
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

SVS304

Understanding and implementing observability for serverless

Lia Vader (she/her)

Sr. Solutions Architect
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

- What is observability?
- Walk through troubleshooting a real-life scenario
- Take approach we see with many customers . . . then cover proper troubleshooting methodology
- Cover how to properly implement:
    - Metrics
    - Logs
    - Traces
    - Dashboards
- Other tools that improve observability experience

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Prerequisites

- This is a 300-level talk
- Experience deploying at least 1 serverless application
- Have consumed Amazon CloudWatch metrics and configured alarms
- Note: Will not be covering partner solutions – but the concepts apply to other tooling

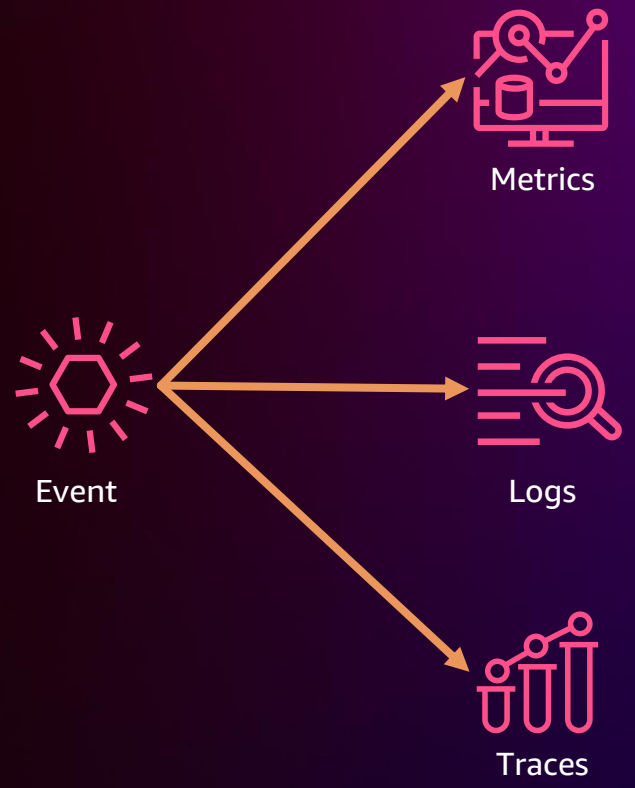
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is observability?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

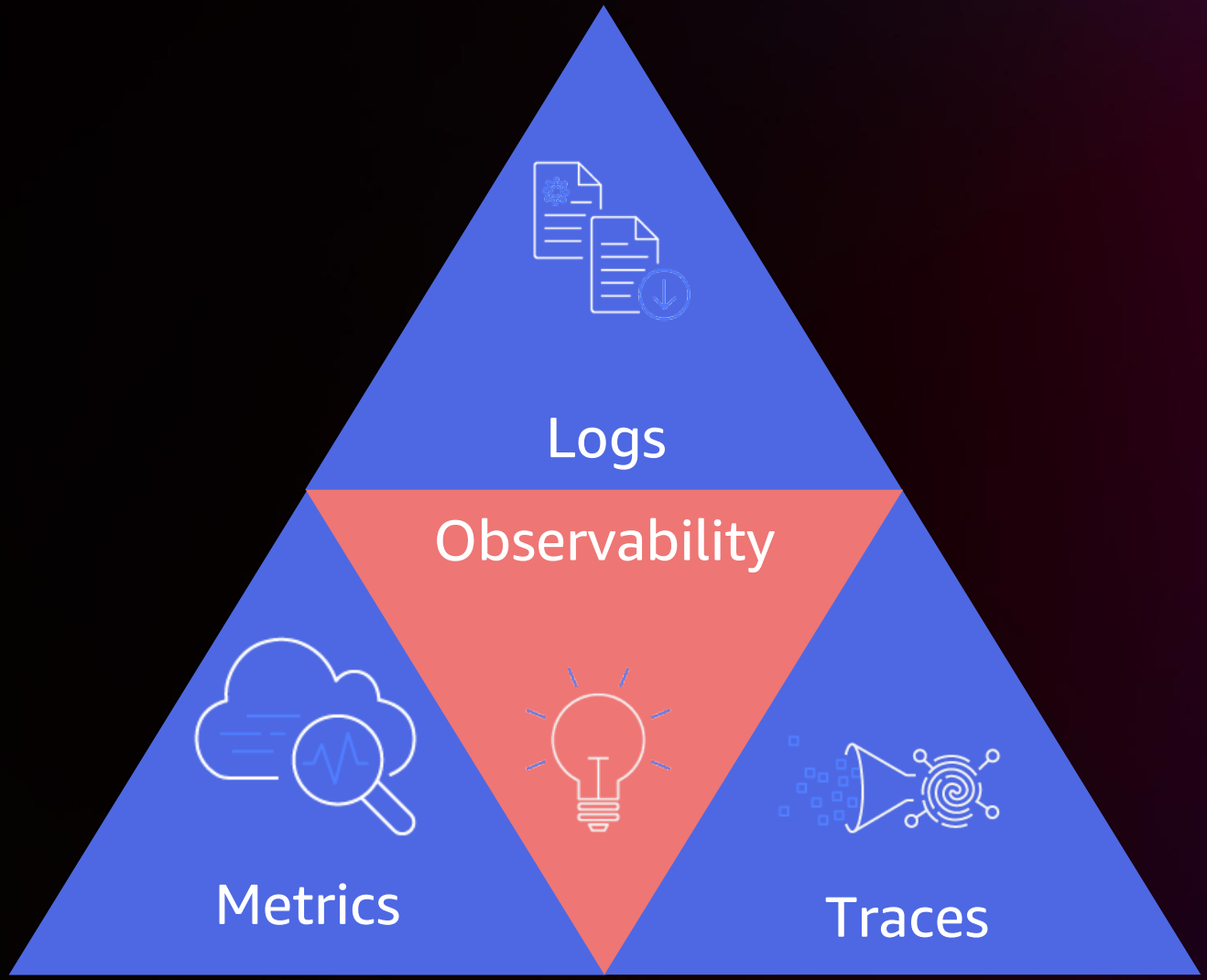
Observability

- More than just monitoring
- Events happen throughout your system
- These events are recorded via metrics, logs, and traces
- Can we correlate these events and analyze them?
- The correlation and analysis answers questions like:
  - Is my application performing as expected?
  - What is the usage?
  - What is the business impact?



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Observability is the goal

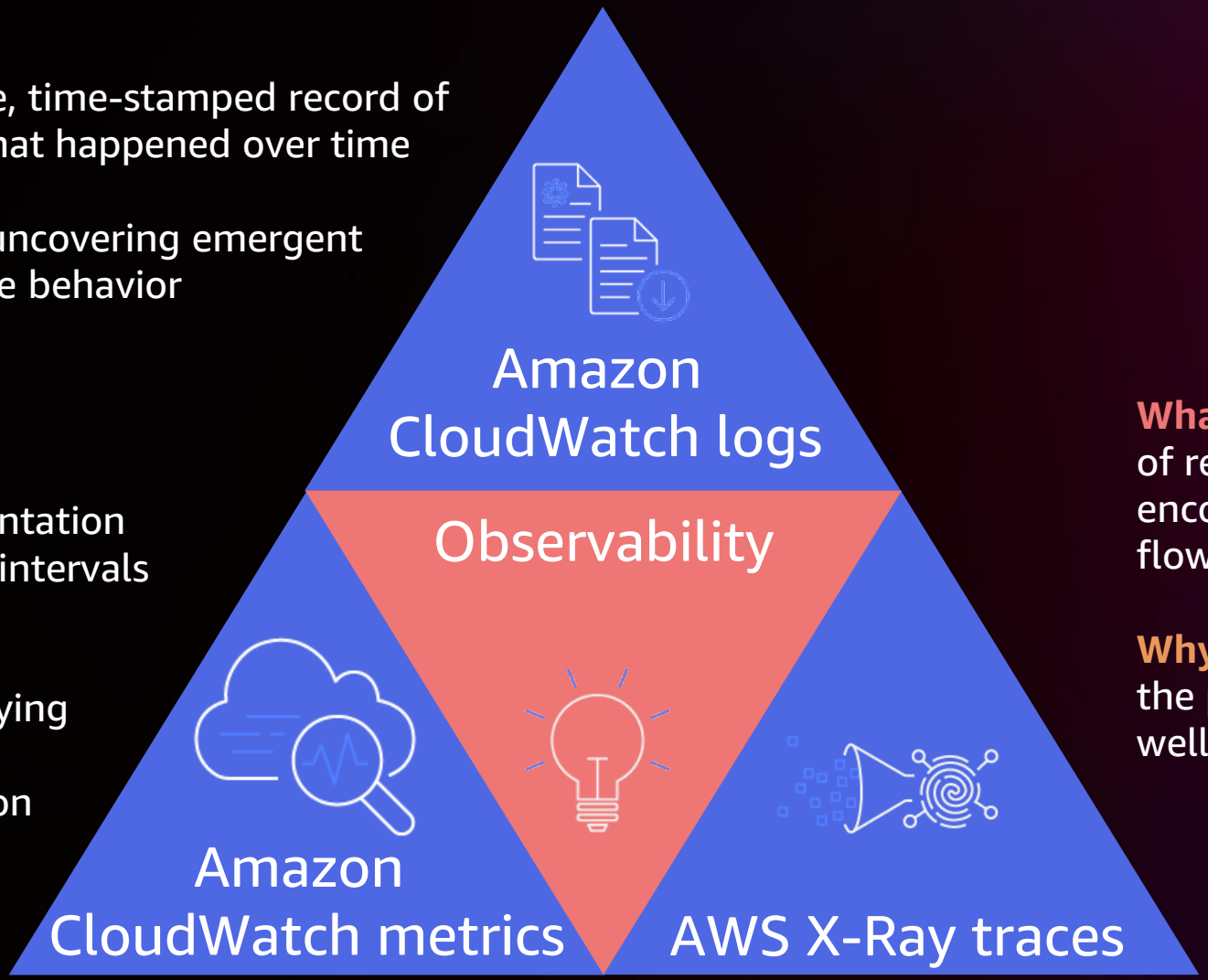


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS observability stack

What: Immutable, time-stamped record of discrete events that happened over time

Why: Useful for uncovering emergent and unpredictable behavior



What: Numeric representation of data measured over intervals of time

Why: Useful for identifying trends, mathematical modeling, and prediction

What: Representation of a series of related distributed events that encode the end-to-end request flow through a distributed system

Why: Provides visibility into both the path traversed by a request as well as the structure of a request

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Observability: AWS offerings



Amazon
CloudWatch



AWS
X-Ray

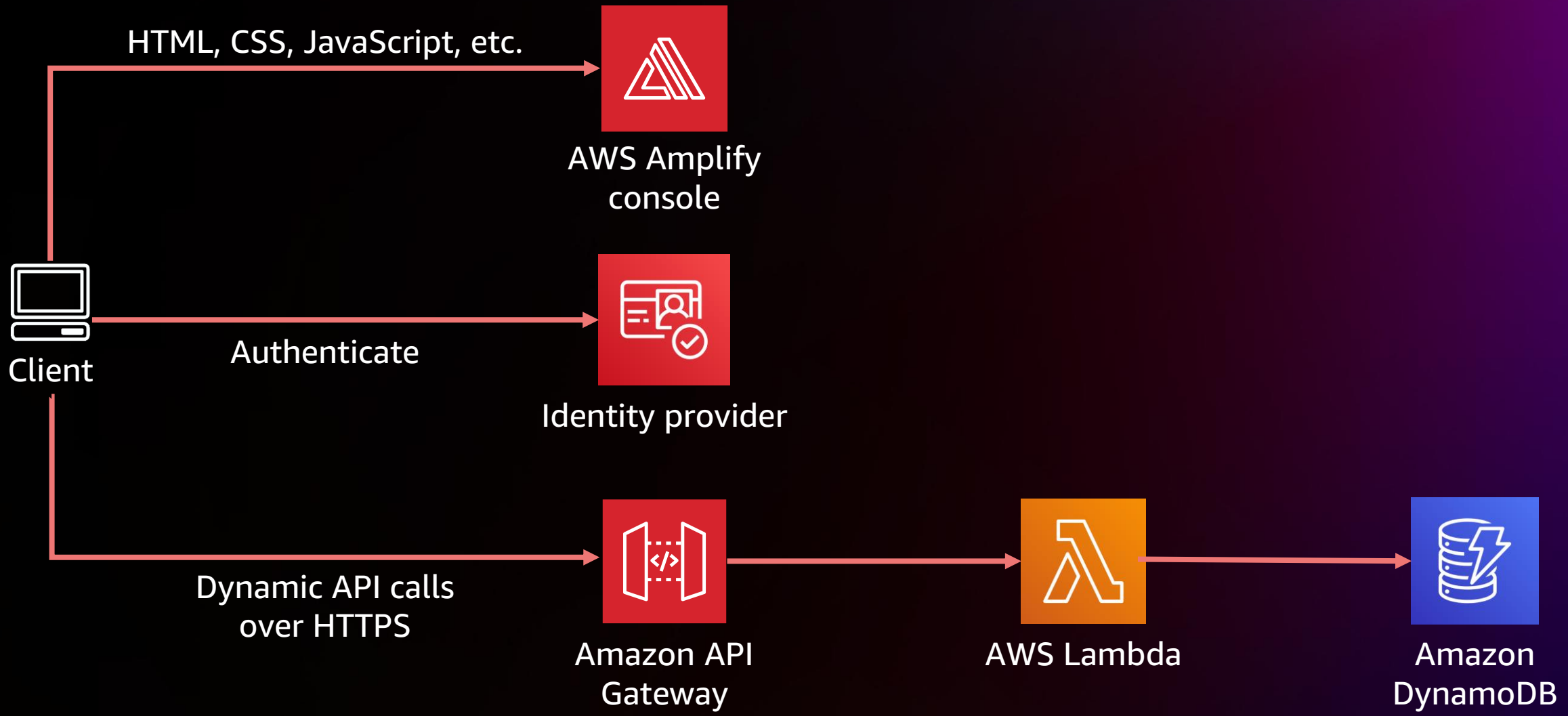
Metrics
Alarms
Logs
Dashboards
Logs Insight
ServiceLens
AWS Lambda Insights
Synthetics
Evidently
RUM

