
AWS
re:Invent

SVS308

Becoming proficient with serverless application observability

Julian Wood

Senior Developer Advocate, AWS Serverless

@julian_wood

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What I am covering today

What is observability?

Standard and custom metrics

Structured logging

Alerts

Tracing

Troubleshooting and discovery

Lambda extensions and more tools



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless applications

Event source

Function

Services (anything)







Changes in data state



Changes in resource state



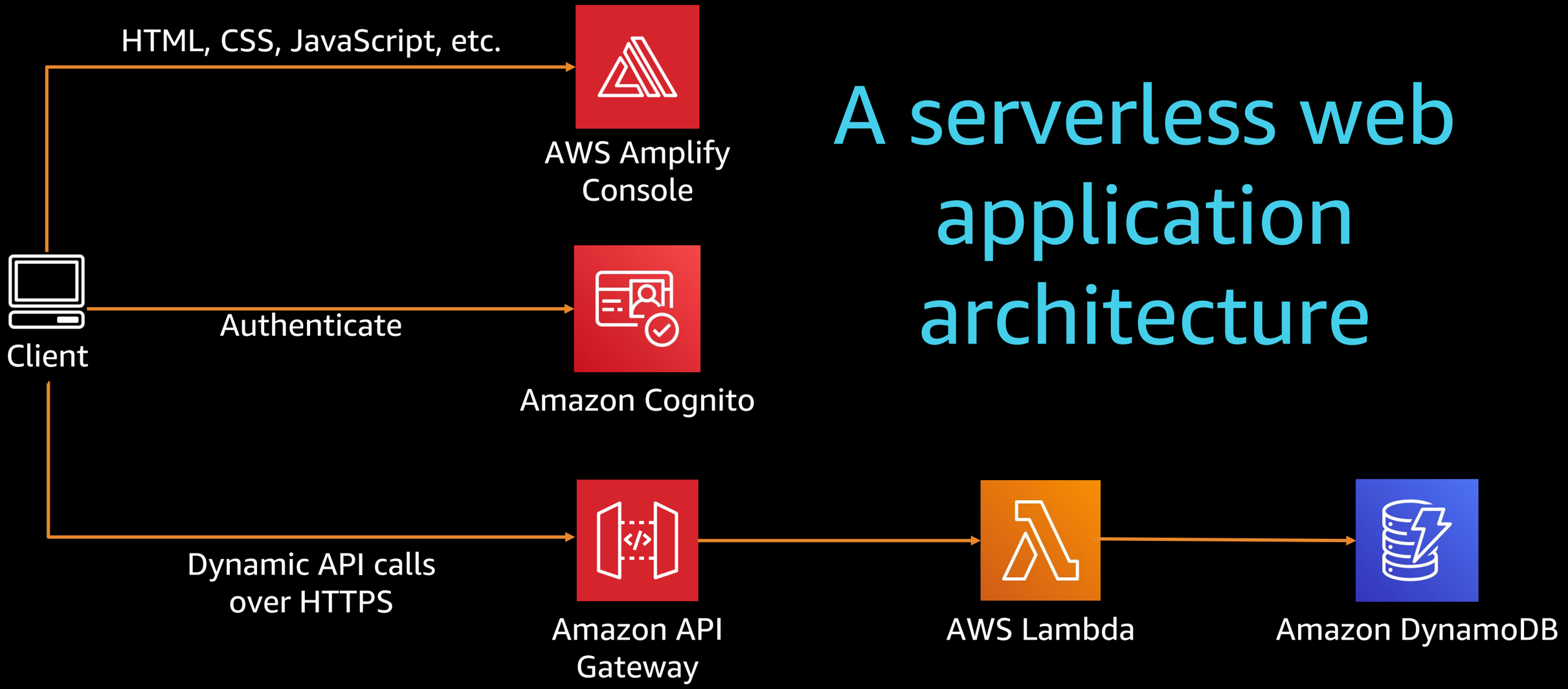
Requests to endpoints



Node.js
Python
Java
Go
Ruby
.Net (C# / PowerShell)
Custom (Runtime API)



A serverless web application architecture



“Everything fails, all the time.”

Werner Vogels

VP & CTO

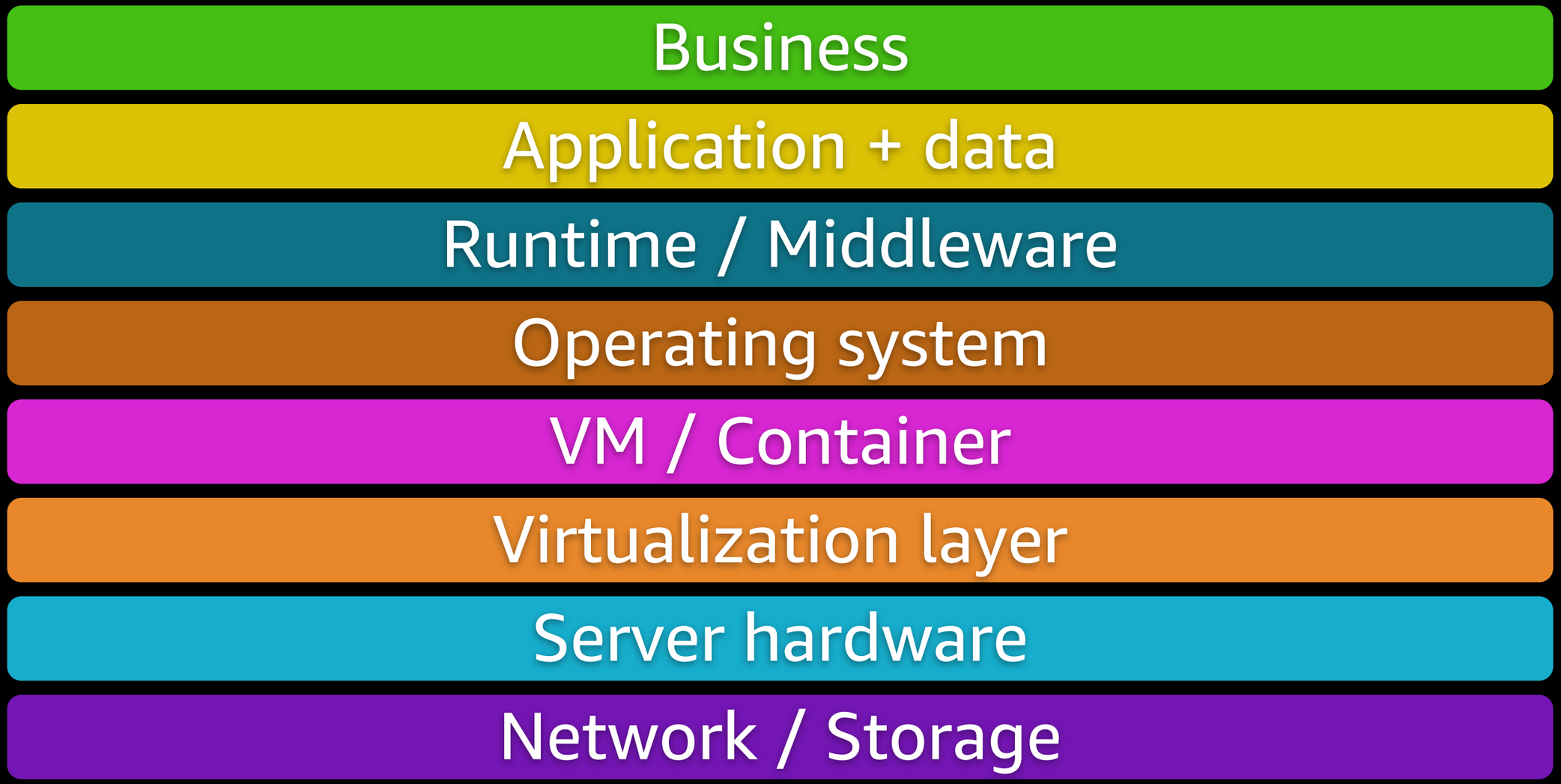
Amazon.com



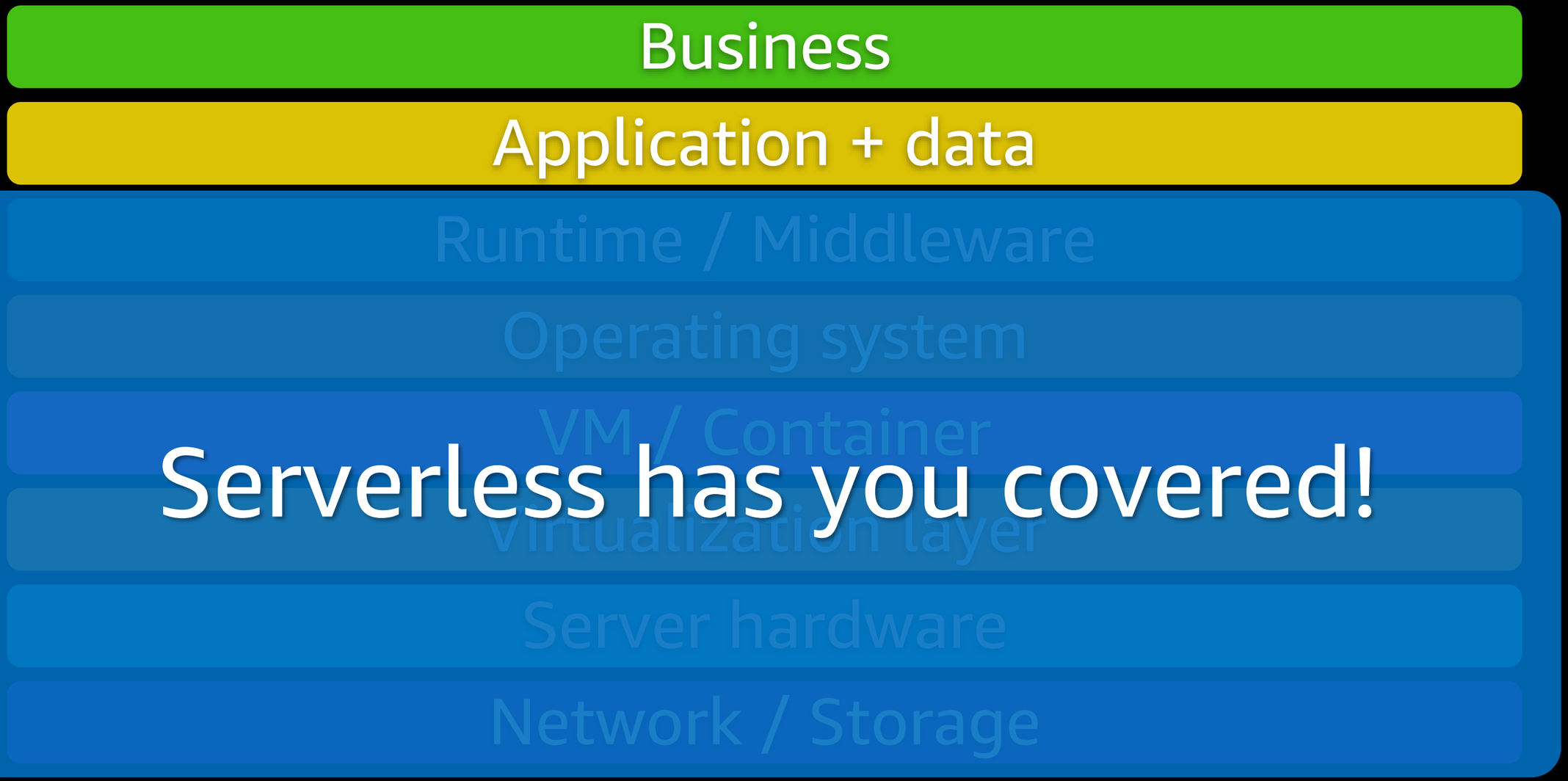
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Traditional application stack



Serverless application stack



What is observability?



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Monitoring more than failures

Is it behaving as expected?

What is the usage?

What is the business impact?

