Debugging AWS Lambda Performance Issues

lumigo

Yan Cui

http://theburningmonk.com
@theburningmonk
yan@lumigo.io

AWS user since 2009

aws serverless HERO since 2018

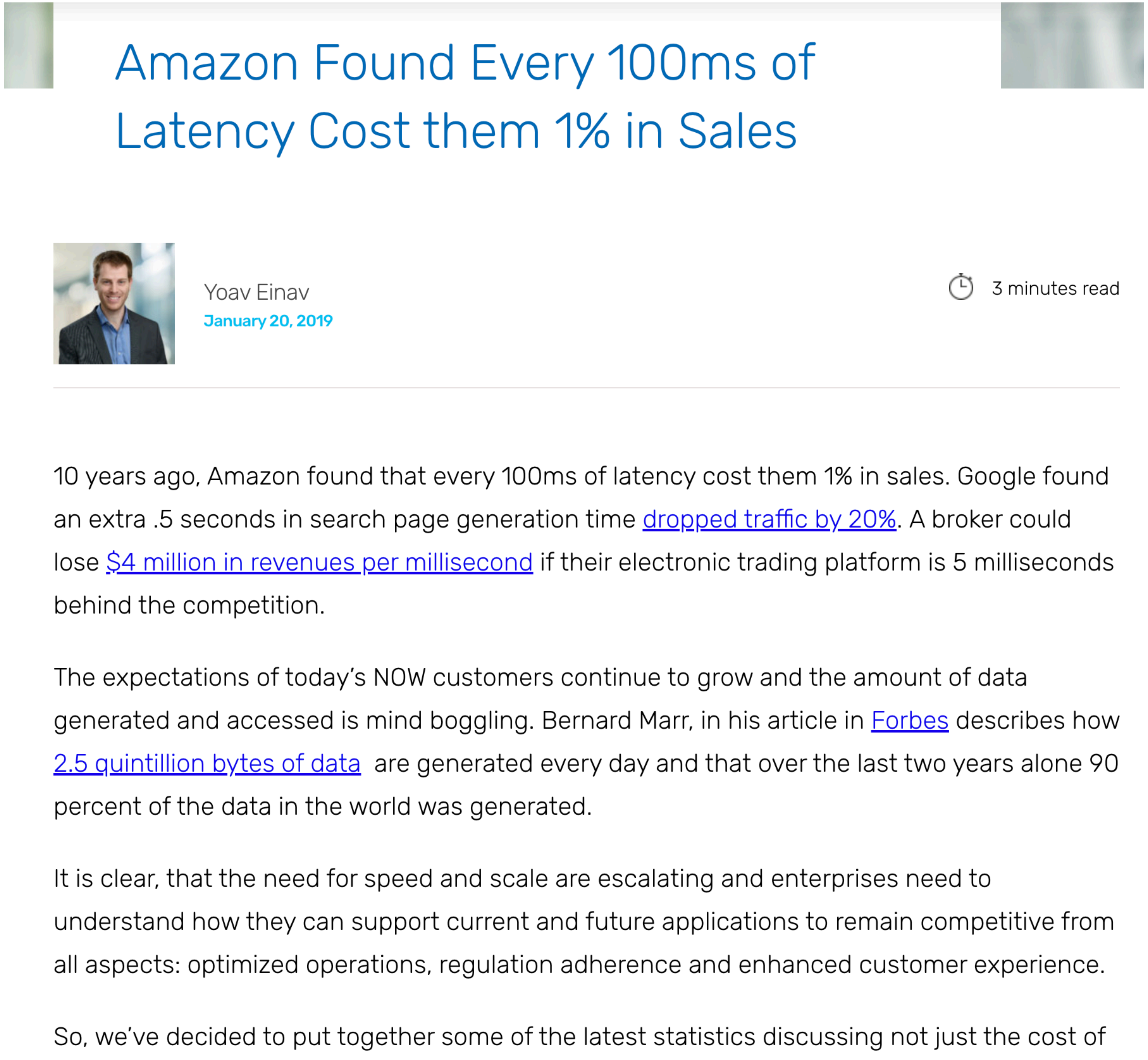
Developer Advocate @ lumigo

Independent Consultant

Aviad Mor

@aviadmor
aviad@lumigo.io

Co-Founder & CTO @ lumigo



Amazon Found Every 100ms of Latency Cost them 1% in Sales

Yoav Einav
January 20, 2019

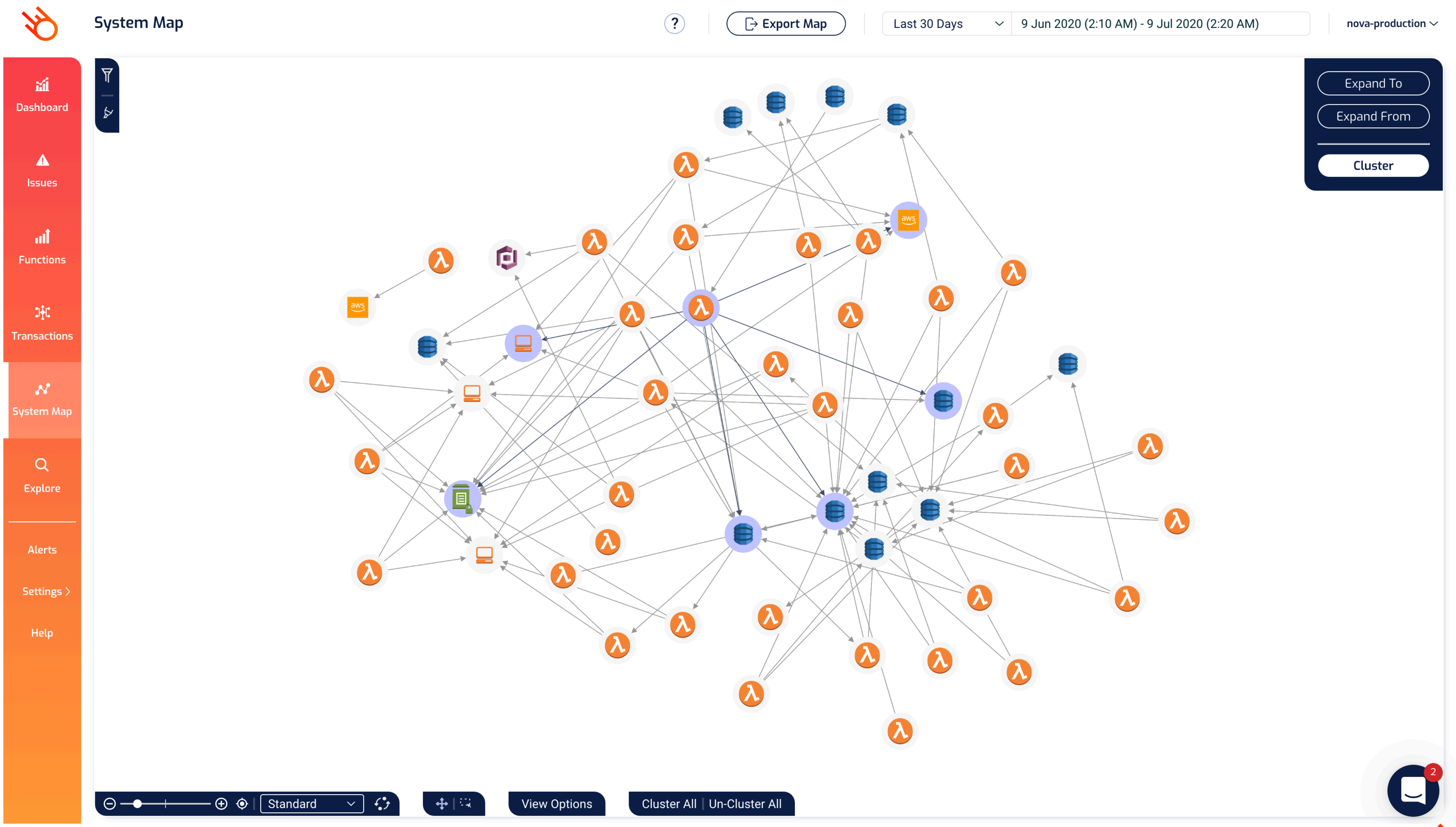
3 minutes read

10 years ago, Amazon found that every 100ms of latency cost them 1% in sales. Google found an extra .5 seconds in search page generation time dropped traffic by 20%. A broker could lose $4 million in revenues per millisecond if their electronic trading platform is 5 milliseconds behind the competition.

