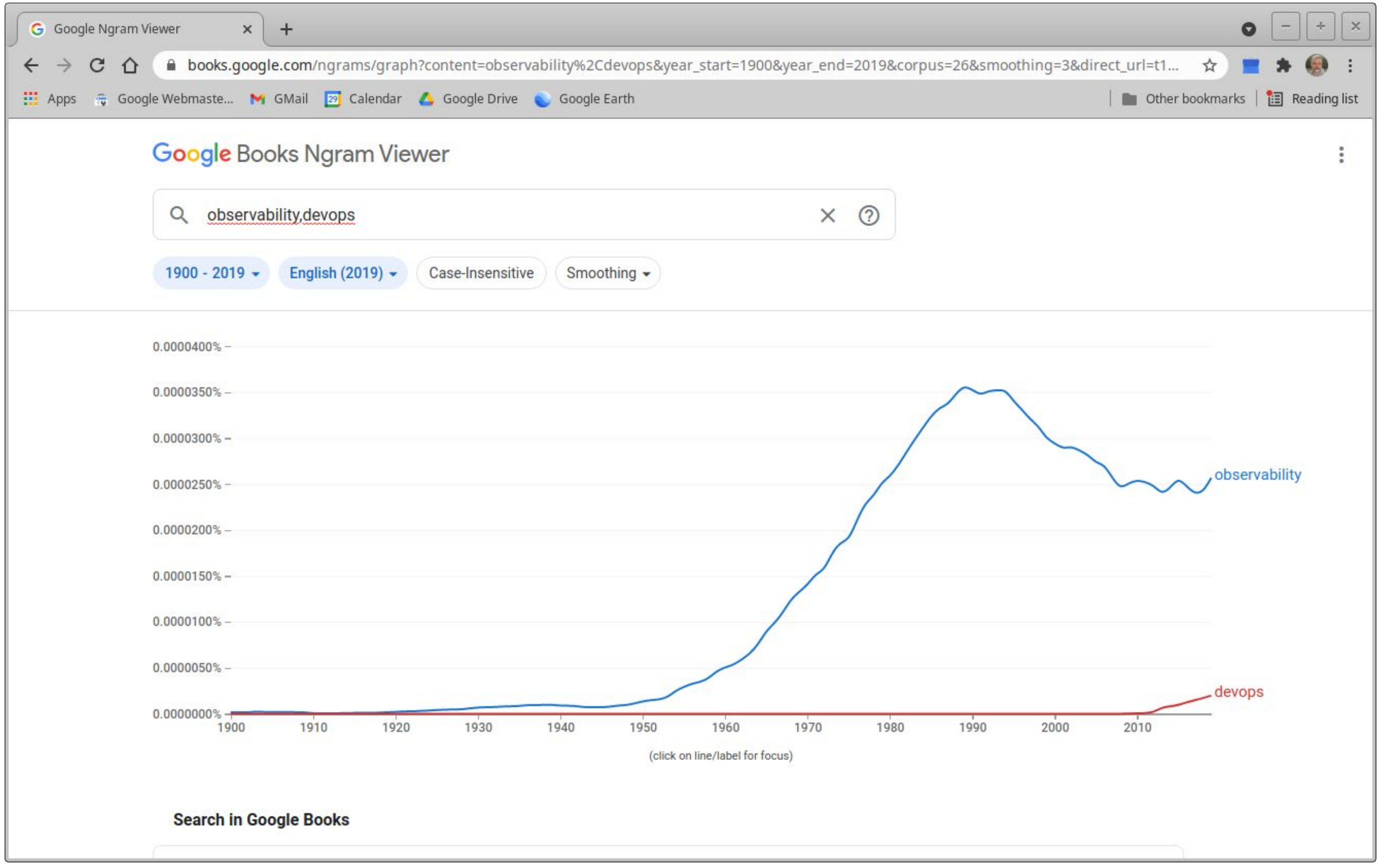
Observability and You

It's 10 PM, do you know what your code is doing?

Keith Gregory
AWS Practice Lead, Chariot Solutions

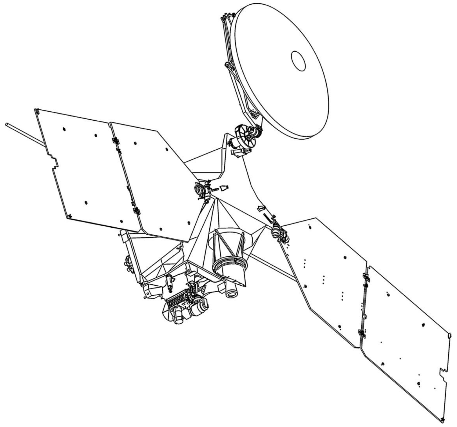
CHARIOT SOLUTIONS

You Keep Using That Word…

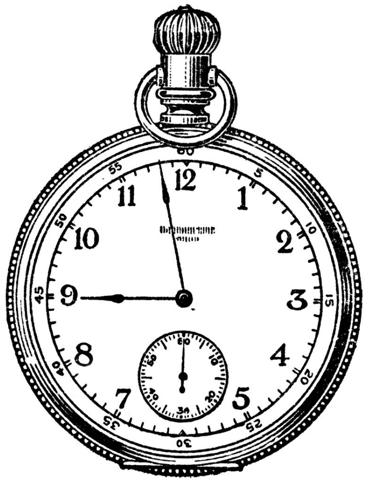


The Observability Triad

Monitoring



Tracing



Logging

Observability Is Like Exercise

You know you should do it, but the couch is comfy and there might be something good on TV!