Monitoring more than failures

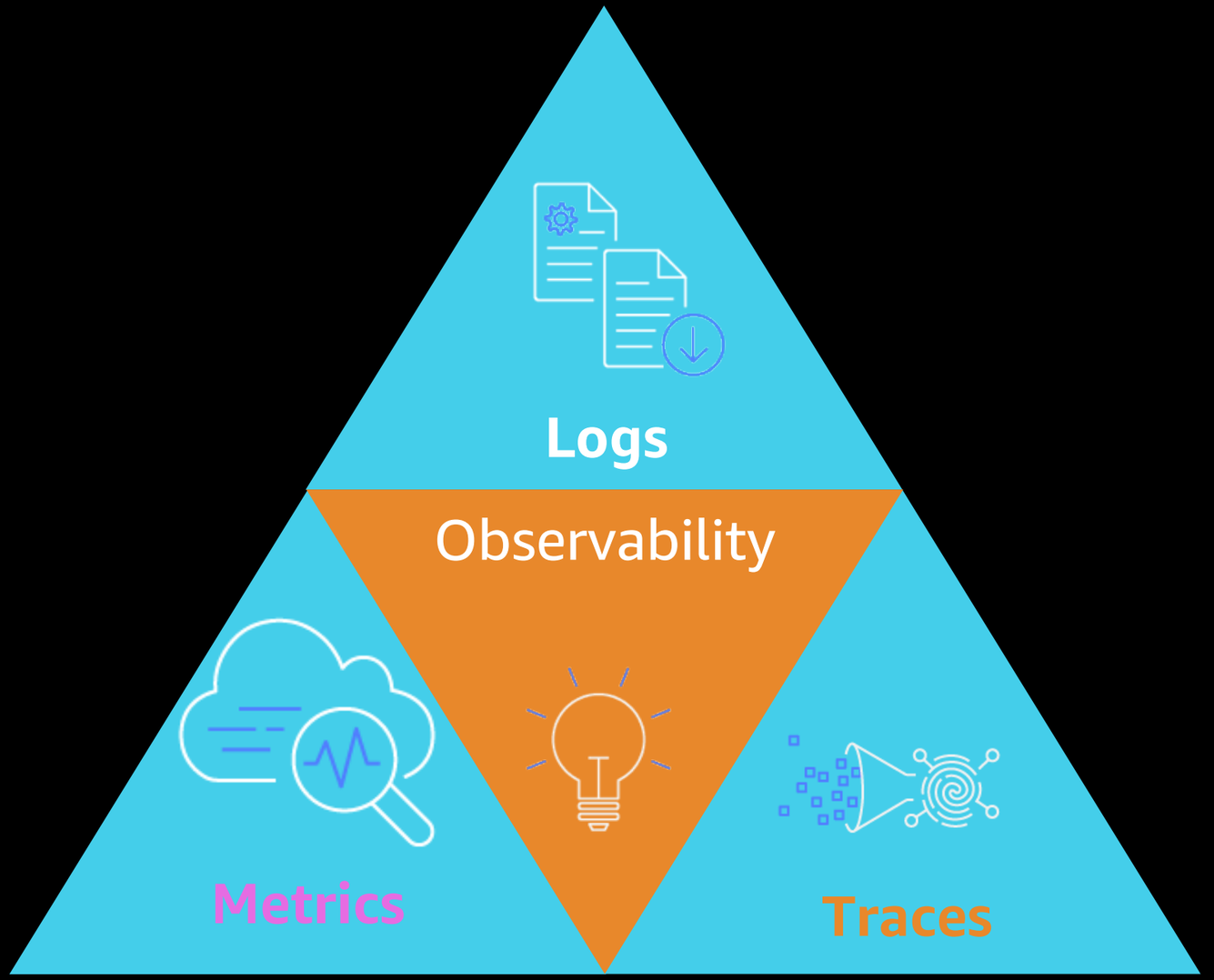
Observability is more about your organizational culture and practices than it is about your tooling

Three pillars of observability tooling

Metrics	Logs	Traces
Numeric data measured at various time intervals (time series data); SLIs (request rate, error rate, duration, CPU%, etc.)	Timestamped records of discrete events that happened within an application or system, such as a failure, an error, or a state transformation	A trace represents a single user's journey across multiple applications and systems (usually microservices)

Definitions from "Distributed Systems Observability," by Cindy Sridharan. Available at: https://www.oreilly.com/library/view/distributed-systems-observability/9781492033431/

Observability is the goal



AWS services for observability



Amazon
CloudWatch

Dashboards
Logs
Metrics
Alarms
Events



AWS X-Ray

Traces
Analytics
Service map

Amazon CloudWatch

1 quadrillion+
(1,000,000,000,000,000+)
Metric observations each month

3.9 trillion
Events each month

Monitors entire infrastructure of AWS and Amazon.com

100 PB
Logs ingested each month

AWS Lambda Ready Partners

DEVELOPER TOOLING SOLUTIONS VALIDATED TO INTEGRATE WITH AWS LAMBDA

















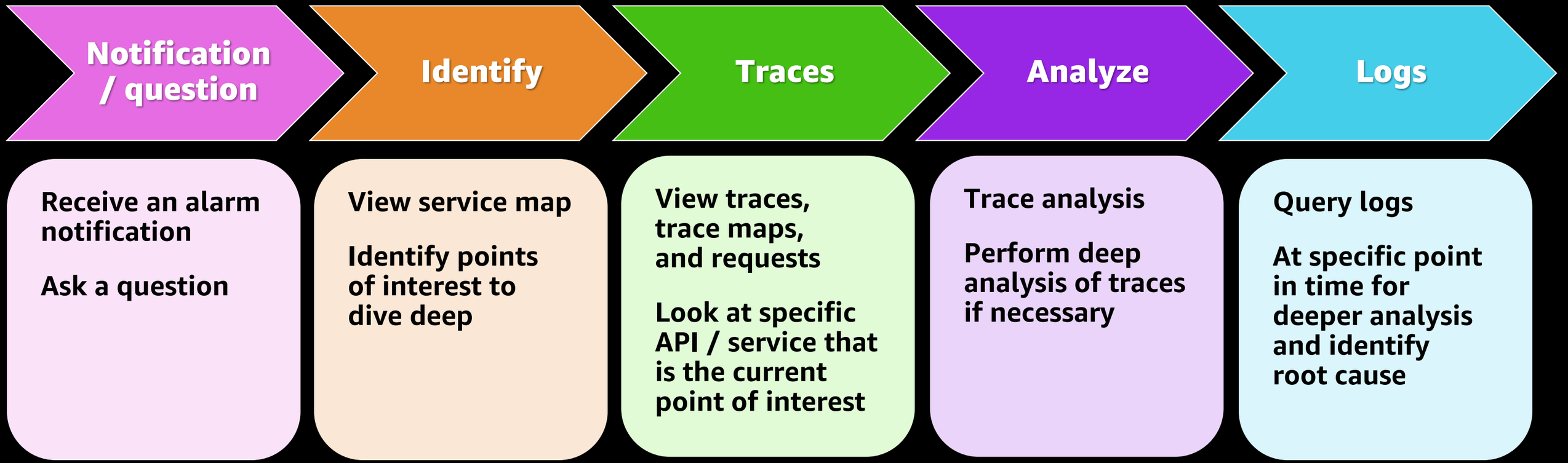


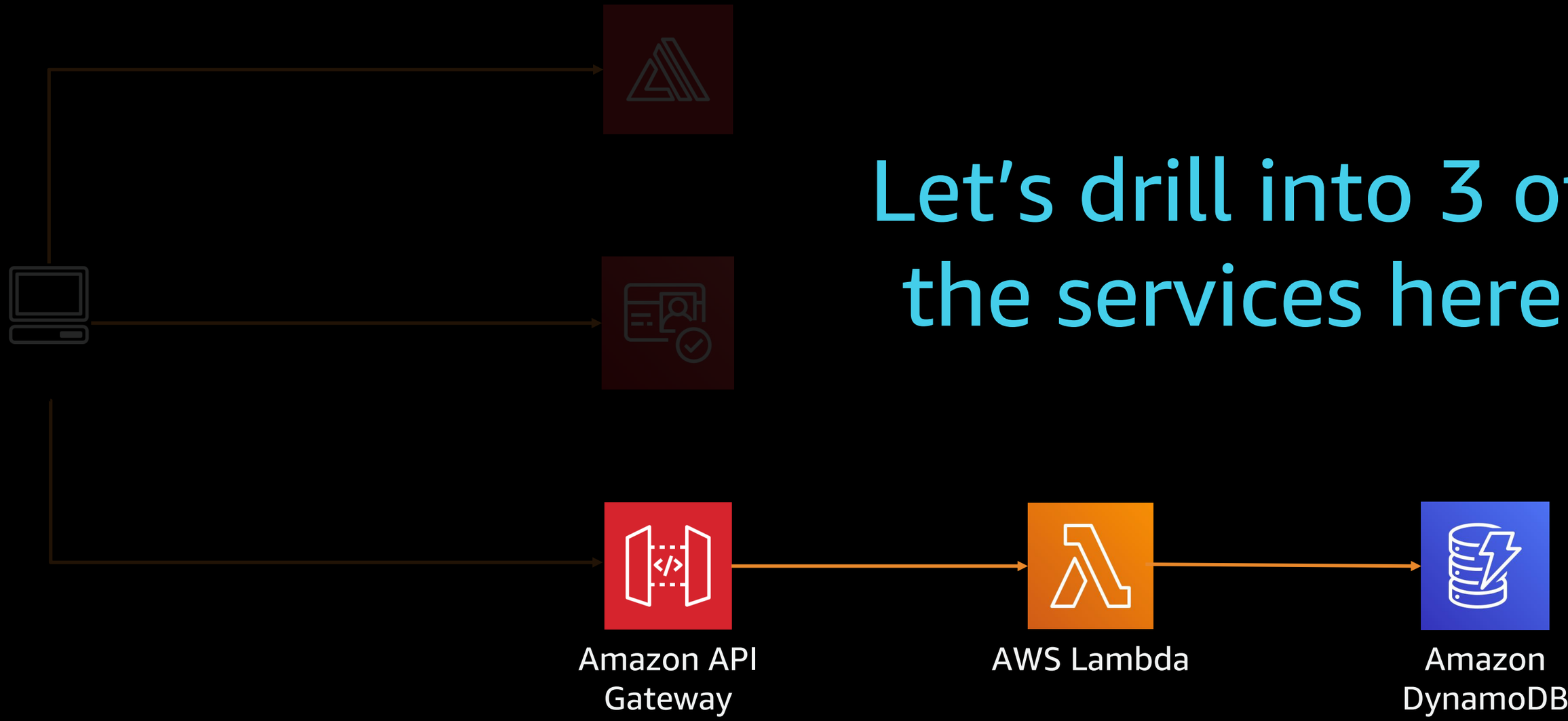






Troubleshooting / query workflow




Let’s drill into 3 of the services here
Amazon API Gateway
AWS Lambda
Amazon DynamoDB