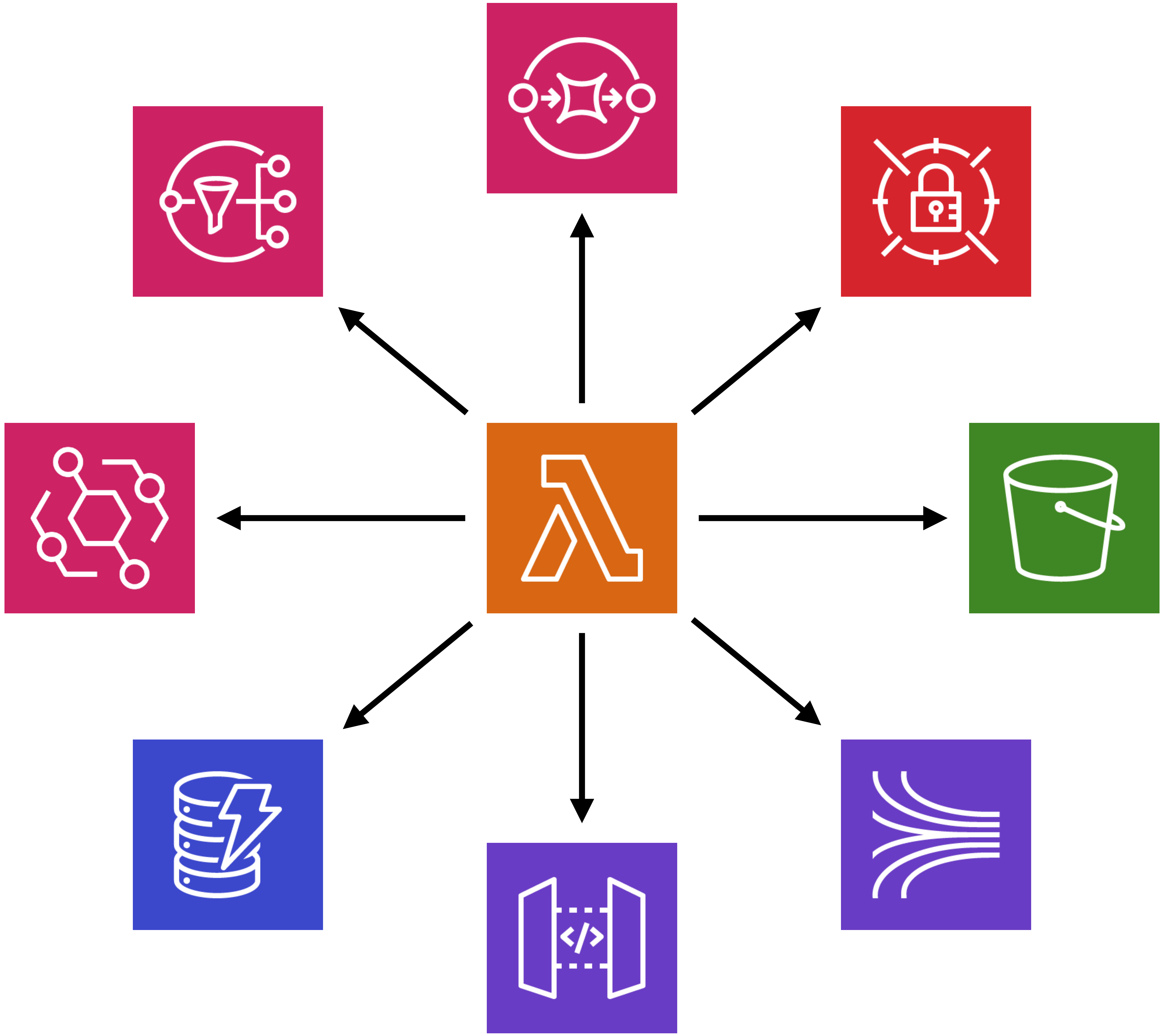
The expectations of today's NOW customers continue to grow and the amount of data generated and accessed is mind boggling. Bernard Marr, in his article in Forbes describes how 2.5 quintillion bytes of data are generated every day and that over the last two years alone 90 percent of the data in the world was generated.

It is clear, that the need for speed and scale are escalating and enterprises need to understand how they can support current and future applications to remain competitive from all aspects: optimized operations, regulation adherence and enhanced customer experience.

