AWS re:Invent

ANT329-R

Amazon ES ingest at Pearson—managing high throughput and scale

Josh Pavel
Senior DevSecOps Engineer
Pearson

Jon Handler
Principal Solutions Architect
Amazon Web Services

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

```
2.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
mp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
20.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085
r.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0
20.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0" 200 4179
r.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0
r.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0
12.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html HTTP/1.0" 200 3985
aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
4.144.152 - - [01/Jul/1995:00:00:13 -0400] "GET / HTTP/1.0" 200 7074
mp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
mp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
mp6.unicomp.net - - [01/Jul/1995:00:00:14 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
aa.net - - [01/Jul/1995:00:00:15 -0400] "GET /images/KSC-logosmall.gif HTTP/1.0" 200 1204
4.144.152 - - [01/Jul/1995:00:00:17 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 304 0
20.110.21 - - [01/Jul/1995:00:00:17 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1713
y391.asahi-net.or.jp - - [01/Jul/1995:00:00:18 -0400] "GET /facts/about_ksc.html HTTP/1.0" 200 3977
-141.eden.com - - [01/Jul/1995:00:00:19 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0916.jpg HTTP/1.0" 200 34
y391.asahi-net.or.jp - - [01/Jul/1995:00:00:19 -0400] "GET /images/launchpalms-small.gif HTTP/1.0" 200 11473
89.154.54 - - [01/Jul/1995:00:00:24 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
s-gw.starway.net.au - - [01/Jul/1995:00:00:25 -0400] "GET /shuttle/missions/51-l/mission-51-l.html HTTP/1.0" 200 6723
ia-30.shadow.net - - [01/Jul/1995:00:00:27 -0400] "GET / HTTP/1.0" 200 7074
89.154.54 - - [01/Jul/1995:00:00:29 -0400] "GET /shuttle/countdown/count.gif HTTP/1.0" 200 40310
a.prodigy.com - - [01/Jul/1995:00:00:33 -0400] "GET /shuttle/missions/sts-71/sts-71-patch-small.gif HTTP/1.0" 200 1205
ia-30.shadow.net - - [01/Jul/1995:00:00:35 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 5866
2.lloyd.com - - [01/Jul/1995:00:00:37 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0613.jpg HTTP/1.0" 200 6171
-pc.moorecap.com - - [01/Jul/1995:00:00:38 -0400] "GET /history/apollo/apollo-13/images/70HC314.GIF HTTP/1.0" 200 1012
89.154.54 - - [01/Jul/1995:00:00:40 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 200 786
l2-01.ix.netcom.com - - [01/Jul/1995:00:00:41 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
```

What is Elasticsearch?

Elasticsearch, Logstash, and Kibana

Sometimes referred to as the “ELK Stack”

Distributed search and analytics engine

Built on Apache Lucene

Easy ingestion and visualization

Rank			DBMS	Score		
Nov 2019	Oct 2019	Nov 2018		Nov 2019	Oct 2019	Nov 2018
1.	1.	1.	Oracle	1336.07	-19.81	+34.96
2.	2.	2.	MySQL	1266.28	-16.78	+106.39
3.	3.	3.	Microsoft SQL Server	1081.91	-12.81	+30.36
4.	4.	4.	PostgreSQL	491.07	+7.16	+50.83
5.	5.	5.	MongoDB	413.18	+1.09	+43.70
6.	6.	6.	IBM Db2	172.60	+1.83	-7.27
7.	7.	↑ 8.	Elasticsearch	148.40	-1.77	+4.94
8.	8.	↓ 7.	Redis	145.24	+2.32	+1.06
9.	9.	9.	Microsoft Access	130.07	-1.10	-8.36
10.	10.	↑ 11.	Cassandra	123.23	+0.01	+1.48

Source: DB-Engines.com, October 2019

It is a database

1

Send data as
JSON via REST APIs

2

Data is indexed—
all fields searchable,
including nested JSON

3

Queries, via REST APIs,
allow fielded matching,
Boolean expressions,
include sorting and analysis





Amazon Elasticsearch Service (Amazon ES) is a **fully managed service** that makes it easy to deploy, manage, and scale Elasticsearch and Kibana

How our customers use Amazon ES



Gain actionable insights

Application API use, infrastructure uptime, security, etc. Your logs are the basis of your insights.



Find root causes

Search your logs for error messages, server ids, etc., to reveal ongoing issues.



React in real time

With fast ingest, fast time to visualize, flexible dashboarding, Amazon ES gives you the means to react.



Search your data

Provide relevant search results for application data.



Centralize logs

Create a single pane of glass view on your infrastructure and application.



Monitor security

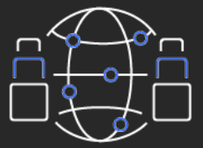
Detect and respond to fraud, denial of service, and other cyber threats.

Benefits of Amazon ES



Fully managed

With APIs and AWS console control, you can deploy your cluster in minutes as a drop-in for an existing cluster.



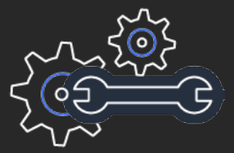
Flexible

Search your data or analyze your logs. Supports AWS and open-source ingestion tools.



Cost-effective

Pay as you go. Reduce operational cost. Right-size your instance types. Use reserved instances.



Highly available

Self-healing, 24x7 monitoring, multi-AZ with a click, automatic backups, AWS support available, metrics in Amazon CloudWatch.



Scalable and high-performance

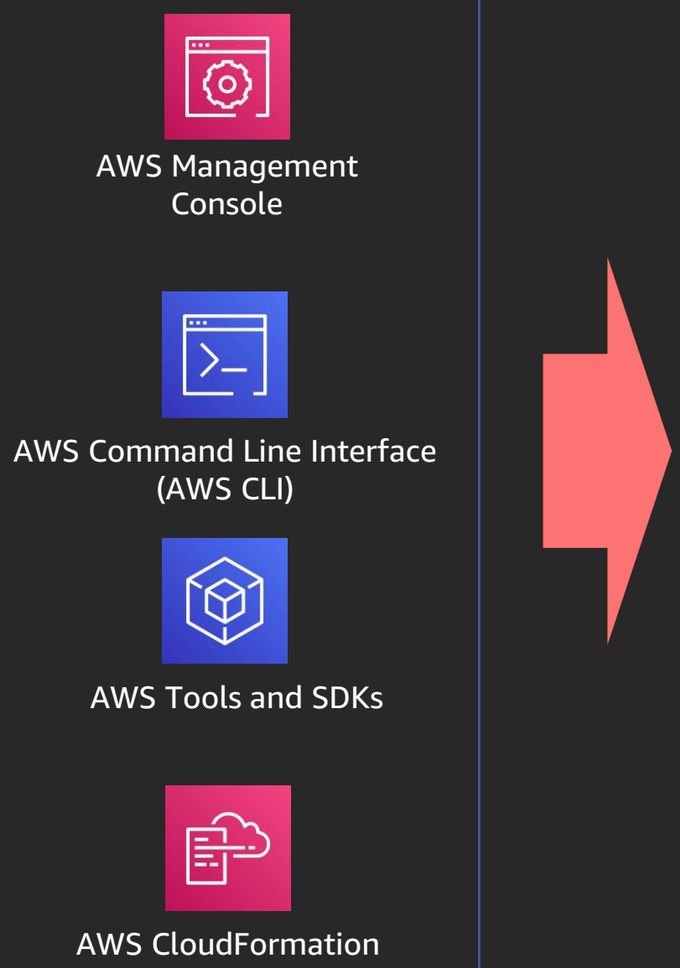
One click to scale, apply Amazon ES version upgrades and patches.

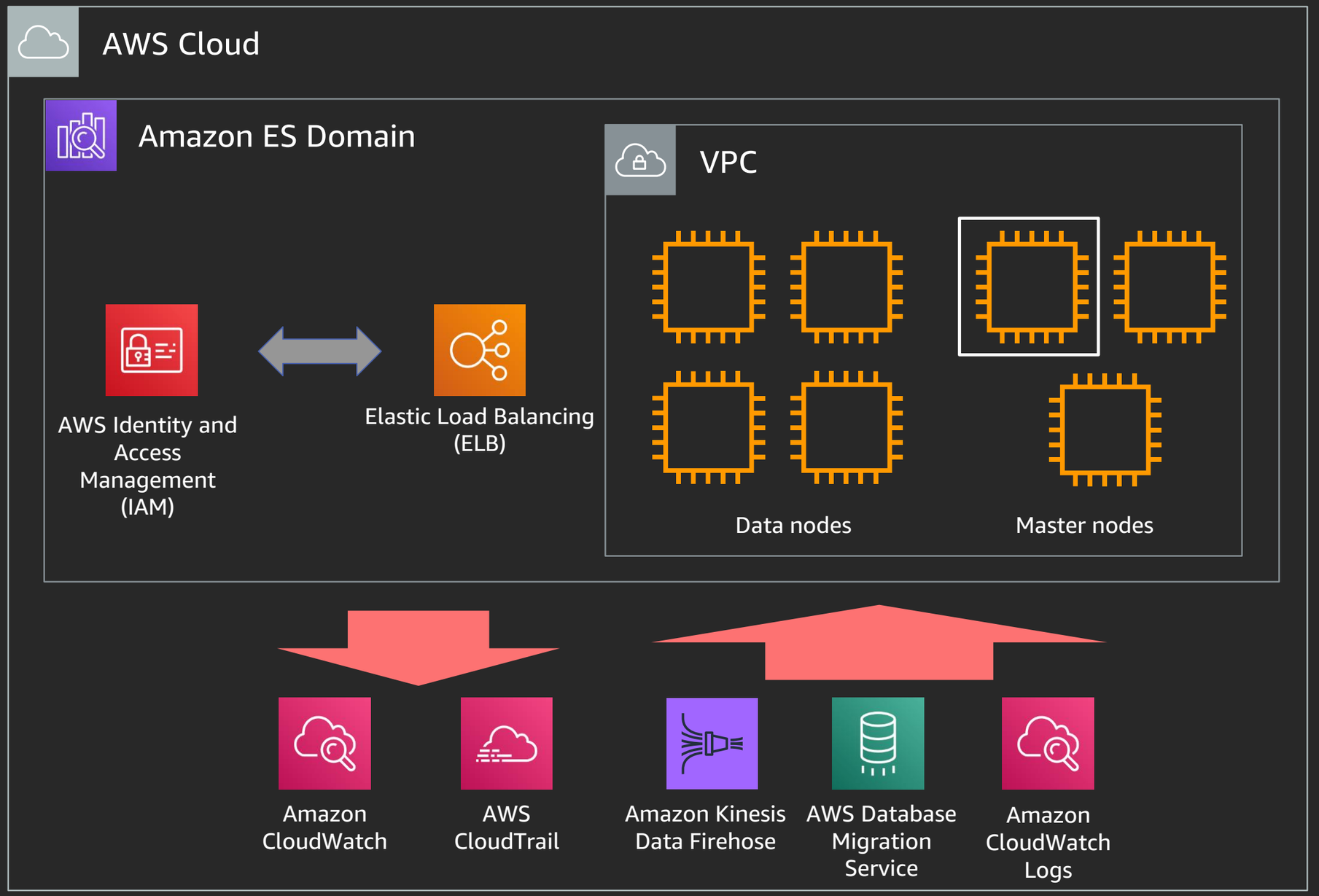


Secure and compliant

Deploy in VPC, Amazon Cognito login, HIPAA, FISMA, SOC, PCI, FedRAMP compliant.

