ABD302

AWS re:INVENT

Real-Time Data Exploration and Analytics with Amazon Elasticsearch Service

Jon Handler – Principal Solutions Architect, AWS

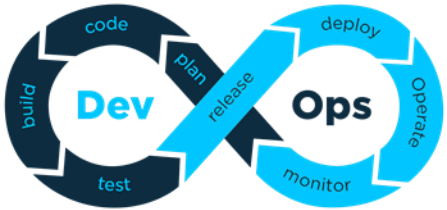
November 2017

AWS re:Invent

aws

THE EXPLOSION OF MACHINE-GENERATED DATA

Machine-generated data is growing **10x faster** than business data



Transition from IT to DevOps

Increase in IoT and Mobile Devices

Cloud-based architectures

Source: insideBigData - The Exponential Growth of Data, February 16, 2017

LOG ANALYTICS FUELING ELASTICSEARCH GROWTH

BENEFITS

- Open source
- Fast time to value
- Easy ingestion
- Easy visualization
- High performance and distributed
- Best analytics and search

Rank	Project Name	Overall Project Rating
1	Linux	100.00
2	Git	31.10
3	MySQL	25.23
4	Node.js	22.75
5	Docker	22.61
6	Hadoop	16.19
7	**Elasticsearch**	**15.72**
8	Spark	14.99
9	MongoDB	14.68
10	Selenium	12.81
11	NPM	12.31
12	Redis	11.61

Source: TechCrunch survey of popular open source software from April'17

AMAZON ELASTICSEARCH SERVICE

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

elasticsearch + kibana

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

ELASTICSEARCH LEADING USE CASES

Application Monitoring & Root-cause Analysis

Provides developers with a high performance, self-service operational monitoring and analytics platform

Security Information and Event Management (SIEM)

Enables security practitioners to centralize and analyze events from across the entire organization

IoT & Mobile

Gives developers and lines of business users real-time location-aware insights into their device fleets

Business & Clickstream Analytics

Provides business users with a real-time view of the performance of their web content and e-commerce platforms

CASE STUDY: EXPEDIA

Application Monitoring & Root-cause Analysis

PROBLEM

Logs, lots and lots of logs. How to cost effectively monitor logs?

Require centralized logging infrastructure

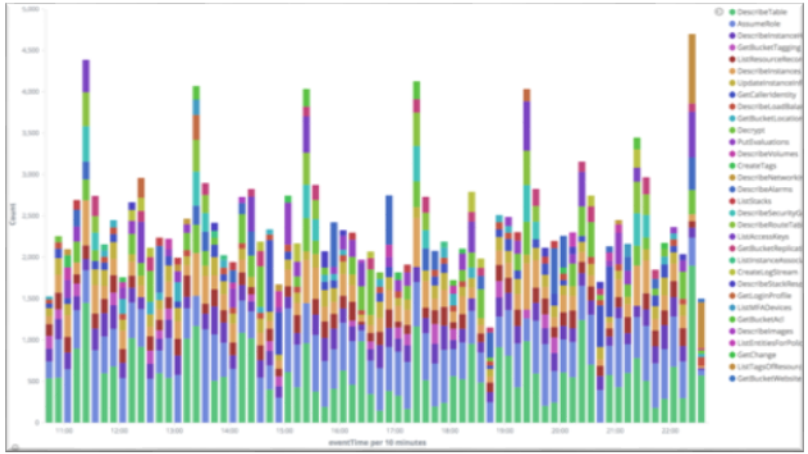
Did not have the man power to manage infrastructure

SOLUTION

Streaming AWS CloudTrail logs, application logs, and Docker startup logs to Elasticsearch

Created centralized logging service for all team members

Using Kibana for visualizations and for Elasticsearch queries



BENEFITS

Quick insights: Able to identify and troubleshoot issues in real-time

Secure: Integrated w/ AWS IAM

Scalable: Cluster sizes are able to grow to accommodate additional log sources

CASE STUDY: FINANCIAL TIMES

Business and Clickstream Analytics

FT FINANCIAL TIMES

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

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

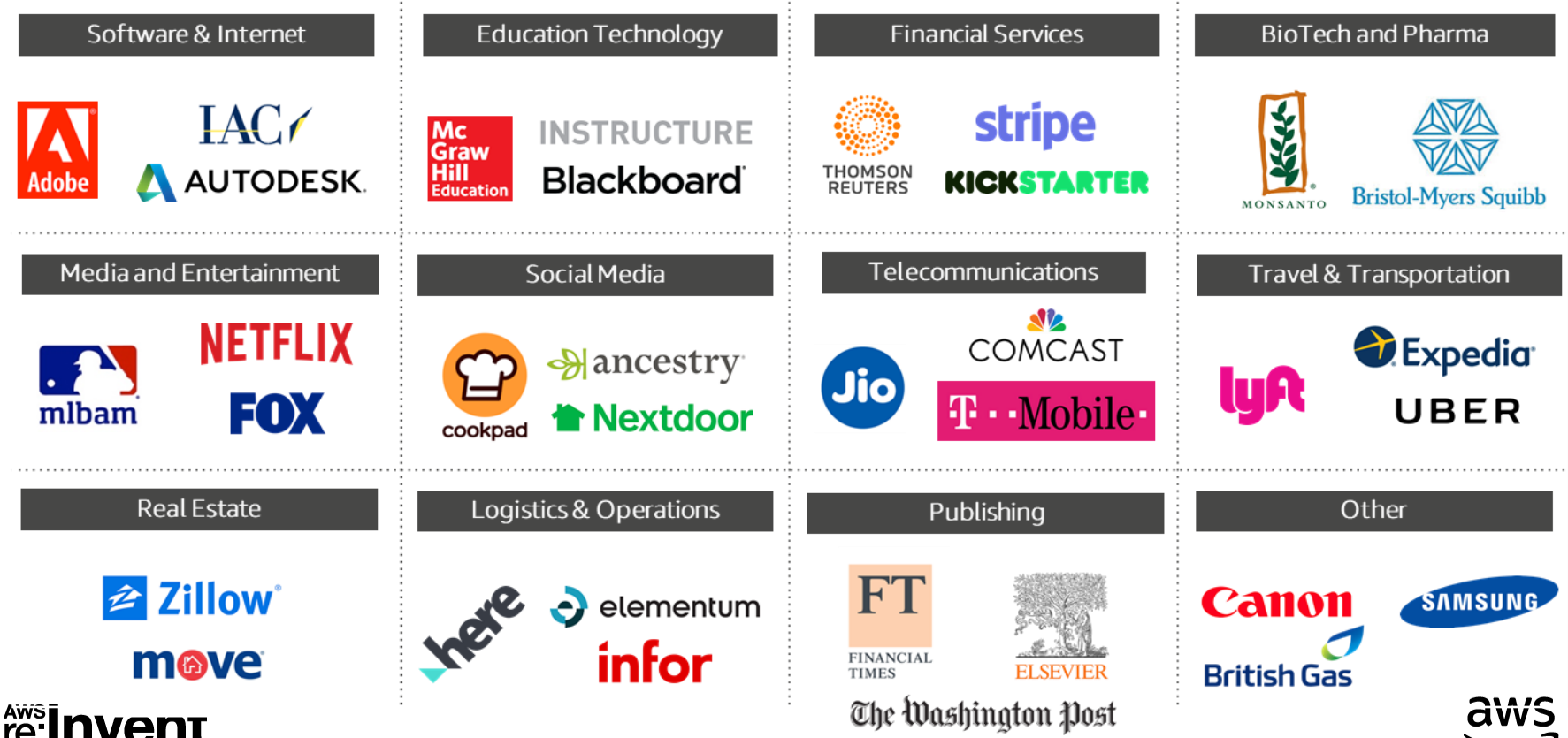
Critical tool for making editorial decisions. Daily editorial meetings start by looking at Lantern dashboard