So, we've decided to put together some of the latest statistics discussing not just the cost of


System Map
?
Export Map
Last 30 Days
9 Jun 2020 (2:10 AM) - 9 Jul 2020 (2:20 AM)
nova-production
Dashboard
Issues
Functions
Transactions
System Map
Explore
Alerts
Settings >
Help
Expand To
Expand From
Cluster
Standard
View Options
Cluster All | Un-Cluster All
2

lumigo



lumigo

observation

majority of performance problems originates from a function’s integration points

macro

how well is this service performing in general?

micro

why did this transaction perform poorly?

macro

identify systemic issues

how well is this service performing in general?

micro

why did this transaction perform poorly?

macro

how well is this service performing in general?

micro

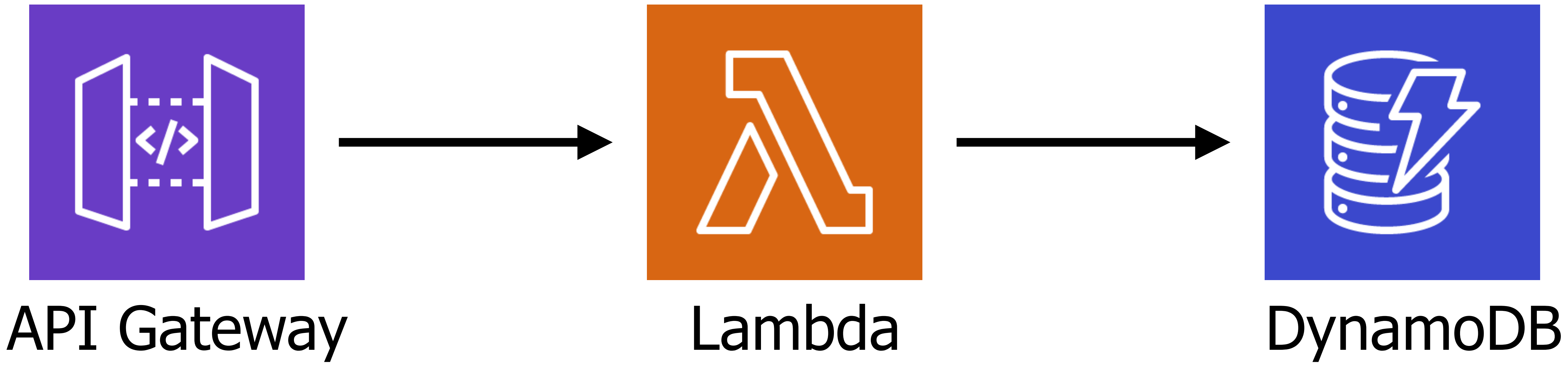
why did this transaction perform poorly?

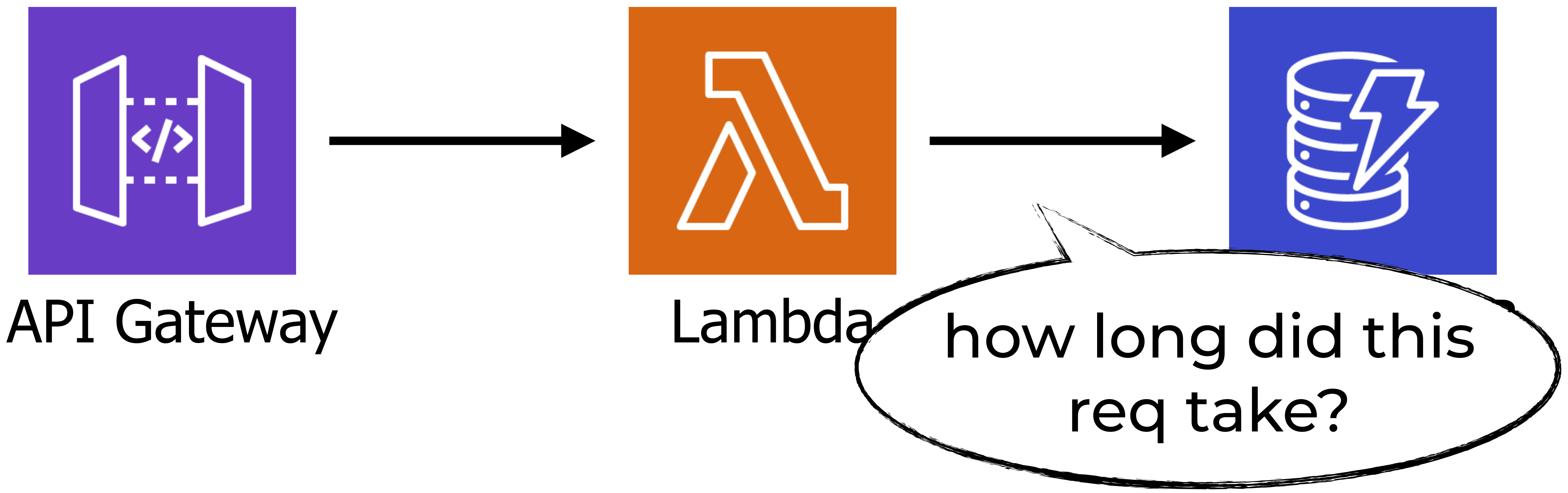
why did this user
get a bad exp?

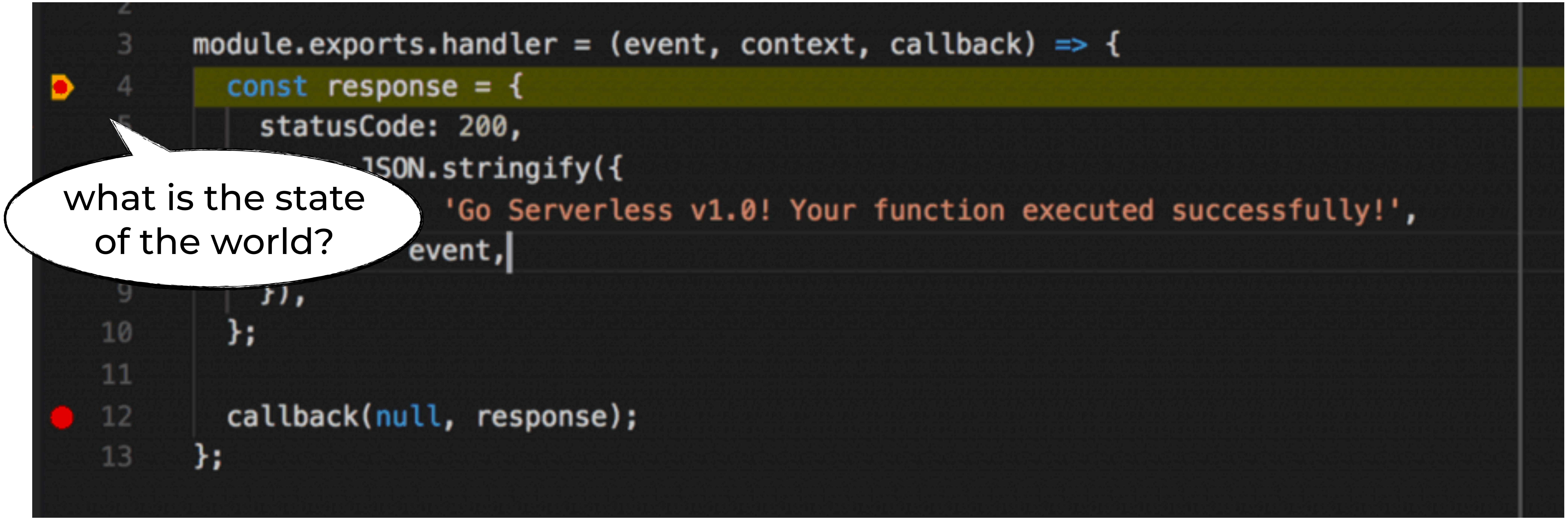
In control theory, observability is a measure of how well **internal states** of a system can be inferred from knowledge of its external outputs.