Service architecture





Amazon ES works with structured JSON containing fields and values


```
{
     "verb": "GET",
     "ident": "-",
     "bytes": 6245,
     "@timestamp": "1995-07-
01T00:00:01",
     "request": "GET
/history/apollo/ HTTP/1.0",
     "host": "199.72.81.55",
     "authuser": "-",
     "@timestamp_utc": "1995-
07-01T04:00:01+00:00",
     "timezone": "-0400",
     "response": 200
}
```

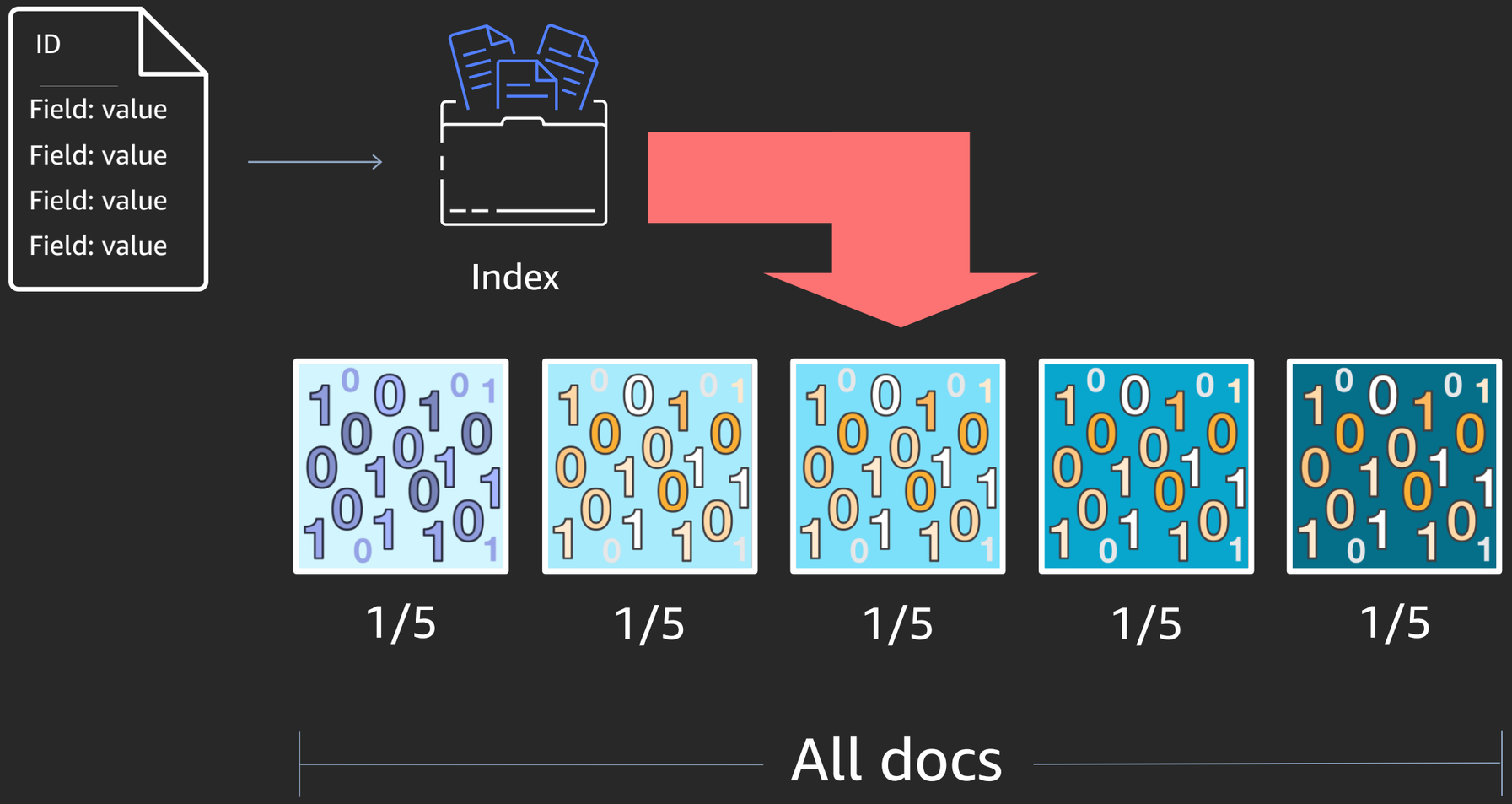
You use the indexing APIs to send data to Amazon ES*

```
POST endpoint/index/type/id
{
     Document
}
```

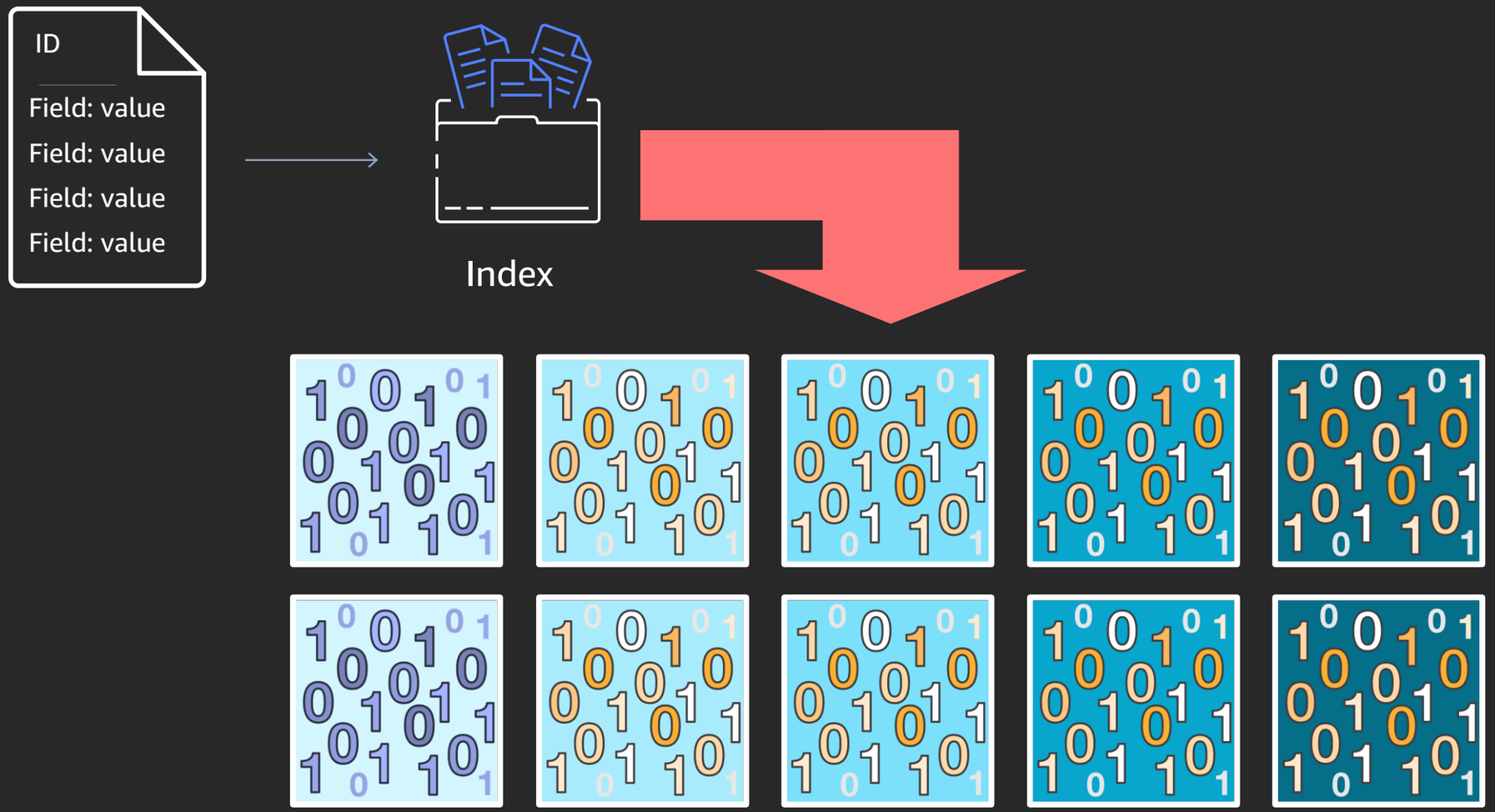
```
POST endpoint/index/_bulk
{ Command }
{ Document }
{ Command }
{ Document }
```

* Your ingestion tools will probably automate this

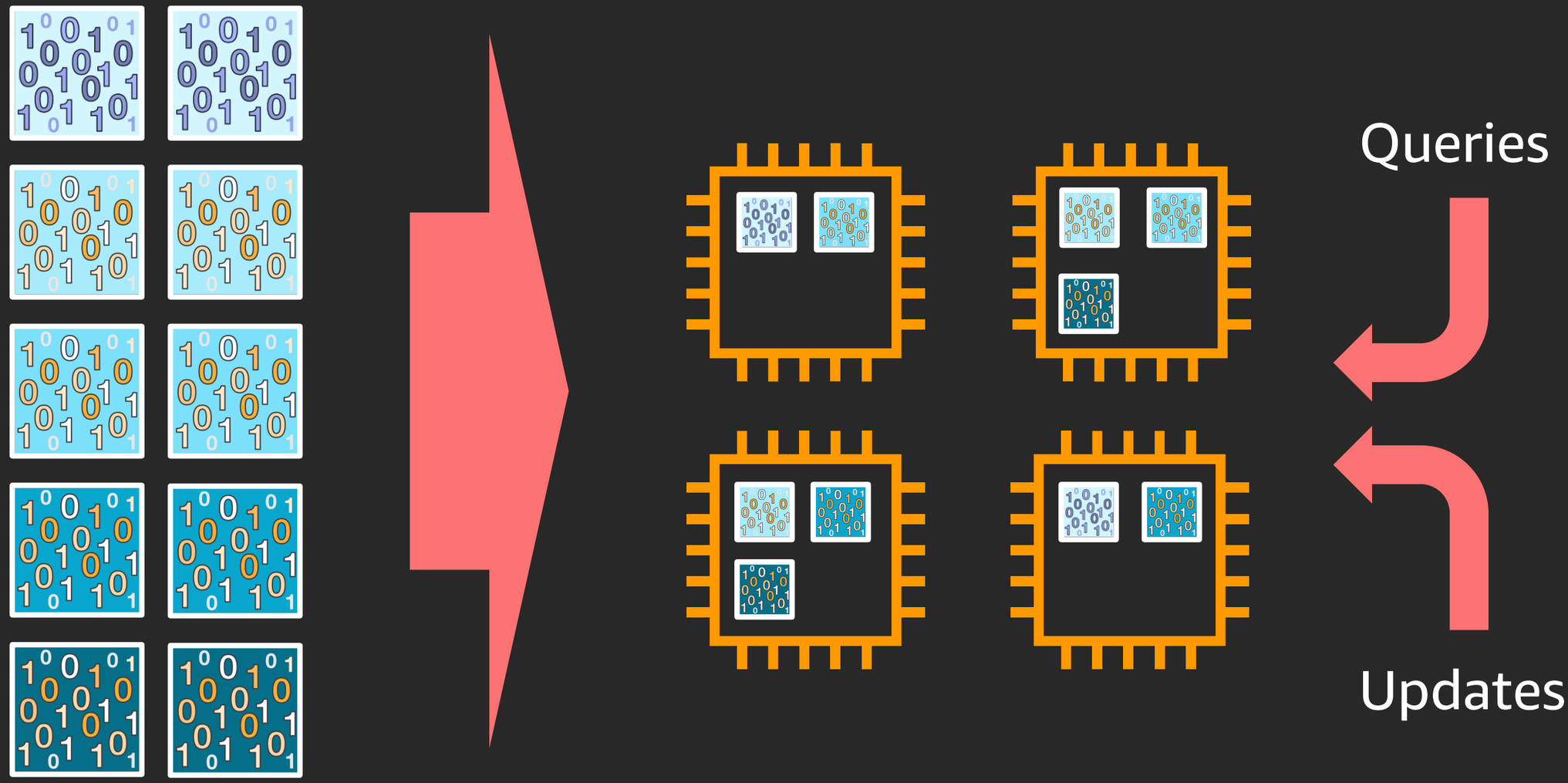
Data is stored in an index comprised of shards