Traces
Analytics
Service map

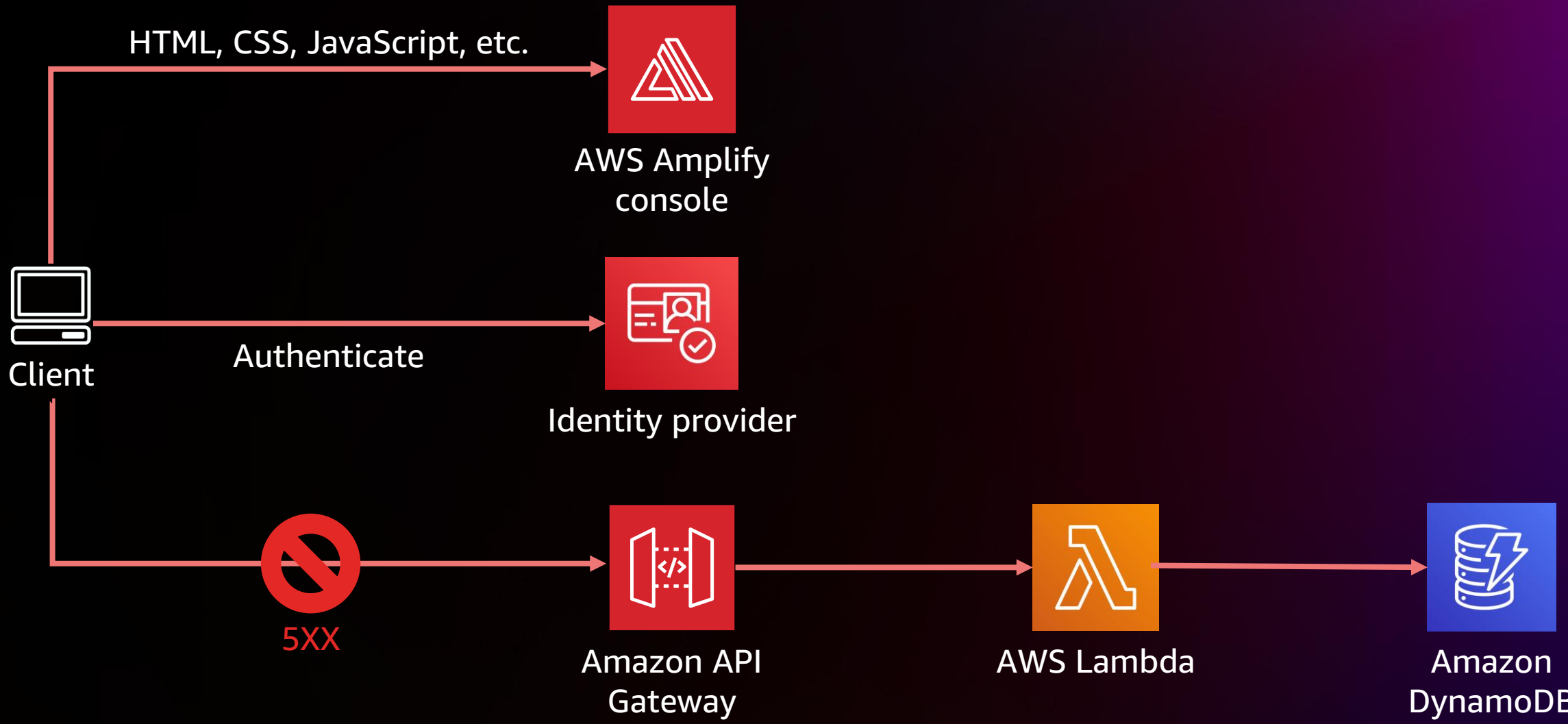
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scenario: The 5XX

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Metrics

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch built-in metrics

AWS Lambda

- **Invocation metrics**

  Invocation count, **Invocation errors**, DeadLetterErrors, DestinationDeliveryFailures, **Throttles**, ProvisionedConcurrencyInvocations, ProvisionedConcurrencySpilloverInvocations

- **Performance metrics**

  Duration, IteratorAge, PostRuntimeExtensionsDuration, OffsetLag

- **Concurrency metrics**

  ConcurrentExecutions, ProvisionedConcurrentExecutions, ProvisionedConcurrencyUtilization, UnreservedConcurrentExecutions

Amazon API Gateway

- **REST APIs**

  API request count, **Latency**, 4XXs, **5XXs**, **IntegrationLatency**, CacheHitCount, CacheMissCount

- **HTTP APIs**

  API request count, **Latency**, 4XXs, **5XXs**, **IntegrationLatency**, DataProcessed

- **WebSocket APIs**

  ConnectCount, MessageCount, **IntegrationError**, ClientError, ExecutionError, **IntegrationLatency**

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Built-in metrics often not enough

What about business/customer metrics?

Measure application performance against business goals

Revenue, signups, pictures uploaded, perceived latency, page views, etc.

How operationally stable is the application?

Continuous integration/deployment feedback time, mean time between failure and recovery, number of on-call pages and time to resolution, etc.

What about caught errors, warnings?

Caught exceptions are not counted as errors on AWS Lambda

What if I want to use other dimensions?

User ID, category, item, tags, environment, etc.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch embedded metric format

AUTOMATICALLY GENERATE METRICS FROM STRUCTURED CLOUDWATCH LOGS

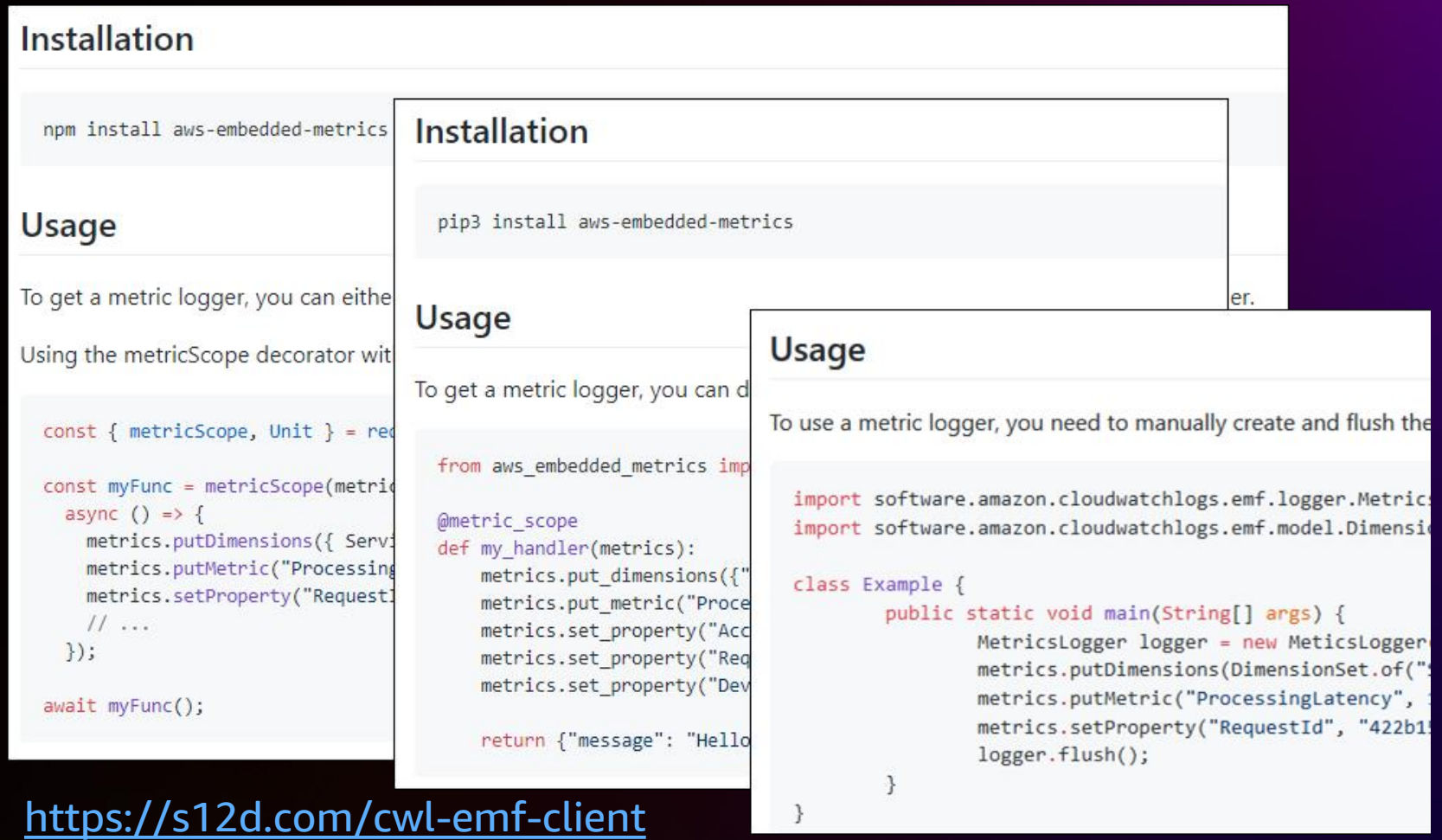
Embed custom metrics alongside detailed log event data

Can send structured format in `PutLogEvents` API call with specific format

Asynchronous

Open-source client libraries available for:

- Node.js
- Python
- Java
- C#



https://s12d.com/cwl-emf-client

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

AT A MINIMUM, MONITOR:





Latency
IntegrationLatency
5xxError
4xxError



Custom metrics

Duration
Errors
Throttling
ConcurrentExecutions
UnreservedConcurrentExecutions



ReadThrottleEvents
WriteThrottleEvents
SystemErrors
ThrottledRequests
UserErrors



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Dashboards

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

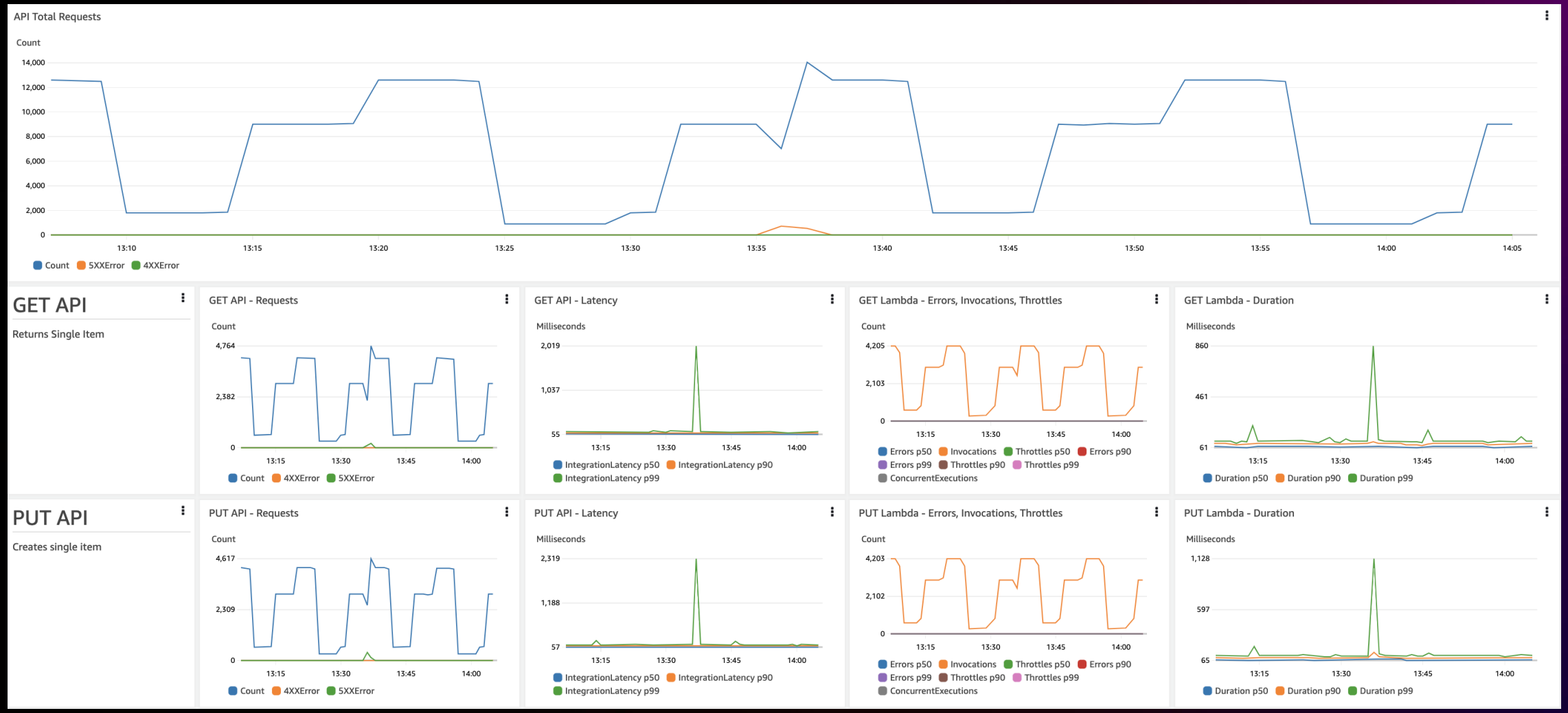
What to include in your dashboard

- System-level dashboards
  - Number of requests
  - Failures
  - Latency
- Customer-level dashboards
  - Overall service health
  - Real user metrics (RUM)
  - Canaries (CloudWatch Synthetics)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://s12d.com/bl-dashboard

