Effective Data Analytics for Modern Applications

Christian Beedgen, CTO, Sumo Logic

Ben Newton, Principal Product Manager, Sumo Logic

Ben Abrams, Lead DevOps Engineer, Cloud Cruiser

November 2016

What to Expect from the Session

- Drivers for Data & Modern Applications
- Designing a Data Analytics Strategy
- Case Study: Cloud Cruiser
- Q&A

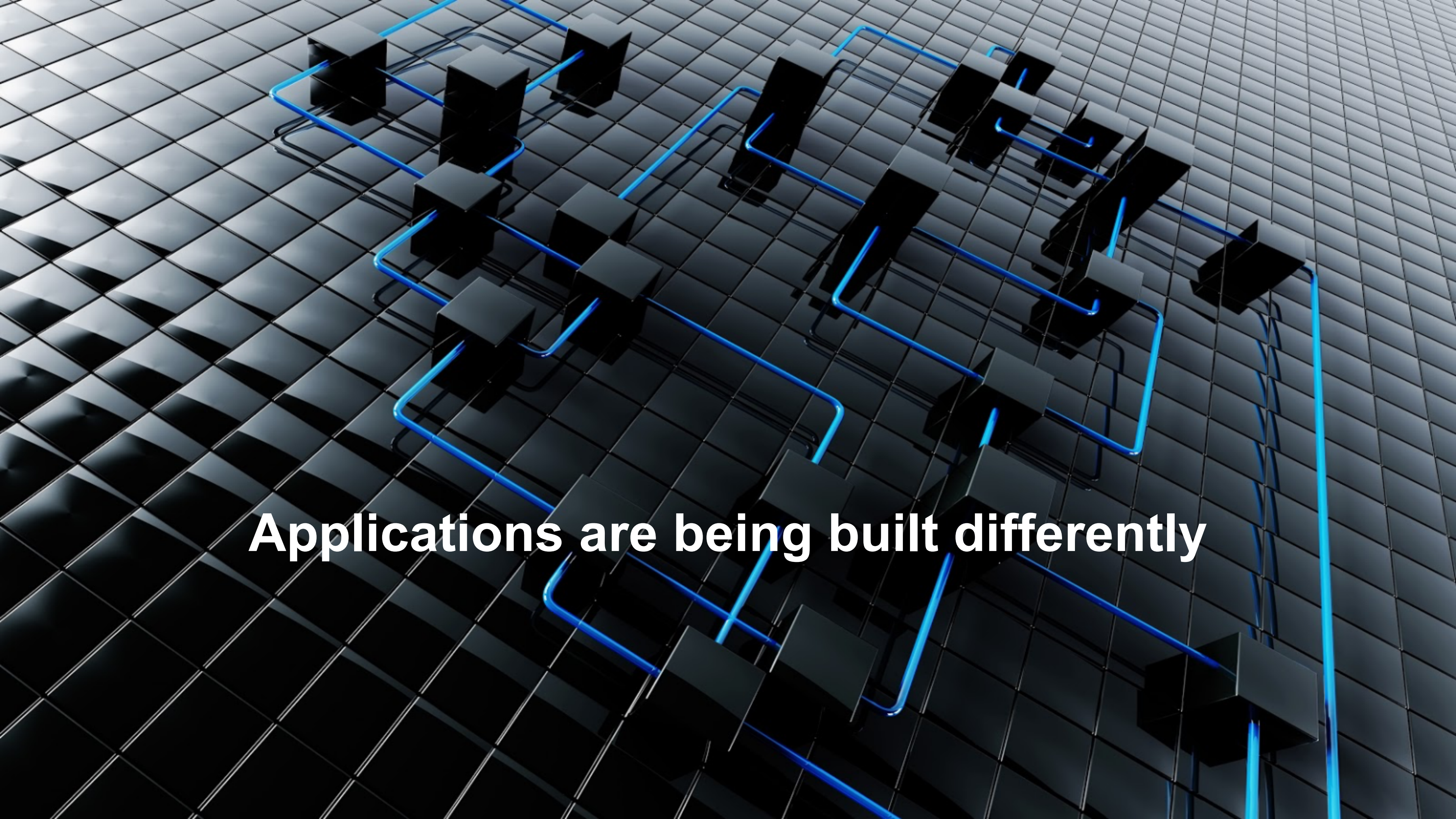
Digital Transformation is Disrupting Every Industry

“Software is eating the world.”

Marc Andreessen

“Every industry that is not bringing software to their business will be disrupted.”

Key Drivers: Customer Experience, Differentiation & Agility



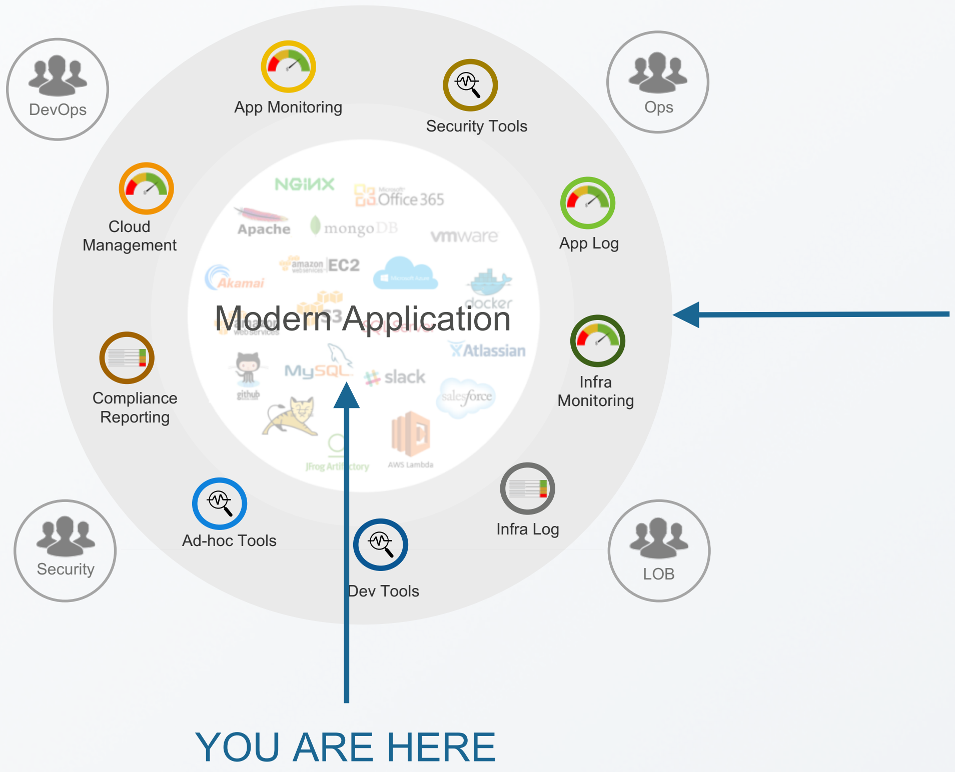
Applications are being built differently

Teams are changing too

Agility can lead to complexity

You can’t fix what you can’t see

With Great Power Comes Great Responsibility



- CEO / Board / Shareholders
- Customers
- Partners
- Customer Success
- CSO / VP Security
- Product Management

Designing a Data Analytics Strategy

Continuous Intelligence

Insights Across Modern Application Lifecycle

Build

Accelerate development and release cycles from code to delivery

Run

Improve performance and reliability through full stack visibility

Secure

Ensure the security and compliance of applications and infrastructure

Continuous Intelligence

Insights Across Modern Application Lifecycle

Build

Accelerate development and release cycles from code to delivery

Run

Improve performance and reliability through full stack visibility

Secure

Ensure the security and compliance of applications and infrastructure

Run

Improve performance and reliability through full stack visibility

What Type of Activities Does Your Data Support?

Monitoring

Detect → Notify

Troubleshooting

Identify → Diagnose → Restore → Resolve

App Intelligence

Understand → Improve → Report

Who Cares About this Data?

