aws SUMMIT

BDA 308

Deep Dive: Log Analytics With Amazon Elasticsearch Service

Case Study: Amazon ES at Expedia Group

Jon Handler

Principal Search Services SA, AWS

Kuldeep Chowhan

Principal Engineer, Expedia Group

AWS Databases and Analytics

Broad and deep portfolio, purpose-built for builders

Business Intelligence & Machine Learning

QuickSight | SageMaker | Comprehend

Relational Databases

Aurora

RDS

Non-Relational Databases

DynamoDB

ElastiCache
(Redis, Memcached)

Neptune
(Graph)

Analytics

DW | Big Data Processing | Interactive

Amazon Redshift | EMR | Athena

Real-time

Elasticsearch Service | Kinesis Data Analytics

Data Lake

S3/Amazon Glacier

AWS Glue
(ETL & Data Catalog)

Data Movement

Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams

Things with purpose

Database characteristics

Relational	Key-value	Document	Graph	Time Series
Referential integrity with strong consistency, transactions, and hardened scale	Low-latency key based queries with high throughput and fast ingestion of data	Indexing and storing documents with support for query on any property	Creating and navigating relations between data easily and quickly	Time-stamped data with large range-scans for summarization and processing
Complex query support via SQL	Simple query methods with filters	Simple query with filters, projections and aggregates	Easily express queries in terms of relations	Computational support for summarized results

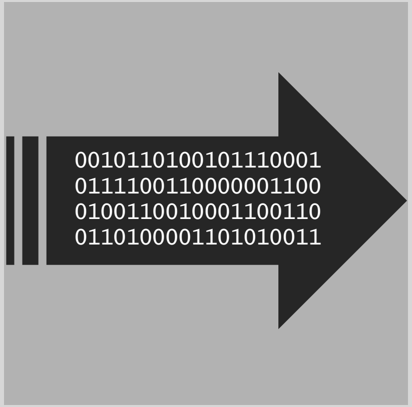
Elasticsearch’s purpose

Text search

Natural language

Boolean queries

Relevance



Streaming data

High-volume ingest

Near real time

Distributed storage

Analysis

Time-based visualizations

Nestable statistics

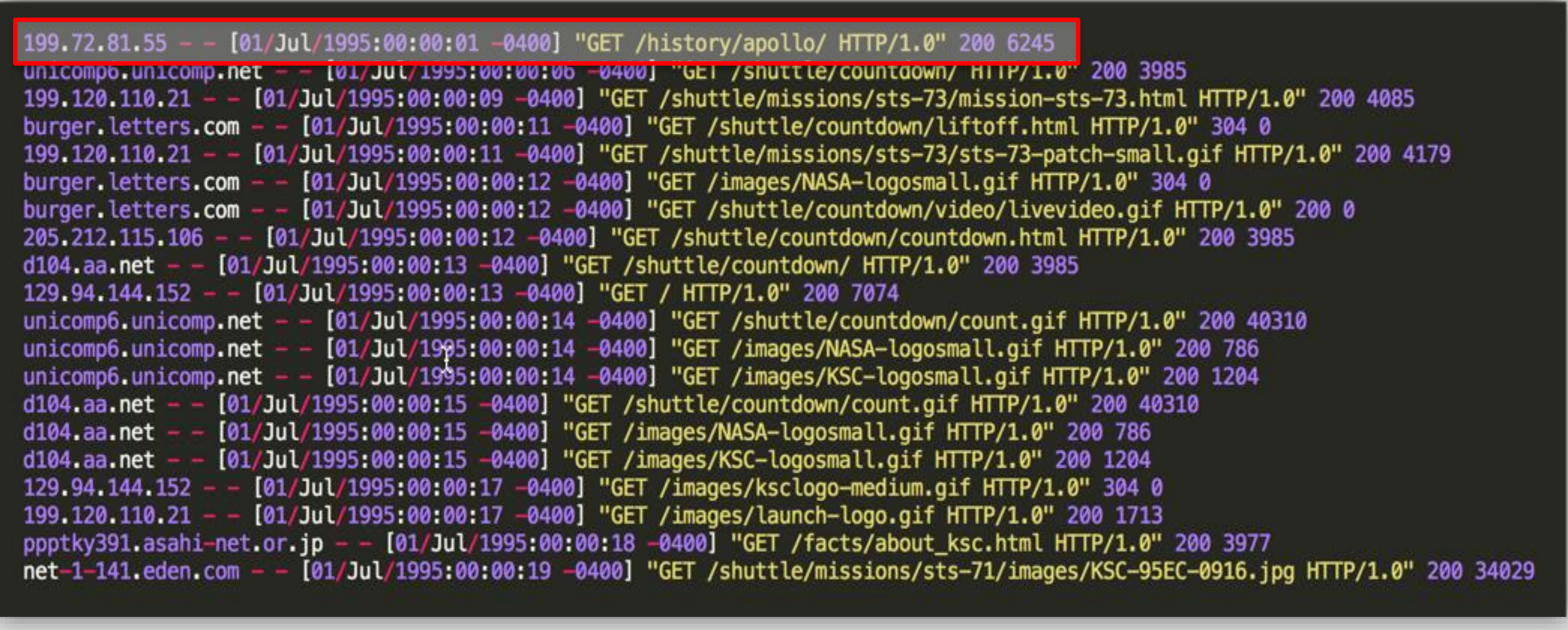
Time series tools

Amazon Elasticsearch Service

Amazon Elasticsearch Service is a **fully managed service** that makes it easy to deploy, manage, and scale Elasticsearch and Kibana in the AWS Cloud

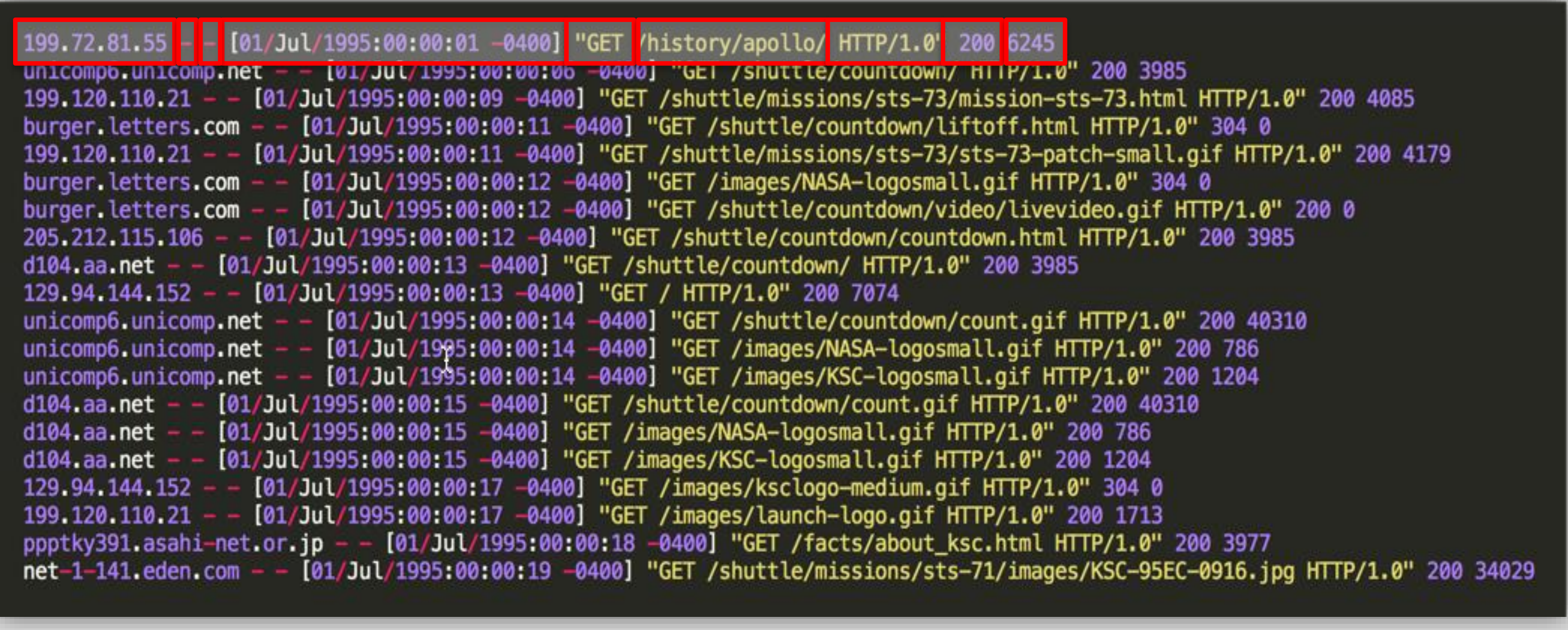
elasticsearch + kibana

Amazon Elasticsearch Service's Storage Layer



```
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085
burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0
199.120.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0" 200 4179
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0
205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html HTTP/1.0" 200 3985
d104.aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
129.94.144.152 - - [01/Jul/1995:00:00:13 -0400] "GET / HTTP/1.0" 200 7074
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
129.94.144.152 - - [01/Jul/1995:00:00:17 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 304 0
199.120.110.21 - - [01/Jul/1995:00:00:17 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713
ppptky391.asahi-net.or.jp - - [01/Jul/1995:00:00:18 -0400] "GET /facts/about_ksc.html HTTP/1.0" 200 3977
net-1-141.eden.com - - [01/Jul/1995:00:00:19 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0916.jpg HTTP/1.0" 200 34029
```

Each log line or other event constitutes a search *document*



```
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085
burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0
199.120.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0" 200 4179
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0
205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html HTTP/1.0" 200 3985
d104.aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
129.94.144.152 - - [01/Jul/1995:00:00:13 -0400] "GET / HTTP/1.0" 200 7074
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
unicomp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
d104.aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
129.94.144.152 - - [01/Jul/1995:00:00:17 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 304 0
199.120.110.21 - - [01/Jul/1995:00:00:17 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713
ppptky391.asahi-net.or.jp - - [01/Jul/1995:00:00:18 -0400] "GET /facts/about_ksc.html HTTP/1.0" 200 3977
net-1-141.eden.com - - [01/Jul/1995:00:00:19 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0916.jpg HTTP/1.0" 200 34029
```

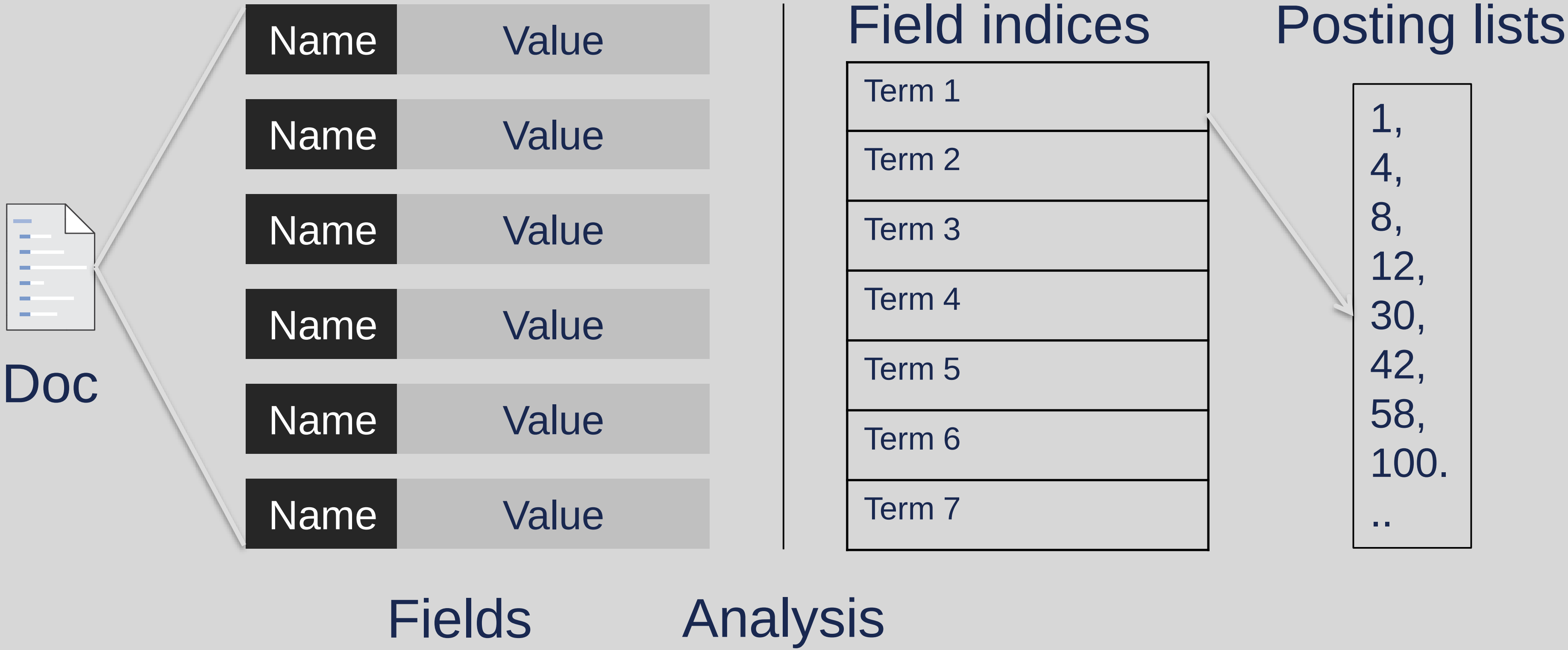
Key Idea

Log lines contain fields

Send JSON to Elasticsearch, with fields and values

```
{
    "host": "199.72.81.55",
    "verb": "GET",
    "request":
        "GET /history/apollo/
         HTTP/1.0",
    "@timestamp":
        "1995-07-01T00:00:01",
    "timezone": "-0400",
    "ident": "-",
    "authuser": "-",
    "response": 200,
    "bytes": 6245
}
```

Lucene creates and stores an index for each field



Doc

Name	Value
Name	Value
Name	Value
Name	Value
Name	Value
Name	Value
Name	Value

Field indices

Term 1
Term 2
Term 3
Term 4
Term 5
Term 6
Term 7

Posting lists

1,
4,
8,
12,
30,
42,
58,
100.
..