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

AMAZON ELASTICSEARCH SERVICE CUSTOMERS



Foundation

Ingest your data

Size your domain

Secure your domain

Add durability

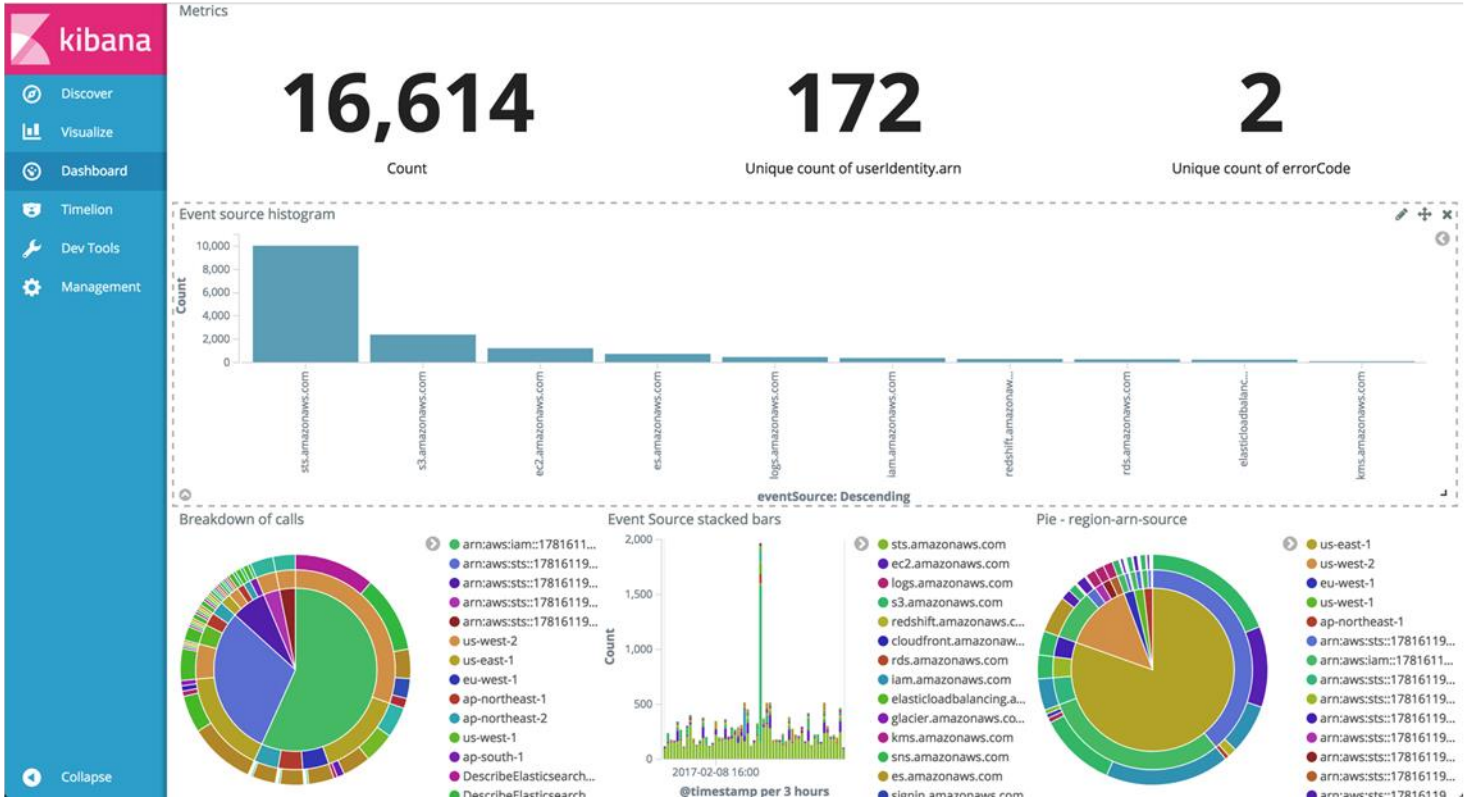
Monitor your domain

Analyze your data

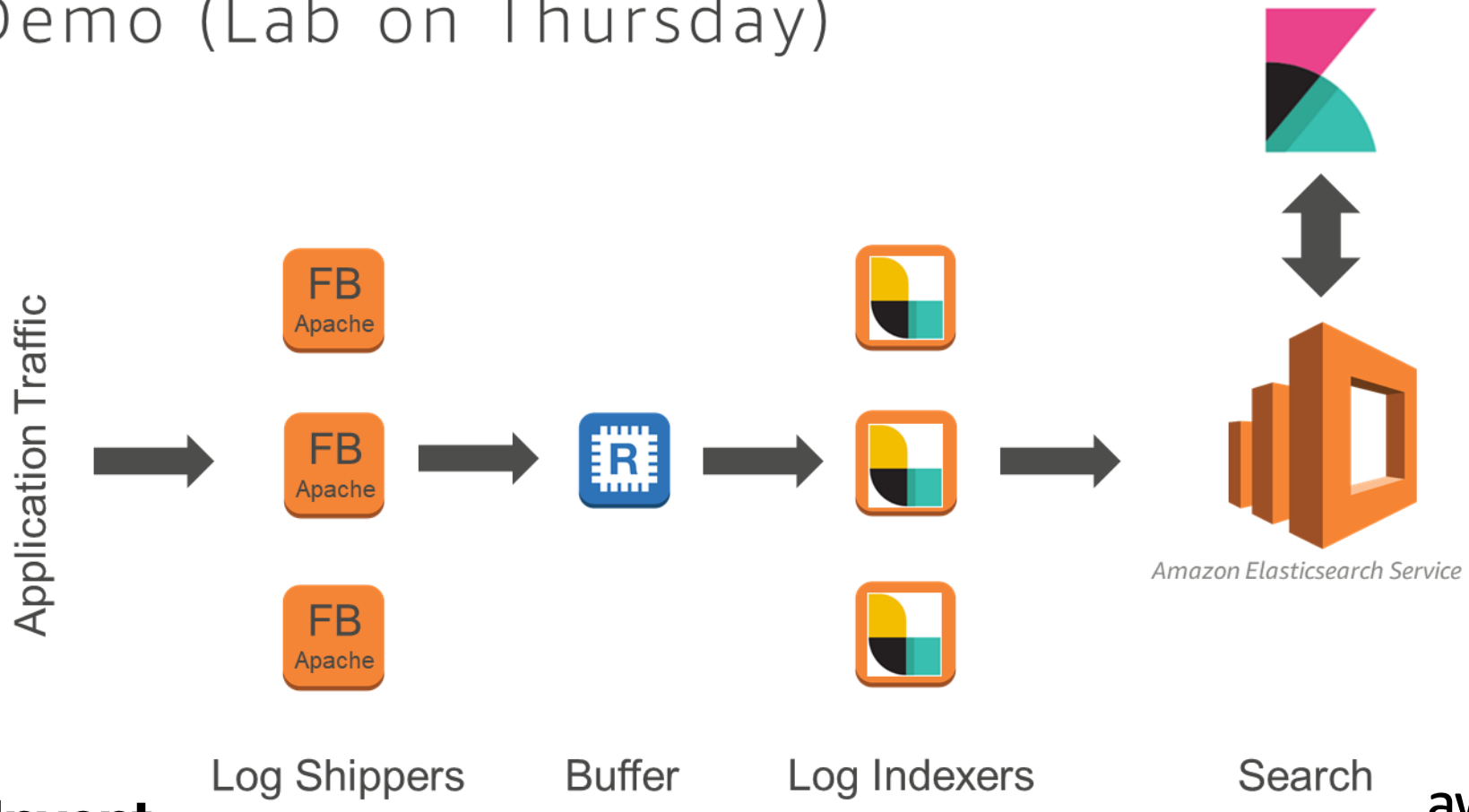
Logs are “human-readable”, but not human-useful!

```
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
```

Kibana provides real-time monitoring



Demo (Lab on Thursday)



Service architecture

AWS SDK

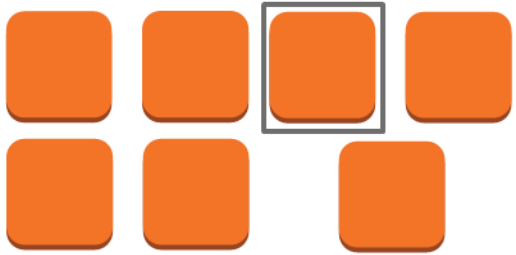
AWS CLI

AWS CloudFormation

Amazon Elasticsearch Service domain

IAM

Elastic Load Balancing



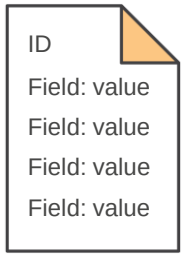
Elasticsearch data nodes

Elasticsearch master nodes

AWS CloudTrail

Amazon CloudWatch

Documents are the core entity



Elasticsearch works with structured JSON

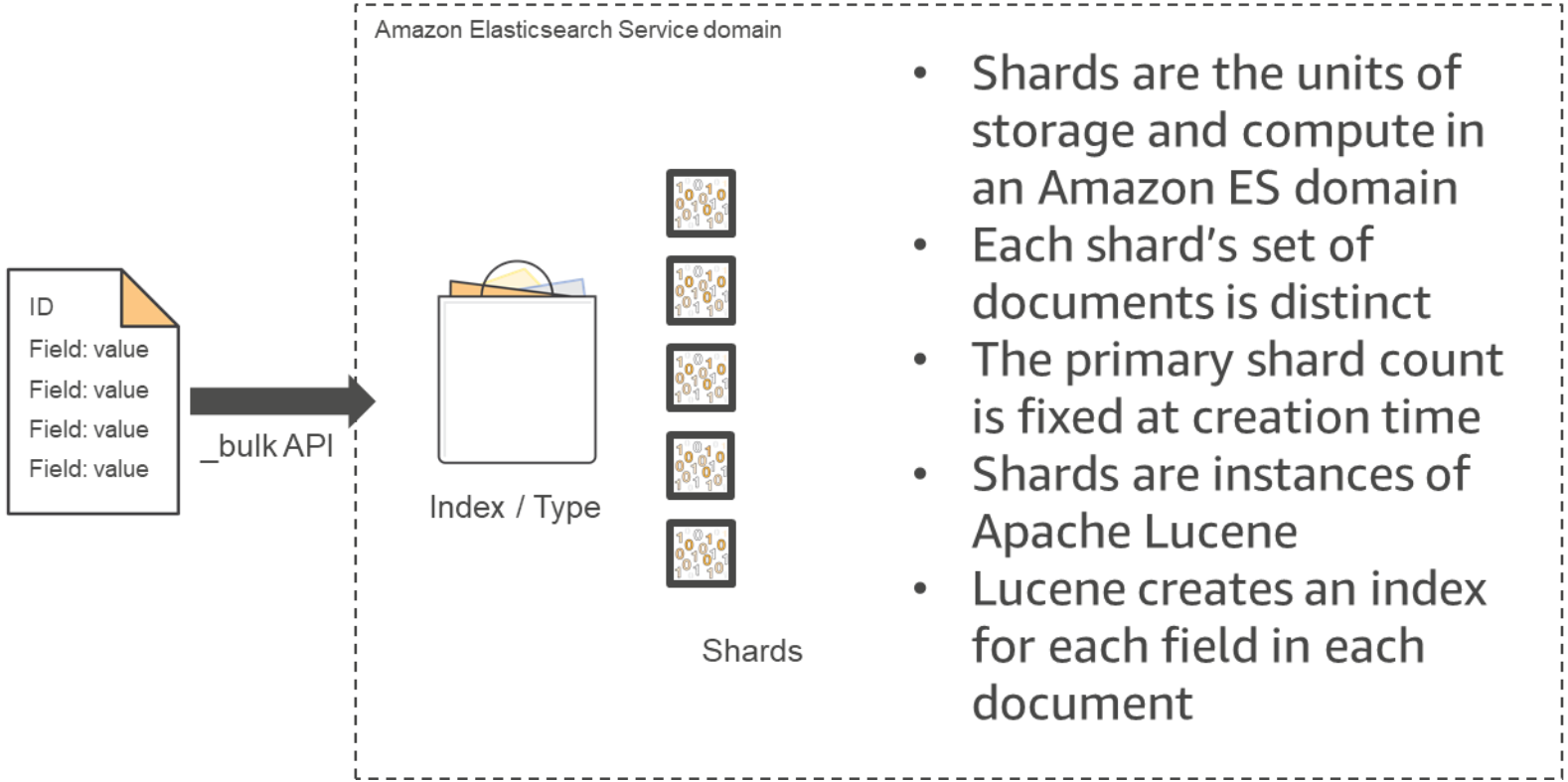
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245

```
{   "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  }
```