CloudWatch dashboard



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch dashboard

CREATE PROGRAMMATICALLY VIA CLOUDFORMATION

- Create dashboards programmatically
- Define the DashboardBody in JSON
  - You specify your widget structures and the properties
- Include this in your CloudFormation template
- Just getting started?
  - Begin using the console to create the Dashboard
  - Click Actions, View/Edit Source
  - Copy the dashboard source, paste into your IDE
  - Add your own edits, copy and paste to keep the same metrics and statistics and only change the dimensions

```
{
    "widgets": [
        {
            "height": 6,
            "width": 5,
            "y": 8,
            "x": 13,
            "type": "metric",
            "properties": {
                "metrics": [
                    [ "AWS/ApiGateway", "IntegrationLatency", "ApiName", "myAPI", "Resource", "/item/{itemid}",
"Stage", "v1", "Method", "GET", { "stat": "p50" } ],
                    [ "...", { "stat": "p99" } ]
                ],
                "view": "timeSeries",
                "stacked": false,
                "region": "us-west-2",
                "stat": "p90",
                "period": 60,
                "title": "GET API - Latency"
            }
        },
        {
            "height": 6,
            "width": 5,
            "y": 8,
            "x": 13,
            "type": "metric",
            "properties": {
                "metrics": [
                    [ "AWS/Lambda", "Errors", "FunctionName", "GetItemFunction", { "stat": "p50" } ],
                    [ ".", "Invocations", ".", ".", { "stat": "Sum" } ],
                    [ ".", "Throttles", ".", ".", { "stat": "p50" } ],
                    [ ".", "Errors", ".", ".", { "stat": "p90" } ],
                    [ ".", "Throttles", ".", ".", { "stat": "p90" } ],
                    [ ".", "ConcurrentExecutions", ".", ".", { "stat": "Average" } ]
                ],
                "view": "timeSeries",
                "stacked": false,
                "region": "us-west-2",
                "period": 60,
                "title": "GET Lambda - Errors, Invocations, Throttles",
                "stat": "p99"
            }
        },
        {
            "height": 6,
            "width": 6,
            "y": 8,
            "x": 13,
            "type": "metric",
            "properties": {
                "metrics": [
                    [ "AWS/Lambda", "Duration", "FunctionName", "GetItemFunction", { "stat": "p50" } ],
                    [ "...", { "stat": "p90" } ]
                ],
                "view": "timeSeries",
                "stacked": false,
                "region": "us-west-2",
                "title": "GET Lambda - Duration",
                "period": 60,
                "stat": "p99"
            }
        }

    ]
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch dashboard

CDK SAMPLE

- Dashboards and widgets can be defined in AWS CDK
- Easier to reproduce across multiple applications and AWS resourses
- Editing and reading code is potentially easier than JSON

```python
from aws_cdk import aws_cloudwatch as cloudwatch

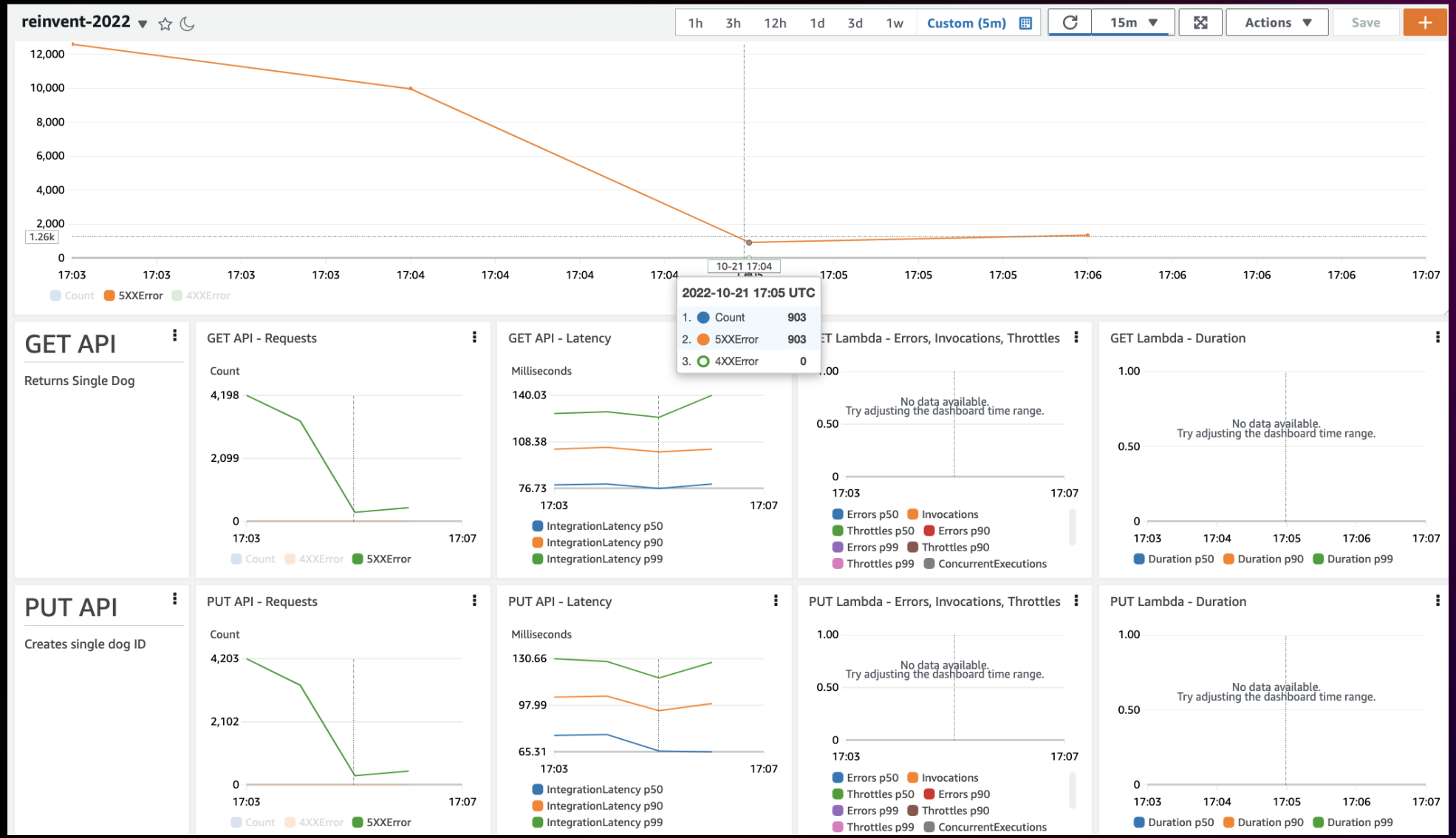
# Create CloudWatch Dashboard to view Lambda Function Metrics
cw_dashboard = aws_cloudwatch.Dashboard(self, "Lambda Dashboard",
    dashboard_name=example_dashboard_name
)
# CloudWatch Dashboard Title
title_widget = aws_cloudwatch.TextWidget(
    markdown="# Dashboard: {}".format(lambda_function.function_name),
    height=1,
    width=24
)
# Create Widgets for CloudWatch Dashboard based on Lambda Function's CloudWatch Metrics
invocations_widget = aws_cloudwatch.GraphWidget(title= "Invocations",
    left=[lambda_function.metric_invocations()],
    width=24)

errors_widget = aws_cloudwatch.GraphWidget(title= "Errors",
    left=[lambda_function.metric_errors()],
    width=24)
...

# Add Widgets to CloudWatch Dashboard
cw_dashboard.add_widgets(title_widget,
                         invocations_widget,
                         errors_widget,
                         duration_widget,
                         throttles_widget,
                         log_widget)
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Troubleshooting



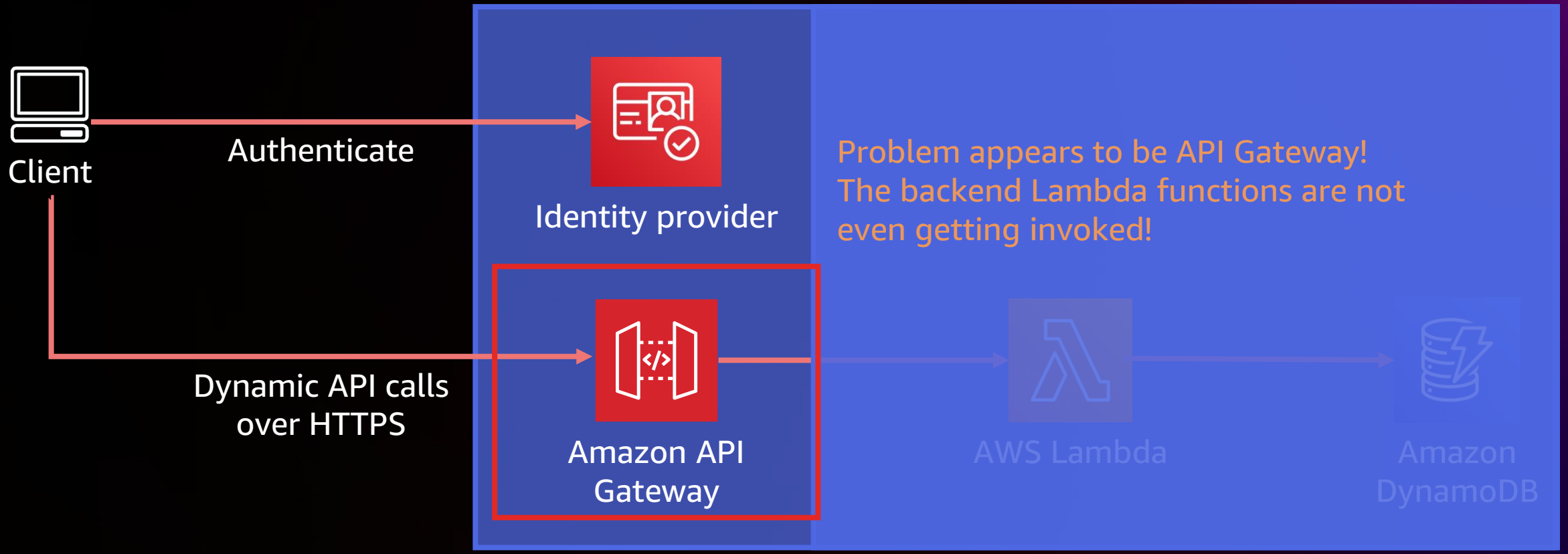
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

CREATE A VIEW OF YOUR SERVICE AND CUSTOMER EXPERIENCE USING DASHBOARDS

Metrics

Dashboard



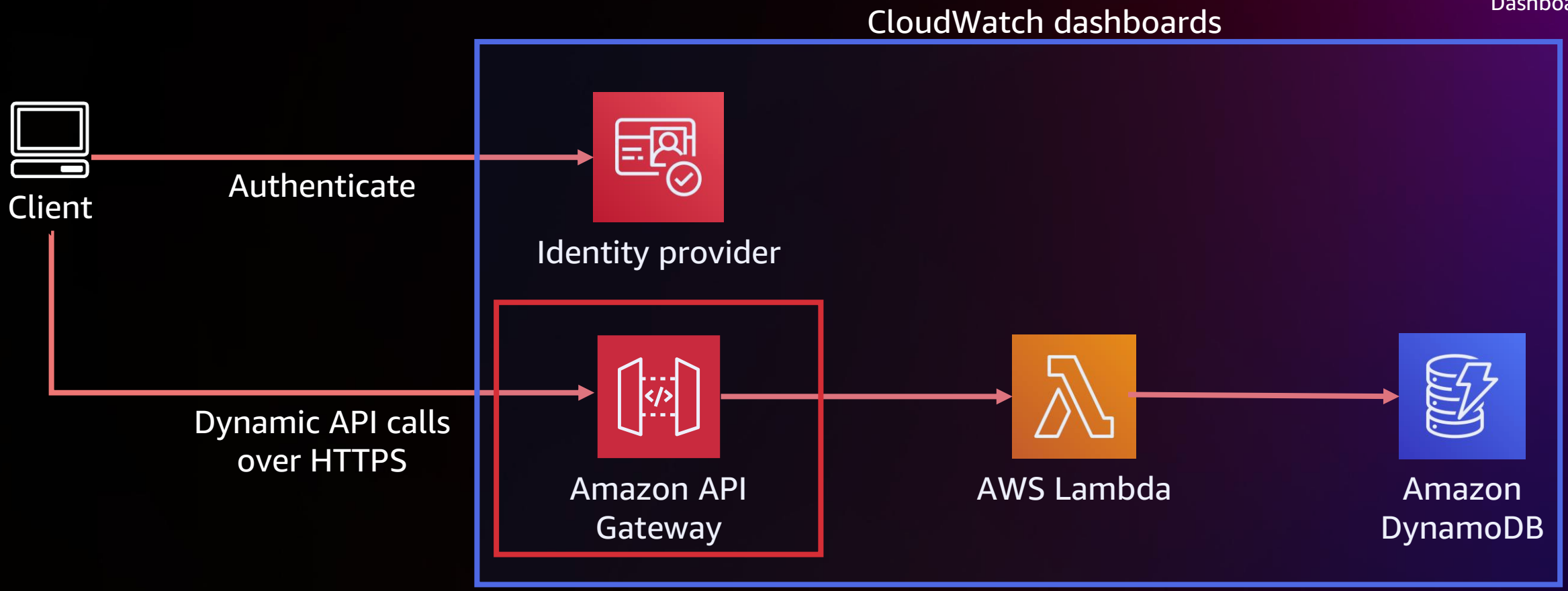
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

CREATE A VIEW OF YOUR SERVICE AND CUSTOMER EXPERIENCE USING DASHBOARDS

Metrics

Dashboard



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

CHECKLIST

1. Identified metrics to monitor
2. Implemented custom metrics
3. Created dashboards to include those metrics
4. We can see the 5XX. Still don’t know why!

What next?