Fields

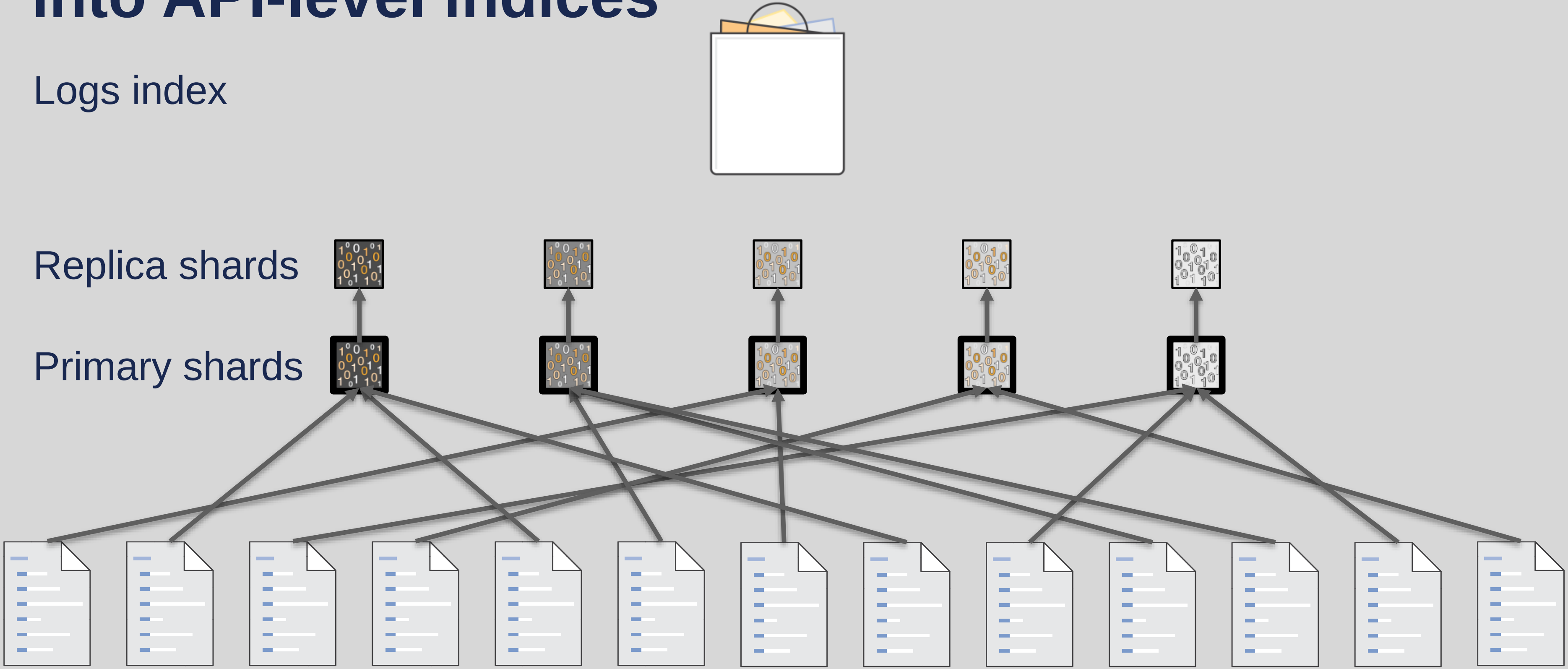
Analysis

Field indices are managed by shards, organized into API-level indices

Logs index

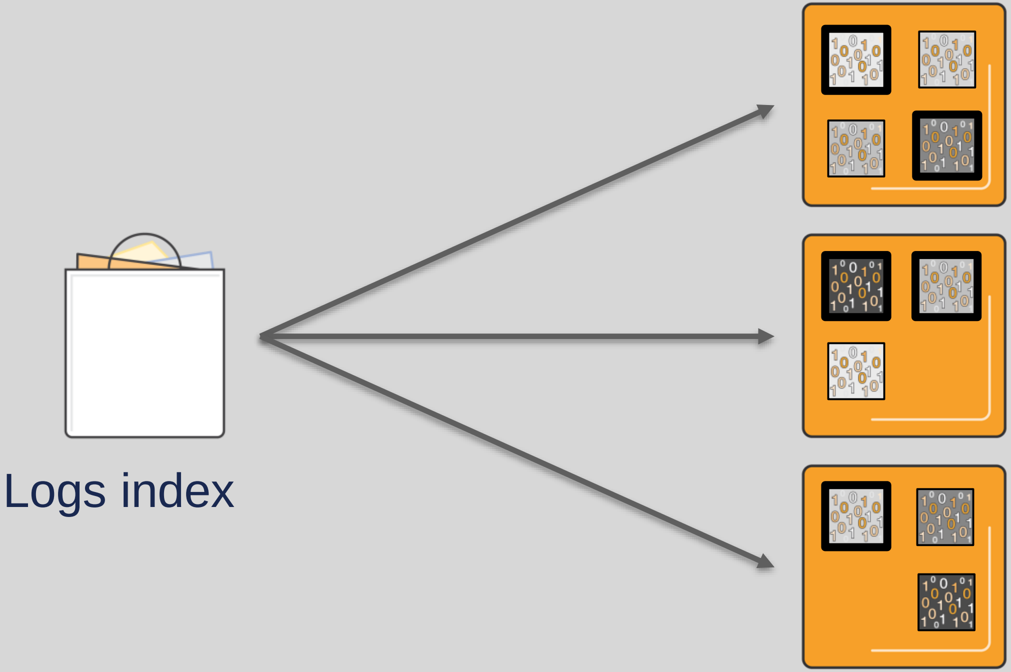
Replica shards

Primary shards



© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Elasticsearch assigns shards to instances



Storage required

- On disk, indices are ~10% larger than source
- Each replica adds an additional 1x storage requirement
- You choose the per-instance storage

Example: a 1 TB corpus will need 2 instances

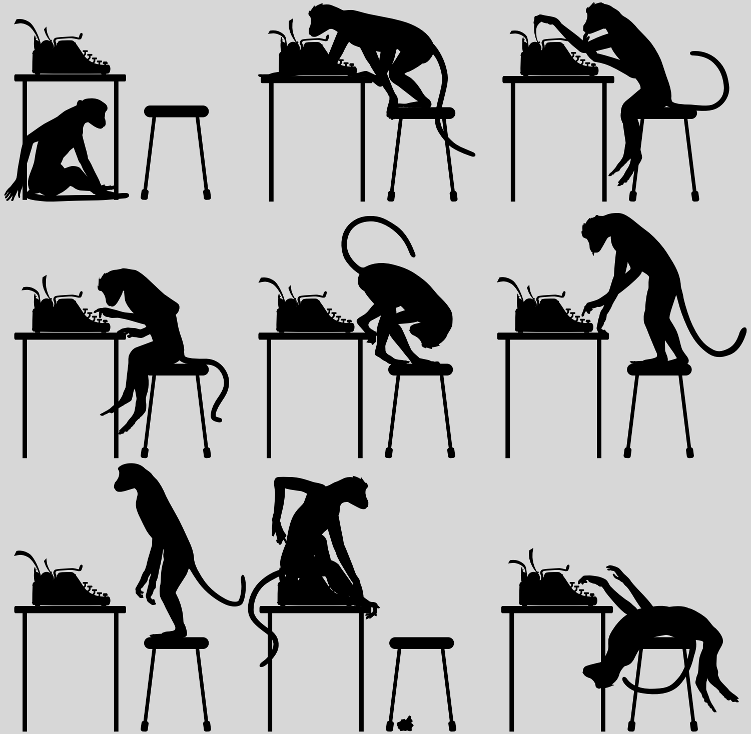
With one replica and 10% inflation, you need 2.2 TB of storage

Choose 1.5 TB of EBS per instance, and you need 2

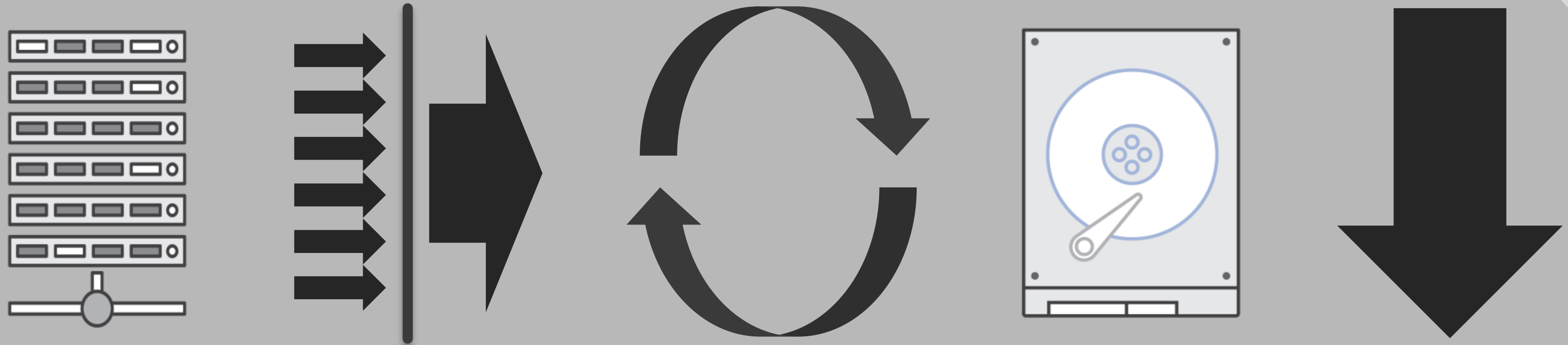
Shards as units of storage

- Set primary shard count based on storage, 40 GB per primary (90 GB for I3 instances)
- Always use at least 1 replica in production
- Keep shard sizes as equivalent as possible

Example: Set shard count = 50 for a 2-TB corpus (2 TB / 40 GB = 50 shards)



Build an ingest pipeline that completes these tasks



Data source

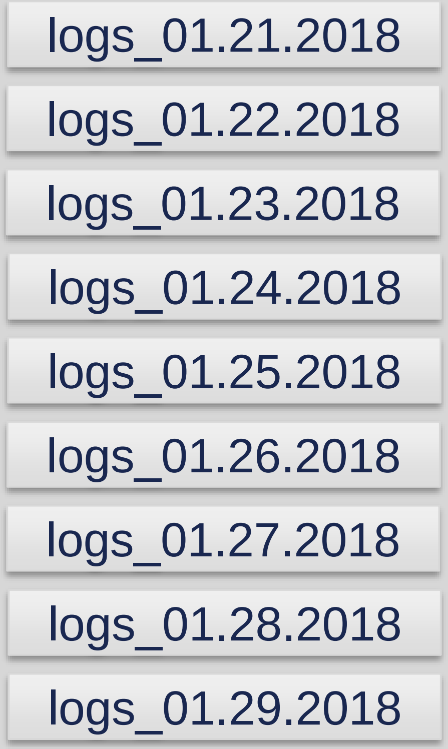
Collect

Transform

Buffer

Deliver

Organize data in daily indexes



logs_01.21.2018

logs_01.22.2018

logs_01.23.2018

logs_01.24.2018

logs_01.25.2018

logs_01.26.2018

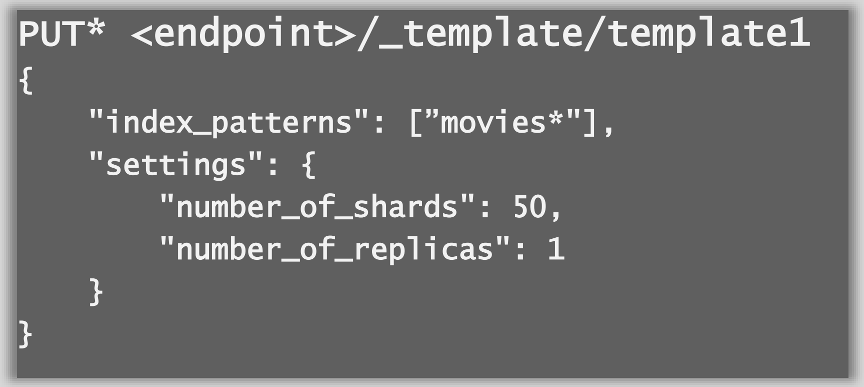
logs_01.27.2018

logs_01.28.2018

logs_01.29.2018

- On ingest, create indexes with a root string, e.g., logs_
- Depending on volume, rotate at regular intervals – normally daily
- Daily indexes simplify index management. Delete the oldest index to create more space on your cluster.

Use templates to set shard count



```
PUT* <endpoint>/_template/template1
{
    "index_patterns": [”movies*"],
    "settings": {
        "number_of_shards": 50,
        "number_of_replicas": 1
    }
}
```

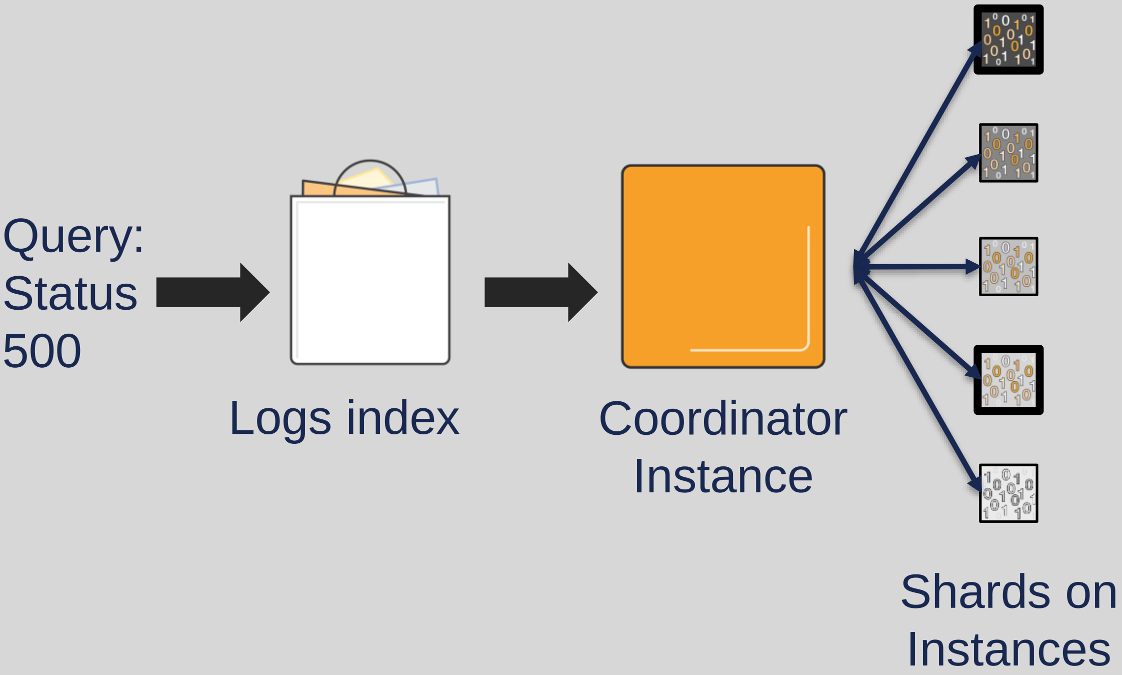
All new indexes that match the index pattern receive the settings