- Documents contain fields – name/value pairs
- Fields can nest
- Value types include text, numerics, dates, and geo objects
- Field values can be single or array
- When you send documents to Elasticsearch they should arrive as JSON

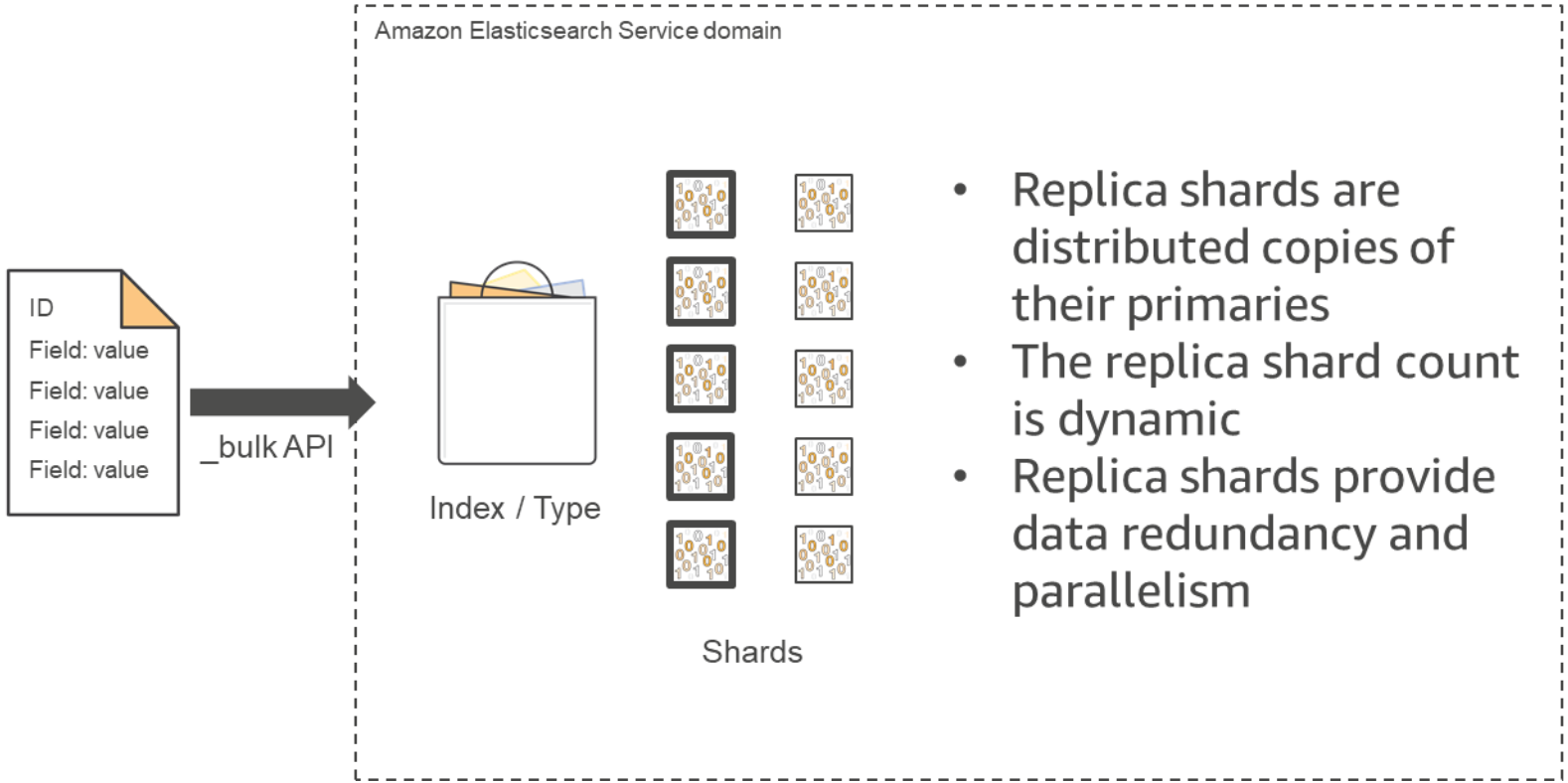
Amazon ES stores documents in an index

Indexes have primary shards



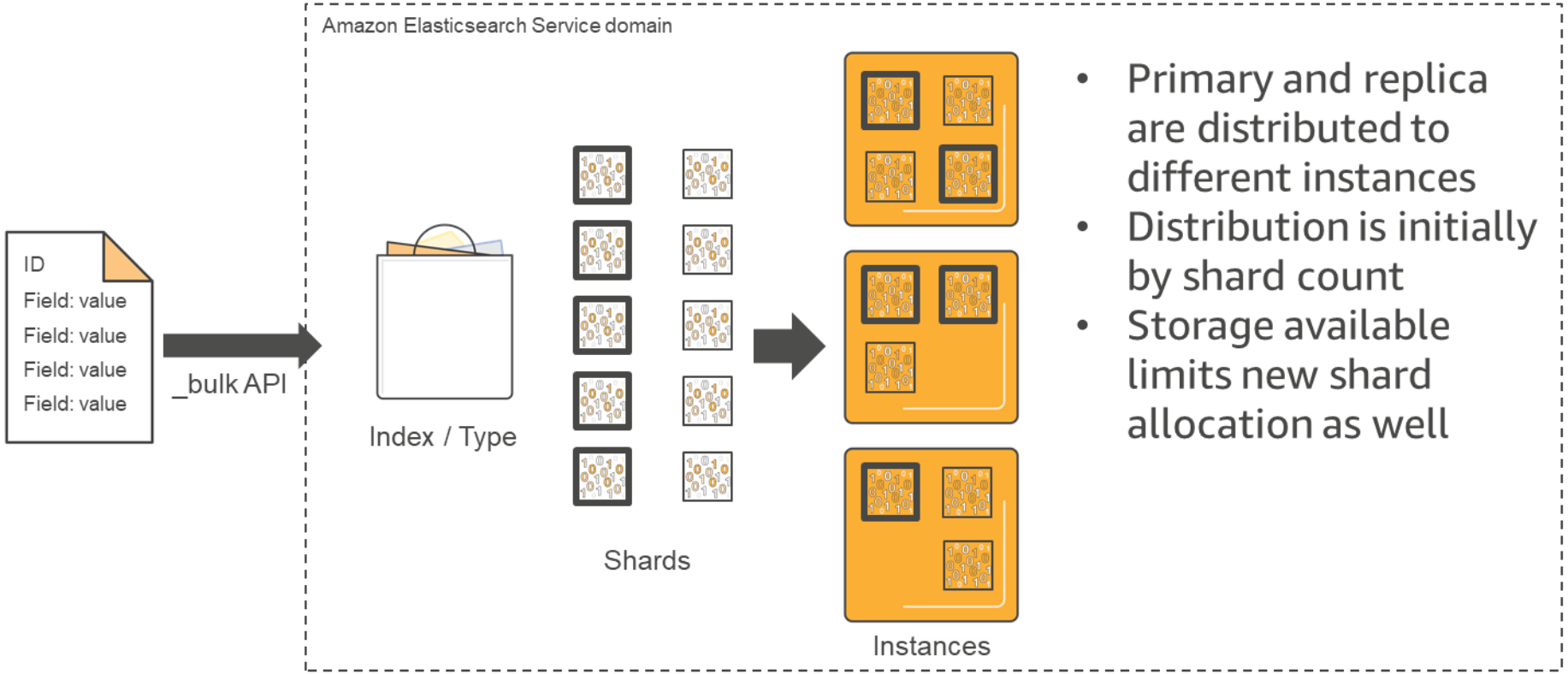
- Shards are the units of storage and compute in an Amazon ES domain
- Each shard's set of documents is distinct
- The primary shard count is fixed at creation time
- Shards are instances of Apache Lucene
- Lucene creates an index for each field in each document

