IoT-Hub Deployment Briefing

1. Executive Summary

We are deploying a Docker-based IoT-System (iothub.com domain) requiring TLS encryption to protect data in transit (GDPR-related security requirements). The Niviu system consists of:

- **IoT-Hub**: Ubuntu server (used as synonym for IoT-Hub) which hosts Docker containers (Traefik reverse proxy, MQTT broker, Gateway Service).
- **IoT-Devices**: Connect to the MQTT broker of the Ubuntu server via MQTT over TLS on port 8883 for sending measurement from the bedside.
- **Windows Clients**: Nurses/doctors can access measurement data via HTTPS on Windows clients accessing the IoT-Hub via https://iothub.com and https://*.iothub.com.

1.1 Access models

- **Option A (Recommended - Hospital Proxy/WAF):** Hospital DNS points iothub.com and *.iothub.com to the hospital Proxy/WAF. The Proxy/WAF terminates client TLS with a hospital-managed certificate and forwards traffic to the Ubuntu server over HTTPS (re-encryption). No certificates need to be installed on Windows clients.
- **Option B (Direct Access):**Hospital DNS points directly to the Ubuntu server Windows clients must trust a local CA certificate distributed via GPO (or script).

1.2 Firewall requirements:

Inbound:	443 (web), 8883 (MQTT/TLS), 2575 (HL7 receiver)
Outbound:	257x (HL7 exporter), 443 (approved updates/registries)

1.3 Security:

End-to-end encryption is enforced (HTTPS + MQTT over TLS). IoT devices validate the MQTT broker via pinned trust (embedded CA/certificate/fingerprint).

2. Recommended Specifications

The following server (virtual machine) specifications are recommended:

vCPUs:	4–8 (8 vCPUs recommended)
RAM:	8 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 the preferred method for remote access to Elixion Medical GmbH for installation and configuration. Alternatively, the IoT-Hub deployment can be done together under supervision.

3. Network Requirements

3.1 DNS Configuration

Route iothub.com and *.iothub.com to the appropriate target (see Option A/Option B below). For Option A, DNS points to the proxy/WAF; for Option B, DNS points directly to the Ubuntu server.

Option A (Hospital Proxy/WAF in front):

- iothub.com A → **[Hospital Proxy/WAF IP or VIP]**
- *.iothub.com A → **[Hospital Proxy/WAF IP or VIP]**

Note: Depending on hospital DNS policy, the record can be implemented as A/AAAA (VIP/IP) or (if supported) CNAME to an internal proxy hostname. Windows clients must be able to resolve iothub.com / *.iothub.com via hospital DNS (internal split-DNS), since access is performed using these hostnames.

Option B (Direct access):

- iothub.com A → **[Ubuntu Server IP]**
- *.iothub.com A → **[Ubuntu Server IP]**

Domain names used by IoT-Hub:

- iothub.com
- api.iothub.com
- admin.iothub.com
- keycloak.iothub.com
- traefik.iothub.com
- grafana.iothub.com

Recommendation: Use a single wildcard domain (*.iothub.com) to simplify routing and certificate handling. DNS should point to the **hospital proxy/WAF (Option A)** or directly to the **Docker host (Option B)**.

3.2 Network Diagram

[IoT Devices (Hospital Network)] → (Port 8883) → [Hospital Firewall] → [Ubuntu Server]

Client connection (Option A – hospital proxy termination):

[Windows Clients] → (HTTPS) → [Hospital Proxy/WAF] → (HTTPS) → [Traefik on Ubuntu Server] → (Internal Services)

Note (Option A): Windows clients resolve *.iothub.com to the **Proxy/WAF**, not directly to the Ubuntu server. The Proxy/WAF forwards traffic to Traefik on the Ubuntu server.

Client connection (Option B – direct access):

[Windows Clients] → (HTTPS) → [Traefik on Ubuntu Server] → (Internal Services)

3.3 IoT Device Connectivity Requirements

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

- **Wi-Fi Credentials / Certificates**: 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:** 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).