***Note: ES 6.0+ syntax**

aws SUMMIT

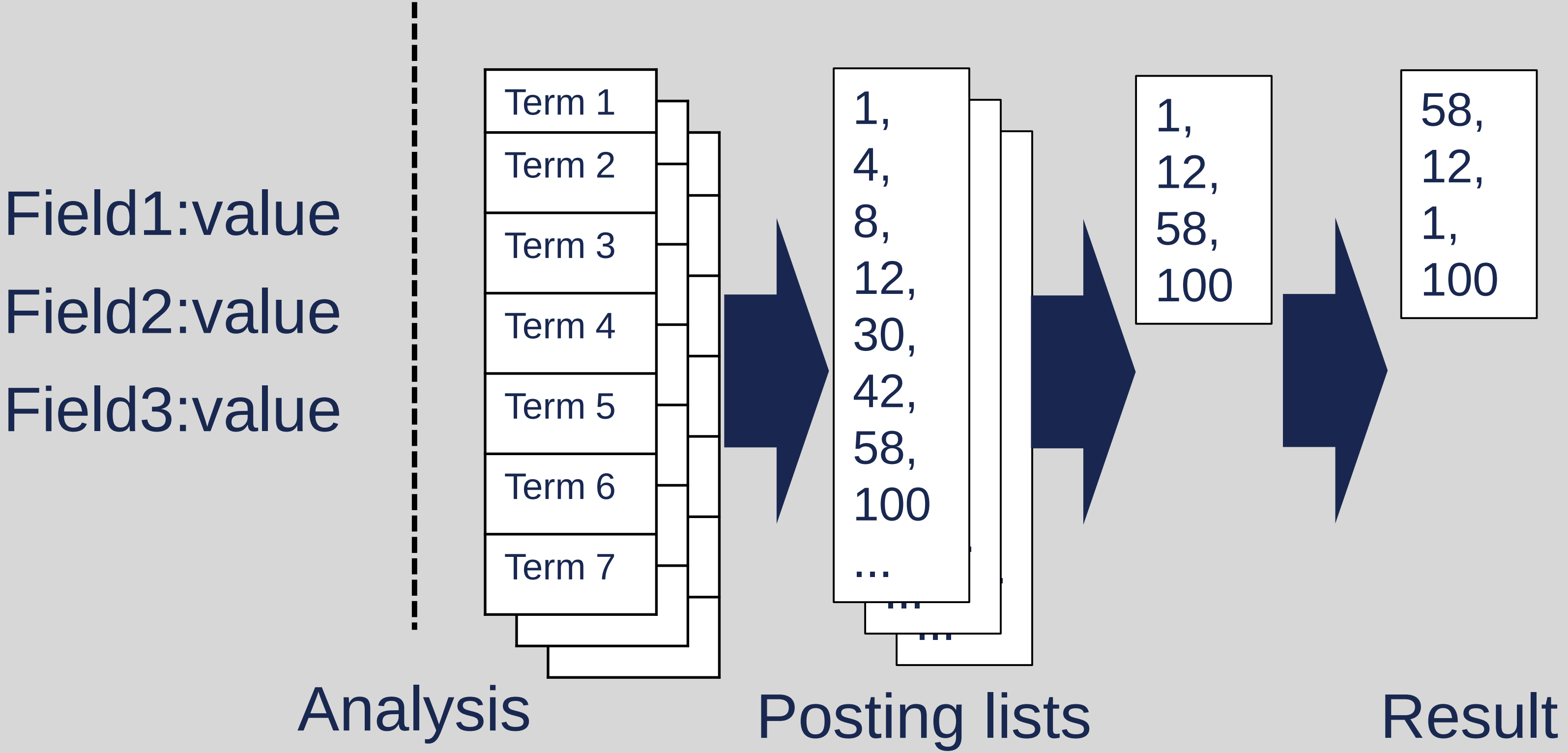
Amazon Elasticsearch Service’s Query Engine

Query distribution

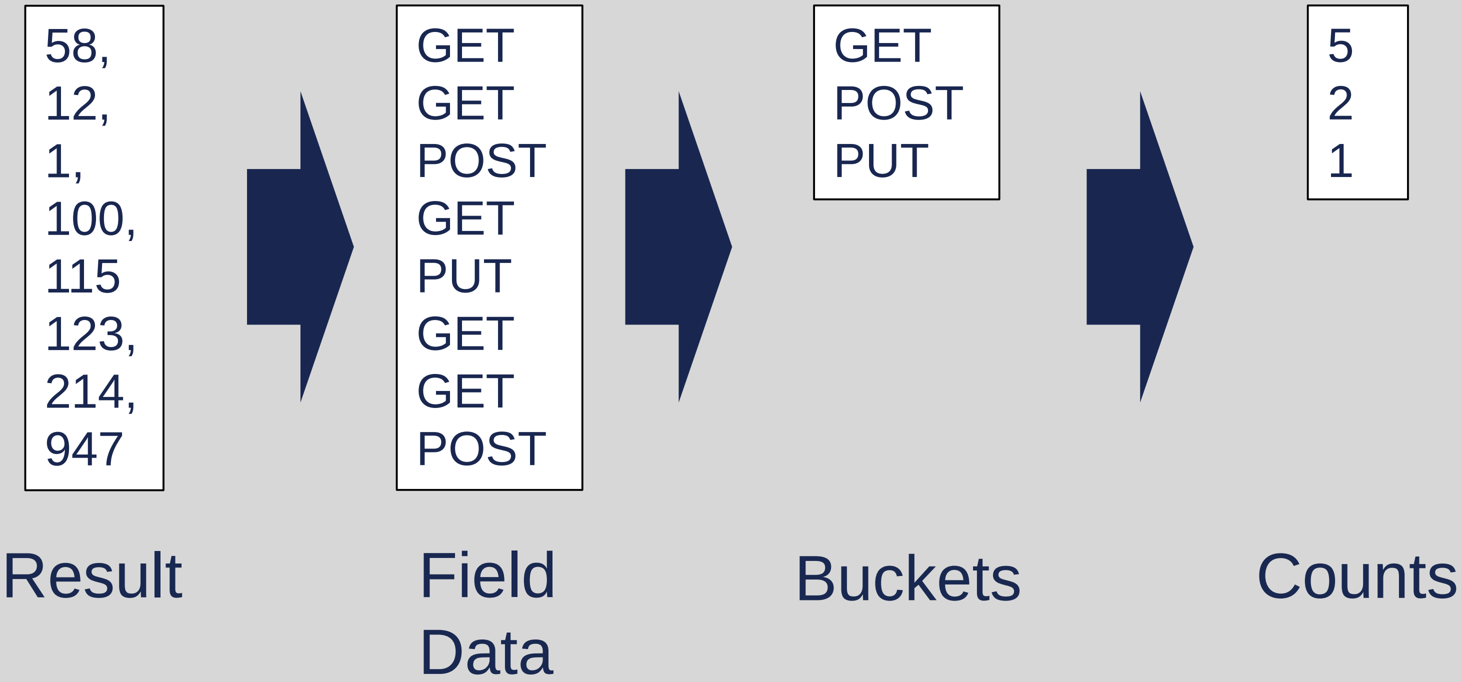


Each shard computes and returns a result to the coordinator, which re-aggregates a final result

Query processing



Analyze field values to get statistics and build visualizations



58,
12,
1,
100,
115
123,
214,
947

GET
GET
POST
GET
PUT
GET
GET
POST

GET
POST
PUT

5
2
1

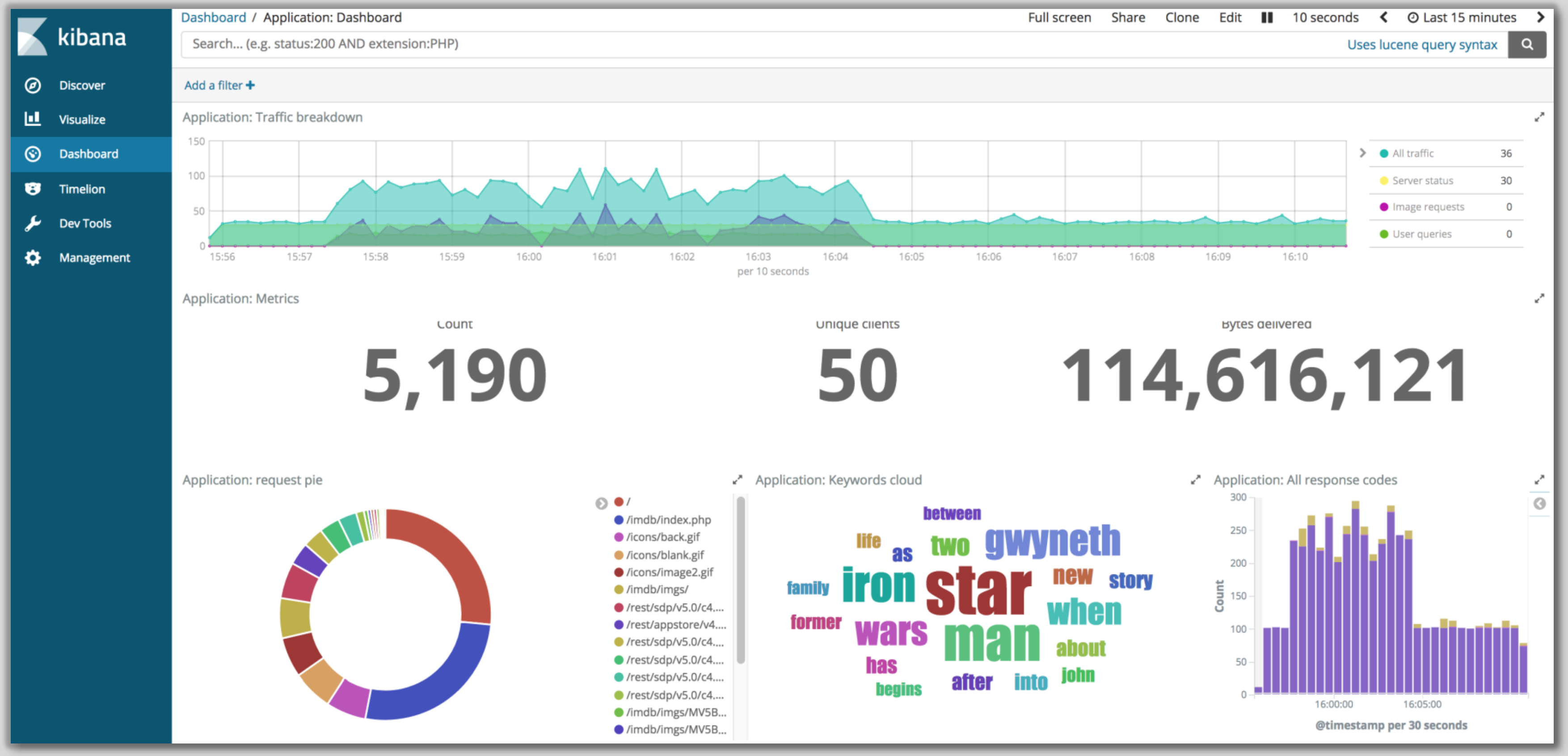
Result

Field Data

Buckets

Counts

Visualize your data



aws SUMMIT

Case study: MirrorWeb

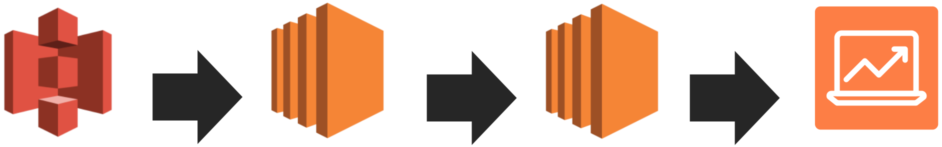
Full text search

PROBLEM

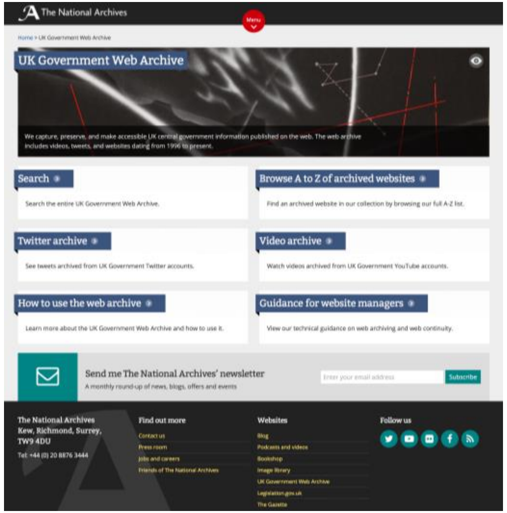
Make the UK Government and UK Parliament’s web archives searchable

Large scale ingestion scenario: 120 TB of data (1.2 MM 100-MB files), duplicates and bad data, Warc format

SOLUTION



Amazon S3 (Storage) → Amazon EC2 (Filtering) → Amazon EC2 (Extraction) → Amazon ES (Search)



The National Archives

Home > UK Government Web Archive

UK Government Web Archive

We capture, preserve, and make accessible UK central government information published on the web. The web archive includes videos, tweets, and websites dating from 1996 to present.

Search

Search the entire UK Government Web Archive.

Browse A to Z of archived websites

Find an archived website in our collection by browsing our full A-Z list.

Twitter archive

See tweets archived from UK Government Twitter accounts.

Video archive

Watch videos archived from UK Government YouTube accounts.

How to use the web archive

Learn more about the UK Government Web Archive and how to use it.

Guidance for website managers

View our technical guidance on web archiving and web continuity.

Send me The National Archives’ newsletter

A monthly round-up of news, blogs, offers and events

Enter your email address

The National Archives
Kew, Richmond, Surrey,
TW9 4DU

Tel: +44 (0) 20 8876 3444

Find out more

- Contact us
- Press room
- Jobs and careers
- Friends of The National Archives

Websites

- Blog
- Podcasts and videos
- Bookshop
- Image library
- UK Government Web Archive
- Legislation.gov.uk
- The Gazette

Follow us

BENEFITS

Scalability: Started on a 9-node, R4.4Xlarge cluster for fast ingest, reduced to 6 R4.Xlarge instances for search. Able to reconfigure the cluster with no down time

Cost effective: Indexed 1.4 billion documents for $337

Fast: 146 MM docs per hour indexed. 14x faster than the previous best for this data set (using Hadoop)

For more on this case, see http://tinyurl.com/ybqwbolq

Case study: Financial Times

Business and Clickstream Analytics

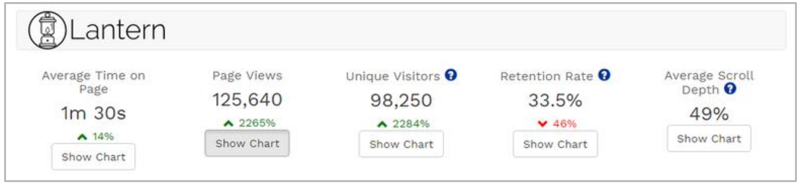
PROBLEM

What stories do our readers care about? What's hot?

Required a custom clickstream analytics solution

Need a solution that delivers analytics in real time

Did not have a team to manage analytics infrastructure



Lantern

Average Time on Page	Page Views	Unique Visitors	Retention Rate	Average Scroll Depth
1m 30s	125,640	98,250	33.5%	49%
▲ 14%	▲ 2265%	▲ 2284%	▼ 46%	
Show Chart	Show Chart	Show Chart	Show Chart	Show Chart

SOLUTION

Streaming user data to Amazon ES for analysis. Created their own custom dashboards for editors and journalists – Lantern.

Lantern - "shines a light" on reader activity for the editors and journalists at the FT

Critical tool for making editorial decisions. Daily editorial meetings start by looking at Lantern dashboard

BENEFITS

Reliability : Lantern is used throughout the day by journalists and editors. Relying on Amazon to manage their systems for maximum uptime.