Its effects are seen over the long term

Eventually, it becomes a habit

The Goals

Avoid surprises

Fix problems quickly when they happen

Monitoring

The current and recent health of your system

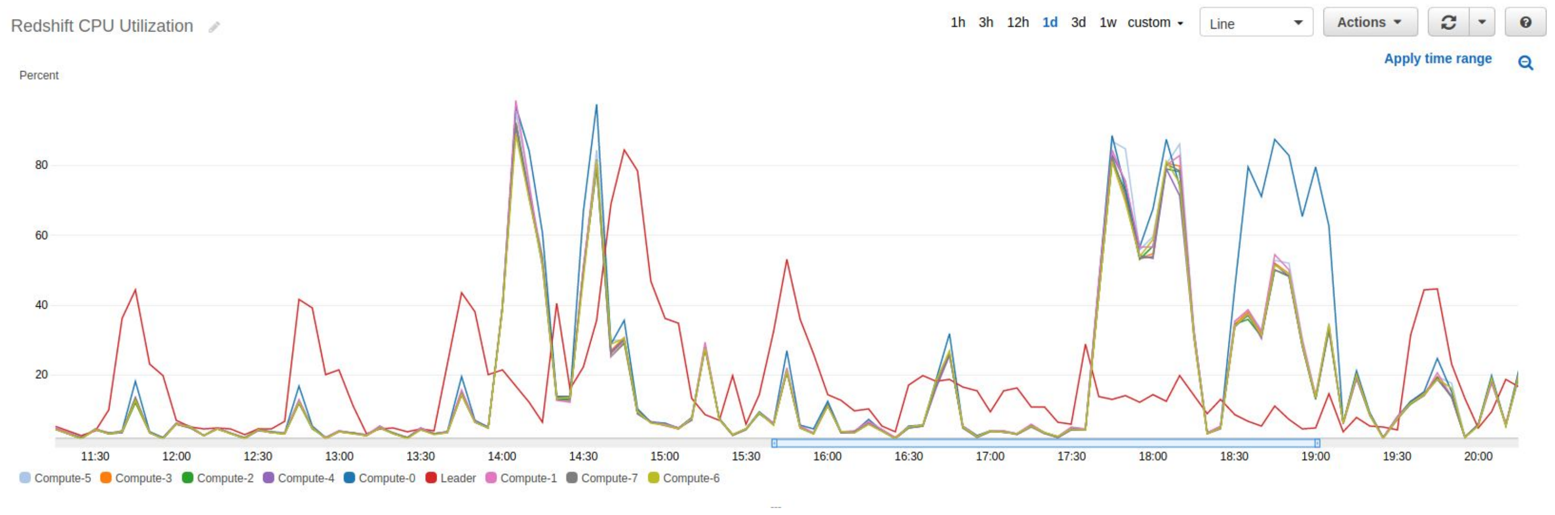
It’s About The Numbers (metrics)

Each metric reports a number

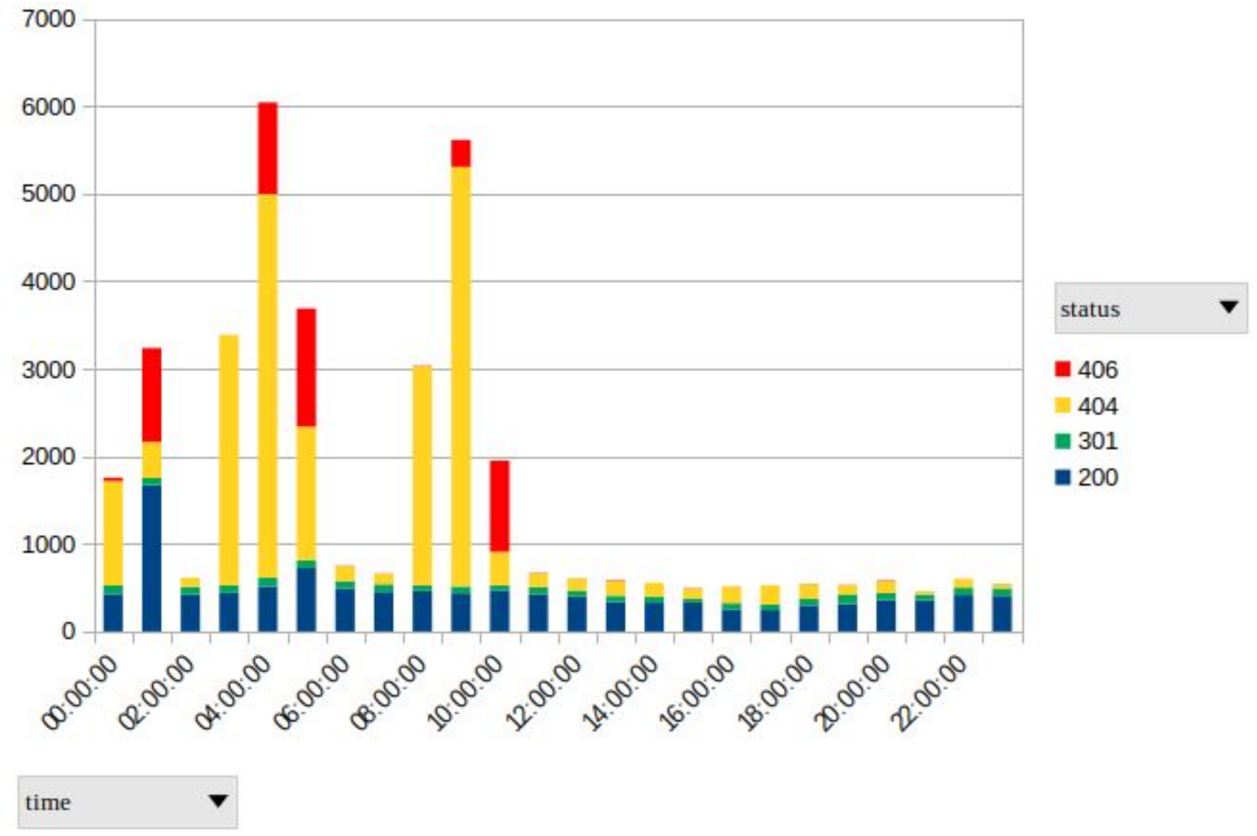
The monitoring system tracks those numbers over time

