INCYTE STUDIOS

SUMMITESP
SUMMITVIEW SYSTEMS UPGRADE 2019
incytestudios.com

How we upgraded a massive oilfield data gathering and analysis platform; reducing storage requirements by 10x, high-use query times by 88%, and significantly dropping AWS spend.

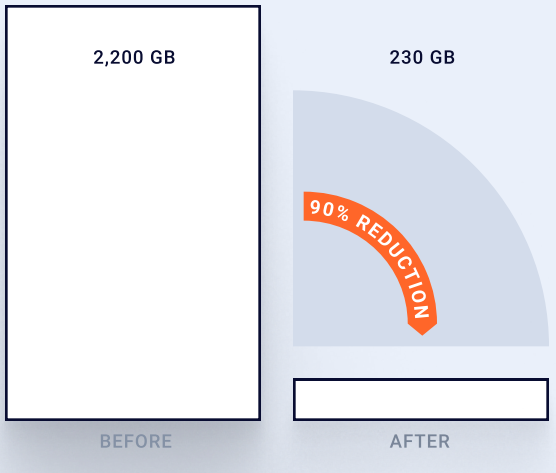
BACKGROUND & SCOPE

SummitView is a oil well monitoring application developed by Summit ESP (a Halliburton subsidiary). SummitView receives telemetry datapoints from thousands of operating oil wells from around the world. Embedded computers are placed near the well and the well's drive. Using SCADA interfaces, telemetry data is gathered and sent to an "ingestor" in the cloud for cataloging the data long term and providing reporting APIs as well as a customer-facing web portal for seeing well performance.

