workday.

Cattle vs. Pets

A Cost-Effective Elasticsearch Architecture to Scale-Out Beyond Petabytes

Leonardo A. Dos Santos (L[e,é]o)

SRECON AMERICAS

Agenda

Who are we?

Why are logs and their aggregations so important for us?

Problem: Our Elasticsearch Scaling Story
> Search: Single View for Customers
> Ingestion: Sharding the Data into Multiple Clusters

Solution: Cattle vs. Pets | Elastic Cluster of Clusters Architecture

The Outcomes

Q&A

Who Are We? Observability

Leonardo Santos

Sr. Distributed Systems Engineer

Workday

Mark Levins

Principal Distributed Systems Engineer

Workday

Valerio Aputini

Principal Distributed Systems Engineer

Workday

Aderval Mendonça

Software Development Engineering Manager

Workday

Why are aggregated logs important for Workday?

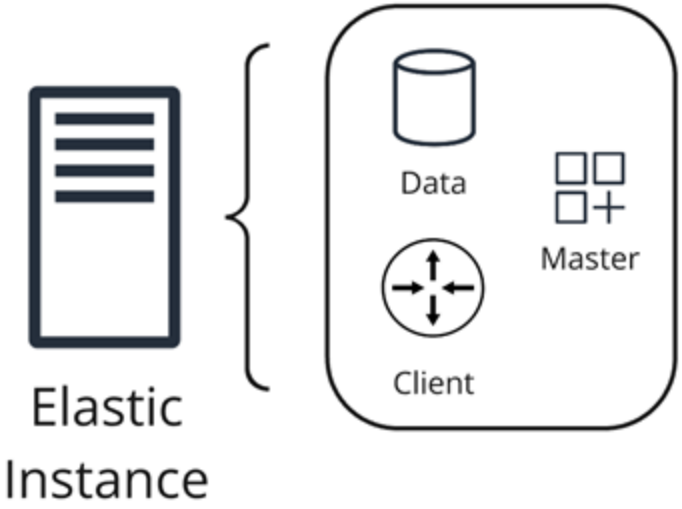
- Thousands (>250k producer hosts) of customers trust Workday to run some of their critical software (HR, Finance, etc)
- Workday Support teams have quite aggressive SLAs & SLOs
- Challenging operating scale for logs:
  - 10s of petabytes and trillions of unstructured indexed logs per month
  - Avg. ~5% growth every month (2x every 18 months)

The Origin of a Scaling Problem:
Our Elastic Scaling Story

Our Elastic Scaling Story

Initial stages before multi-cluster

Everyone starts with a PoC/MVP

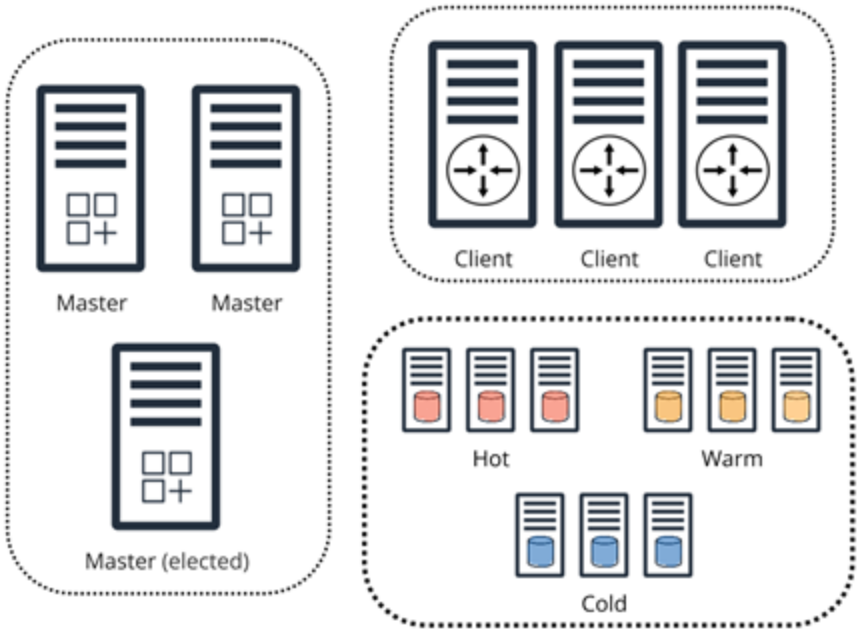


Our Elastic Scaling Story

Initial stages before multi-cluster

Then progress to a **single cluster**

- **Vertical** (up) Scaling sounds as a way to go… Until it's not!
- **Horizontal** (out) Scaling is the next, but you **reach the limit of data nodes** (↑ intra-cluster communication)
- **Vertical** scaling **again**, but inside **single cluster**



Our Elastic Scaling Story

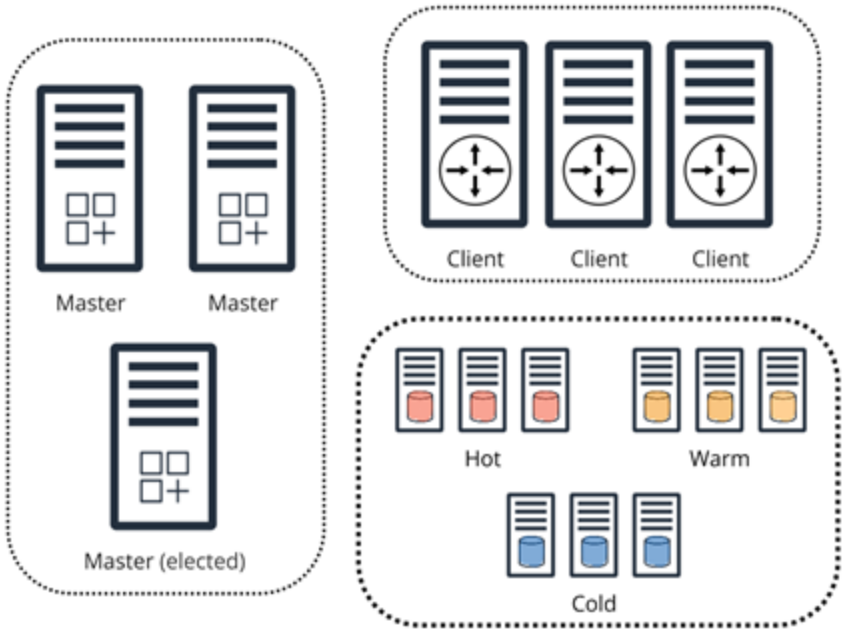
How to keep scaling?

Single-cluster limits are reached

- Intra-cluster communication & Single Master Limit
- Elastic's recommendation to avoid >200 nodes

How to keep scaling?