Indexes have replica shards



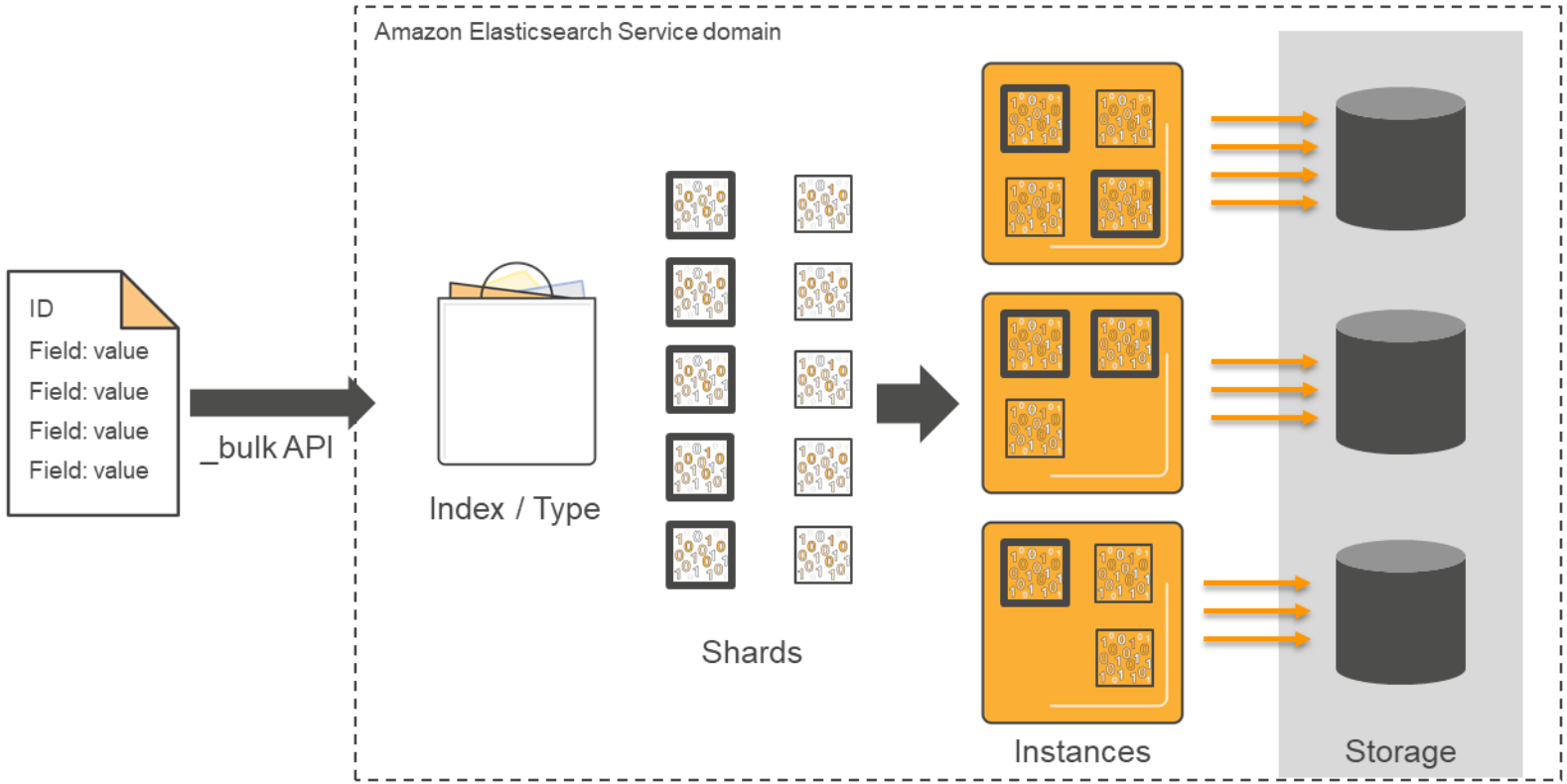
- Replica shards are distributed copies of their primaries
- The replica shard count is dynamic
- Replica shards provide data redundancy and parallelism

Shards are distributed across instances



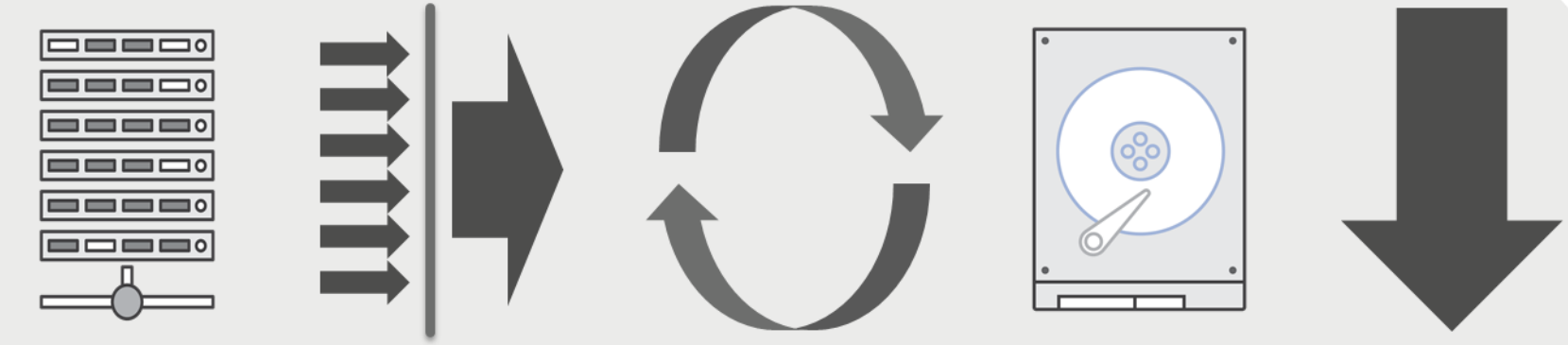
- Primary and replica are distributed to different instances
- Distribution is initially by shard count
- Storage available limits new shard allocation as well

Shards (Lucene) store indexes on disk

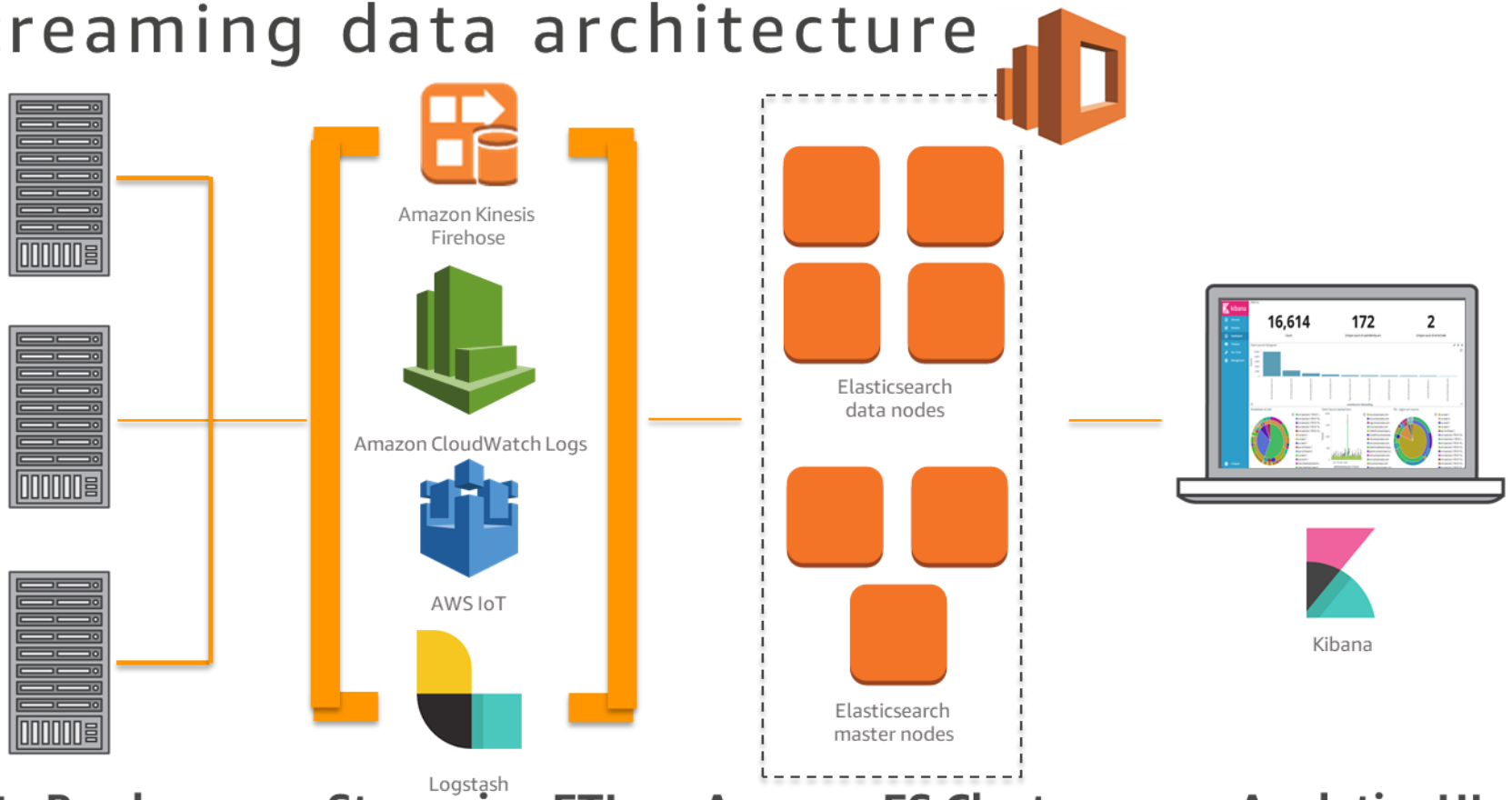


- ✓ Foundation
- Ingest your data
- Size your domain
- Secure your domain
- Add durability
- Monitor your domain
- Analyze your data

