Serverless Sherlock Returns

Cracking Fargate Mysteries with New Relic Observability

Avinash Shashikant Dalvi
Senior Staff Engineer @Nushift Technologies
AWS Community Builder

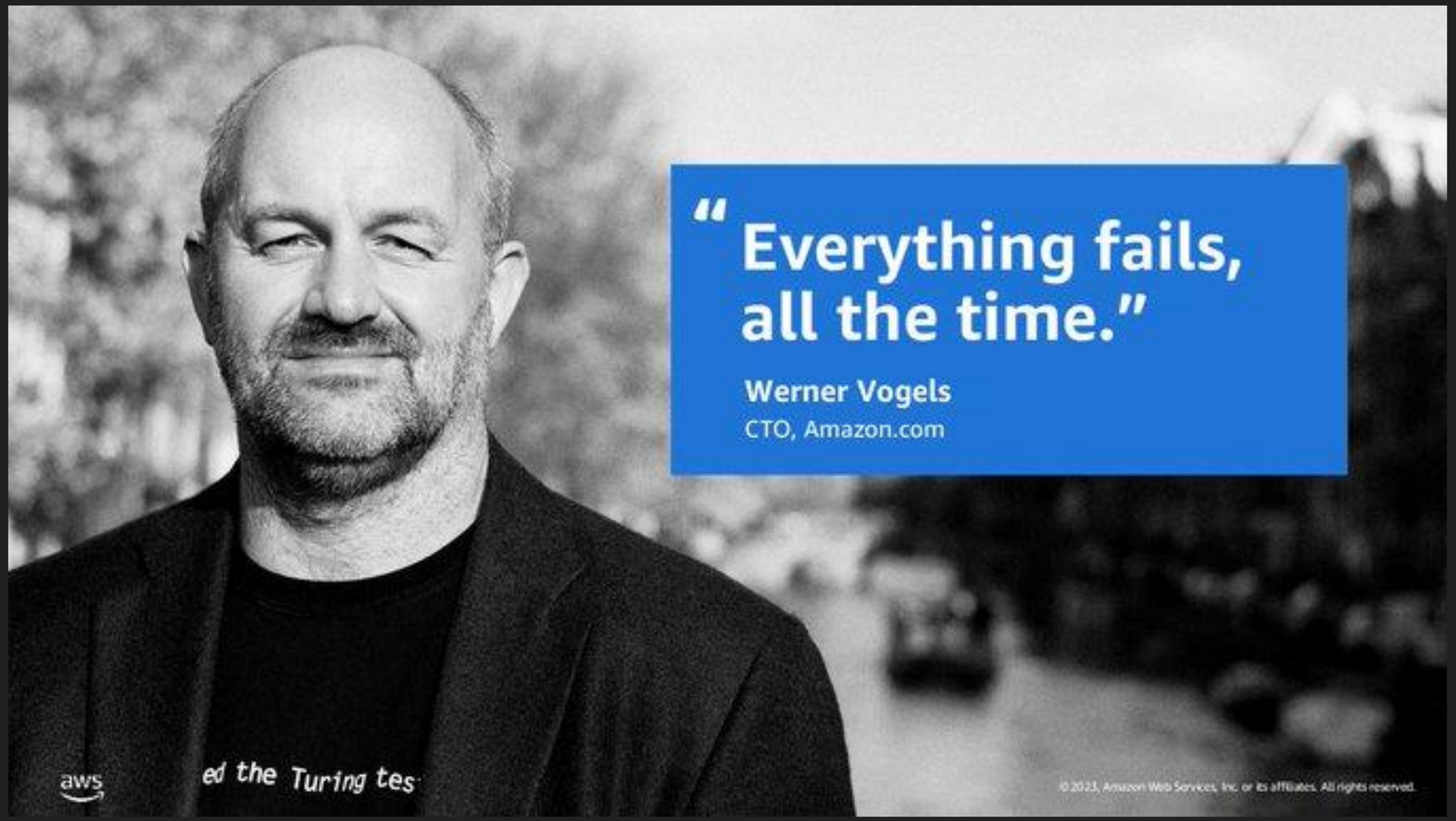

The
Detective
Journey

AWS Fargate in 30 Seconds

“AWS Fargate: Run containers—without managing servers, clusters, or VMs.”