In control theory, observability is a measure of how well **internal states** of a system can be inferred from knowledge of its external outputs.

what do we need
to collect?


API Gateway
Lambda
DynamoDB


API Gateway
Lambda
how long did this req take?



```
module.exports.handler = (event, context, callback) => {
  const response = {
    statusCode: 200,
        JSON.stringify({
      'Go Serverless v1.0! Your function executed successfully!',
      event,
    }),
  };

  callback(null, response);
};
```

In control theory, observability is a measure of how well **internal states** of a system can be inferred from knowledge of its external outputs.

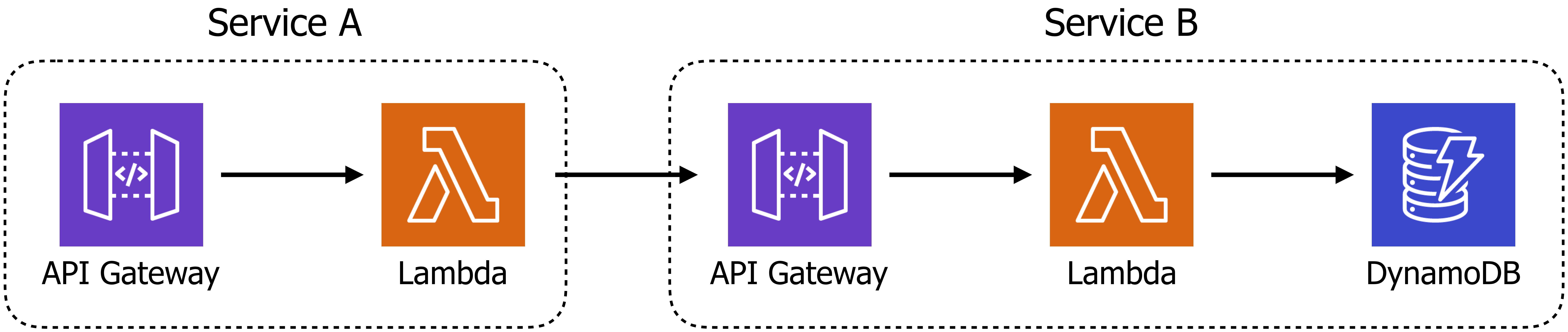
what are the most important outputs to collect?

macro

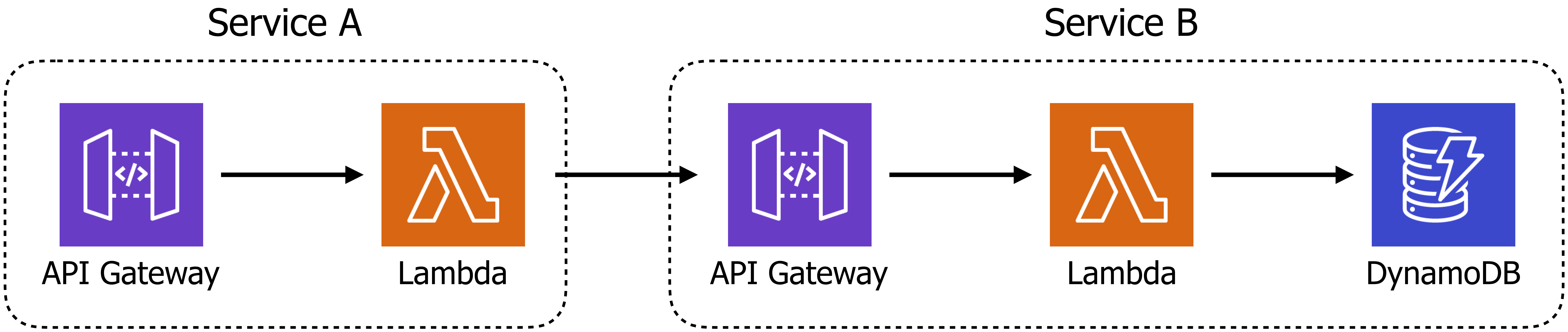
how well is this service performing in general?

