Hydro Power Plant SCADA - Proof of Concept

Digital Twin and predictive monitoring for a small hydro plant with two Francis turbine units

Technical summary - Focus: what was demonstrated and the role and benefits of CrateDB

> This Proof of Concept simulates a complete small hydro power plant, ingests its real-time sensor telemetry, visualizes the live process as a digital twin, raises operational alarms, and applies machine learning for anomaly detection and predictive maintenance. CrateDB serves as the central time-series and analytics datastore at the heart of the system.

1. What the PoC Demonstrates

The system models a realistic hydro plant and exercises a full industrial IoT / SCADA data lifecycle end to end:

- **Full sensor simulation -** 83 tags across two Francis turbine units and common plant systems: electronic, mechanical, PLC/digital and calculated signals (turbine speed, flow, head, bearing temperatures, shaft vibration, cavitation, generator power/voltage/efficiency, dam levels, grid frequency, transformer condition, safety sensors).
- **Real-time ingestion -** continuous high-frequency telemetry streamed into the datastore.
- **Live SCADA and Digital Twin -** animated process diagram (reservoir, penstock, turbines, generators, transformer, grid), live gauges and trend charts, all driven by the data.
- **Alarm engine -** multi-level limit checking (LO_LO / LO / HI / HI_HI and digital states) with multilingual messages and acknowledgement, persisted for audit.
- **Anomaly detection (AI) -** unsupervised models flag abnormal operating conditions per unit and identify the most-deviating sensor.
- **Predictive maintenance (RUL) -** per-component health index and Remaining Useful Life estimate derived from degradation trends.
- **What-if fault injection -** operators inject sensor faults or trip a unit and watch alarms, anomalies and health react in real time.

2. Why a Time-Series / Analytics Database Was Needed

An industrial monitoring system like this places several simultaneous demands on its datastore. It must:

- absorb a continuous high-rate stream of sensor readings without back-pressure;
- retain long history for trends, audits and model training;
- answer both what is happening now (latest values) and how it behaved over time (aggregations) with low latency;
- store mixed data - numeric telemetry, alarm events and semi-structured AI output together;
- be queryable with standard SQL by the backend, the AI services and dashboards alike.

CrateDB was selected to meet all of these needs with a single technology.

3. CrateDB in This System

CrateDB is the single source of stored truth. Three concerns live in dedicated tables:

Table	Purpose	Notable feature used
telemetry	All sensor readings over time (bulk of the data)	Time-based partitioning by month + sharding
alarms	Alarm events with multilingual messages, values, limits, ack state	Primary key + UPSERT for set/clear/ack lifecycle

anomalies / health	AI anomaly scores and per-component health / RUL	Dynamic OBJECT columns for flexible AI detail

4. Key CrateDB Benefits Realized

4.1 Built for time-series and IoT scale

CrateDB ingests high-velocity sensor data continuously while remaining queryable. Automatic monthly partitioning and sharding keep both writes and time-range queries fast as history grows - exactly the access pattern of a SCADA historian, with no manual archiving logic in the application.

4.2 Standard SQL - one query language for everyone

The backend API, the Python AI/ML services and the analytics dashboard all query CrateDB with familiar ANSI SQL over its HTTP and PostgreSQL-wire interfaces. There is no proprietary query language to learn, and existing SQL/PostgreSQL tooling connects directly.

4.3 Real-time reads alongside heavy writes

CrateDB serves the latest-value-per-tag snapshot and time-windowed aggregations (averages, min/max, downsampling for charts) concurrently with the constant ingest stream. The same store powers both the live SCADA view and historical trend analysis.

4.4 Powerful analytical SQL

The PoC relies on CrateDB analytics directly in the database: time bucketing (date_bin, date_trunc) to align signals for ML features, and window functions (ROW_NUMBER() OVER (PARTITION BY ...)) to retrieve the latest record per component. Computation happens close to the data, not in the application.

4.5 Flexible schema for mixed and AI data

Alongside strictly-typed numeric columns, CrateDB's dynamic OBJECT type stores semi-structured AI output (feature deviations, model details) without rigid migrations - so numeric telemetry and evolving machine-learning results coexist in one database.

4.6 Distributed by design and operationally simple

CrateDB is a distributed, horizontally-scalable database that runs as a single node for this PoC but scales out to a cluster unchanged - the same schema and queries would handle far more turbines, plants and sensors. Its built-in admin web UI for ad-hoc SQL and monitoring sped up development and validation.

4.7 One datastore instead of several

Crucially, CrateDB removed the need to bolt together separate systems for time-series storage, relational/event data and analytical queries. A single database covers ingest, live reads, historical analytics, alarm records and AI results - reducing moving parts, operational overhead and integration risk for the PoC and any future production build.

5. Summary

> The PoC proves that a complete hydro-plant monitoring stack - simulation, real-time SCADA, alarming, anomaly detection and predictive maintenance - can be built around CrateDB as its central data platform. CrateDB delivered IoT-scale time-series ingestion, standard SQL for every consumer, simultaneous real-time and historical querying, in-database analytics, flexible storage for AI output, and linear scalability - all from a single, operationally simple datastore.