4. Firewall Rules

Ensure the Ubuntu server is accessible with these rules:

Direction	Port	Protocol	Source/Destination	Purpose
Inbound	443	TCP	Internal Networks	HTTPS Traffic (Web Apps)
Inbound	8883	TCP	IoT Devices (Hospital Network)	MQTT over TLS
Outbound	443	TCP	Approved update registries / OS mirrors	OS updates + container image pulls (and other explicitly approved endpoints)
Inbound	2575	TCP	Internal Networks	HL7 Receiver
Outbound	257x (configurable)	TCP	Internal Networks / PDMS	HL7 Exporter

5. Security Configuration

5.1 SSL Certificate Options (Trusted HTTPS without browser warnings)

Option A (Recommended): Hospital-managed TLS via Proxy / Reverse Proxy (No client certificate distribution)

Use this option if the hospital operates an HTTPS proxy / reverse proxy (WAF / load balancer) and wants to avoid installing CA certificates on every client device.

1. DNS: Hospital DNS resolves iothub.com and *.iothub.com to the **hospital proxy/WAF** (VIP/IP). The proxy/WAF forwards traffic to the Ubuntu server.
2. Client-facing HTTPS: The hospital proxy terminates TLS for iothub.com and required subdomains using a hospital-managed certificate (e.g., wildcard *.iothub.com). The certificate may be issued by the hospital enterprise PKI (internal CA) or a public CA, depending on hospital policy.
3. Upstream to IoT-Hub: The proxy forwards requests to the Ubuntu server as the upstream target **over HTTPS** (TLS re-encryption).
4. Upstream HTTPS trust: The proxy **is configured to trust** the upstream certificate presented by the IoT-Hub server (proxy-side trust store / validation), according to hospital policy. No certificate deployment is

required on Windows clients; any upstream trust is handled on the proxy/WAF only.

5. No client changes: No certificate installation is required on client devices.

Option B: Local CA (mkcert) + Client Trust Distribution

Use this option if clients access the IoT-Hub directly and you can distribute a local CA to all clients.

- **Locate CA Certificate**: Provided as rootCA.pem (generated on Ubuntu server).
- **Deploy via Group Policy (GPO)**:
    1. **Path**: Computer Configuration → Policies → Windows Settings → Security Settings → Public Key Policies → Trusted Root Certification Authorities
    2. **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"
  ```

5.2 MQTT Security

- IoT devices connect to the MQTT broker on the IoT Hub server using **MQTT over TLS**.
- **Devices connect via IP address.** TLS is used for encryption in transit.
- **Certificate validation is enforced on the device side via a pinned trust anchor** (e.g., embedded CA certificate or pinned broker certificate / fingerprint).
- Since devices connect via IP, **hostname/SNI-based validation is not required**.
- Protocol/Port: mqtts://<iot-hub-ip>:8883 (TLS)
- DNS: Not required for IoT devices. Devices connect to the MQTT broker via IP address only.

5.3 Gateway Service

- **Port:** 443 for admin.iothub.com
- **Port:** 2575 for HL7 ORM messages import
- **Port:** 257x (configurable) for HL7 ORU messages export

6. Compliance & Maintenance

6.1 GDPR Compliance

All web traffic uses HTTPS (TLS 1.2+) as a technical measure to protect data in transit (GDPR-related security requirements).

6.2 Certificate Renewal

Option A (Hospital-managed TLS termination)

- Client-facing certificate renewal: Managed by the hospital proxy/PKI (certificate for iothub.com / *.iothub.com).
- Upstream (proxy → Ubuntu) connection:
  - If configured as HTTPS, certificate renewal/trust is managed according to hospital policy (proxy-side trust store / validation).

Option B (Local CA / mkcert + client trust distribution)

- **Manual renewal (on Ubuntu server):**