“Dimensions” allow metrics to be sliced and diced

Example: Redshift CPU Utilization



One Metric, Multiple Dimensions



CHARIOT SOLUTIONS

“The Four Golden Signals”

Latency	*How long do requests take?*
Traffic	*How much work is coming in?*
Saturation	*How close are you to system limits?*
Errors	*What percent of requests fail?*

Values Versus Buckets

(Percentile) Buckets

- Reduces noise: outliers don’t change average, don’t get lost
- Aggregation across collection periods is potentially invalid

Discrete values are useful for “right now” action

- Presented as sum or min/max, *rarely* average

Outliers Are Important!

- Every outlier should prompt you to ask “why?”
  - Why is one node handling more requests than the others?
  - Should there be so many login failures?
- Missing data is the most important outlier
  - It usually means that part of your system is down
  - Or unable to report what it’s doing

Alerting

Identify anomalous behavior and wake someone up

Recognize trends and scale your infrastructure

Tracing

Helping you find bottlenecks

CHARIOT SOLUTIONS

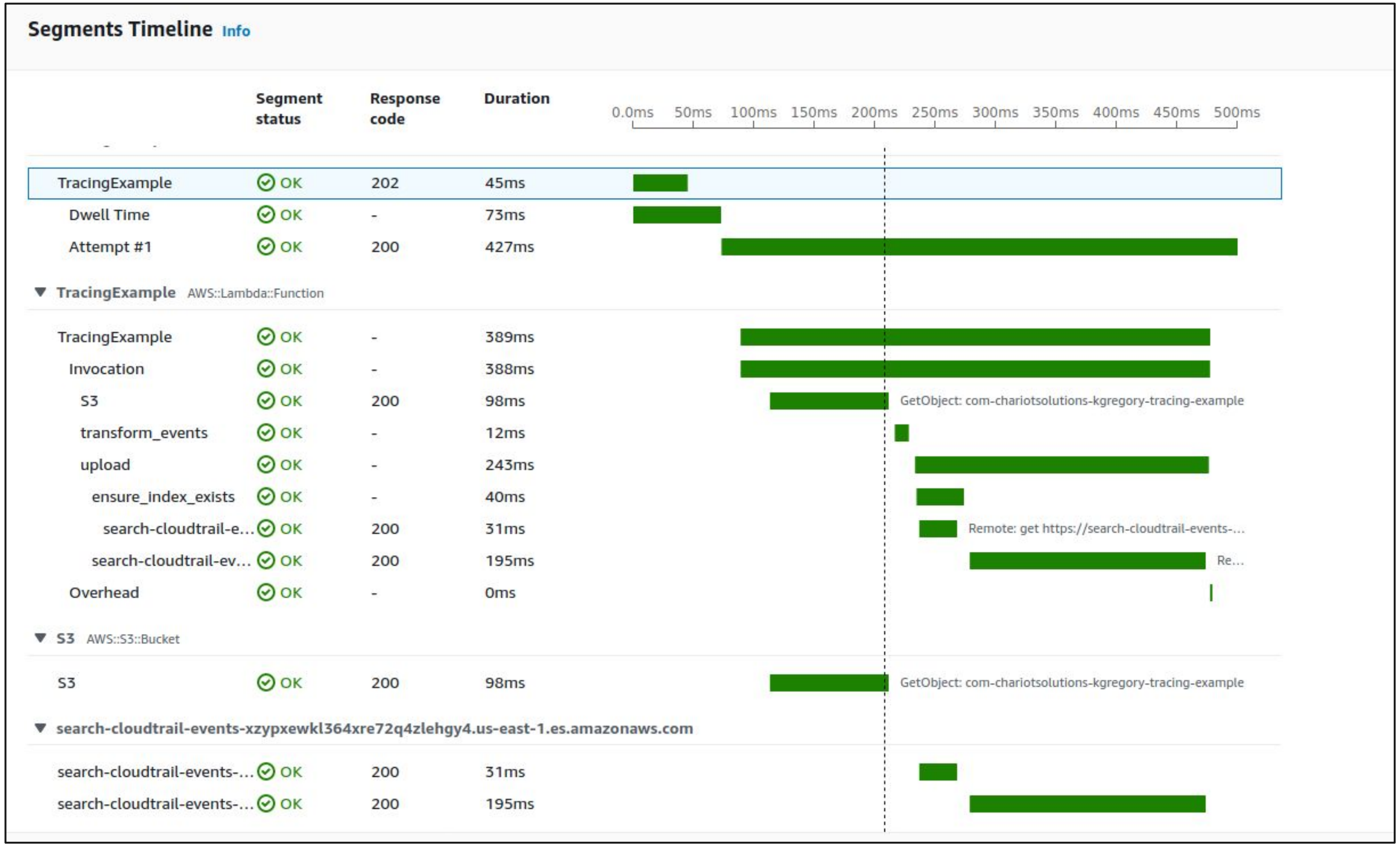
How It Works

Uniquely identify Requests and Spans

Runtime reports start/end timestamps for each span

