THUNDRA™

Building 100% Serverless Blog Site Application with Right Observability

July 10, 2019

Serkan ÖZAL

WHO AM I?

- Founder & CEO/CTO @ Thundra
- Co-organizer of
  - Serverless Turkey Meetup
  - ServerlessDays İstanbul 2019 (3rd of October)
- Oracle Open Source Contributor
- In serverless era since 3 years
- PhD candidate

@serkan_ozal

serkan-ozal

AGENDA

- What We Gonna Do?
- What We Gonna Use?
- The Architecture
- Monitoring with CloudWatch
- Monitoring with Thundra
    - How to Setup
    - Local & Distributed Tracing
    - Invocation Tagging
    - Async Monitoring
- More Thundra Features

WHAT WE GONNA DO?

The Blog Site Application

- Send Blog Post
- Get Blog Post
- Search Blog Post
- Delete Blog Post

Reference implementation available at

github.com/thundra-io/serverless-blog-site-workshop

WHAT WE GONNA USE?

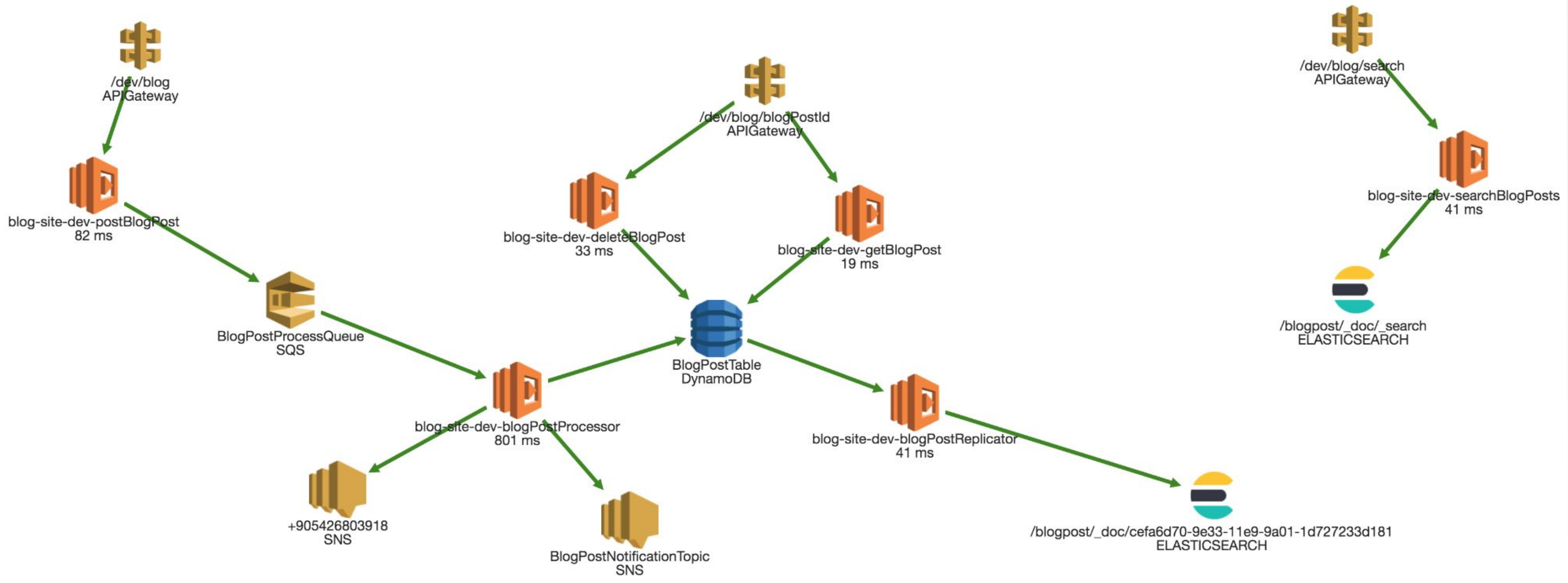
aws

node js

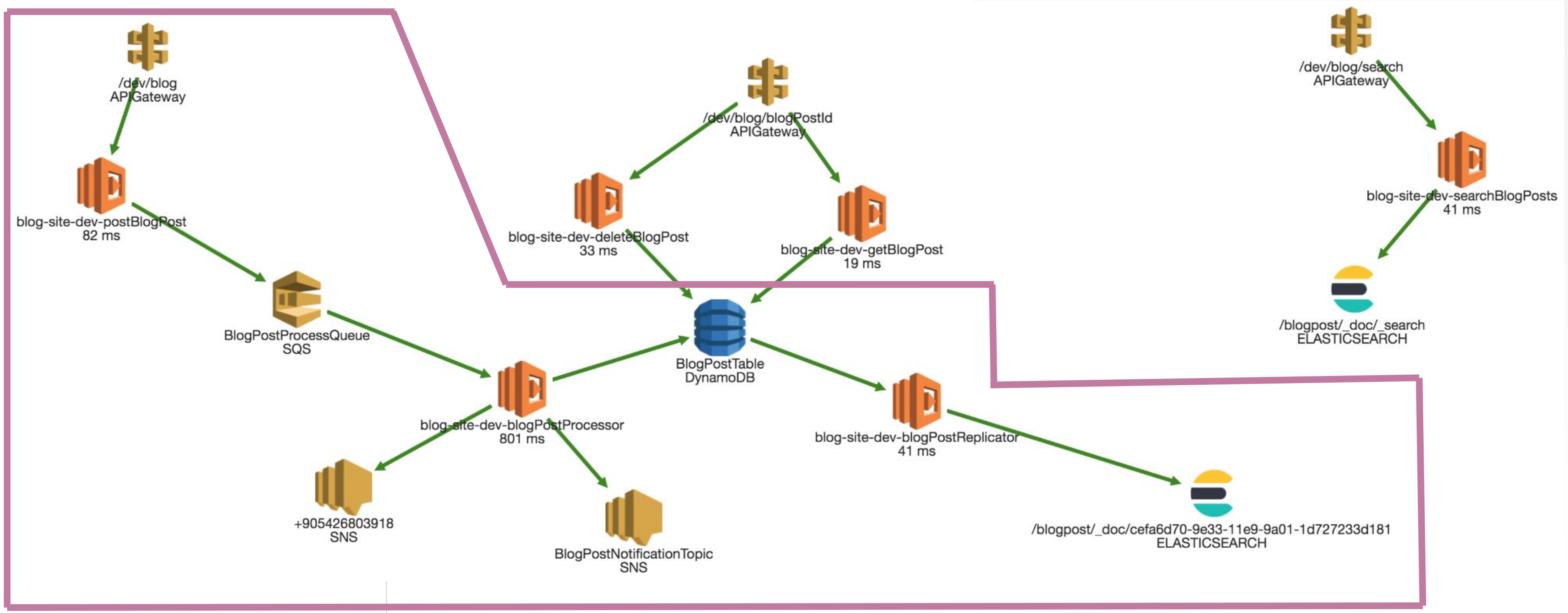
serverless

THUNDRA

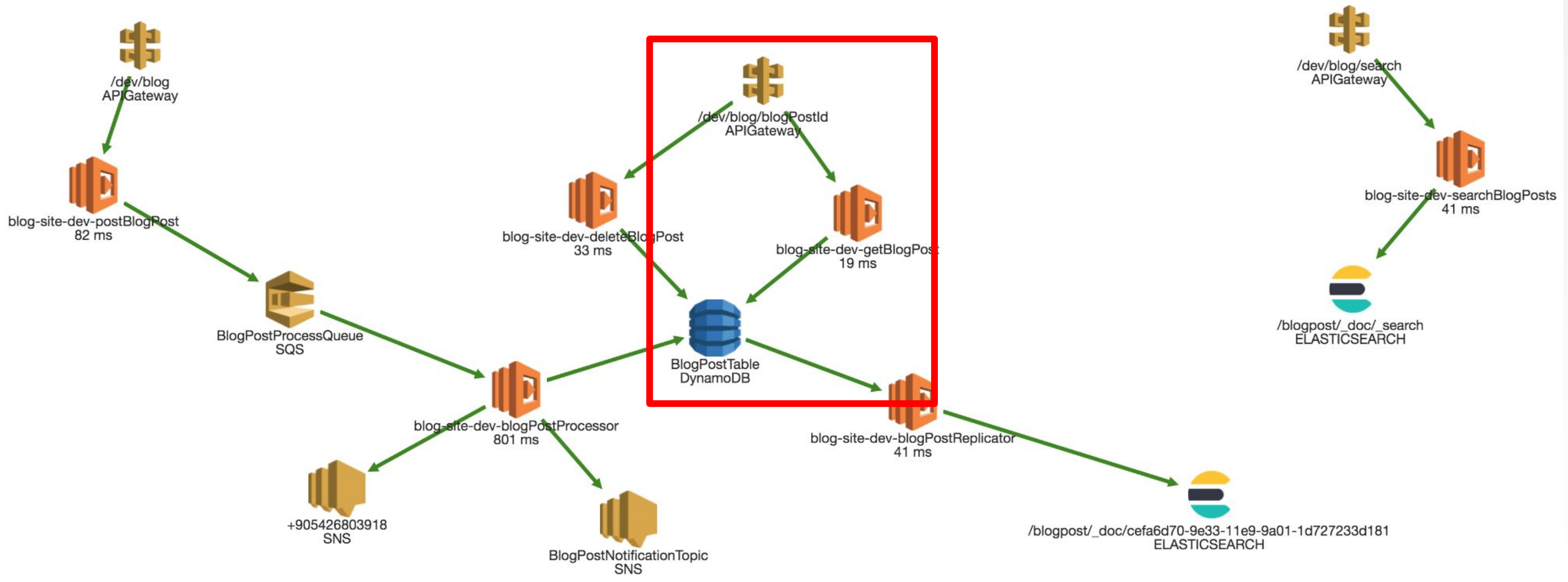
THE ARCHITECTURE


/dev/blog
APIGateway
blog-site-dev-postBlogPost
82 ms
BlogPostProcessQueue
SQS
/dev/blog/blogPostId
APIGateway
blog-site-dev-deleteBlogPost
33 ms
blog-site-dev-getBlogPost
19 ms
BlogPostTable
DynamoDB
blog-site-dev-blogPostProcessor
801 ms
+905426803918
SNS
BlogPostNotificationTopic
SNS
blog-site-dev-blogPostReplicator
41 ms
/blogpost/_doc/cefa6d70-9e33-11e9-9a01-1d727233d181
ELASTICSEARCH
/dev/blog/search
APIGateway
blog-site-dev-searchBlogPosts
41 ms
/blogpost/_doc/_search
ELASTICSEARCH