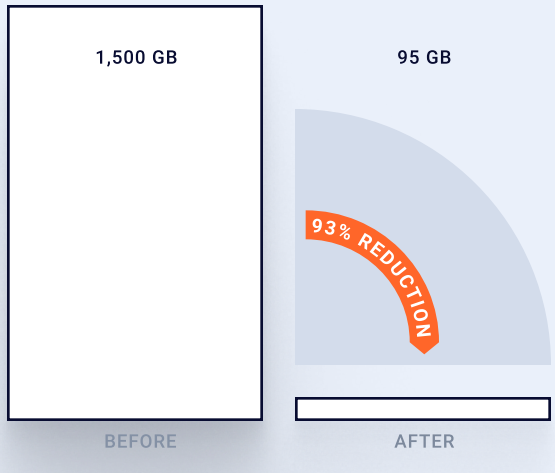
OBJECTIVES

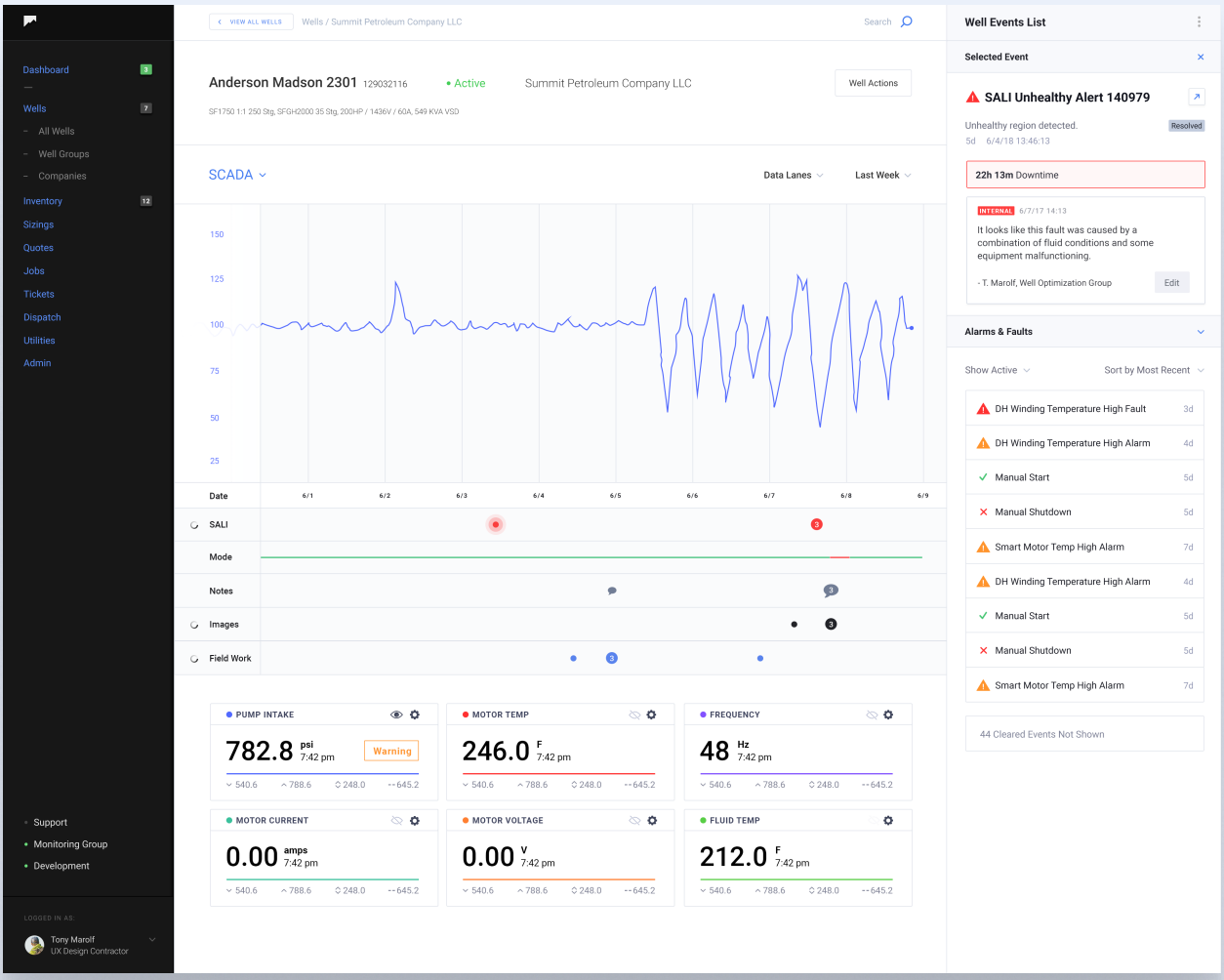
As SummitView found more success, the operational complexity of the product grew. Incyte's task was to relieve stress on the database, increase reliability of the ingestion, as well as update the web portal code to enable faster and more robust graphing and reporting of the data.

Telemetry table database size, before and after upgrade.



Telemetry table index size, before and after upgrade.





SummitView web interface.

PROBLEM STATEMENT

SummitView's underlying database was an Amazon AWS RDS instance of Postgres 9. The schema for readings (telemetry received by the thousands of wells) was not partitioned or indexed in a way that would support continued growth of the product. The database performance was suffering under daily use with routine reporting queries taking over 45 seconds to deliver results. The customer facing web portal would take upwards of 5-10 seconds on some wells to draw the correct graphs. The web-portal code was also somewhat dated and had lost its primary developer (another contractor)

CHALLENGES

In addition to the primary web portal, there were several third-party applications that depended on the original schema and API. Incyte had to adopt a "first do no harm" policy while simultaneously preventing data loss, adding new ingestor processes, and updating the reporting framework. The web portal graphing code also needed to be replaced in a way that would not break any other portions of the site.