CloudWatch metrics



© 2020, 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

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**

CloudWatch built-in metrics

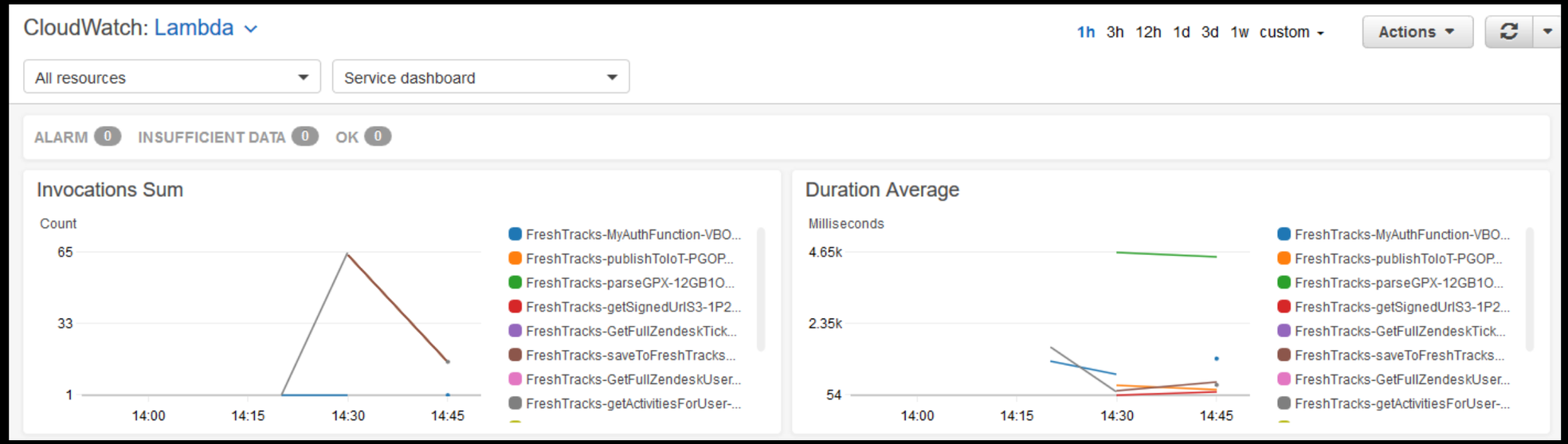


Amazon DynamoDB

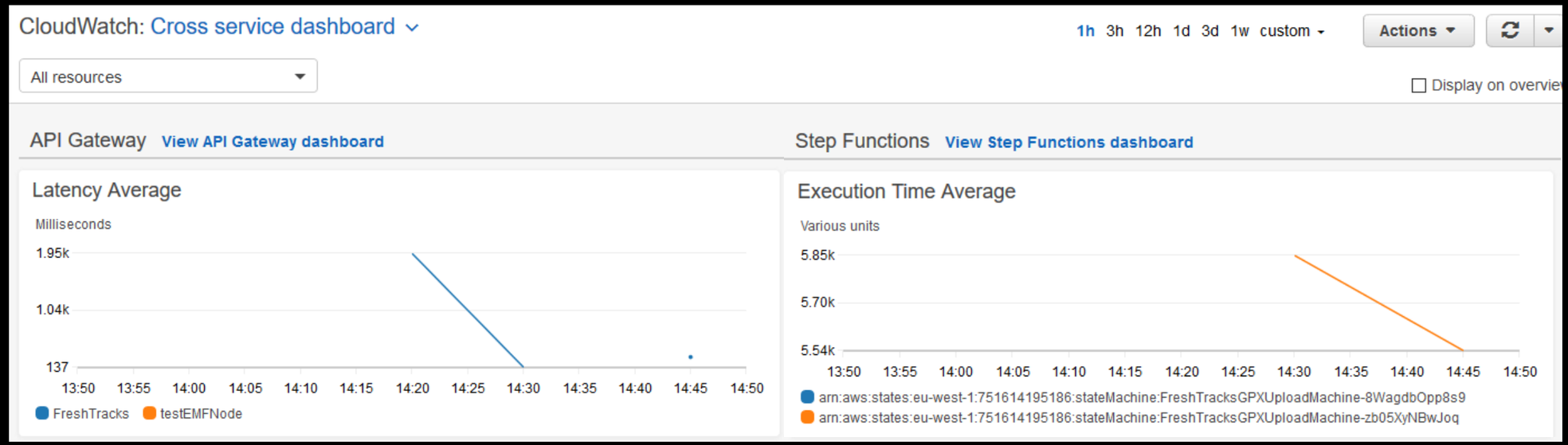
AccountMaxReads,
AccountMaxTableLevelReads,
AccountMaxTableLevelWrites,
AccountMaxWrites,
AccountProvisionedReadCapacityUtilization,
AccountProvisionedWriteCapacityUtilization,
ConditionalCheckFailedRequests,
ConsumedReadCapacityUnits,
ConsumedWriteCapacityUnits,
MaxProvisionedTableReadCapacityUtilization,
MaxProvisionedTableWriteCapacityUtilization,
OnlineIndexConsumedWriteCapacity,
OnlineIndexPercentageProgress,
OnlineIndexThrottleEvents,
PendingReplicationCount,
ProvisionedReadCapacityUnits,
ProvisionedWriteCapacityUnits,
ReadThrottleEvents, ReplicationLatency,
ReturnedBytes, ReturnedItemCount,
ReturnedRecordsCount,
SuccessfulRequestLatency, **SystemErrors**,
TimeToLiveDeletedItemCount,
ThrottledRequests, TransactionConflict,
UserErrors, **WriteThrottleEvents**

Per-service and cross-service dashboards

Per-service metrics dashboard



Cross-service metrics dashboard



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.

Creating custom metrics

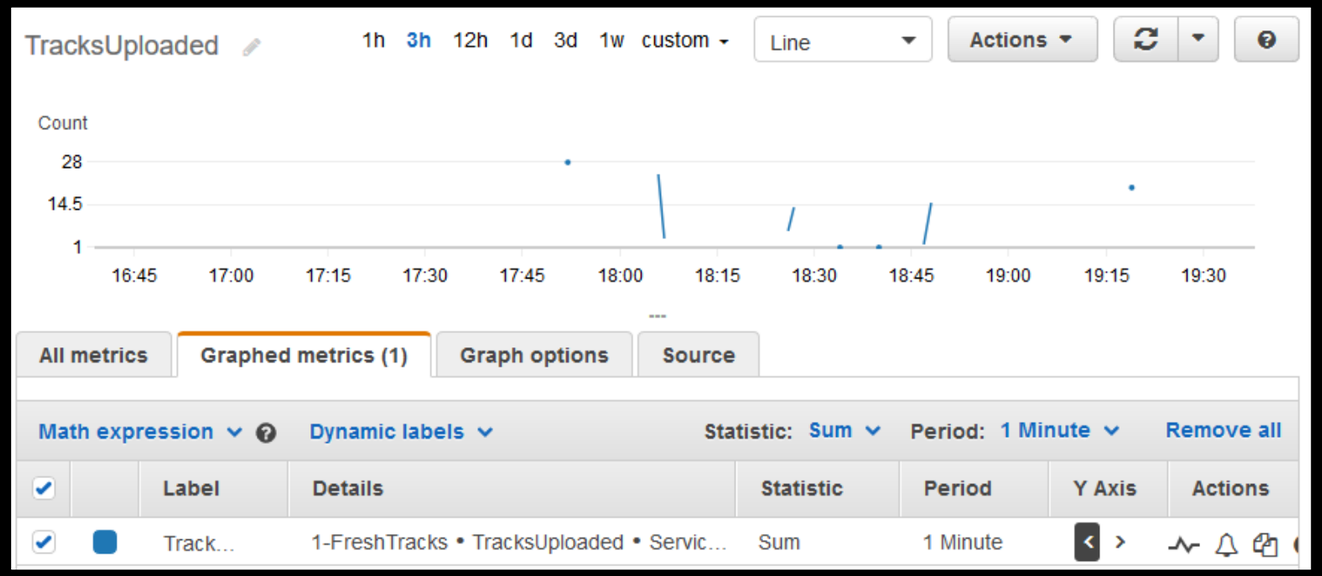
Useful for application, business, and operations metrics

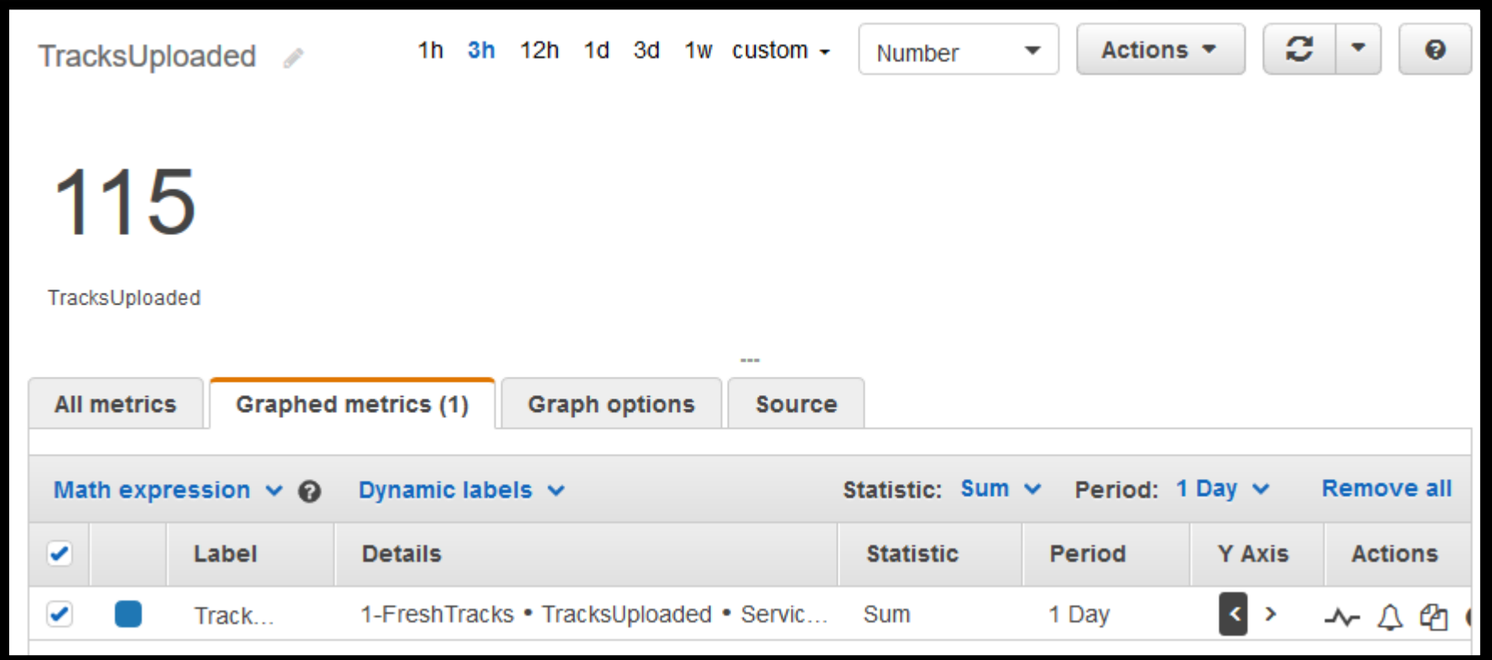
- Use built-in capabilities of the AWS SDK to call the CloudWatch putMetricData API call
- Charged by metric and by put call of data into a metric

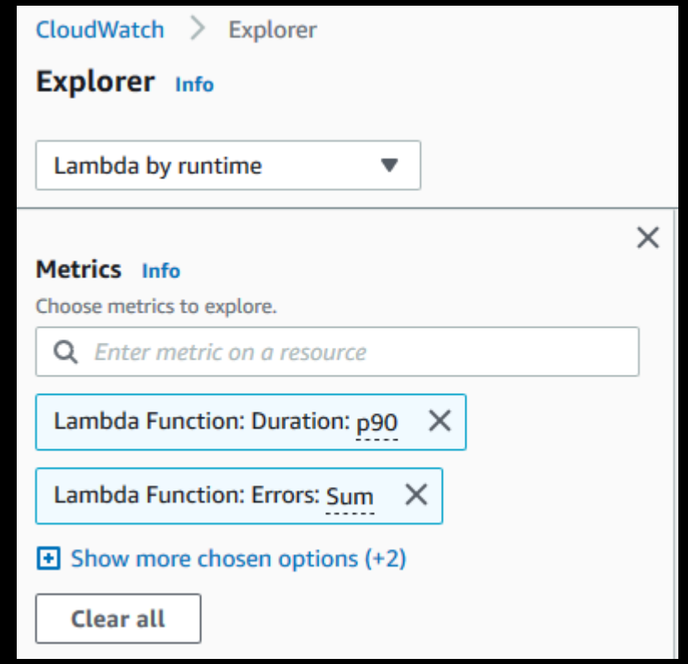
```
  msg.map((stat) => {
    params.MetricData.push({
      'MetricName': 'wait-times',
      'Dimensions': [
        {'Name': 'Type','Value': 'ride'},
        {'Name': 'Ride','Value': stat.rideId},
      ],
      'Unit': 'Seconds','Value': (stat.wait * 60)}
    )
  })
  // Send to CloudWatch
  console.log(await cloudwatch.putMetricData(params).promise())
}
```

Improved by embedded metric format (covering shortly)

Visualize with CloudWatch metrics graphs







CloudWatch Metrics Explorer

https://s12d.com/cw-me NEW

CloudWatch logging



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Built-in logging

API Gateway logging

- REST: Two levels of logging, ERROR and INFO
  - Set globally in stage, or override per method
  - Optionally log method request / body content
- HTTP APIs and WebSocket APIs with logging variables

Lambda logging

- Logging directly from your code with your language's equivalent of `console.log()` – basic request information included
- JSON structured logging via `PutMetricData` API or embedded metric format, which includes invocation information



Metric filters

- Build metrics based on log filters, jump back to logs
- Two subscription filters per log group NEW

