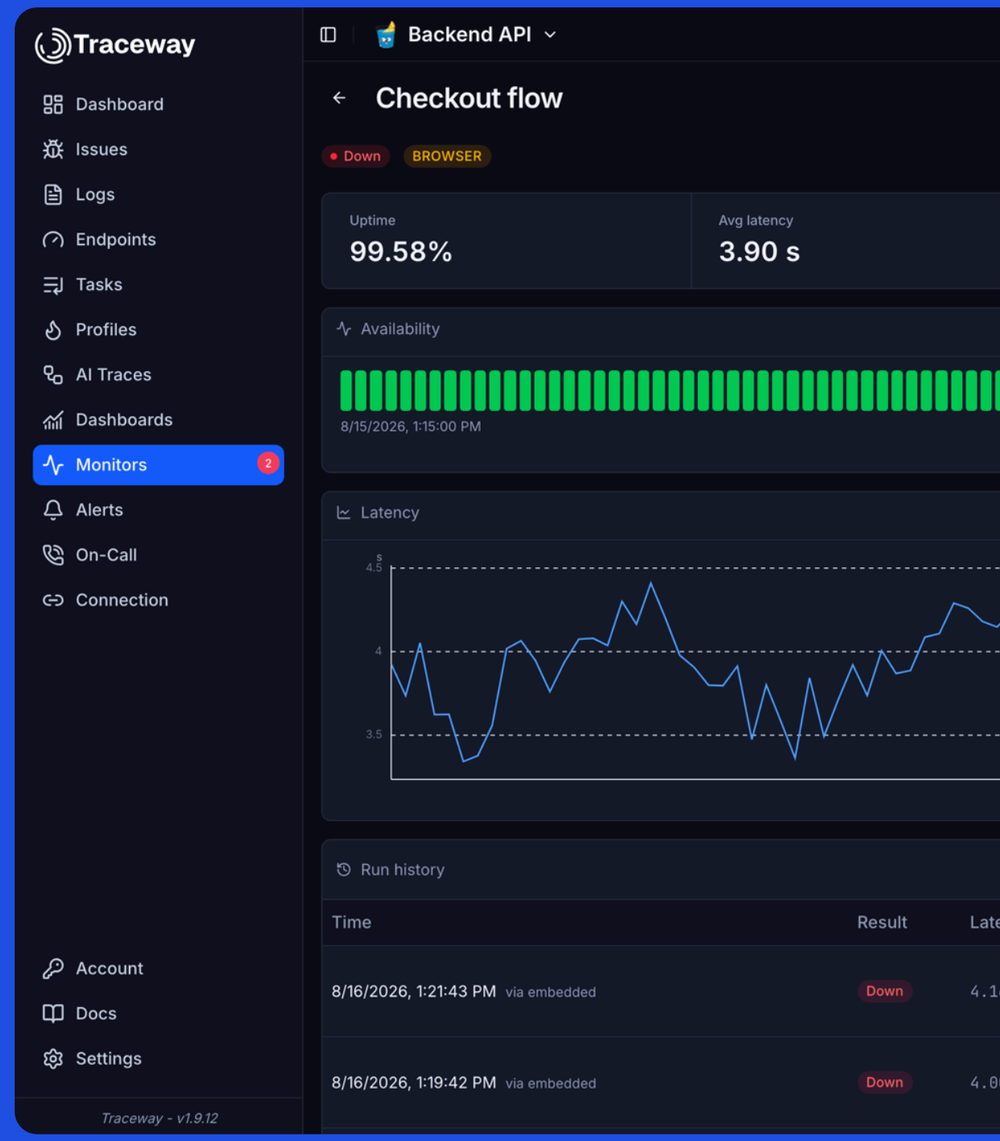
Traceway

Follow the trail to the bug.

Traceway shows you what broke in production and why, usually before your customers notice.

tracewayapp.com



The problem

Most teams hear about outages from their customers.

Too many tabs

The error is in one tool, the logs in another, the graphs in a third.

Long nights

Someone spends the evening grepping servers to work out what one request did.

A fix on a guess

Nobody can reproduce it, so someone ships a likely fix and hopes.