Monitoring

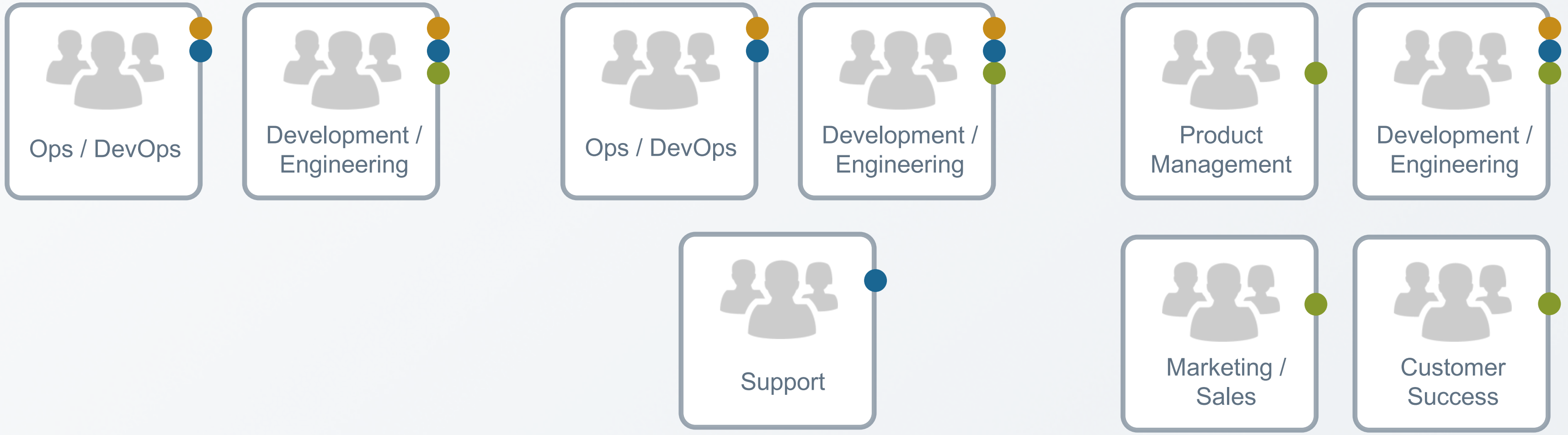
Focus on **User Visible** Functionality

Troubleshooting

Focus on **End-to-End** Visibility

App Intelligence

Focus on **User Activity** & **Visibility**



Stakeholders **Stakeholders** **Stakeholders**

Use Data to Solve Real Problems

Monitoring

Focus on **User Visible** Functionality

What's important to your business? Can you measure it?

Measure and monitor **user visible** metrics

Build **fewer, higher impact, real-time monitors**

Troubleshooting

Focus on **End-to-End** Visibility

You **can't fix** what you **can't measure**

Comprehensive metrics coverage is essential

Correlate metrics with **logs** to reduce resolution time

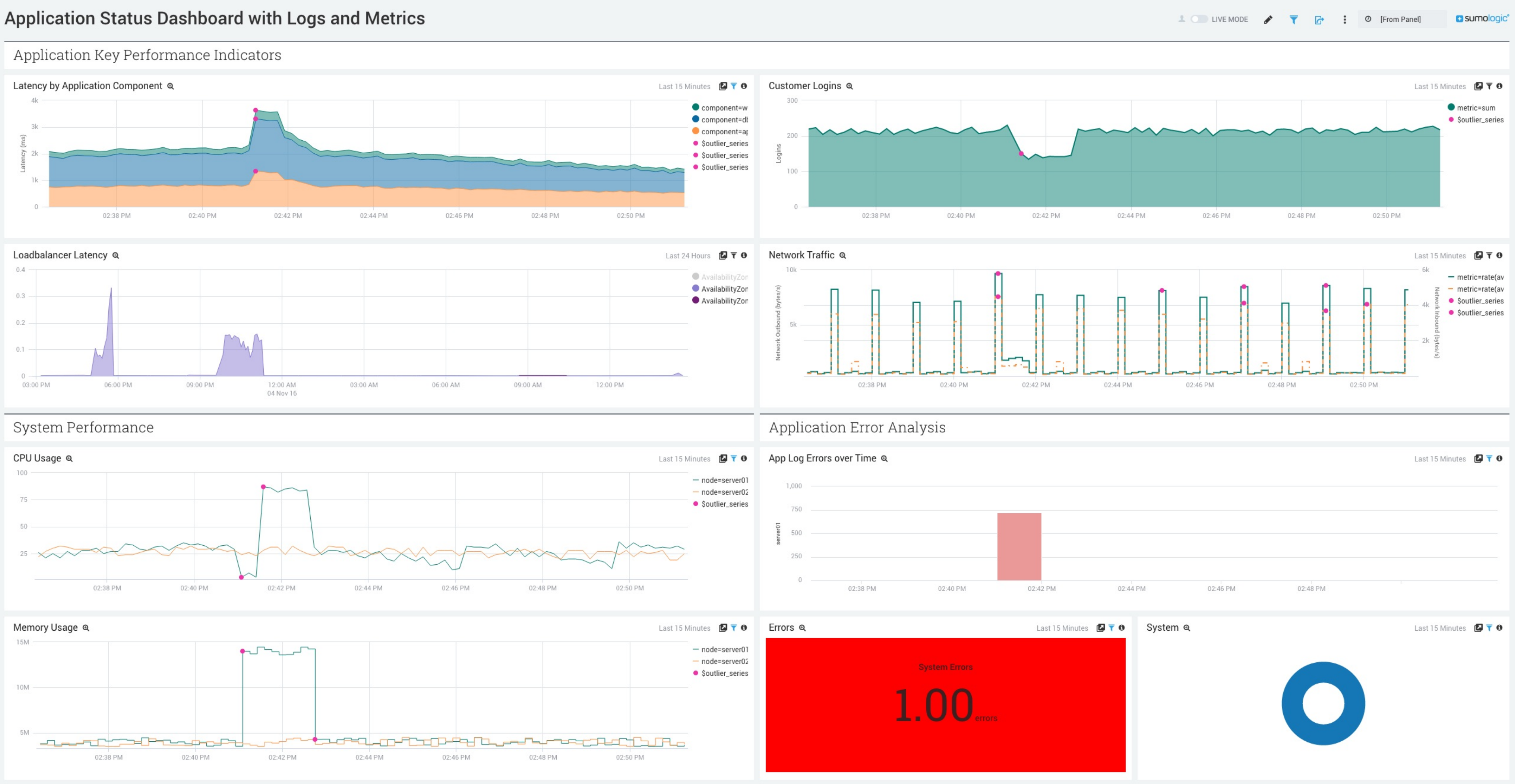
App Intelligence

Focus on **User Activity** & **Visibility**

You **can't improve** what you **can't measure**

You need both **activity metrics** and **detailed logs**

Up to date data drives better **data-driven decisions**


Application Status Dashboard with Logs and Metrics
LIVE MODE
[From Panel]
sumologic
Application Key Performance Indicators
Latency by Application Component
Last 15 Minutes
Latency (ms)
component=w
component=dl
component=ap
$outlier_series
Customer Logins
Logins
metric=sum
$outlier_series
Loadbalancer Latency
Last 24 Hours
AvailabilityZor
03:00 PM
06:00 PM
09:00 PM
12:00 AM 04 Nov 16
03:00 AM
06:00 AM
09:00 AM
12:00 PM
Network Traffic
Network Outbound (bytes/s)
Network Inbound (bytes/s)
metric=rate(av
$outlier_series
System Performance
CPU Usage
node=server01
node=server02
Application Error Analysis
App Log Errors over Time
server01
Memory Usage
Errors
System Errors
1.00 errors
System
02:38 PM
02:40 PM
02:42 PM
02:44 PM
02:46 PM
02:48 PM
02:50 PM

Data From all Parts of Your Stack

Infrastructure

AWS ECS

docker

Microsoft Windows

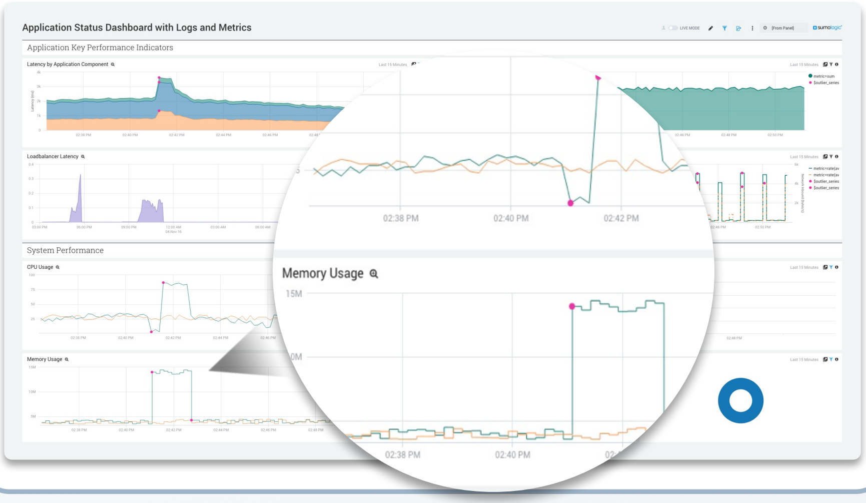
kubernetes

Apache MESOS

CPU, Memory, etc.

