🌍 EnviroHealth-Monitor: Technical Project Report

A Distributed Real-Time Pipeline for Urban Environmental Analytics

Date: January 2026

Institution: MSc Computer Science & Data Science @ ESILV

Project Demo: https://youtu.be/IVGqxV3CXvA

👥 The Team

The following team members contributed to the implementation and demonstration of this distributed system:

- **Hussnain Amanat Ali: Project Lead & Spark Logic**
- **Muhammad Irfan: Infrastructure & YARN Management**
- **Ayman Berri: Monitoring & Dashboard Visualization**

---

1. Executive Summary

EnviroHealth-Monitor addresses the demand for real-time monitoring of urban environmental conditions in metropolitan areas like Paris. By moving away from traditional high-latency batch processing, this project delivers an end-to-end streaming architecture with sub-5-second latency. The system provides a 360-degree view of both Infrastructure Health and Environmental Analytics.

---

2. Distributed Infrastructure

Unlike standard local deployments, this project utilizes a professional cloud-based distributed architecture:

- **Cluster Scale: A 6-node Hadoop cluster (1 Master, 5 Workers) deployed on Microsoft Azure.**
- **Hardware Capacity: Total cluster power of 48 GB RAM and 48 vCores (8GB/8-vCores per node).**
- **Resource Orchestration: Managed via Apache YARN, ensuring all nodes maintain a "RUNNING" state during high-compute Spark tasks.**

---

3. The Data Pipeline

The system is designed as a decoupled, four-stage distributed pipeline to ensure high availability and fault tolerance.

A. Ingestion Layer

Real-time data is ingested from the OpenAQ and OpenWeather APIs. Python-based producers simulate IoT sensors by publishing raw JSON messages to dedicated Kafka topics.

B. Processing Layer (Apache Spark)

Apache Spark Structured Streaming serves as the core processing engine. It performs:

- **Strict Schema Enforcement: Transforming raw JSON into typed DataFrames.**
- **Windowed Aggregations: Calculating real-time pollution averages (PM2.5) every few seconds.**
- **Correlation: Joining weather and air quality streams to identify health risk patterns.**

C. Storage & Data Lake

- **Time-Series Storage: Processed metrics are written to InfluxDB for temporal analysis.**
- **Cold Storage: Raw and processed data is permanently saved in HDFS, partitioned by year and month for future batch auditing.**

---

4. Monitoring & Visualization (TIG Stack)

The project implements the TIG Stack (Telegraf, InfluxDB, Grafana) to monitor system and application performance simultaneously:

- **Infrastructure Monitoring: Telegraf agents collect real-time RAM usage from the Azure VMs, visualized in a "Usage Graph" to prevent cluster bottlenecks.**
- **Functional Visualization: Grafana dashboards display live "Pollution Deep-Dives" and "Worker Audits" (Batch Count: 8.11), proving pipeline integrity.**

---

5. Technical Challenges Solved

1. **OS Integration: Successfully deployed the full stack on Ubuntu 24.04 LTS, overcoming modern repository and GPG key compatibility issues.**

2. **Distributed Consistency: Synchronized configuration files (yarn-site.xml, core-site.xml) across 6 nodes to ensure seamless task distribution.**
3. **Real-time Synchronization: Coordinated separate API streams into a unified Kafka-Spark flow for multi-dimensional analysis.**

---

6. Conclusion

EnviroHealth-Monitor successfully demonstrates a modern distributed Big Data stack. The system achieves low latency and high reliability, proving that cloud-distributed clusters can provide vital real-time environmental insights while maintaining robust infrastructure monitoring.