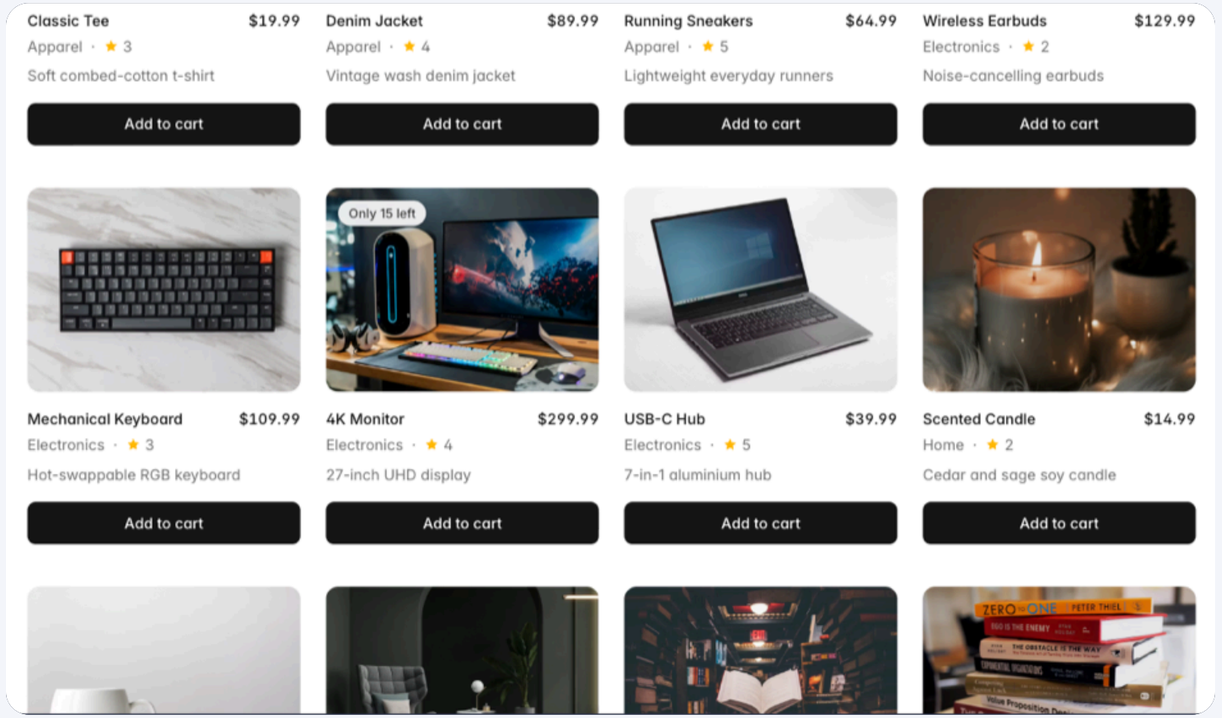


A story

This is Ada.

She’s an engineer at a webstore.

And it’s a Friday evening.



Ada and the store are made up. The screens in this story are the real Traceway product.

Friday, 18:40

Checkout starts failing.

Let’s play this evening out twice. First, the way it usually goes.