TECHNOLOGIES USED

AWS
POSTGRES 11
TIMESCALEDB
VUE.JS
HIGHCHARTS
FALCON
PYTHON
JAVASCRIPT

SOLUTION

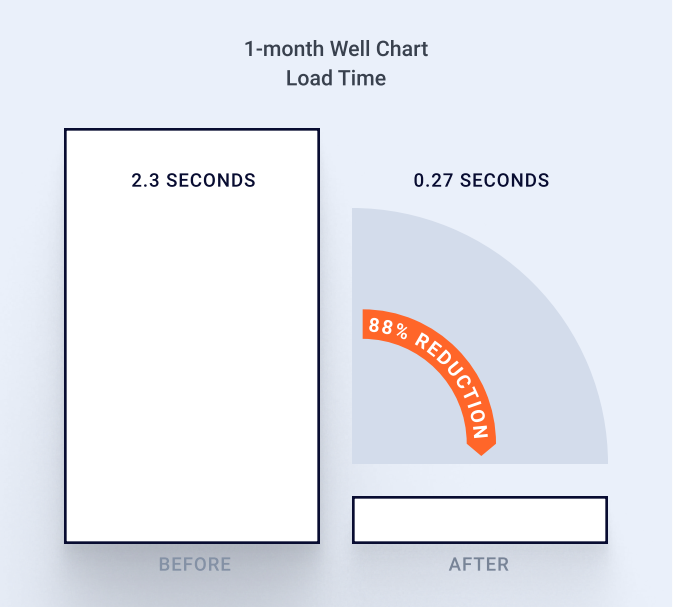
Incyte began a 3-pronged approach to the challenges above:

1. Porting the the Chart.js graphing code to HighCharts
2. Porting the Django API to Python3 + Falcon
3. Replacing the Postgres RDS instance with EC2 Postgres+Timescale

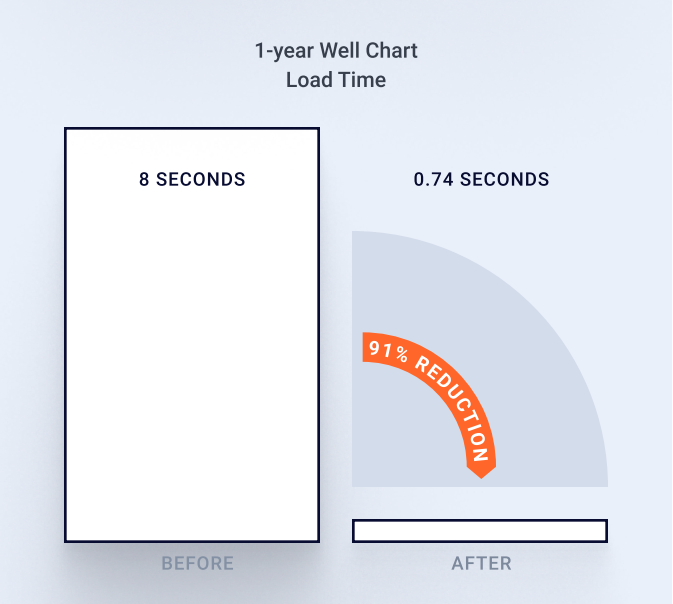
Incyte replaced the tool's primary database with Postgres 11 + TimescaleDB. TimescaleDB is designed to alleviate several issues of collecting large amounts of sensor data. The built-in hypertables of Timescale allowed time-based partitioning of the existing data as well as the familiarity of Postgres. After designing a new storage schema making use of hyper-tables, Incyte wrote converters and migrations to move the old data into the new format.

Next, Incyte provided a "view" over the new data that mimicked the older format, allowing 100% backward compatibility with older queries and reports. Once this view was working, Incyte converted the ingestors and writers to send data in the newer format. Older consumers of the data were then also converted, and the old tables removed to save space. During database development, the updates to the API and graphing code were done simultaneously.

To mitigate scheduling risk, each of the 3 efforts were designed to be deployed in isolation, so that if any of them were delayed the customer would still see benefits. Thankfully, all three were completed on time and shipped to the customer's satisfaction.



DATA HEAVY PAGES LOADING 10X FASTER



BENEFITS

All work was completed on time and shipped to Summit's customers. The size of the telemetry tables went from 2.2TB data with 1.5TB of indexes down to 230GB data with 95GB of indexes, a 10x storage size reduction.

Query performance for reporting and charting saw similar speed ups. 30 days worth of readings for a well went from taking 2.3 seconds down to .27 seconds; 1 year of data from 8 seconds down to .74s. The customer also saw reduced cost in AWS as far less powerful hardware was required to support the more efficient data use.