influxdb™

How Enprove Built a Monitoring and Energy Analytics SaaS Platform on InfluxDB

Enprove

How Enprove Built a Monitoring and Energy Analytics SaaS Platform on InfluxDB

Company in brief

Enprove is a leading energy consultancy that helps energy-intensive industries transition to greener solutions. Through audits, expert advice, and their innovative SaaS platform, Enelyzer, Enprove drives sustainable change. More than just a platform, Enelyzer embodies visionary thinking, deep energy expertise, and a passionate commitment to a better future.

Technologies used:

InfluxDB Cloud Dedicated, AWS, Telegraf, Grafana, MQTT, Modbus

Case overview

An innovative SaaS platform, Enelyzer leads customers toward sustainable energy practices. As demand for real-time energy insights grew, Enelyzer engineers faced performance challenges due to their relational database backend. Enelyzer engineers integrated InfluxDB, a time series database, which improved data ingestion, querying speed, and scalability. This modernization enabled real-time insights, efficient customer onboarding, and enhanced reporting capabilities, positioning Enelyzer for continued growth and innovation.

> " The functionality and performance improvements in Enelyzer could not be possible without InfluxDB. Using InfluxDB Cloud Dedicated has been a great experience so far for us."

Abdullah Sabaa Allil, Technical Lead, Enprove

The business challenge

As industries face increasing pressure to reduce their environmental impact, adopting sustainable practices is critical for long-term success. Enelyzer provides strategic energy and sustainability insights by collecting and transforming energy consumption data from various channels into actionable insights. This helps energy-intensive industries and property managers improve both operational efficiency and sustainability.

Unlike the traditional Excel spreadsheets that are still widely used in the industry, Enelyzer offers a more interactive approach: asking targeted questions and delivering advanced insights. Key features like carbon footprint tracking and Cell Global Identity (CGI) reporting empower businesses to streamline energy management.

Enelyzer is experiencing rapid growth, with plans to expand to over 75,000 time series in the near future.This dataset will include categories such as energy usage data, cost data, energy storage metrics, operational and environmental conditions. To achieve this, they need a robust time series management system capable of handling real-time ingestion, querying, and the high cardinality of energy data. The system must also scale seamlessly with Enelyzer's growth, ensuring continued performance for both new and existing customers.

The technical challenge

Enelyzer aims to provide customers with real-time or near real-time data querying and analytical capabilities. To deliver insights, reports, and visualizations without delay, Enelyzer requires a robust technical infrastructure capable of ingesting, accessing, and processing data in real-time.

With this pursuit in mind, Enelyzer engineers originally built a monolithic technical architecture supported by a PostgreSQL database. As Enelyzer scaled its customer base and workload size, performance bottlenecks emerged in the monolithic system, slowing down data ingestion and querying speeds. In response, Enelyzer engineers modernized its architecture, implementing a distributed system and adding a second relational database, Microsoft SQL Server. The engineering team rewrote the backend in Scala and Rust. With two relational databases supporting different functionalities and a fully redesigned microservices architecture, Enelyzer once again delivered real-time insights and continued to scale.

However, the initial success the Enelyzer engineering team found with their new distributed architecture was short-lived. As Enelyzer's customer base continued to scale, the application encountered limitations. The technical infrastructure struggled to keep up with growth. The engineering team once again faced persistent issues with storage, performance, limited write capabilities, inadequate time series indexing and compression, inefficient querying, and scalability constraints. Despite optimization efforts, the relational database backend couldn't meet the evolving needs of Enelyzer customers.

Another factor impacting Enelyzer's performance was the volume of data stored in Microsoft SQL Server and PostgreSQL databases. With recent growth, Enelyzer's engineering teams required a new solution—one that ingests, queries, and processes data in real-time and stores massive amounts of data

without compromising performance. Beyond high storage capacity, Enelyzer's engineers needed a database that could reduce the size of customer datasets through downsampling while preserving essential insights.

The challenges faced by Enelyzer's engineers weren't due to poor architecture or inadequate technology. It all comes down to the data. With a focus on the energy sector, Enelyzer collects high cardinality time series data. Relational databases were not built to handle the requirements of time series data, which requires specialized tooling. Enelyzer's engineers decided to modernize one last time. After evaluating several time series databases, the engineering team added InfluxDB to their technical infrastructure.

The solution

In the next technology stack upgrade, Enelyzer achieved the desired performance boost. Microsoft SQL Server was replaced with InfluxDB Cloud Dedicated, which now handles all time series data, while Postgres continues to store customers' contextual data. The team also rewrote the backend, shifting from Scala to primarily Rust with a small amount of Scala., and added Telegraf for data collection. After incorporating InfluxDB and Telegraf, Enelyzer could once again deliver real-time querying and analytics to customers at a much greater scale.

Despite many changes, Enelyzer engineers attribute much of the performance gains to InfluxDB Cloud Dedicated. Unlike relational databases, which rely on indexes to optimize query performance, InfluxDB Cloud Dedicated's custom partitioning influences both query optimization and storage strategies. Enelyzer built a custom partitioning strategy that organizes data by measurement (similar to a database table) and point (individual data entry including timestamp, fields, and tags). This strategy physically isolates data from each IoT device and enhances read and write performance by reducing concurrency and minimizing resource transactions.