  ```
  mkcert -force-renew "*.iothub.com"
  ```

- **Restart services:** Restart performed by Elixion during maintenance.
- **Auto-renewal:** According to local operational policy (recommended monitoring/alerting for expiration).

6.3 Monitoring

- **Logs**:
  - All services : Accessible via Grafana (https://grafana.iothub.com).
- **Alerts** configured for:
  - Unauthorized access attempts.
  - Certificate expiration (30-day advance notice).

7. IT Action Summary

1. **DNS:** Add wildcard record for *.iothub.com → Hospital Proxy/WAF VIP/IP (Option A) or → Ubuntu IP (Option B).
2. **Firewall**: Open ports 443, 8883, 2575, 257x (configurable)
3. **HTTPS Trust (choose one)**

Option A (Recommended): Configure hospital proxy with a hospital-managed certificate for iothub.com / *.iothub.com and reverse-proxy to the Ubuntu server (no client certificate distribution).
Option B: Distribute rootCA.pem via GPO/script to all Windows clients.

8. Attachments

- Network topology diagram.

9. Support Contact

Ulas Arican
+49 (0) 174 7643106
arican@elixionmedical.com

10. Troubleshooting

10.1 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

- Incorrect wildcard record (*.iothub.com) or missing DNS entry.
- Propagation delays in DNS updates or local DNS cache issues.

Investigation

- Use nslookup api.iothub.com (Windows) or dig api.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 an internal DNS resolver, ensure wildcard records are configured correctly.

10.2 SSL / Certificate Trust Issues (Option A vs Option B)

Before investigating, confirm whether the deployment uses Option A (hospital-managed TLS via proxy) or Option B (local CA + client distribution).

If Option A:

Symptoms

- Browsers show certificate warnings even though hospital certificate is expected.
- Only some subdomains work; others show name mismatch.

Possible Causes

- Proxy is not terminating TLS for all required subdomains (missing SAN/wildcard coverage).
- Proxy forwards to wrong upstream host or does not preserve SNI/Host header as required.
- Upstream HTTPS is enabled but the proxy does not trust the upstream certificate/CA.

Resolution

1. Ensure the hospital certificate covers iothub.com and all required subdomains (prefer wildcard *.iothub.com).
2. Verify proxy routes for listed subdomains (api/admin/keycloak/traefik/grafana).
3. If upstream uses HTTPS, import/trust the upstream CA/certificate on the proxy (not on clients) or configure upstream trust according to policy.

If Option B:

Symptoms

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

Possible Causes

- CA certificate (rootCA.pem) was not imported in the client’s trusted root store.
- Using an outdated certificate or incorrect domain name (SAN mismatch).
- 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.

10.3 Firewall and Port Forwarding Issues

Symptoms

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

Possible Causes

- Firewall rules not matching the intended source/destination or protocol/port settings.
- Routing / firewall rules blocking traffic from the device network to the IoT Hub server (port 8883).
- Overly restrictive outbound rules preventing Docker containers from updating or communicating.

Investigation

- Test port reachability (e.g., telnet iothub.com 443 or nc -vz <iot-hub-ip> 8883).
- Check firewall logs and routing 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 routing/firewall rules allow the device network to reach the IoT Hub server on port 8883
3. Permit necessary outbound traffic for system updates or container image pulls.

10.4 Docker & Container Configuration

Symptoms

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

Possible Causes

- Incorrect Docker Compose file referencing wrong ports or volumes.
- Conflict between container ports and host ports.
- 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.

10.5 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

- The GPO is not linked to the correct Organizational Unit (OU).
- Clients are not connected to the domain or are in the wrong OU.
- 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.

10.6 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, Gateway etc.) to detect potential issues early.
- Maintain documentation of changes for compliance and future reference.

11. Version History

Version	Effective date	Author	Change description
02	02.02.2026	Ulas Arican	Option A/B DNS routing added. HTTPS end-to-end to Traefik clarified. Firewall + MQTT TLS updated.
01	08.01.2026	Ulas Arican	First version