Export logs to Amazon ElastiCache or Amazon S3

- Explore with Kibana or Amazon Athena / Amazon QuickSight

CloudWatch embedded metric format

AUTOMATICALLY GENERATE METRICS FROM STRUCTURED CLOUDWATCH LOGS

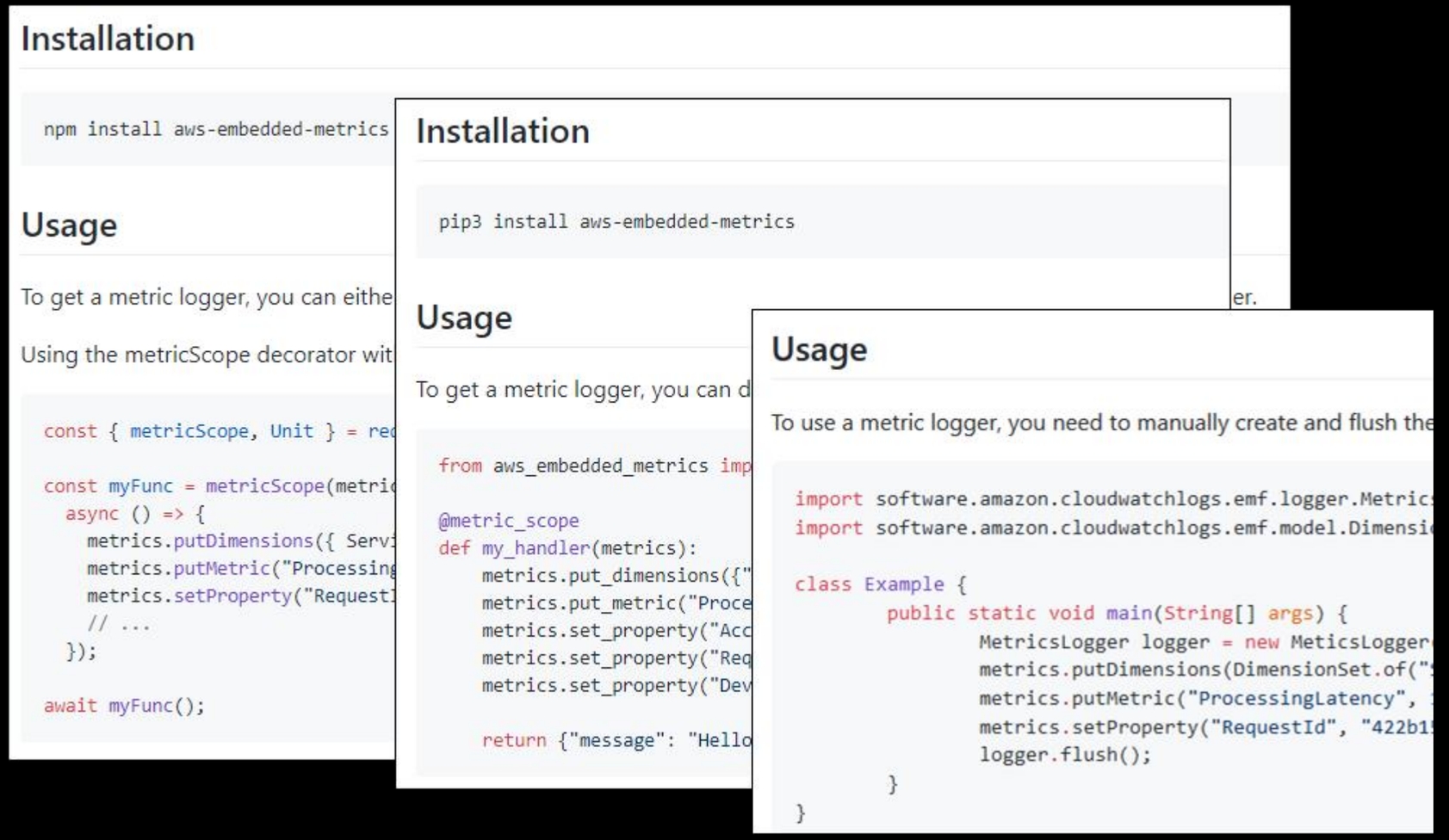
Embed custom metrics alongside detailed log event data

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

Asynchronous

Open-source client libraries available for

- Node.js
- Python
- Java NEW



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

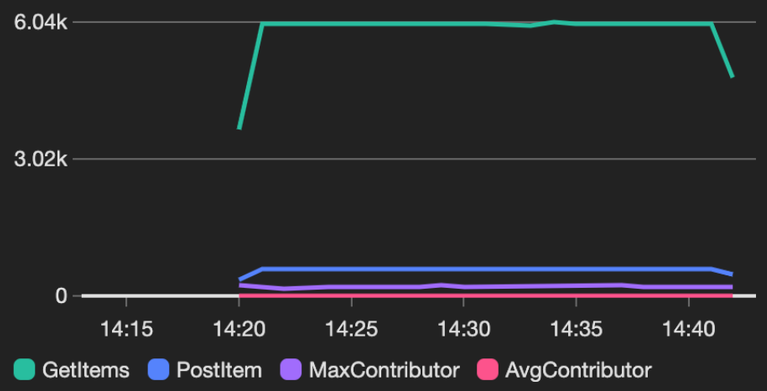
CloudWatch embedded metric format

```
message = {
   PriceInCart    100
   QuantityInCart    2
   ProductId     a23390f3
   CategoryId     bca4cec1
   UserId     31ba3930
   CartId     58dd189f
   Environment      prod
   LogLevel     INFO
   Timestamp      2019-12-11 12:44:40.300473
   Message     Added 2 items 'a23390f3' to cart
'58dd189f'
```

```
[...]
"_aws": {
"functionVersion": "$LATEST",
  "Timestamp": 1576064416496,
    "CloudWatchMetrics": [{
      "Namespace": "ecommerce-cart",
      "Dimensions": [
        ["Environment", "CategoryId"]
      ],
      "Metrics": [
        {"Name": "PriceInCart", "Unit": "None"},
        {"Name": "QuantityInCart", "Unit": "None"}
      ]
    }]
  }
}
```