Cost savings: Able to easily tune their cluster to meet their needs with minimal management overhead

aws SUMMIT

Amazon Elasticsearch Service

Benefits of Amazon Elasticsearch Service

Supports open-source APIs and tools

Drop-in replacement with no need to learn new APIs or skills

Easy to use

Deploy a production-ready Elasticsearch cluster in minutes

Scalable

Resize your cluster with a few clicks or a single API call

Secure

Deploy into your VPC and restrict access using security groups and IAM policies

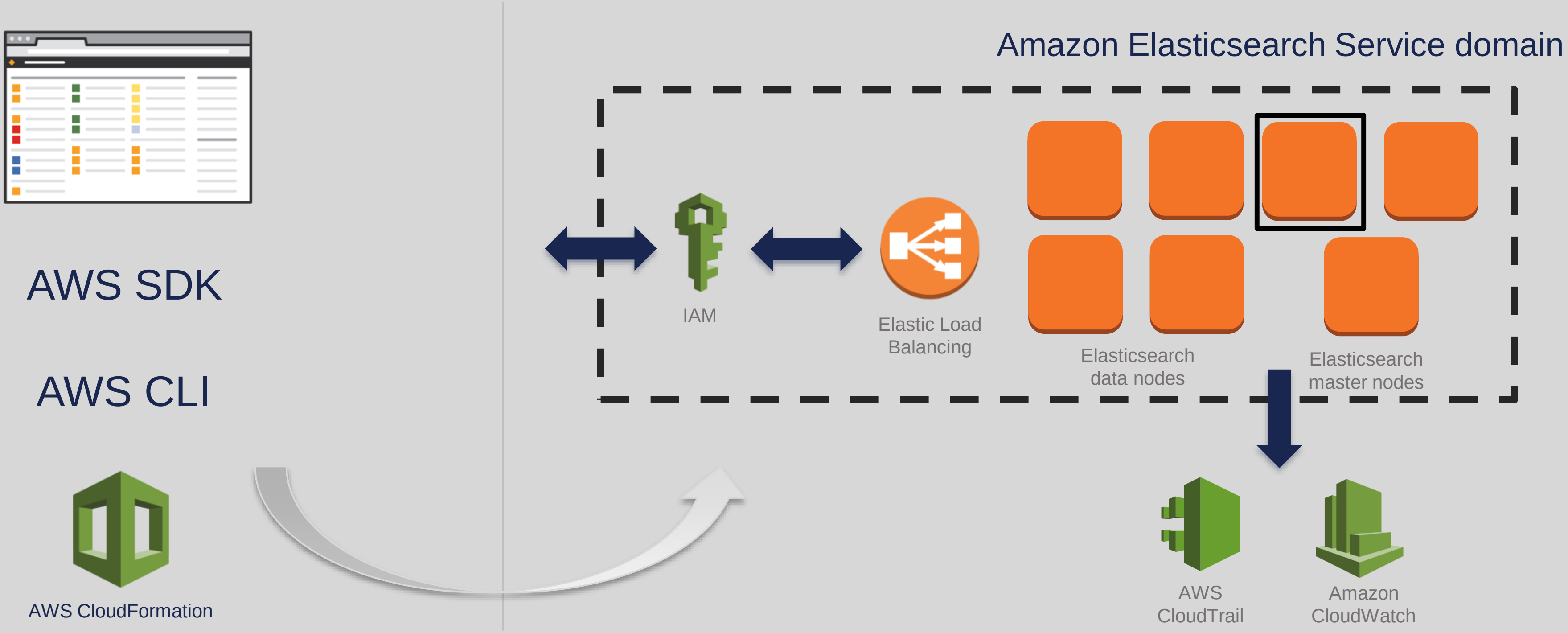
Highly available

Replicate across Availability Zones, with monitoring and automated self-healing

Tightly integrated with other AWS services

Seamless data ingestion, security, auditing and orchestration

Service architecture



AWS SDK

AWS CLI

AWS CloudFormation

Amazon Elasticsearch Service domain

IAM

Elastic Load Balancing

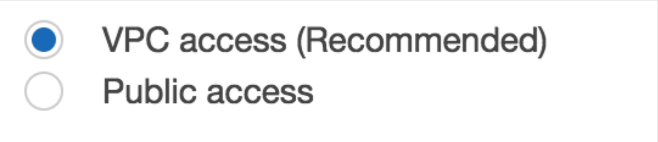
Elasticsearch data nodes

Elasticsearch master nodes

AWS CloudTrail

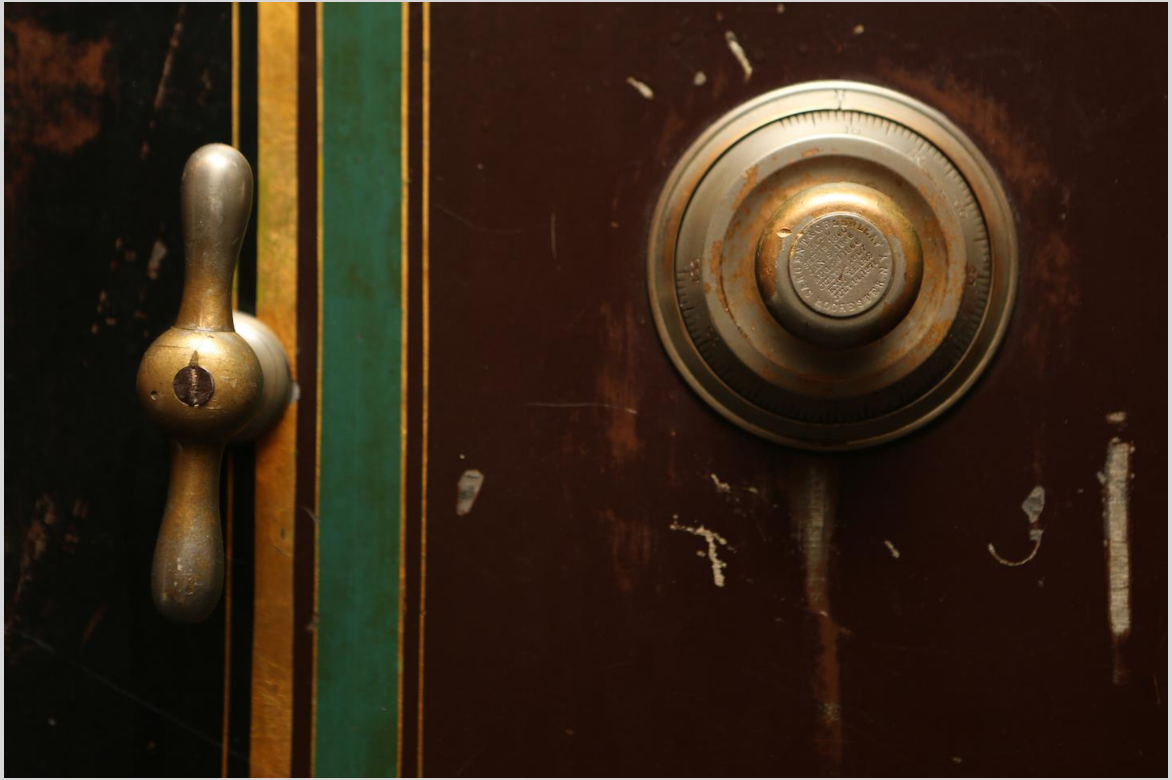
Amazon CloudWatch

Security



● VPC access (Recommended)

○ Public access

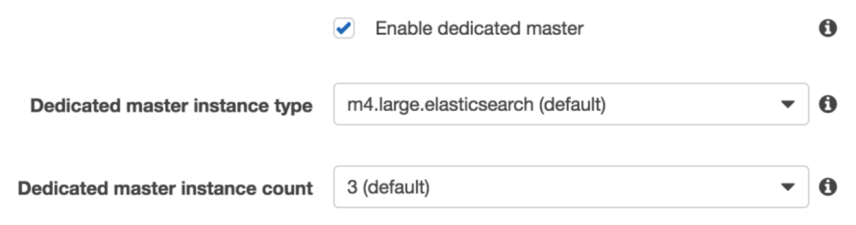


Public endpoints – IAM

Private endpoints – IAM and security groups

Encryption

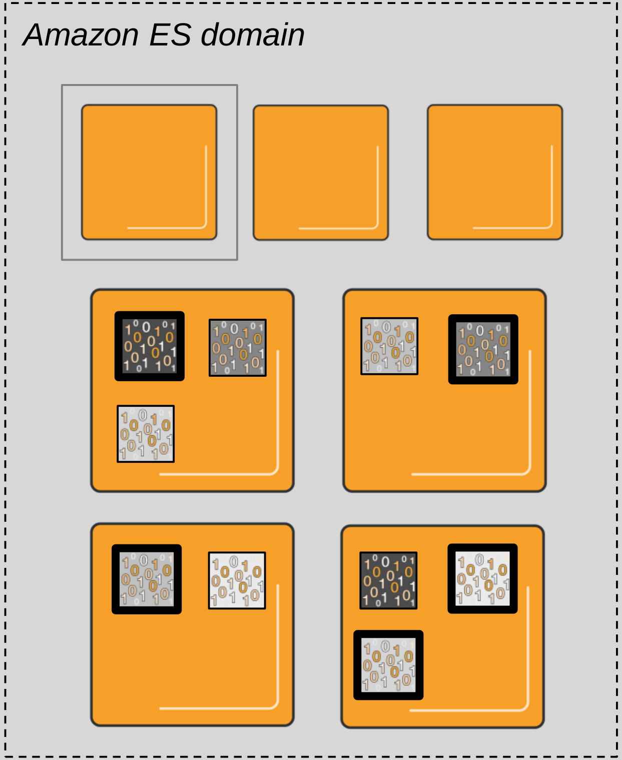
Use three dedicated master instances in production



☑ Enable dedicated master

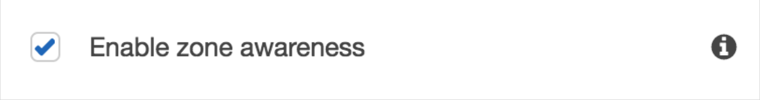
Dedicated master instance type: m4.large.elasticsearch (default)

Dedicated master instance count: 3 (default)



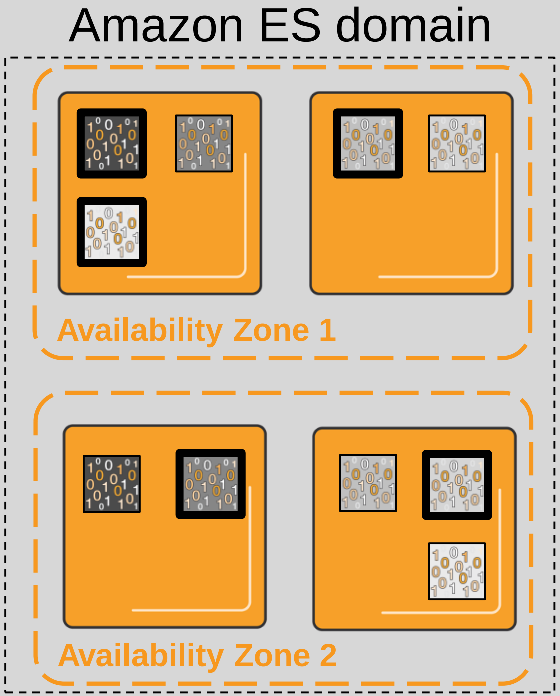
Master instances orchestrate and make your cluster more stable

Use zone awareness in production



☑ Enable zone awareness

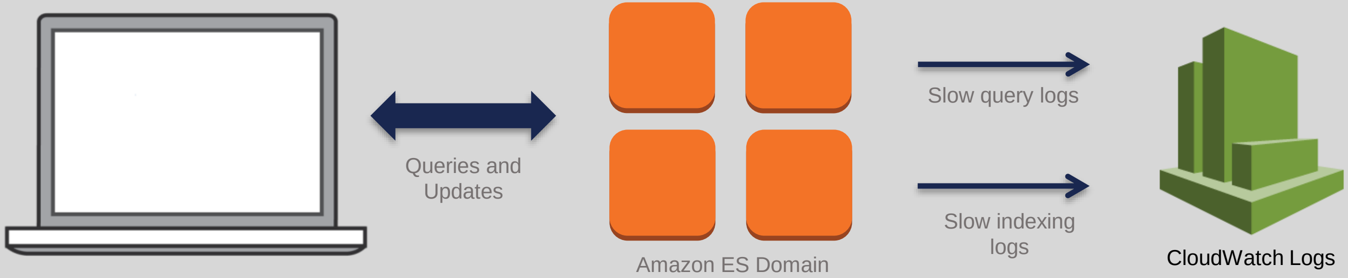
100% data redundancy in two zones makes your cluster more highly available



Set CloudWatch metrics and alarms

Name	Metric	Threshold	Periods
ClusterStatus.red	Maximum	>= 1	1
ClusterIndexWritesBlocked	Maximum	>= 1	1
CPUUtilization/MasterCPUUtilization	Average	>= 80%	3
JVMMemoryPressure/Master...	Maximum	>= 80%	3
FreeStorageSpace	Minimum	<= (25% of avail space)	1
AutomatedSnapshotFailure	Maximum	>= 1	1

Monitor Elasticsearch slow logs



- Easy console setup
- Integrated with CloudWatch Logs
- Set thresholds to receive log events corresponding to slow queries and slow indexing

```
• index.search.slowlog.threshold.query.warn
• index.search.slowlog.threshold.query.info
• index.search.slowlog.threshold.query.debug
• index.search.slowlog.threshold.query.trace

• index.search.slowlog.threshold.fetch.warn
• index.search.slowlog.threshold.fetch.info
• index.search.slowlog.threshold.fetch.debug
• index.search.slowlog.threshold.fetch.trace

• index.indexing.slowlog.threshold.index.warn
• index.indexing.slowlog.threshold.index.info
• index.indexing.slowlog.threshold.index.debug
• index.indexing.slowlog.threshold.index.trace

• index.indexing.slowlog.level: trace
• index.indexing.slowlog.source: 255
```

Pay only for what you use

Instance hours

For data and master instances

EBS GB/Mo

For volumes deployed

AWS data transfer

For transfer out

Amazon Elasticsearch Service usage

@ Expedia

Kuldeep Chowhan

Principal Engineer
@ Expedia Group, Inc.

@this_is_kuldeep

+175 Amazon ES clusters

+500 EC2 instances

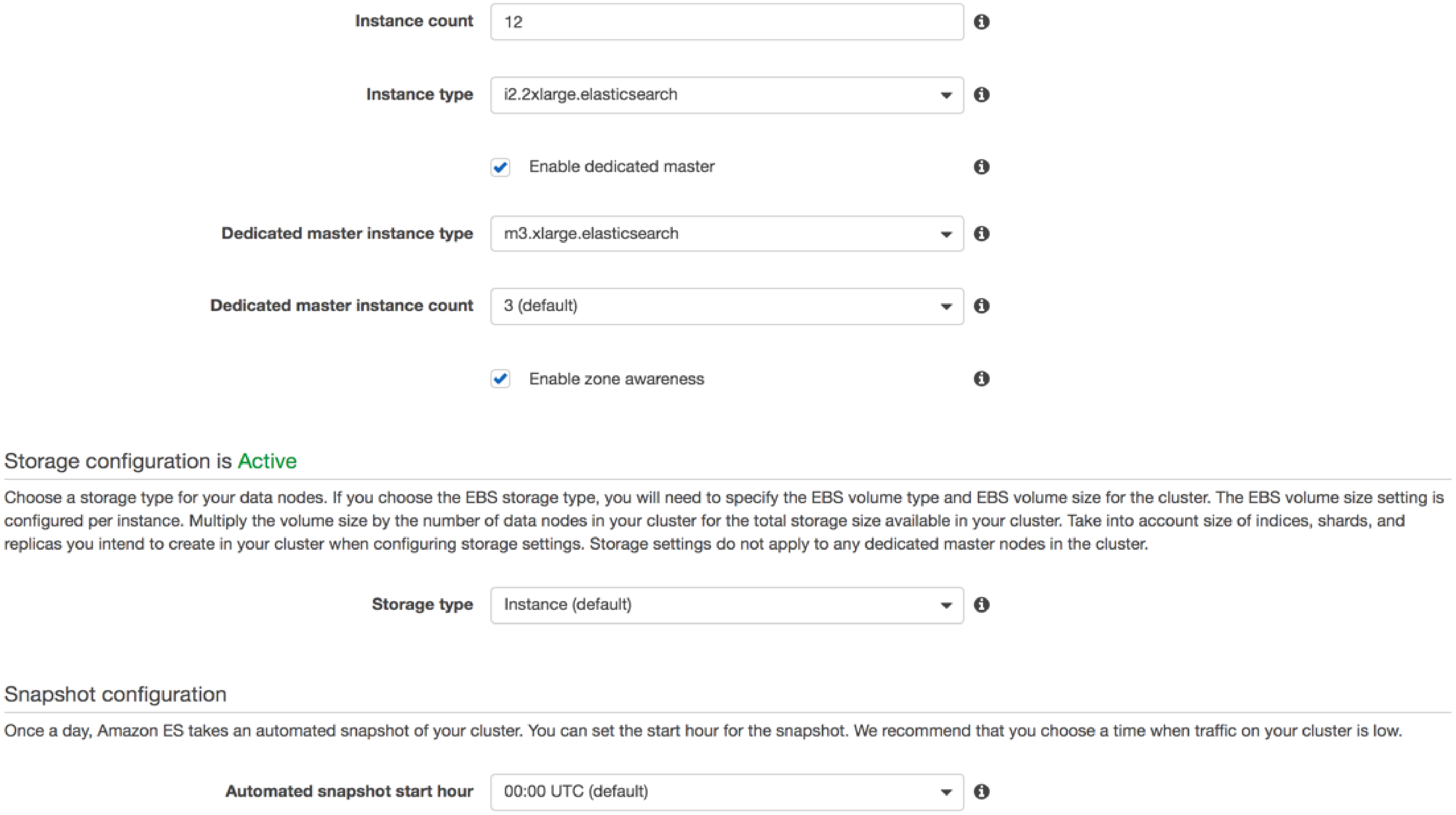
+40B documents

+35 TB of data

Why did we choose Amazon Elasticsearch Service?