- Shift to a multi-cluster approach (linear)
- Split the problem in two:
  - Presentation (Search)
  - Data Ingestion



The Search Path

A Single View for Customers

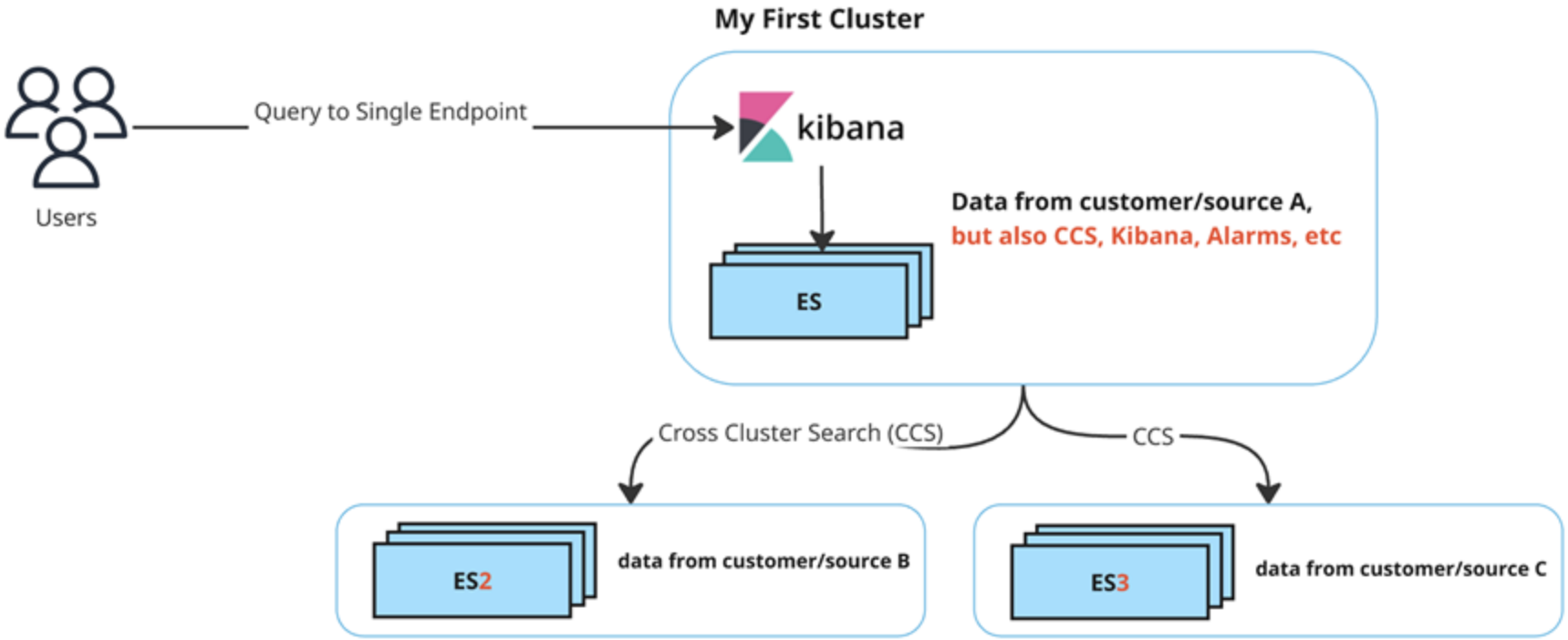
Our Elastic Scaling Story

The Search Path | A Single View for Customers

- **Cross Cluster Search (CCS)** allows one cluster as search entry point
- Then, this **first single cluster** becomes critical, handling **customer data** and **presentation metadata** (e.g., alarms, dashboards, saved objects, etc)
- **Heterogeneity introduces** complexity in operations, with numerous exceptions (if/else) in code

Our Elastic Scaling Story

The Search Path | A Single View for Customers

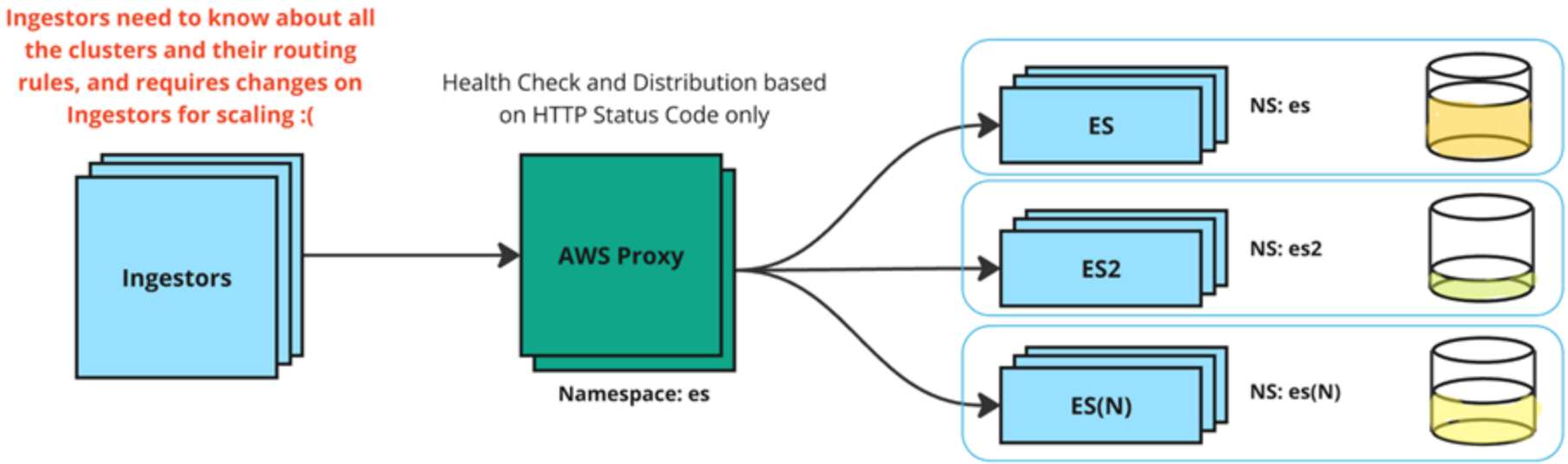


The Ingestion Path

Splitting the Data in Multiple Linear Clusters

Our Elastic Scaling Story

The Ingestion Path | Splitting the Data in Multiple Linear Clusters



Our Elastic Scaling Story

The Ingestion Path | Splitting the Data in Multiple Linear Clusters

- Same **heterogeneity issues** as in the Search Path
- **Source aware routing** paths to reach data clusters directly with **no centralized strategy**
- **Traffic imbalances** causing **hot-spots** across clusters (e.g., one cluster handling 90% of the data)
- Still, specific data source can grow and their **cluster will hit the data nodes limit** (>200 nodes)

Our Elastic Scaling Story

The Ingestion Path | Splitting the Data in Multiple Linear Clusters

Adding any new cluster causes:

- **Naming convention** issues
- **Multiple deployment models**
- Unsustainable **increase in operational load**
- With the constant onboarding of new logs and use cases

→ **Urgency for re-design!**

Cattle vs. Pets

An Elastic Cluster of Clusters Architecture

Cattle vs. Pets

An Elastic Cluster of Clusters Architecture

Requirements

- **Horizontal scalability** for the **next 10 years**
- **Separation** of **Data** and **Presentation** layers
- Support for **specialized cluster profiles**
- **Cost-effectiveness**
- Infrastructure-as-Code Agnosticism
- Standard **Naming Conventions**
- Keep **Stupid Simple**
- **Self-healing**