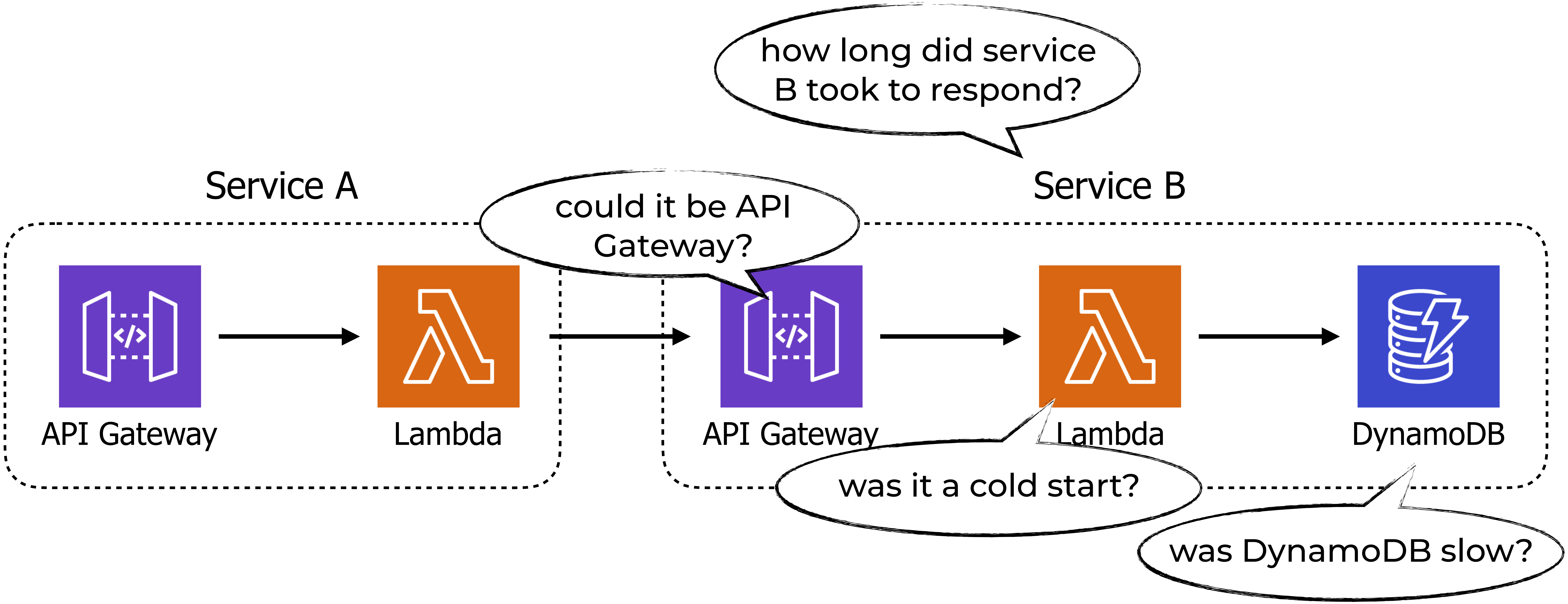
micro

why did this transaction perform poorly?


Service A
Service B
API Gateway
Lambda
API Gateway
Lambda
DynamoDB

how long did service B took to respond?



how long did service B took to respond?
Service A
could it be API Gateway?
Service B
API Gateway
Lambda
API Gateway
Lambda
DynamoDB
was it a cold start?
was DynamoDB slow?

lumigo

How do we collect these external output to help us infer the application's internal state to debug performance issues?

log them manually

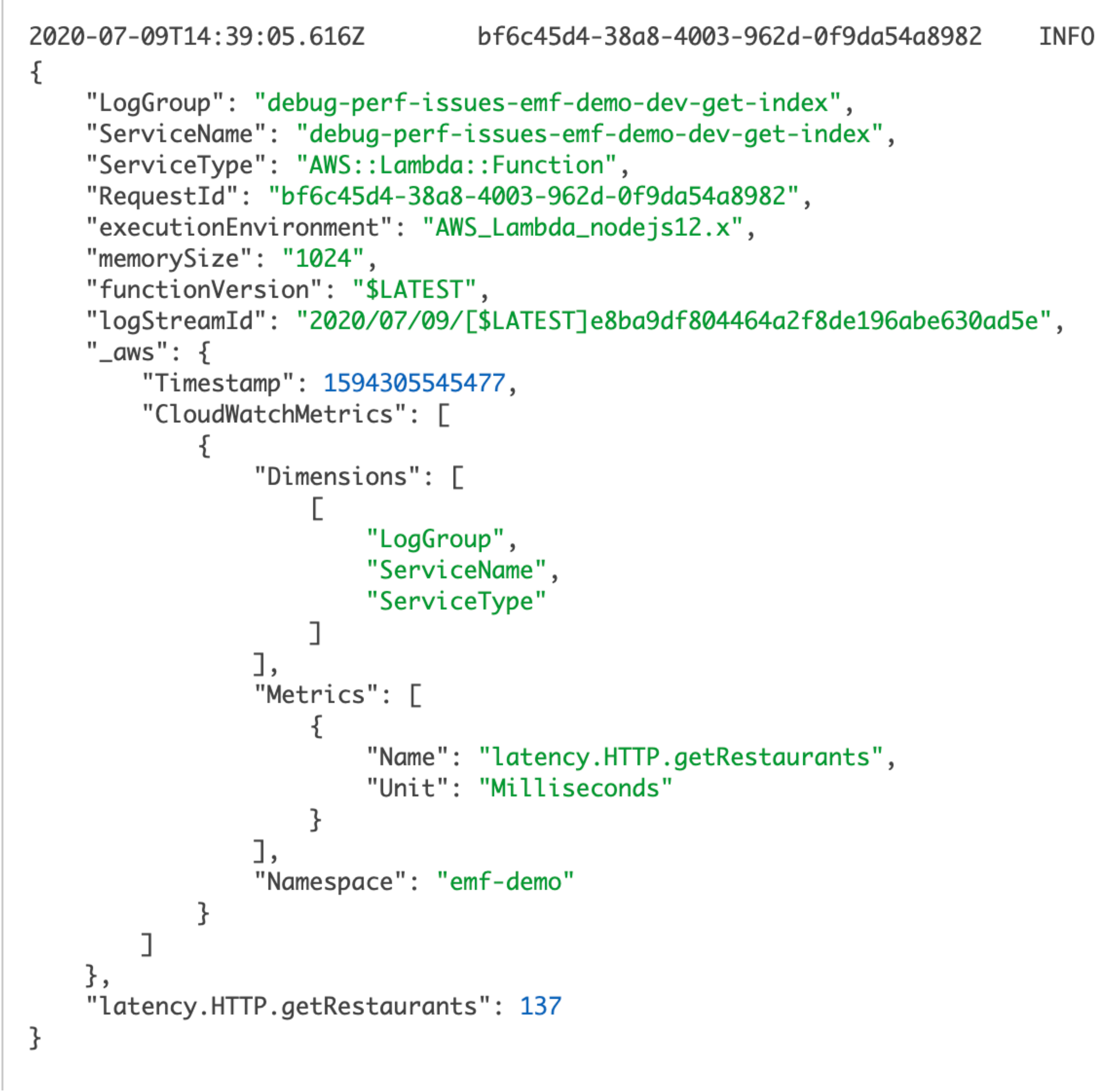
```
const start = Date.now()
await doSomething()
const end = Date.now()
const latency = end - start
console.log(`doSomething took ${latency}ms`)
```

log them manually

```
const start = Date.now()
await doSomething()
const end = Date.now()
const latency = end - start
console.log(`doSomething took ${latency}ms`)
```