Shards are primary or replica

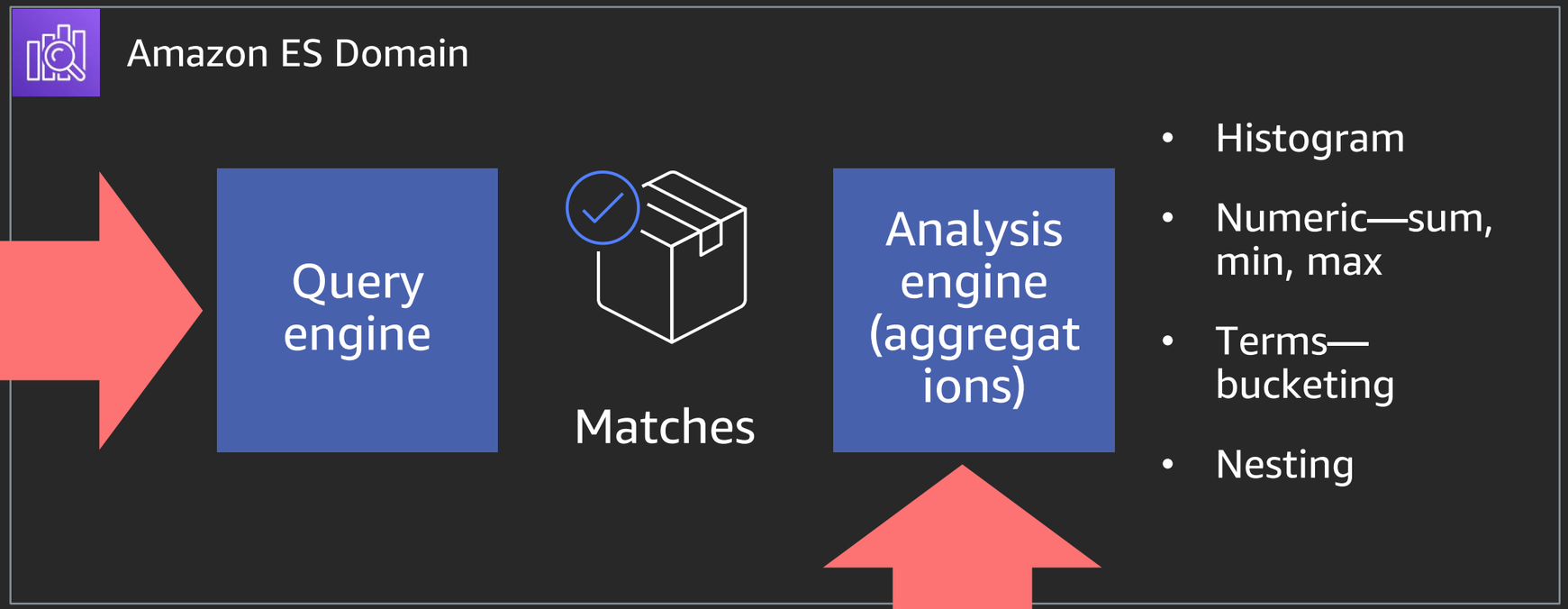


Amazon ES distributes shards to data nodes



You use aggregations to analyze log data

```
GET logs/log/_search
{
  "query": {
    "term": {
      "verb.keyword": {
        "value": "GET"
      }
    }
  }
}
```





Amazon ES analysis with aggregations

Host	Timestamp	Verb	Request	Http	Status	Size
199.72.81.55	[01/Jul/1995:00:00:01	GET	/history/apollo/	HTTP/1.0	200	6245
unicomp6.unicomp.net	[01/Jul/1995:00:00:06	GET	/shuttle/countdown/	HTTP/1.0	200	3985
199.120.110.21	[01/Jul/1995:00:00:09	GET	/shuttle/missions/sts-73/mission-sts-73.html	HTTP/1.0	200	4085
burger.letters.com	[01/Jul/1995:00:00:11	GET	/shuttle/countdown/liftoff.html	HTTP/1.0	304	0
199.120.110.21	[01/Jul/1995:00:00:11	GET	/shuttle/missions/sts-73/sts-73-patch-small.gif	HTTP/1.0	200	4179
burger.letters.com	[01/Jul/1995:00:00:12	GET	/images/NASA-logosmall.gif	HTTP/1.0	304	0
burger.letters.com	[01/Jul/1995:00:00:12	GET	/shuttle/countdown/video/livevideo.gif	HTTP/1.0	200	0
205.212.115.106	[01/Jul/1995:00:00:12	GET	/shuttle/countdown/countdown.html	HTTP/1.0	200	3985
d104.aa.net	[01/Jul/1995:00:00:13	GET	/shuttle/countdown/	HTTP/1.0	200	3985
129.94.144.152	[01/Jul/1995:00:00:13	GET	/	HTTP/1.0	200	7074

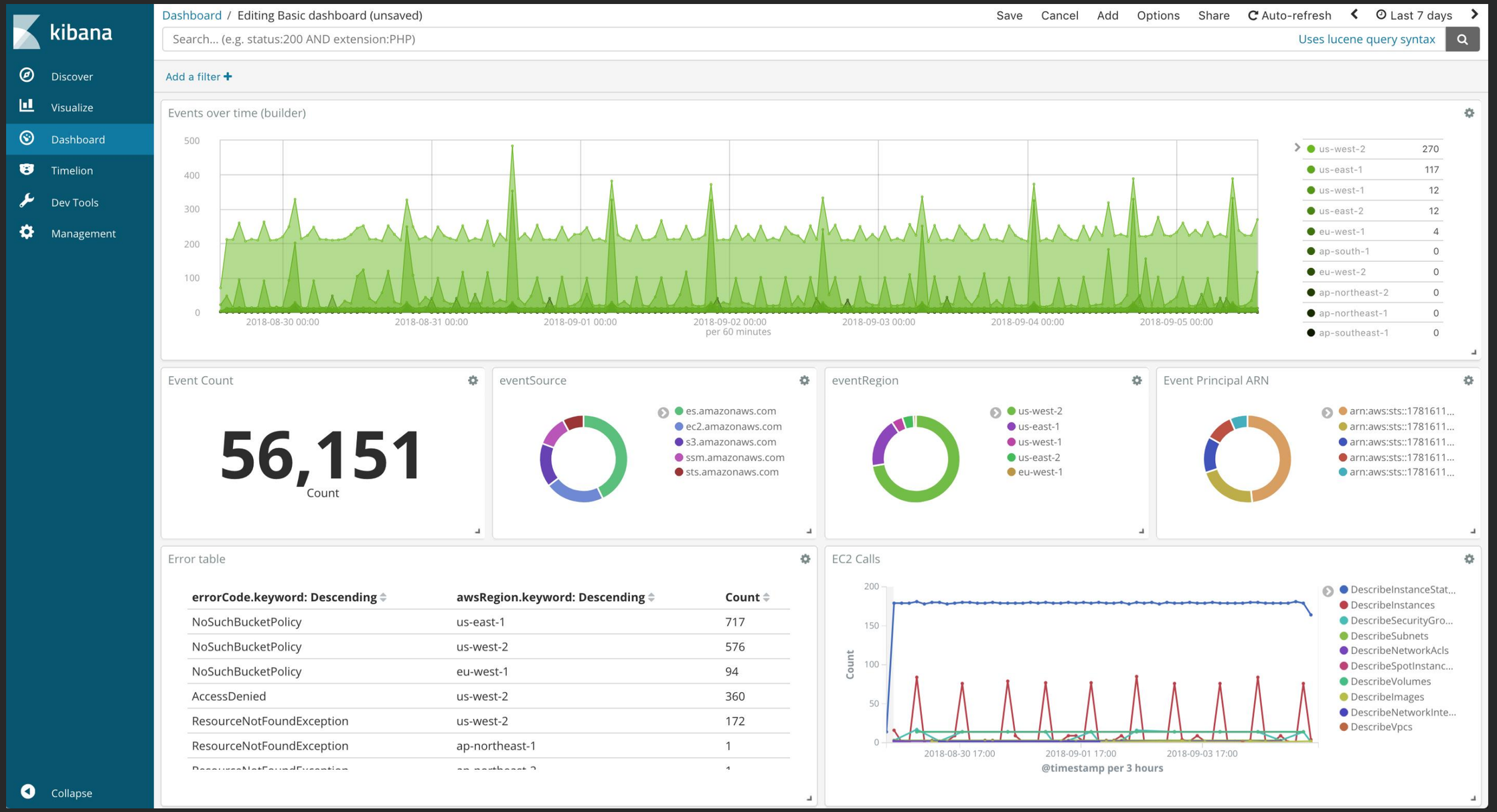
Filter: host="burger*" OR host="199.120*"

Host	Timestamp	Verb	Request	Http	Status	Size
199.72.81.55	[01/Jul/1995:00:00:01	GET	/history/apollo/	HTTP/1.0	200	6245
unicomp6.unicomp.net	[01/Jul/1995:00:00:06	GET	/shuttle/countdown/	HTTP/1.0	200	3985
199.120.110.21	[01/Jul/1995:00:00:09	GET	/shuttle/missions/sts-73/mission-sts-73.html	HTTP/1.0	200	4085
burger.letters.com	[01/Jul/1995:00:00:11	GET	/shuttle/countdown/liftoff.html	HTTP/1.0	304	0
199.120.110.21	[01/Jul/1995:00:00:11	GET	/shuttle/missions/sts-73/sts-73-patch-small.gif	HTTP/1.0	200	4179
burger.letters.com	[01/Jul/1995:00:00:12	GET	/images/NASA-logosmall.gif	HTTP/1.0	304	0
burger.letters.com	[01/Jul/1995:00:00:12	GET	/shuttle/countdown/video/livevideo.gif	HTTP/1.0	200	0
205.212.115.106	[01/Jul/1995:00:00:12	GET	/shuttle/countdown/countdown.html	HTTP/1.0	200	3985
d104.aa.net	[01/Jul/1995:00:00:13	GET	/shuttle/countdown/	HTTP/1.0	200	3985
129.94.144.152	[01/Jul/1995:00:00:13	GET	/	HTTP/1.0	200	7074