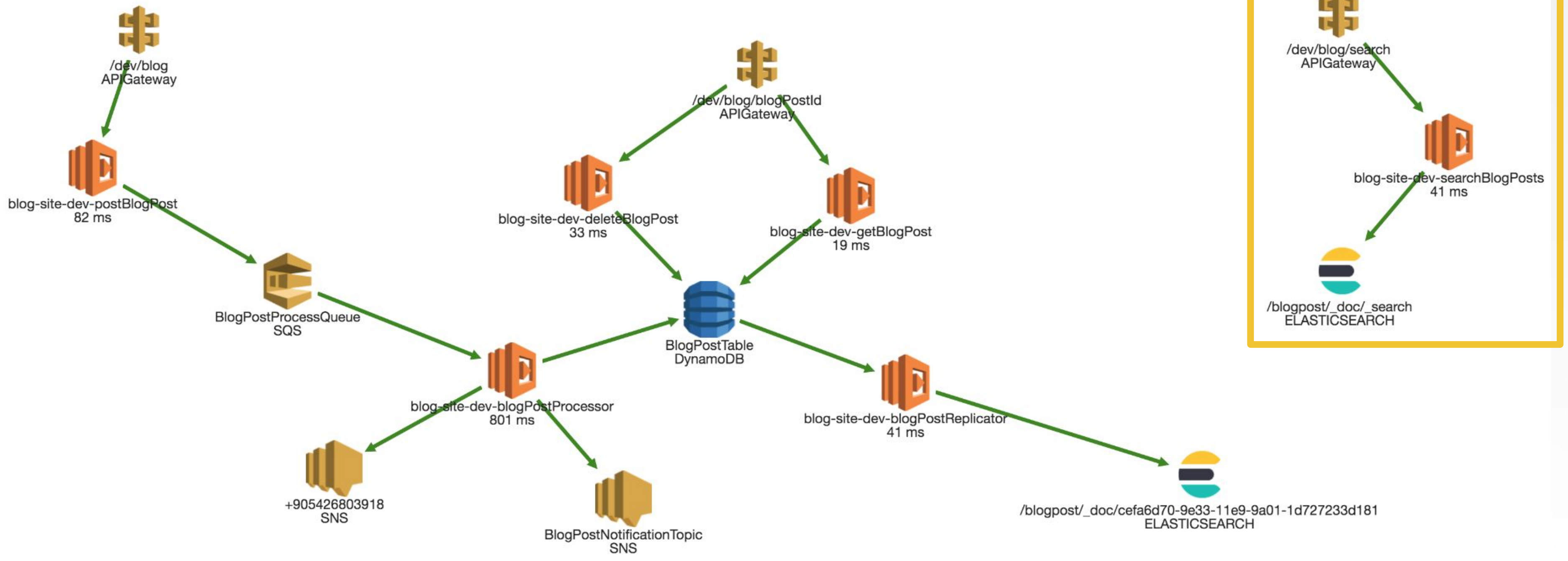
Send Blog Post


/dev/blog
APIGateway
blog-site-dev-postBlogPost
82 ms
BlogPostProcessQueue
SQS
blog-site-dev-blogPostProcessor
801 ms
+905426803918
SNS
BlogPostNotificationTopic
SNS
/dev/blog/blogPostId
APIGateway
blog-site-dev-deleteBlogPost
33 ms
blog-site-dev-getBlogPost
19 ms
BlogPostTable
DynamoDB
blog-site-dev-blogPostReplicator
41 ms
/blogpost/_doc/cefa6d70-9e33-11e9-9a01-1d727233d181
ELASTICSEARCH
/dev/blog/search
APIGateway
blog-site-dev-searchBlogPosts
41 ms
/blogpost/_doc/_search
ELASTICSEARCH

