How Understory Cut Debugging Time by 80% with Dash0

UNDERSTORY

The Problem:

Reliance on disjointed AWS CloudWatch logs and metrics made debugging slow and painful.

The Solution:

Dash0 provided a unified view of traces, logs, and metrics without requiring proprietary SDK lock-in.

The Result:

A new engineer reduced a potential week-long debugging session to a single day, and the team now relies on one tool instead of three.

Creating Memories, Not Managing Logs

The mission at Understory is simple: empower experienced hosts to create amazing memories. As a platform helping them run and grow their businesses, Understory handles operations so customers can focus on hospitality.

But for Lars, the Technical Co-founder of Understory, the reality of building that platform wasn't always as seamless as the user experience he aimed to provide.

When building from scratch, you want engineers focused on innovation, not digging through logs. Yet that's where they found themselves. Despite a modern stack, their observability was immature. Engineers were digging through disconnected logs and wasting development time trying to understand failures.

This is the story of how Understory transformed their "immature" observability setup into a powerful, unified engine that saves their engineers days of work every single week.

The Challenge: Sifting Through the Noise

Understory adopted modern standards like OpenTelemetry from day one. However, generating telemetry data is one thing; making it useful is another.

> Our observability journey at Understory was very immature. We didn't really have a good way to get the data out and actually start using it.

The team relied on built-in AWS infrastructure. When an issue arose, the workflow was manual and fragmented. They sifted through CloudWatch logs, checked metrics separately, and tried to tie signals together.

It was a classic "needle in a haystack" scenario. The data was there, but the insight was buried.

> It was really spending a lot of time digging around trying to find what we needed. We lacked the thing that tied them all together.

The Switch: Seamless Integration Without Lock-In

The turning point came through a professional connection. What started as a conversation turned into a demo that addressed Understory's frustrations.

For Lars, the decision to test Dash0 wasn't just about the features,it was about the architecture. One of the biggest fears for any technical co-founder is vendor lock-in. Nobody wants to rewrite their codebase to accommodate a monitoring tool.

"What really spoke to us was that we didn't have to actually start doing anything different from our end," Lars explains. "We weren't locking ourselves into a proprietary SDK. It was built on open standards."

Because Understory was already using OpenTelemetry, the integration was almost trivial. They simply configured their OpenTelemetry (OTEL) collector to ship data to Dash0. There was no heavy lifting, no code refactoring, and no closed doors.

> We could just point all of our outputs to you guys from the collector, and then we would start seeing the value. It was very much a two-way decision, we didn't close any doors behind us.

The Impact: Turning Weeks into Days

The impact became clear quickly. The team moved from a fragmented system view to a unified pane of glass where logs, metrics, and traces lived together.

The real test came just weeks after implementation. Understory had recently onboarded a new engineer who was still learning the platform's architecture. During his first week, he encountered a complex bug that required deep triage across multiple services.

> If we hadn't had Dash0 at that point, they would have spent probably most of the week trying to figure stuff out, behind us.

Instead, he searched across the stack, followed traces, and narrowed down the issue within the first day. "He was narrowing into where the actual issue was within the first day," Lars says.

Dash0 turned a four-to-five-day investigation into a one-day fix.

A "Phenomenal" Developer Experience

Beyond the software itself, the relationship between Understory and Dash0 has strengthened their engineering workflow. Lars describes the support experience as "close to phenomenal."

"Having the Slack channel that we do... we definitely have an experience of being heard," he says. "Issues are being resolved on the day usually."

In the beginning, engineers still opened CloudWatch out of habit.

> Now people are just going straight into Dash0. It provides insight across everything.

Conclusion: Mature Telemetry for Modern Teams

For Understory, Dash0 has delivered exactly what a modern engineering team needs: clarity without complexity.

When asked how he describes Dash0 to industry colleagues, Lars puts it simply: "They're taking all the different telemetry signals that you can find and shoving it into one UI where you can correlate everything in a much more seamless way."

For technical leaders evaluating their stack, Lars offers one final piece of advice:

> Try out the MCP and hook it up to Claude code and see what it does, because that is purely magical.

Ready to unify your telemetry?

Stop digging through CloudWatch logs and start seeing the full picture.

Learn more about Dash0 today

dash0

dash0.com