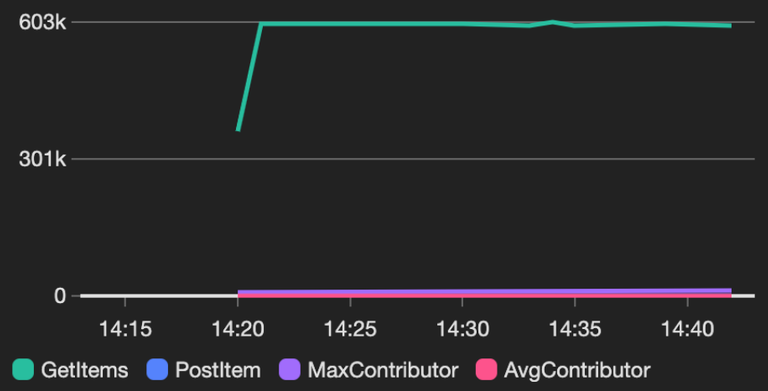
Requests

Various units



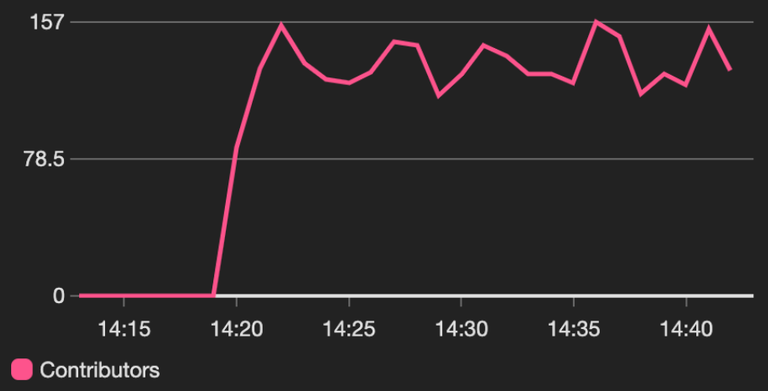
Items

No unit



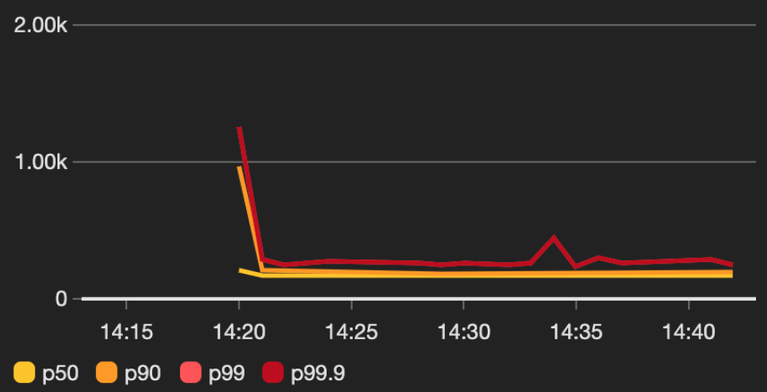
Number of contributors

No unit



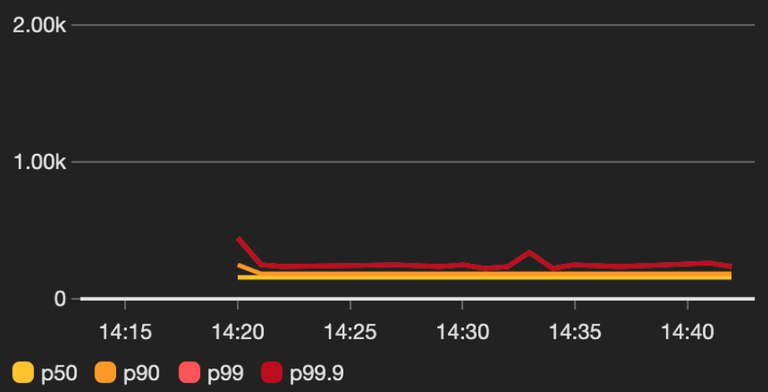
API Gateway Latency

Milliseconds



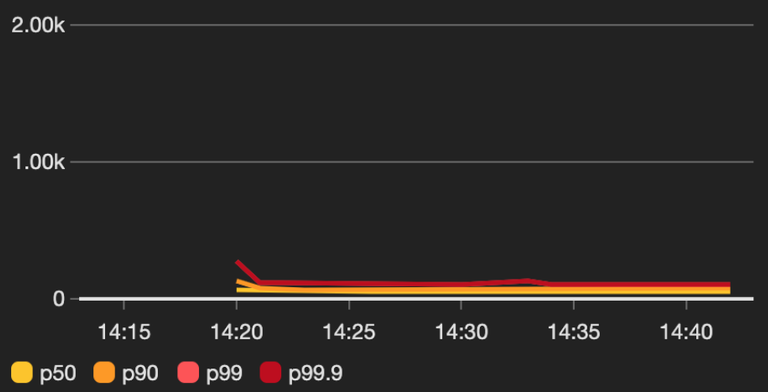
GetItems Duration

Milliseconds



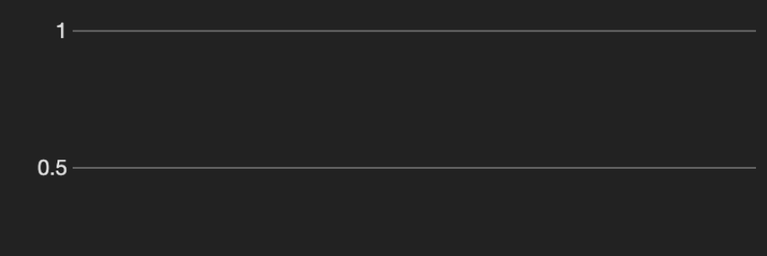
PostItem Duration

Milliseconds



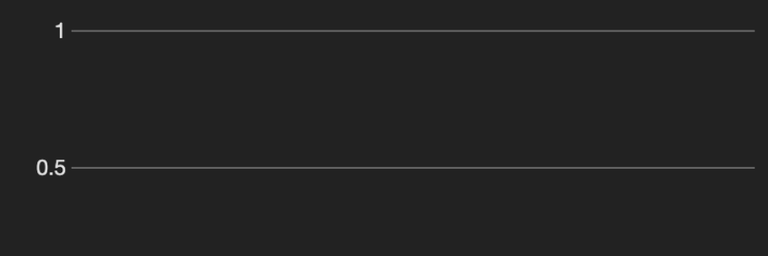
Backend Errors

Count



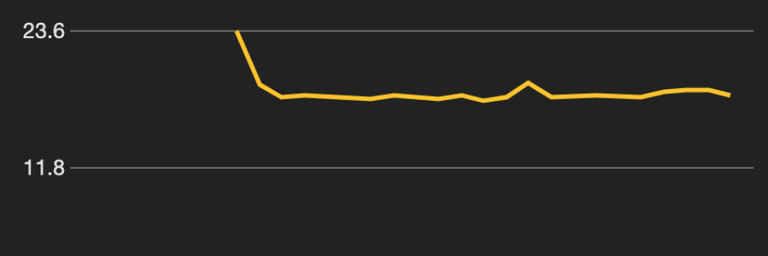
Throttles

Count



Concurrent Executions

Count



Amazon CloudWatch Logs Insights



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

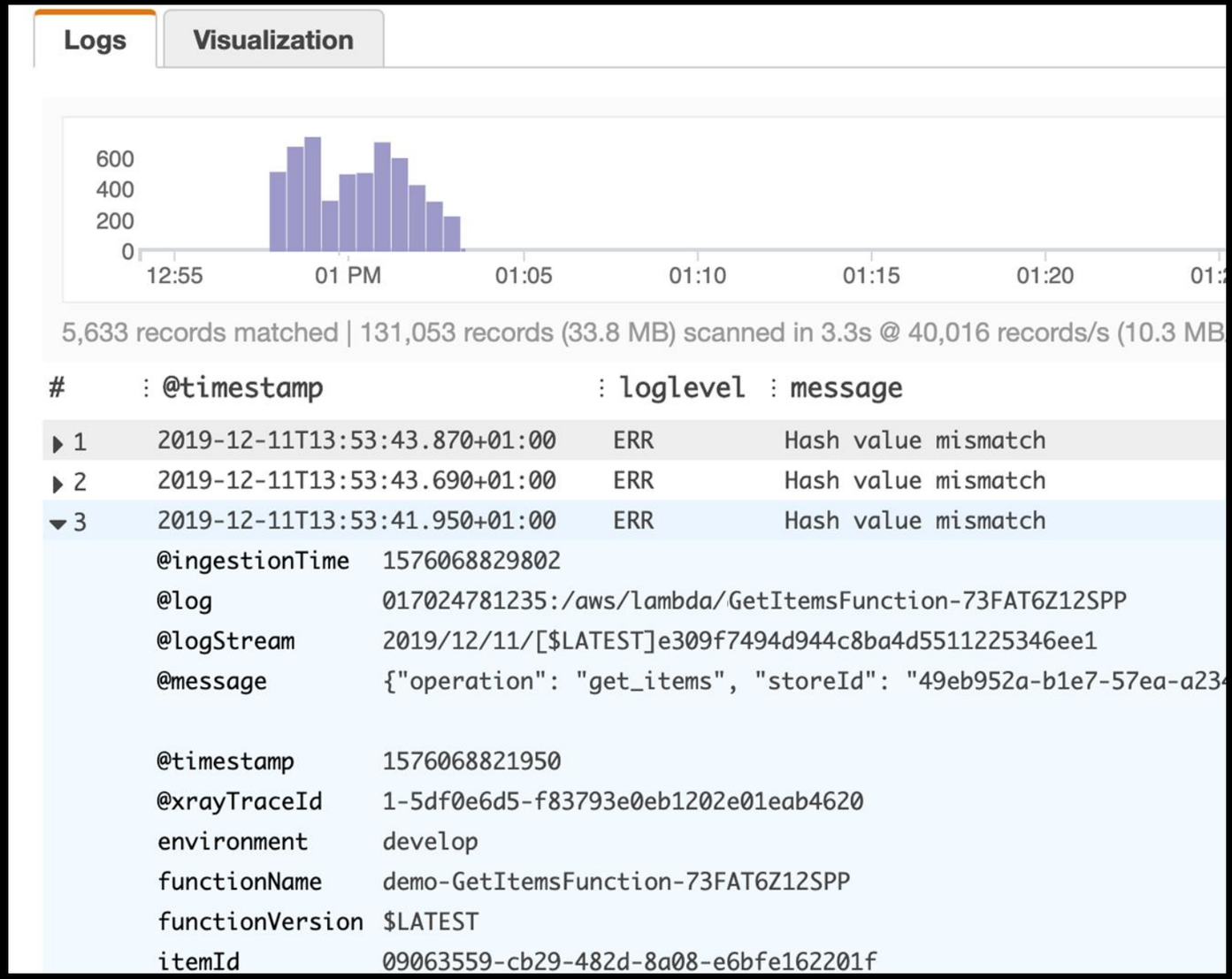
Amazon CloudWatch Logs Insights

Interactively search and analyze your log data in Amazon CloudWatch Logs

Drive actionable intelligence to address operational issues without needing to provision servers or manage software

- Processes structured log data
- Flexible purpose-built query language
- Query up to 20 log groups
- Save queries

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



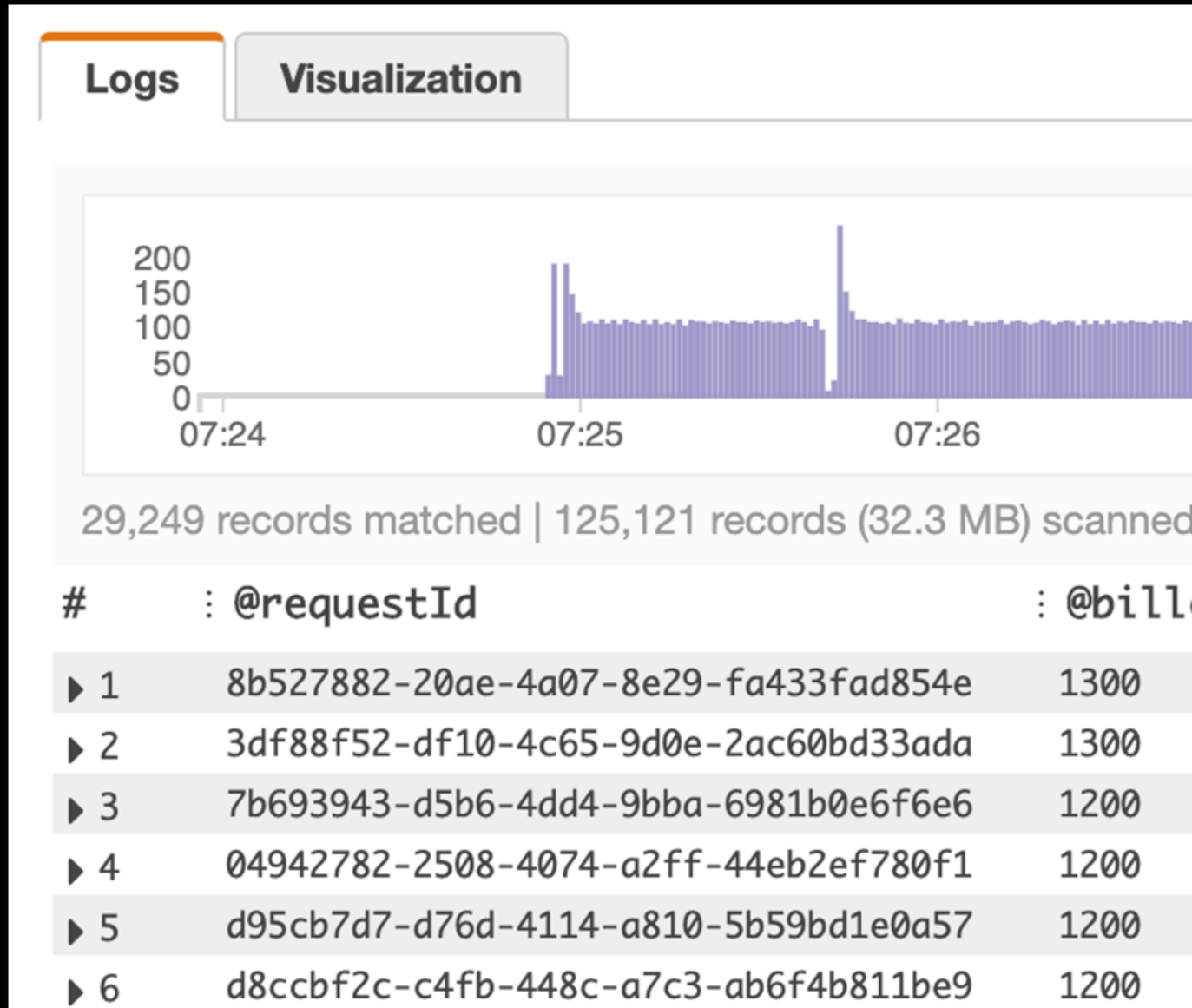
Top 100 most expensive invocations

```
filter @type = "REPORT"

| fields @requestId,
@billedDuration

| sort by @billedDuration
desc

| limit 100
```



Checking Lambda performance

P90 latency, total invokes, and max latency for each 5-minute window

```
filter @type =
"REPORT"
| stats
avg(@duration),
max(@duration),
min(@duration),
pct(@duration,
90),
count(@duration)
by bin(5m)
```

Logs | Visualization | Export results | Add to dashboard

Showing 48 of 2,352 records matched

14,276 records (4.8 MB) scanned in 3.8s @ 3,745 records/s (1.3 MB/s)

Show histogram

#	bin(5m)	avg(@duration)	max(@duration)	min(@duration)	pct(@duration, 90)	count(@duration)
1	2020-10-21T22:00:	2665.7931	5299.88	241.51	4454.03	48
2	2020-10-21T21:45:	2711.1933	4923.12	237.99	4477.22	48
3	2020-10-21T21:30:	2567.0235	5169.22	304.31	4253.03	48
4	2020-10-21T21:15:	2671.319	5105.17	338.53	4199.89	48
5	2020-10-21T21:00:	2528.9225	4577.06	231.42	4114	48
6	2020-10-21T20:45:	2683.5394	5182.84	319.67	4127.31	48
7	2020-10-21T20:30:	2443.1156	5007.75	250.96	3692.27	48
8	2020-10-21T20:15:	2627.3131	5049.13	304.19	4218.91	48
9	2020-10-21T20:00:	2705.4185	5044.18	251.02	4620.97	48
10	2020-10-21T19:45:	2693.3875	5037.74	270.23	4496.73	48
11	2020-10-21T19:30:	2692.1761	13310.84	255.78	4237.32	96