Aggregator figures out how long everything took

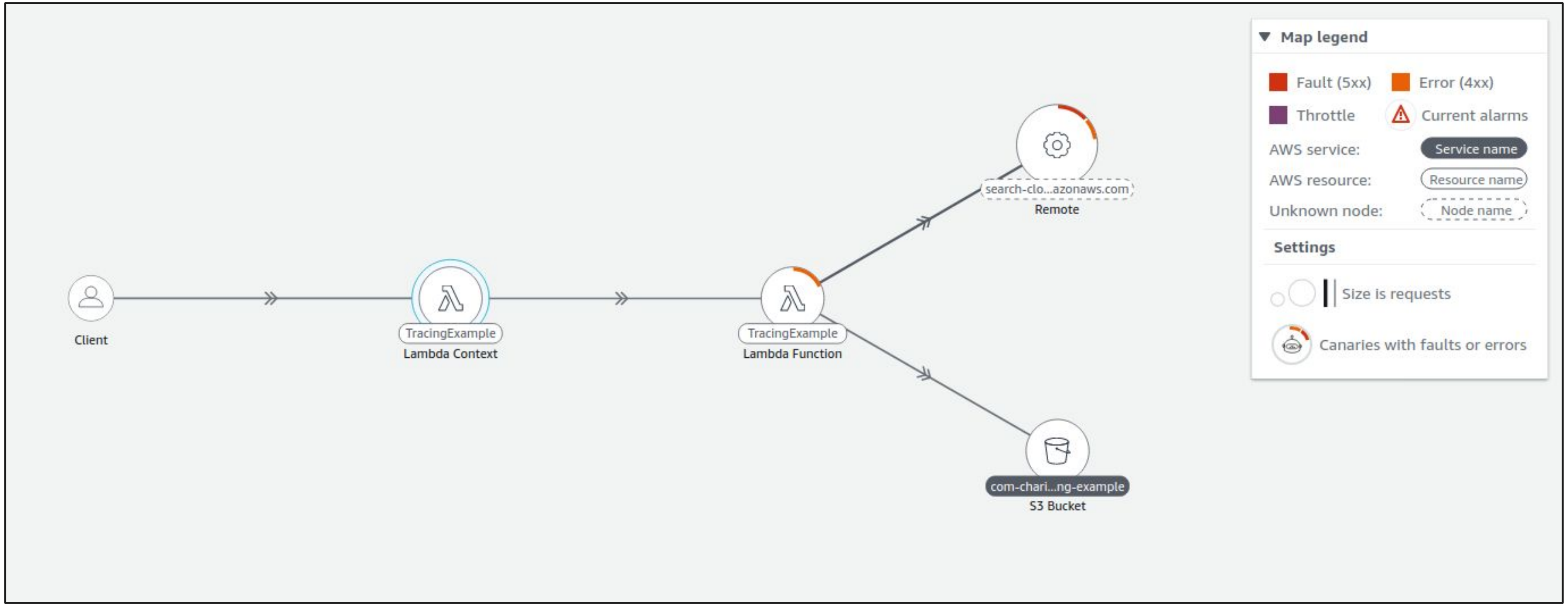
Execution-time Breakdown



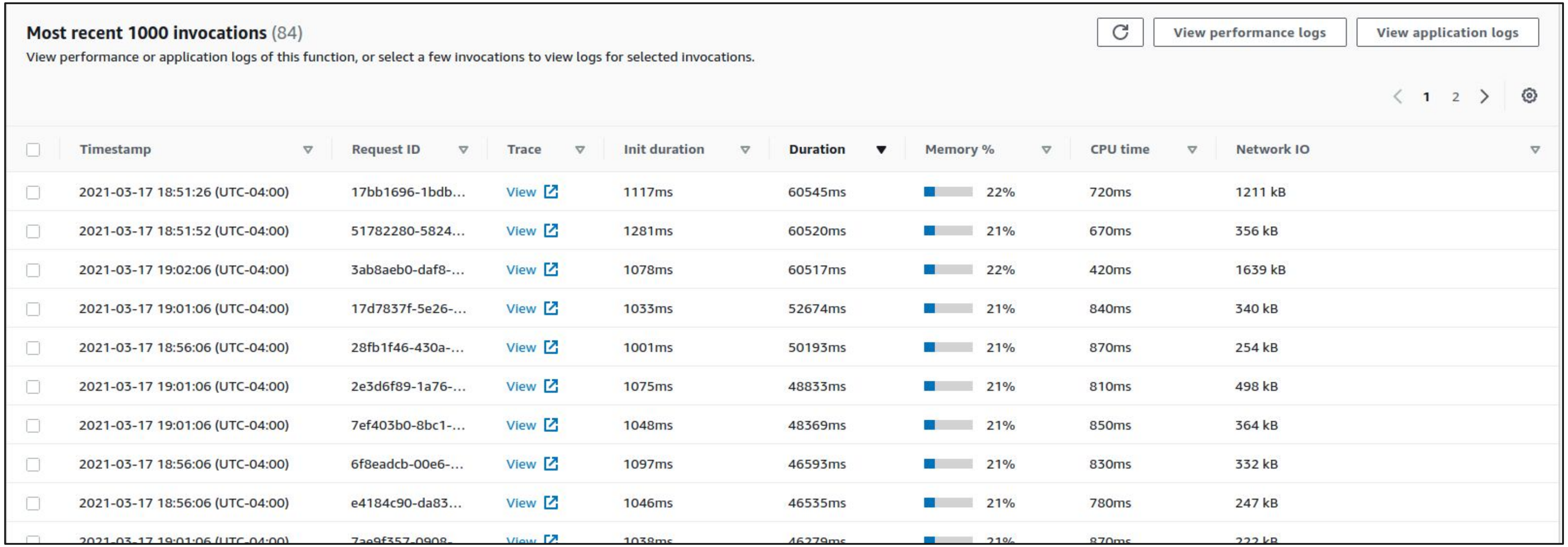
Segments Timeline Info

	Segment status	Response code	Duration	
TracingExample	OK	202	45ms	
Dwell Time	OK	-	73ms	
Attempt #1	OK	200	427ms	
▼ TracingExample AWS::Lambda::Function				
TracingExample	OK	-	389ms	
Invocation	OK	-	388ms	
S3	OK	200	98ms	GetObject: com-chariotsolutions-kgregory-tracing-example
transform_events	OK	-	12ms	
upload	OK	-	243ms	
ensure_index_exists	OK	-	40ms	
search-cloudtrail-e…	OK	200	31ms	Remote: get https://search-cloudtrail-events-…
search-cloudtrail-ev…	OK	200	195ms	Re…
Overhead	OK	-	0ms	
▼ S3 AWS::S3::Bucket				
S3	OK	200	98ms	GetObject: com-chariotsolutions-kgregory-tracing-example
▼ search-cloudtrail-events-xzypxewkl364xre72q4zlehgy4.us-east-1.es.amazonaws.com				
search-cloudtrail-events-…	OK	200	31ms	
search-cloudtrail-events-…	OK	200	195ms	

Service Map


Client
TracingExample
Lambda Context
TracingExample
Lambda Function
search-clo...azonaws.com
Remote
com-chari...ng-example
S3 Bucket
Map legend
Fault (5xx)
Error (4xx)
Throttle
Current alarms
AWS service:
Service name
AWS resource:
Resource name
Unknown node:
Node name