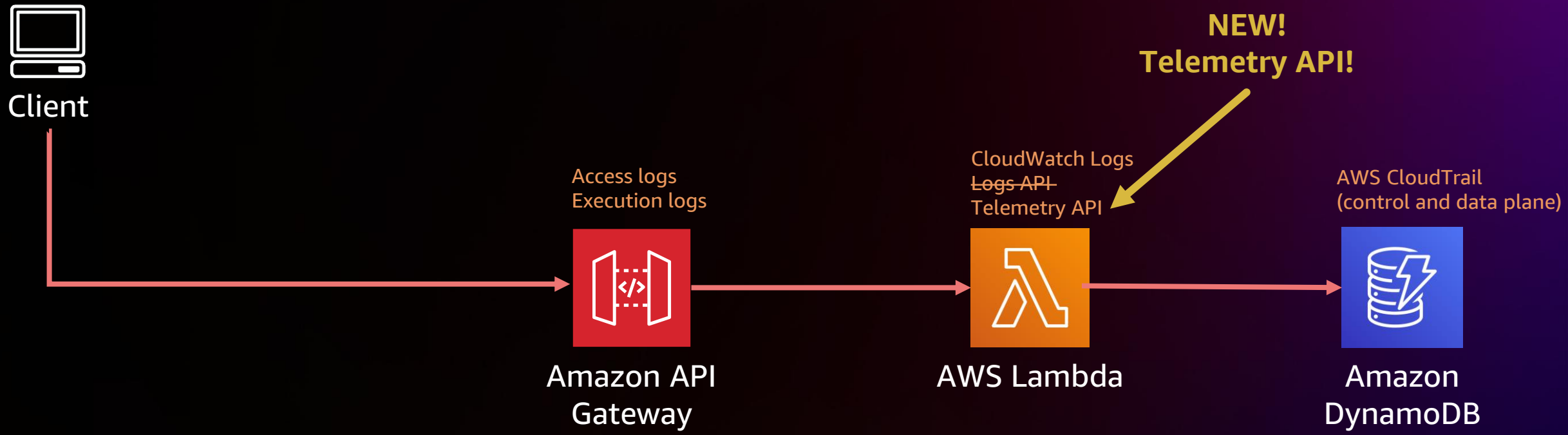
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Logging

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Logging

LOGGING OPTIONS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

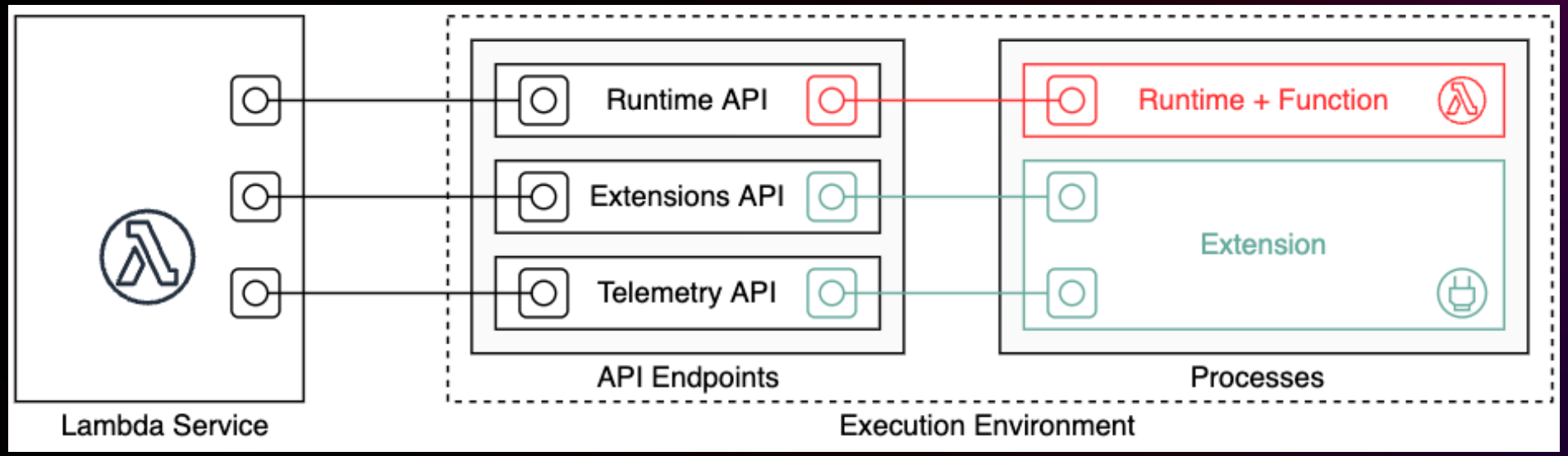
Lambda extensions

Easily integrate Lambda with your favorite monitoring, observability, security, and governance tools

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda extensions

- Easier to integrate your existing tools with Lambda
- No complex installation or configuration
- Tools integrate deeply into the Lambda environment



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using extensions

Deploy extensions as Lambda layers

- Using AWS Management Console, AWS CLI, AWS CloudFormation, AWS SAM, Serverless Framework, Terraform, Stackery

Use extensions

- Consume capabilities of an extension by configuring the options in your Lambda functions

https://s12d.com/extensions-intro

Extensions API

- Build your own extension
- Provides a way for tools to get greater control during function initialization, invocation, and shutdown

https://s12d.com/extensions-building

Telemetry API

- Allow extensions to receive enhanced telemetry data (logs, platform traces, performance metrics)

https://s12d.com/extensions-telemetry

Most customers use extensions without needing to know about the capabilities of the extensions API that enables them

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Extensions available today

AWS LAMBDA READY PARTNER, AWS PARTNER, AND AWS TOOLS USING LAMBDA EXTENSIONS













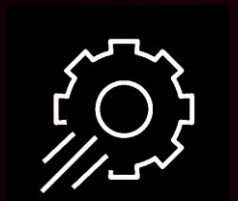








Amazon CloudWatch
Lambda Insights



AWS AppConfig











This is not a complete list. To view all AWS Partners for this category, visit AWS Partner Solutions Finder.
This list of partners is current as of October 21, 2022.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

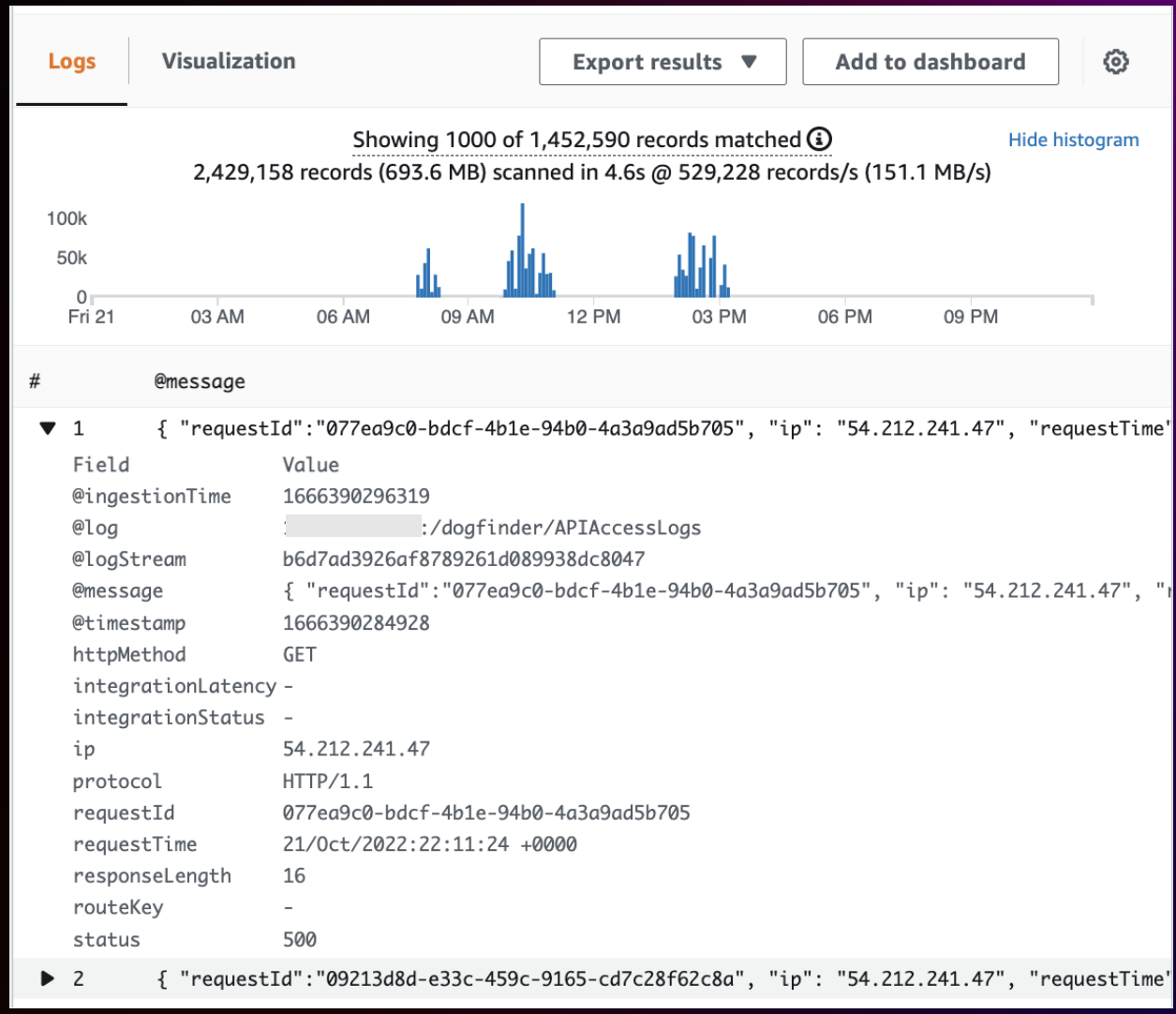
Finding errors in logs

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch Logs Insights

FINDING 5XX ERRORS IN API GATEWAY ACCESS LOGS

```
fields @message, status
| filter status >= 500 and status < 600
```