Process Metrics

System Logs/Events


Application Status Dashboard with Logs and Metrics
Application Key Performance Indicators
Latency by Application Component
Loadbalancer Latency
System Performance
CPU Usage
Memory Usage
15M
02:38 PM
02:40 PM
02:42 PM

- Rollups vs. Detailed
- What resolution makes sense?
- Is real-time necessary?

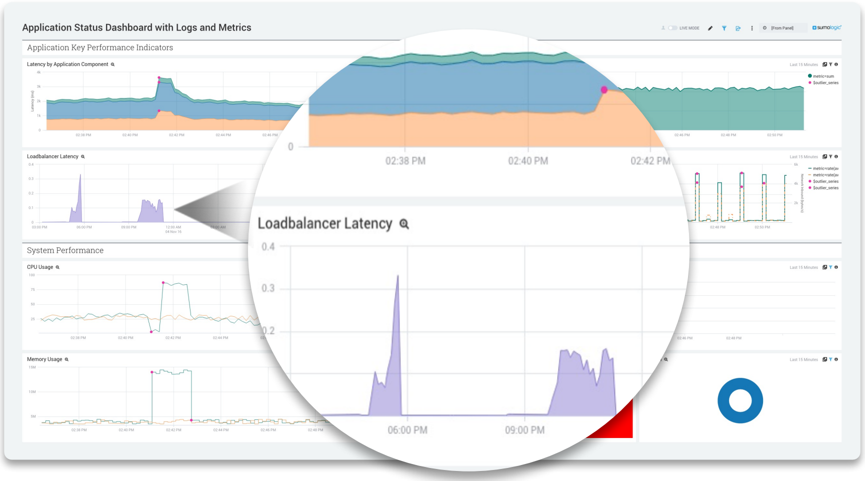
Data From all Parts of Your Stack

Platform

Errors

Latency

Stack Traces


Application Status Dashboard with Logs and Metrics
Application Key Performance Indicators
Latency by Application Component
Loadbalancer Latency
System Performance
CPU Usage
Memory Usage
02:38 PM
02:40 PM
02:42 PM
06:00 PM
09:00 PM

- Rollups vs. Detailed
- Coverage of all components
- Detailed logs for investigations
- Architecture in the metadata

Data From all Parts of Your Stack

Custom

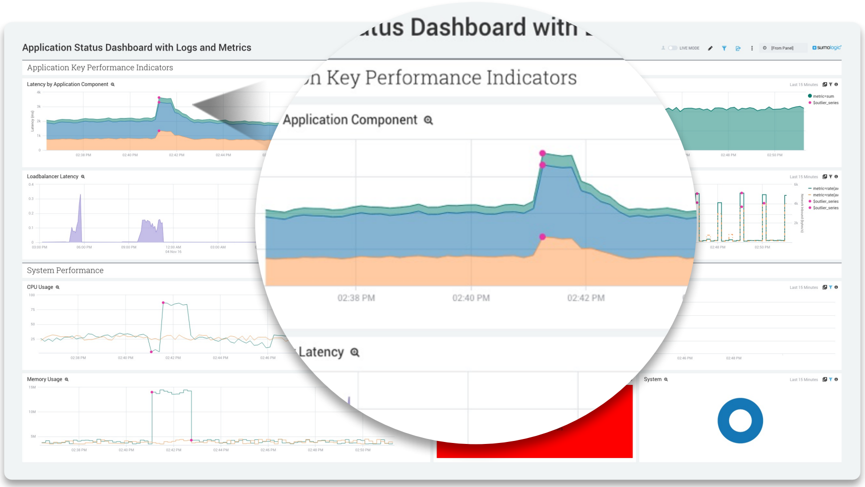
AWS ELB

AWS DynamoDB

User Visits

User Activity

Transactions


Application Status Dashboard with Logs and Metrics
Application Key Performance Indicators
Latency by Application Component
Loadbalancer Latency
System Performance
CPU Usage
Memory Usage
System
Application Component
Latency
02:38 PM
02:40 PM
02:42 PM

- How is your service measured?
- What frustrates users?
- How does the business measure itself?
- The business in the metadata

Data From all Parts of Your Stack

Infrastructure

AWS ECS

docker

Microsoft Windows

kubernetes

Apache MESOS

Samples

CPU, Memory, etc.

Process Metrics

System Logs / Events

Platform

amazon web services

Java

RAILS

node JS

Samples

Errors

Latency

Stack Traces

Custom

AWS ELB

AWS DynamoDB

Apache

NGINX

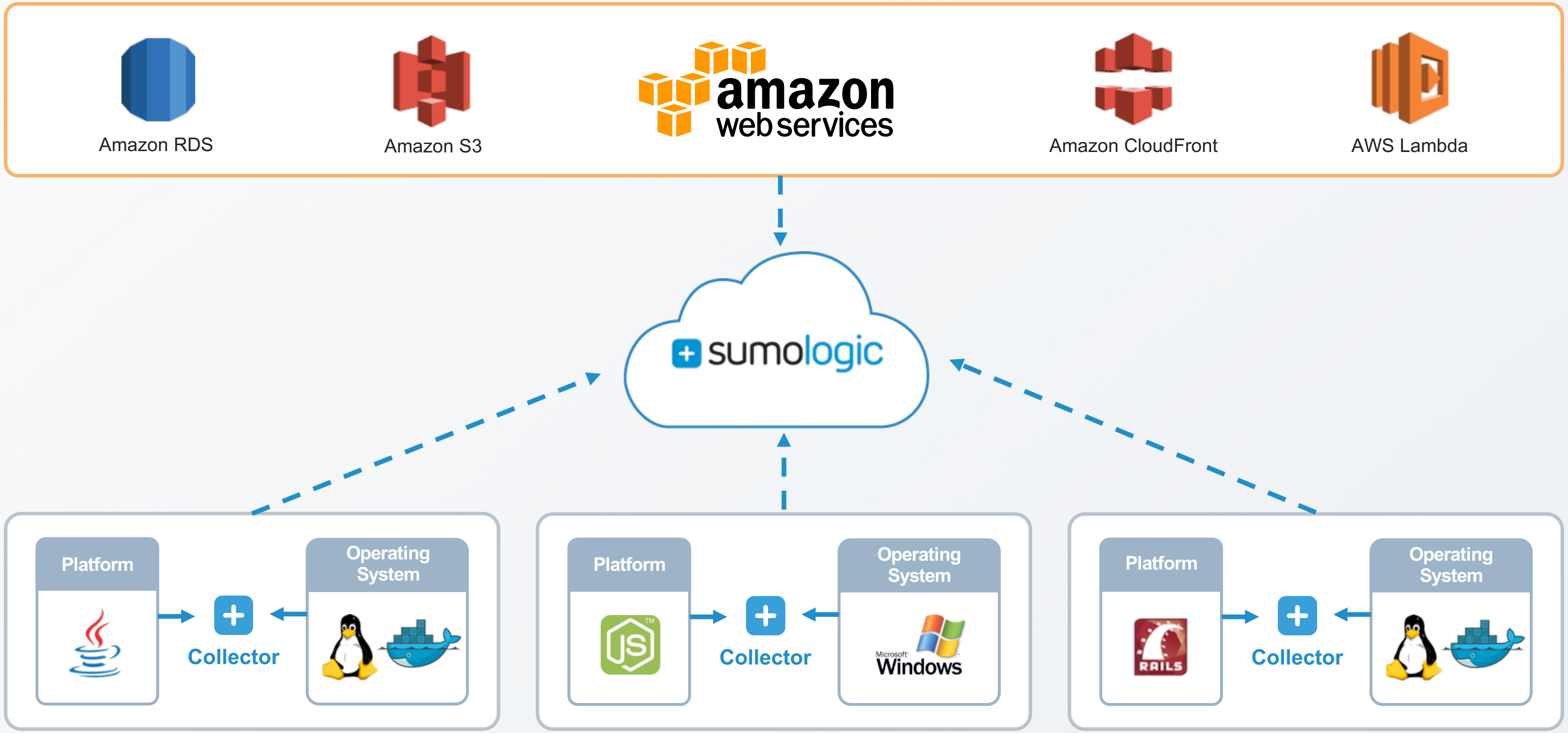
Samples

User Visits

User Activity

Transactions

How Do You Collect the Data You Need?