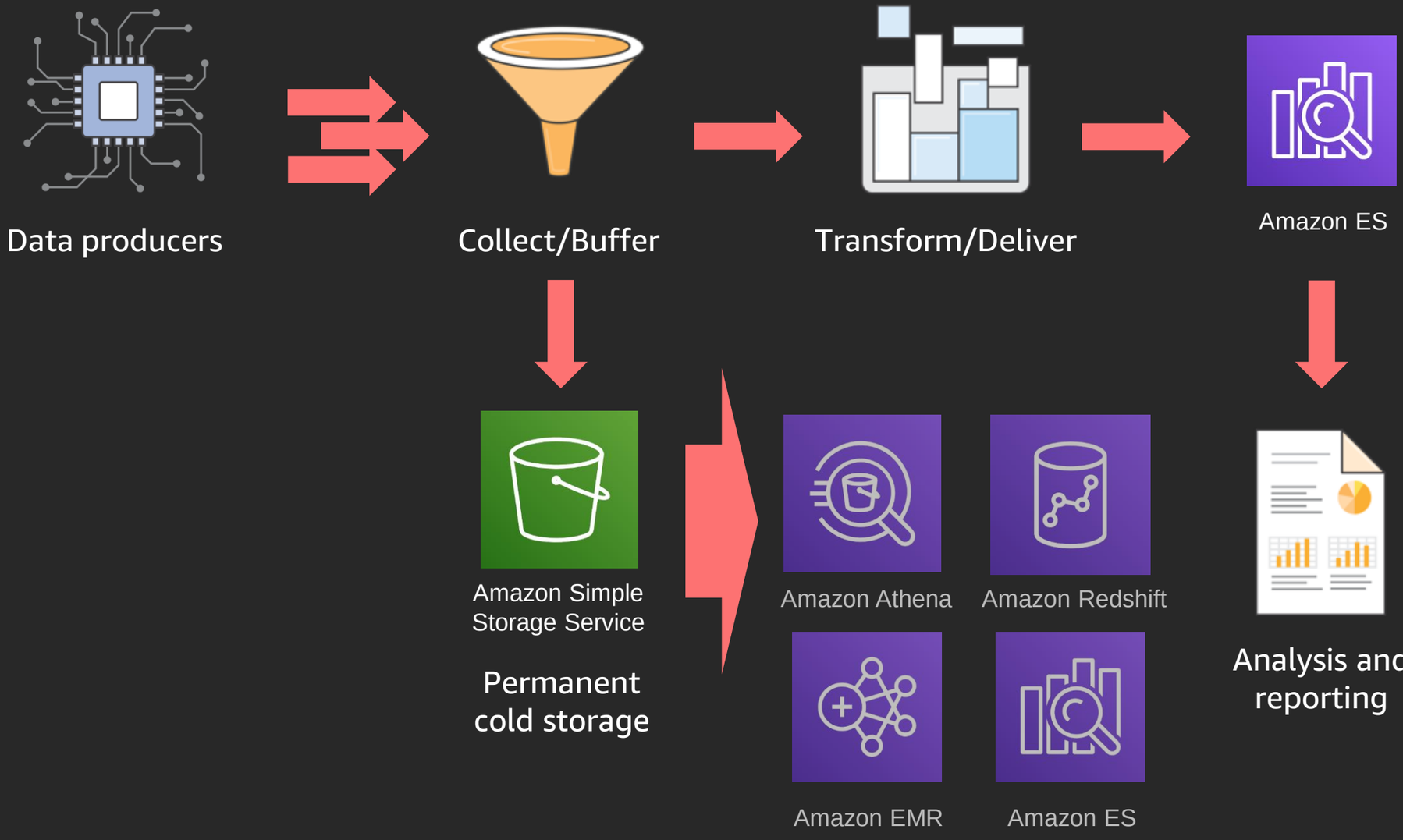
Aggregate: Size sum

Host	Timestamp	Verb	Request	Http	Status	Size
199.72.81.55	[01/Jul/1995:00:00:01	GET	/history/apollo/	HTTP/1.0	200	6245
unicomp6.unicomp.net	[01/Jul/1995:00:00:06	GET	/shuttle/countdown/	HTTP/1.0	200	3985
199.120.110.21	[01/Jul/1995:00:00:09	GET	/shuttle/missions/sts-73/mission-sts-73.html	HTTP/1.0	200	4085
burger.letters.com	[01/Jul/1995:00:00:11	GET	/shuttle/countdown/liftoff.html	HTTP/1.0	304	0
199.120.110.21	[01/Jul/1995:00:00:11	GET	/shuttle/missions/sts-73/sts-73-patch-small.gif	HTTP/1.0	200	4179
burger.letters.com	[01/Jul/1995:00:00:12	GET	/images/NASA-logosmall.gif	HTTP/1.0	304	0
burger.letters.com	[01/Jul/1995:00:00:12	GET	/shuttle/countdown/video/livevideo.gif	HTTP/1.0	200	0
205.212.115.106	[01/Jul/1995:00:00:12	GET	/shuttle/countdown/countdown.html	HTTP/1.0	200	3985
d104.aa.net	[01/Jul/1995:00:00:13	GET	/shuttle/countdown/	HTTP/1.0	200	3985
129.94.144.152	[01/Jul/1995:00:00:13	GET	/	HTTP/1.0	200	7074

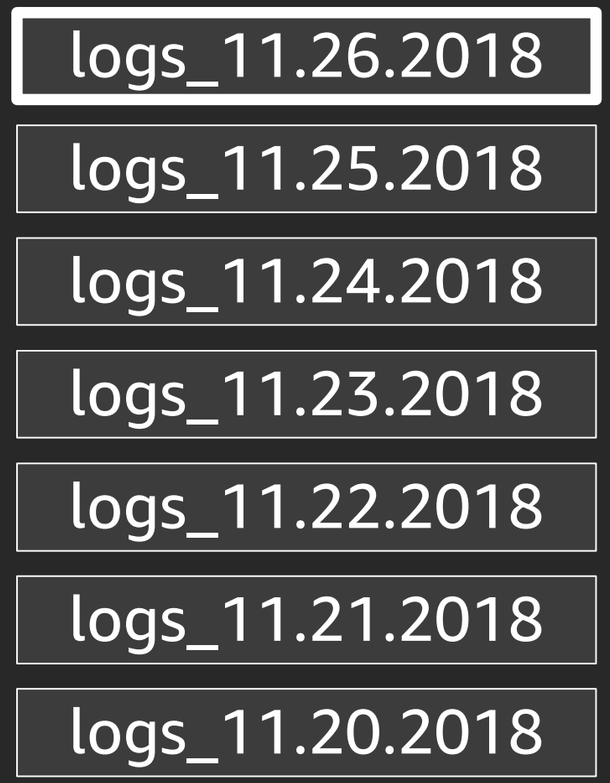
Example: AWS CloudTrail



Log analytics data flow



When you use Amazon ES for log analysis, create daily indexes

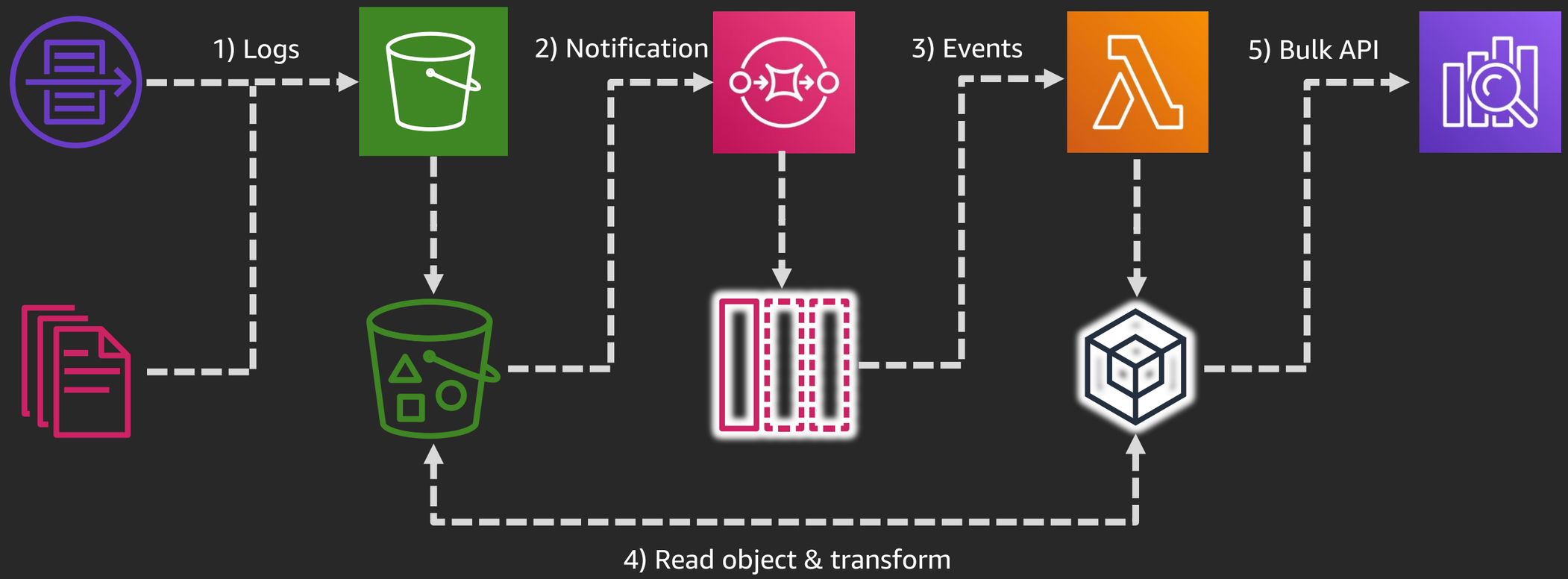


On ingest, create indexes with a root string, e.g., logs_

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

Amazon S3 event notifications approach



Best practice: use R5s for log analytics

Best practice: active shard count < vCPU count

Best practice: use a _template to manage sharding

Josh Pavel

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

- The problem
- The design
- The process
- The results
- What’s next?

ALWAYS LEARNING



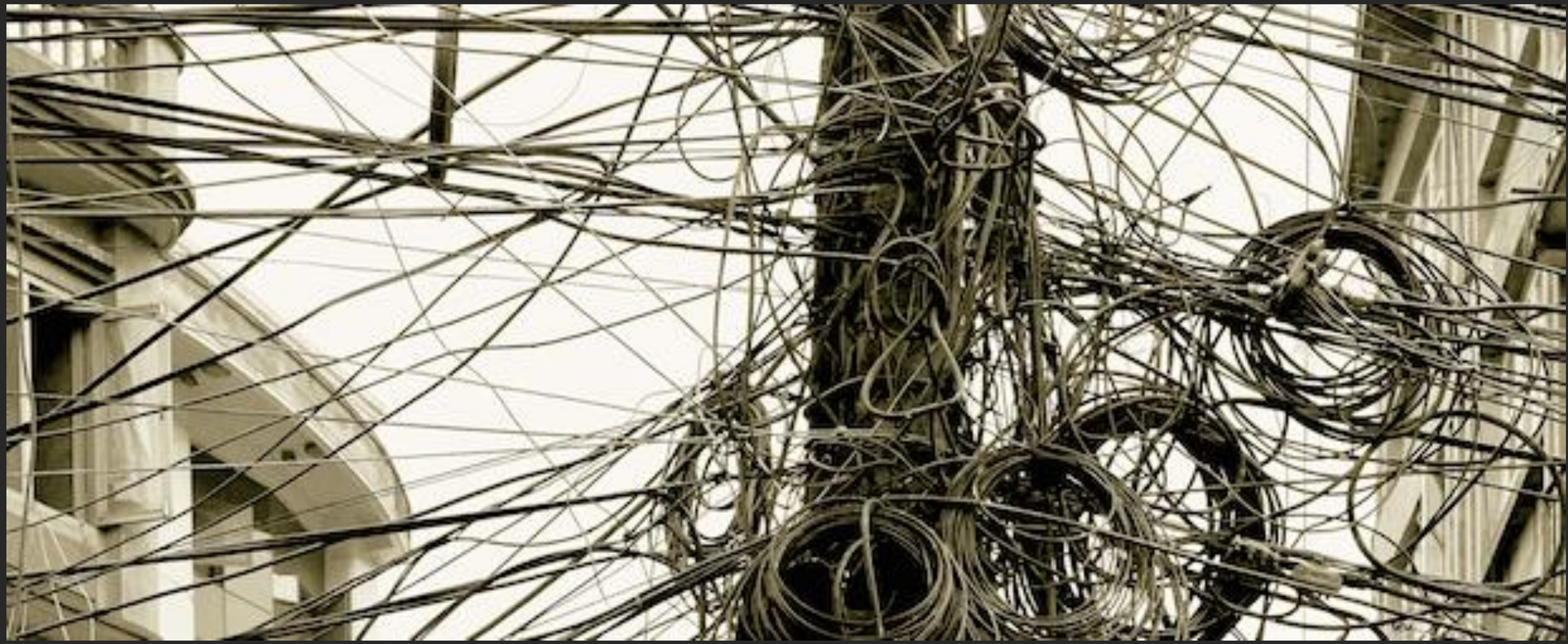
Pearson

The problem

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Data chaos!



"Think of a bicycle wheel; it goes round and round, but it is moving forwards, not standing still. The same circuit around the hub of the wheel becomes part of the forward movement of the bicycle as a whole."