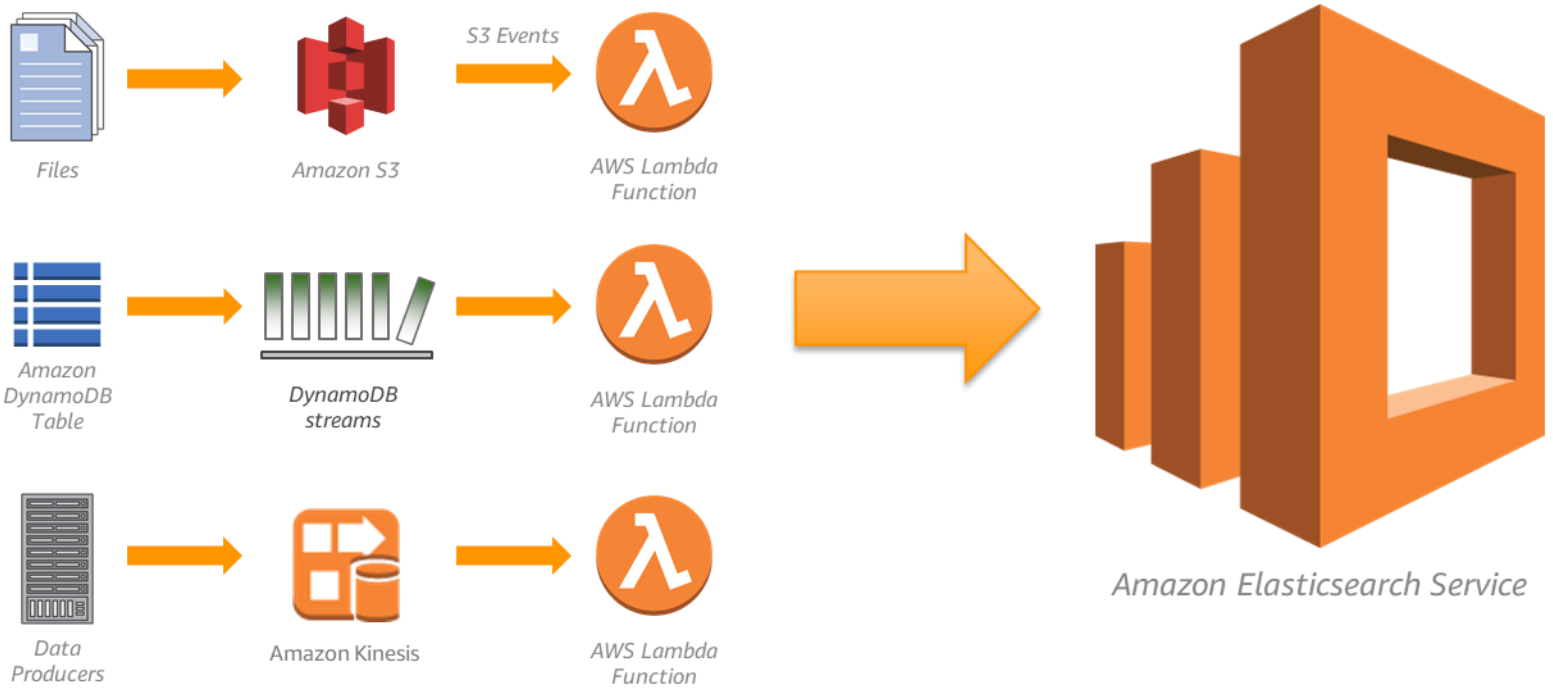
The components of ingestion



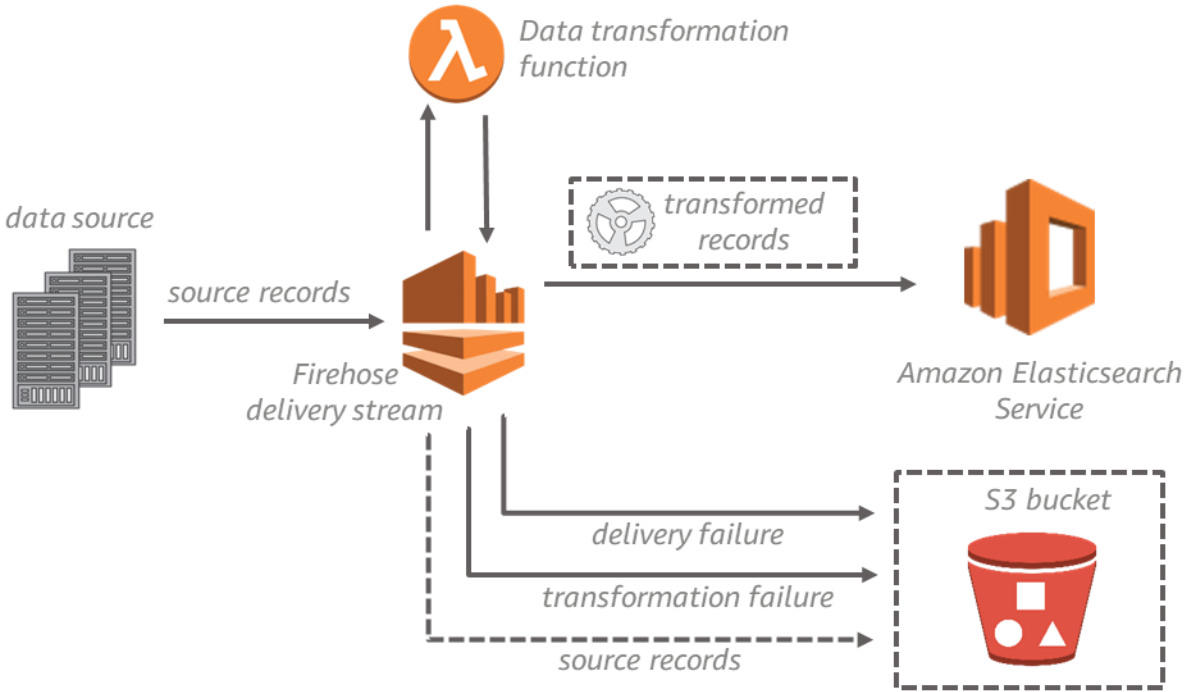
Streaming data architecture



Amazon Lambda to Amazon ES



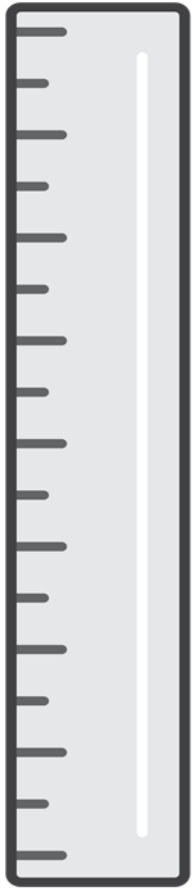
Kinesis Firehose delivery architecture



- For public access domains
- Easily transform data
- Serverless with built-in batching, index rollover, error handling

- ✓ Foundation
- ✓ Ingest your data
- Size your domain
- Secure your domain
- Add durability
- Monitor your domain
- Analyze your data

Use storage for sizing when creating a new domain



How many instances?

- The index size will be about the same as the corpus of source documents
  - **Double this** if you are deploying an index replica
- Size based on storage requirements
  - Either local storage or up to 1.5 TB of Amazon Elastic Block Store (EBS) per instance

Example: a 2 TB corpus will need 4 instances

Assuming a replica and using EBS

Given 1.5 TB of storage per instance, this gives 6TB of storage

How many shards?

- Best practice: shards should be < 50GB
- Divide index size by ~40GB to get initial shard count
- Active shards per instance ~= vCPUs
- Always use at least 1 replica for production!

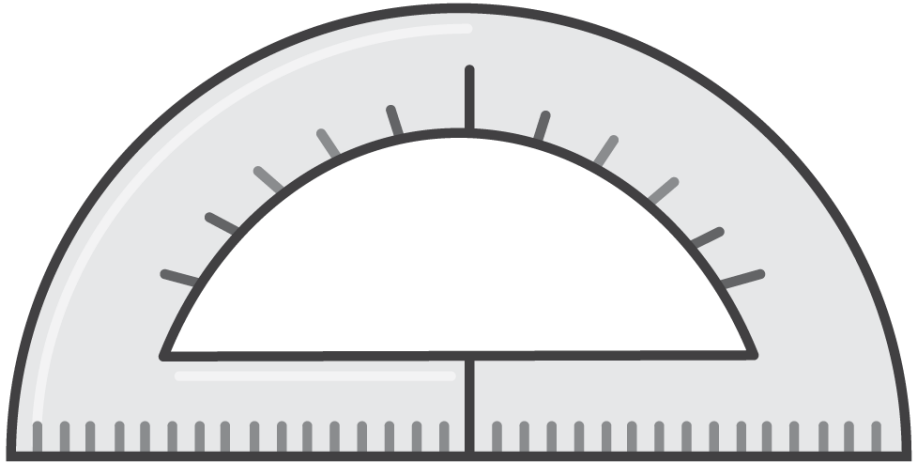
Example: 2 TB corpus will need 50 shards

2,000 GB / 40GB per shard = 50 shards

Which instances?

Instance	Workload
T2	Entry point. Dev and test. OK for dedicated masters.
M3, M4	Equal read and write volumes.
R3, R4	Read-heavy or workloads with high memory demands (e.g., aggregations).
C4	High concurrency/indexing workloads
I2	Up to 1.6 TB of SSD instance storage.

Adjust based on workload



Best practices for right-sizing

Write-heavy (streaming)

- Pay attention to concurrency!
    - Writes: all shards and replicas
- ++Index.refresh_interval, more capacity
- Increase primary shard count and data instances for more parallelism