Easy to set up

Set up for high availability



Instance count 12

Instance type i2.2xlarge.elasticsearch

☑ Enable dedicated master

Dedicated master instance type m3.xlarge.elasticsearch

Dedicated master instance count 3 (default)

☑ Enable zone awareness

Storage configuration is Active

Choose a storage type for your data nodes. If you choose the EBS storage type, you will need to specify the EBS volume type and EBS volume size for the cluster. The EBS volume size setting is configured per instance. Multiply the volume size by the number of data nodes in your cluster for the total storage size available in your cluster. Take into account size of indices, shards, and replicas you intend to create in your cluster when configuring storage settings. Storage settings do not apply to any dedicated master nodes in the cluster.

Storage type Instance (default)

Snapshot configuration

Once a day, Amazon ES takes an automated snapshot of your cluster. You can set the start hour for the snapshot. We recommend that you choose a time when traffic on your cluster is low.

Automated snapshot start hour 00:00 UTC (default)

Security

Elasticsearch access policy example

```
{
      "Effect": "Allow",
      "Principal": {
            "AWS": "arn:aws:iam::xxxxx:root"
      },
      "Action": "es:*",
      "Resource": “arn:aws:es:us-west-2:xxxxx:domain/xxxxx/*"
},
{
      "Effect": "Allow",
      "Principal": {
            "AWS": "*"
      },
      "Action": "es:Http*",
      "Resource": "arn:aws:es:us-west-2:xxxx:domain/xxxxx/*",
      "Condition": {
            "IpAddress": {
                  "aws:SourceIp": [
                        "0.0.0.0/28"
                  ],
```

Monitoring & backups

Different log analytics architectures using Elasticsearch Service @ Expedia

Different Log Analytics Architectures

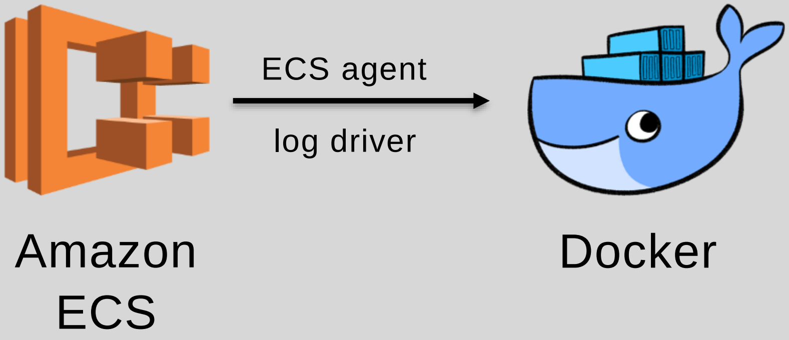
- Docker startup logs to Elasticsearch
- CloudTrail log analytics using Elasticsearch Service
- Distributed tracing platform using Elasticsearch Service

Docker start-up logs to Elasticsearch

Docker startup logs to Elasticsearch

Amazon
ECS

Docker startup logs to Elasticsearch



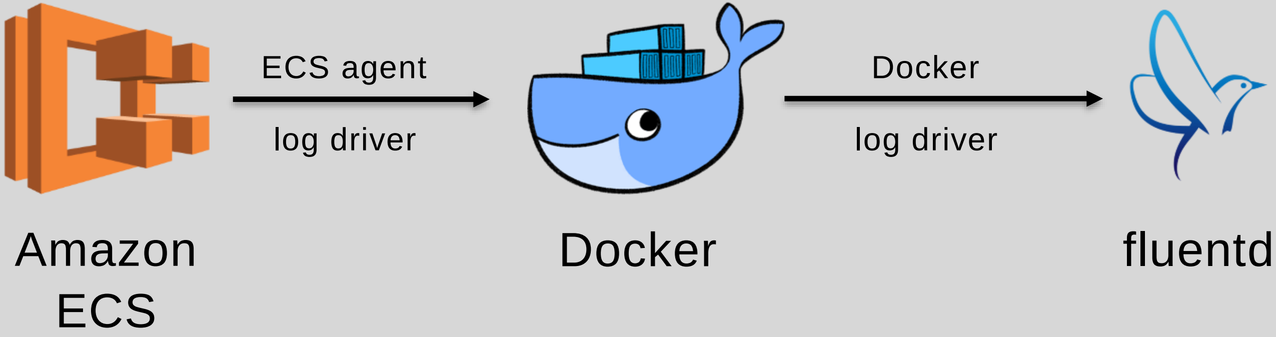
ECS agent

log driver

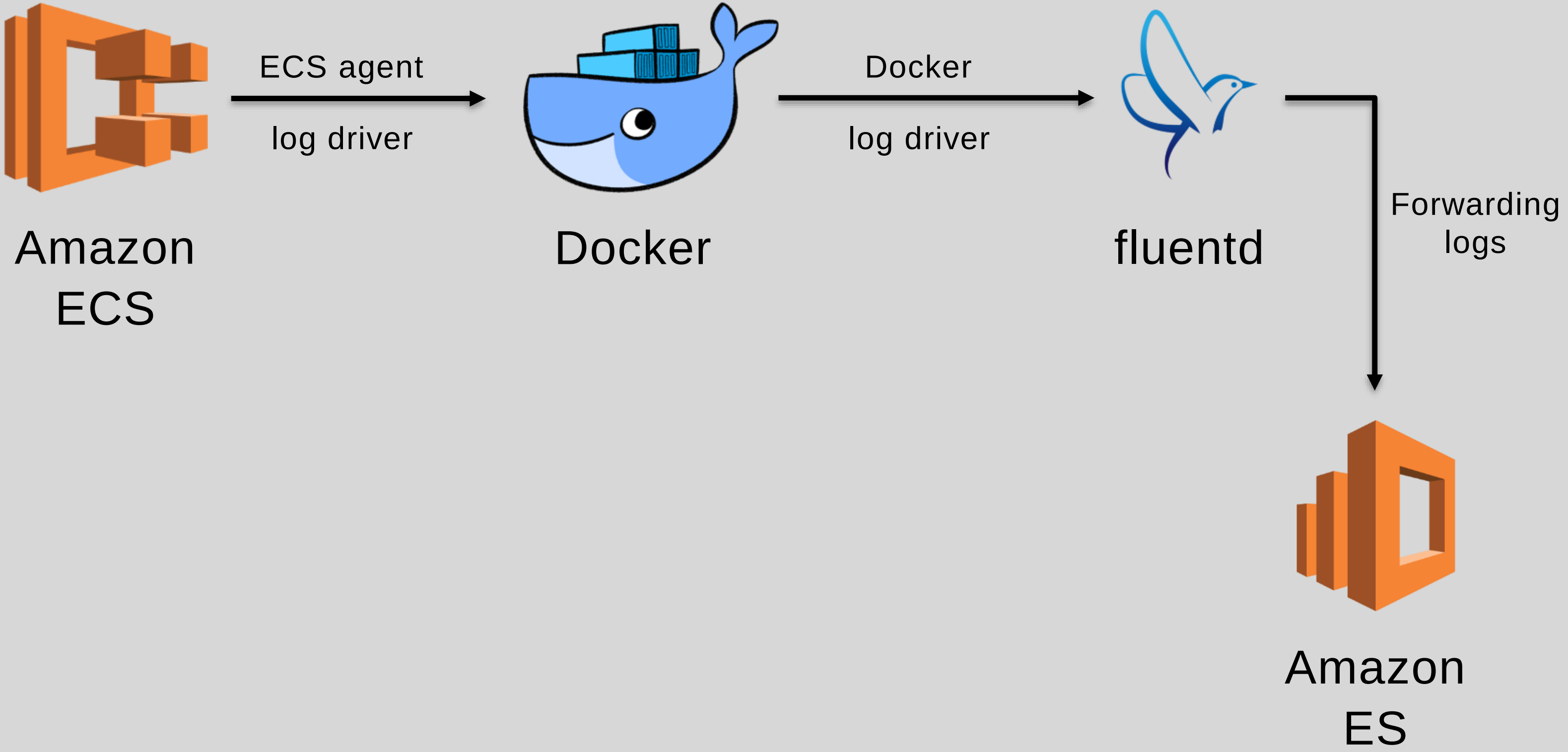
Amazon ECS

Docker

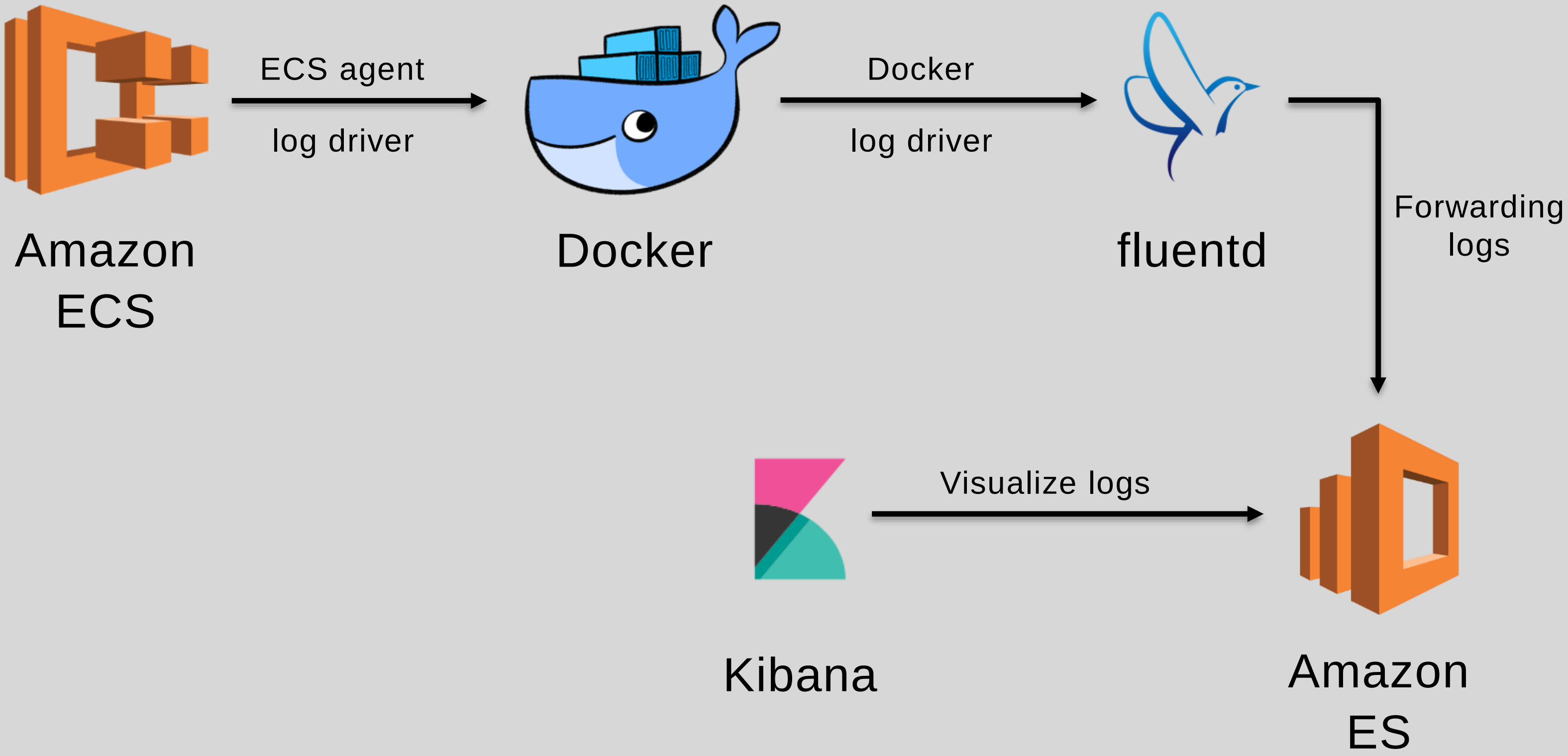
Docker startup logs to Elasticsearch



Docker startup logs to Elasticsearch



Docker startup logs to Elasticsearch



Docker fluentd log_driver configuration

```
{
    "log_driver": "fluentd",
    "options": {
        "fluentd-address": “<fluentd>:24224",
        "tag": "#{ImageName}"
    }
}
```

fluentd configuration to receive Docker logs

```
<source>
   @type forward
   port 24224
   bind 0.0.0.0
</source>
<match *.**>
   @type copy
```

fluentd to ES configuration

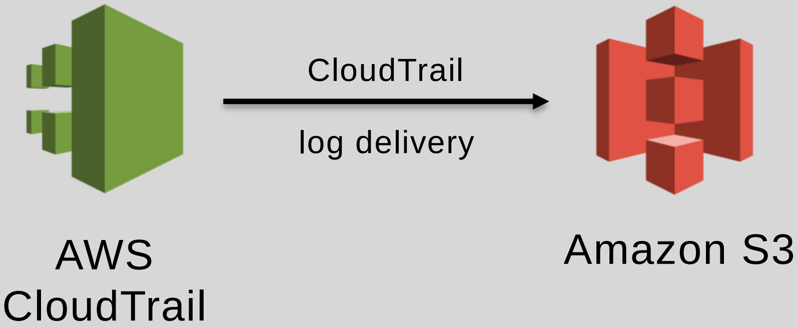
```
<match *.**>
   @type copy
   <store>
      @type elasticsearch
      host <elasticsearch domain>
      include_tag_key true
      tag_key @log_name
      flush_interval 1s
   </store>
```