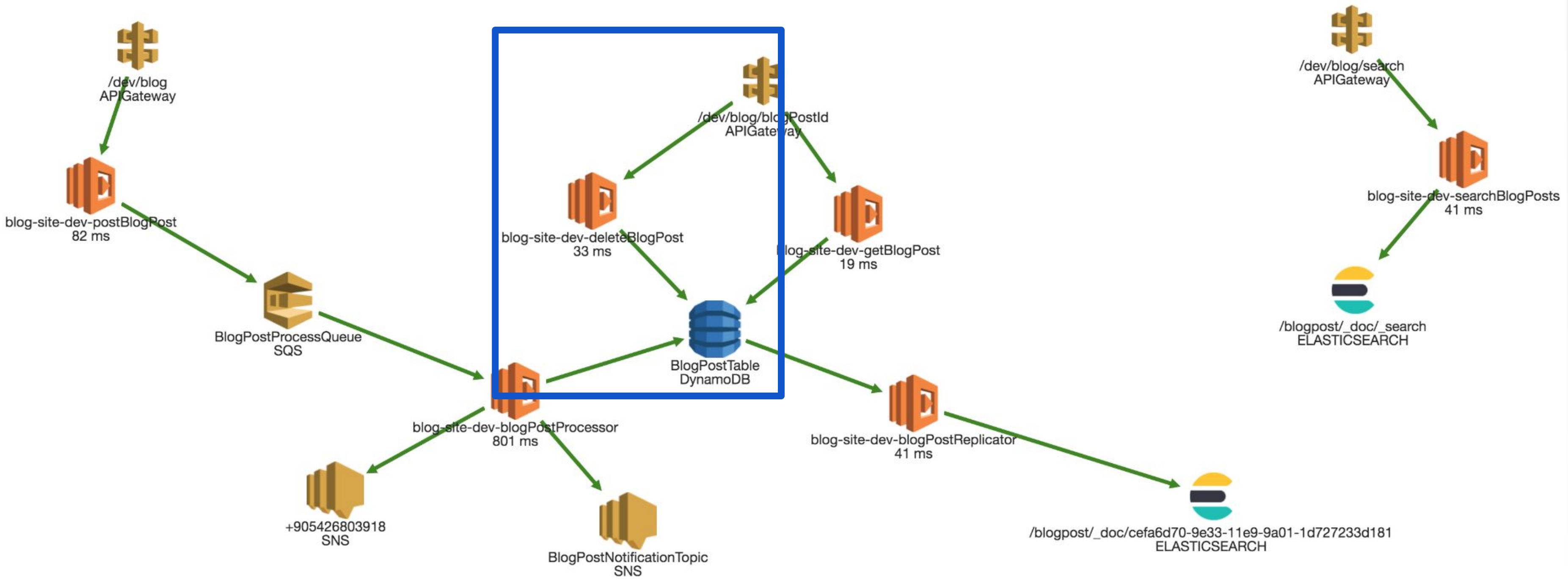
Get Blog Post


/dev/blog
APIGateway
blog-site-dev-postBlogPost
82 ms
BlogPostProcessQueue
SQS
blog-site-dev-blogPostProcessor
801 ms
+905426803918
SNS
BlogPostNotificationTopic
SNS
/dev/blog/blogPostId
APIGateway
blog-site-dev-deleteBlogPost
33 ms
blog-site-dev-getBlogPost
19 ms
BlogPostTable
DynamoDB
blog-site-dev-blogPostReplicator
41 ms
/blogpost/_doc/cefa6d70-9e33-11e9-9a01-1d727233d181
ELASTICSEARCH
/dev/blog/search
APIGateway
blog-site-dev-searchBlogPosts
41 ms
/blogpost/_doc/_search
ELASTICSEARCH

Search Blog Post

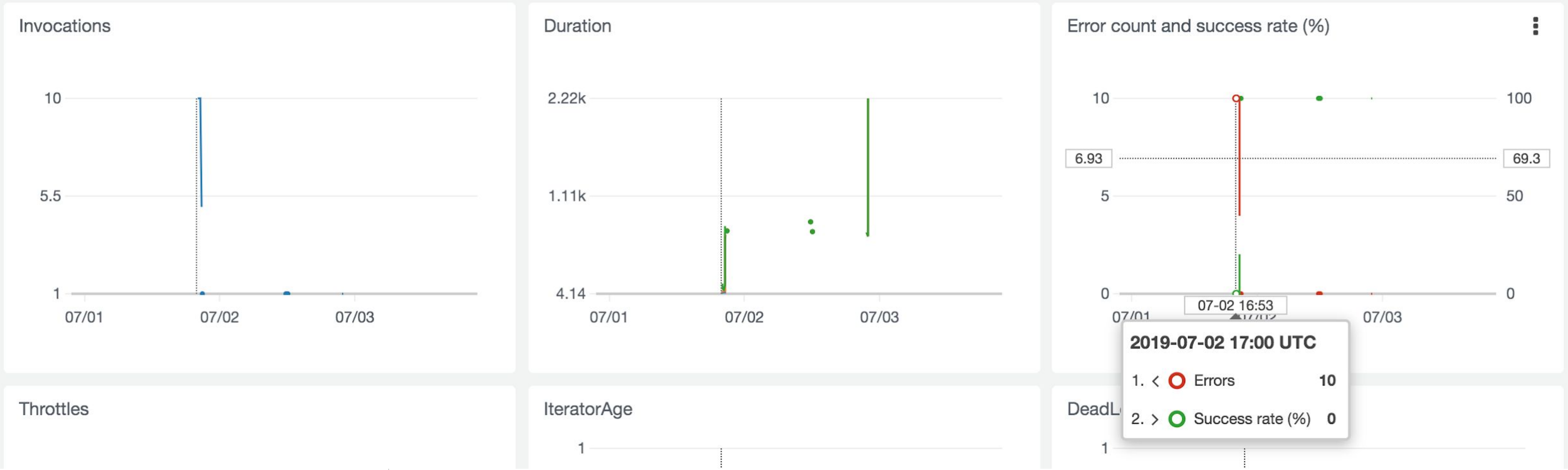

/dev/blog
APIGateway
blog-site-dev-postBlogPost
82 ms
BlogPostProcessQueue
SQS
blog-site-dev-blogPostProcessor
801 ms
+905426803918
SNS
BlogPostNotificationTopic
SNS
/dev/blog/blogPostId
APIGateway
blog-site-dev-deleteBlogPost
33 ms
blog-site-dev-getBlogPost
19 ms
BlogPostTable
DynamoDB
blog-site-dev-blogPostReplicator
41 ms
/blogpost/_doc/cefa6d70-9e33-11e9-9a01-1d727233d181
ELASTICSEARCH
/dev/blog/search
APIGateway
blog-site-dev-searchBlogPosts
41 ms
/blogpost/_doc/_search
ELASTICSEARCH

Delete Blog Post


/dev/blog
APIGateway
blog-site-dev-postBlogPost
82 ms
BlogPostProcessQueue
SQS
/dev/blog/blogPostId
APIGateway
blog-site-dev-deleteBlogPost
33 ms
blog-site-dev-getBlogPost
19 ms
BlogPostTable
DynamoDB
blog-site-dev-blogPostProcessor
801 ms
+905426803918
SNS
BlogPostNotificationTopic
SNS
blog-site-dev-blogPostReplicator
41 ms
/blogpost/_doc/cefa6d70-9e33-11e9-9a01-1d727233d181
ELASTICSEARCH
/dev/blog/search
APIGateway
blog-site-dev-searchBlogPosts
41 ms
/blogpost/_doc/_search
ELASTICSEARCH