Settings
Size is requests
Canaries with faults or errors

Outliers Are Important!



Most recent 1000 invocations (84)

View performance or application logs of this function, or select a few invocations to view logs for selected invocations.

Timestamp	Request ID	Trace	Init duration	Duration	Memory %	CPU time	Network IO
2021-03-17 18:51:26 (UTC-04:00)	17bb1696-1bdb...	View	1117ms	60545ms	22%	720ms	1211 kB
2021-03-17 18:51:52 (UTC-04:00)	51782280-5824...	View	1281ms	60520ms	21%	670ms	356 kB
2021-03-17 19:02:06 (UTC-04:00)	3ab8aeb0-daf8-...	View	1078ms	60517ms	22%	420ms	1639 kB
2021-03-17 19:01:06 (UTC-04:00)	17d7837f-5e26-...	View	1033ms	52674ms	21%	840ms	340 kB
2021-03-17 18:56:06 (UTC-04:00)	28fb1f46-430a-...	View	1001ms	50193ms	21%	870ms	254 kB
2021-03-17 19:01:06 (UTC-04:00)	2e3d6f89-1a76-...	View	1075ms	48833ms	21%	810ms	498 kB
2021-03-17 19:01:06 (UTC-04:00)	7ef403b0-8bc1-...	View	1048ms	48369ms	21%	850ms	364 kB
2021-03-17 18:56:06 (UTC-04:00)	6f8eadcb-00e6-...	View	1097ms	46593ms	21%	830ms	332 kB
2021-03-17 18:56:06 (UTC-04:00)	e4184c90-da83...	View	1046ms	46535ms	21%	780ms	247 kB
2021-03-17 19:01:06 (UTC-04:00)	7ae9f357-0908-...	View	1038ms	46279ms	21%	870ms	222 kB

Should you enable in Production?

