

# DealerSocket and ClearScale Partner to Optimize Data Pipeline Application



## Executive Summary

[DealerSocket](#) is a leading provider of innovative software for the automotive industry. The company offers a suite of seamlessly integrated products to help dealerships sell and service vehicles more profitably while delivering exceptional customer experiences. Among these products are robust solutions for customer relationship management (CRM), vehicle inventory management, analytics reporting, and more.

Employing the latest technologies, the company continually develops new solutions and product integrations to help its customers gain and maintain a competitive edge.

## The Challenge

In today's fast-paced business landscape, every second counts. To help dealerships make data-driven decisions on the fly and ensure essential information is at their fingertips, DealerSocket developed an application to generate real-time data pipelines. Integrated into its already efficient product suite, the solution would rapidly gather, process, and push relevant data across integration points — streamlining processes, eliminating double entry of information, unifying customer profiles, leveraging trend and buying habit data, and much more.

The solution was built using [Apache Kafka](#), an open-source, distributed streaming platform, with the Kafka deployment running in DealerSocket's [AWS environment](#). It features built-in scalability to accommodate increases in data and workload complexity.

The application worked well in the company's development environment. Before moving it to production, however, DealerSocket wanted to conduct a comprehensive analysis to ensure the solution's scalability and optimal performance under real-world conditions.

Specifically, DealerSocket sought to partner with a third-party company that could assist it in assessing and optimizing the application architecture and code, the Kafka systems configuration, and the underlying [Kubernetes](#) cluster orchestration system. A stress testing strategy — and the actual stress testing — was also needed to validate the solution.

That required the prospective partner organization to possess the rather unique combination of extensive experience and expertise in both Kafka AWS deployments and in the use of Kubernetes for automating deployment, scaling and managing containerized applications. [ClearScale, an AWS Premier Consulting Partner](#) that specializes in delivering customized cloud systems integration, application development, and managed services, fit the bill.

## The ClearScale Solution

Responding to DealerSocket's specific request, the ClearScale proposed an initial information-gathering phase to review the business objectives and technical requirements of the DealerSocket application. An in-depth code and systems audit would follow, yielding recommendations for enhancements.

Next up would be a stress test strategy, test implementation, and a plan for continued performance optimization. By teaming up throughout the entire project, ClearScale and DealerSocket could combine their respective strengths and knowledge to generate the most comprehensive assessment and optimization plans.

### The Code and Systems Audits

Following a review of the data pipeline application's functional and non-functional requirements, separate code audits were conducted of the Java, C#, and AngularJS components. The code was assessed against best practices to identify opportunities for simplification and performance enhancements, and to eliminate any potential performance bottlenecks.

The deliverables included an audit report and list of recommendations for application optimization. Among them was following Java naming conventions for the auto-generated code, using the "Dead Letter Queue" approach for capturing processing errors, and employing log compaction for the continuous data capture. It was also recommended that certain tools be employed to continue to control and improve code quality.

As with the code audit, the systems audit started with a study of the application requirements with a focus on the general architecture, data flow, and deployment pipeline. An in-depth assessment was then conducted of the entire infrastructure. That included the AWS EC2, the Linux OS, the Kubernetes cluster deployment, and the Kafka architecture — from the Java settings through the Apache Zookeeper configuration.

In addition to outlining the findings, the ClearScale and DealerSocket team created a remediation plan that specified action items for enhancements and best practices at the various infrastructure levels. For example, at the AWS level, spreading brokers across multiple Availability Zones could prevent the Kafka service from becoming unavailable in the event of an Availability Zone failure. At the Zookeeper level, disabling the client limit could help prevent running out of allowed connections.