Cattle vs. Pets

An Elastic Cluster of Clusters Architecture

Load-Balancer

- **New LB** layer with auto-scaling for HAProxy servers
- HAProxy with redirects to special cluster profiles
  - Redirects based on **HTTP headers** or **Paths**
  - **Not tagged** traffic redirected to the **Generic Data Cluster-of-Clusters**
- **TCP Health-Check Service** scores clusters based on the USE Method (Utilization, Saturation, Errors) *[Brendan Gregg]*

Load-Balancer

LB routes and balances the traffic to the data clusters based on the healthiness scores

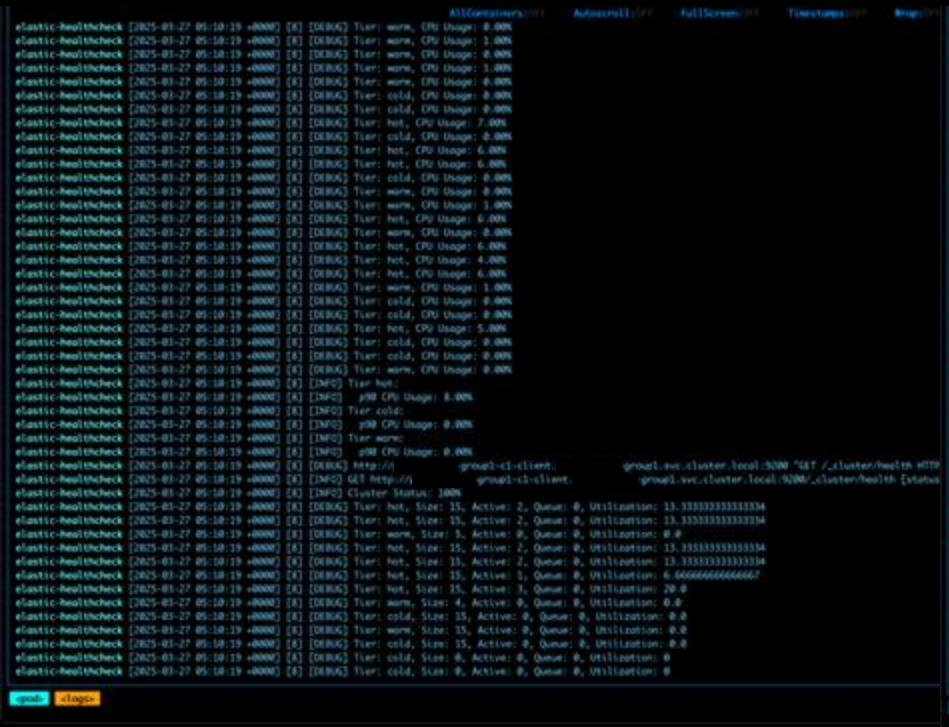
$$(a)\ cpu \ = \ 100 \ - \ \frac{p90(usage(\%)) + p90(threadpools(\%))}{2}$$

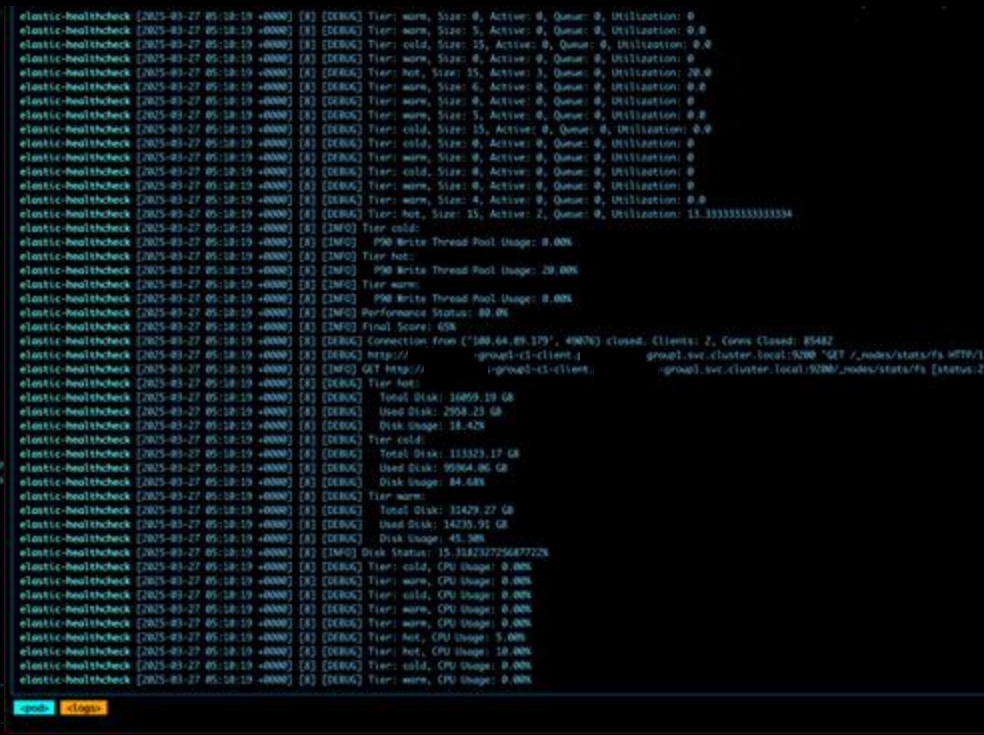
$$(b)\ disk \ = \ MIN(freespace(\%))$$

$$(c)\ clusterstatus \ = \ (red \ = 0\ ,\ yellow \ = 50\ ,\ green \ = 100)$$

$$score \ = clusterstatus\ if\ (clusterstatus \ == \ 0)\ else\ \frac{a+b+c}{3}$$

TCP Health-Check Service





```
backend group1_backend
  balance roundrobin
  timeout check 5s
  server group1-c1 group1-c1-client.group1.svc.cluster.local:9200 check weight 100 agent-check agent-addr group1-c1-es-healthcheck.group1.svc.cluster.local agent-port 9600 agent-inter 5s
  server group1-c2 group1-c2-client.group1.svc.cluster.local:9200 check weight 100 agent-check agent-addr group1-c2-es-healthcheck.group1.svc.cluster.local agent-port 9600 agent-inter 5s
  ...
```