CloudTrail log analytics using Elasticsearch Service

CloudTrail log analytics using Elasticsearch

AWS
CloudTrail

CloudTrail log analytics using Elasticsearch

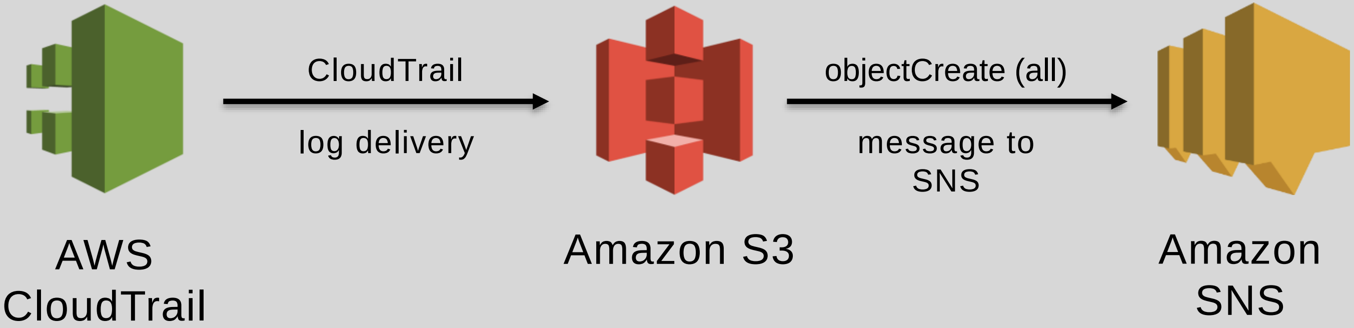


AWS CloudTrail

CloudTrail log delivery

Amazon S3

CloudTrail log analytics using Elasticsearch



AWS CloudTrail

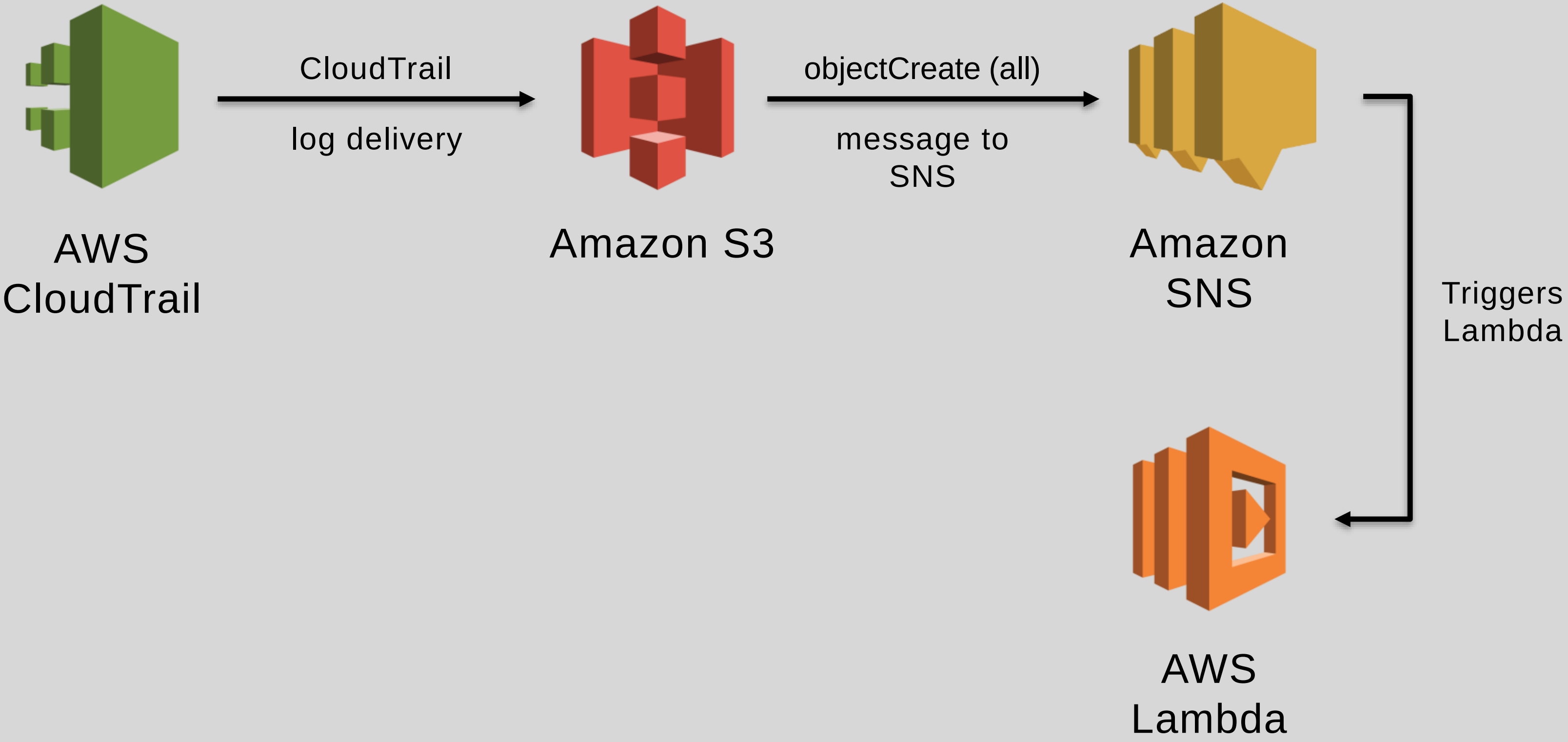
CloudTrail log delivery

Amazon S3

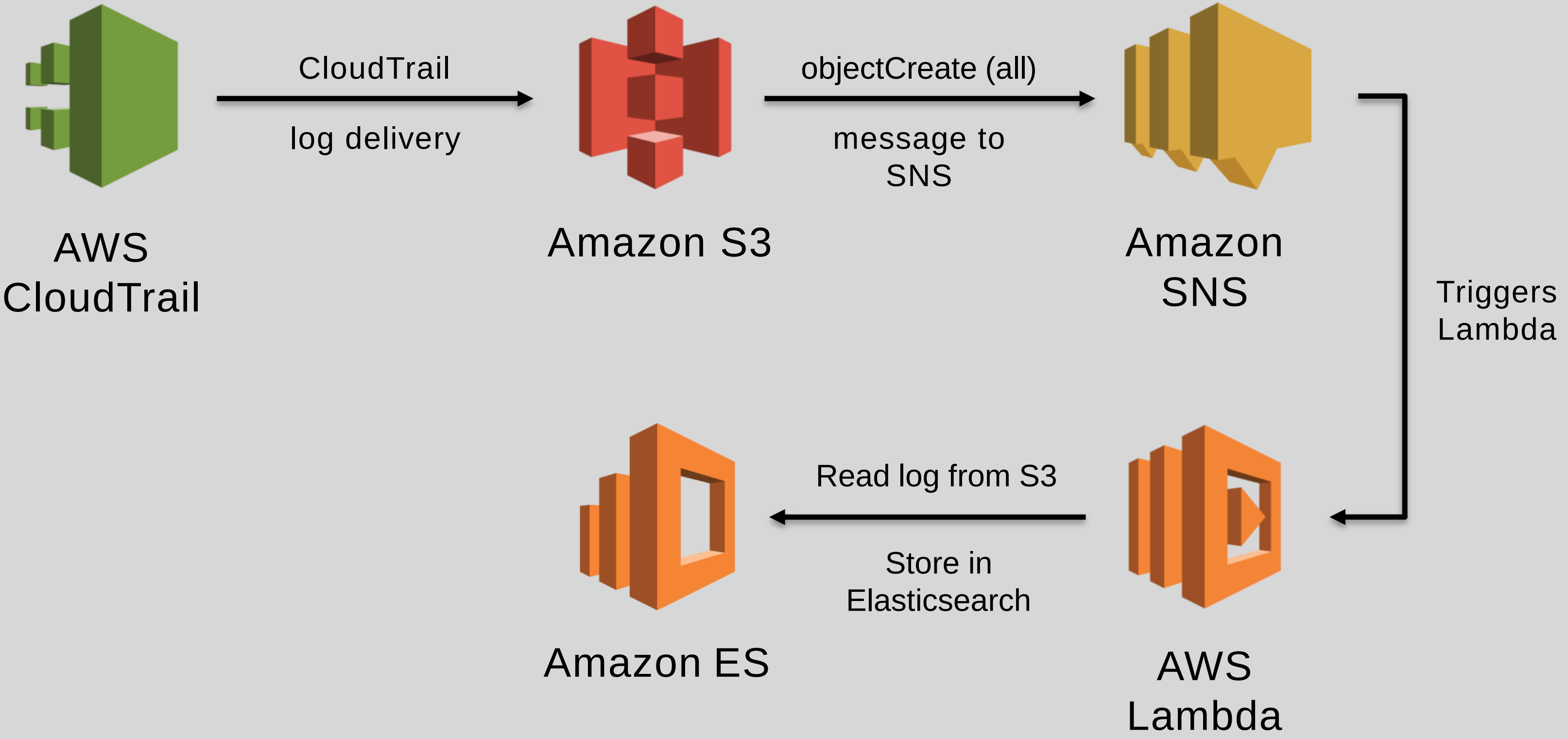
objectCreate (all) message to SNS

Amazon SNS

CloudTrail log analytics using Elasticsearch



CloudTrail log analytics using Elasticsearch



AWS CloudTrail

CloudTrail log delivery

Amazon S3

objectCreate (all) message to SNS

Amazon SNS

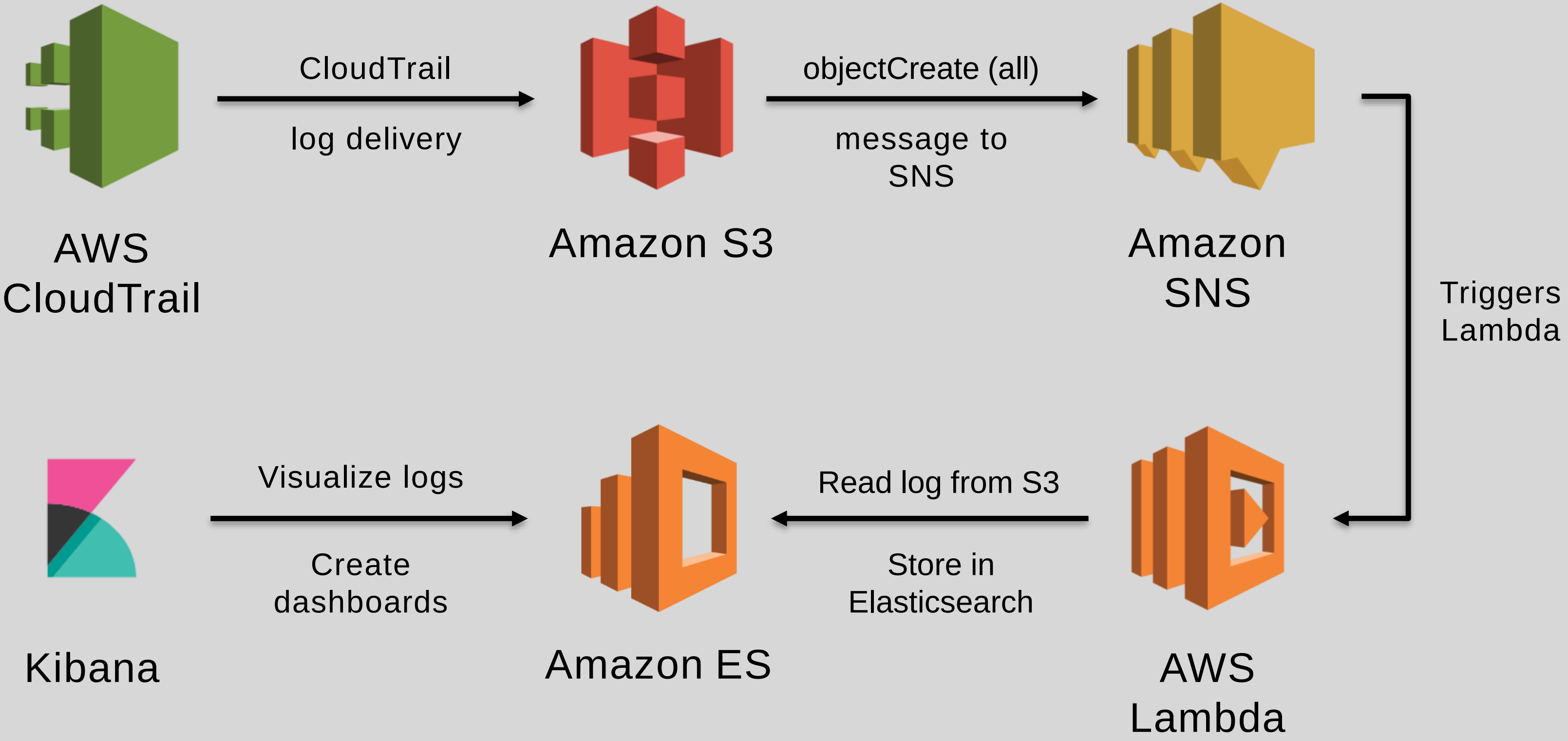
Triggers Lambda

AWS Lambda

Read log from S3 Store in Elasticsearch

Amazon ES

CloudTrail log analytics using Elasticsearch



AWS CloudTrail

CloudTrail log delivery

Amazon S3

objectCreate (all) message to SNS

Amazon SNS

Triggers Lambda

AWS Lambda

Read log from S3

Store in Elasticsearch

Amazon ES

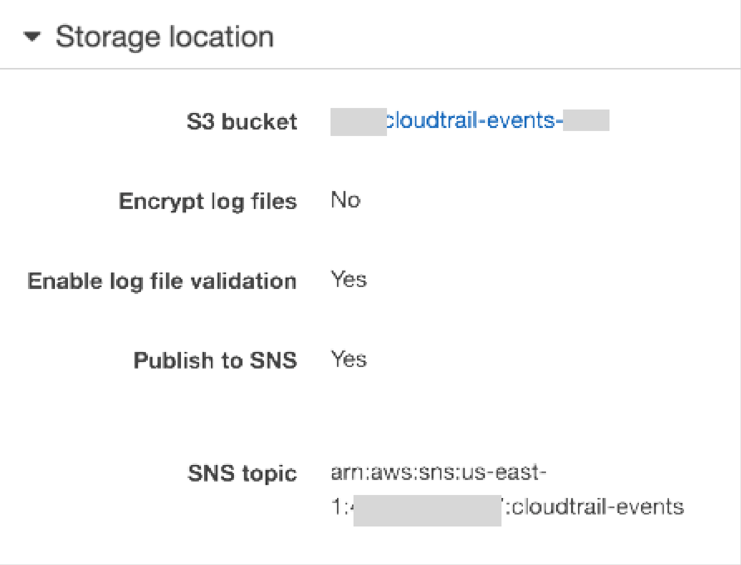
Visualize logs

Create dashboards

Kibana

CloudTrail log analytics using Elasticsearch

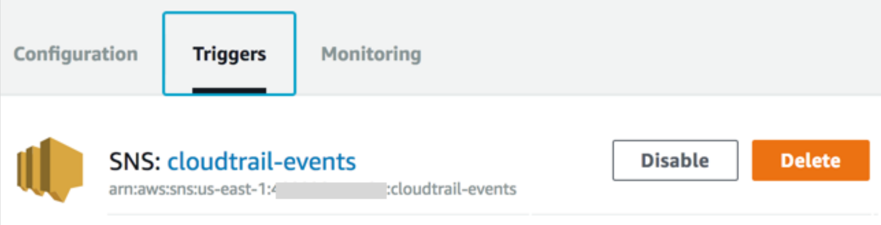
CloudTrail to S3 and SNS



▾ Storage location

S3 bucket	[redacted]cloudtrail-events-[redacted]
Encrypt log files	No
Enable log file validation	Yes
Publish to SNS	Yes
SNS topic	arn:aws:sns:us-east-1:[redacted]:cloudtrail-events