N.T. Wright

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

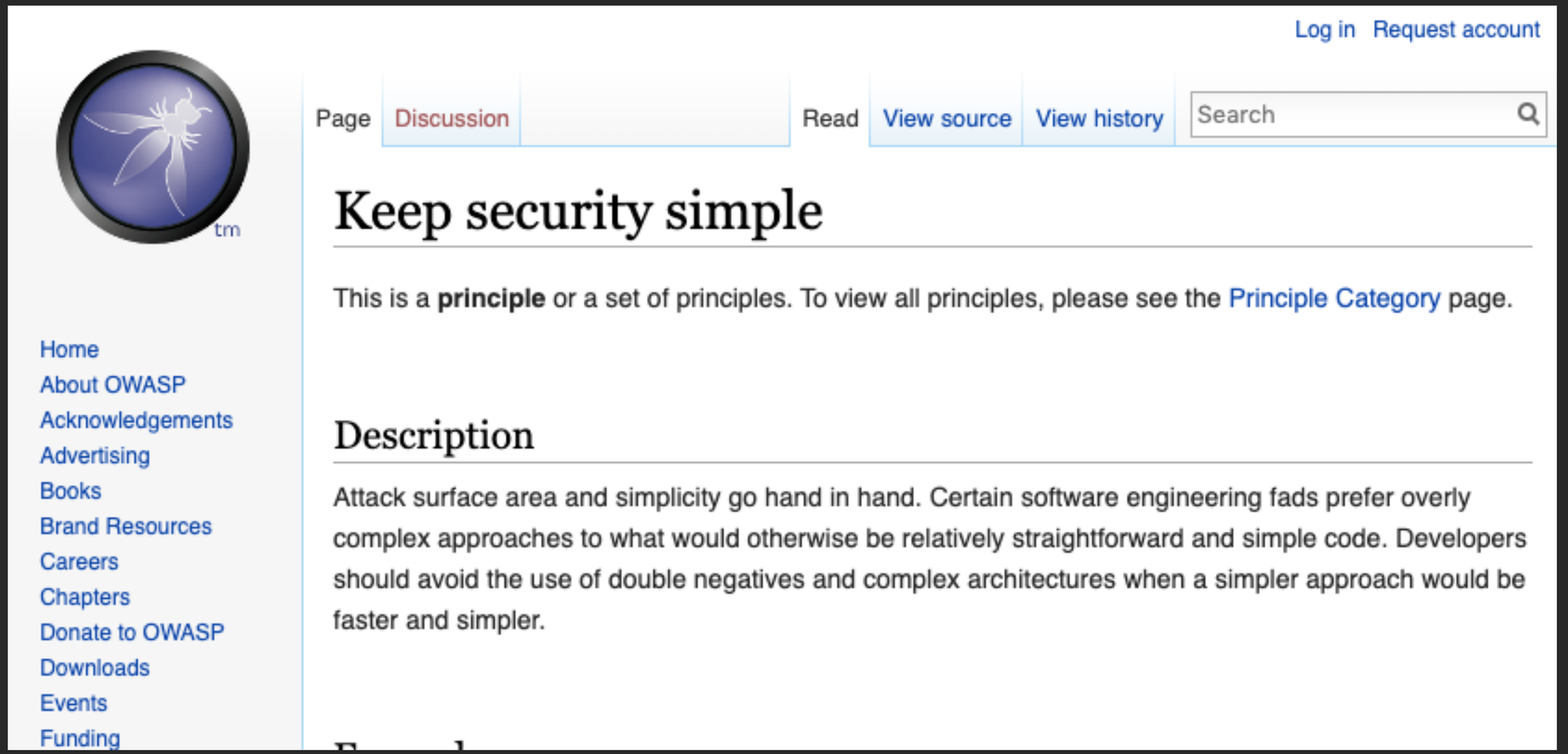
"If I have seen further, it is by standing on the shoulders of giants."

Sir Isaac Newton

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Keep it simple

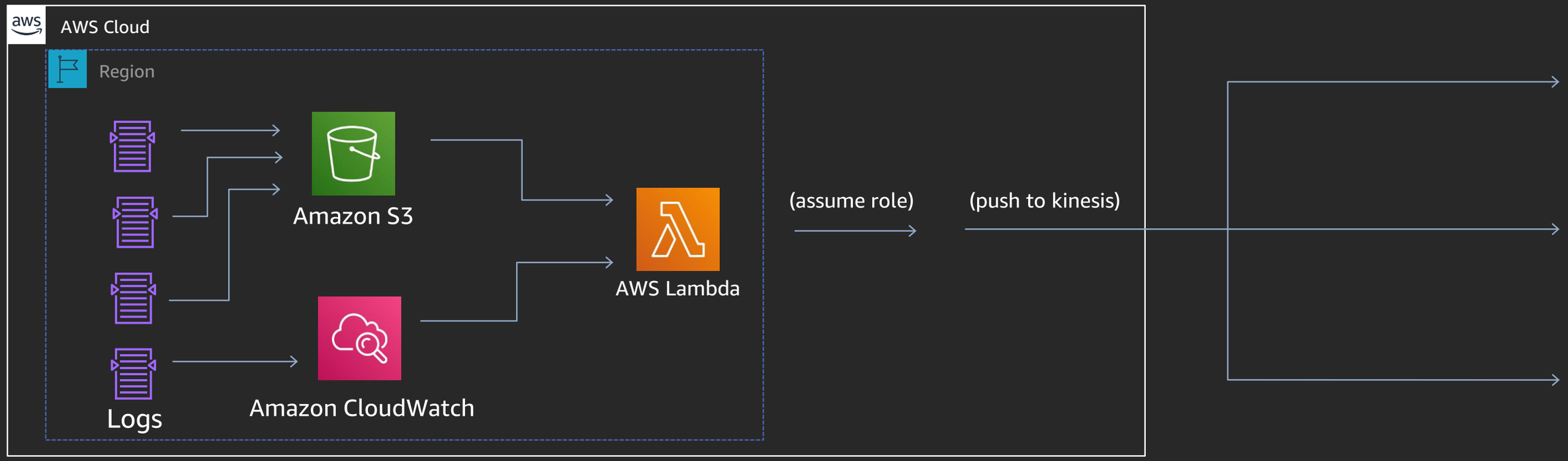
Simplicity is a security principle (OWASP)



The design (client account)

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

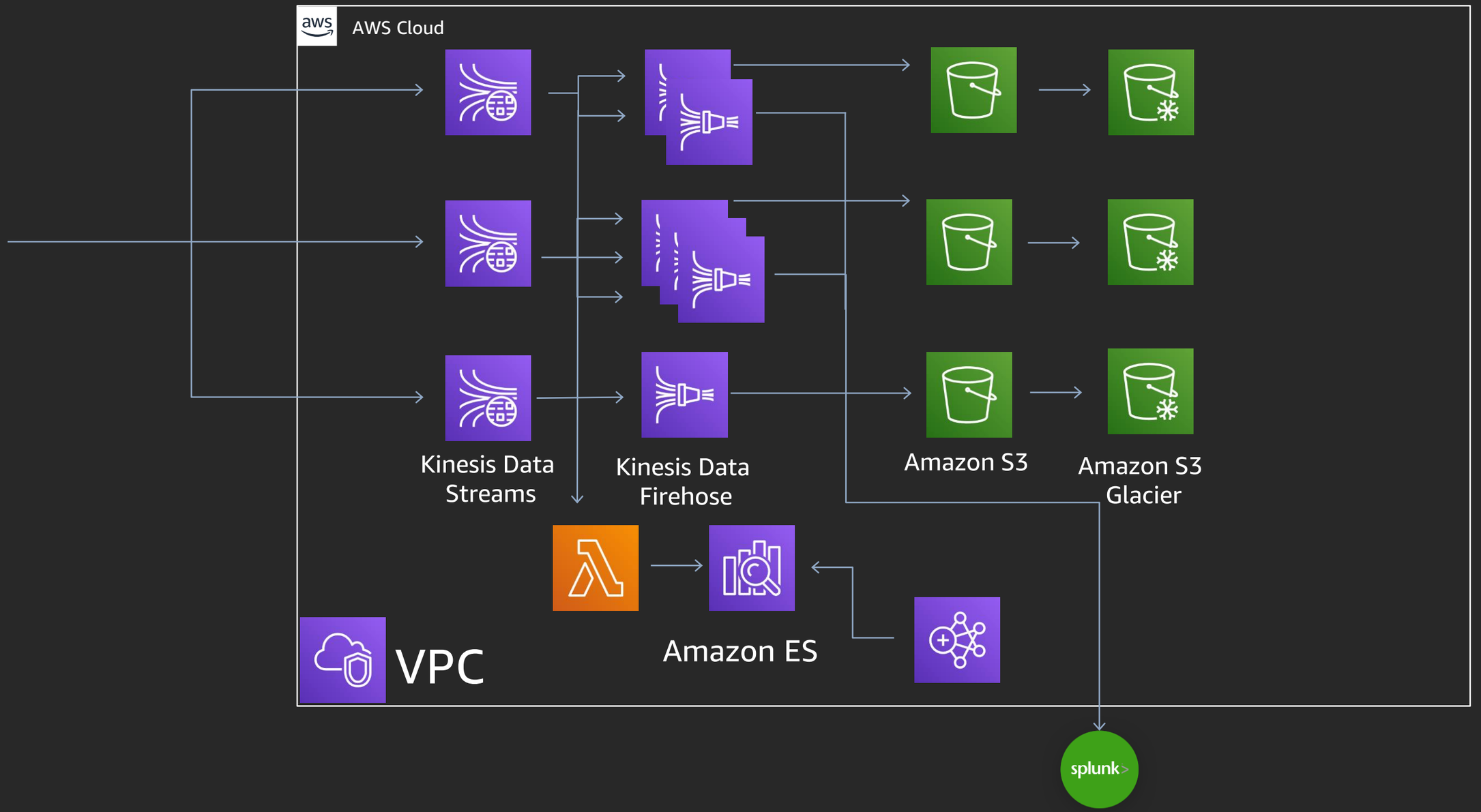
Client side


aws
AWS Cloud
Region
Amazon S3
AWS Lambda
(assume role)
(push to kinesis)
Logs
Amazon CloudWatch

The design (centralized account)