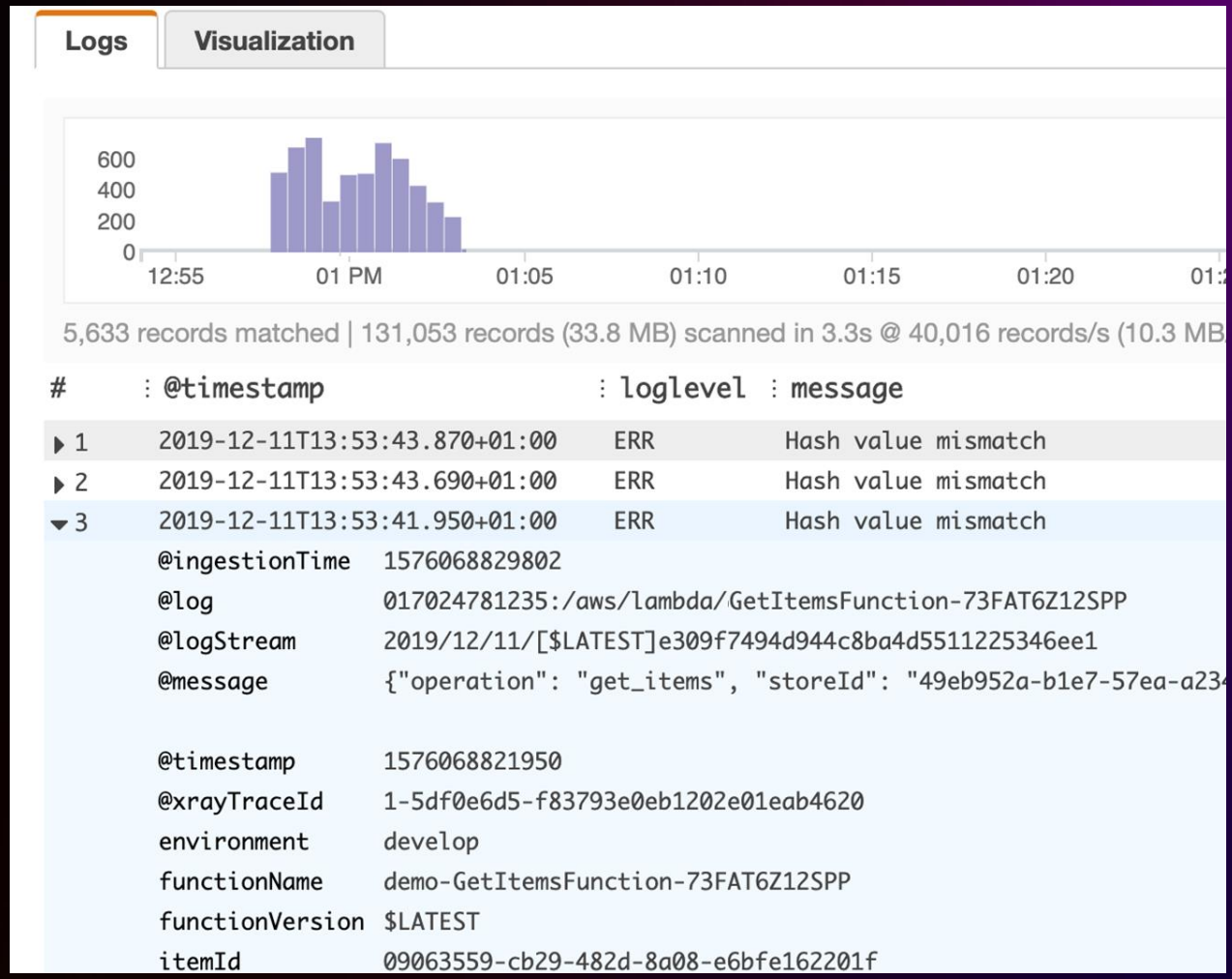


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch Logs Insights

FINDING LAMBDA ERRORS

```
fields Timestamp, LogLevel, Message
| filter LogLevel == "ERR"
| sort @timestamp desc
| limit 100
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

More useful queries!

THE LAMBDA OPERATOR GUIDE HAS A CATALOG OF HELPFUL CLOUDWATCH INSIGHTS QUERIES

Description	Example query syntax

Percentage of cold starts in total invocations	`filter @type = "REPORT"`
	`| stats sum(strcontains(@message, "Init Duration"))/count(*) * 100 as`
	`coldStartPct, avg(@duration)`
	`by bin(5m)`

Percentile report of Lambda duration	`filter @type = "REPORT"`
	`| stats`
	`avg(@billedDuration) as Average,`
	`percentile(@billedDuration, 99) as NinetyNinth,`
	`percentile(@billedDuration, 95) as NinetyFifth,`
	`percentile(@billedDuration, 90) as Ninetieth`
	`by bin(30m)`

Percentile report of Lambda memory usage	`filter @type="REPORT"`
	`| stats avg(@maxMemoryUsed/1024/1024) as mean_MemoryUsed,`
	`min(@maxMemoryUsed/1024/1024) as min_MemoryUsed,`
	`max(@maxMemoryUsed/1024/1024) as max_MemoryUsed,`
	`percentile(@maxMemoryUsed/1024/1024, 95) as Percentile95`

Invocations using 100% of assigned memory	`filter @type = "REPORT" and @maxMemoryUsed=@memorySize`
	`| stats`
	`count_distinct(@requestId)`
	`by bin(30m)`

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://s12d.com/queries

Logging

RECAP

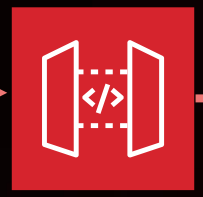

Metrics


Dashboard


Logs


Client

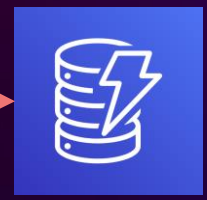
Access logs
Execution logs


Amazon API Gateway

CloudWatch Logs
~~Logs API~~
Telemetry API


AWS Lambda

CloudTrail
(control and data plane)


Amazon DynamoDB

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

CHECKLIST

1. Identified metrics to monitor
2. Implemented custom metrics
3. Created dashboards to include those metrics
4. We queried logs

What next?

*Note that execution logs would have told us the answer, typically not enabled in production

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tracing

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

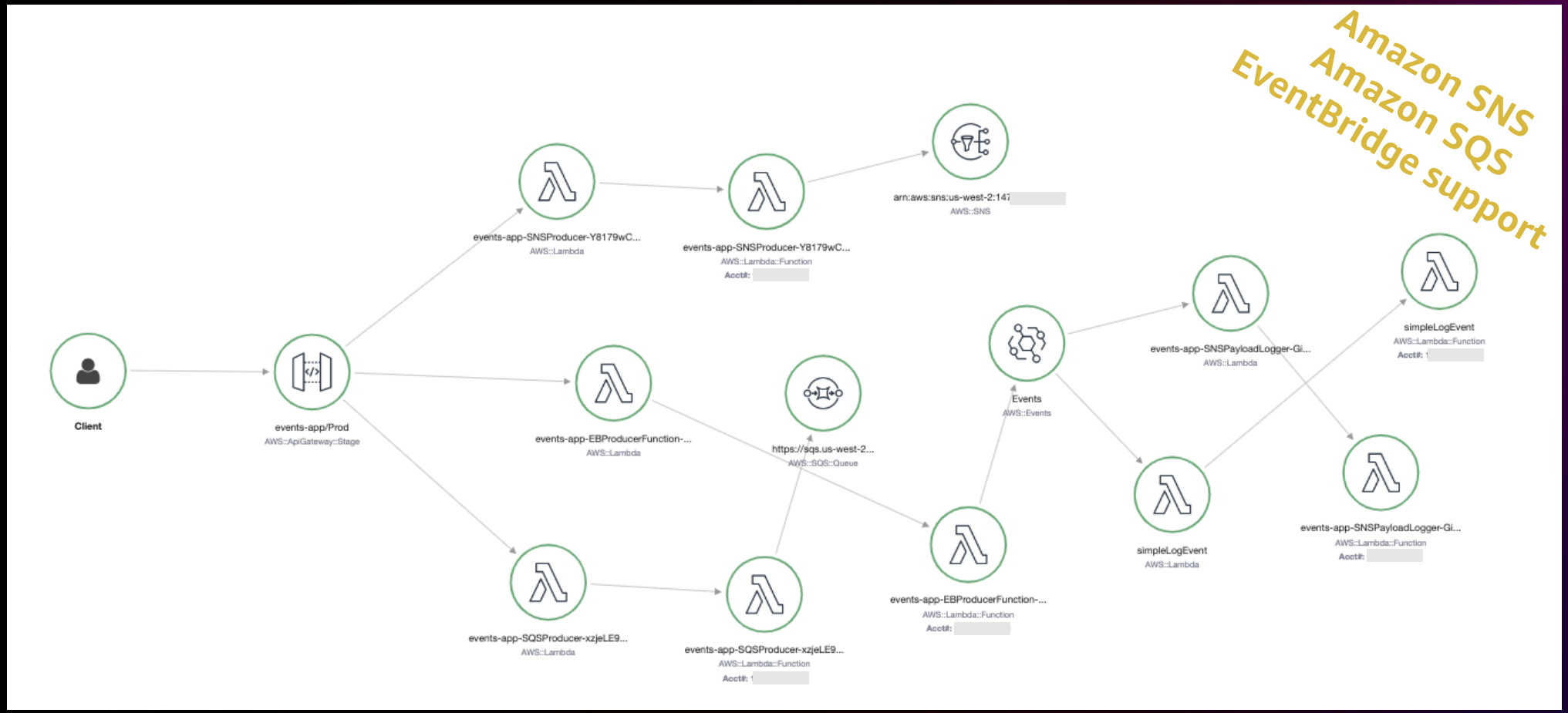
END-TO-END VIEW OF REQUESTS FLOWING THROUGH AN APPLICATION



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

END-TO-END VIEW OF REQUESTS FLOWING THROUGH AN APPLICATION



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

ENABLE TRACING ON LAMBDA AND API GATEWAY RESOURCES

AWS SAM

```
Globals:
  # Enable X-Ray tracing on API Gateway
  Api:
    TracingEnabled: true
  # Set default values for all Lambda functions
  Function:
    Tracing: Active
    Timeout: 10
```

AWS CDK

```
const lambda_backend = new lambda.Function(this, "lambdaFunction", {
  runtime: lambda.Runtime.NODEJS_14_X,
  handler: "index.handler",
  code: lambda.Code.fromAsset("src"),
  tracing: lambda.Tracing.ACTIVE
})

const api = new apigateway.RestApi(this, "RestAPI", {
  deployOptions: {
    metricsEnabled: true,
    loggingLevel: apigateway.MethodLoggingLevel.INFO,
    dataTraceEnabled: true,
    tracingEnabled: true,
    stageName: 'prod'
  },
})
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