Use Data Analytics to Your Advantage

Monitoring

Focus on **User Visible** Functionality

Measure and monitor **what matters to your users**

Send notifications to **incident management platforms** (PagerDuty, VictorOps, etc.) and/or **Collaboration Tools** (Slack, etc.), rather than flood your engineers with email

Troubleshooting

Focus on **End-to-End** Visibility

Late/Delayed Metrics mean late/delayed resolution – **real-time matters**

Tie your **Playbook** instructions directly to the data (search this, look at this, etc.)

Correlate performance (Metrics) with what happened (Logs) to resolve issues quickly

App Intelligence

Focus on **User Activity** & **Visibility**

Use Metadata to make the data reflect your view of the world, not vice versa

Good user activity data will improve your product and your user experience

Keep long term trends of your data to understand your progress

Case Study: Cloud Cruiser

CLOUD CRUISER

Cloud Cruiser

Meter and Manage Your Cloud Spend

Ben Abrams, Lead DevOps Engineer

Official Title:
Lead “DevOps” Engineer

Preferred Title:
“Supreme Unicorn Hunter of Planet Earth and the Entire Galaxy Besides”

“My team supports all aspects of Engineering (Dev, QA, Ops, Sales, and Product). I like to think of us as an Application Delivery Team.”

“We provide a SaaS app which enables you to easily collect, meter, and understand your cloud spend in AWS, Azure, and GCP.”

“Our SaaS app manages 100’s of millions of cloud spend.”

“Our customers are large enterprises and mid-market players globally distributed across all verticals.”

Our Tech Stack

Application

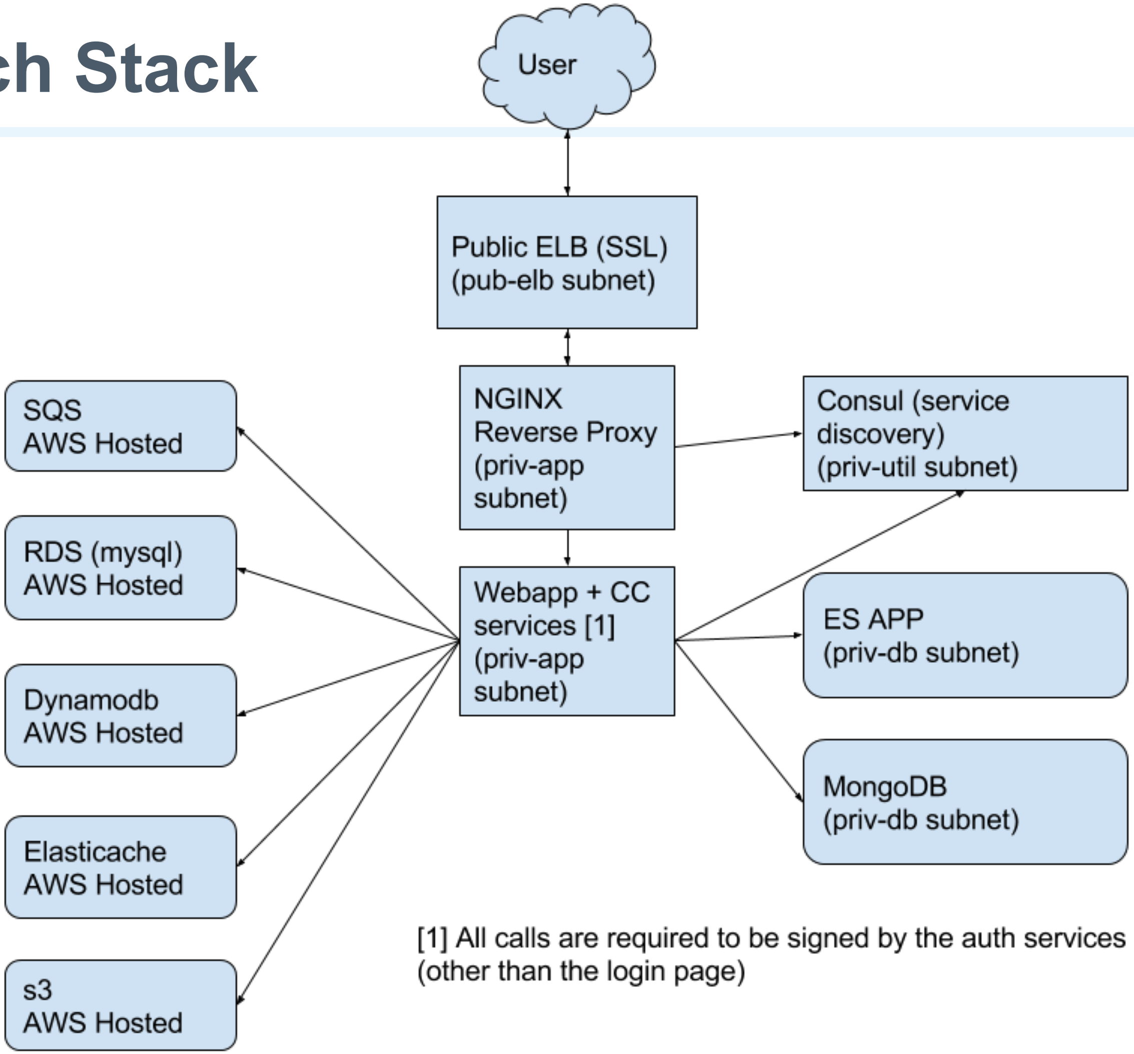
- Microservices written in Java using dropwizard framework
- Angularjs + Tomcat (webapp)
- Elasticsearch
- DynamoDB
- Elasticache (memcached)
- Blob Storage (s3)
- Quartz Scheduler (RDS)

Infrastructure

- AWS (300-500 instances)
- Linux (ubuntu)
- Chef, Terraform, Packer
- Ruby
- Consul
- Nginx (reverse proxy)
- Elasticsearch + MongoDB
- Jenkins
- Sensu

CLOUD CRUISER

Our Tech Stack



[1] All calls are required to be signed by the auth services (other than the login page)

CLOUD CRUISER

Why We Came to Sumo Logic

Burdens of ELK

- Operational burden / distraction
- Security
- Scale + Cost

Metrics

- Had trust in Sumo Logics' ability to deliver – already a happy log customer
- Prevent another tool being managed/added

CLOUD CRUISER

Value We Got from Sumo Logic

Logs

- Reduced operational burden
- Reduced cost
- Increased confidence in log integrity
- Was able to reduce the number of people needing VPN
- Alerting based on searches did not need ops handholding (previously did with Sensu)

Metrics

- Increased visibility in system and application health
- Used in an ongoing effort with application and infrastructure changes in which we were able to reduce our monthly AWS bill by over 100%

The **Rollout**…

What are we using to get this?

- Chef: automation of config and collector install
- Application Graphite Metrics from Dropwizard
- Other graphite metrics forwarded by Sensu to Sumo Logic

Naming Conventions

Logs

Our Schema:
_sourceCategory=$ENV/$LOG_TYPE/$SERVER_ROLE

Breakdown:

- **ENV**: prod-west
- **LOG_TYPE**: nginx_access
- **SERVER_ROLE**: this corresponds to a chef role

Metrics

Our Schema:
_sourceCategory=$ENV/metrics/$METRIC_TYPE/$METRIC_SOURCE

Breakdown:

- **ENV**: prod-west
- **METRIC_TYPE**: graphite, statsd, host
- **METRIC_SOURCE**: who sent the actual metrics. This corresponds to a chef role. Remember to consider metric forwarders

Deploying with Chef

Base Sumo Logic Config and Install

```
 remote_file
"#{Chef::Config[:file_cache_path]}/sumocollector.deb" do
      source node['sumologic']['collectorDEBUrl']
    end

    dpkg_package 'sumocollector' do
      source
"#{Chef::Config[:file_cache_path]}/sumocollector.deb"
      action :install
    end

    service 'collector' do
      action [:enable, :start]
    end
```

```
template node['sumologic']['sumo_conf_path'] do
  cookbook node['sumologic']['conf_config_cookbook']
  source conf_source
  sensitive true
  owner 'root'
  group 'root'
  mode '0600'
  variables(accessID: credentials[:accessID],
            accessKey: credentials[:accessKey])
  notifies :restart, 'service[collector]', :delayed
end

directory node['sumologic']['sumo_json_path'] do
    owner 'root'
    group 'root'
    mode '0755'
    action :create
  end
```

Don't log secrets!