Depending on implementation, may add overhead

… But production is where you see your real workload

Answer: YES!

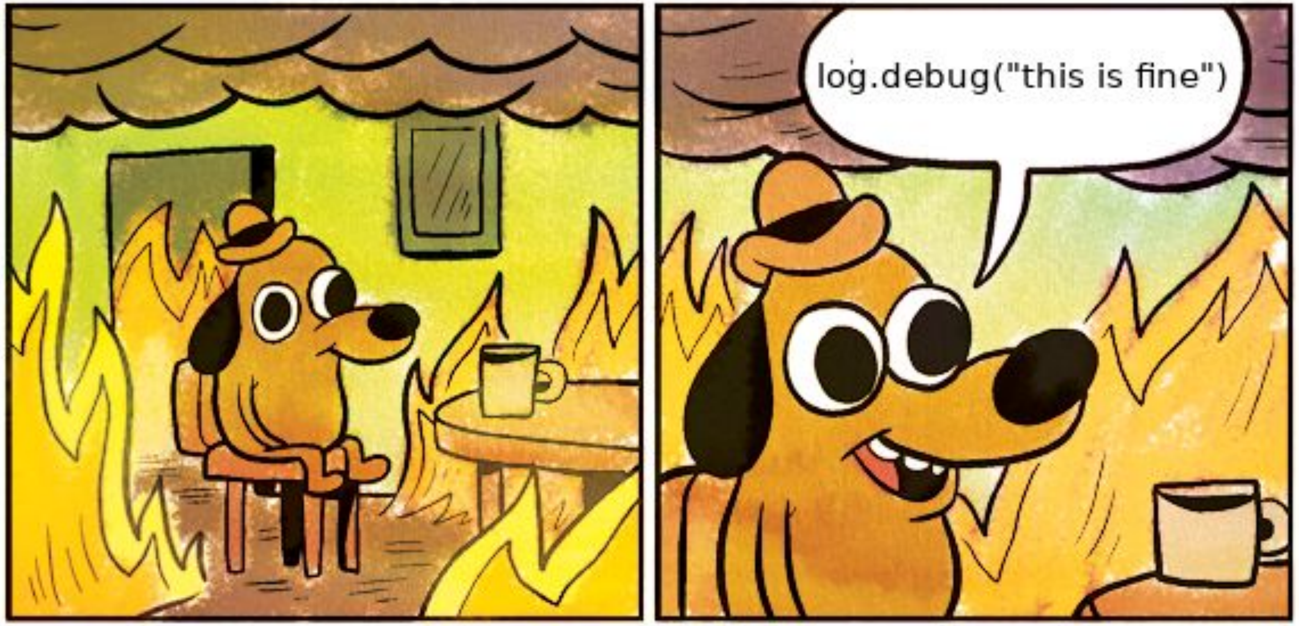
Logging

How you debug production problems at 3 AM

Effective Logs Must …

- Convey appropriate urgency
  - Errors should wake people up
  - Debug messages should help developers solve problems
- Provide enough information
  - What is the current state of my program?
  - What is it about to do, and why?
- Without being overwhelming

This Is Not Effective



adapted with permission from
http://gunshowcomic.com/648

Structured Logging

Logs formatted for searching, not reading

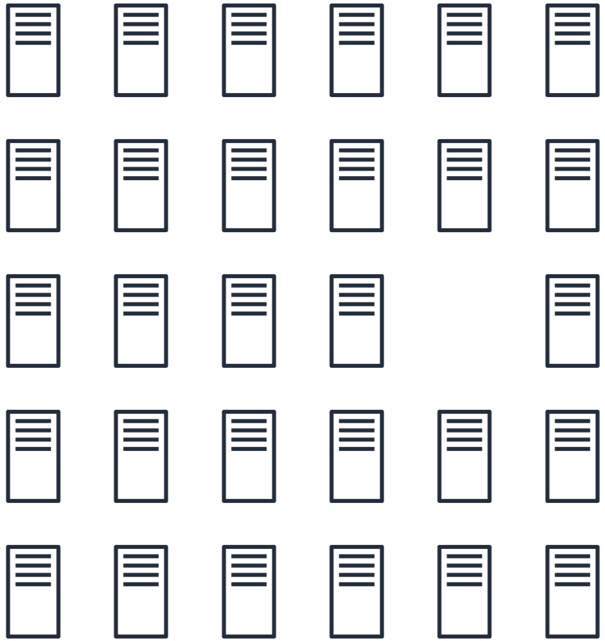
JSON is the perfect structured log format