turn these into metrics
using **Metric Filter**

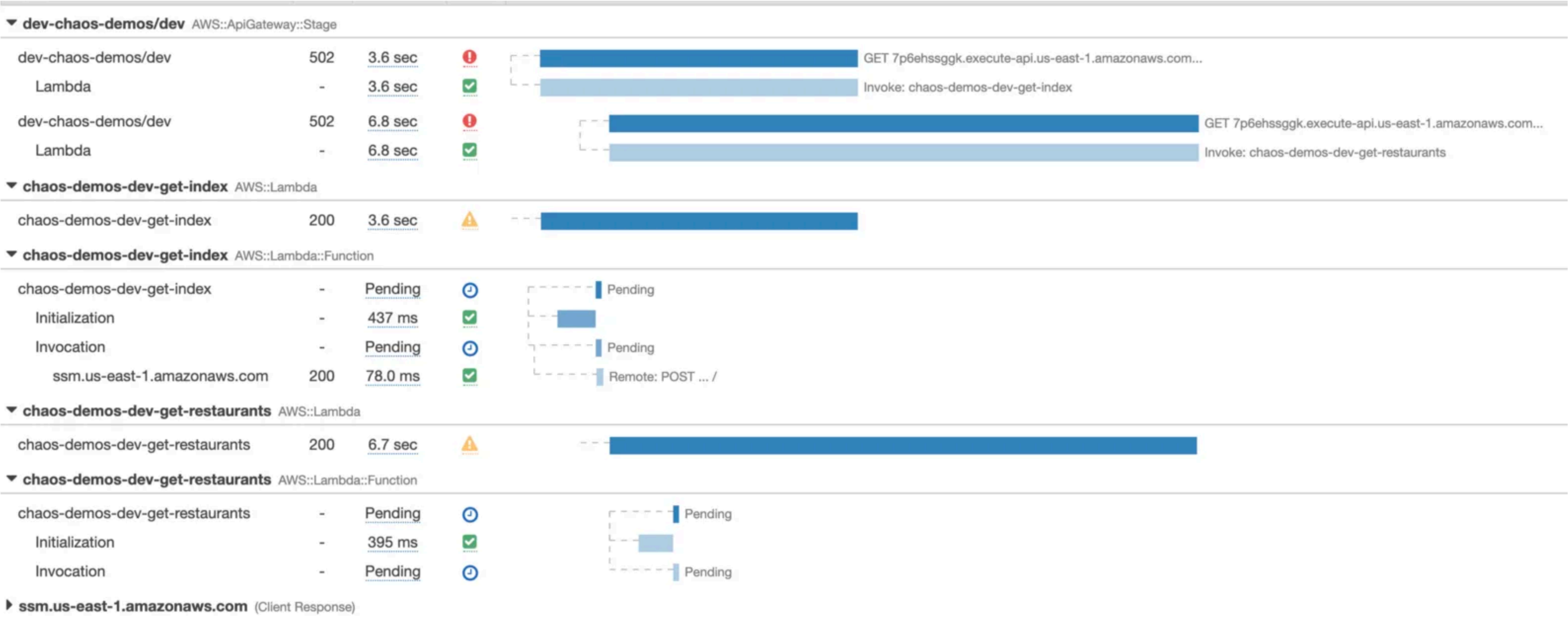
Embedded Metric Format (EMF)



```
2020-07-09T14:39:05.616Z        bf6c45d4-38a8-4003-962d-0f9da54a8982    INFO
{
    "LogGroup": "debug-perf-issues-emf-demo-dev-get-index",
    "ServiceName": "debug-perf-issues-emf-demo-dev-get-index",
    "ServiceType": "AWS::Lambda::Function",
    "RequestId": "bf6c45d4-38a8-4003-962d-0f9da54a8982",
    "executionEnvironment": "AWS_Lambda_nodejs12.x",
    "memorySize": "1024",
    "functionVersion": "$LATEST",
    "logStreamId": "2020/07/09/[$LATEST]e8ba9df804464a2f8de196abe630ad5e",
    "_aws": {
        "Timestamp": 1594305545477,
        "CloudWatchMetrics": [
            {
                "Dimensions": [
                    [
                        "LogGroup",
                        "ServiceName",
                        "ServiceType"
                    ]
                ],
                "Metrics": [
                    {
                        "Name": "latency.HTTP.getRestaurants",
                        "Unit": "Milliseconds"
                    }
                ],
                "Namespace": "emf-demo"
            }
        ]
    },
    "latency.HTTP.getRestaurants": 137
}
```