MONITORING WITH CLOUDWATCH

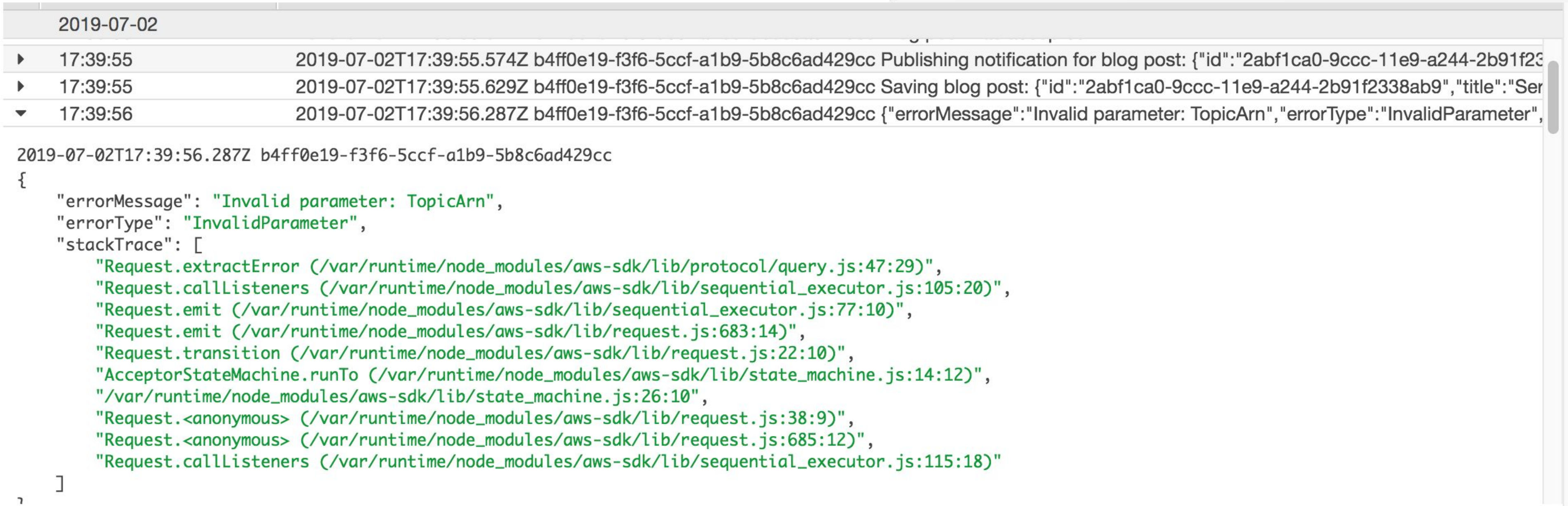
So What?



2019-07-02 17:00 UTC

1. < ○ Errors 10
2. > ○ Success rate (%) 0

Finding Needles in Haystacks



```
2019-07-02
▸ 17:39:55    2019-07-02T17:39:55.574Z b4ff0e19-f3f6-5ccf-a1b9-5b8c6ad429cc Publishing notification for blog post: {"id":"2abf1ca0-9ccc-11e9-a244-2b91f23
▸ 17:39:55    2019-07-02T17:39:55.629Z b4ff0e19-f3f6-5ccf-a1b9-5b8c6ad429cc Saving blog post: {"id":"2abf1ca0-9ccc-11e9-a244-2b91f2338ab9","title":"Ser
▾ 17:39:56    2019-07-02T17:39:56.287Z b4ff0e19-f3f6-5ccf-a1b9-5b8c6ad429cc {"errorMessage":"Invalid parameter: TopicArn","errorType":"InvalidParameter",

2019-07-02T17:39:56.287Z b4ff0e19-f3f6-5ccf-a1b9-5b8c6ad429cc
{
    "errorMessage": "Invalid parameter: TopicArn",
    "errorType": "InvalidParameter",
    "stackTrace": [
        "Request.extractError (/var/runtime/node_modules/aws-sdk/lib/protocol/query.js:47:29)",
        "Request.callListeners (/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:105:20)",
        "Request.emit (/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:77:10)",
        "Request.emit (/var/runtime/node_modules/aws-sdk/lib/request.js:683:14)",
        "Request.transition (/var/runtime/node_modules/aws-sdk/lib/request.js:22:10)",
        "AcceptorStateMachine.runTo (/var/runtime/node_modules/aws-sdk/lib/state_machine.js:14:12)",
        "/var/runtime/node_modules/aws-sdk/lib/state_machine.js:26:10",
        "Request.<anonymous> (/var/runtime/node_modules/aws-sdk/lib/request.js:38:9)",
        "Request.<anonymous> (/var/runtime/node_modules/aws-sdk/lib/request.js:685:12)",
        "Request.callListeners (/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:115:18)"
    ]
```

MONITORING WITH THUNDRA

How to Setup

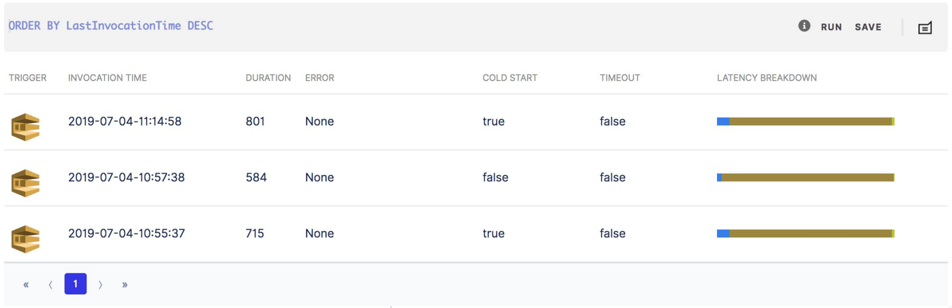
- Add Thundra layer

```
layers:
  - arn:aws:lambda:${self:provider.region}:269863060030:layer:thundra-lambda-node-layer:15
```

