

The current version is # ident "@ (#) \$Format:Food:docs/zabbix\_monitoring.md:Lange  
François:lanfr144@school.lu:2026/06/11 08:26:59:Lange  
François:lanfr144@school.lu:2026/06/11  
08:26:59:1701828b122e0c319e59134ca6511a42ecad9297:: \$"

# Zabbix Telemetry & Monitoring Guide

## Overview

The Local Food AI project enforces strict DevSecOps observability by streaming live hardware and database telemetry metrics to an external Zabbix server (192.168.130.170:8081).

[!IMPORTANT] Offline Local Fallback: If the network to the external server is down or unavailable, the Zabbix monitoring dashboard is fully functional and accessible locally at `http://localhost:8081` when running the local Docker Compose stack.

## Accessing the Dashboard

1. Open your browser and navigate to `http://192.168.130.170:8081` (or `http://localhost:8081` if offline/local).
2. Log in using your Zabbix credentials (default: Admin / zabbix).
3. On the left sidebar, click Monitoring > Dashboards.
4. Select the Food AI RAG Telemetry (Live) dashboard.

## Key Metrics Monitored

The dashboard automatically queries the SNMP daemons running inside the Docker containers to monitor:

- Memory Consumption: Evaluates the massive RAM usage required by the Ollama Llama3.2:3B LLM during clinical evaluations.
- CPU Spikes: Identifies processing bottlenecks during the 3GB OpenFoodFacts MATCH AGAINST queries.
- Database Row Count Check: Displays the real-time record count of `food_db.products_core` to monitor the background CSV ingestion progress.

## Verifying Alerts

1. Click Monitoring > Problems.
2. If `snmpd` inside a container crashes or is unreachable, Zabbix will trigger an Agent Unreachable High-Severity Alert.
3. If the Database Server container crashes, Zabbix will trigger an alert via the Application Python `snmp_notifier.py` wrapper which sends asynchronous trap payloads indicating critical RAG failures.