- It’s well-defined
- It’s extensible
- It’s the format expected by popular search engines

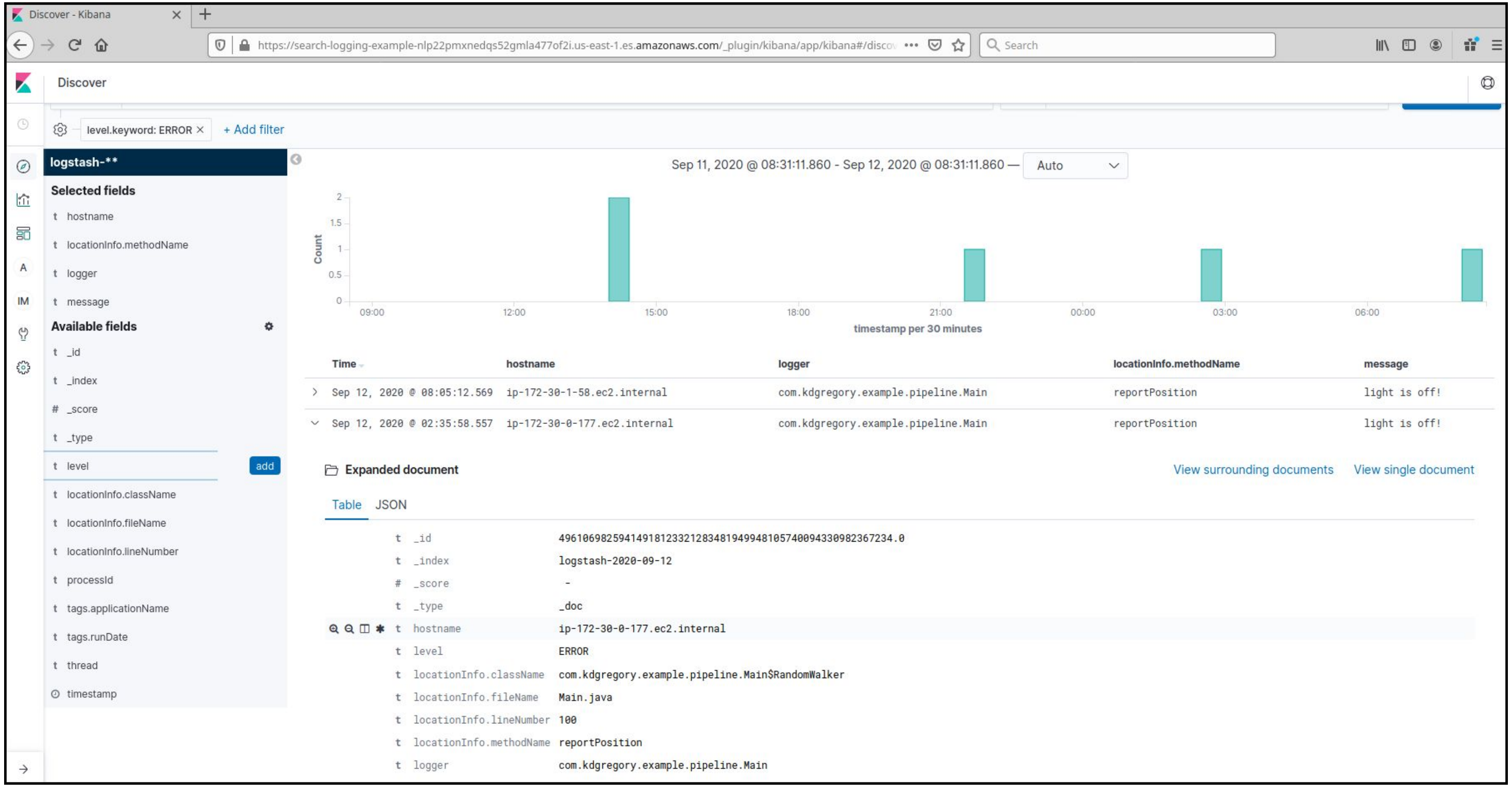
Centralized Logging

Every node has its own log

In the cloud, machines disappear without a trace



Search engines FTW



Discover - Kibana
https://search-logging-example-nlp22pmxnedqs52gmla477of2i.us-east-1.es.amazonaws.com/_plugin/kibana/app/kibana#/disco
Search
Discover
level.keyword: ERROR
+ Add filter
logstash-**
Sep 11, 2020 @ 08:31:11.860 - Sep 12, 2020 @ 08:31:11.860 —
Auto
Count
timestamp per 30 minutes
Expanded document
View surrounding documents
View single document
Table
JSON
add

Selected fields

- t hostname
- t locationInfo.methodName
- t logger
- t message

Available fields

- t _id
- t _index
- # _score
- t _type
- t level
- t locationInfo.className
- t locationInfo.fileName
- t locationInfo.lineNumber
- t processId
- t tags.applicationName
- t tags.runDate
- t thread
- timestamp

Time	hostname	logger	locationInfo.methodName	message
Sep 12, 2020 @ 08:05:12.569	ip-172-30-1-58.ec2.internal	com.kdgregory.example.pipeline.Main	reportPosition	light is off!
Sep 12, 2020 @ 02:35:58.557	ip-172-30-0-177.ec2.internal	com.kdgregory.example.pipeline.Main	reportPosition	light is off!

