



How LaneOne Satisfied Performance Expectations from Day 1



LaneOne provides an “all-inclusive” premium experience for top concerts—packaging great seats with parking or ride-share credit, a preferred entrance, special food and drink options, and concierge-grade customer support. Every part of this experience is powered by their cloud-based app.

Given that LaneOne’s mission is to deliver an exceptional end-to-end experience, their first publicized on-sale had to go flawlessly. Compounding the pressure, their launch offered exclusive access to American Express card members for the best seats to the Eagles 2018 tour. This meant that the first test for their new application would be a highly-publicized on-sale for an act that routinely sells out stadiums in a matter of minutes. They needed confidence that their application could handle extreme loads—before a single customer ever accessed the site.

Using Tricentis Flood to load test at both the protocol level (Gatling) and browser level (Flood Chrome), they iteratively assessed and optimized the system’s performance characteristics until their goals were met. By the time of the launch, LaneOne could rest assured that their brand-new application would meet the high expectations of customers as well as eminent partners such as American Express and Ticketmaster.

In the words of Josh Cronemeyer, CTO of LaneOne:

Event ticketing for some of the world's most successful touring acts like the Eagles and Harry Styles means you have to be ready for big traffic spikes. We needed to a way to optimize the perform ance of our individual components (services), as well as understand and fine-tune the performance from the perspective of the end user. Tricentis Flood was recommended to me by a colleague. It looked like exactly what we needed: a cloud-based testing platform that would let us distribute both protocol-level testing and browser-level testing.

Our software platform is all cloud based: a mixture of microservices and serverless approaches. We started testing by using Gatling to test our services in isolation. The tests were easy to write and try out locally. Then, we'd provision our production-like environment and let Flood distribute our tests to a grid (distributed load test infrastructure). After we were satisfied with the performance characteristics of each service, we would run our tests at a higher level of abstraction with Flood Chrome’s browser-level tests.

Running load tests this way was great for helping us understand how to configure the individual services to work together. We got answers to questions like:

- What should the Amazon DynamoDB read/write capacity be?
- How many web and background (Heroku) dynos does each service need?
- What are the right number of (Sidekiq) workers for our various background queues?
- Is our postgres instance (PostgreSQL database server) appropriately sized?

That process gave the team more confidence that the system would perform well at launch. We could also answer business questions about how many simultaneous purchasers we could accommodate and how many orders per minute we could process.

By the time we launched, we had tremendous confidence in the system's performance characteristics. Moreover, we were able to give the business confidence that we could support enough concurrent purchasers and order throughput to see us through a successful marketing push from our partners at American Express and Ticketmaster.

Since that initial launch, we've been selling tickets for the Eagles and Harry Styles without any scaling issues. We really appreciate how Flood lets us quickly and efficiently get about the business of load testing. Our next steps will be to automate load tests as part of our continuous delivery pipeline.

About Tricentis Flood

Tricentis Flood is a cloud based distributed load testing platform for everyone. Flood lets you run globally distributed load tests with your favorite open source tools, making the creation, execution, and analysis of load tests easy for Dev/Test teams, so they can obtain fast feedback on performance, availability, reliability and scalability. Whether you need to load test a single URL, simulate realistic browser behavior with Chrome, or execute large concurrency and volume with JMeter and Gatling, Flood provides a simple and affordable platform for scaling load tests on demand.