INSTRUMENT YOUR CODE

```python
import json
import os
import boto3
from datetime import datetime
from aws_embedded_metrics import metric_scope

# Patch libraries to instrument downstream calls
# See https://docs.aws.amazon.com/xray/latest/devguide/xray-sdk-python-patching.html for more details
from aws_xray_sdk.core import patch_all

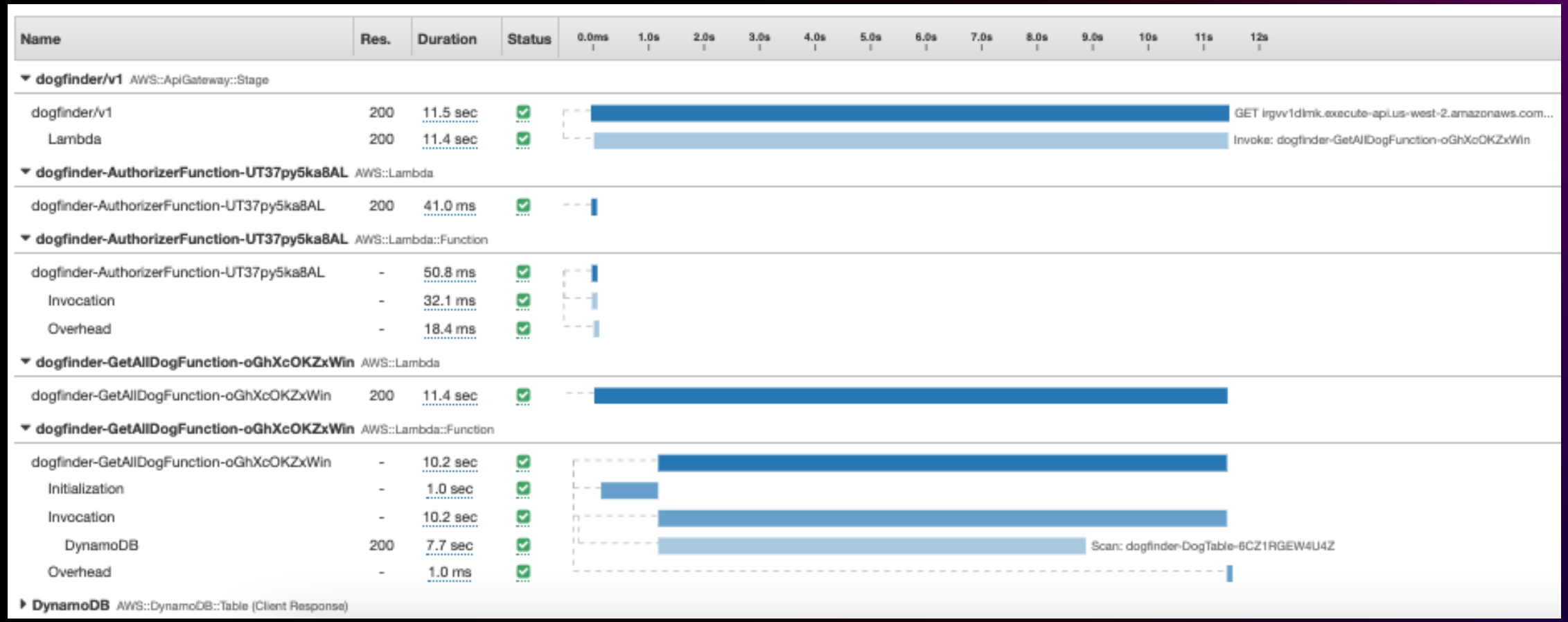
patch_all()

...
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

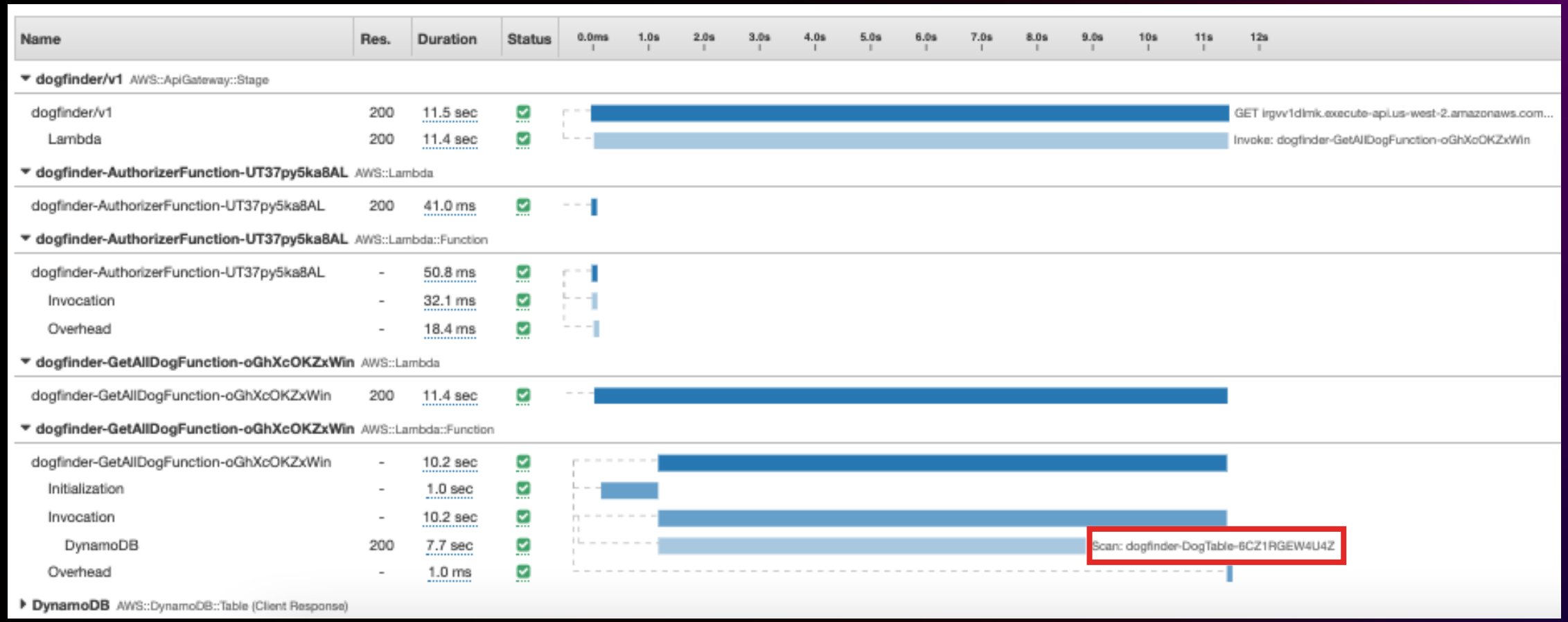
https://s12d.com/xray-sdk

AWS X-Ray trace example



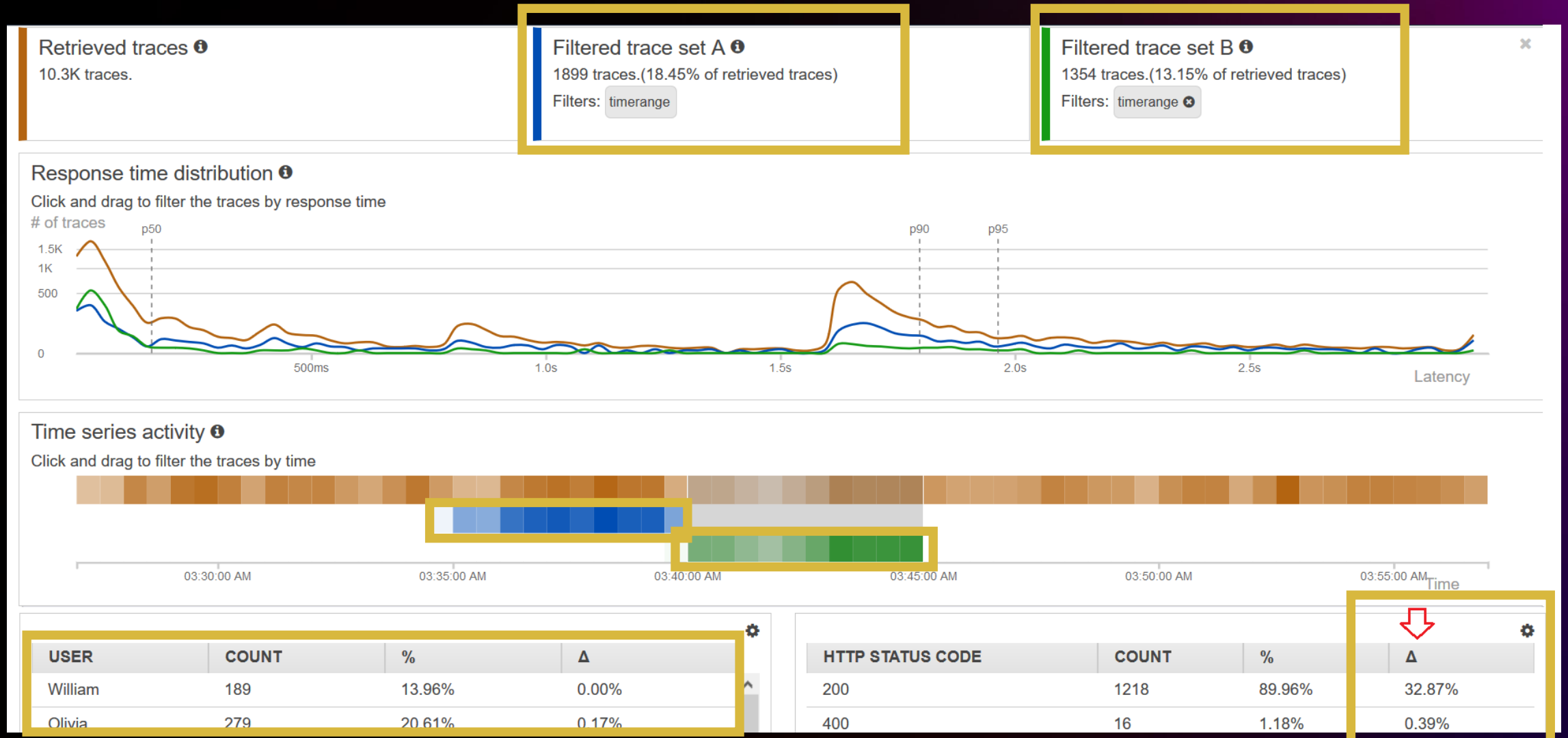
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray trace example



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

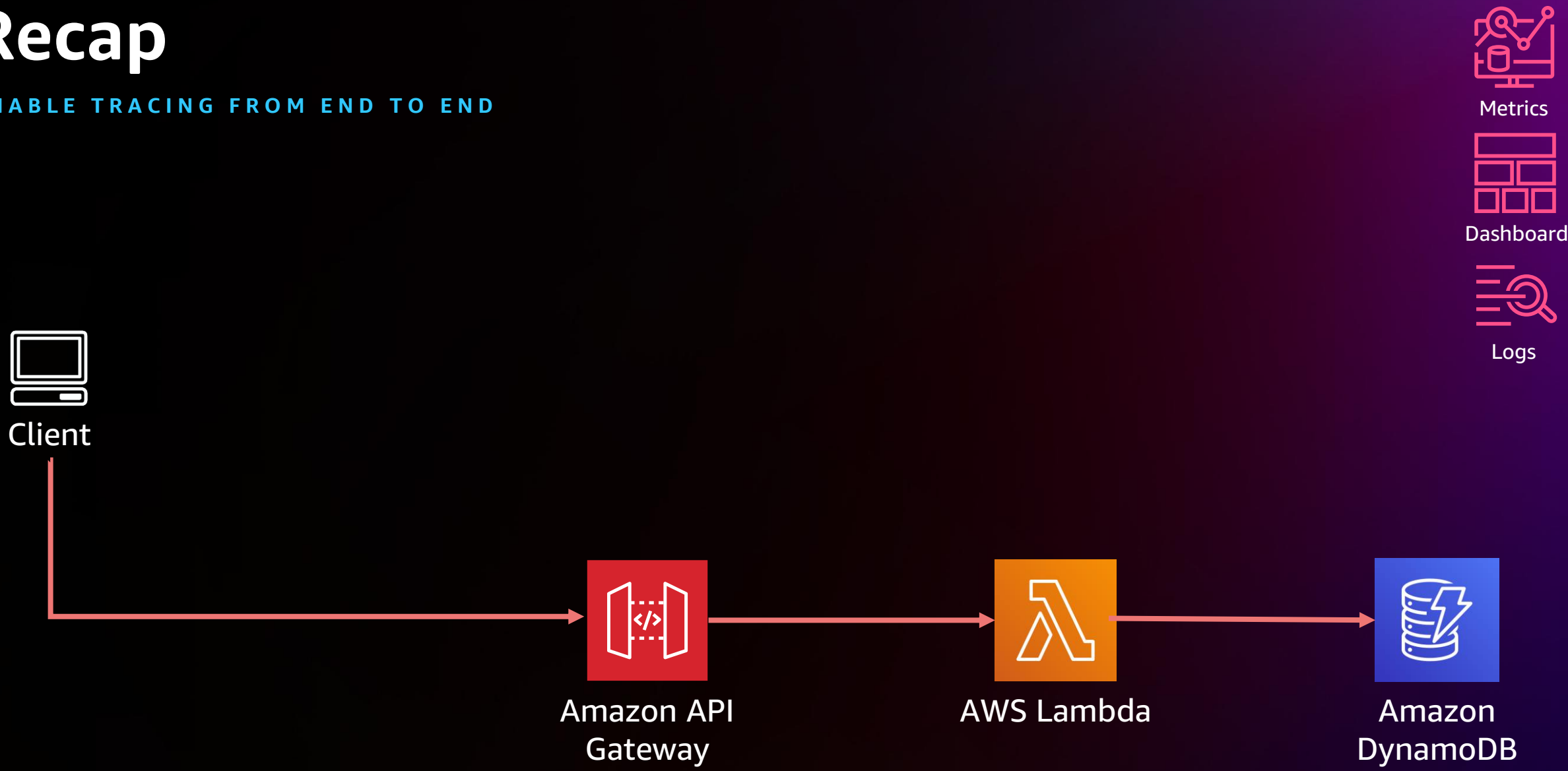
AWS X-Ray Analytics example



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

ENABLE TRACING FROM END TO END



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

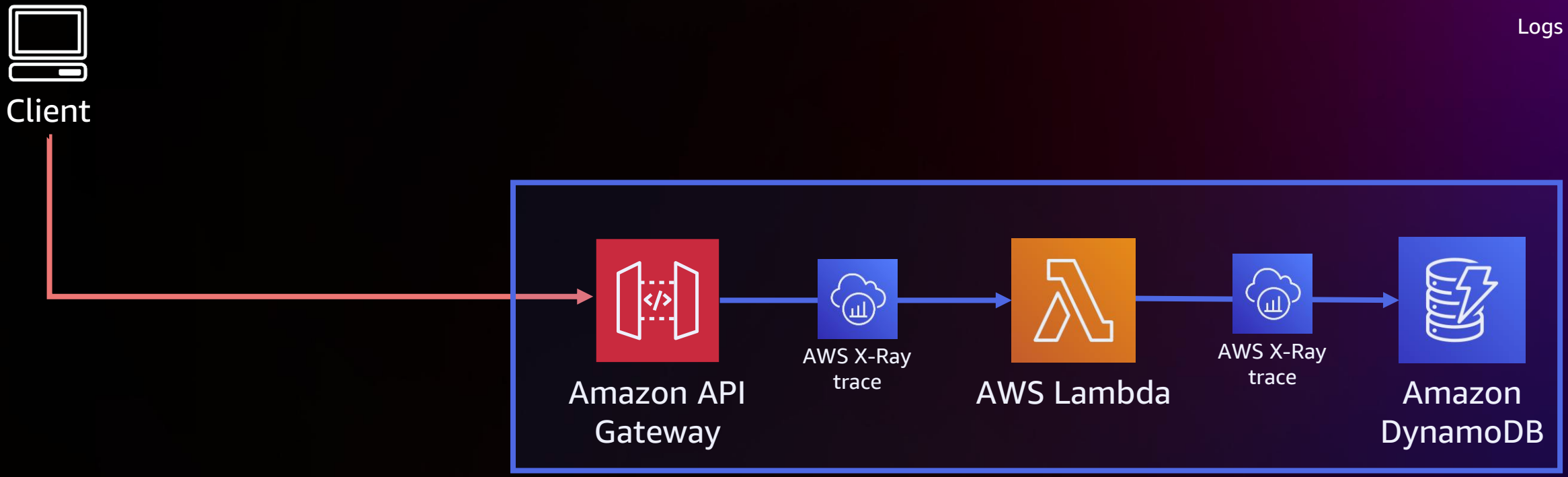
Recap

ENABLE TRACING FROM END TO END

Metrics

Dashboard

Logs



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

CHECKLIST

1. Identified metrics to monitor
2. Implemented custom metrics
3. Created dashboards to include those metrics
4. Queried logs
5. Enabled tracing

Wait . . . we should have seen it in AWS X-Ray!?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A different view: CloudWatch ServiceLens

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch ServiceLens

- Ties together CloudWatch metrics and logs, in addition to traces from AWS X-Ray
- Gives y

CloudWatch ServiceLens

Amazon CloudWatch

Metrics
Logs
Events
Alarms
Dashboards



AWS X-Ray

Traces
Analytics
Service map
Latency detection (server and client)



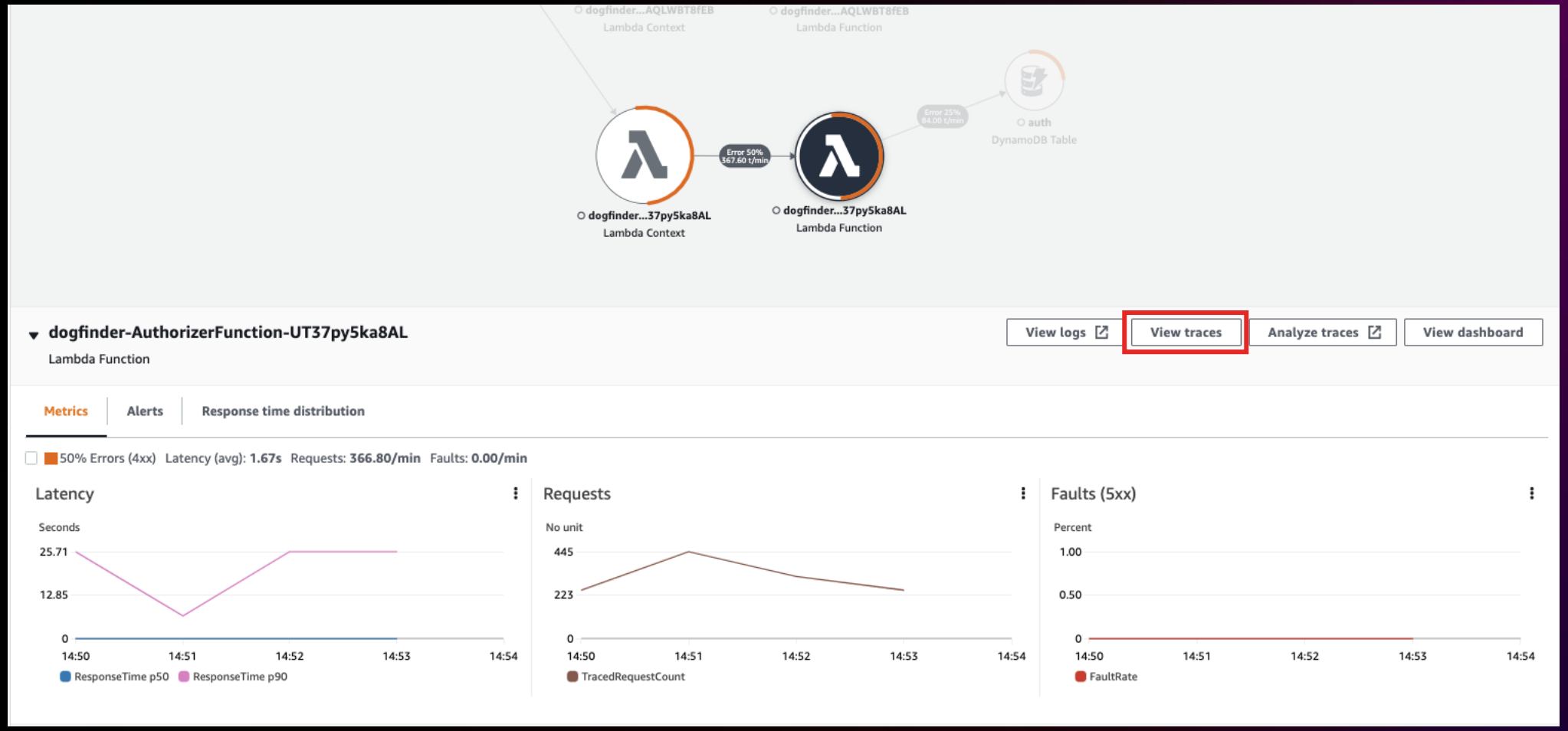
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudWatch ServiceLens



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

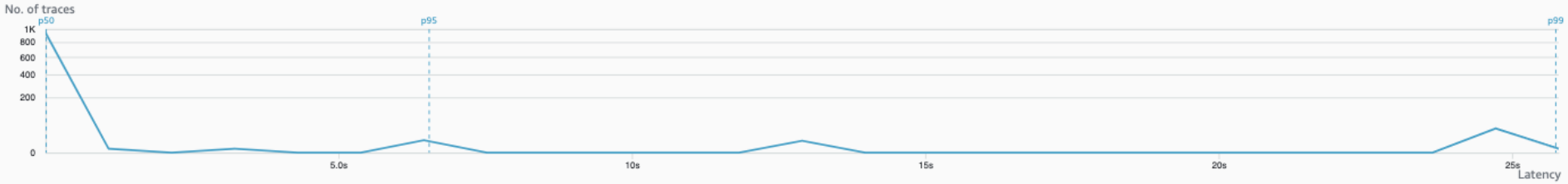
Drill down . . . Click on Lambda function



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Refine query by response time distribution

Drag and drop on the graph to select a time frame



Sort by response code

Traces (1000)

This table shows the most recent traces with an average response time of 1.22s. It shows as many as 1000 traces.

Add to dashboard

Start typing to filter trace list

1 2 3 4 5 6 7 ... 67

ID	Trace status	Timestamp	Response code	Response Time	Duration	HTTP Method	URL Address
...68a8a9fa4ff24f3a3025d77d	Fault (5xx)	4.7min (2022-10-21 14:53:30)	500	25.707s	25.724s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...291cd58c483133e3150457a3	Fault (5xx)	4.0min (2022-10-21 14:54:08)	500	25.707s	25.707s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...2573c2535c085029292a9aa8	Fault (5xx)	4.1min (2022-10-21 14:54:06)	500	25.794s	25.81s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...220ef8e536f8cfd20d8a9813	Fault (5xx)	4.7min (2022-10-21 14:53:29)	500	25.696s	25.709s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...365d39b97ead283e05708238	