These are from an encrypted data bag (defined elsewhere/ out of scope)

CLOUD CRUISER

Log Collector Setup

```
role = node.roles[0]

  # syslog
  syslog_excludes =
node['cc']['sumologic']['syslog']['filters']
  sumo_source_local_file 'localfile-syslog' do
    description 'Syslog'
    source_json_directory node['sumologic']['sumo_json_path']
    category "#{node.chef_environment}/syslog/#{role}"
    path_expression '/var/log/syslog'
    filters [
      syslog_excludes['dhcp']
    ]

    only_if { node['platform_family'].include? 'debian' }
  end
```

```
# cc service logs
  sumo_source_local_file 'localfile-microservice' do
    description 'Microservice Log File'
    source_json_directory node['sumologic']['sumo_json_path']
    category "#{node.chef_environment}/microservice/#{role}"
    path_expression '/var/log/cc/*/*.log'
    multiline_processing_enabled true
    use_autoline_matching false
    manual_prefix_regexp '^[A-Z]+\s+\[\d{4}-\d{2}-
\d{2}\s+\d{2}:\d{2}:\d{2}\,\d{3}\].*'
    only_if { node.roles.include? 'microservice_base' }
  end
```

CLOUD CRUISER

Metric Collector Setup

```
  # host metrics
  template
"#{node['sumologic']['sumo_json_path']}/systemstats-
default.json" do
    source 'systemstats.json.erb'
    action :create
    notifies :restart, 'service[collector]', :delayed
    variables(category:
"#{node.chef_environment}/metrics/systemstats/#{role}",
              description: 'Host Metrics',
              name: 'systemstats-default',
              interval:
node['cc']['sumologic']['systemstats_frequency'])
  end
```

```
# host metrics template
{
    "api.version": "v1",
    "source":
        {
            "name": "<%= @name %>",
            "sourceType": "SystemStats",
            <% if @category %>
            "category": "<%= @category %>",
            <% end %>
            <% if @hostName %>
            "hostName": "<%= @hostName %>",
            <% end %>
            <% if @description %>
            "description": "<%= @description %>",
            <% end %>
            "interval": <%= @interval %>
        }
}
```

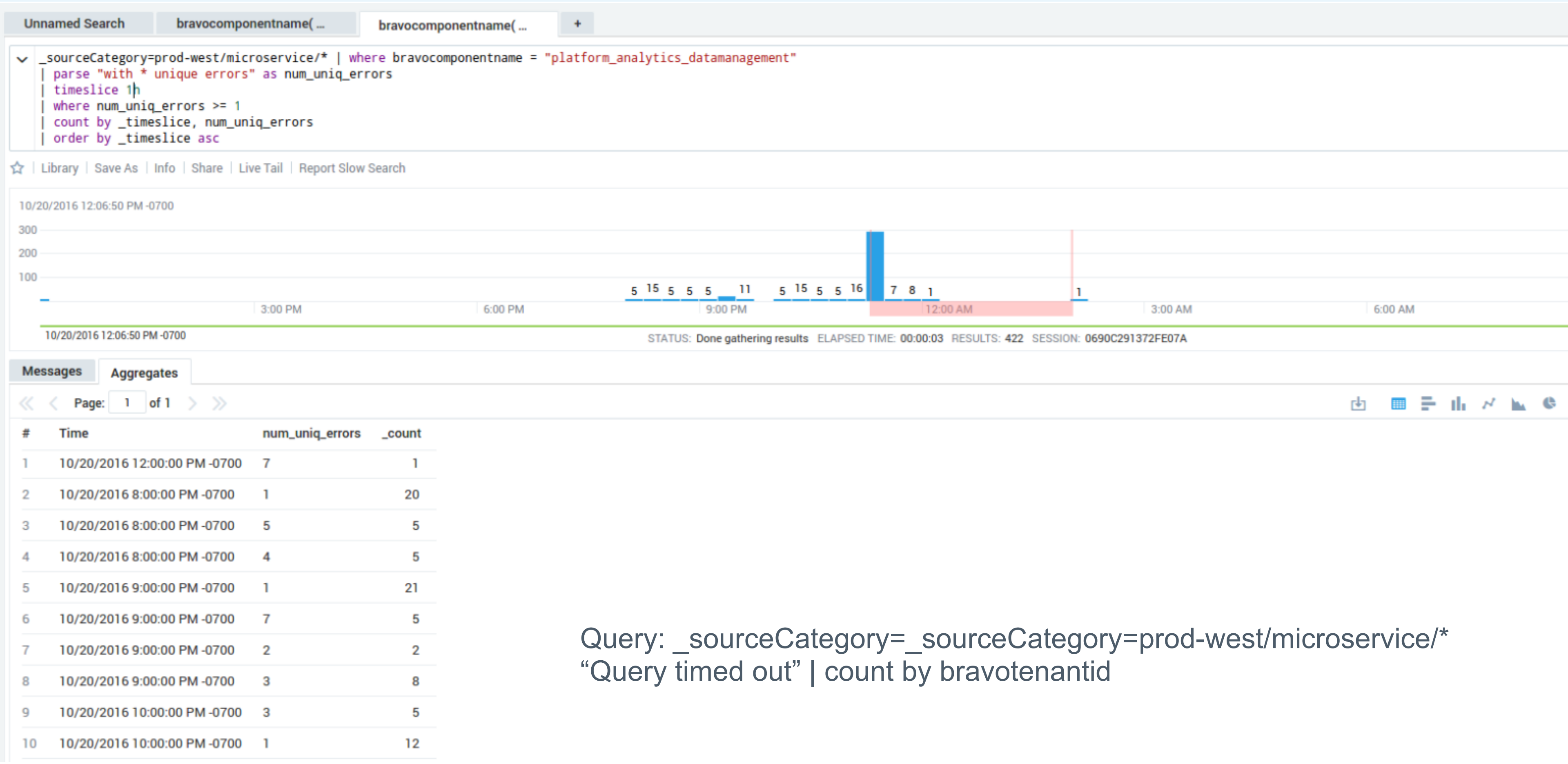
Graphite Metrics

```
  # graphite metrics
  template "#{node['sumologic']['sumo_json_path']}/graphite-
default.json" do
    source 'graphite.json.erb'
    action :create
    notifies :restart, 'service[collector]', :delayed
    variables(category:
"#{node.chef_environment}/metrics/graphite/#{role}",
              description: 'Graphite Metrics',
              protocol: 'TCP',
              port:
node['cc']['sumologic']['dropwizard']['port'],
              name: 'graphite-default')
    only_if { node.roles.include? 'microservice_base' }
  end
```

```
# graphite metrics template
{
    "api.version": "v1",
    "source":
        {
            "name": "<%= @name %>",
            <% if @category %>
            "category": "<%= @category %>",
            <% end %>
            <% if @description %>
            "description": "<%= @description %>",
            <% end %>
            <% if @protocol %>
            "protocol": "<%= @protocol %>",
            <% end %>
            <% if @port %>
            "port": <%= @port %>,
            <% end %>
            "sourceType": "Graphite"
        }
}
```

Search Examples

Searching Logs



```
_sourceCategory=prod-west/microservice/* | where bravocomponentname = "platform_analytics_datamanagement"
| parse "with * unique errors" as num_uniq_errors
| timeslice 1h
| where num_uniq_errors >= 1
| count by _timeslice, num_uniq_errors
| order by _timeslice asc
```