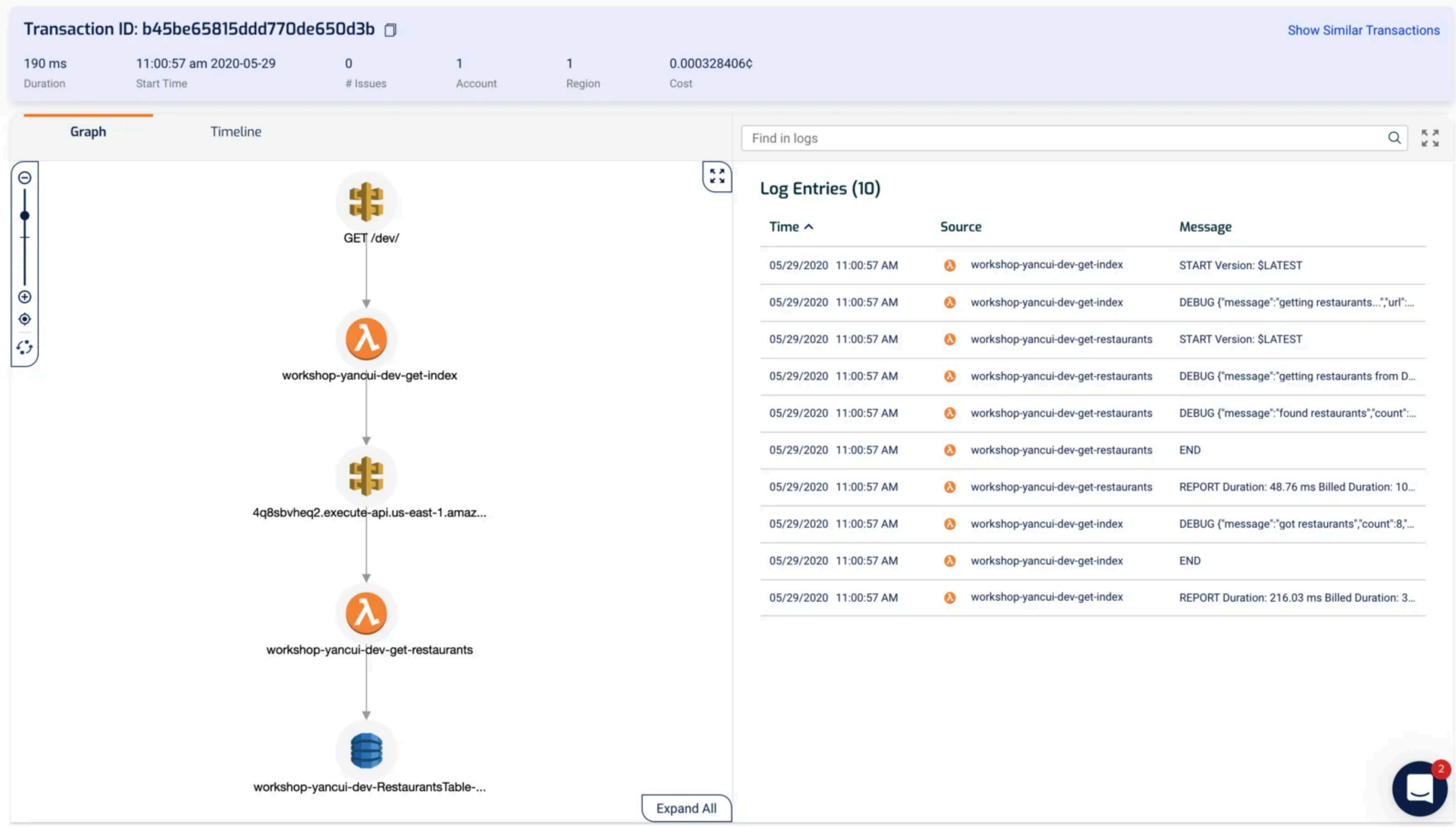
intercept all HTTP requests

X-Ray



▼ **dev-chaos-demos/dev** AWS::ApiGateway::Stage				
dev-chaos-demos/dev	502	3.6 sec	❗	GET 7p6ehssggk.execute-api.us-east-1.amazonaws.com...
Lambda	-	3.6 sec	☑	Invoke: chaos-demos-dev-get-index
dev-chaos-demos/dev	502	6.8 sec	❗	GET 7p6ehssggk.execute-api.us-east-1.amazonaws.com...
Lambda	-	6.8 sec	☑	Invoke: chaos-demos-dev-get-restaurants
▼ **chaos-demos-dev-get-index** AWS::Lambda				
chaos-demos-dev-get-index	200	3.6 sec	⚠	
▼ **chaos-demos-dev-get-index** AWS::Lambda::Function				
chaos-demos-dev-get-index	-	Pending	🕘	Pending
Initialization	-	437 ms	☑	
Invocation	-	Pending	🕘	Pending
ssm.us-east-1.amazonaws.com	200	78.0 ms	☑	Remote: POST ... /
▼ **chaos-demos-dev-get-restaurants** AWS::Lambda				
chaos-demos-dev-get-restaurants	200	6.7 sec	⚠	
▼ **chaos-demos-dev-get-restaurants** AWS::Lambda::Function				
chaos-demos-dev-get-restaurants	-	Pending	🕘	Pending
Initialization	-	395 ms	☑	
Invocation	-	Pending	🕘	Pending
▶ **ssm.us-east-1.amazonaws.com** (Client Response)				

Lumigo



Transaction ID: b45be65815ddd770de650d3b

Show Similar Transactions

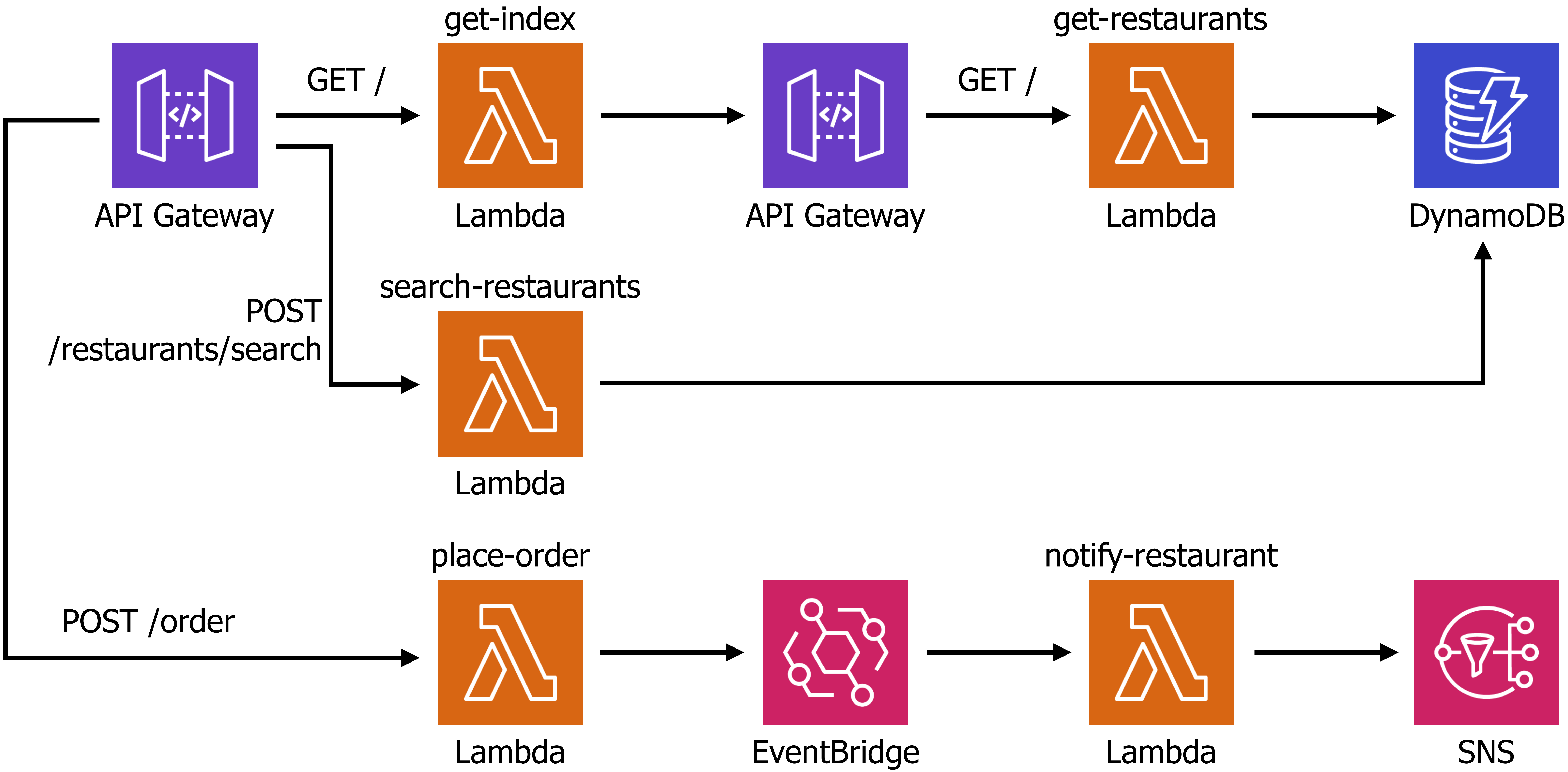
190 ms	11:00:57 am 2020-05-29	0	1	1	0.000328406¢
Duration	Start Time	# Issues	Account	Region	Cost