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Centralized logging account



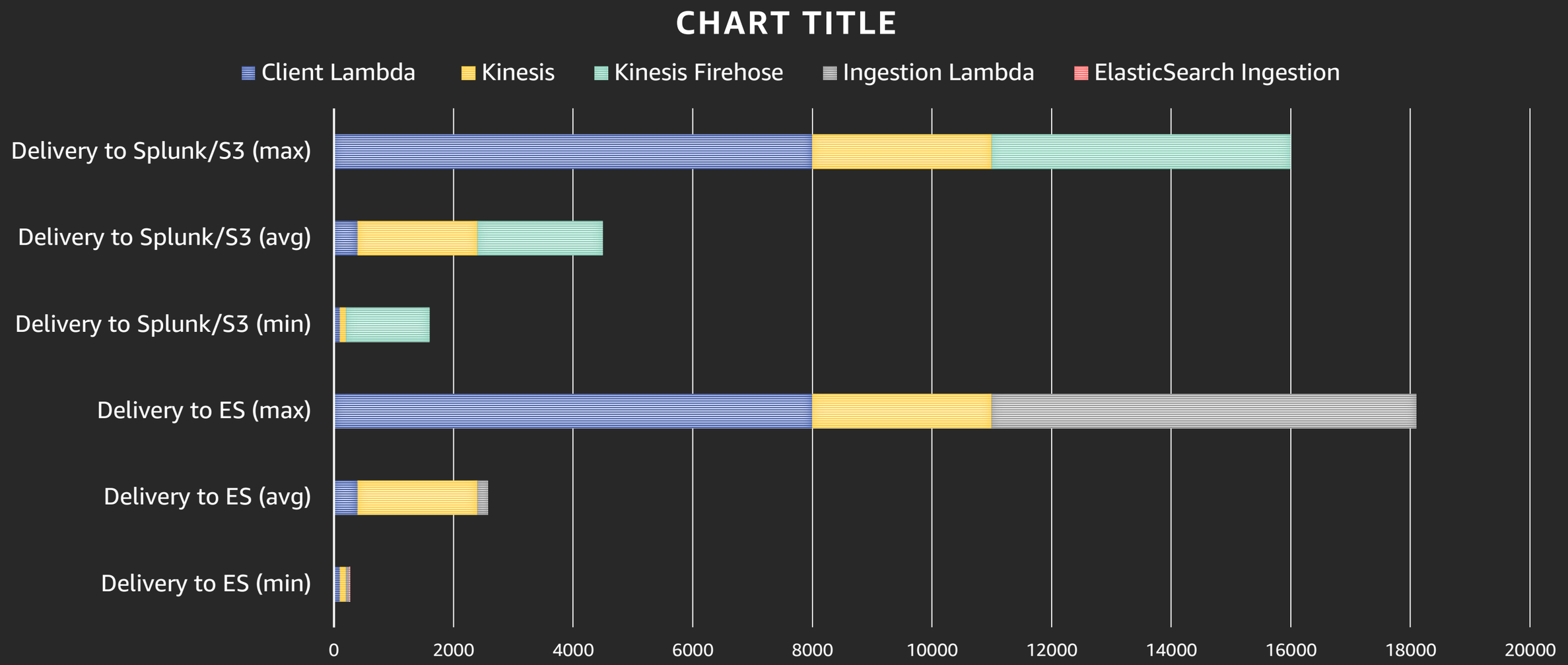
The results

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

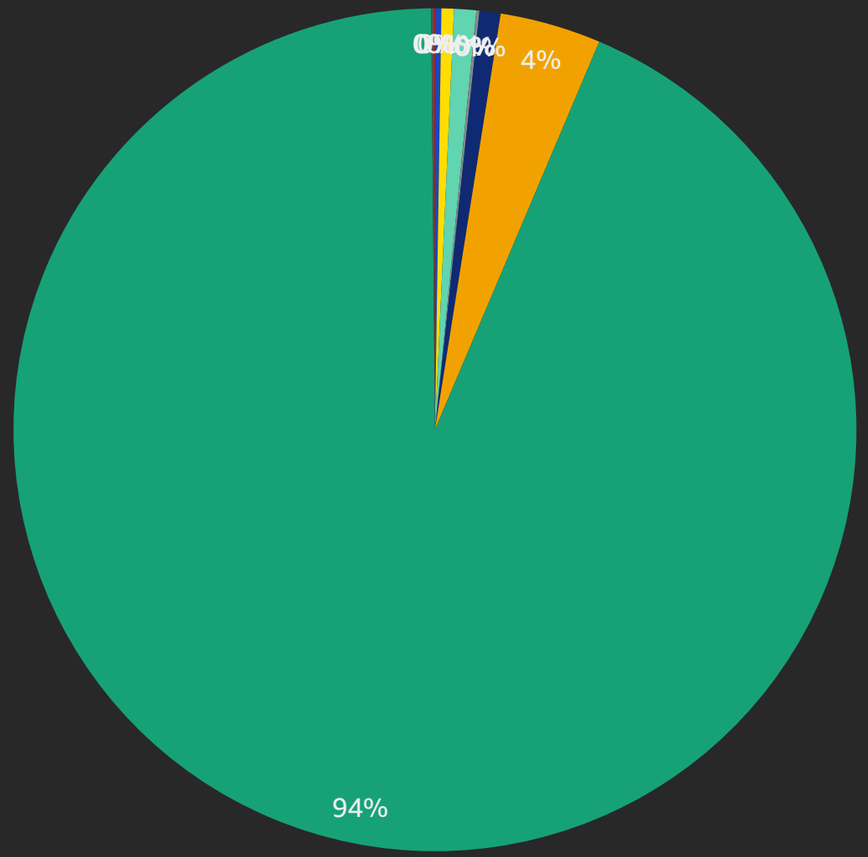
Ingestion rates

	Current	Building/Testing for…
Accounts	14	~ 1000
Data size	4 TB / 30 days	500 TB/ 30 days
Log lines	Up to 17,000/sec (30 days: 44,676,000,000)	Up to 100,000/sec (30 days: 262,800,000,000)

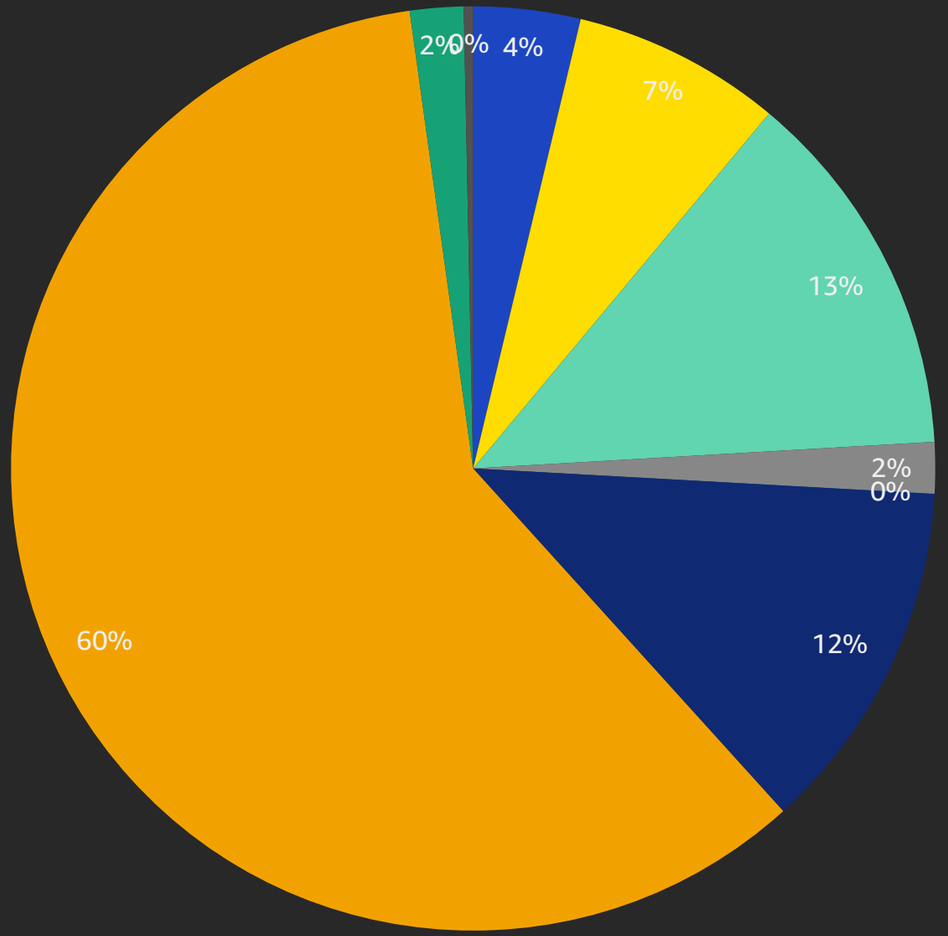
Processing time (in ms)



Cost breakdown



Cost breakdown (the other 6%)



Kinesis - Stream 1
Kinesis - Stream 2
Kinesis - Stream 3
Kinesis - Stream 4
Kinesis Firehose
Ingestion Lambda
ElasticSearch - Storage
ElasticSearch - Master
S3
Glacier

Lessons learned

- BULK and BATCH
  - Bulk add to Kinesis
  - Bulk add to Amazon ES (using the Python library)
  - Batch process events in lambdas, batch size reading from Kinesis

These two edits made all the difference

```
# Client Lambda

kinesis = global_session.client('kinesis')

    try:

        r = kinesis.put_records(

            StreamName=stream_name,

            Records=records

        )

    except Exception as e:

       logger.critical("ERROR putting
       records to kinesis stream: " +
       str(e))
```

```
# Ingestion Lambda

status = helpers.bulk(

        es,

        to_send,

        raise_on_error=False, # Leave this
        off so that errors still get
        logged, but we don't stall and re-
        process events.

        request_timeout=30

    )

    print("DEBUG - Response status: " +
       str(status))
```

Lessons learned

- Credential DURATION matters
  - This is a constant system; when you authorize matters
  - 50% of our logs were centralized logging authorizations!

Lessons learned

- Kinesis START POINT matters
  - “LATEST” sounds good; you want “TRIM_HORIZON”
- DLQ and RETRY
  - CI/CD pipeline problems—not a problem!

Type

You can set the starting position to one of the following values:

`AT_SEQUENCE_NUMBER`: Start streaming from the position denoted by the sequence number specified in the `SequenceNumber` field.

`AFTER_SEQUENCE_NUMBER`: Start streaming right after the position denoted by the sequence number specified in the `SequenceNumber` field.

`AT_TIMESTAMP`: Start streaming from the position denoted by the time stamp specified in the `Timestamp` field.

`TRIM_HORIZON`: Start streaming at the last untrimmed record in the shard, which is the oldest data record in the shard.

`LATEST`: Start streaming just after the most recent record in the shard, so that you always read the most recent data in the shard.