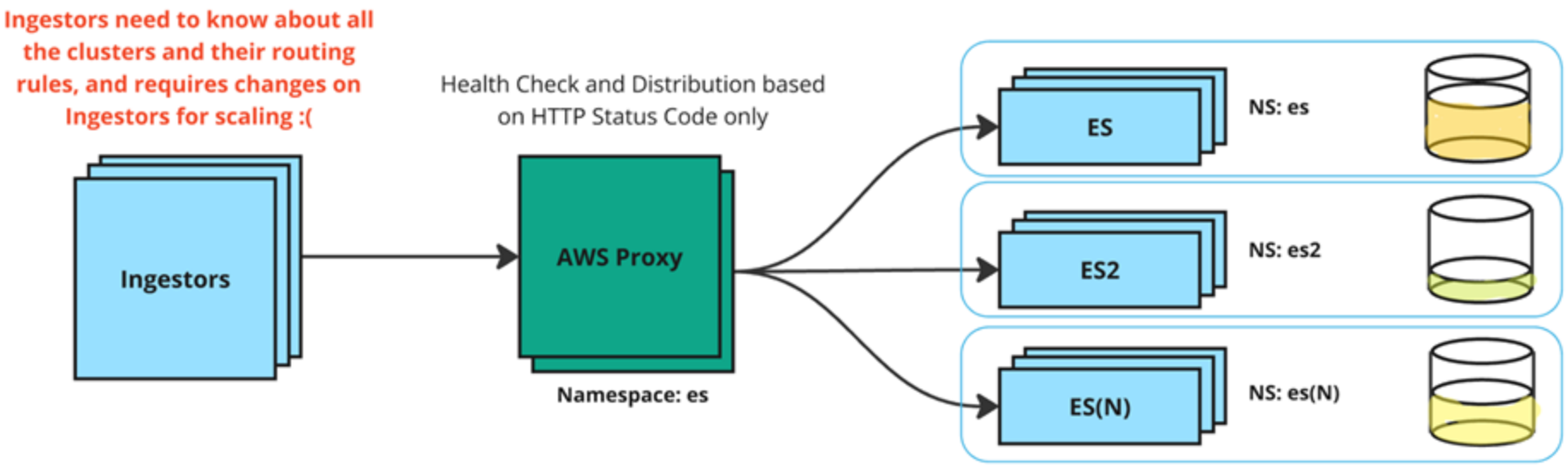
Management Cluster

- A **single namespace** with **minimal Elastic cluster** acting as the **entry point for Search and Ingestion**
- **Doesn’t store real customer data**
- Keeps all the **centralized tooling** and **services** (e.g., LB, Kibana, Cross-Cluster-Search, Search-Path Metadata, Auditor Service)
- **LB layer routes** traffic to the Data Clusters **using a USE Weighted Round-Robin** method
  - TCP Health-check Service runs on every cluster

