



Since 1946, Fender has revolutionized music and culture as one of the world's leading musical instrument manufacturers, marketers and distributors.

**Environment:**

- AWS Lambda microservices written in Go
- AWS Lambda sending Cloudwatch logs to Honeycomb

**Need:**

- A powerful and intuitive interface for debugging and troubleshooting problems.
- Fast search results across high-cardinality fields such as unique customer IDs.

**Love:**

*"The goal of the platform team is to create and support services that support our web and mobile applications and provide data and analytics to the business. In that way we are able to understand and support players.*

*We have an ELK stack, but the query language is cumbersome and the UI is not as easy to use as Honeycomb.*

*Honeycomb and Lambda work very well together. Without Honeycomb, it was very difficult to get all the Cloudwatch logs correlated, but now we can just do a quick search by request or userID with no difficulties."*

**- Michael Garski**  
Director of Engineering, Fender

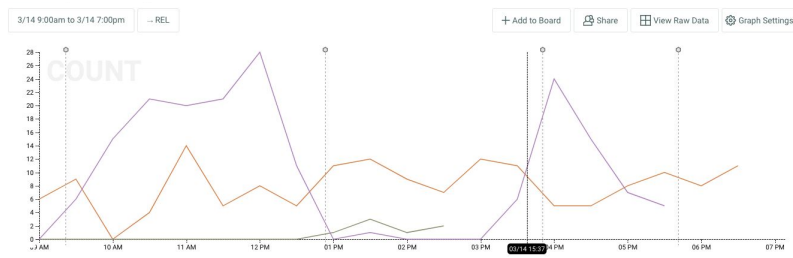
**Challenge:**

Fender's fanatical customer focus means systems observability is a necessity. To understand and support their users, they must have visibility into their users' experience of their service. The Fender Platform team maintains their own ELK environment to aggregate logs and investigate problems, but have found the platform slow and difficult for team members to learn to use.

**Solution:**

Fender installed Honeycomb specifically to get the benefit of searching across all their Cloudwatch logs from Lambda at once, and have been especially pleased with the speed at which they can zero in on an issue.

Recently, the Fender Platform team deployed an update to the subscription management service for Fender Play users. Soon afterward, they noticed errors related to the billing systems, which started to spike over the course of the morning.



Using Honeycomb, they determined that the issue was related to differences between their test and production environments and could confirm, thanks to the Honeycomb Marker they'd set, that this problem was definitely caused by their recent deployment. They were able to roll back to a stable environment within five minutes.

**Results:**

Honeycomb allowed Fender to drill into their data and identify the exact users impacted by the issue, allowing their Customer Support team to contact those users **before** those specific users contacted Customer Support to report a problem.

*"If we did not have Honeycomb it may have taken an additional 30 minutes or so to determine what the issue was by poking around in ELK or Cloudwatch logs. Honeycomb's visualization and honeymarkers made it obvious that the issue was related to a recent release. On top of that, Honeycomb allowed us to determine the affected users and pass that information on to our support team."*