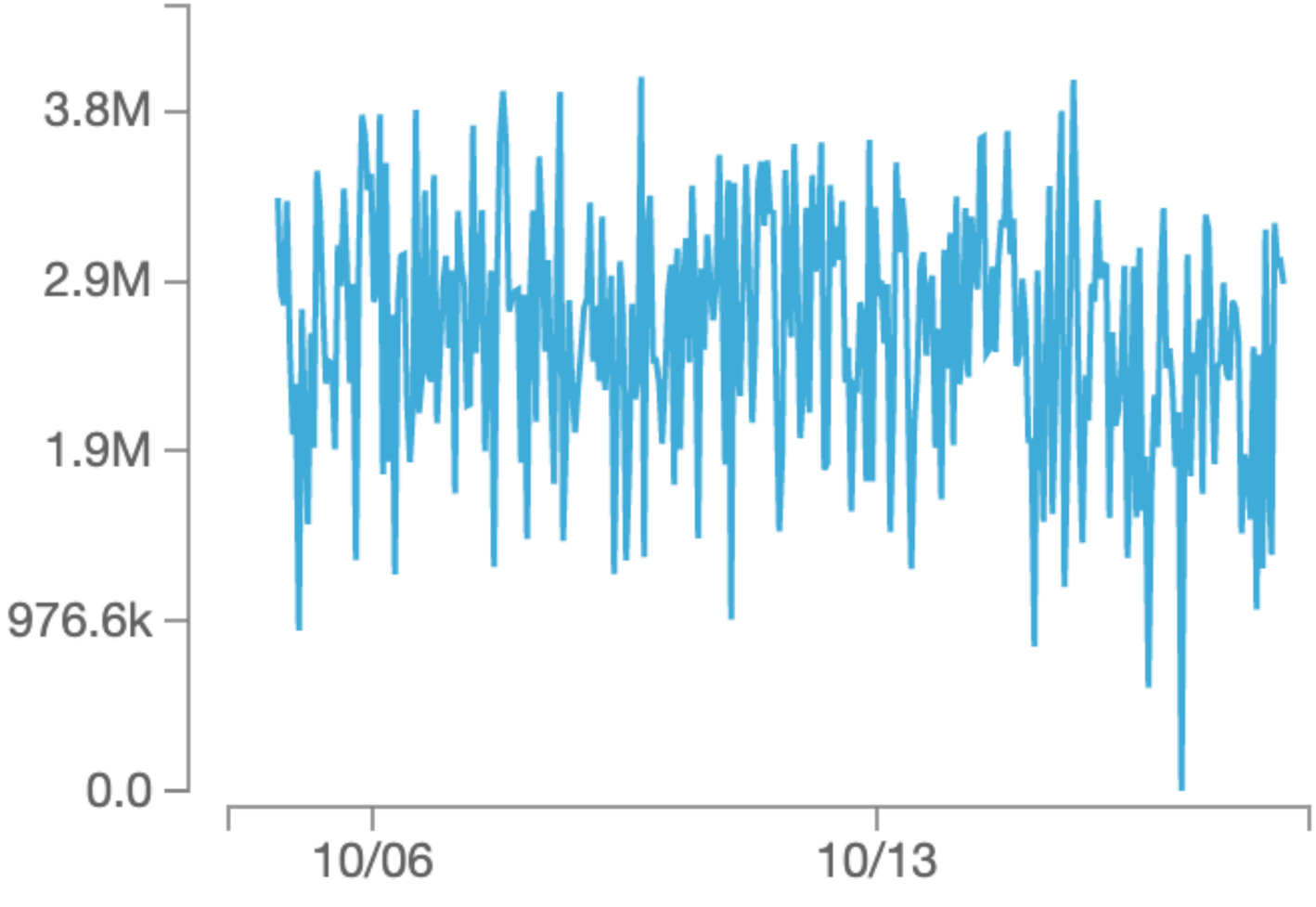
Type: String

Valid Values: `AT_SEQUENCE_NUMBER` | `AFTER_SEQUENCE_NUMBER` | `TRIM_HORIZON` | `LATEST` | `AT_TIMESTAMP`

Required: Yes

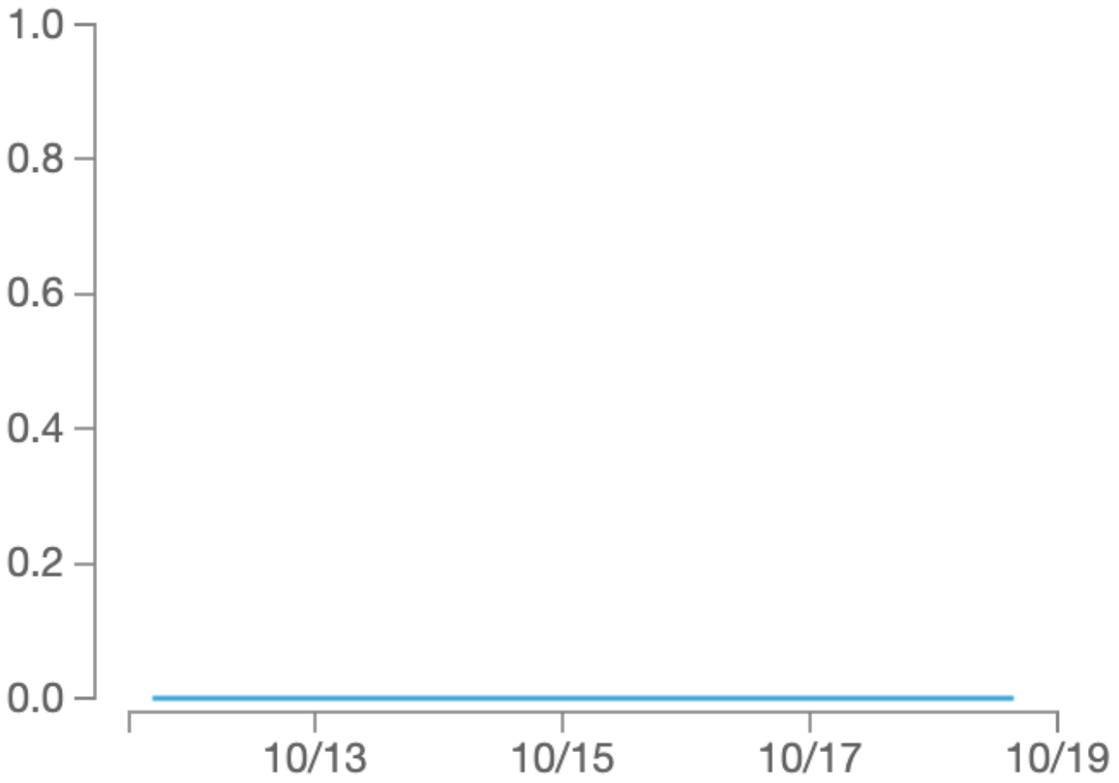
Get Records Iterator Age (Milliseconds) — Maximum

GetRecords.IteratorAgeMilliseconds



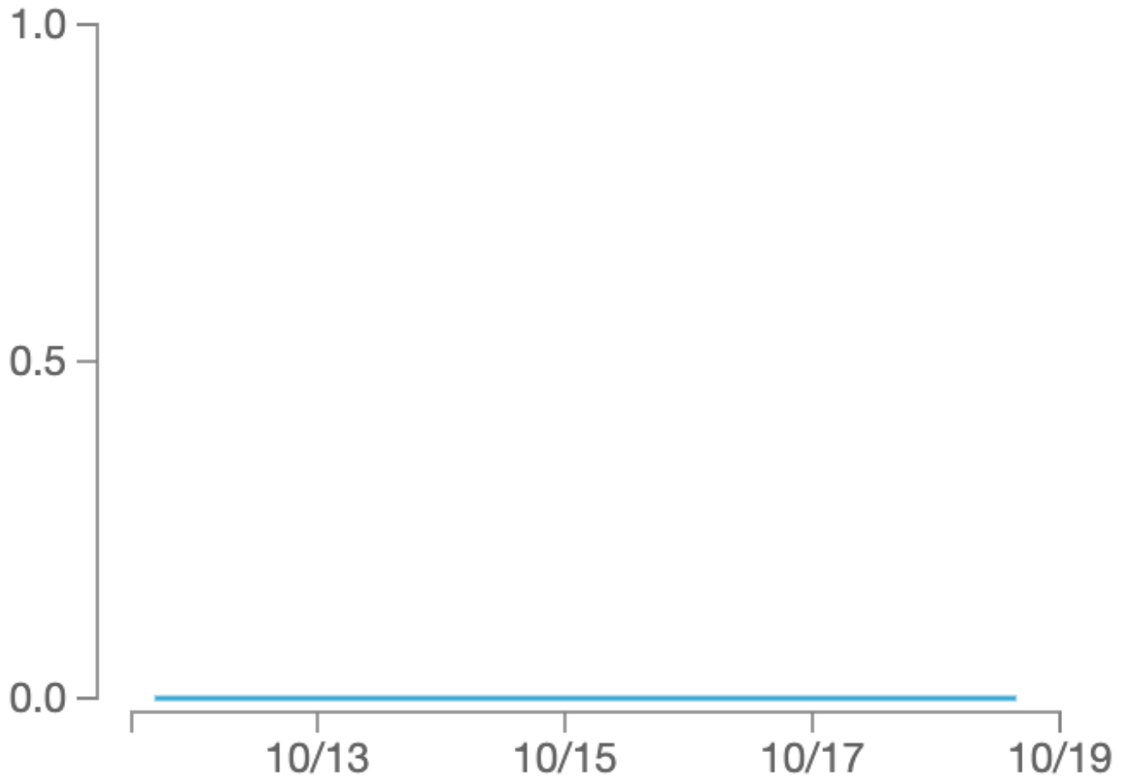
Read Throughput Exceeded (Count / Request) — Average

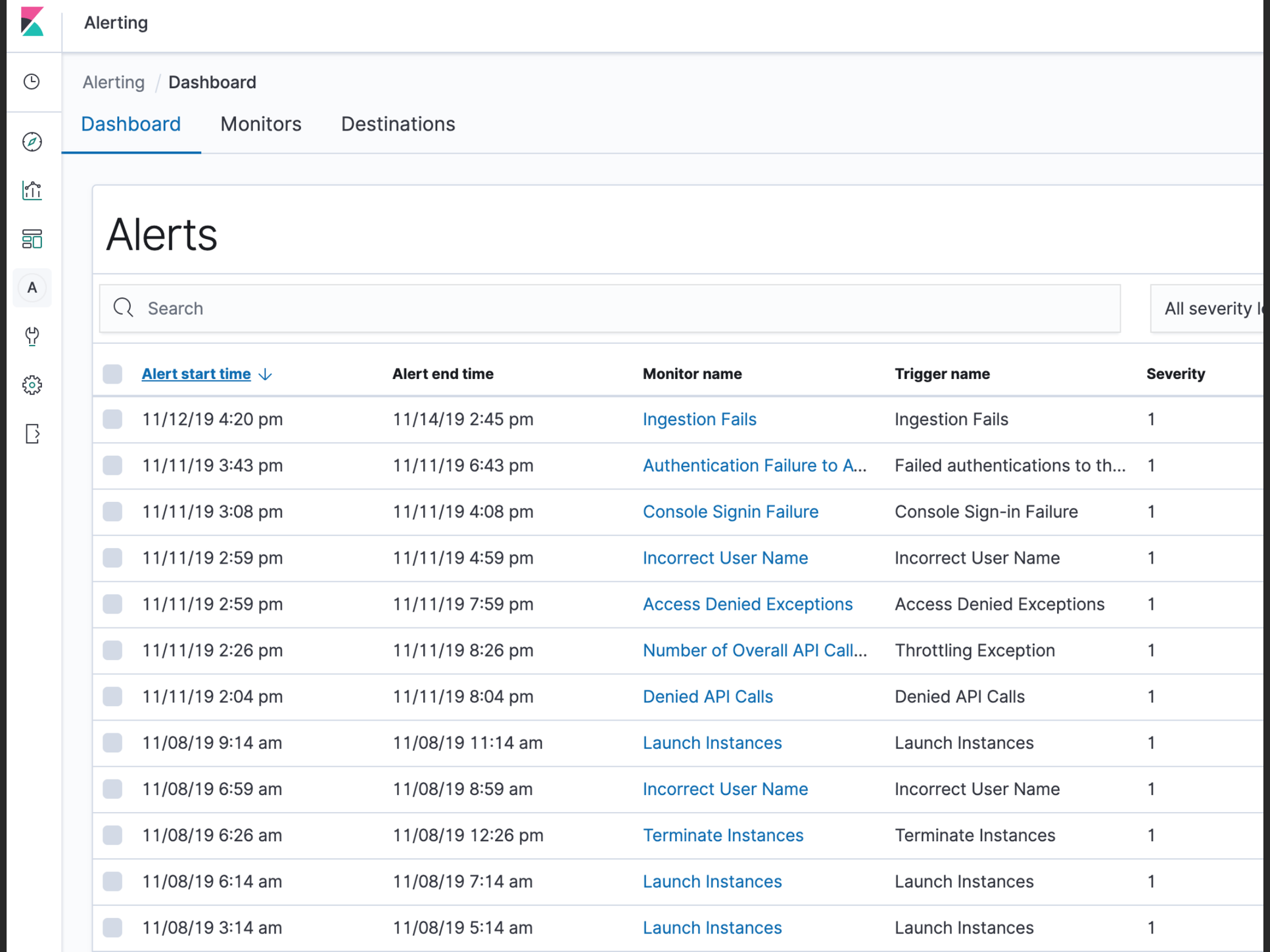
ReadProvisionedThroughputExceeded



Write Throughput Exceeded (Count / Request) — Average

WriteProvisionedThroughputExceeded




Alerting
Alerting / Dashboard
Dashboard
Monitors
Destinations
Alerts
Search
All severity l
Alert start time
Alert end time
Monitor name
Trigger name
Severity
11/12/19 4:20 pm
11/14/19 2:45 pm
Ingestion Fails
Ingestion Fails
1
11/11/19 3:43 pm
11/11/19 6:43 pm
Authentication Failure to A...
Failed authentications to th...
1
11/11/19 3:08 pm
11/11/19 4:08 pm
Console Signin Failure
Console Sign-in Failure
1
11/11/19 2:59 pm
11/11/19 4:59 pm
Incorrect User Name
Incorrect User Name
1
11/11/19 2:59 pm
11/11/19 7:59 pm
Access Denied Exceptions
Access Denied Exceptions
1
11/11/19 2:26 pm
11/11/19 8:26 pm
Number of Overall API Call...
Throttling Exception
1
11/11/19 2:04 pm
11/11/19 8:04 pm
Denied API Calls
Denied API Calls
1
11/08/19 9:14 am
11/08/19 11:14 am
Launch Instances
Launch Instances
1
11/08/19 6:59 am
11/08/19 8:59 am
Incorrect User Name
Incorrect User Name
1
11/08/19 6:26 am
11/08/19 12:26 pm
Terminate Instances
Terminate Instances
1
11/08/19 6:14 am
11/08/19 7:14 am
Launch Instances
Launch Instances
1
11/08/19 3:14 am
11/08/19 5:14 am
Launch Instances
Launch Instances
1

CCL Kibana Alerting APP 6:04 PM

Number of times an API call was denied

Monitor Denied API Calls just entered alert status. Please investigate the issue.
- Trigger: Denied API Calls
- Severity: 1
- Period start: 2019-11-11T22:04:35.201Z
- Period end: 2019-11-11T23:04:35.201Z
- Total failures: 66
- Metadata (truncated to 10):

Index: [redacted]-cloudtrail-2019-11-11, Region: us-east-1, EventSource: eks.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: us-west-1, EventSource: eks.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: ca-central-1, EventSource: eks.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: us-west-2, EventSource: eks.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: us-west-2, EventSource: eks.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: us-west-2, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-west-2, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-central-1, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-central-1, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-central-1, EventSource: workspaces.amazonaws.com