Cattle vs. Pets

An Elastic Cluster of Clusters Architecture

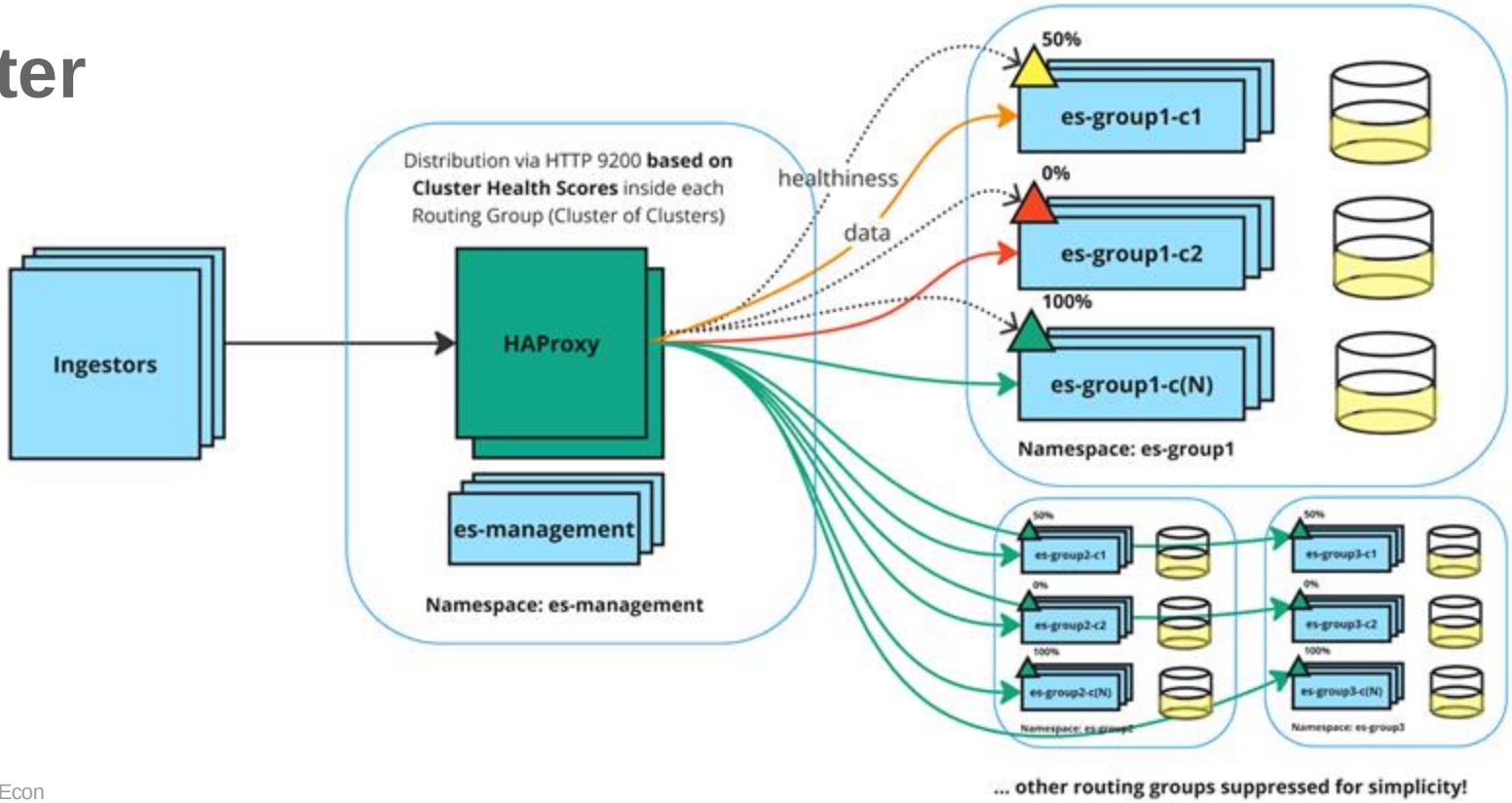
Before

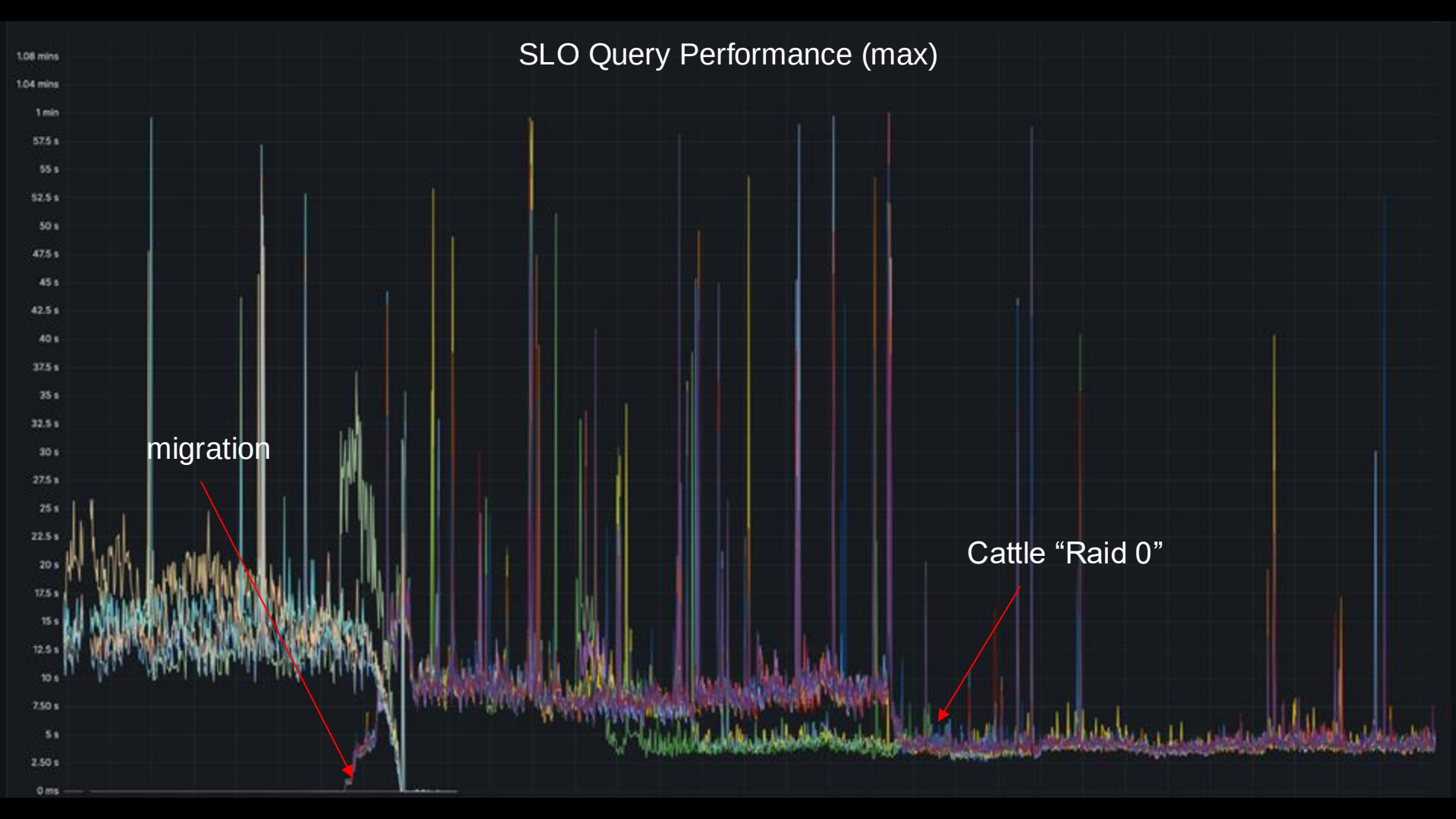


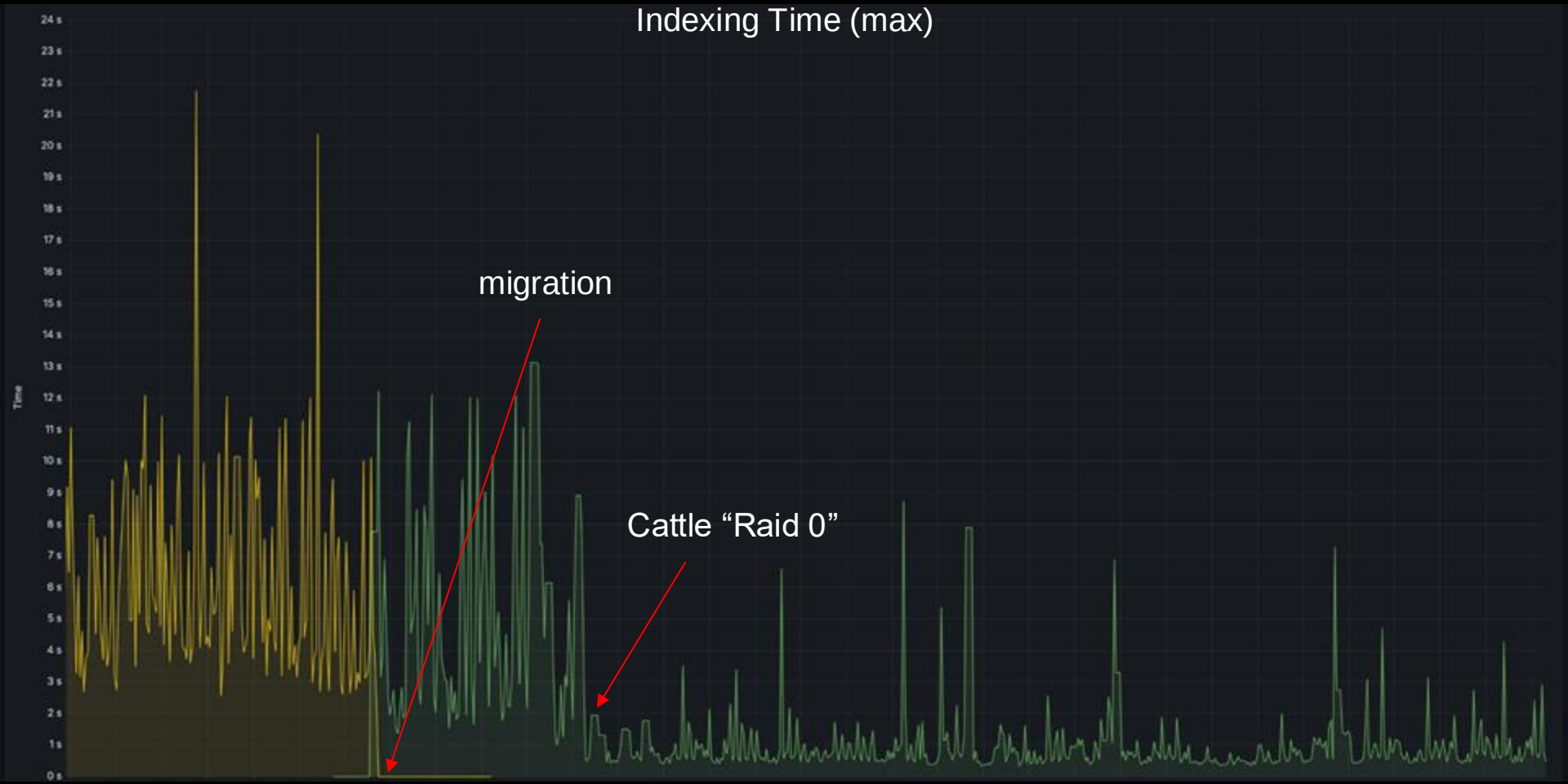
Cattle vs. Pets

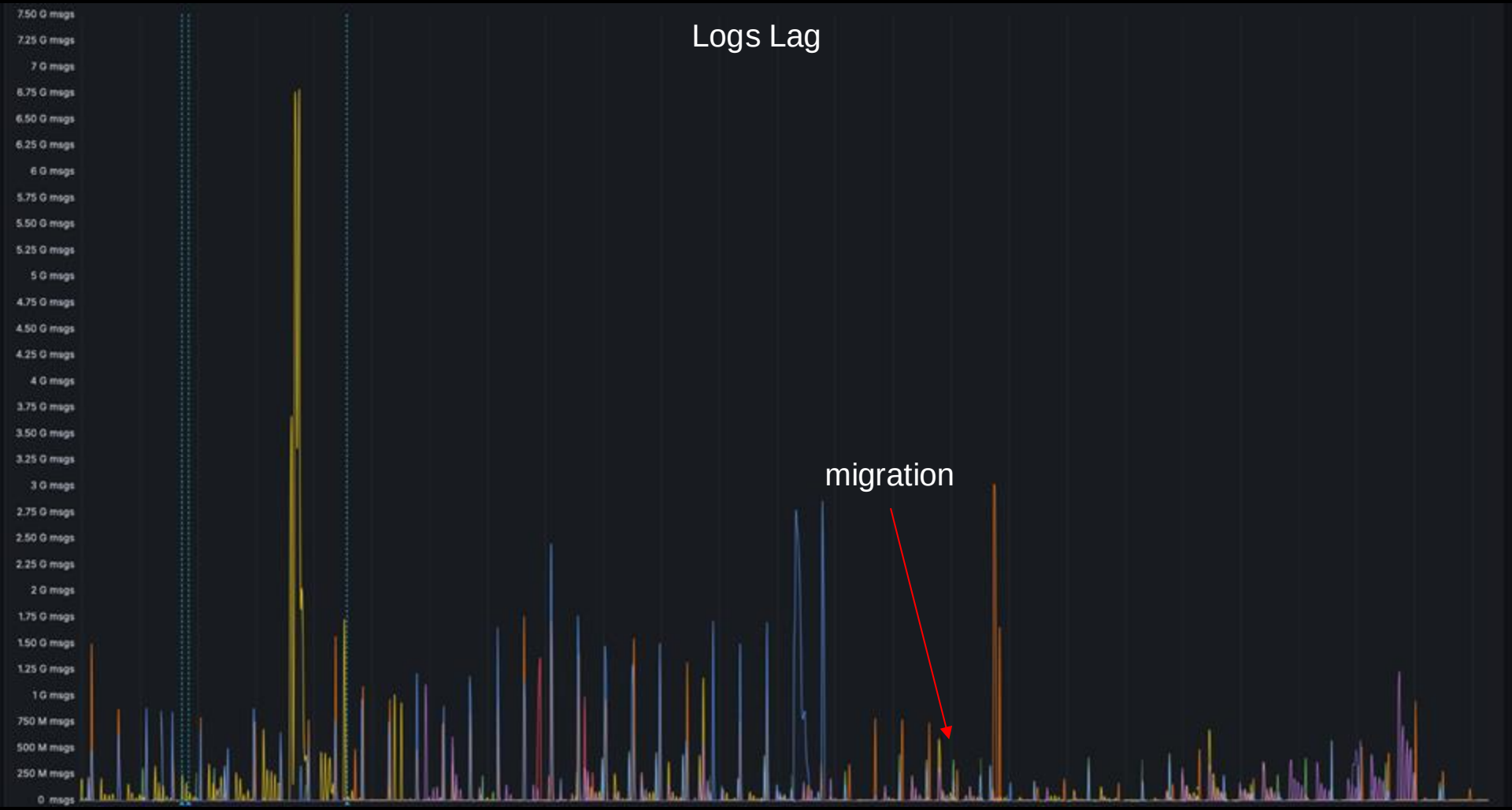
An Elastic Cluster of Clusters Architecture

After




SLO Query Performance (max)
1.08 mins
1.04 mins
1 min
57.5 s