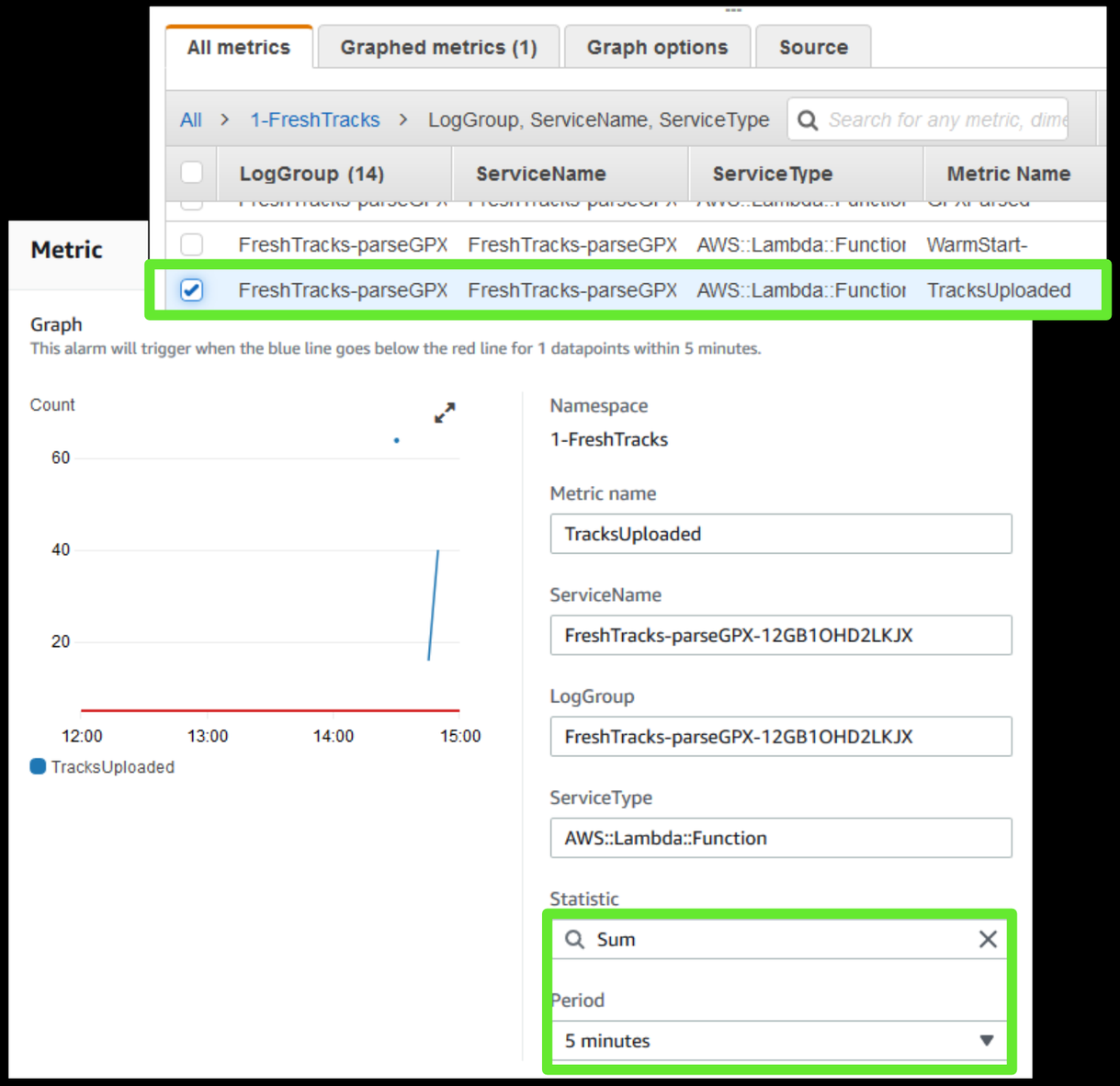
Creating alerts

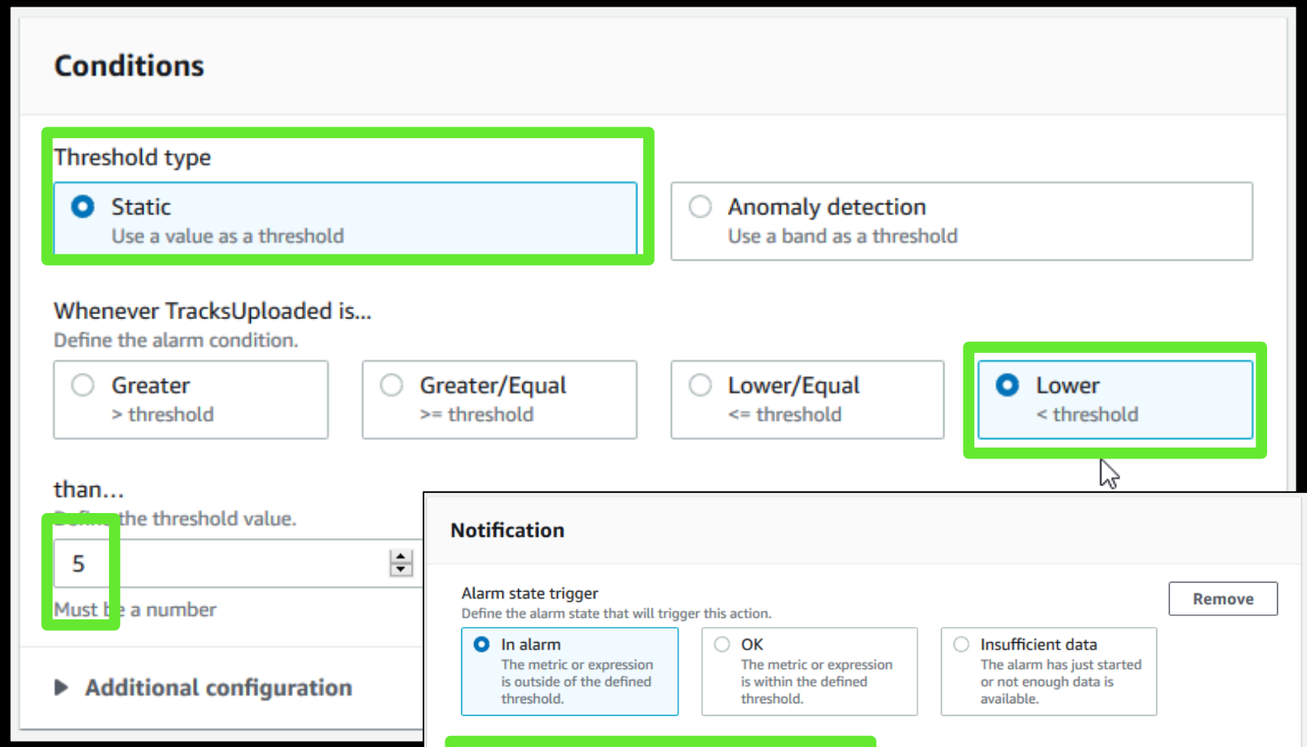


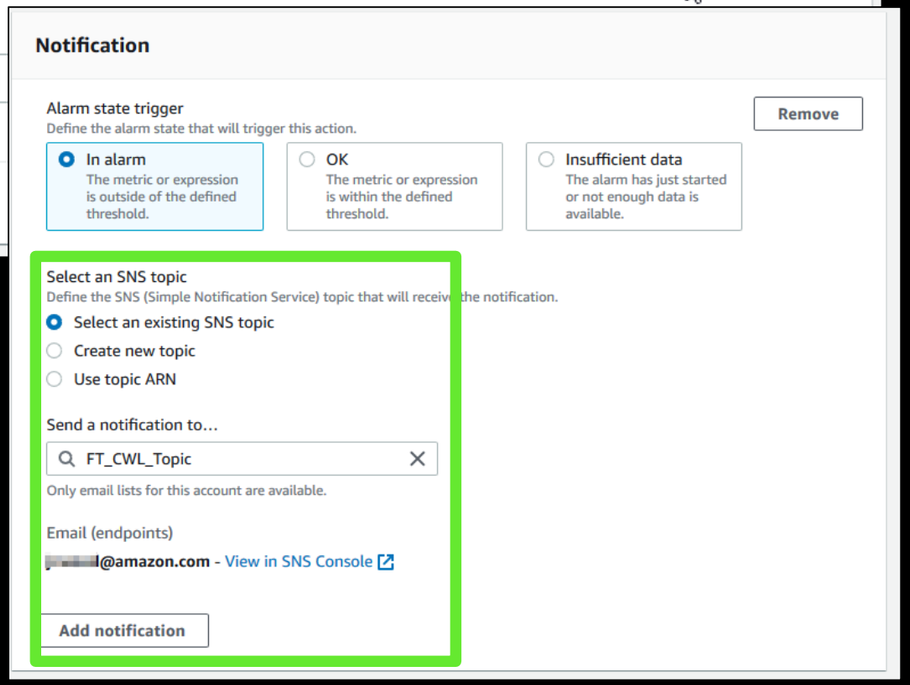
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

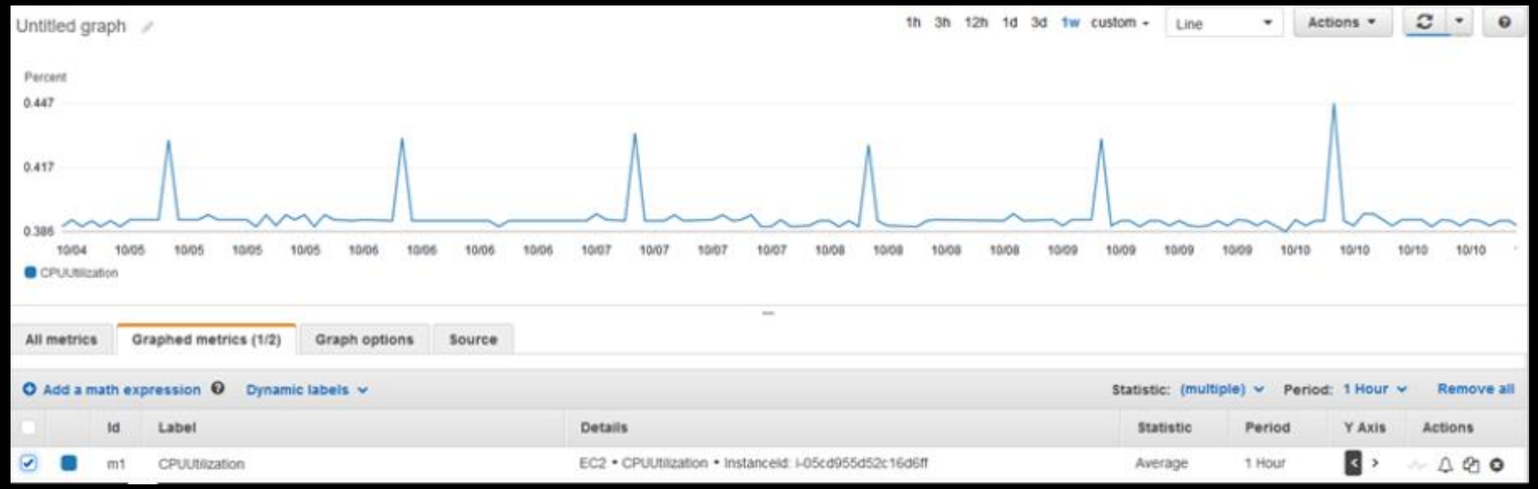
Creating CloudWatch alarms

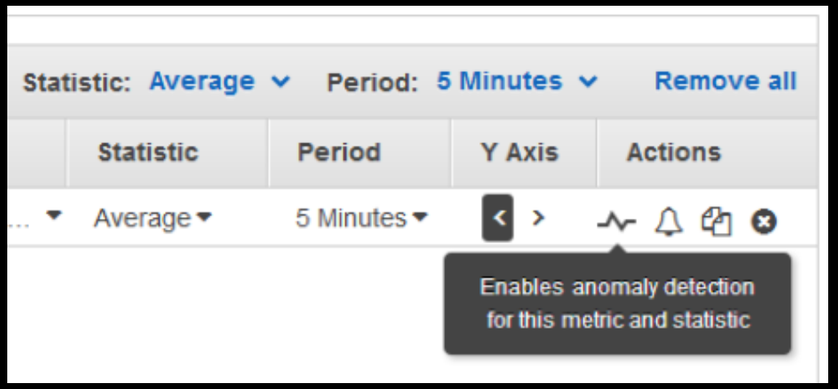


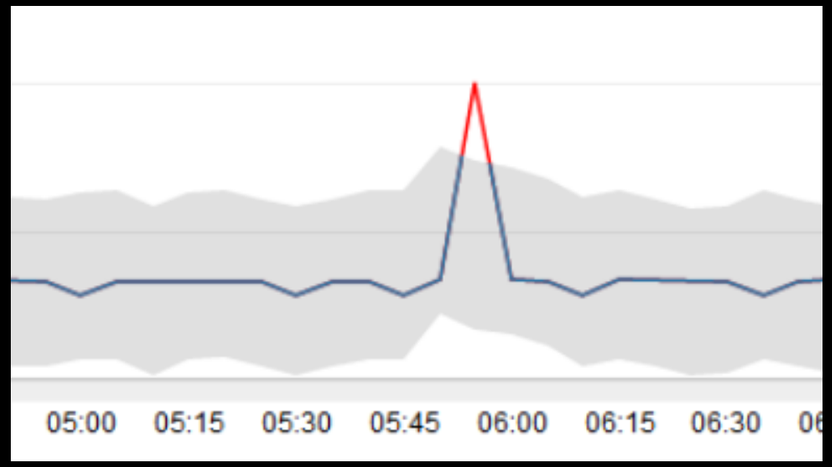


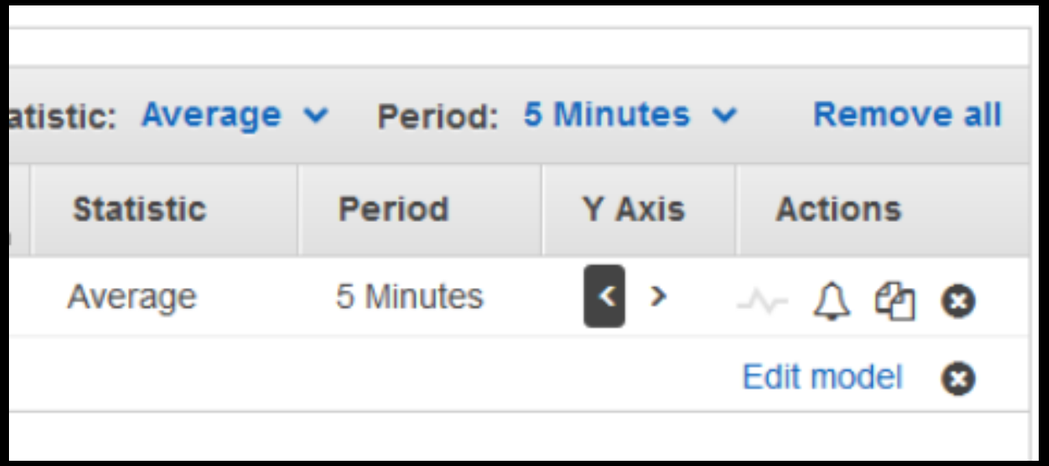


Using CloudWatch anomaly detection alarms









Tracing



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS X-Ray

End-to-end view of requests flowing through an application

- Lambda – instruments incoming requests for all supported languages and can capture calls made in code