Fault (5xx)	4.2min (2022-10-21 14:54:00)	500	0.074s	208.421s	DELETE	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...46a5426778e9d0b5150cc1eb	Fault (5xx)	4.1min (2022-10-21 14:54:04)	500	25.744s	25.744s	GET	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...60b408b91c52736517173f76	Fault (5xx)	4.1min (2022-10-21 14:54:06)	500	25.707s	25.723s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...79a78cab6132932a6bcc2ffe	Fault (5xx)	4.7min (2022-10-21 14:53:25)	500	25.746s	25.762s	GET	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...193752660ec3db631b07ce79	Fault (5xx)	4.7min (2022-10-21 14:53:29)	500	25.748s	25.748s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...508bdc65010c2c262bb42723	Fault (5xx)	4.3min (2022-10-21 14:53:54)	500	6.496s	6.496s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...0a0e49ba620b05a56e1cd3d7	Fault (5xx)	4.7min (2022-10-21 14:53:30)	500	25.681s	25.697s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...3fee545c16abe9e86bcd0040	Fault (5xx)	4.0min (2022-10-21 14:54:07)	500	25.745s	25.762s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...7f26aaf30b8431af204e8c35	Fault (5xx)	4.1min (2022-10-21 14:54:06)	500	25.73s	244.989s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...0c68bbd66bb8aa8359e63b12	Fault (5xx)	4.1min (2022-10-21 14:54:01)	500	25.706s	25.722s	DELETE	https://irgvv1dlmk.execute-api.us-west-2.amazona...
...1970039355a2a44a66566f2d	Fault (5xx)	4.0min (2022-10-21 14:54:08)	500	25.735s	25.751s	PUT	https://irgvv1dlmk.execute-api.us-west-2.amazona...

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Segments Timeline Info

	Segment status	Response code	Duration	0.0ms 2.0s 4.0s 6.0s 8.0s 10s 12s 14s 16s 18s 20s 22s 24s 26s
▼ dogfinder/v1 AWS::ApiGateway::Stage				
dogfinder/v1	Fault (5xx)	500	25.71s	PUT https://irgvv1dlmk.execute-api.us-west-
▼ dogfinder-AuthorizerFunction-UT37py5ka8AL AWS::Lambda				
dogfinder-AuthorizerFunction-UT37py5ka8AL	Error (4xx)	200	25.70s	
▼ dogfinder-AuthorizerFunction-UT37py5ka8AL AWS::Lambda::Function				
dogfinder-AuthorizerFunction-UT37py5ka8AL	Error (4xx)	-	25.71s	
Invocation	Error (4xx)	-	25.69s	
DynamoDB	Fault (5xx)	400	25.68s	GetItem: auth
Overhead	OK	-	19ms	
▼ DynamoDB AWS::DynamoDB::Table				
DynamoDB	Error (4xx)	400	25.68s	GetItem: auth

Logs Info

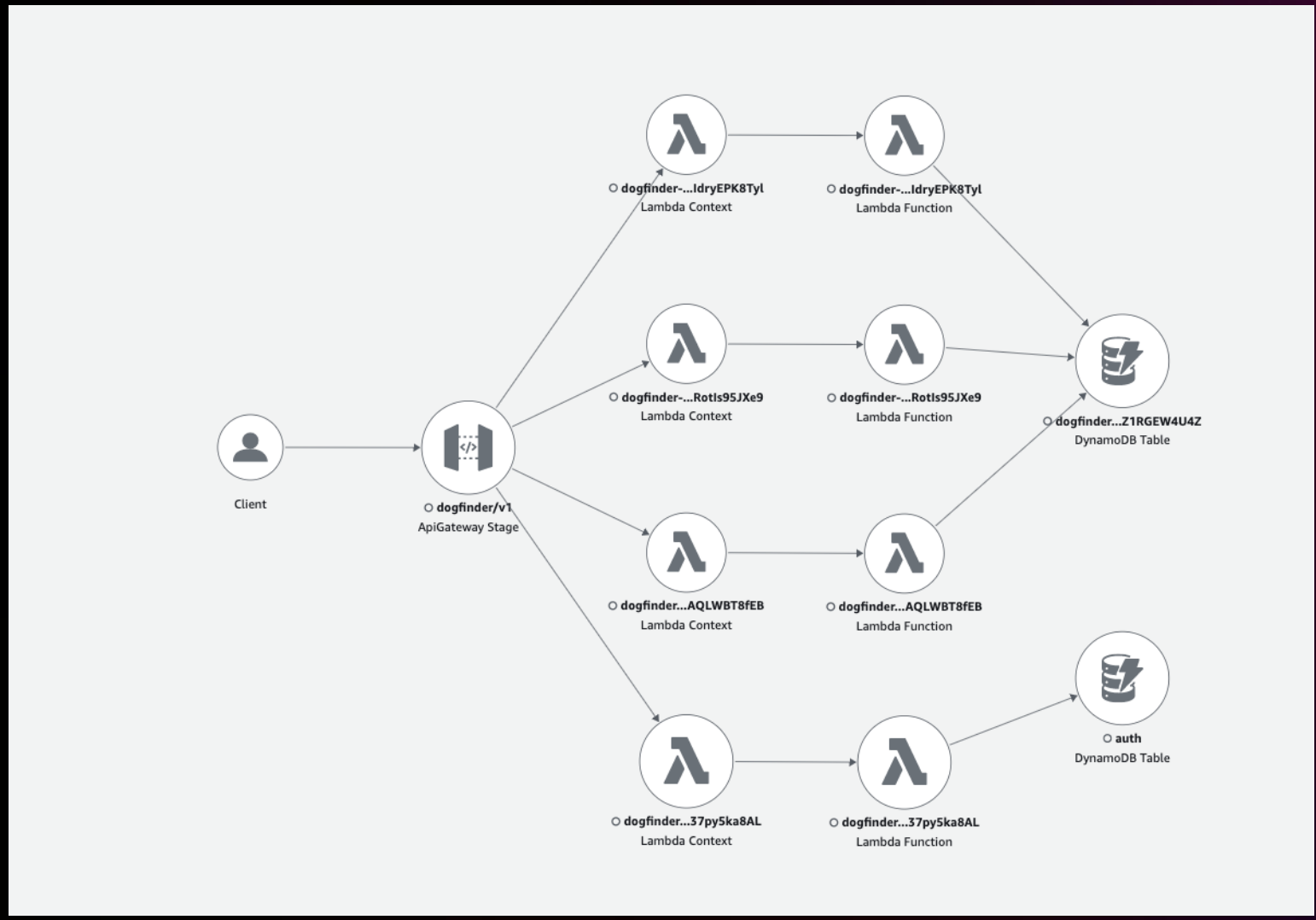
All logs for this trace

View in CloudWatch Logs Insights

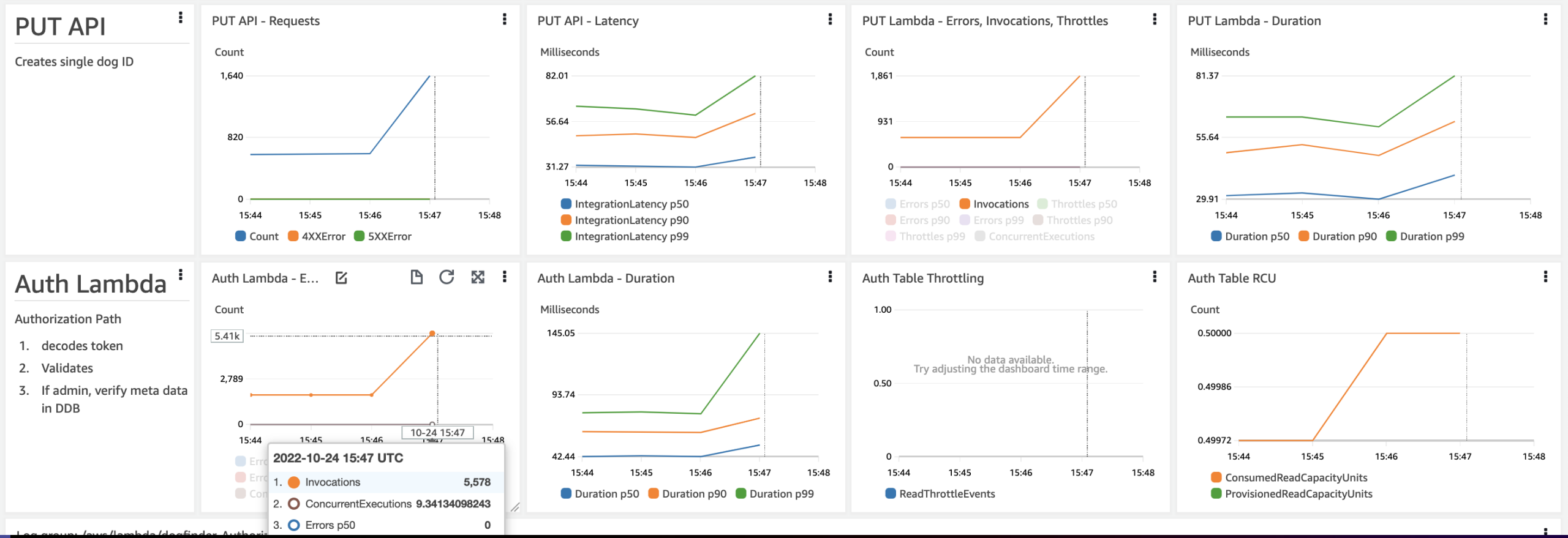
#	@log	@timestamp	@message
1	147791281060:API-Gateway-Execution-Logs_irgvv1dlmk/v1	2022-10-21T21:53:30.256Z	(631aae54-1e3e-4269-99a6-c61dd6462d10) Extended Request Id: aYAyIHbJvHcFx-w=
2	147791281060:API-Gateway-Execution-Logs_irgvv1dlmk/v1	2022-10-21T21:53:30.257Z	(631aae54-1e3e-4269-99a6-c61dd6462d10) Starting authorizer: slhyhv for request: 631aae54-1e3e-
3	147791281060:API-Gateway-Execution-Logs_irgvv1dlmk/v1	2022-10-21T21:53:30.257Z	(631aae54-1e3e-4269-99a6-c61dd6462d10) Sending request to https://lambda.us-west-2.amazonaws.c