# Kafka Load Testing Solution Architecture

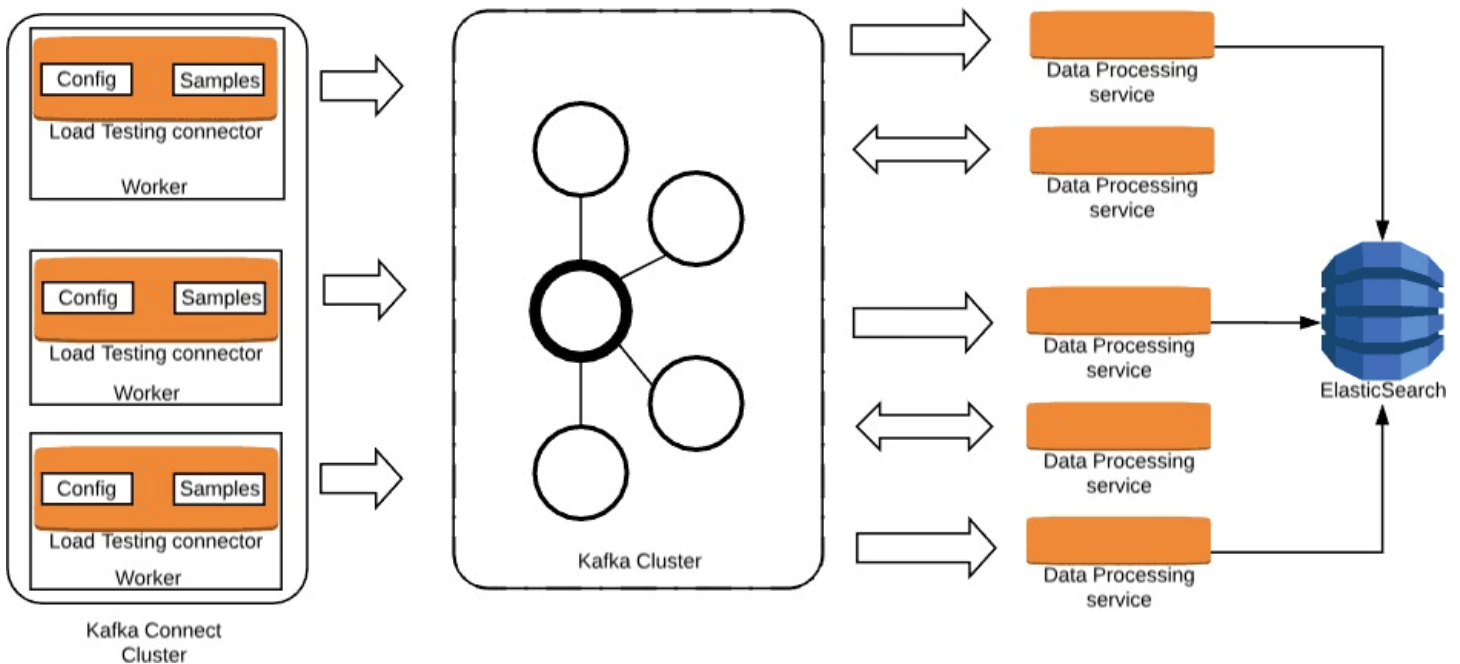
## The Stress Testing Strategy and Implementation

The second part of ClearScale’s partnership with DealerSocket was the development and implementation of a stress test strategy, including scope, targets, tools, scenarios, performance measurements, testing support, and the results analysis.

The way the data pipeline application was set up, the MSSQL continuous data capture (CDC) source connectors read the changes from databases and then send them to the Kafka cluster. Data processing services transform the data, returning results to Kafka or sending them to ElasticSearch. Using real databases to test this process would have been cost prohibitive and impractical. Nonetheless, any load test needed to replicate real-world scenarios as closely as possible to ensure the application worked as expected.

ClearScale and DealerSocket assessed the options. It was determined that the best approach was to employ a Kafka Source Connector that could play pre-recorded samples that DealerSocket would provide based on real-world data. Running the connector in distributed mode would allow for easily scaling the load to the cluster, simulating actual processes.

## Kafka Load Testing Diagram



The initial load tests identified application load limits and other areas for optimization. This information enabled DealerSocket to further finetune the application prior to moving it to production. A monitoring solution was also developed so the application could continue to be assessed and improved.

## The Benefits

The recommendations for optimizing DealerSocket's application that resulted from its collaboration with ClearScale have been implemented and tested. The company now has a proven, high-performing solution for generating real-time data pipelines that super-charges the efficiency and value of its product suite. It scales reliably, handles large data sets, and is regularly monitored to ensure it continues to perform at the level DealerSocket and its customers expect.