Enable X-Ray Tracing

```
Globals:
  Function:
    Tracing: Active
```

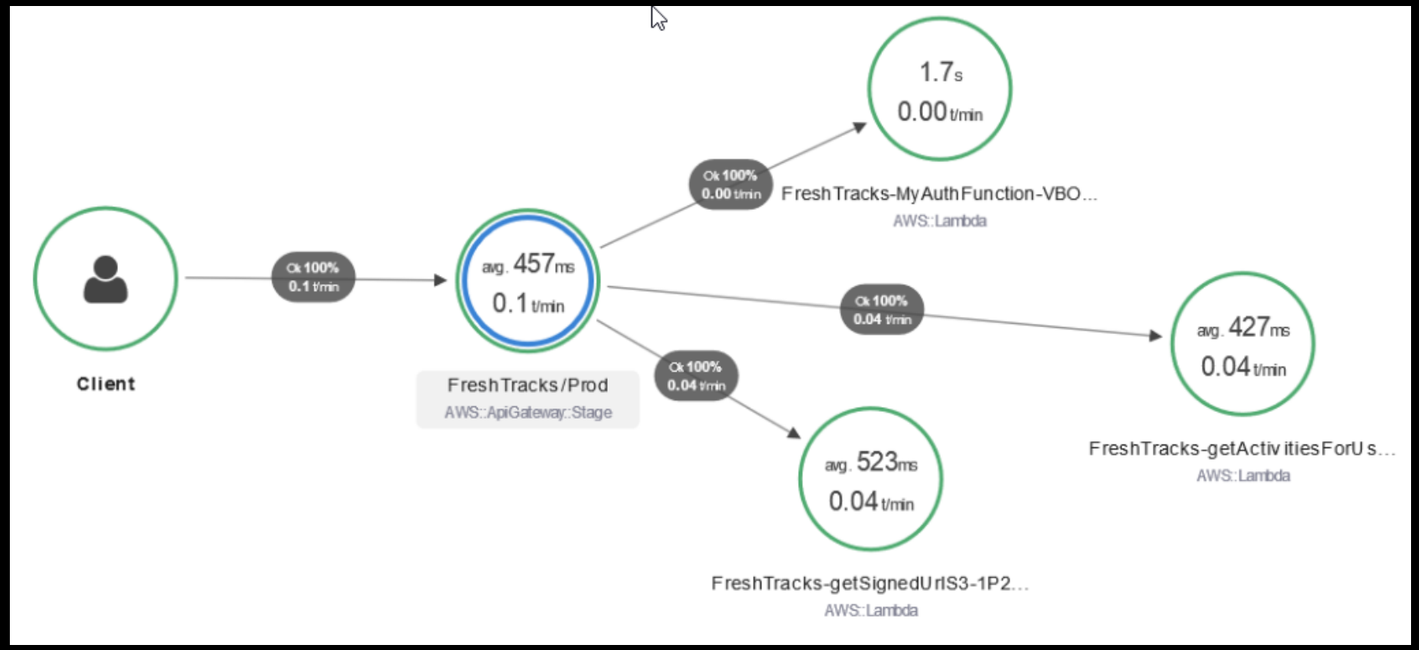
- API Gateway – inserts a tracing header into HTTP calls as well as reports data back to X-Ray itself

Enable active tracing Info

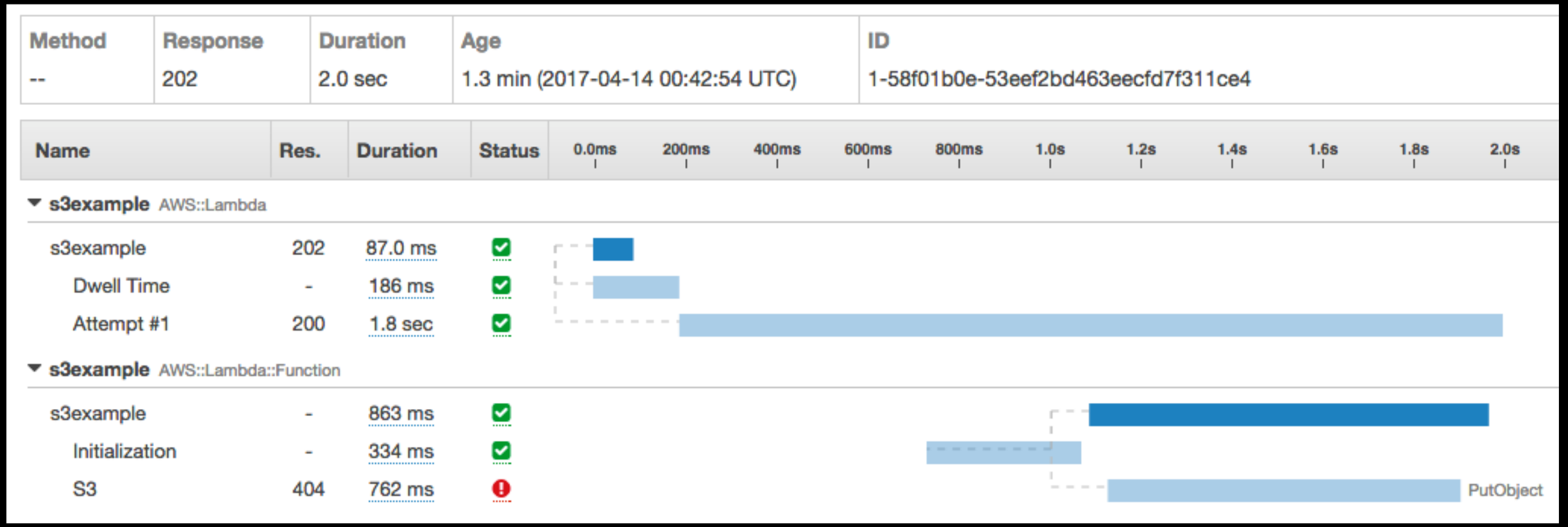
```
Globals:
  Api:
    TracingEnabled: True
```

```
const AWSXRay = require('aws-xray-sdk-core');
const AWS = AWSXRay.captureAWS(require('aws-sdk'));

const documentClient = new AWS.DynamoDB.DocumentClient();
```