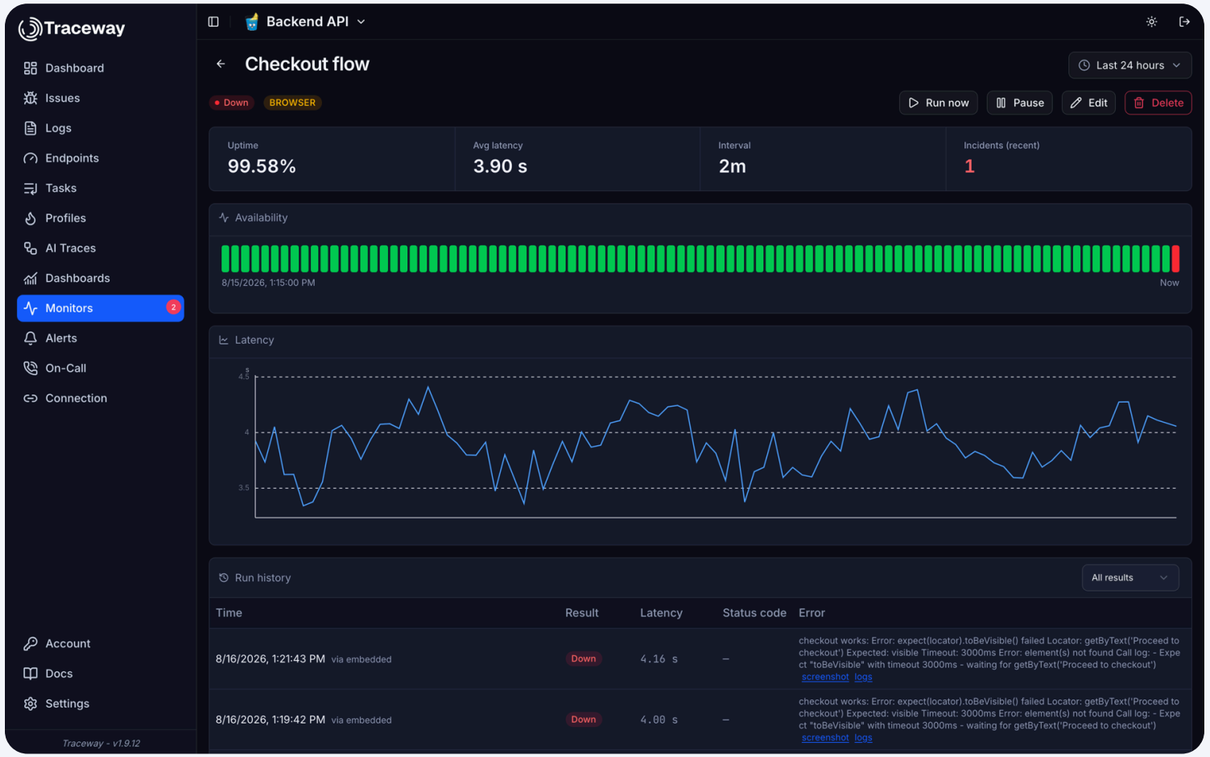
Take one · the usual way

Ada’s Friday, without Traceway.

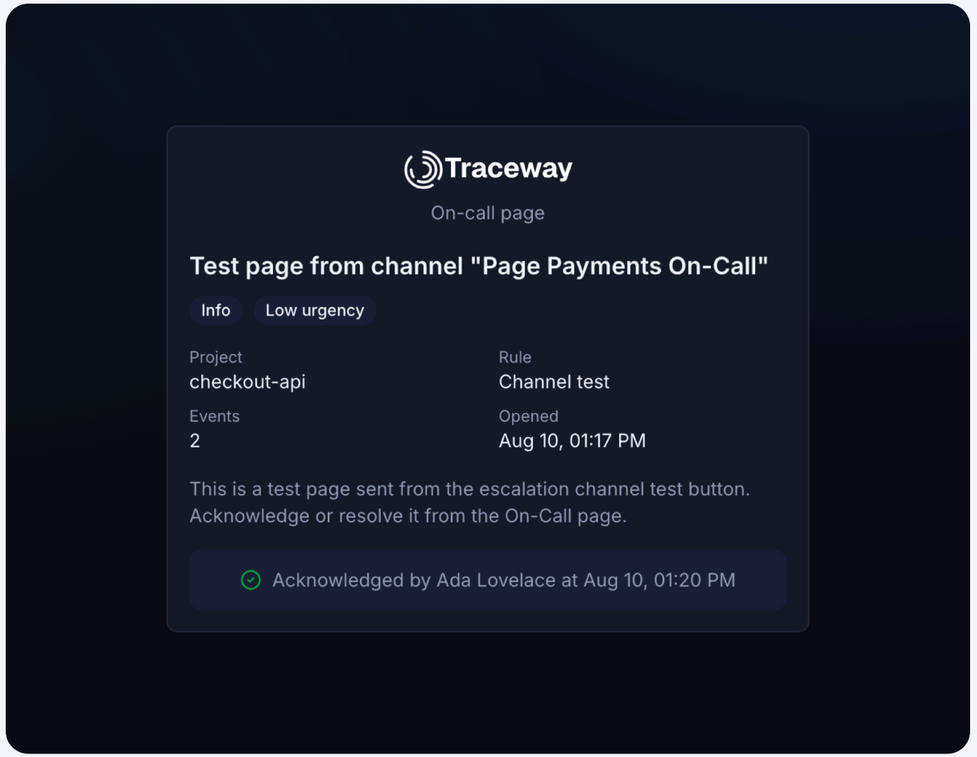
18:40	Payments start failing. Nobody knows yet.
19:30	Support forwards an email from a customer.
20:15	Ada opens a laptop at dinner. Three dashboards.
21:40	Still grepping logs. Can’t reproduce it locally.
23:10	Ships something that should fix it. Goes to bed hoping.

Take two · with Traceway · 18:41

Ada’s phone buzzes a minute in.



A check that clicks through checkout fails, and Traceway opens an incident.



Ada taps acknowledge right in the message. No login needed.

Browser checks are self-hosted only today. HTTP and TCP checks run on every plan.

Take two · 18:43

The errors are already sorted.

Every repeat of the same error lands in one issue. Ada gets a short list instead of a wall of logs.

What broke, how often, since when.



