

Function Faster With New Relic Serverless for AWS Lambda

Bring the power of New Relic One to Serverless on AWS

Amazon Web Services (AWS) Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume—there is no charge when your code is not running.

With Lambda, you can run code for virtually any type of application or backend service, all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

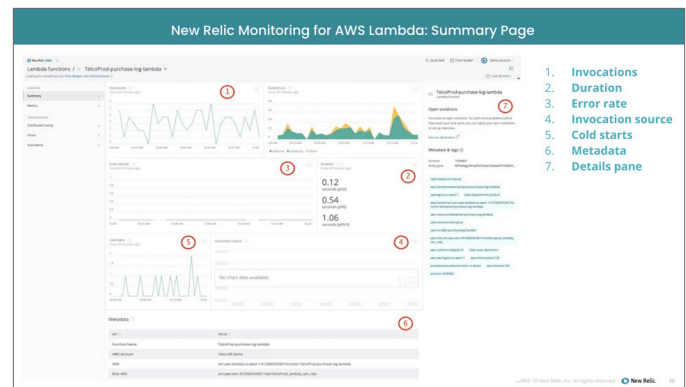
This approach comes with a cost, however, and that cost is complexity: multiple languages, many services, and myriad paths through the system.

Building software has never been easier—but monitoring it has never been harder

While it has never been easier to build software, understanding and monitoring your systems has never been more difficult. The benefits of architectures using this approach are clear, yet the tools used to build, monitor, and troubleshoot these composite applications haven't kept up.

New Relic Monitoring for AWS Lambda's curated experience provides for:

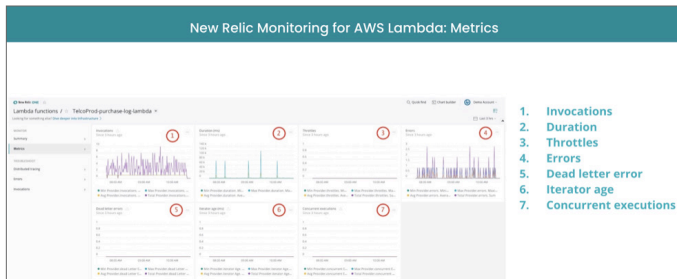
- **Faster troubleshooting and alerting:** When errors occur in your AWS Lambda application, you need alerts and immediate insights into those errors; you can't waste time diving into logs to troubleshoot complex outages.
- **Managing application scale:** Scaling a DIY solution can be full-time work. You'll save time and resources with tooling built to handle your growing workloads.
- **Controlling complex environments:** This is especially important if you manage a distributed system running AWS Lambda functions alongside other services, whether they are legacy services, or more modern components built on Kubernetes or other container-based environments. You need to see everything at a high level while also having the ability to drill down into individual errors quickly when something goes wrong. And this all needs to happen in one place, in real time—using a single tool, on one platform.



Complete observability into your serverless applications with New Relic

Modern software developers need complete observability into their serverless applications, with the same depth and breadth they've come to expect in their legacy, monolithic apps. Key data for this observability includes:

- 1. Metrics to understand system health:** Aggregate metrics are essential to understand what's happening at the highest level.



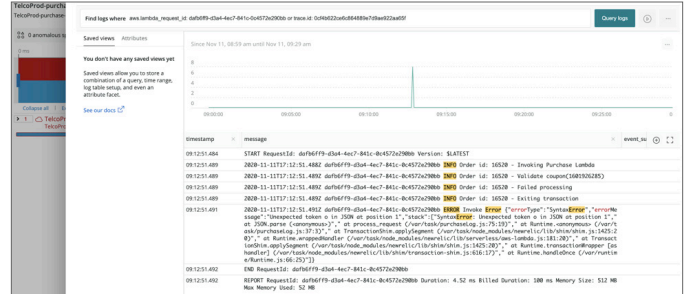
- 2. Traces of end-to-end paths:** When you discover a problem, whether from an alert, a support ticket, or an observation you make, you need to understand the end-to-end path through your distributed system. Looking at these traces helps you better understand the data flows and identify offending components, even if they are far downstream.

```

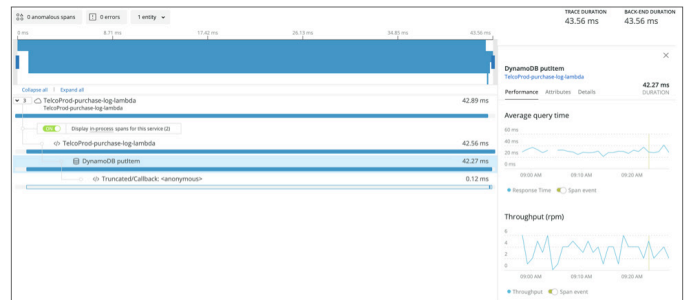
Copy to clipboard
at                                     JSON.parse <anonymous>
  at /var/task/purchaseLog.js           75:19 process_request
  at /var/task/purchaseLog.js           37:13 Runtime.<anonymous>
  at /var/task/node_modules/newrelic/lib/shim/shim.js 1425:20 TransactionShim.applySegment
  at /var/task/node_modules/newrelic/lib/serverless/aws-... 181:20 Runtime.wrappedHandler
  at /var/task/node_modules/newrelic/lib/shim/shim.js           1425:20 TransactionShim.applySegment
  at /var/task/node_modules/newrelic/lib/shim/transaction... 616:17 Runtime.transactionWrapper [as handler]
  at /var/runtime/Runtime.js           66:25 Runtime.handleOnce
    
```

ATTRIBUTE	VALUE
AWS function name	TelcoProd-purchase-log-lambda
AWS handler	purchaseLog.handler
AWS runtime	nodejs12.x
Trace ID	0cf4b622e6c864889e7d9ae922aa65f
Sampled	true
Timestamp	November 11, 2020 12:12:51.000
Request method	POST
Request URI	/TelcoProd_gateway_api_resource
Request headers content type	application/json
Request headers host	bimiyokigc4.execute-api-us-west-1.amazonaws.com
Request headers user agent	Ruby
Error class	SyntaxError
Error message	Unexpected token o in JSON at position 1

- 3. Detailed data for root cause analysis:** Once you've found the source, you need to find the cause. Rich, code-level, HTTP request details and errors in the context of your metrics and traces enable you to determine what is wrong. You shouldn't have to download a bunch of logs across multiple functions and services, aggregate them, and start running custom queries to get the answer.



- 4. Relationships for context:** Serverless, highly iterative systems experience lots of change, and that change frequently occurs in unplanned ways. There could be unexpected or unknown upstream or downstream systems causing an issue. When a problem arises, you need to see the full context around the system.



Serverless monitoring made easy with New Relic One

Infrastructure flexibility on AWS has led to an evolution from microservices to what are essentially nanoservice architectures. While organizations may maintain a variety of microservices with internal or public facing APIs,

each of those microservices may comprise many other service components, all of which can be modified and scaled independently.

While this flexibility can increase agility, it can increase complexity making observability into your environment that much more difficult. New Relic's curated serverless monitoring experience reduces this complexity, allowing you to focus on your business itself, rather than focusing on the underlying software components that drive your business.

“Customers can expect a great experience with Chegg, because New Relic gives our engineers insights on the behavior of their applications, so they can continuously improve that experience.”

Steve Evans

Vice President Engineering Services, Chegg