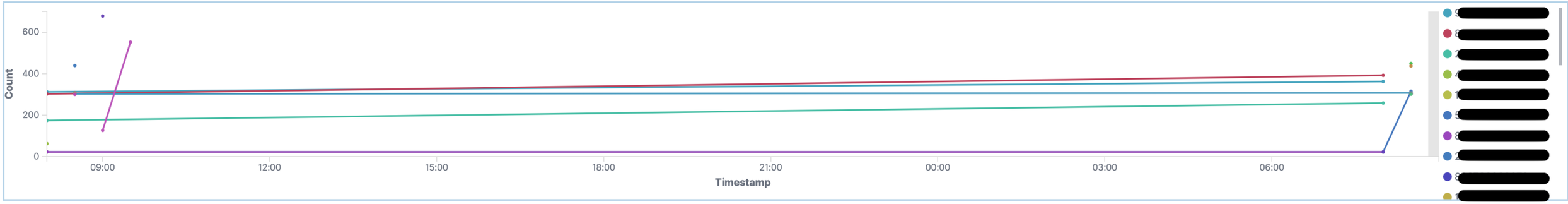
CCL Kibana Alerting APP 6:26 PM

Number of overall API calls exceeds the threshold

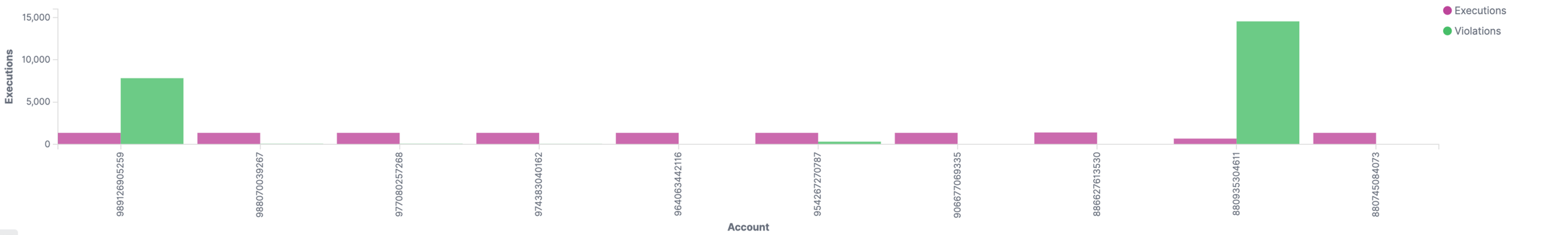
Monitor Number of Overall API Calls Exceeding a Threshold in the last hour.
- Trigger: Throttling Exception
- Severity: 1
- Period start: 2019-11-11T22:26:01.685Z
- Period end: 2019-11-11T23:26:01.685Z
- Total failures: 33
- Metadata (truncated to 10):

Index: [redacted]-cloudtrail-2019-11-11, Region: eu-west-2, EventSource: ec2.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: us-west-2, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-west-2, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-central-1, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-west-2, EventSource: ec2.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-west-2, EventSource: ec2.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-west-2, EventSource: ec2.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-central-1, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: eu-central-1, EventSource: workspaces.amazonaws.com
Index: [redacted]-cloudtrail-2019-11-11, Region: us-east-1, EventSource: workspaces.amazonaws.com

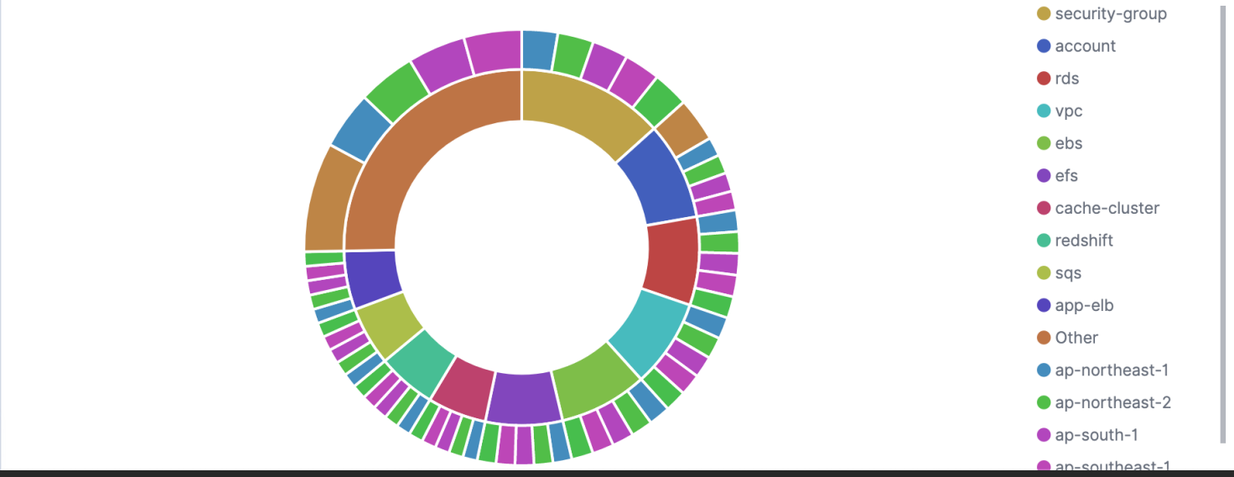
PCS (Prod) - Top 10 Account Executions Over Time



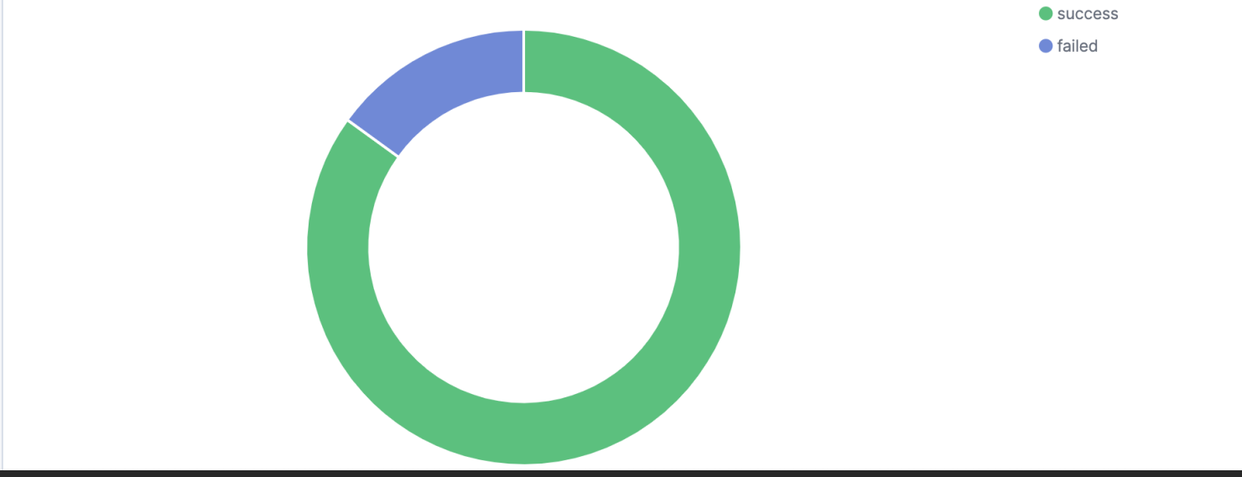
PCS (Prod) - Top 10 Account Report



PCS (Prod) - Top 10 Resource Executions By Region



PCS (Prod) - Error Rate



Lessons learned (continued)

- Watch Amazon ES field growth
  - (I’m looking at you, CloudTrail)—"index.mapping.total_fields.limit”

Amazon ES tuning algorithm (for IaC, run in lambda)

Storage is cheap, shards are not:

- In an unmanaged solution, you can handle outliers by pinning them to specially tuned data nodes.

Our rules of thumb With AWS ES:

- Keep shards at 20 GB (max) – most of ours are well under that (because of daily rotation)
- Keep shards per node < 3000
- Re-evaluate strategy around 10,000 shards (your mileage will vary)

Scale up the NUMBER of data nodes when calculated shards required are greater than available:

- (#data_nodes) X (#_shards_per_node) = TOTAL SHARDS AVAILABLE
- (TOTAL SHARD NEEDS FOR DAILY INDICES) X (ES RETENTION) = TOTAL SHARDS REQUIRED

Scale up the SIZE of data nodes when indexing latency is an issue

Amazon ES tuning algorithm:

For IaC, run in lambda to check/set:

```
PUT /_cluster/settings
{
  "persistent": {
    "cluster.max_shards_per_node": 1500
  }
}
```

On index creation:

```
"settings": {
 "index.mapping.total_fields.limit": 5000,
 "index.number_of_shards": 2,
 "index.number_of_replicas": 1
},
```

Lessons learned (continued)

- Nested fields
  - Developers love arrays; Elasticsearch/Kibana does not

Lessons learned (continued)

- Index rotation
  - Simplifies retention; complicates access controls

Lessons learned (continued)

- Have a data organization/segregation plan – “<account>-<type>-<unique>”

Lessons learned (continued)

- ”Big data” problems
  - Kibana’s server.maxPayloadBytes setting is preset in AWS:
  - “Payload content length greater than maximum allowed: 1048576”

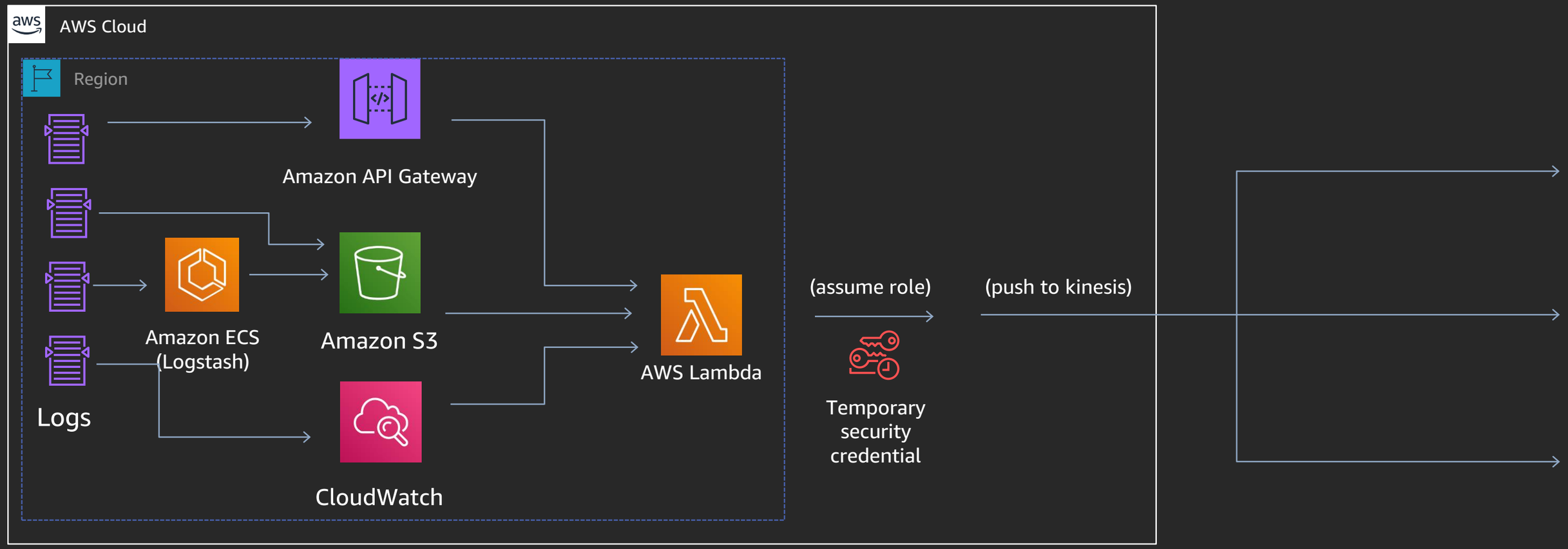
Benefits of design (as compared to alternates we evaluated)

- FLEXIBLE and SCALABLE
- SECURE
- EFFECTIVE
- COSTING

What’s next

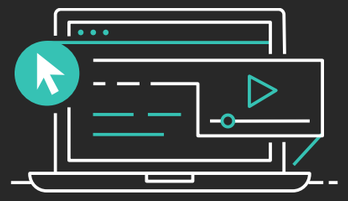
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Client side (alternate, P.O.C.)



Learn big data with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate data analytics skills



New free digital course, Data Analytics Fundamentals, introduces Amazon S3, Amazon Kinesis, Amazon EMR, AWS Glue, and Amazon Redshift



Classroom offerings, including Big Data on AWS, feature AWS expert instructors and hands-on labs



Validate expertise with the **AWS Certified Big Data - Specialty** exam or the new **AWS Certified Data Analytics - Specialty** beta exam

Visit aws.amazon.com/training/paths-specialty/

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws training and certification

Thank you!

Josh Pavel
joshua.pavel@pearson.com

Jon Handler
handler@amazon.com

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.