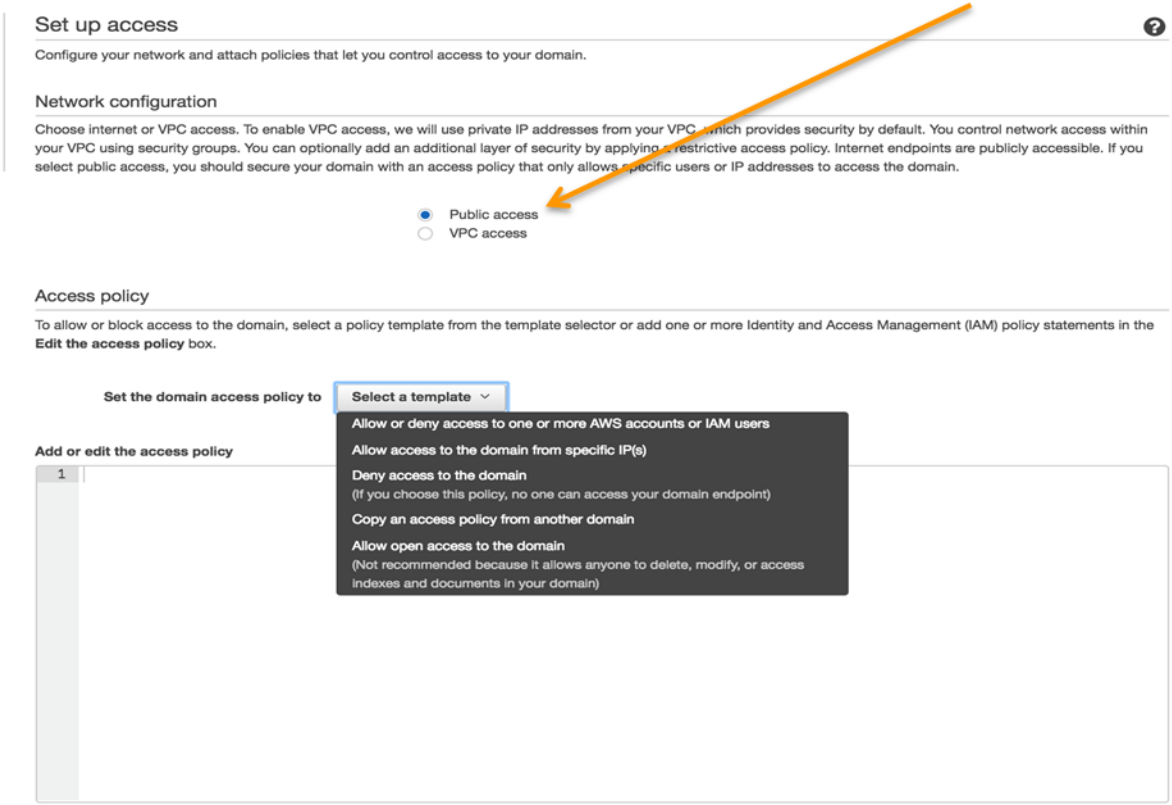
Read-heavy (full text)

- Pay attention to concurrency!
    - Reads: one of each shard
- Increase replica shard count and data instances for more parallelism

- Monitor, monitor, monitor!
- Test for shards per node; Estimate 1 shard per vCPU for best throughput
- Elasticsearch places shards based on count per node; Beware unbalanced storage

- ✓ Foundation
- ✓ Ingest your data
- ✓ Size your domain
- Secure your domain
- Add durability
- Monitor your domain
- Analyze your data

- Choose public access or VPC
- Public access domain endpoints are on the internet
- IAM policies for access control



We strongly recommend against using an “open access” policy

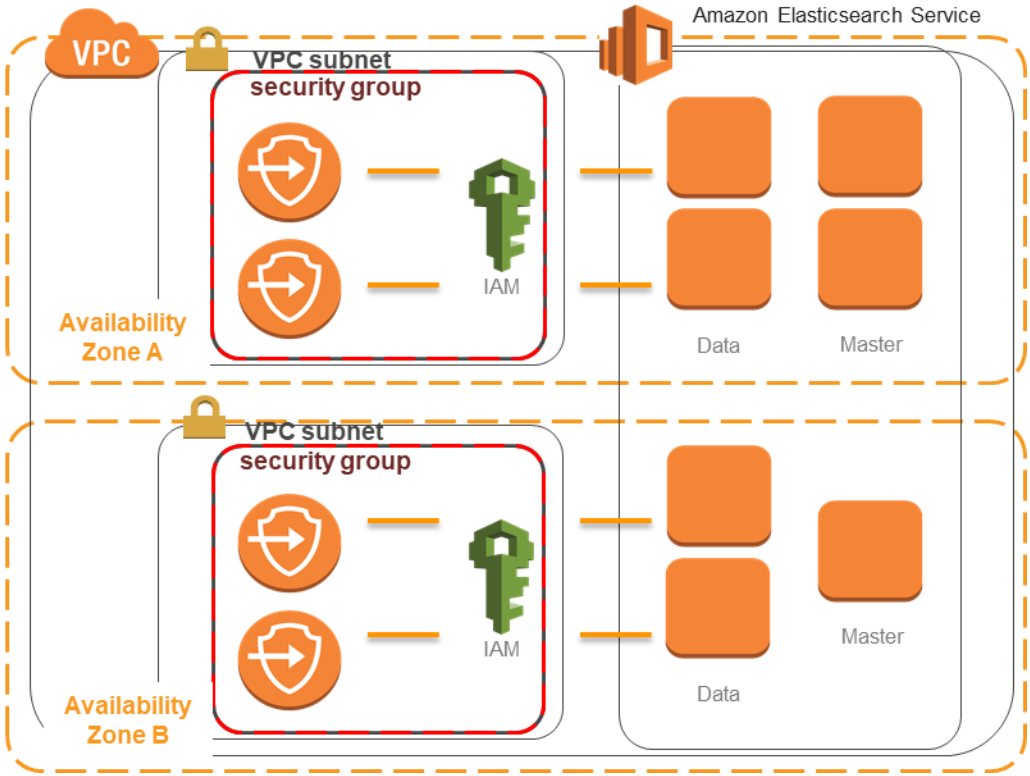
Policy skeleton

```
{
    "Version": "2012-10-17",
    "Statement": [ {
        "Effect":...
        "Principal": ...
        "Action": [...],
        "Resource": ...,
        "Condition": ...
        } ]
}
```

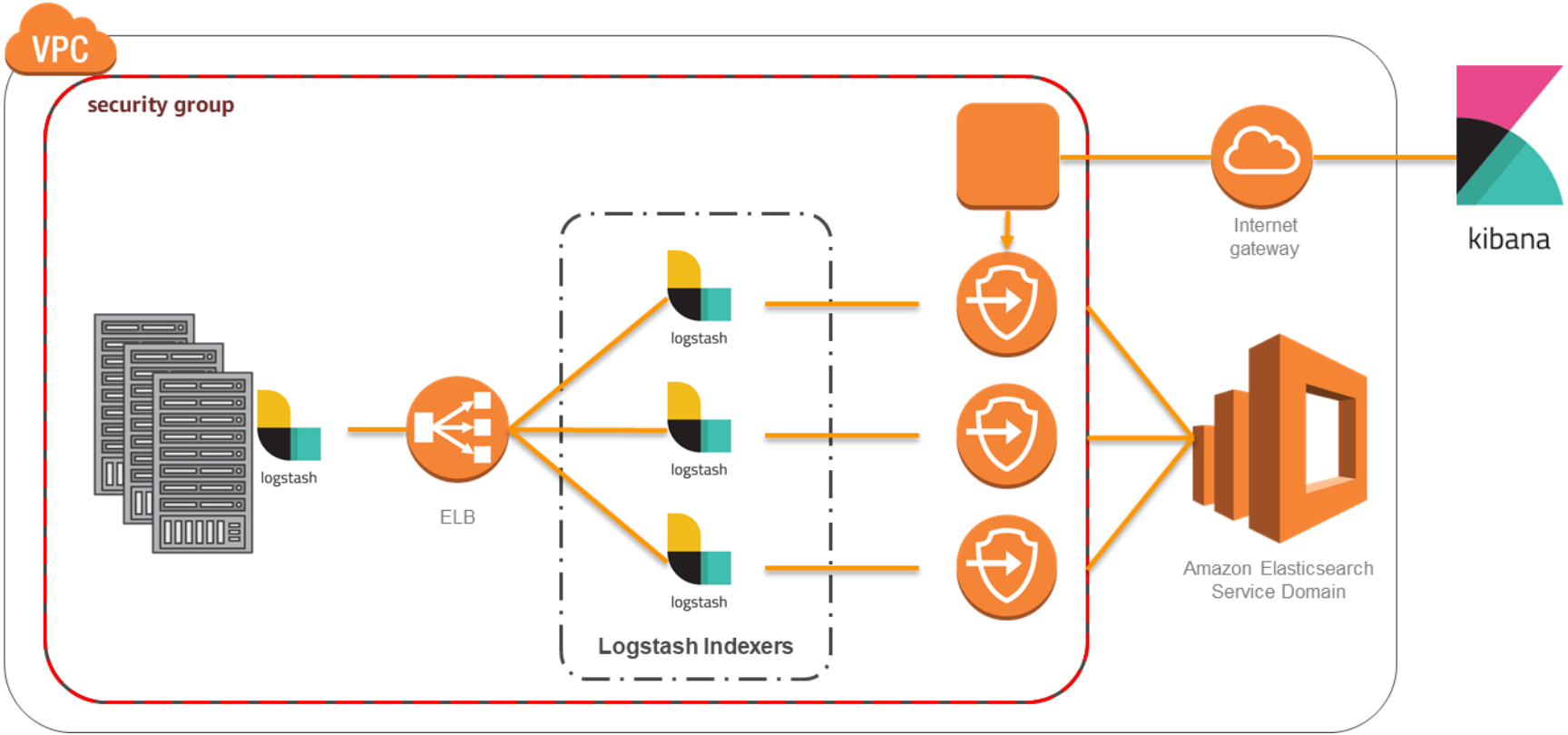
- Effect: Allow or Deny
- Principal: AWS account ID
- Action
  - HTTP verbs
  - Service actions
- Resource: Amazon ES domain/index
- Condition: IP Address

* NEW * Amazon Elasticsearch Service VPC Support

- Private networking between your VPC and Amazon Elasticsearch Service
- Traffic does not traverse the public internet
- Use IAM policies and security groups for authentication and access control