Expanded document

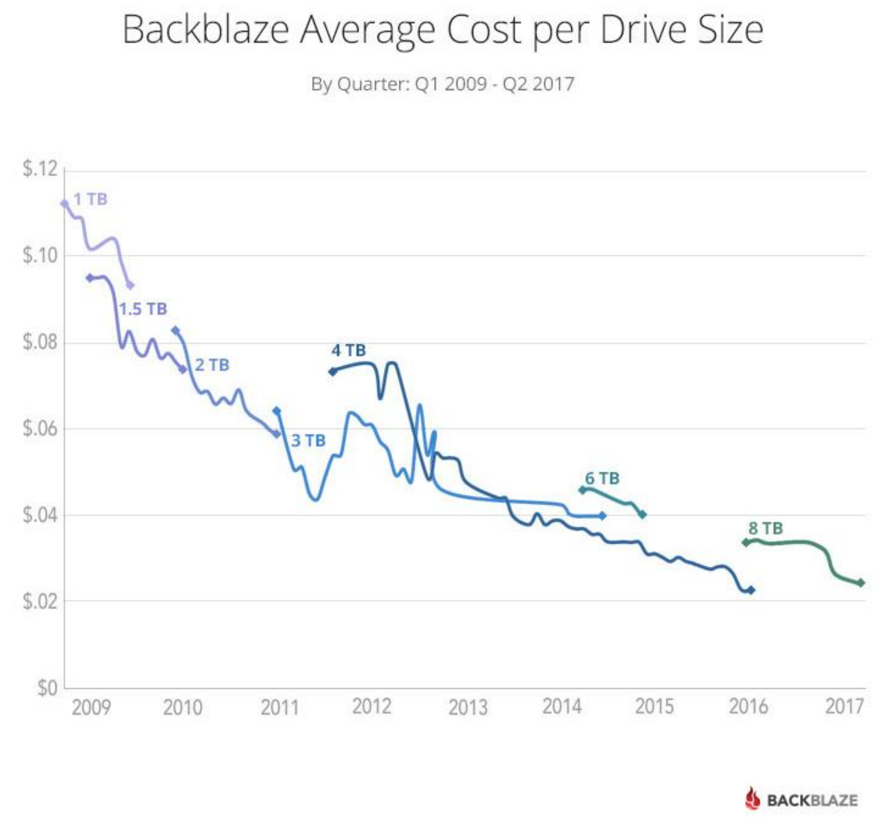
t	_id	49610698259414918123321283481949948105740094330982367234.0
t	_index	logstash-2020-09-12
#	_score	-
t	_type	_doc
t	hostname	ip-172-30-0-177.ec2.internal
t	level	ERROR
t	locationInfo.className	com.kdgregory.example.pipeline.Main$RandomWalker
t	locationInfo.fileName	Main.java
t	locationInfo.lineNumber	100
t	locationInfo.methodName	reportPosition
t	logger	com.kdgregory.example.pipeline.Main

Observability Culture

It’s not enough to say “let’s do this!”

Observe ALL The Things!

Don’t Let Storage Hold You Back



https://www.backblaze.com/blog/wp-content/uploads/2017/07/chart-cost-per-drive-2017.jpg

Always Provide Value

It’s easy to report the wrong things

Applications that “cry wolf” are ignored

Make Dashboards **Visible**

Show only the information that is actionable

Raspberry Pi + Big Screen TV = Instant NOC

“Observability Infected”

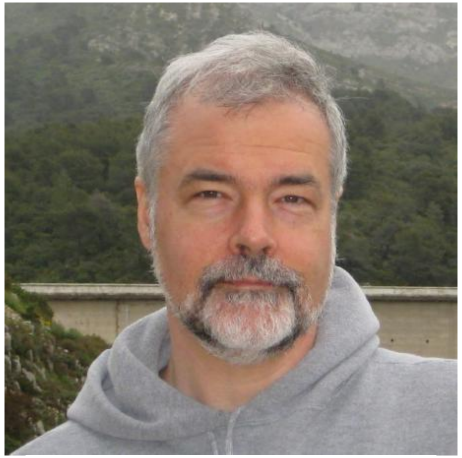
One bad experience with *un*-observability

After debugging, leave your logging in-place (like writing a test to isolate a bug)

Encourage teams to ask questions about their app

Questions?

About Me



AWS Practice Lead at Chariot Solutions

Programming since 1977,
professionally since 1984,
on AWS since 2008

https://www.kdgregory.com/

https://github.com/kdgregory/

@ChariotKGregory

Technology in the Service of Business.

Chariot Solutions is the Greater Philadelphia region’s top IT consulting firm specializing in software development, systems integration, mobile application development and training.

Our team includes many of the top software architects in the area, with deep technical expertise, industry knowledge and a genuine passion for software development.

Visit us online at chariotsolutions.com.