55 s
52.5 s
50 s
47.5 s
45 s
42.5 s
40 s
37.5 s
35 s
32.5 s
30 s
27.5 s
25 s
22.5 s
20 s
17.5 s
15 s
12.5 s
10 s
7.50 s
5 s
2.50 s
0 ms
migration
Cattle “Raid 0”

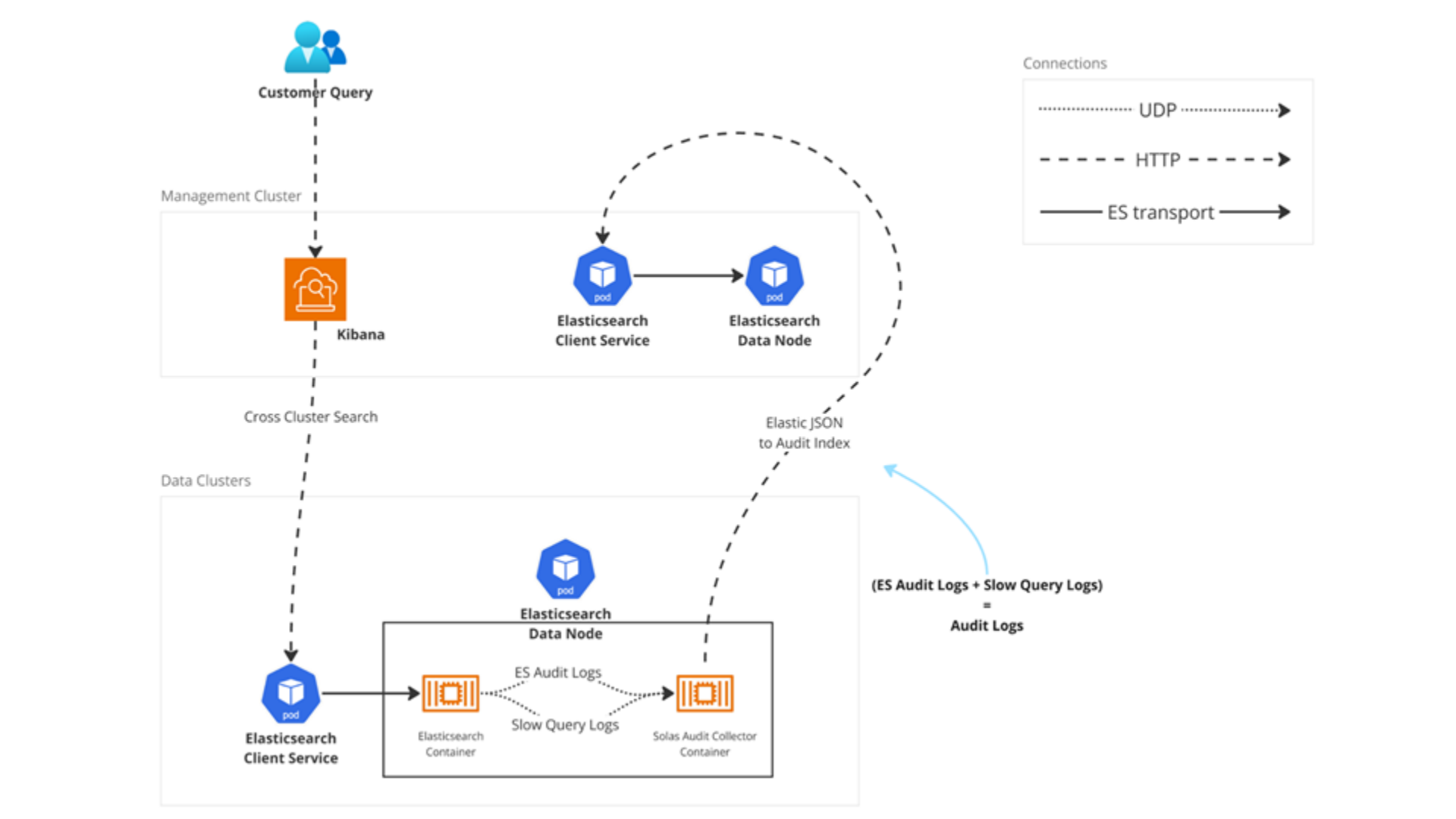

Indexing Time (max)
migration
Cattle “Raid 0”
Time
24 s
23 s
22 s
21 s
20 s
19 s
18 s
17 s
16 s
15 s
14 s
13 s
12 s
11 s
10 s
9 s
8 s
7 s
6 s
5 s
4 s
3 s
2 s
1 s
0 s