10/20/2016 12:06:50 PM -0700

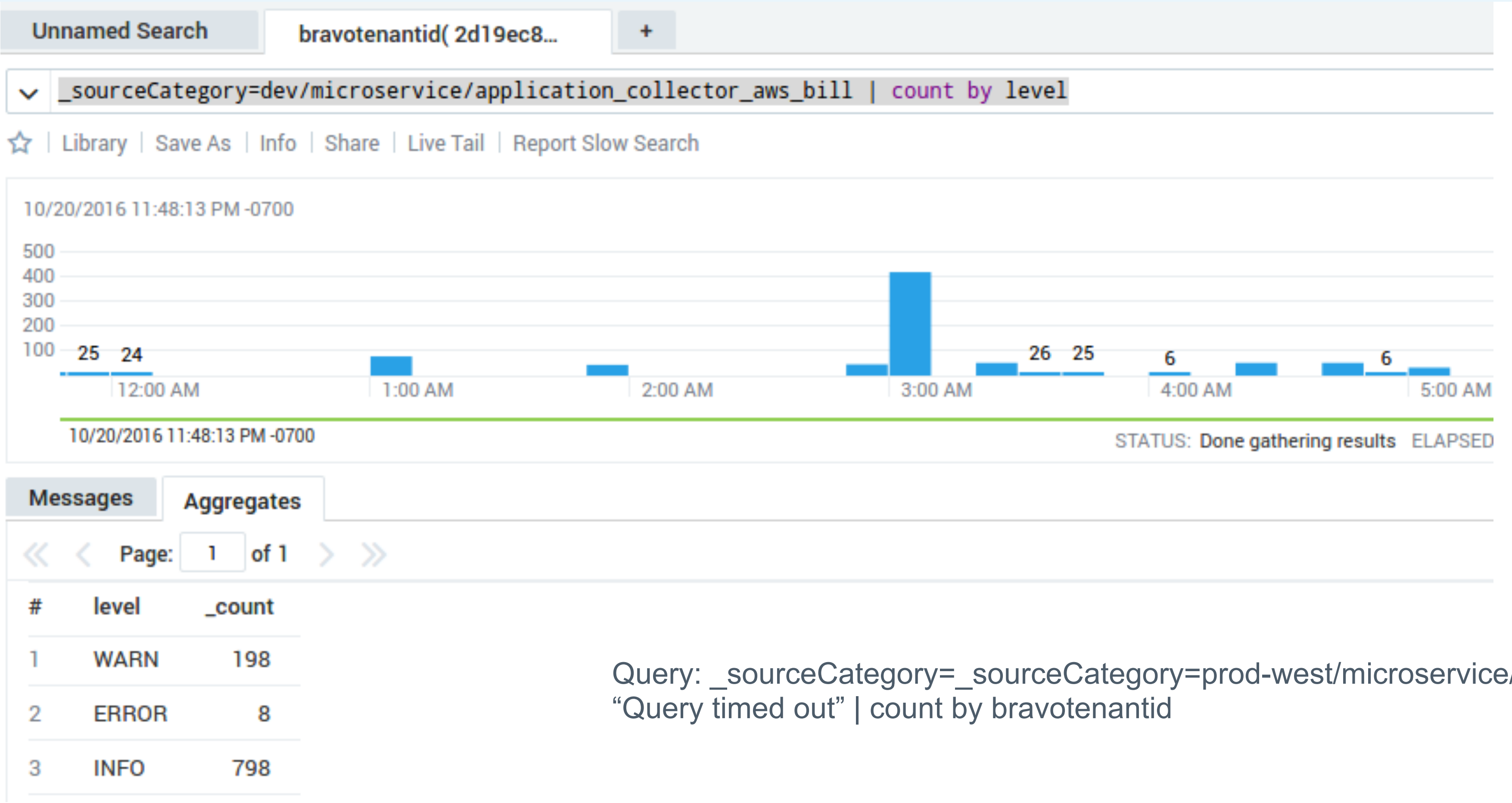
10/20/2016 12:06:50 PM -0700

STATUS: Done gathering results ELAPSED TIME: 00:00:03 RESULTS: 422 SESSION: 0690C291372FE07A

#	Time	num_uniq_errors	_count
1	10/20/2016 12:00:00 PM -0700	7	1
2	10/20/2016 8:00:00 PM -0700	1	20
3	10/20/2016 8:00:00 PM -0700	5	5
4	10/20/2016 8:00:00 PM -0700	4	5
5	10/20/2016 9:00:00 PM -0700	1	21
6	10/20/2016 9:00:00 PM -0700	7	5
7	10/20/2016 9:00:00 PM -0700	2	2
8	10/20/2016 9:00:00 PM -0700	3	8
9	10/20/2016 10:00:00 PM -0700	3	5
10	10/20/2016 10:00:00 PM -0700	1	12

Query: _sourceCategory=_sourceCategory=prod-west/microservice/* "Query timed out" | count by bravotenantid

Searching Logs


Unnamed Search
bravotenantid(2d19ec8...
+
Library | Save As | Info | Share | Live Tail | Report Slow Search
Messages
Aggregates
Page: 1 of 1

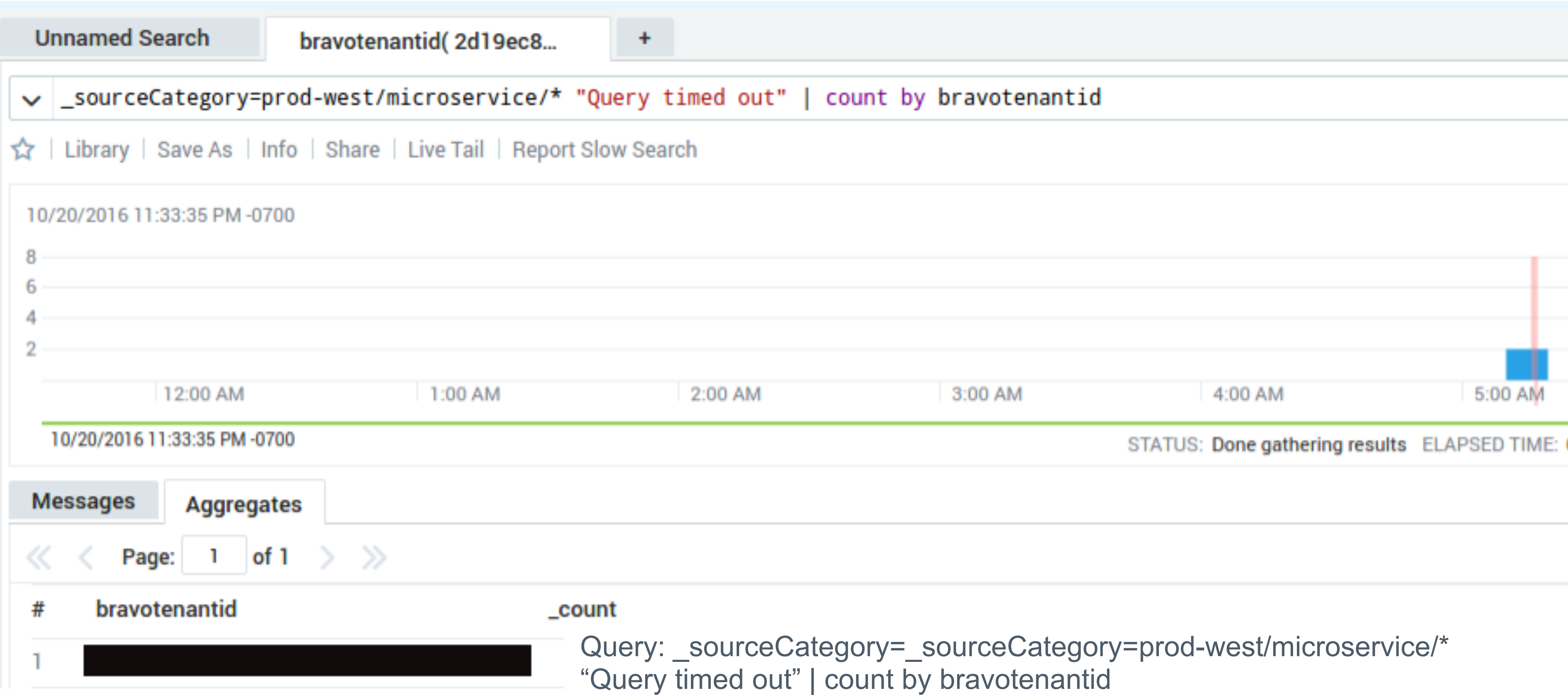
```
_sourceCategory=dev/microservice/application_collector_aws_bill | count by level
```

10/20/2016 11:48:13 PM -0700
500
400
300
200
100
25
24
26
25
6
6
12:00 AM
1:00 AM
2:00 AM
3:00 AM
4:00 AM
5:00 AM
10/20/2016 11:48:13 PM -0700
STATUS: Done gathering results ELAPSED

#	level	_count
1	WARN	198
2	ERROR	8
3	INFO	798

Query: _sourceCategory=_sourceCategory=prod-west/microservice/* "Query timed out" | count by bravotenantid