4	147791281060:API-Gateway-Execution-Logs_irgvv1dlmk/v1	2022-10-21T21:53:30.257Z	(631aae54-1e3e-4269-99a6-c61dd6462d10) Incoming identity: **********
5	147791281060:API-Gateway-Execution-Logs_irgvv1dlmk/v1	2022-10-21T21:53:30.257Z	(631aae54-1e3e-4269-99a6-c61dd6462d10) Endpoint request URI: https://lambda.us-west-2.amazonaw
6	147791281060:API-Gateway-Execution-Logs_irgvv1dlmk/v1	2022-10-21T21:53:30.257Z	(631aae54-1e3e-4269-99a6-c61dd6462d10) Endpoint request headers: {X-Amz-Date=20221021T215330Z,
7	147791281060:API-Gateway-Execution-Logs_irgvv1dlmk/v1	2022-10-21T21:53:30.257Z	(631aae54-1e3e-4269-99a6-c61dd6462d10) Endpoint request body after transformations: {"type":"T
8	147791281060:/aws/lambda/dogfinder-AuthorizerFunction-UT37py5ka8AL	2022-10-21T21:53:30.269Z	START RequestId: fce01035-ff31-4ab3-8d67-fb18e99ce812 Version: $LATEST

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

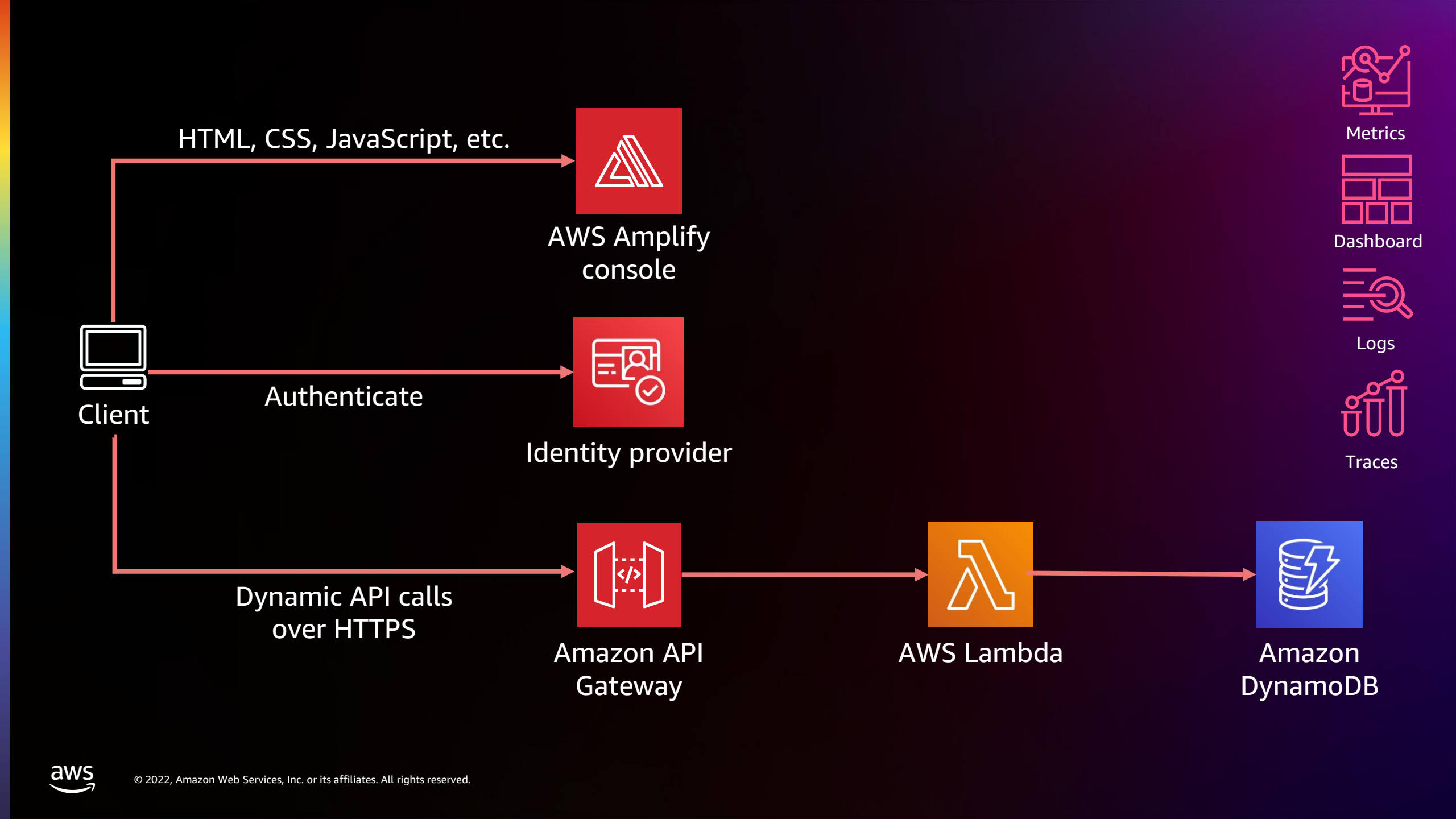
After scaling DynamoDB table



Adjusted dashboard

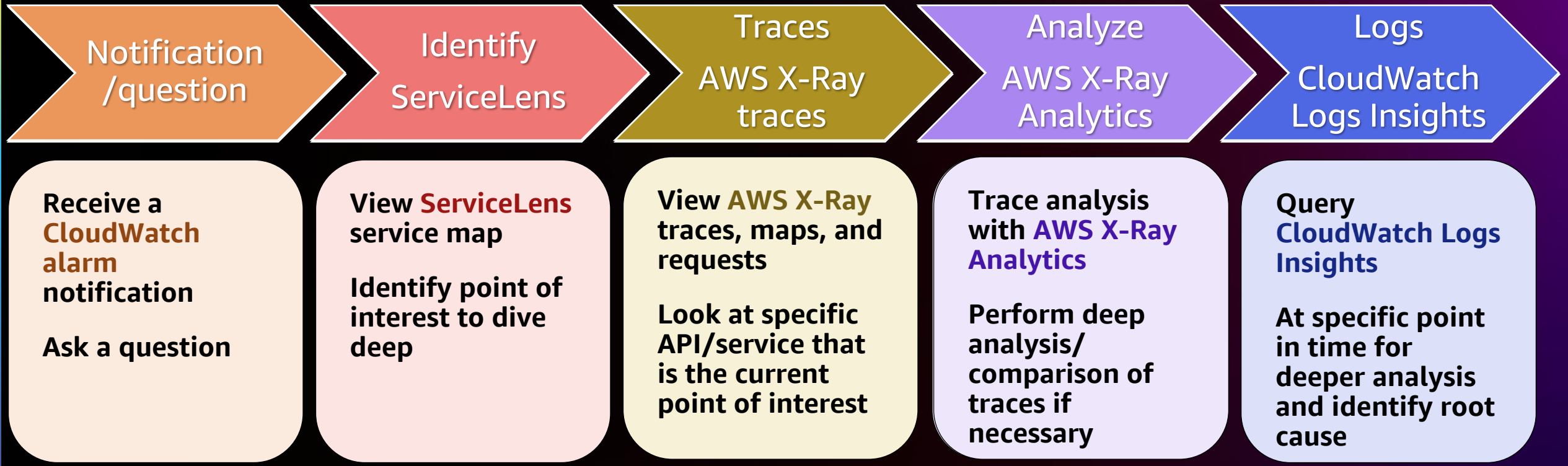


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Troubleshooting/query workflow



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scenario recap

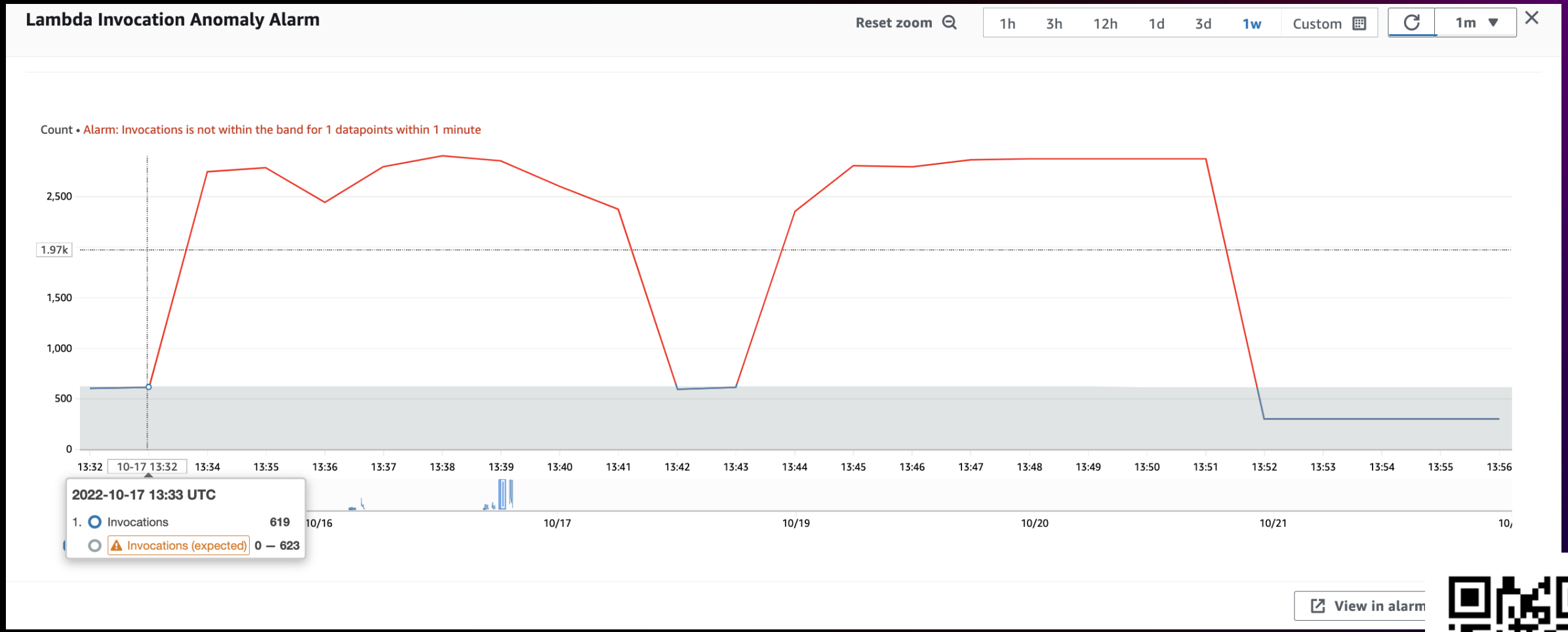
- Know the service metrics – create alarms on the critical metrics
  - Include these alarms in your infrastructure as code when you create the application
  - Set alarm thresholds appropriately
- Use Embedded Metric Format to create application-level metrics
- Create dashboards
  - Include each route of your API and/or logical components
  - Proper statistics: P50, P90, P99 vs. average, min/max
- Enable tracing and instrument your code using the AWS X-Ray SDK
- Use CloudWatch ServiceLens for end-to-end visibility and troubleshooting
- Identify ad hoc CloudWatch Logs Insights queries
  - Include in your run books and dashboards

aws

What else could have helped us?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using CloudWatch anomaly detection alarms



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

https://s12d.com/anomaly

Lambda Powertools library

A SUITE OF UTILITIES FOR AWS LAMBDA FUNCTIONS

Logging: Output as structured JSON

Tracing: Send traces to AWS X-Ray

Metrics: Custom metrics with embedded metric format

Utilities: Parameters, Amazon SQS batch processing, etc.

Python

https://github.com/awslabs/aws-lambda-powertools-python

Java

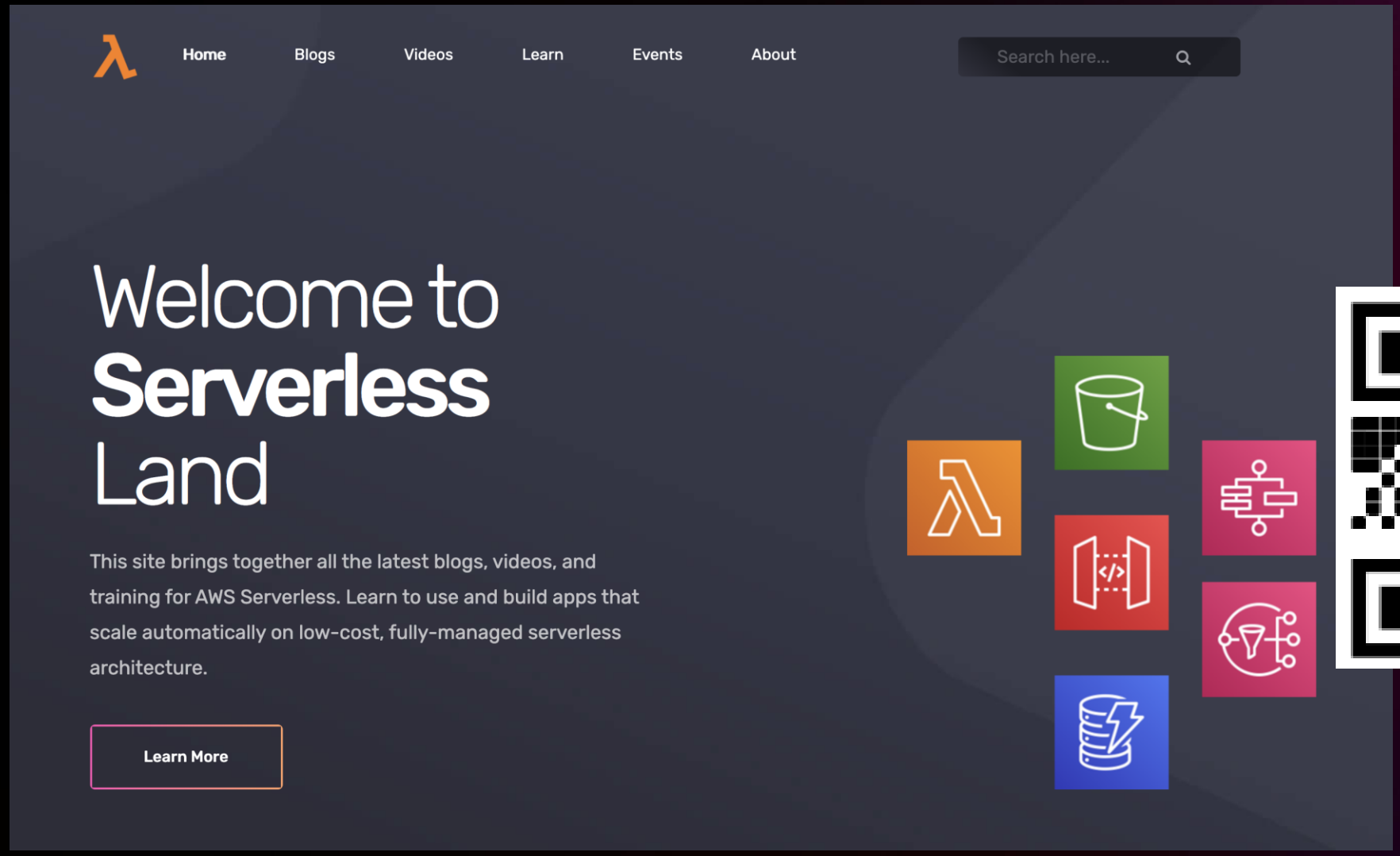
https://github.com/awslabs/aws-lambda-powertools-java

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You should also know . . .

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverlessland.com



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Other helpful resources

- Well-Architected Serverless Lens
  - https://s12d.com/serverless-lens



- Serverless observability workshop
  - https://serverless-observability.workshop.aws/



- Full-featured serverless sample application
  - https://s12d.com/serverless-samples



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.