Take two · 18:46

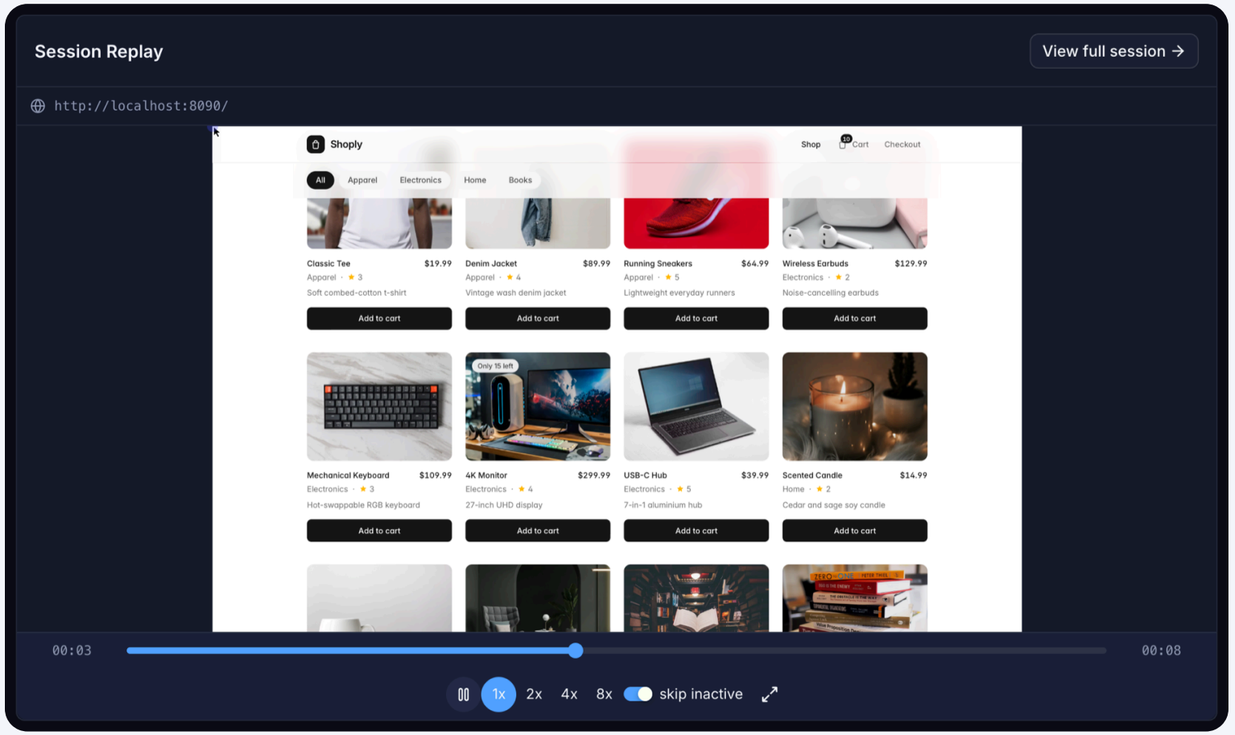
Ada opens one failed checkout.



Every step it took, with the logs from that request under the step that wrote them.

Take two · 18:48

Then watches what the customer did.



The replay plays back the clicks right before the error.



The stack trace points at the real file and line.

Take two · 18:55

Ada hands it to a coding agent.

```
$ npx skills add tracewayapp/traceway --all

> /traceway debug <link to the issue>

It reads the issue, the trace and the logs,
then finds the cause in your code.
```

Works with Claude Code, Cursor, Codex, or any MCP client.

The CLI is read-only. The agent can look at production but can’t touch it.

So, how did it go?

Same bug, very different Friday.

Without Traceway

Found by

A customer, 50 minutes in

Root cause

A guess, at 23:10

Ada’s Friday

Gone, and orders with it

With Traceway

Found by

A monitor, 1 minute in

Root cause

The exact line, at 18:48

Ada’s Friday

Back at dinner by 19:00

The times are from this story, not measured customer data.

What you get

What changes for your team.

You hear about it first

Pages that escalate until someone picks up.

One place to look

Errors, traces, logs and replays, linked.

Straight to the line

Readable stack traces and AI agents.

A bill you can predict

No per-host or per-seat fees.

No lock-in

OpenTelemetry and MIT licensed.

Help getting there

We set it up with you.

How we can help

Three ways to work with us.

Traceway Cloud

We host it. You set it up with AI skills.

Consulting

We analyze your architecture and processes, propose a rollout plan, and help you do it.

Dedicated DevOps

We monitor and manage Traceway inside your own cloud.

Traceway

Give your next Friday a better ending.

Book a 30-minute call

Book
calendly.com/dusan-tracewayapp/30min

Our services
tracewayapp.com/services