliance and future reference.