- Get and set Thundra API key

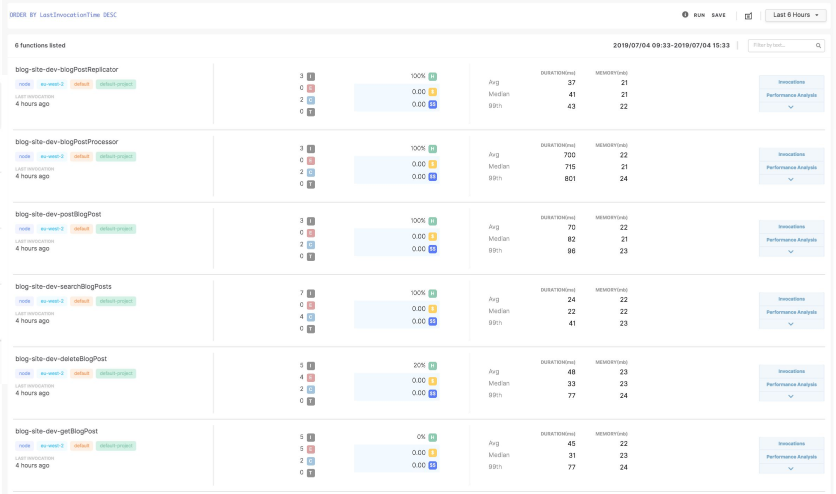
```
environment:
  thundra_apiKey: <YOUR API KEY HERE>
```

- Happy monitoring!!!



ORDER BY LastInvocationTime DESC

TRIGGER	INVOCATION TIME	DURATION	ERROR	COLD START	TIMEOUT	LATENCY BREAKDOWN
	2019-07-04-11:14:58	801	None	true	false	
	2019-07-04-10:57:38	584	None	false	false	
	2019-07-04-10:55:37	715	None	true	false	



ORDER BY LastInvocationTime DESC

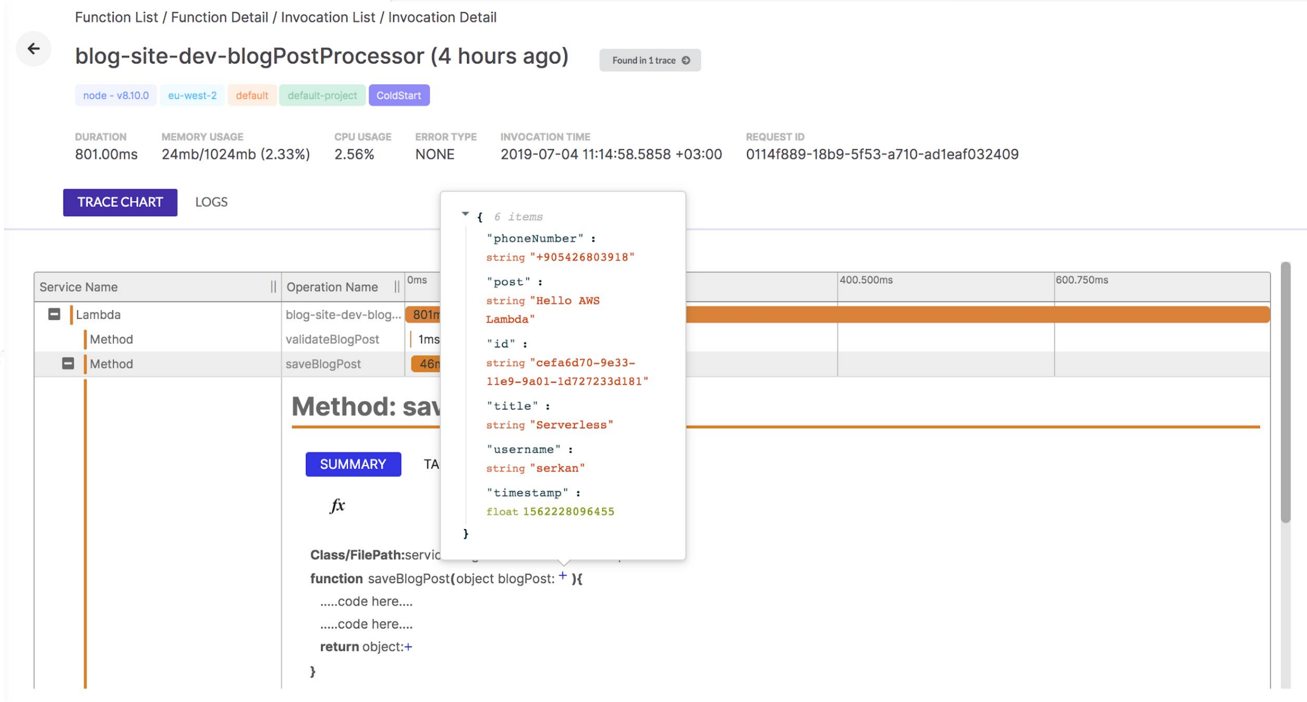
Last 6 Hours

6 functions listed — 2019/07/04 09:33-2019/07/04 15:33

Function	Tags	Last invocation	I	E	C	T	H	$	$$	Duration(ms) Avg	Duration(ms) Median	Duration(ms) 99th	Memory(mb) Avg	Memory(mb) Median	Memory(mb) 99th
blog-site-dev-blogPostReplicator	node, eu-west-2, default, default-project	4 hours ago	3	0	2	0	100%	0.00	0.00	37	41	43	21	21	22
blog-site-dev-blogPostProcessor	node, eu-west-2, default, default-project	4 hours ago	3	0	2	0	100%	0.00	0.00	700	715	801	22	21	24
blog-site-dev-postBlogPost	node, eu-west-2, default, default-project	4 hours ago	3	0	2	0	100%	0.00	0.00	70	82	96	22	21	23
blog-site-dev-searchBlogPosts	node, eu-west-2, default, default-project	4 hours ago	7	0	4	0	100%	0.00	0.00	24	22	41	22	22	23
blog-site-dev-deleteBlogPost	node, eu-west-2, default, default-project	4 hours ago	5	4	2	0	20%	0.00	0.00	48	33	77	23	23	24
blog-site-dev-getBlogPost	node, eu-west-2, default, default-project	4 hours ago	5	5	2	0	0%	0.00	0.00	45	31	77	22	23	24