Searching Logs



Unnamed Search | bravotenantid(2d19ec8...

```
_sourceCategory=prod-west/microservice/* "Query timed out" | count by bravotenantid
```

Library | Save As | Info | Share | Live Tail | Report Slow Search
10/20/2016 11:33:35 PM -0700
8
6
4
2
12:00 AM
1:00 AM
2:00 AM
3:00 AM
4:00 AM
5:00 AM
10/20/2016 11:33:35 PM -0700

STATUS: Done gathering results ELAPSED TIME:

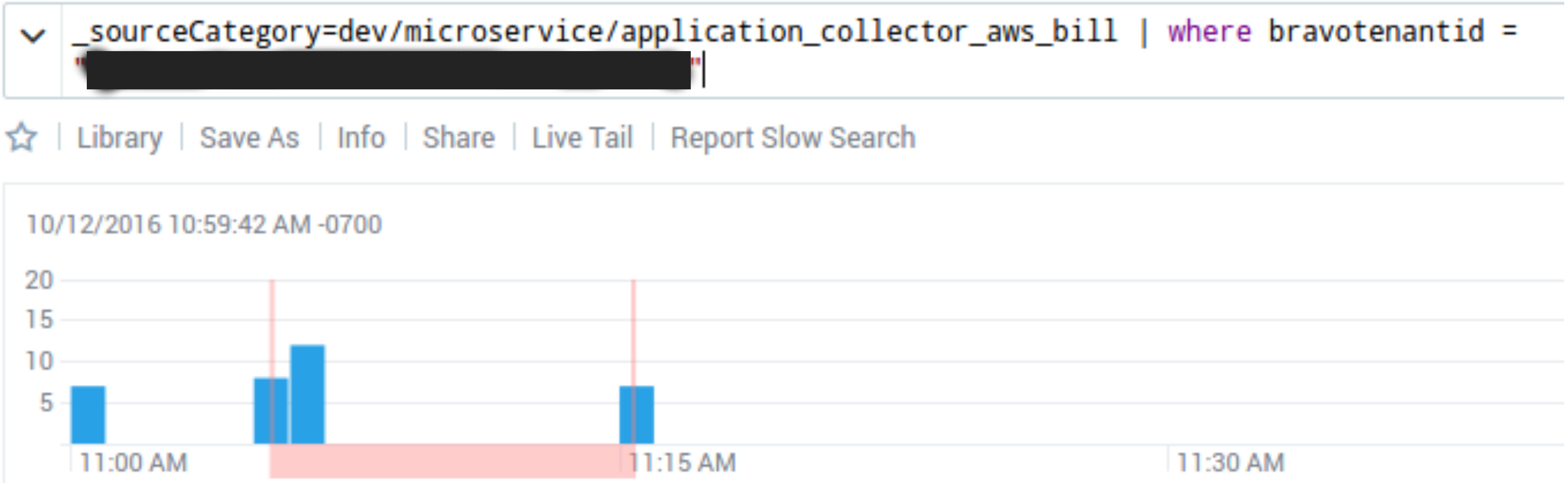
Messages | Aggregates

Page: 1 of 1

#	bravotenantid	_count
1	[redacted]	

Query: _sourceCategory=_sourceCategory=prod-west/microservice/* “Query timed out” | count by bravotenantid

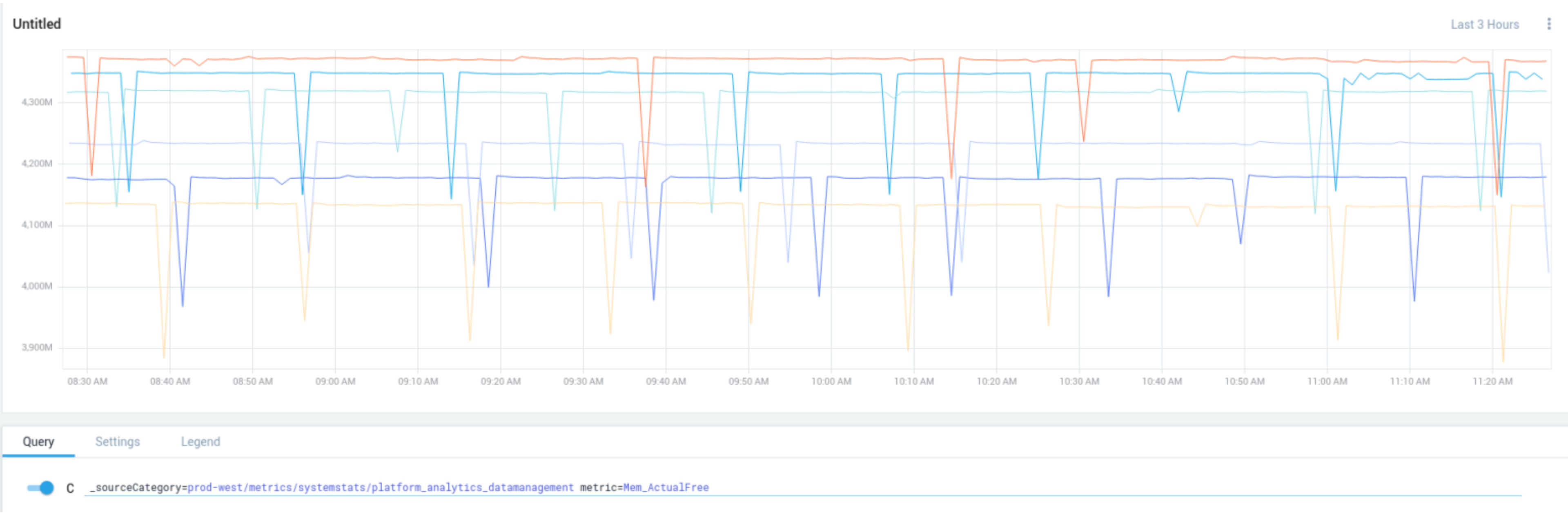
Searching Logs



```
_sourceCategory=dev/microservice/application_collector_aws_bill | where bravotenantid =
"[redacted]"
```

Query:
_sourceCategory=_sourceCategory=dev/microservice/application_collector_
aws_bill | where bravotenantid = "SOME_TENANT_UID"

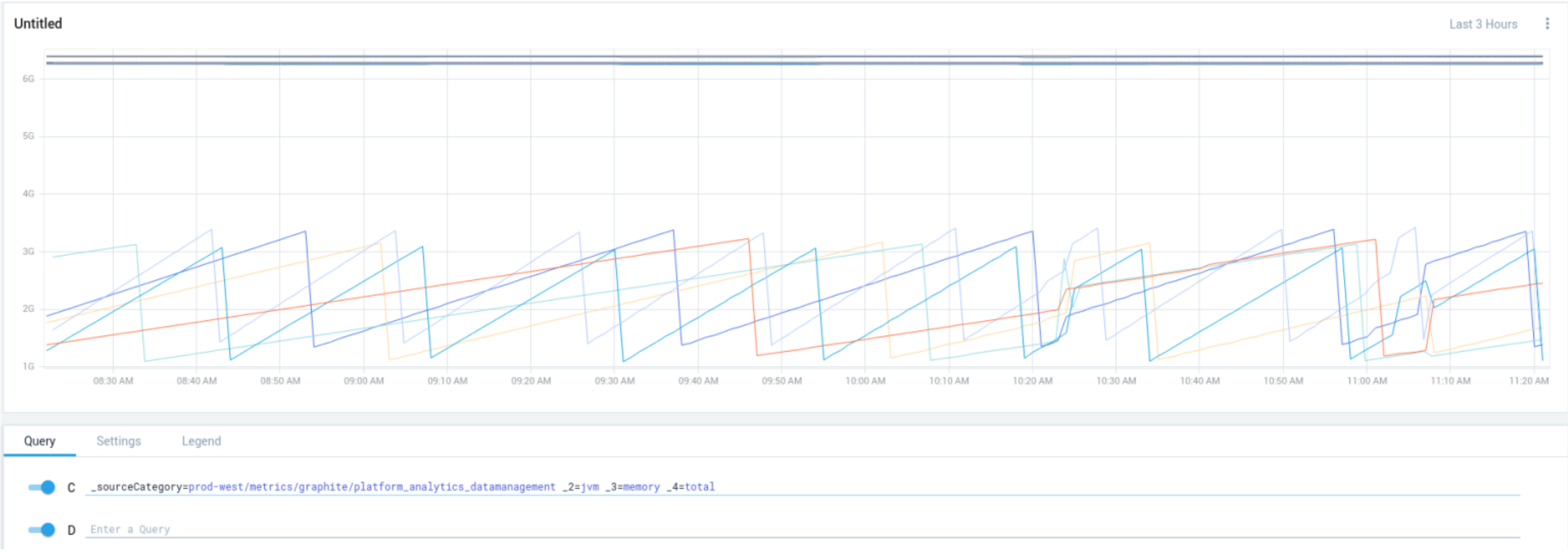
Searching Metrics



```
C  _sourceCategory=prod-west/metrics/systemstats/platform_analytics_datamanagement metric=Mem_ActualFree
```