- Serverless containers
- No infrastructure management

Photo by frank mckenna on Unsplash


“ Everything fails, all the time.”
Werner Vogels
CTO, Amazon.com
aws
ed the Turing tes
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The Mystery Unfolds

Symptoms

- Average response times jumped from 200ms to 1200ms within 30 minutes
- Error rates spiked by 25%
- Unexplained task restarts

With each passing minute, user complaints were piling up, and our on-call team was scrambling for answers.

Without direct server access, where do you start solving this mystery?

Why Fargate Failure is mysterious

In the world of serverless, we've traded control for convenience. But when things go wrong, we find ourselves in a fog of abstraction.

Key challenges

- Limited visibility into the underlying infrastructure
- Ephemeral nature of Fargate task
- Lack of direct access to the host
- Difficulty in local reproduction
- Standard metrics and logs offer limited insight into dynamic issues.

Photo by Tracy Higashi on Unsplash

In the Dark...

“Logs everywhere. But no answers.”

- Overwhelming logs: High volume of unstructured log data from distributed services.
- Lack of context: Logs provide event details but lack correlation and causality.
- Investigative burden: Manual searching and filtering prolong incident resolution.

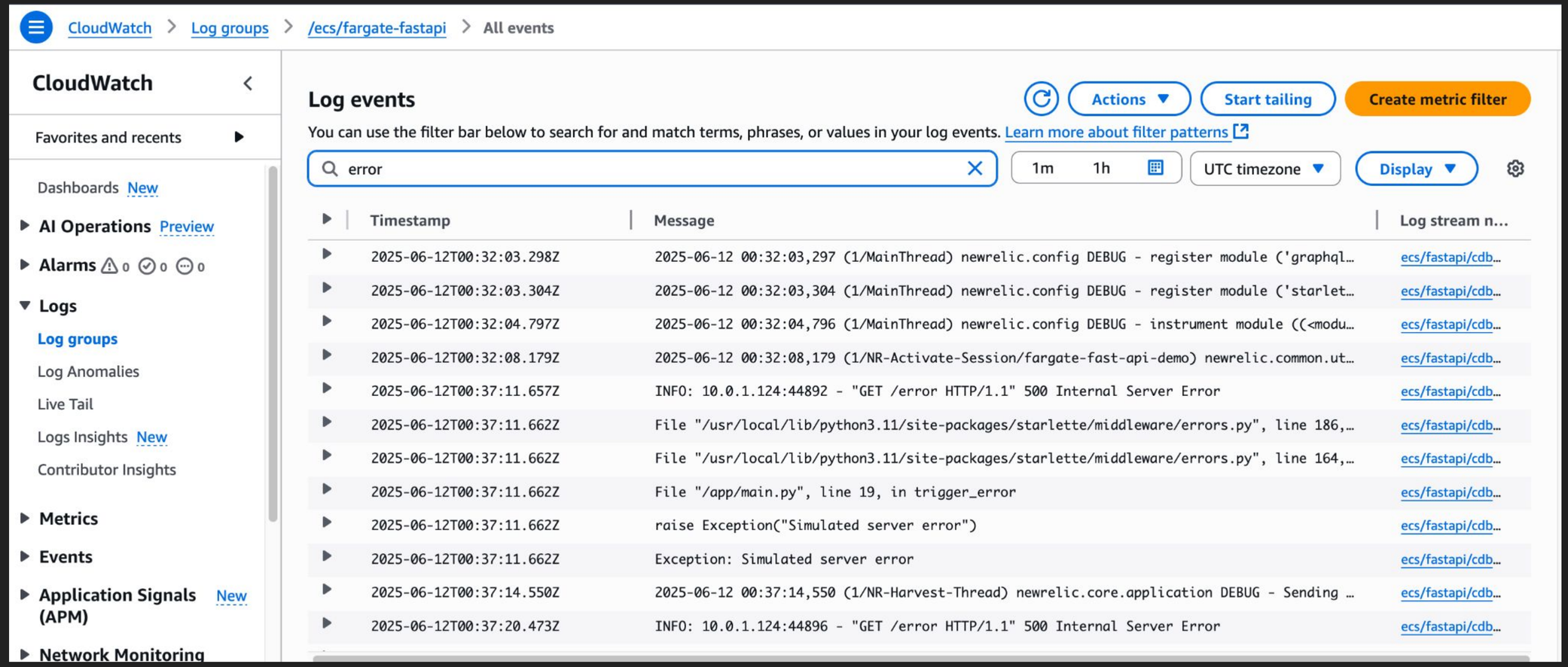
The Classic Clue Hunt

“CloudWatch shows you what happened—but not why.”

- Symptom visibility
- Fragmented evidence
- Manual correlation

The Classic Clue Hunt

“CloudWatch shows you what happened—but not why.”



A Mystery from My ECS Days

- Months chasing a 502 timeout
- Logs pointed everywhere—app, DB, network
- We manually tuned infra, scaled up, added application logs
- Using PHP Xdebug done profiling of app flow

The real culprit?


What I Wish I Had
What I Wish I Had:
• Shows me the full request/transaction path, not just logs
• Connects errors, slowdowns, and dependencies automatically
• Flags the exact culprit (code, DB, or infra)
• Turns a sea of logs into a connected story
Makes me feel like a Sherlock: not a log archaeologist!