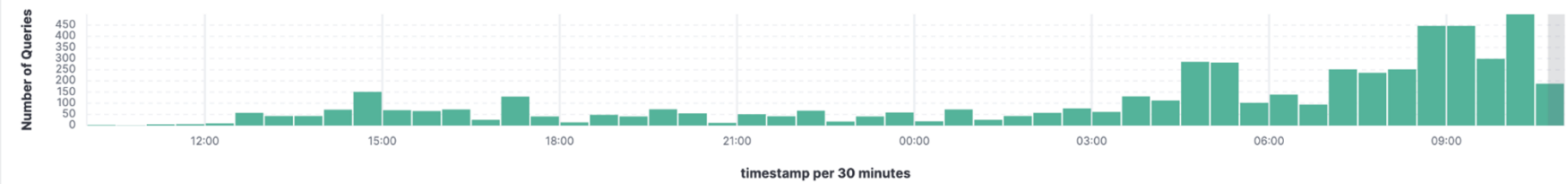

Logs Lag
migration
7.50 G msgs
7.25 G msgs
7 G msgs
6.75 G msgs
6.50 G msgs
6.25 G msgs
6 G msgs
5.75 G msgs
5.50 G msgs
5.25 G msgs
5 G msgs
4.75 G msgs
4.50 G msgs
4.25 G msgs
4 G msgs
3.75 G msgs
3.50 G msgs
3.25 G msgs
3 G msgs
2.75 G msgs
2.50 G msgs
2.25 G msgs
2 G msgs
1.75 G msgs
1.50 G msgs
1.25 G msgs
1 G msgs
750 M msgs
500 M msgs
250 M msgs
0 msgs

~17% savings in EC2 instances for largest Region (57% in smaller)

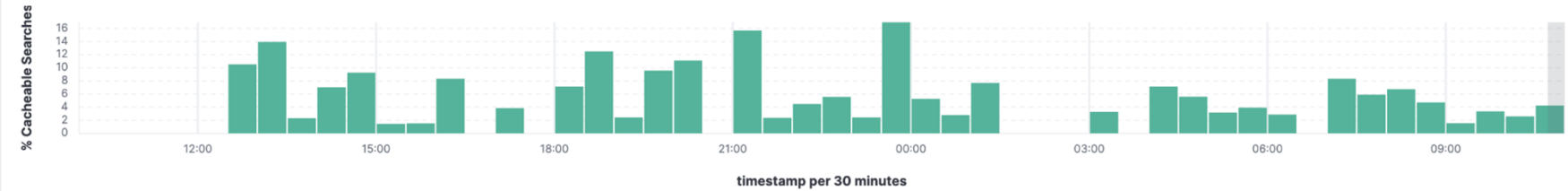
Elastic Auditing Service

- Elasticsearch **lacks** a **centralized** way to track **slow queries**, query stats, top offenders, etc., especially at this scale
- Sidecar UDP server aggregating audit and slow logs from data clusters & shipping to mgmt cluster
- Improves the end-to-end customer experience (query latency)
- Helps on analysis and troubleshooting of **performance issues across clusters**
- Paves the road for **more tooling** (e.g., rate-limiting of top offenders to meet our SLOs, FE Caching)

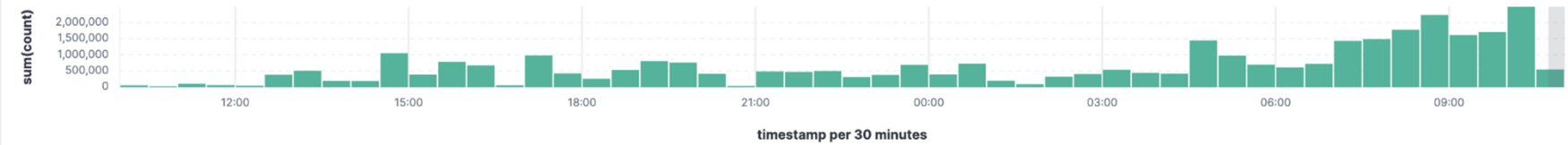

Customer Query
Management Cluster
Kibana
Elasticsearch
Client Service
Elasticsearch
Data Node
Cross Cluster Search
Elastic JSON
to Audit Index
Data Clusters
Elasticsearch
Data Node
ES Audit Logs
Slow Query Logs
Elasticsearch
Client Service
Elasticsearch
Container
Solas Audit Collector
Container
(ES Audit Logs + Slow Query Logs)
=
Audit Logs
Connections
UDP
HTTP
ES transport



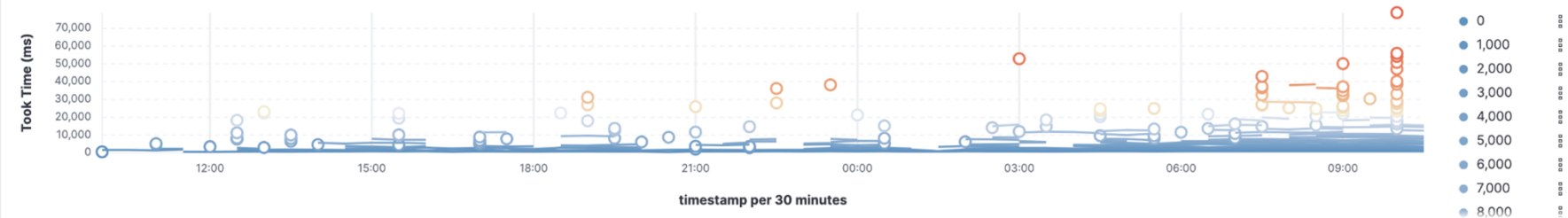
[redacted] Cacheable Searches



[redacted] Unique Sub-Queries ⓘ



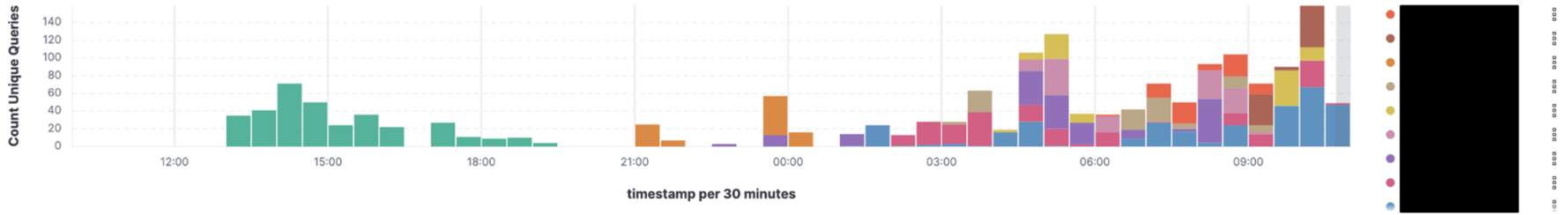
[redacted] Elapsed Time (legend = intervals)



