



**Performance Engineering  
Practices Improve  
Systems Performance  
by 200% in 6 Months**

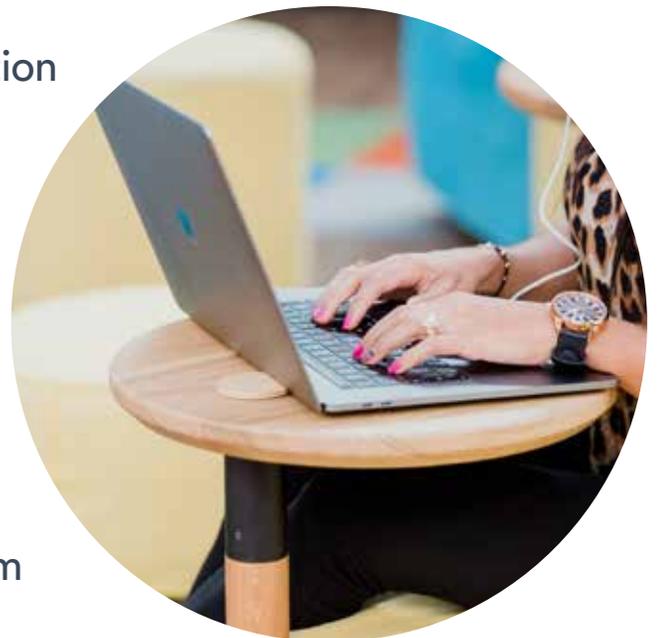
# Results

- 200% performance improvement in critical cases in 6 months
- Fully automated testing allows for over 300 test executions a day
- Issues are flagged at the beginning of the software development life cycle
- Improved information analysis
- Tests run in isolated performance environments before being deployed to production
- Time and money savings achieved by timely performance analysis of prototypes

## Executive Summary

PSL's client, a leading provider of integration software, achieved improved and consistent performance in complex, high-load products by deploying a team of highly skilled nearshore performance engineers.

The nearshore performance team increased the number of test executions, enhanced analysis processes, and improved system performance across the board by applying data-driven insights from automated testing.



# Tech Stack



Jenkins

Terraform

kibana

elastic

## Background

Our client, a leading provider of software integration platforms designed to connect SaaS and enterprise applications, works with companies in industries ranging from automotive and transportation to entertainment and customer goods. Its platform allows customers to create a highly efficient network of applications to fully leverage the potential of different components and accelerate development.

With so many companies and industries depending on the stability and speed of the client's platform, performance concerns are a top priority. Ensuring customers' products function even under the most extreme conditions is essential. So, when the client needed to build a team of highly skilled performance engineers, they turned to PSL, a leading nearshore software development outsourcing provider in Latin America, to help structure and execute performance engineering best practices.

# Solution

To achieve more accurate predictions and in-depth insights without dedicating countless hours to running manual tests, the PSL team began by automating all performance tests using Hudson and later migrated to Jenkins. They used Terraform to create environments in AWS, Elasticsearch as a database, and Kibana to display performance dashboards. The team went from running 6 tests a day to executing between 200 to 300 performance tests in the same time.

Thanks to fully-automated testing, results were now easier to track and changes to the code safer to deploy. This also gave the technical and business teams the metrics they needed to set achievable goals and fix performance issues before production.

The PSL team developed a way for the client's development team to test changes to the code in safe, isolated environments. It was achieved by creating environments in AWS and performing load testing using JMeter. Moreover, the strategy only allowed one change or factor to be altered at a time. This allowed them to accurately identify effects on server runtime later, saving time and cutting costs in the long run.

Under the guidance of the transversal PSL performance engineering team, critical use cases saw performance improvements of more than 200% in less than six months. The end product, which was improved and iterated on through performance testing and tuning, resulted in the client securing larger contracts. In the future, PSL will be working closely to develop new features for end users and continue to ensure a faster, more stable system.

# Challenge

Even though the client was already implementing performance engineering practices, they wanted to optimize and achieve higher levels of speed and performance, but they were having a hard time finding local resources.

End user expectations are high for quality and performance and are constantly climbing, which is why PSL's client was facing pressure to go even further with their performance testing capabilities.

# Solution

In this case, the client was hoping to use performance testing to ensure that its many applications would perform flawlessly in high-load environments, efficiently analyze large amounts of data, and predict how applications will act in the future to start building more robust applications from the start.