SNS to Lambda trigger



SNS: cloudtrail-events

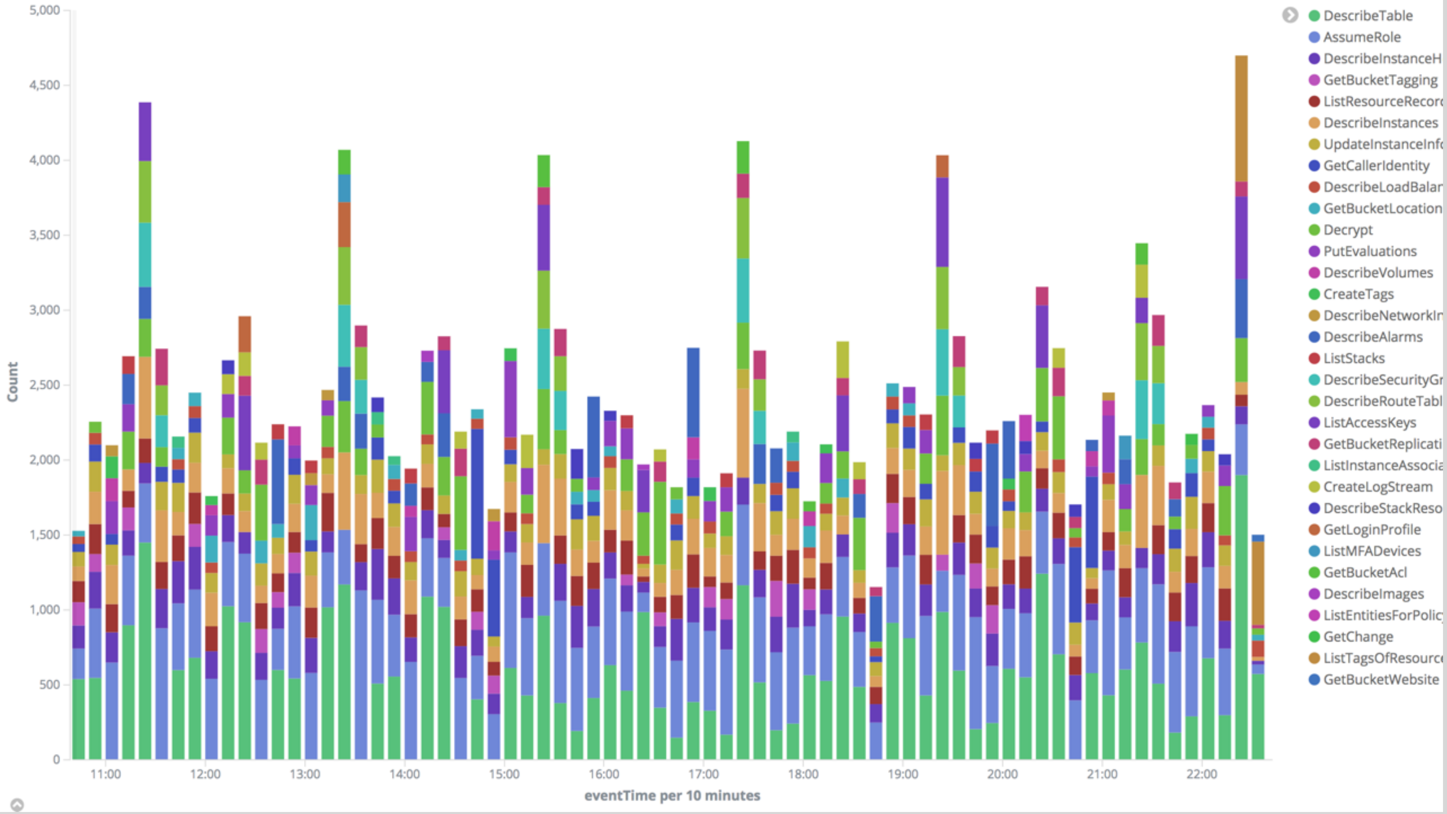
arn:aws:sns:us-east-1:[redacted]:cloudtrail-events

CloudTrail logs from S3 to Elasticsearch

```
try:
    response = s3.get_object(Bucket=s3Bucket, Key=s3ObjectKey)
    content =
gzip.GzipFile(fileobj=StringIO(response['Body'].read())).read()
for record in json.loads(content)['Records']:
    recordJson = json.dumps(record)
    logger.info(recordJson)
    indexName = 'ct-' + datetime.datetime.now().strftime("%Y-%m-%d")
    res = es.index(index=indexName, doc_type='record', id=record['eventID'],
body=recordJson)
    logger.info(res)
return True
```

How did we use this CloudTrail log data that is in Elasticsearch?

Top 10 AWS API calls (per 10 mins) dashboard

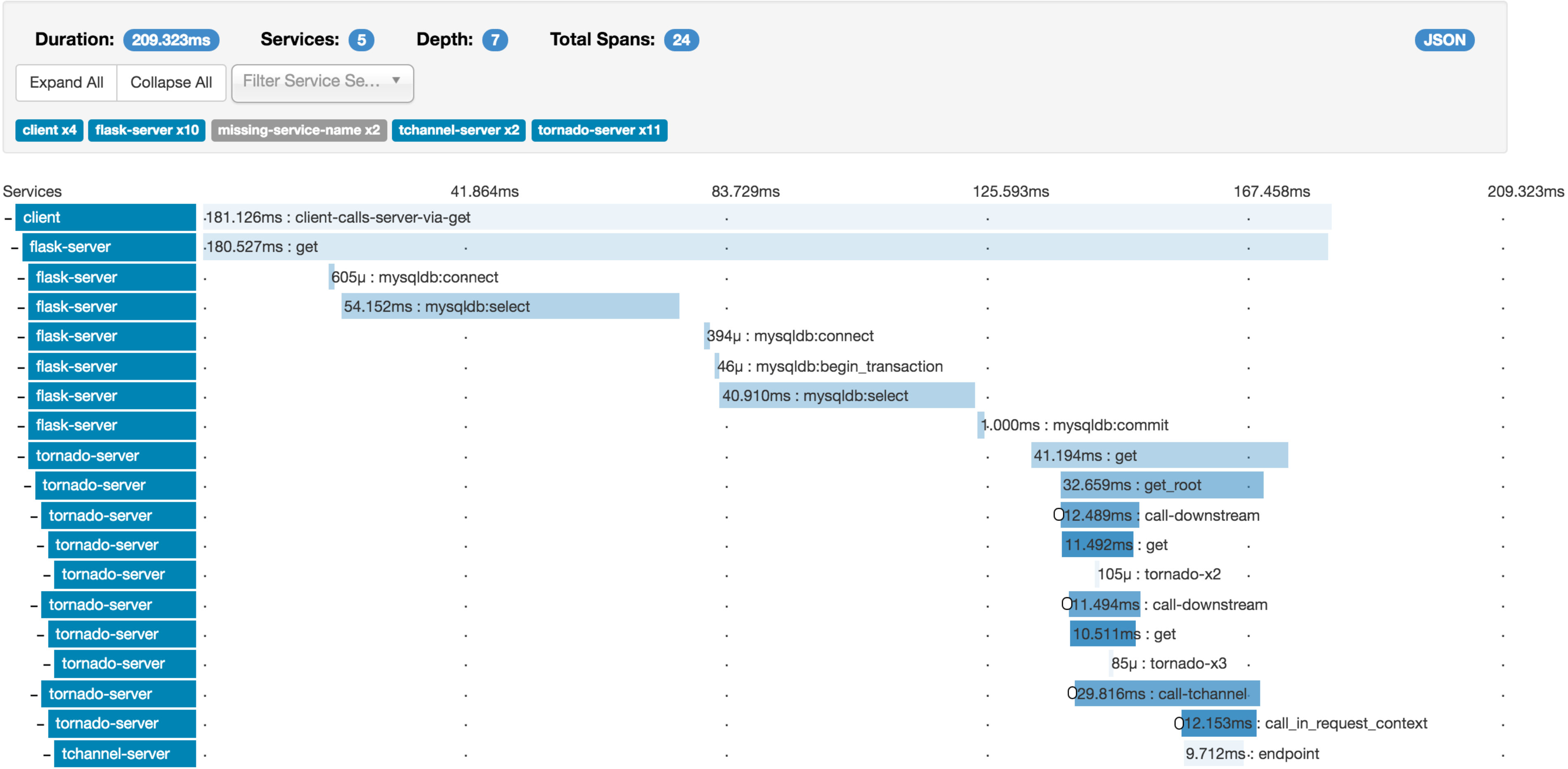


This solution is open sourced at

https://github.com/ExpediaDotCom/cloudtrail-log-analytics

as a serverless application

Distributed tracing platform using Elasticsearch Service



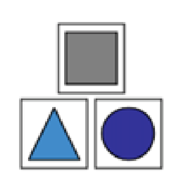
Duration: 209.323ms Services: 5 Depth: 7 Total Spans: 24 JSON

Expand All | Collapse All | Filter Service Se... ▼

client x4 | flask-server x10 | missing-service-name x2 | tchannel-server x2 | tornado-server x11

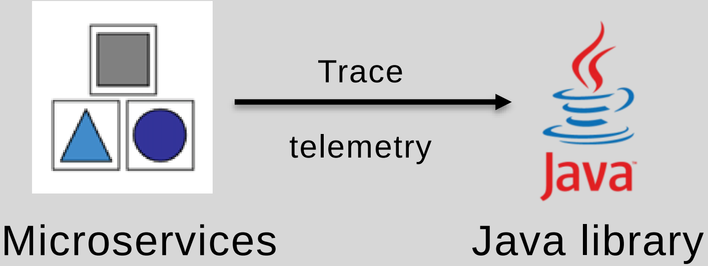
```
Services            41.864ms   83.729ms   125.593ms   167.458ms   209.323ms
– client                    181.126ms : client-calls-server-via-get
 – flask-server             180.527ms : get
  – flask-server            605μ : mysqldb:connect
  – flask-server            54.152ms : mysqldb:select
  – flask-server            394μ : mysqldb:connect
  – flask-server            46μ : mysqldb:begin_transaction
  – flask-server            40.910ms : mysqldb:select
  – flask-server            1.000ms : mysqldb:commit
  – tornado-server          41.194ms : get
   – tornado-server         32.659ms : get_root
    – tornado-server        12.489ms : call-downstream
     – tornado-server       11.492ms : get
      – tornado-server      105μ : tornado-x2
    – tornado-server        11.494ms : call-downstream
     – tornado-server       10.511ms : get
      – tornado-server      85μ : tornado-x3
    – tornado-server        29.816ms : call-tchannel
     – tornado-server       12.153ms : call_in_request_context
      – tchannel-server     9.712ms : endpoint
```