InfluxDB was built on top of open source software with DataFusion as its query execution engine. It was important for the Enelyzer engineering team to grow with their database partner. For the last few months, Enelyzer engineers worked alongside the InfluxDB and DataFusion teams to fit InfluxDB to their specific use case. This included fixing bugs and introducing new features, specifically for the Enelyzer platform.

Architecture

As part of the open data ecosystem, InfluxDB is compatible with many technical products and services within the ecosystem. It's highly compatible nature allows the Enelyzer engineers to deliver features faster using existing tools and utilities within the open data ecosystem.
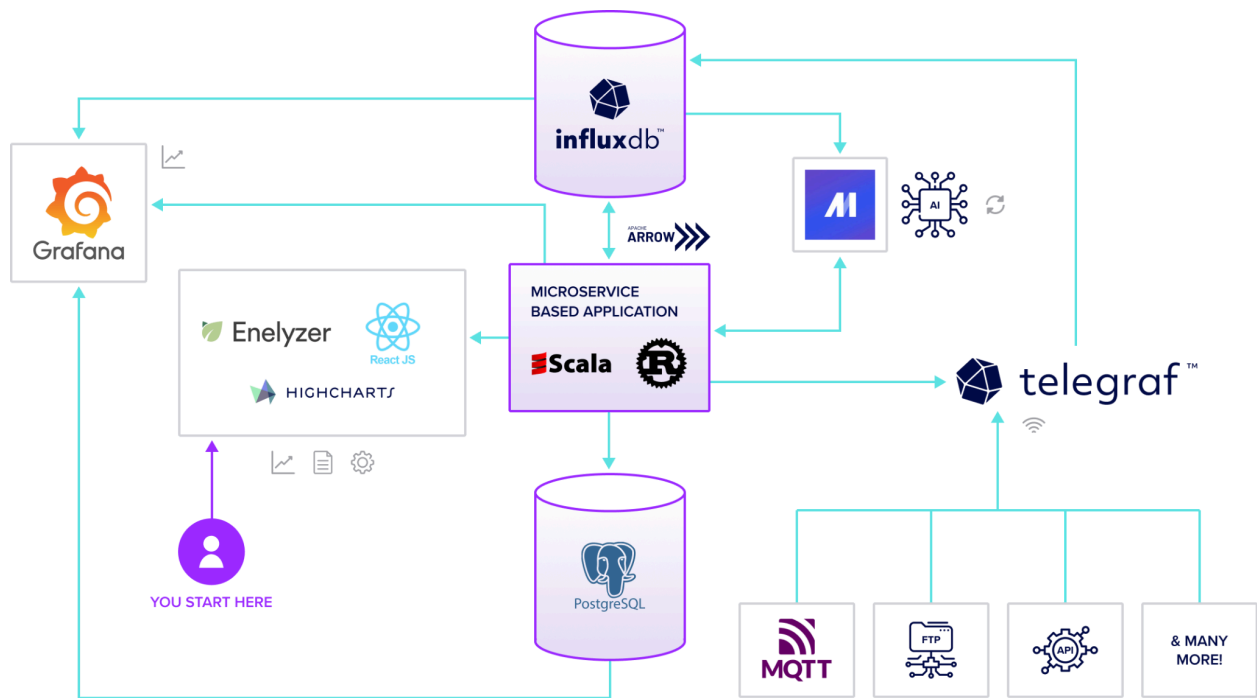
Enelyzer offers two customer-facing options for data visualization, charting, and analytics. Enelyzer's UI, built with React and Highcharts, provides intuitive charts, visualizations, and custom reports. For customers needing more advanced analysis and alerting, Enelyzer integrates directly with Grafana.

Data enters Enelyzer's ecosystem through the open source plugin agent Telegraf. Using Telegraf, Enelyzer continuously ingests data from diverse sources—such as MQTT, API, FTP, PLCs, and third-party or proprietary hardware—in real-time, avoiding vendor lock-in. Telegraf's plugins for data collection, output, and transformations enable Enelyzer to maintain a unified data pipeline regardless of input and output sources.

In its latest modernization, Enelyzer's engineering team added Mage AI to the stack. Mage AI manages task scheduling and ETL pipelines with InfluxDB, transforming incoming data as part of Enelyzer's data normalization process. They also introduced ontologies with reasoning capabilities to support machine learning models.

Enelyzer offers two customer-facing options for data visualization, charting, and analytics. Enelyzer's UI, built with React and Highcharts, provides intuitive charts, visualizations, and custom reports. For customers needing more advanced analysis and alerting, Enelyzer integrates directly with Grafana.

Results

InfluxDB's performance gains and streamlined processes translate to real benefits for Enelyzer customers. Fast querying and aggregations are essential, as customers rely on Enelyzer for timely reports and audits. Additionally, when specific alerts are set—for instance, to notify users if consumption surpasses or drops below a defined threshold—data querying needs to be swift for these alerts to serve their purpose effectively.