Graph
Timeline
GET /dev/
workshop-yancui-dev-get-index
4q8sbvheq2.execute-api.us-east-1.amaz...
workshop-yancui-dev-get-restaurants
workshop-yancui-dev-RestaurantsTable-...
Expand All
Find in logs
2

Log Entries (10)

Time ^	Source	Message
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-index	START Version: $LATEST
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-index	DEBUG {"message":"getting restaurants...","url":...
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-restaurants	START Version: $LATEST
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-restaurants	DEBUG {"message":"getting restaurants from D...
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-restaurants	DEBUG {"message":"found restaurants","count":...
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-restaurants	END
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-restaurants	REPORT Duration: 48.76 ms Billed Duration: 10...
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-index	DEBUG {"message":"got restaurants","count":8,"...
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-index	END
05/29/2020 11:00:57 AM	workshop-yancui-dev-get-index	REPORT Duration: 216.03 ms Billed Duration: 3...

DEMO!



get-index

get-restaurants

GET /

GET /

API Gateway

Lambda

API Gateway

Lambda

DynamoDB

search-restaurants

POST
/restaurants/search

Lambda

place-order

notify-restaurant

POST /order

Lambda

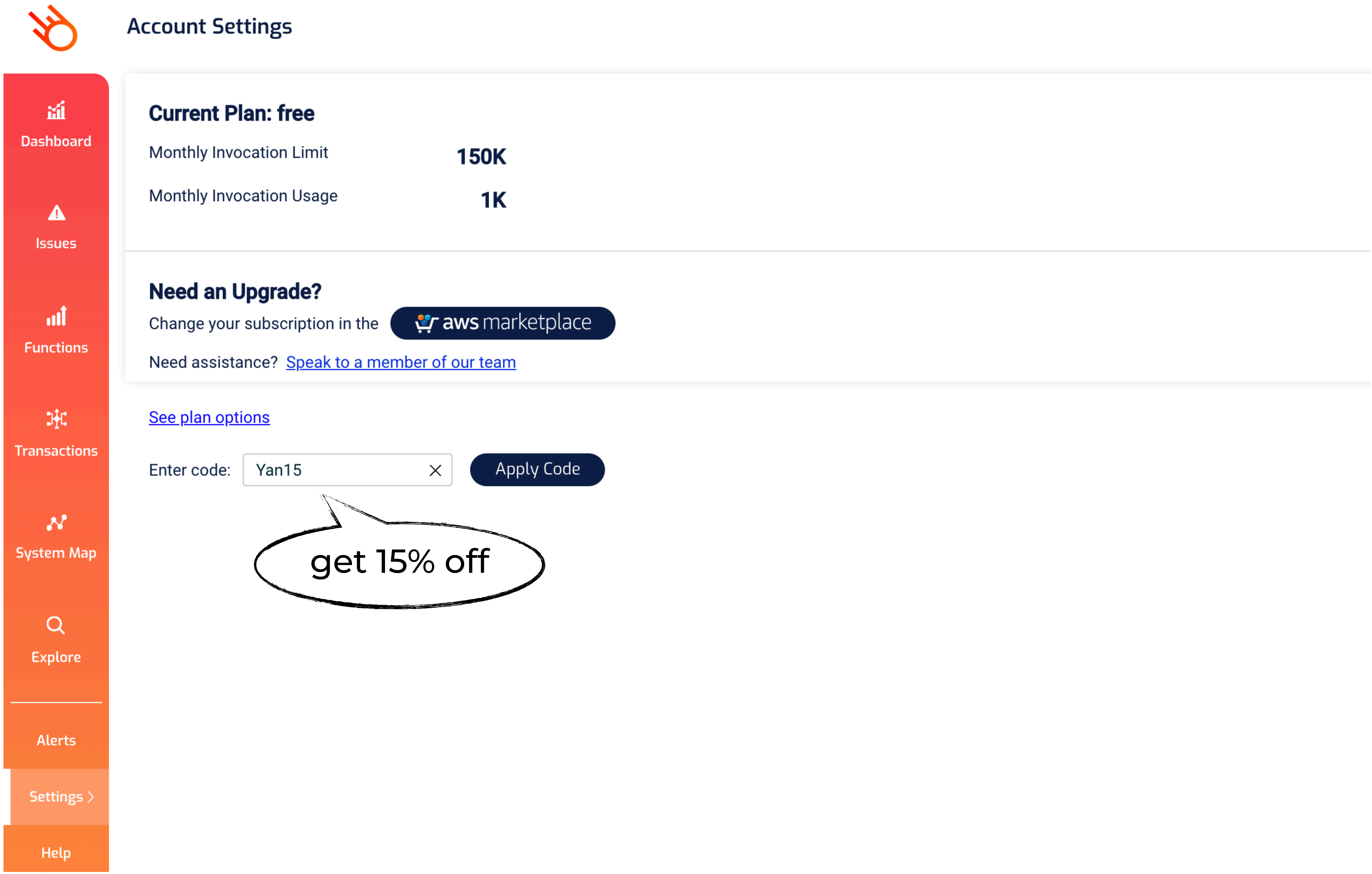
EventBridge

Lambda

SNS

lumigo

github.com/theburningmonk/debugging-lambda-perf-issues-demo



Account Settings

Current Plan: free

Monthly Invocation Limit	150K
Monthly Invocation Usage	1K

Need an Upgrade?

Change your subscription in the aws marketplace

Need assistance? Speak to a member of our team

See plan options

Enter code: Yan15

get 15% off

"That's all Folks!"

@theburningmonk

theburningmonk.com

github.com/theburningmonk

yan@lumigo.io

lumigo