Local Tracing

```
environment:
  thundra_agent_lambda_trace_instrument_traceableConfig: service.blogPostService.*[traceArgs=true,traceReturnValue=true]
```



Function List / Function Detail / Invocation List / Invocation Detail

blog-site-dev-blogPostProcessor (4 hours ago)

Found in 1 trace

node - v8.10.0 | eu-west-2 | default | default-project | ColdStart

DURATION	MEMORY USAGE	CPU USAGE	ERROR TYPE	INVOCATION TIME	REQUEST ID
801.00ms	24mb/1024mb (2.33%)	2.56%	NONE	2019-07-04 11:14:58.5858 +03:00	0114f889-18b9-5f53-a710-ad1eaf032409

TRACE CHART LOGS

Service Name	Operation Name	0ms	400.500ms	600.750ms
Lambda	blog-site-dev-blog...	801m		
Method	validateBlogPost	1ms		
Method	saveBlogPost	46r		

```
{ 6 items
  "phoneNumber" :
  string "+905426803918"
  "post" :
  string "Hello AWS
  Lambda"
  "id" :
  string "cefa6d70-9e33-
  11e9-9a01-1d727233d181"
  "title" :
  string "Serverless"
  "username" :
  string "serkan"
  "timestamp" :
  float 1562228096455
}
```

Method: sav

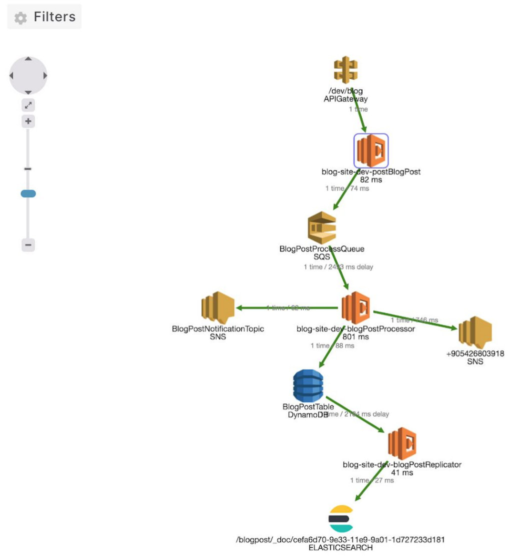
SUMMARY TA

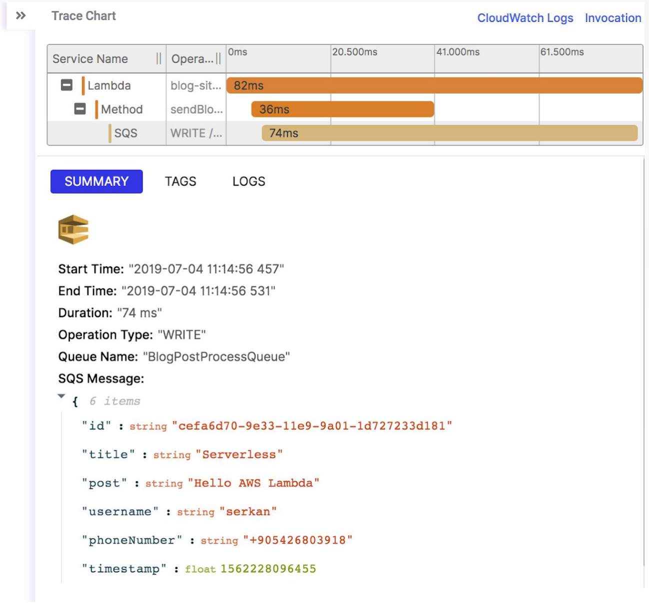
fx

```
Class/FilePath:servic
function saveBlogPost(object blogPost: + ){
  .....code here....
  .....code here....
  return object:+
}
```

Distributed Tracing

- Integrated with all of the most used AWS services
- Supports multiple upstream transaction