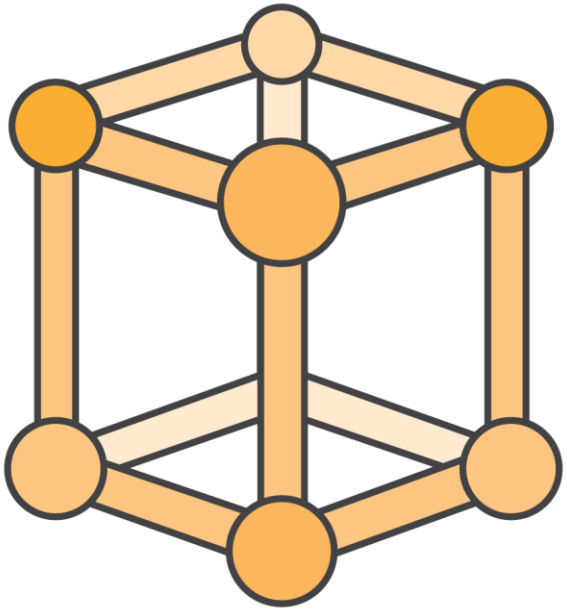


Send log data to Amazon ES in your VPC

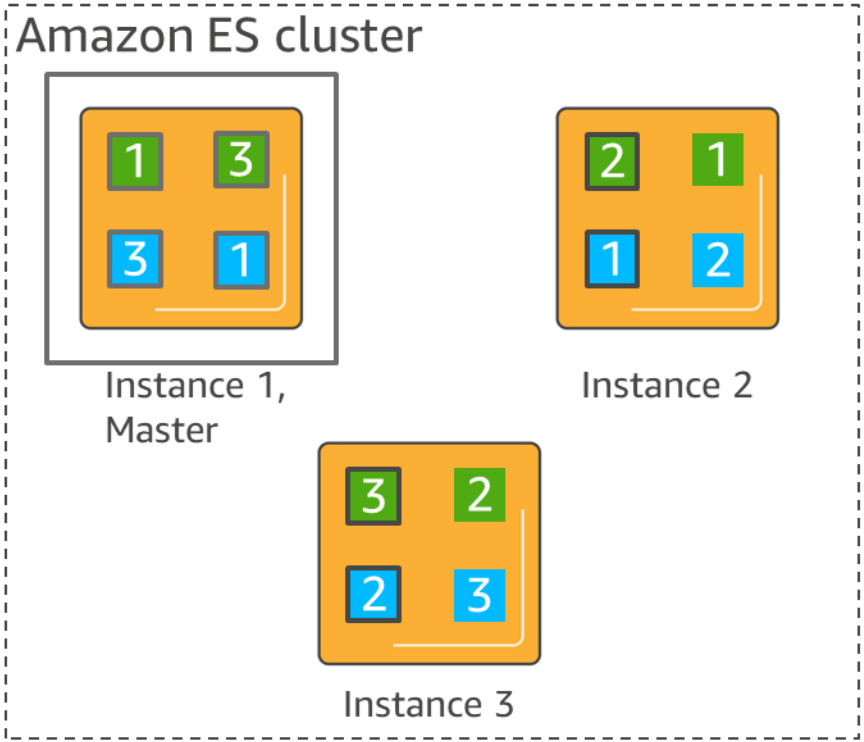


- ✓ Foundation
- ✓ Ingest your data
- ✓ Size your domain
- ✓ Secure your domain
- Add durability
- Monitor your domain
- Analyze your data

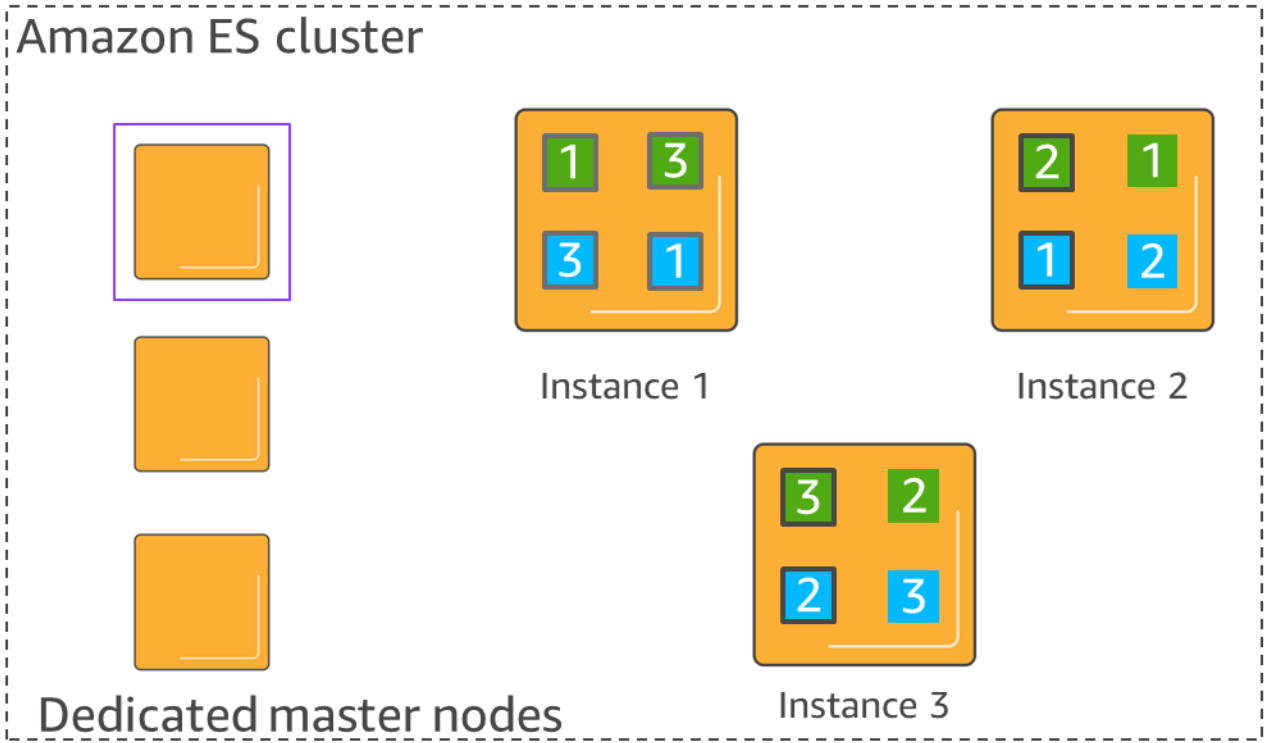
Dedicated master instances improve cluster stability



Cluster with no dedicated masters



Cluster with dedicated masters



Data nodes: queries and updates

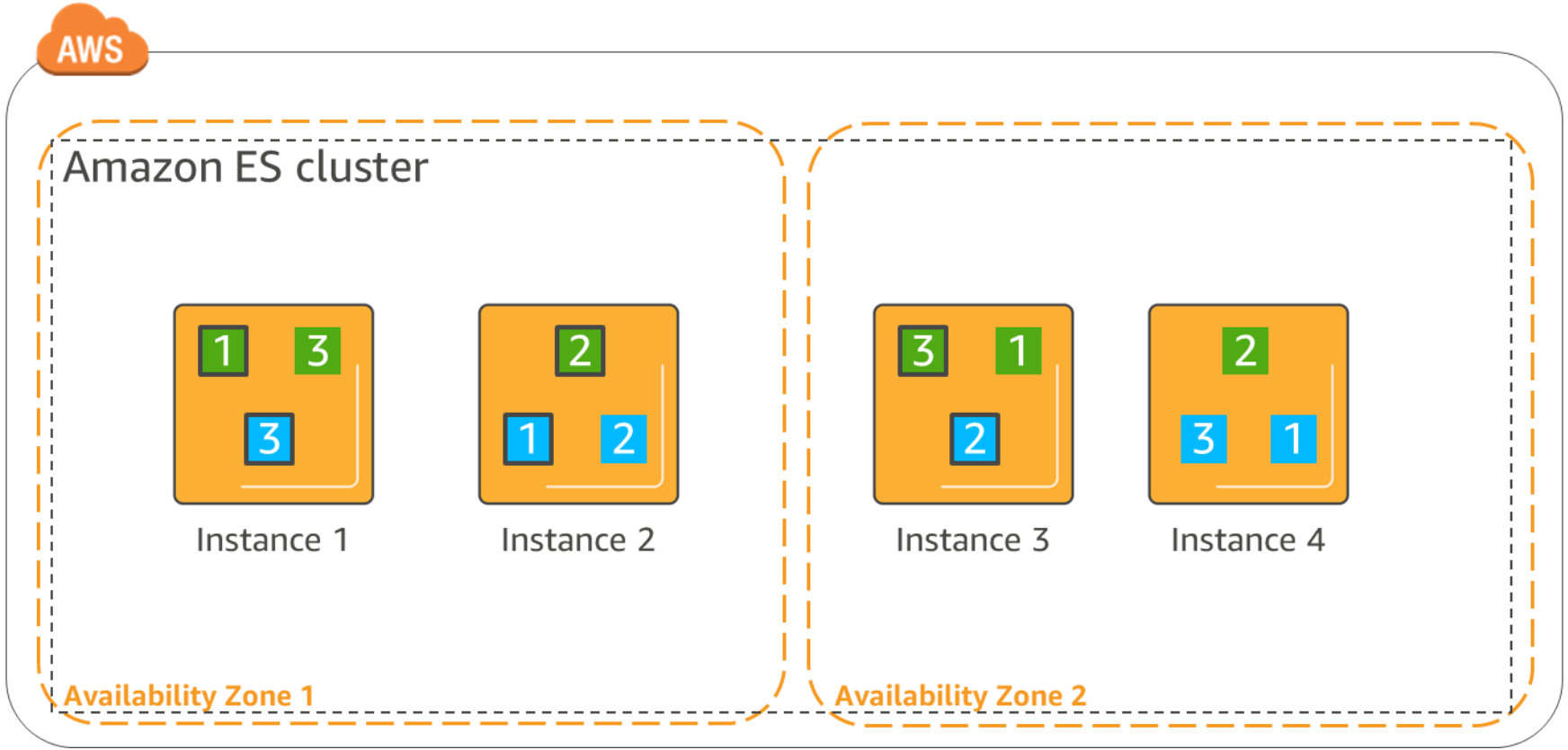
Master node recommendations

Number of data nodes	Master node instance type
< 10	m3.medium+
< 20	m4.large+
<= 50	c4.xlarge+
50-100	c4.2xlarge+