Integrating Telegraf into its data pipeline refined Enelyzer's customer onboarding. This process often involves handling both new real-time data and extensive historical data, both of which are crucial for identifying trends and patterns, and making accurate forecasts. In some cases, customers bring up to six years of historical data, translating to billions of data points. Telegraf collects and sends this historical data to InfluxDB quickly and efficiently, enabling a smooth, expedited onboarding experience.

Enelyzer's billing feature: before and after the modernization

Customers use the Enelyzer platform to bill tenants for energy consumption, accounting for various time-dependent parameters, like contract duration, property occupancy, and shared energy sources. Calculating these parameters determines "consumption intervals," which are permutations of all variables used to calculate energy consumption and generate reports.

Previously, Enelyzer queried Microsoft SQL Server to generate the report. After completing the calculation, Enelyzer stored the report in Postgres. To stay current, these persisted reports required recalculations with each dependency change and daily updates. This approach was slow, error-prone, and inefficient, as 70% of the report generation time was spent on querying energy data, taking around five minutes per report.

The current Enelyzer solution retrieves input data and parameters to determine consumption intervals directly from InfluxDB, eliminating the need to persist data. Querying InfluxDB enables Enelyzer to generate real-time reports, continuously updating data and eliminating dependency tracking.

Next steps

Future plans for the Enelyzer platform include several high-priority roadmap items, focusing on forecasting, advanced reporting, and auditing. Enelyzer engineers aim to enhance auditing capabilities to support a consultancy-driven part of the company. Additionally, the team plans to implement IoT edge processing with InfluxDB OSS, enabling calculations and pre-normalizations at the edge to streamline integration.

The Enelyzer team is also experimenting with Mage AI for forecasting and predictive analytics. MAGE AI's backend system powers the platform's calculations, reporting, and auditing, communicating with InfluxDB via a high-speed Arrow Flight gRPC connection rather than a traditional REST interface. Enelyzer values InfluxDB's foundation on open-source technologies like Apache DataFusion, Arrow, and Parquet, as these projects provide active communities, allowing for easy bug identification and feature tracking.

About InfluxData

InfluxData is the creator of InfluxDB, the leading time series platform. More than 1,900 customers use InfluxDB to collect, store, and analyze all time series data at any scale. Developers can query and analyze their time-stamped data in real-time to discover, interpret, and share new insights to gain a competitive edge. InfluxData is a remote-first company with a globally distributed workforce. For more information, visit www.influxdata.com.

InfluxDB documentation, downloads & guides

Get InfluxDB
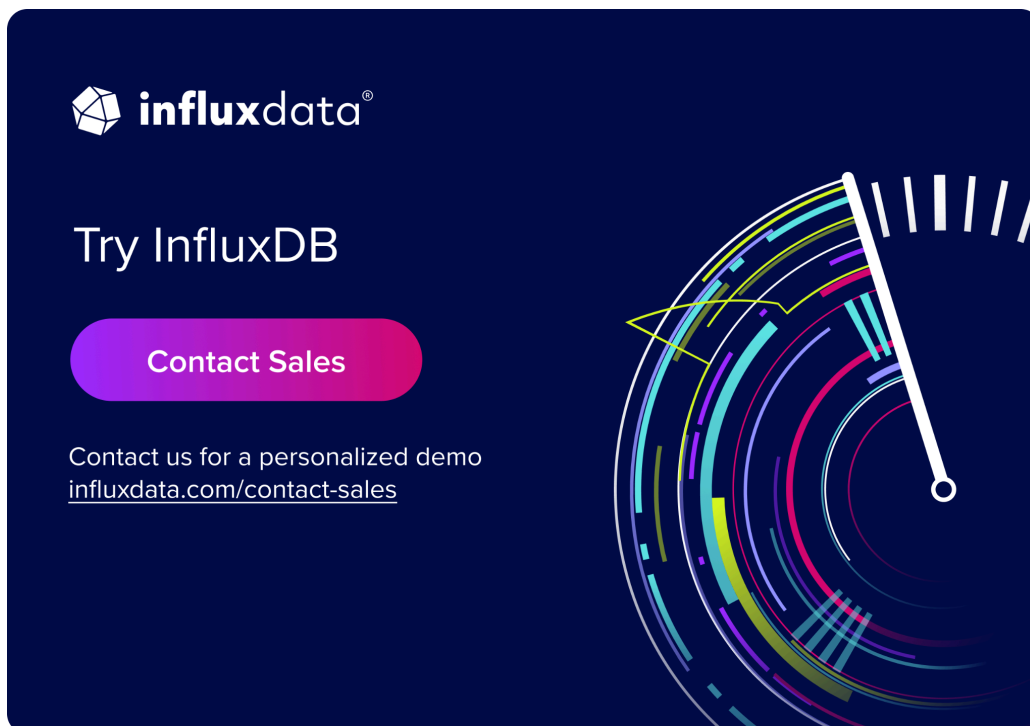
Try InfluxDB Cloud for Free

Contact Sales

Get documentation

Additional tech papers

Join the InfluxDB community