X-Ray trace example



AWS X-Ray Analytics example



USER	COUNT	%	Δ
William	189	13.96%	0.00%
Olivia	279	20.61%	0.17%

HTTP STATUS CODE	COUNT	%	Δ
200	1218	89.96%	32.87%
400	16	1.18%	0.39%

CloudWatch ServiceLens

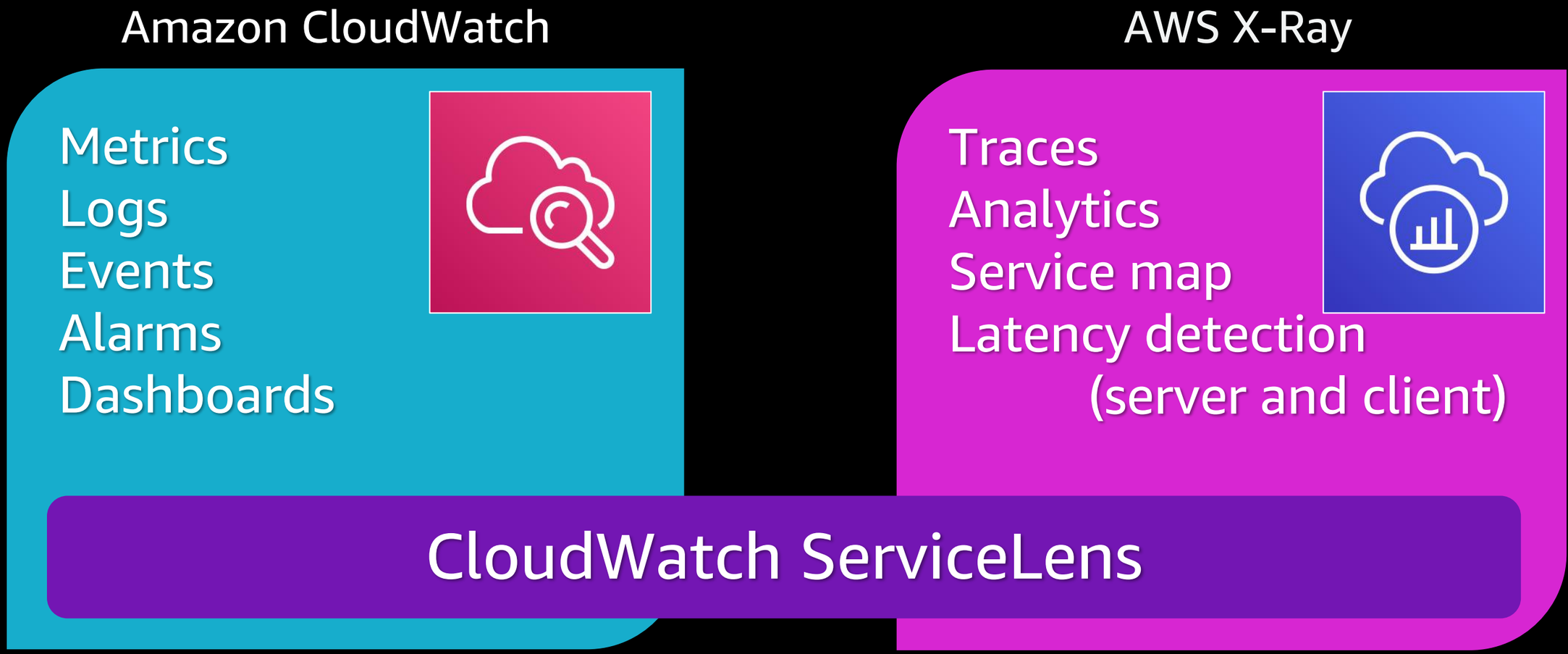


AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

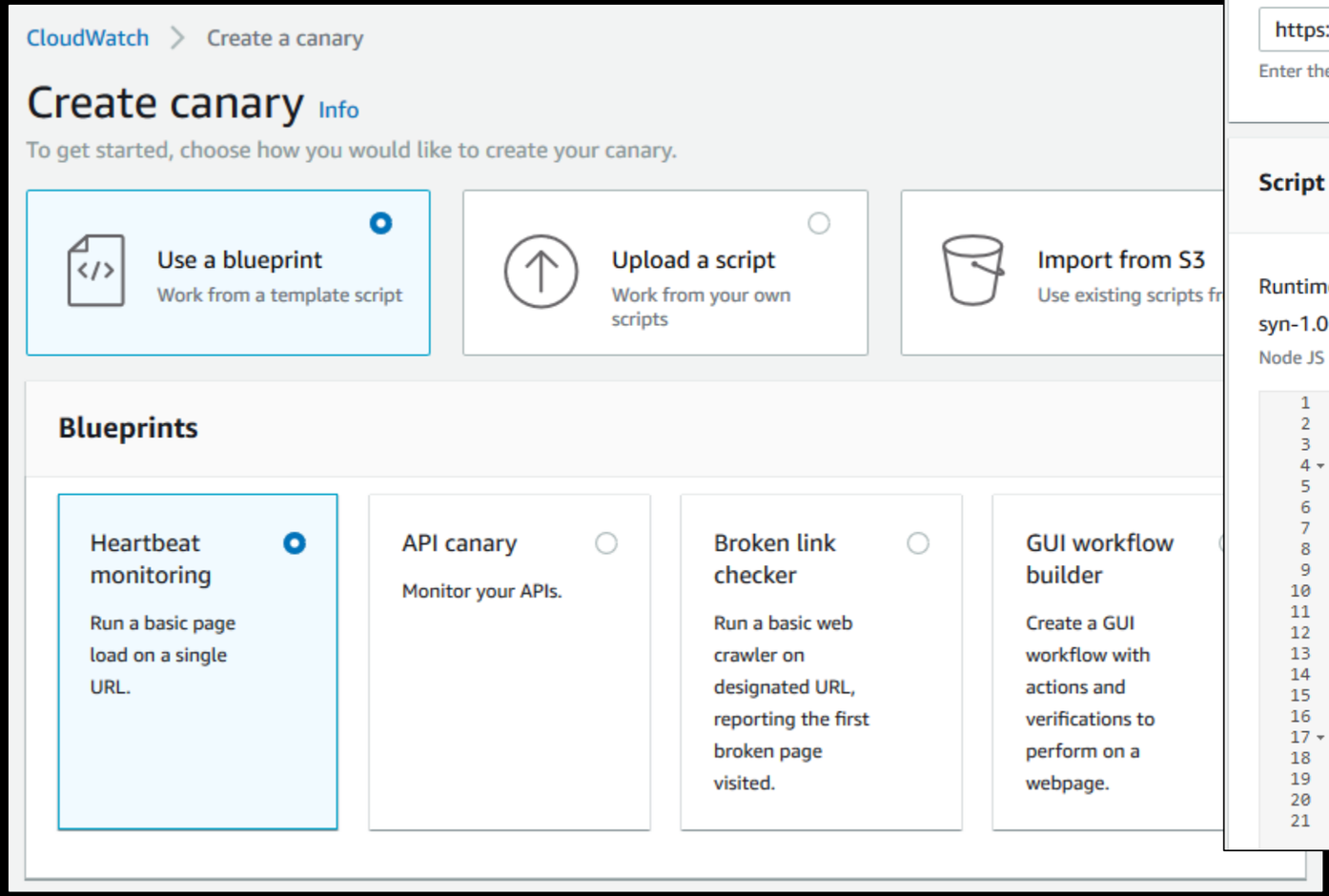
CloudWatch ServiceLens

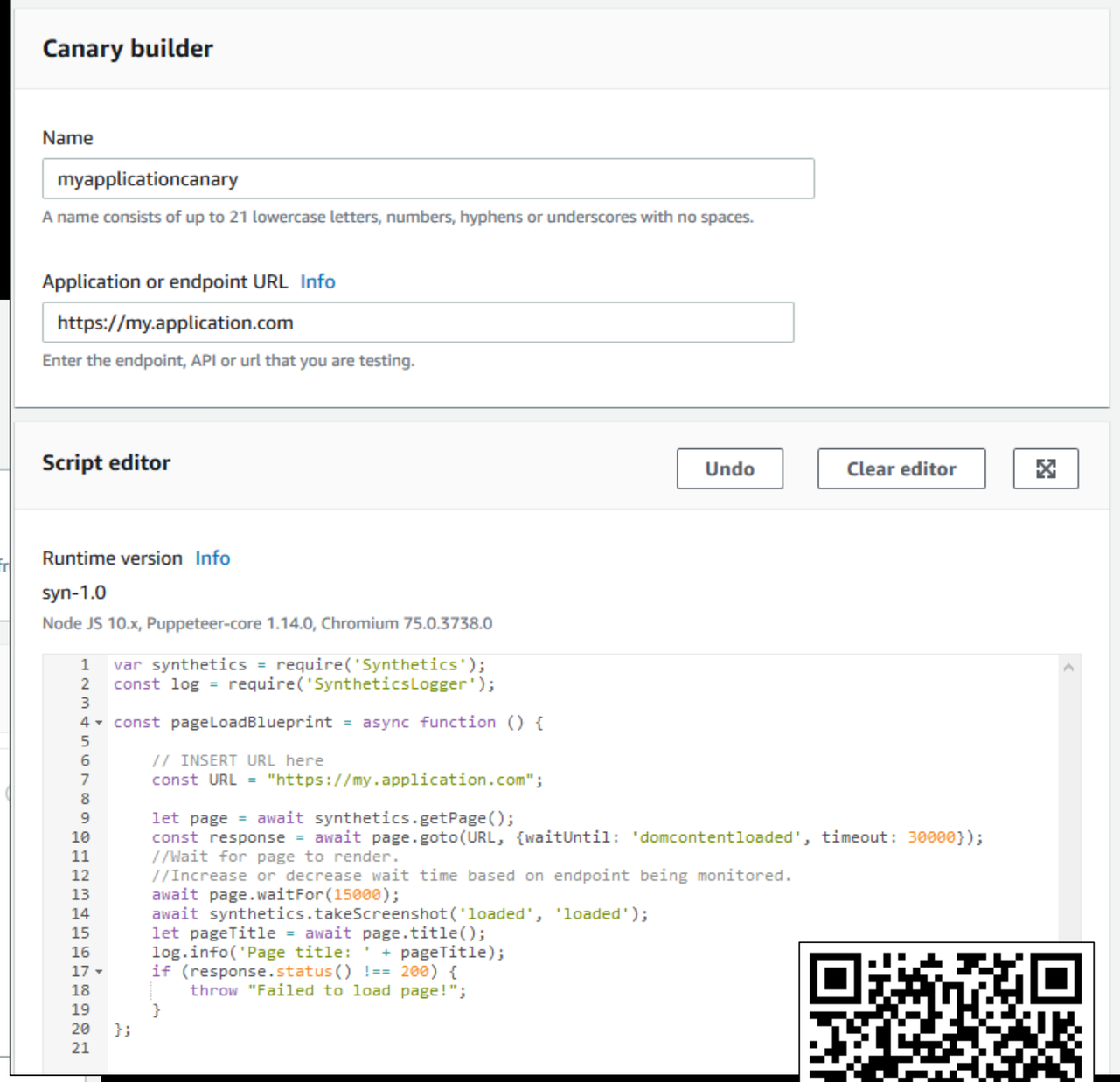
- Ties together CloudWatch metrics and logs, in addition to traces from AWS X-Ray
- Gives you a complete view of your applications and their dependencies



CloudWatch Synthetics

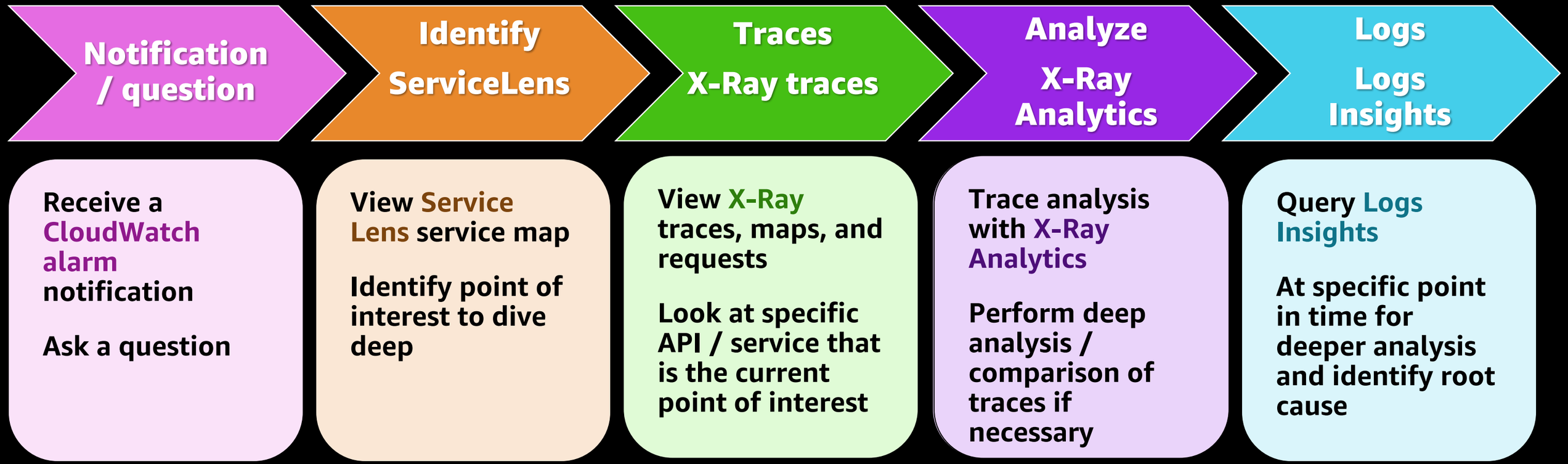
MONITOR WEB APPLICATIONS USING MODULAR, LIGHTWEIGHT CANARY TESTS





https://s12d.com/cw-synthetics-canaries

Troubleshooting / query workflow



NEW

Lambda Extensions: In preview

A new way to easily integrate Lambda with your favorite monitoring, observability, security, and governance tools



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda extensions

NEW

A NEW WAY TO INTEGRATE WITH THE LAMBDA EXECUTION ENVIRONMENT

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

Capture diagnostic information before, during, and after function invocation

Automatically instrument your code without needing code changes

Fetch configuration settings or secrets before the function invocation

Detect and alert on function activity through hardened security agents

Using extensions

NEW

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

Logs API

- Consume platform and function logs directly from Lambda



https://s12d.com/extensions-logs

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

Extensions available today

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





















Amazon CloudWatch
Lambda Insights



AWS AppConfig











NEW

Lambda container image support

Deploy functions defined as container images to Lambda



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda container image support

DEPLOY FUNCTIONS DEFINED AS CONTAINER IMAGES TO LAMBDA

NEW

https://s12d.com/lambda-containers

1. Create container image using familiar tools
2. Deploy image to Amazon Elastic Container Registry (Amazon ECR)
3. Create function using AWS CLI, AWS Management Console, AWS SAM
4. Upon invoke, Lambda starts and runs the container as defined in the image

Deployed images need to be event-driven, stateless applications

Can use set of reliable and secure base images provided by Lambda to build functions

Locally test and deploy functions using either Docker CLI or AWS SAM

Lambda Powertools

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 NEW

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



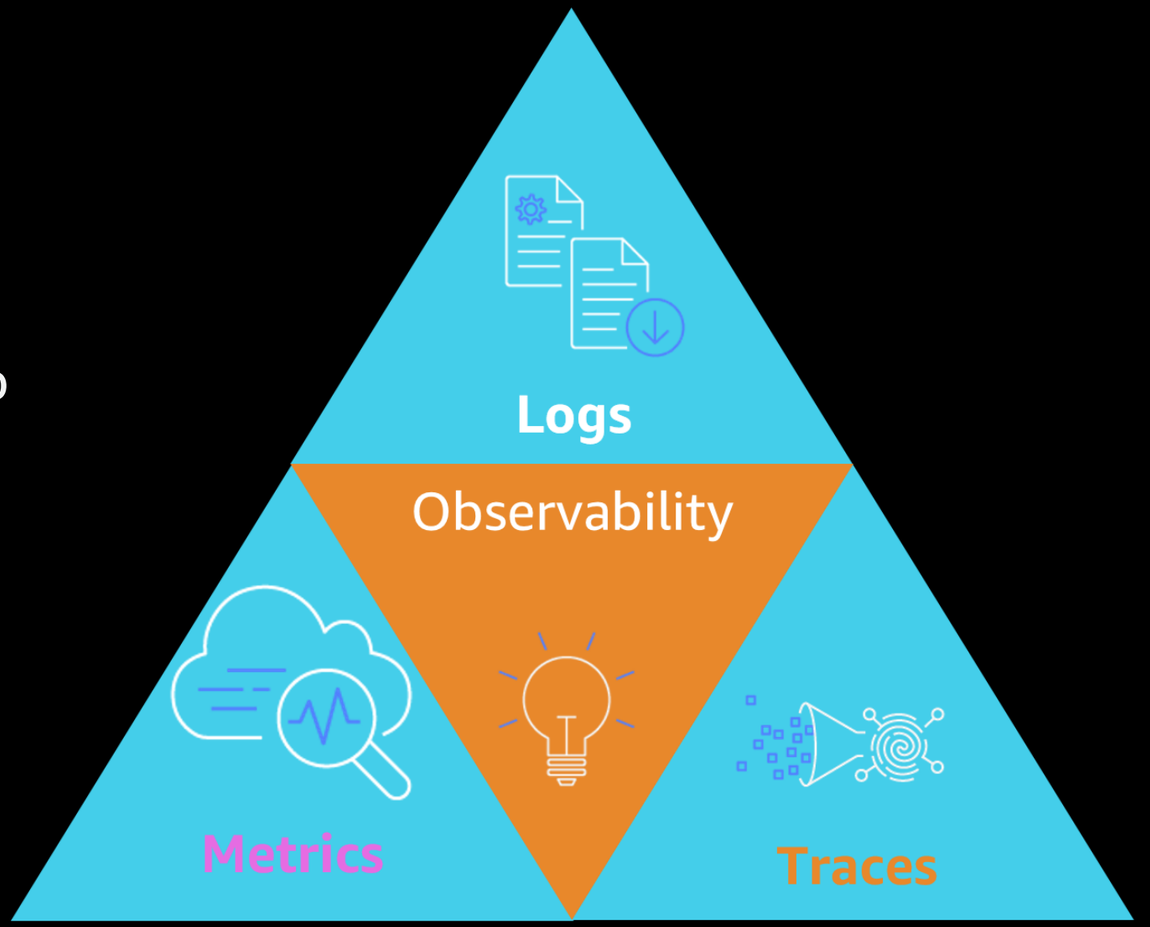
Summary

Observability is more than just monitoring metrics

- A culture focused on gathering as much data required to make the right data-driven decisions
- Understanding states between up and down

Amazon CloudWatch and AWS X-Ray give you tools to gain visibility into how your systems are performing

- Embedded metric format is a powerful 2 for 1
- Use Logs Insights to interactively query and analyze log data
- ServiceLens gives you that “single-pane” view into your applications
- Extensions – more easily use your existing tools



Serverlessland.com

Home Blogs Videos Learn Events About

Welcome to **Serverless** Land

This site brings together all the latest blogs, videos, and training for AWS Serverless. Learn to use and build apps that scale automatically on low-cost, fully-managed serverless architecture.



Thank you!

Julian Wood

@julian_wood

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.