- In production, always use an odd number of masters, >= 3
- Master nodes can be smaller than data nodes

Zone awareness provides data redundancy in 2 zones

Cluster with zone awareness



- ✓ Foundation
- ✓ Ingest your data
- ✓ Size your domain
- ✓ Secure your domain
- ✓ Add durability
- Monitor your domain
- Analyze your data

CloudWatch 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

Slow Logs

- Easy console set up
- 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`

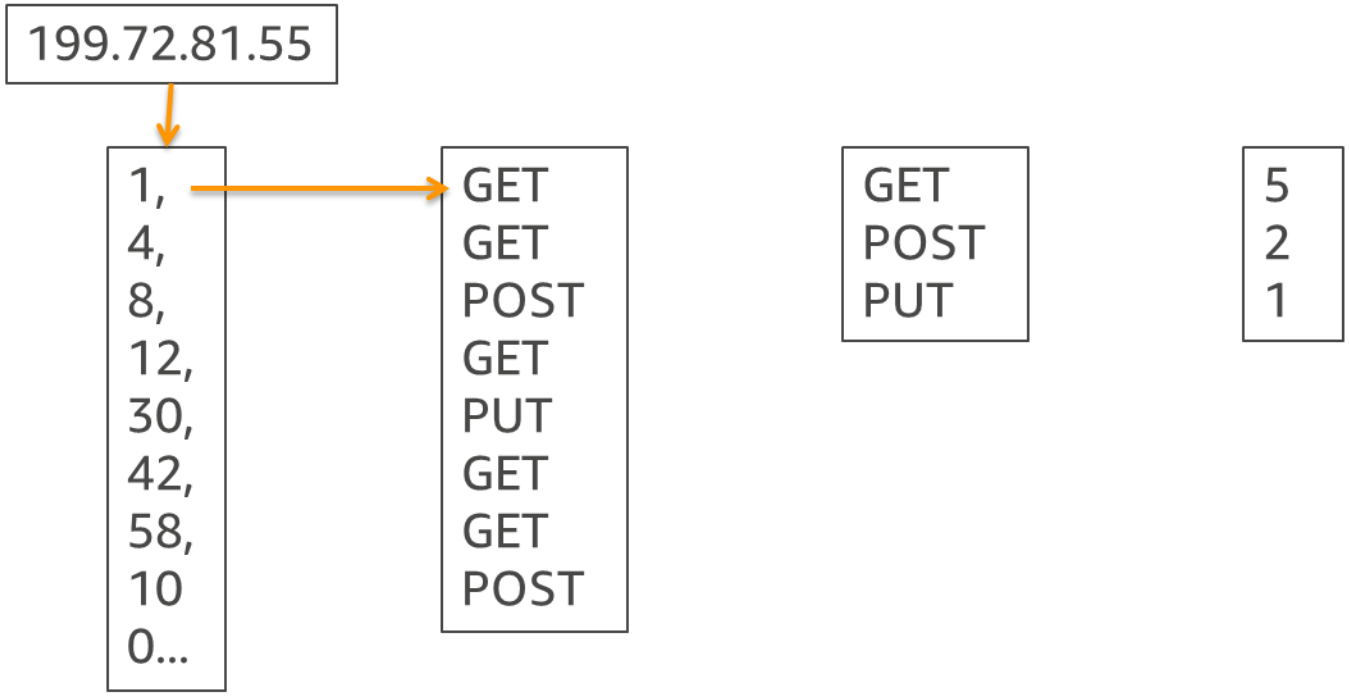
- `index.search.slowlog.threshold.fetch.warn`
- `index.search.slowlog.threshold.fetch.info`
- `index.search.slowlog.threshold.fetch.debug`
- `index.search.slowlog.threshold.fetch.trace`

- `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`

- ✓ Foundation
- ✓ Ingest your data
- ✓ Size your domain
- ✓ Secure your domain
- ✓ Add durability
- ✓ Monitor your domain
- Analyze your data

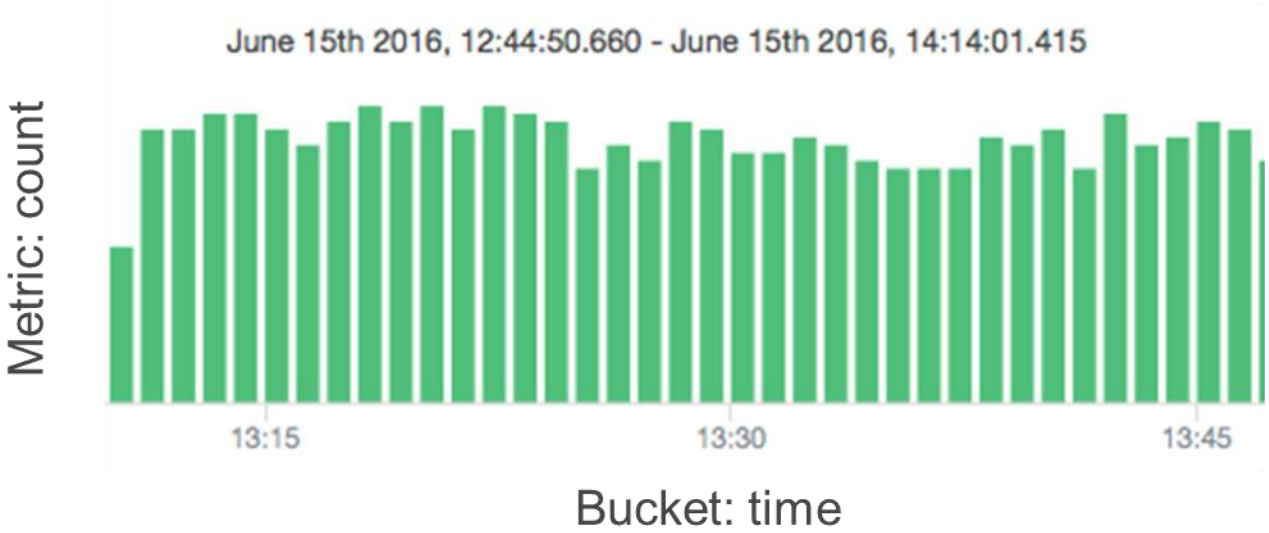
host:199.72.81.55 with <histogram of verb>



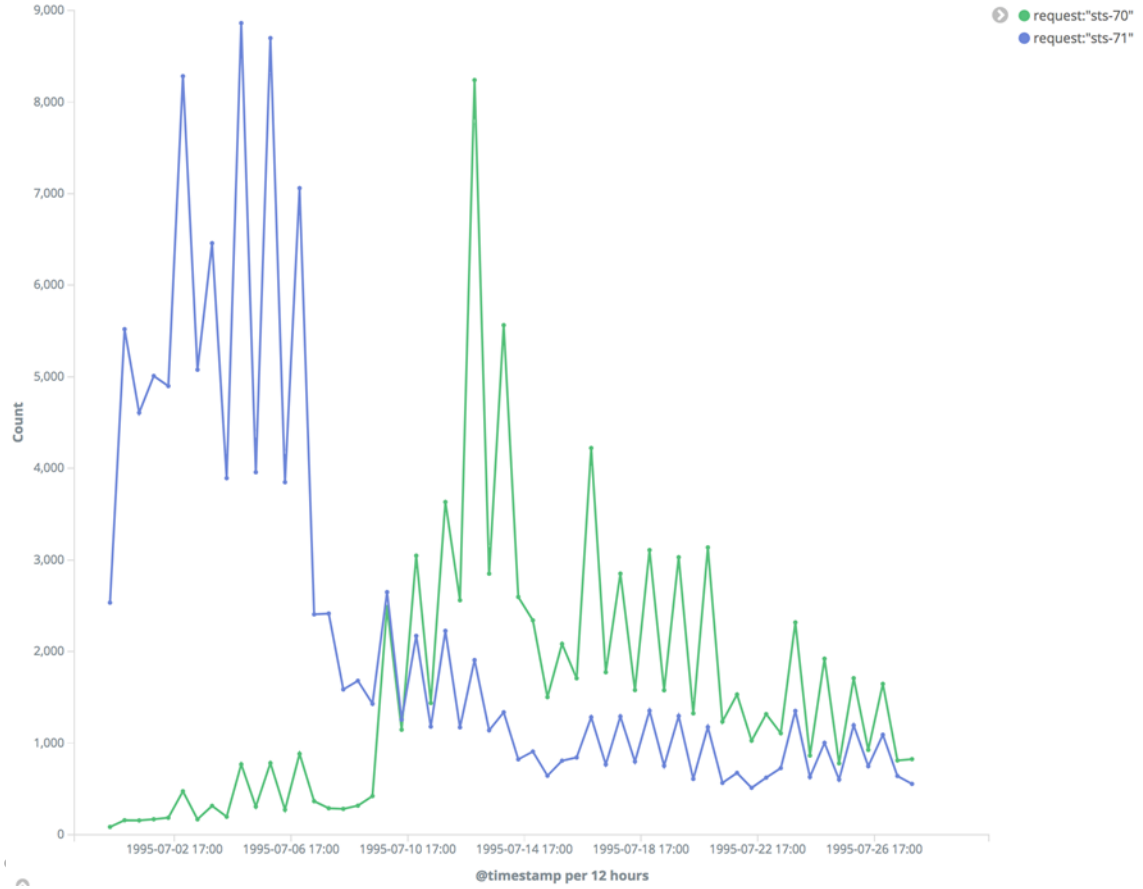
Amazon ES aggregations

Buckets – a collection of documents meeting some criterion

Metrics – calculations on the content of buckets



More complicated aggregations



Amazon Elasticsearch Service

Run Elasticsearch in the AWS Cloud with Amazon Elasticsearch Service

Deploy, scale, ingest, secure, monitor, and analyze

Start sending your log data today!

elasticsearch

kibana

AWS re:Invent

aws

AWS re:Invent

THANK YOU!