Distributed tracing platform using Elasticsearch



Microservices

Distributed tracing platform using Elasticsearch



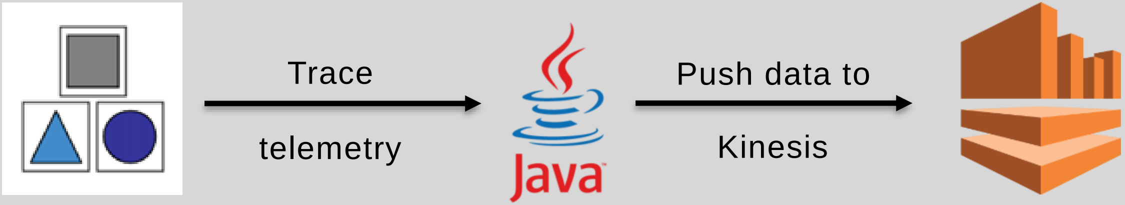
Trace

telemetry

Microservices

Java library

Distributed tracing platform using Elasticsearch



Trace

telemetry

Push data to

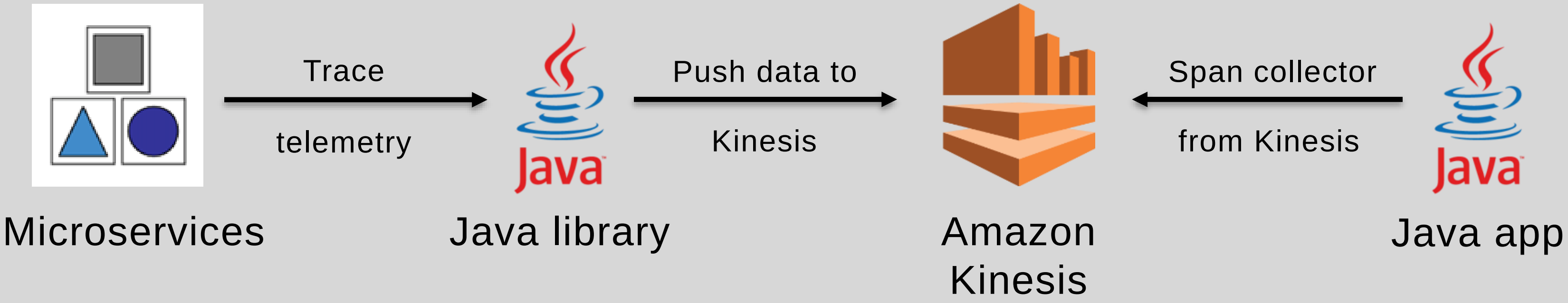
Kinesis

Microservices

Java library

Amazon Kinesis

Distributed tracing platform using Elasticsearch



Microservices

Trace telemetry

Java library

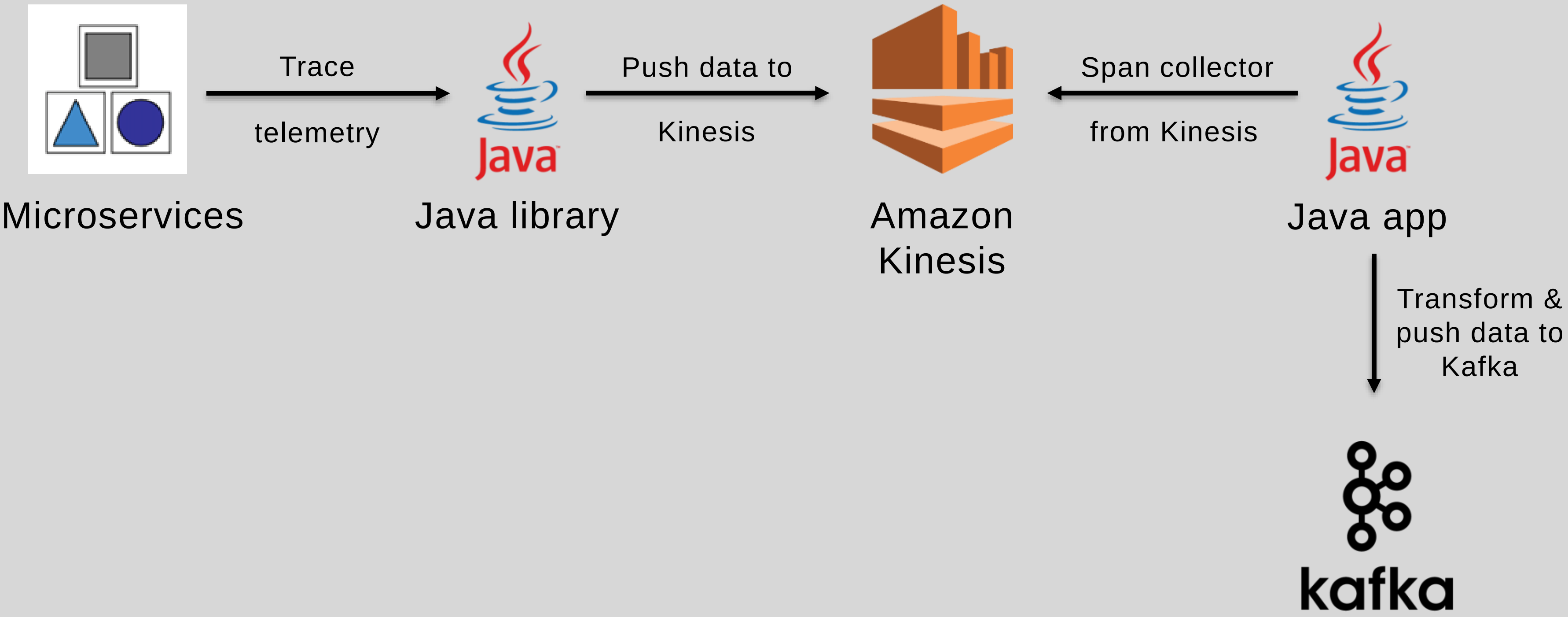
Push data to Kinesis

Amazon Kinesis

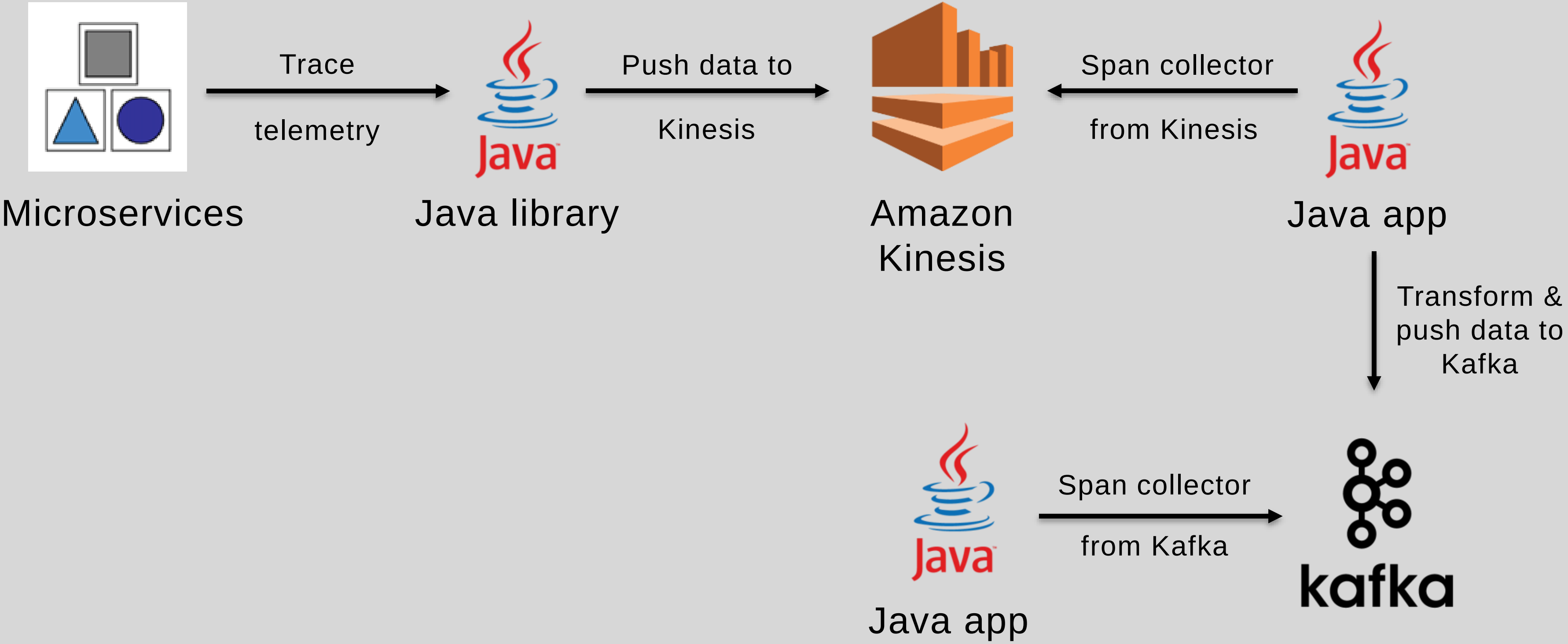
Span collector from Kinesis

Java app

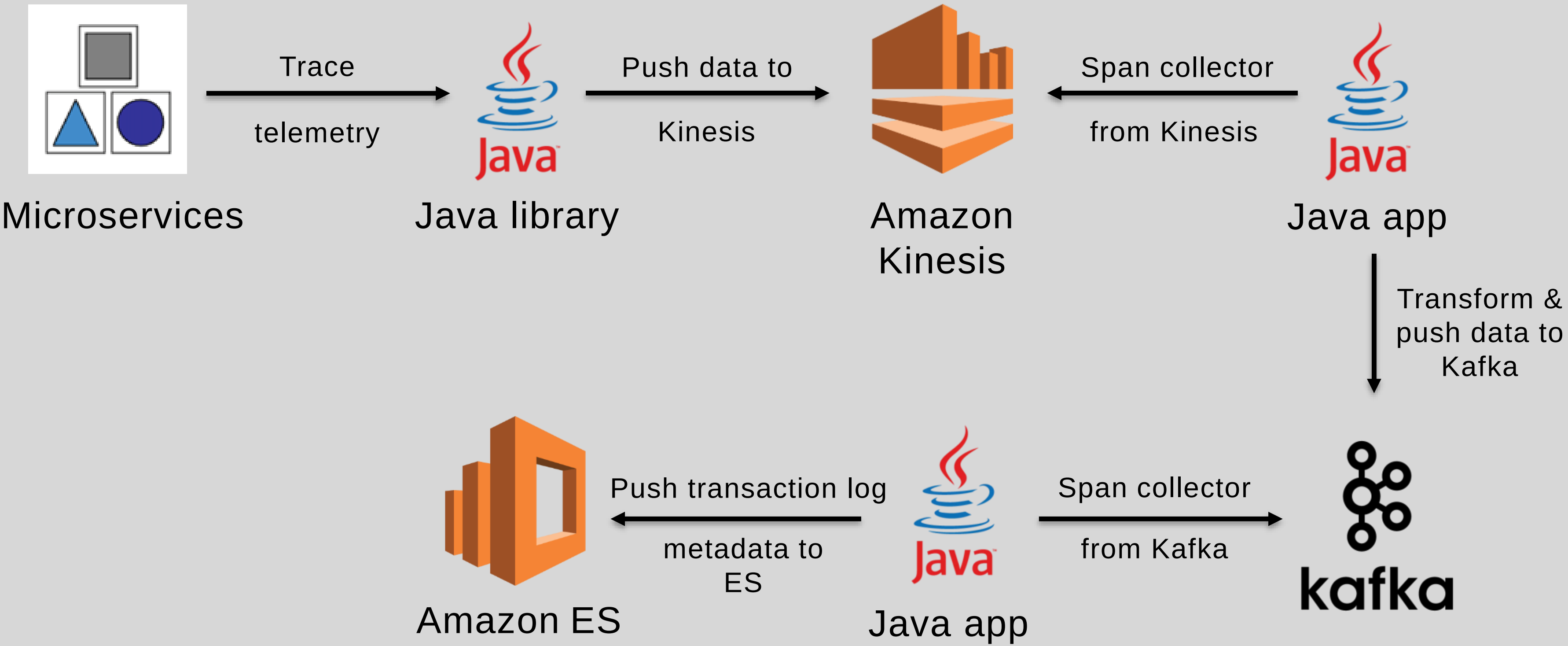
Distributed tracing platform using Elasticsearch



Distributed tracing platform using Elasticsearch



Distributed tracing platform using Elasticsearch



Microservices

Trace telemetry

Java library

Push data to Kinesis

Amazon Kinesis

Span collector from Kinesis

Java app

Transform & push data to Kafka

kafka

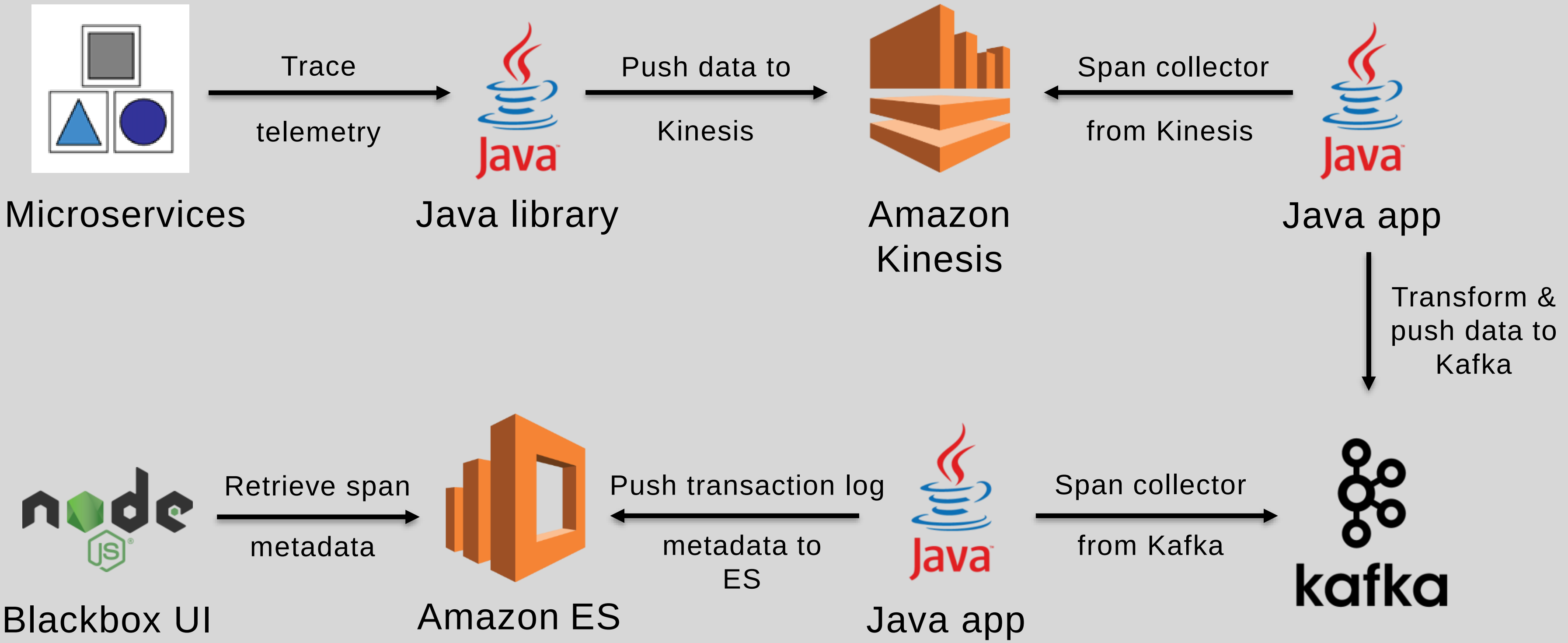
Span collector from Kafka

Java app

Push transaction log metadata to ES

Amazon ES

Distributed tracing platform using Elasticsearch



Haystack document in Elasticsearch

```
{
      "_index": "transactions-2017-10-26-04",
      "_type": "transactions_logs",
      "_id": "AV9W9lCq5Jdrc3Uj0vCg",
      "_score": null,
      "_source": {
            "transactionid": "5e66cad8-d7ea-49e8-94c8-24d2298d4cdc"
      },
      "fields": {
            "startTime": [
                  1508992503395
            ]
      },
      "sort": [
            1508992503395
      ]
}
```

How is Elasticsearch used in distributed tracing?

Time-based queries for traces

Filtering traces by services

Things to keep in mind

- Scaling of cluster results in a new cluster with the data being synchronized
- Monitor and optimize the cluster yourself
- No upgrade button between Elasticsearch versions (wish list)
- Monitoring doesn’t show how much disk space in use (wish list)

Wrap up

Benefits of using Amazon Elasticsearch Service:

- Easy to set up
- Setup for high availability
- Security
- Monitoring and backup

Please complete the session survey in the summit mobile app.

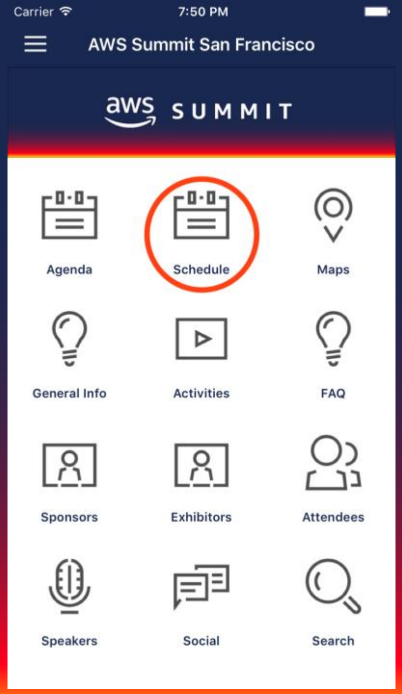
aws SUMMIT

Thank you!

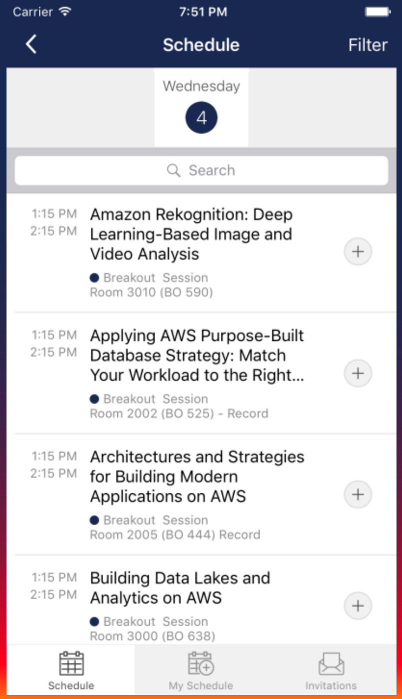
@this_is_kuldeep

Submit Session Feedback

1. Tap the **Schedule** icon.



2. Select the session you attended.



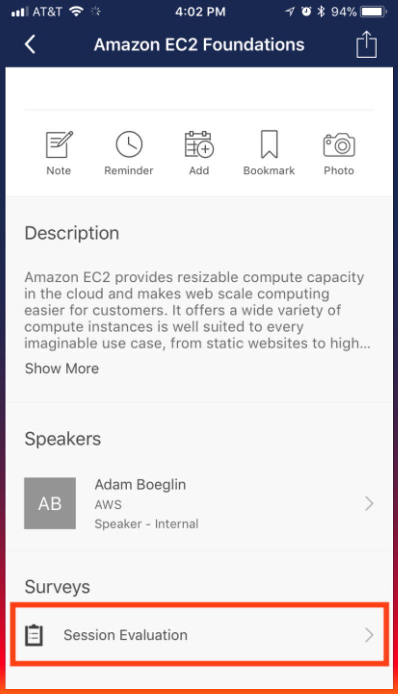
1:15 PM
2:15 PM
Amazon Rekognition: Deep Learning-Based Image and Video Analysis
● Breakout Session
Room 3010 (BO 590)

1:15 PM
2:15 PM
Applying AWS Purpose-Built Database Strategy: Match Your Workload to the Right...
● Breakout Session
Room 2002 (BO 525) - Record

1:15 PM
2:15 PM
Architectures and Strategies for Building Modern Applications on AWS
● Breakout Session
Room 2005 (BO 444) Record

1:15 PM
2:15 PM
Building Data Lakes and Analytics on AWS
● Breakout Session
Room 3000 (BO 638)

3. Tap **Session Evaluation** to submit your feedback.



Description

Amazon EC2 provides resizable compute capacity in the cloud and makes web scale computing easier for customers. It offers a wide variety of compute instances is well suited to every imaginable use case, from static websites to high...

Show More

Speakers

AB
Adam Boeglin
AWS
Speaker - Internal

Surveys

Session Evaluation