Query: _sourceCategory=prod-west/metrics/systemstats/platform_analytics_datamanagement metric=Mem_ActualFree

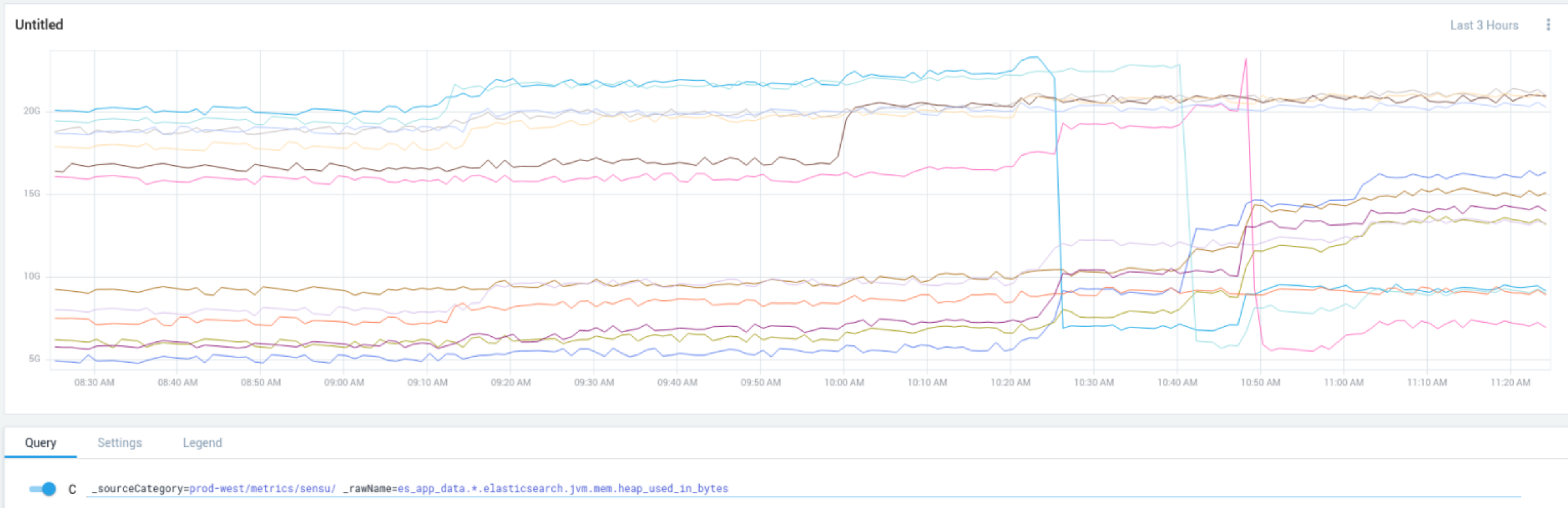
Searching Metrics



```
C  _sourceCategory=prod-west/metrics/graphite/platform_analytics_datamanagement _2=jvm _3=memory _4=total
D  Enter a Query
```

Query: _sourceCategory=prod-west/metrics/graphite/platform_analytics_datamanagement _2=jvm _3=memory _4=total

Searching Metrics



```
_sourceCategory=prod-west/metrics/sensu/ _rawName=es_app_data.*.elasticsearch.jvm.mem.heap_used_in_bytes
```

Query: _sourceCategory=prod-west/metrics/sensu/
_rawName=es_app_data.*.elasticsearch.jvm.mem.heap_used_in_bytes

Dashboards for Metrics and Logs

Dashboards tell you what’s going on…

- Relevant Data gives you birds eye view
- Cut troubleshooting time
- Service specific

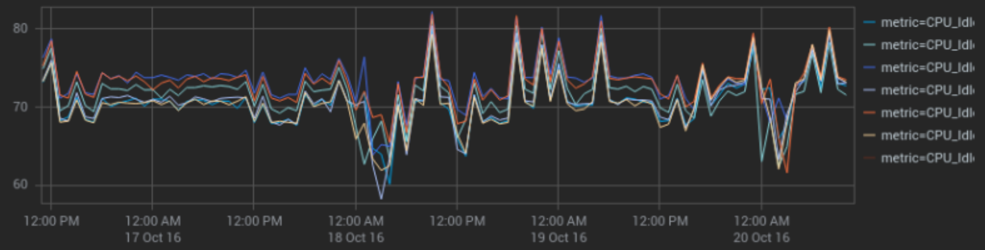
Prod-west Analytics Datamanagement Perf

Done Editing

Host Metrics (local) +

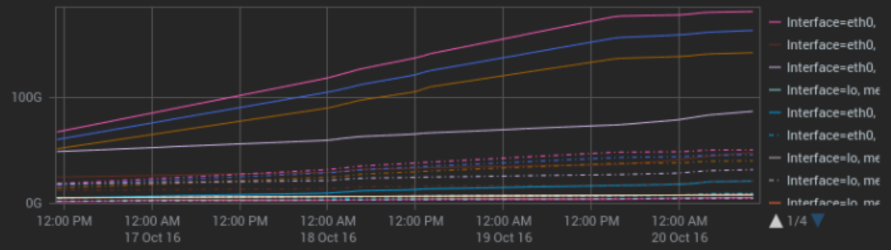
CPU

Last 4 Days



Network

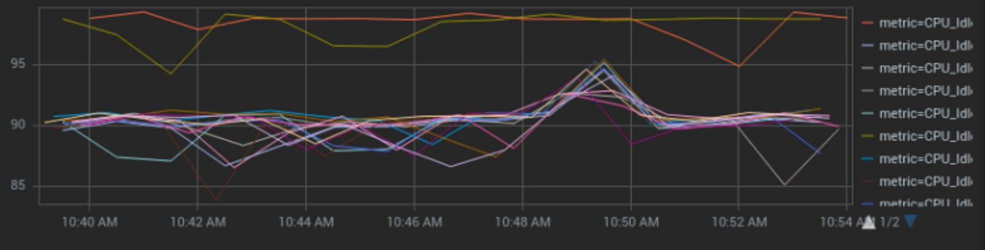
Last 4 Days



ES Host Metrics +

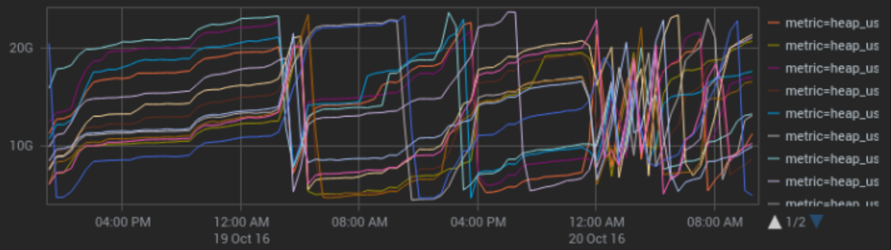
ES CPU Idle

Last 15 Minutes



Prod ES data Nodes Heap

Last 2 Days



Other ES Metrics

ES Cluster St...

Last Minute

Healthy

ES Data Nodes Consul ...

Today

THERE IS NO DATA TO DISPLAY.

SHOW IN SEARCH

ES Queries Timed out b...

Today

[redacted]

No data is good

You should normally see:

Healhy -> no data -> no data

You might see some consul timeouts this indicates load (gc) more than an actual issue. If you see timeouts check the tenant id and determine if its a data size issue or if the user is not filtering their reports.

Key Takeaways

Monitoring

Focus on **User Visible** Functionality

Measure and monitor what **matters**

Troubleshooting

Focus on **End-to-End** Visibility

You need **Logs and Metrics** to solve real problems

App Intelligence

Focus on **User Activity** & **Visibility**

Use Metadata to make the data reflect your view of the world

Thank you!

Come Visit Sumo Logic at Booth #604

Remember to complete your evaluations!