The Detective's Toolkit

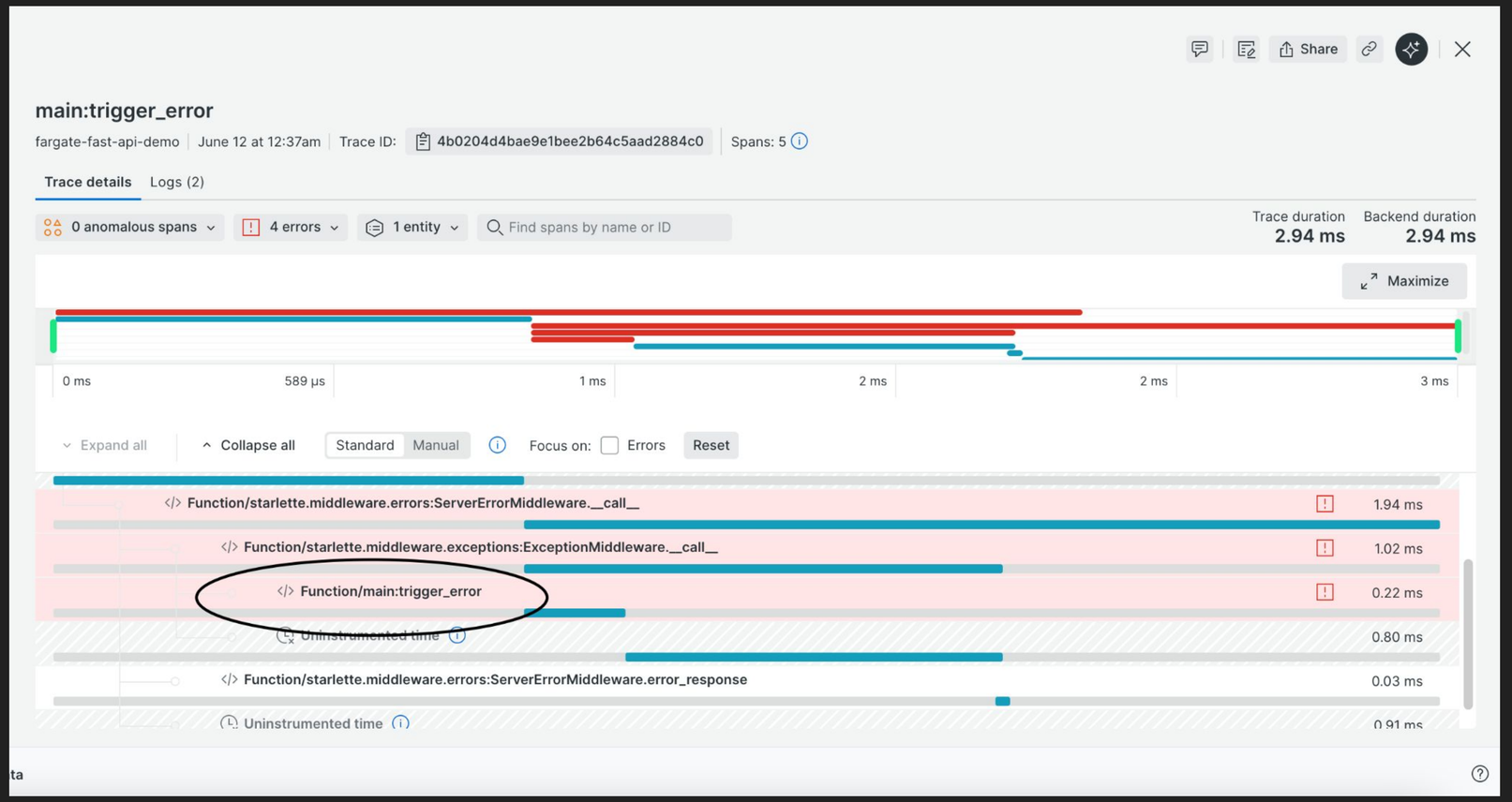
The Breakthrough

“New Relic pulls the clues together.”

- Correlation engine: New Relic automatically correlates logs, traces, metrics, and errors.
- Unified view: Presents a cohesive narrative from scattered observability data.
- Real-time insights: Enables fast diagnosis with visual linkages across stack layers.

The Breakthrough

“New Relic pulls the clues together.”



Want to Be a Sherlock?

“Add New Relic to your Fargate stack. Solve mysteries faster.”

Startup.sh

```
#!/bin/bash
envsubst < /app/newrelic.ini > /app/newrelic_runtime.ini
export NEW_RELIC_CONFIG_FILE=/app/newrelic_runtime.ini
exec newrelic-admin run-program uvicorn main:app --host 0.0.0.0 --port 8080

```

Want to Be a Sherlock?

Dockerfile

```
FROM python:3.11-slim

WORKDIR /app

# Install system dependencies
RUN apt-get update && apt-get install -y wget procps gettext && \
    rm -rf /var/lib/apt/lists/*

# Install Python dependencies including New Relic agent
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt newrelic

# Copy app code and config
COPY . .

# Copy static New Relic config file with placeholders
COPY newrelic.ini /app/newrelic.ini

# Copy custom startup script
COPY start.sh /app/start.sh
RUN chmod +x /app/start.sh

# Use entrypoint script to configure and run app
CMD ["/app/start.sh"]

```

Why Not X-Ray?

- AWS X-Ray — Great for AWS Service-Level Tracing
- Smoothly visualizes service maps for AWS API Gateway, Lambda, DynamoDB, ECS, etc
- Provides trace-level insight into AWS-managed services
- Lack of all programming language supports
- Still new

The Case Closed

- Reduced MTTR: New Relic cuts mean time to resolution by correlating clues instantly.
- Proactive diagnostics: Go beyond alerts—understand root causes before users are impacted.
- Mission accomplished: Detective work complete; issue understood, explained, and resolved.

Photo by Agnieszka De Val on Unsplash

“As any good detective knows, preventing crime is as important as solving it”

Implementing Proactive Measures

Key strategies

- Implement comprehensive logging, tracing, and status monitoring from the start.
- Set up proactive alerting using CloudWatch Alarms.
- Use AWS Fargate Platform latest version or later for enhanced debugging capabilities.
- Regularly review and optimize your task definitions.
- Embrace Infrastructure as Code for consistent, reproducible deployments.
- Implement and regularly check your application's status page for early warning signs.

“In the end, it’s not just about solving the mystery—it’s about preventing the next one.”

Thank You!

To Connect with me scan this code