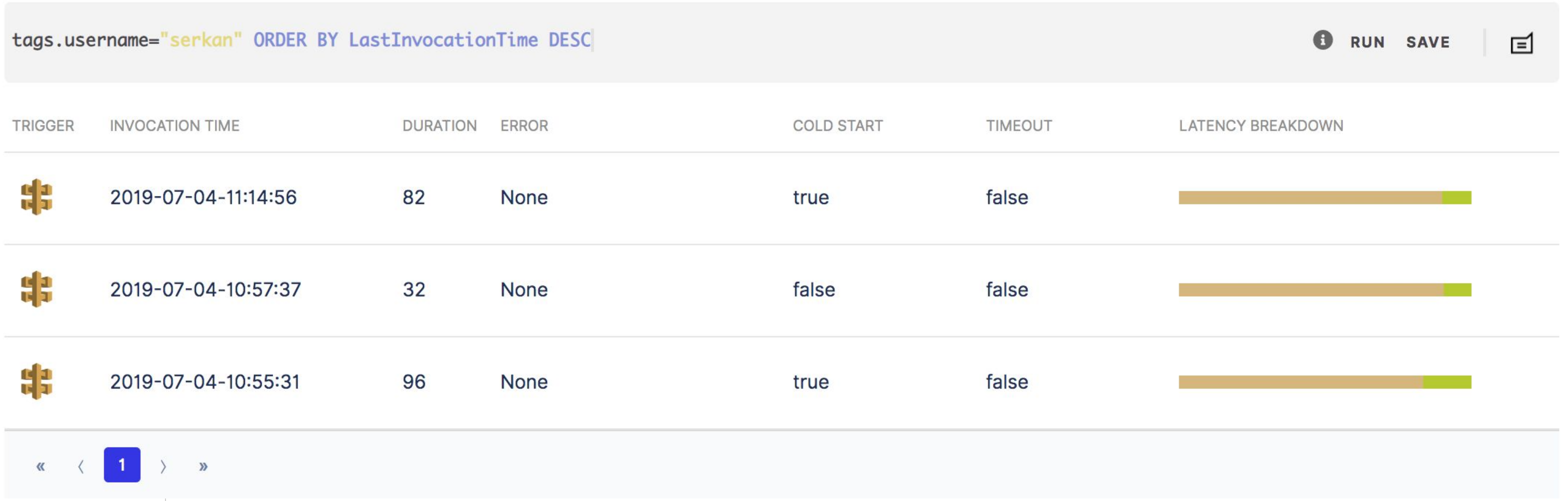
Service Name	Opera...	0ms	20.500ms	41.000ms	61.500ms
Lambda	blog-sit...	82ms			
Method	sendBlo...	36ms			
SQS	WRITE /...	74ms			

Start Time: "2019-07-04 11:14:56 457"
End Time: "2019-07-04 11:14:56 531"
Duration: "74 ms"
Operation Type: "WRITE"
Queue Name: "BlogPostProcessQueue"
SQS Message:

```
{ 6 items
  "id" : string "cefa6d70-9e33-11e9-9a01-1d727233d181"
  "title" : string "Serverless"
  "post" : string "Hello AWS Lambda"
  "username" : string "serkan"
  "phoneNumber" : string "+905426803918"
  "timestamp" : float 1562228096455
```

Invocation Tagging

```
thundra.InvocationSupport.setTag('username', blogPost.username);
```

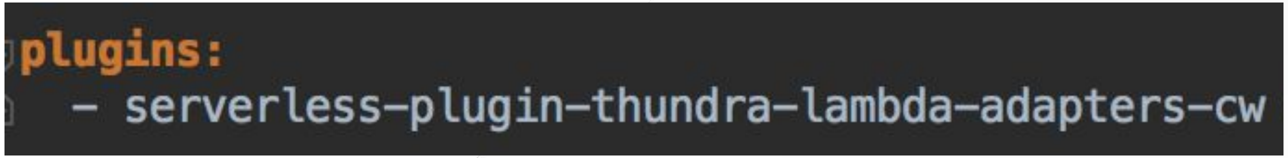


```
tags.username="serkan" ORDER BY LastInvocationTime DESC
```

TRIGGER	INVOCATION TIME	DURATION	ERROR	COLD START	TIMEOUT	LATENCY BREAKDOWN
	2019-07-04-11:14:56	82	None	true	false	
	2019-07-04-10:57:37	32	None	false	false	
	2019-07-04-10:55:31	96	None	true	false	

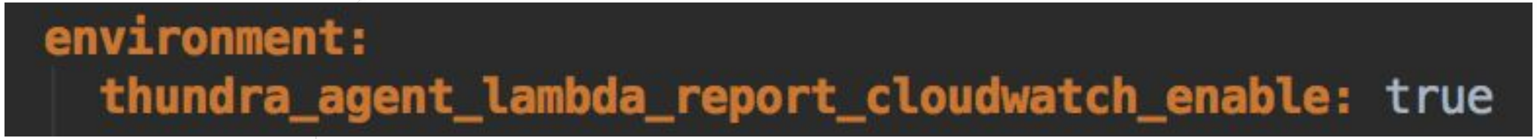
Async monitoring

- No network delay
- Even works in VPC
- Add Thundra Serverless CloudWatch plugin



```
plugins:
  - serverless-plugin-thundra-lambda-adapters-cw
```

- Enable CloudWatch based reporting



```
environment:
  thundra_agent_lambda_report_cloudwatch_enable: true
```

MORE THUNDRA FEATURES

OpenTracing API/Spec Compatible

- OpenTracing compatible API
- OpenTracing specification compatible data model
- Easy to integrate with other APM solutions
  - Honeycomb
- github.com/opentracing/specification/blob/master/specification.md

Intelligent Sampling

- Samples periodically at every
  - N invocation
  - N seconds/minutes
- Samples when
  - duration exceeds given threshold
  - invocation fails with any (or specific) error
  - CPU/Memory usage exceeds threshold
- Custom: Implement your own sampler

Alerting

- Highly customizable alerts to create your own way of keeping eye on your system.
- Actionable insights to reduce the MTTR
- Customizable notification rules to your preferred channel and channels
- Different severity levels to prevent the alert fatigue in your organization

Your Data at Your Instance

- Thundra never see your data
- Splunk
  - stored at your Splunk deployment
  - Thundra provides Splunk App to visualize your data
  - or run your own queries
- Elasticsearch
  - stored at your Elasticsearch instance/clıster
  - Thundra provides Kibana plugin to visualize your data
  - or run your own queries

THUNDRA

Thank you !