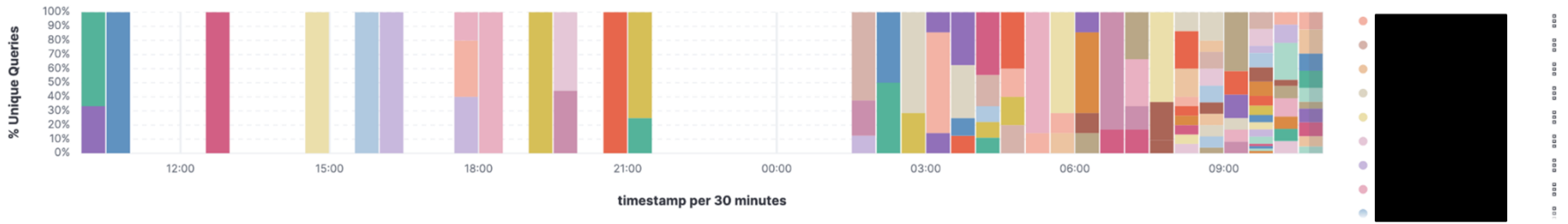
5435 documents

t query.raw_query	{"size":0,"query":{"bool":{"filter":[{"bool":{"should":[{"match_phrase":{"server_type":{"query":"authgwy","slop":0,"zero_terms_query":"NONE","boost":1.0}}},{"match_phrase":{"[redacted]":{"query":[redacted],"slop":0,"zero_terms_query":"NONE","boost":1.0}}}],"adjust_pure_negative":true,"minimum_should_match":"1","boost":1.0}},{"bool":{"should":[{"term":{"[redacted]keyword":{"value":"[redacted]","boost":1.0}}}],"adjust_pure_negative":true,"minimum_should_match":"1","boost":1.0}},{"term":{[redacted].keyword":{"value"[redacted]boost":1.0}}},{"range":{"@timestamp":{"from":"2025-03-24T12:42:52.812Z","to":"2025-03-24T17:42:52.812Z","include_lower":true,"include_upper":true,"format":"strict_date_
t query.term.[redacted]keyword	[redacted]
# query.track_total_hits	2,147,483,647
timestamp	Mar 24, 2025 @ 10:42:54.961
# took_time	2
# took_time_avg	16
# took_time_max	254
t took_time_max_index	[redacted]-2025.03.24-000302
t took_time_max_node	[redacted]-group1-c5-data-hot-b-21
# took_time_min	0
t took_time_min_index	[redacted]-2025.03.24-000306
t took_time_min_node	[redacted]-group1-c15-data-hot-c-12
# took_time_sum	1,137

[redacted] - Top 10 - Unique Queries Per User



[redacted] - Unique Queries Per User (%)

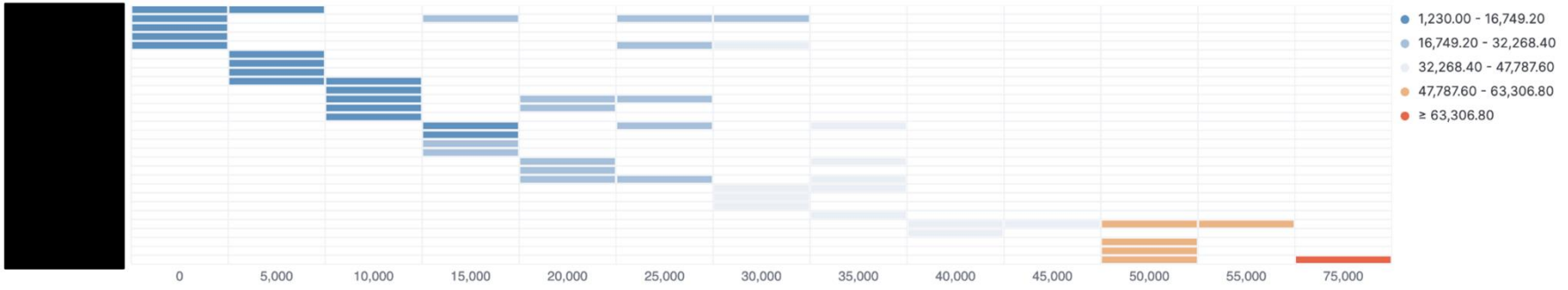


[redacted] Simple Search

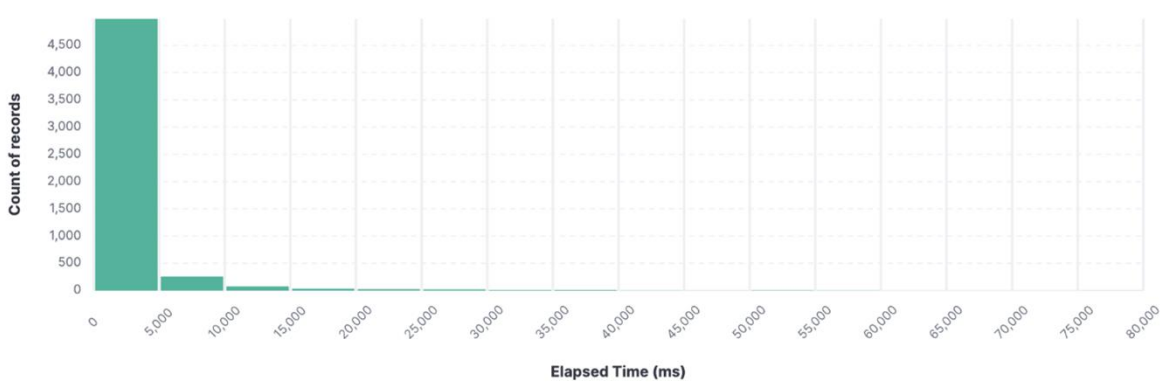
5435 documents

Time ↓	username	hits_sum	took_time_max	took_time_max_node	count	ip
Mar 24, 2025 @ 10:42:54.961	[redacted]	0	254	[redacted]-group1-c5-data-hot-b-21	70	[redacted]
Mar 24, 2025 @ 10:42:41.396		712	16	[redacted]-group1-c13-data-hot-c-18	340	
Mar 24, 2025 @ 10:42:39.835		0	21	[redacted]-group1-c14-data-cold-b-17	406	
Mar 24, 2025 @ 10:42:38.097		3,507	414	[redacted]-group1-c5-data-hot-c-14	1,078	
Mar 24, 2025 @ 10:42:15.187		33	127	[redacted]-group1-c14-data-hot-c-26	397	
Mar 24, 2025 @ 10:42:14.918		5	281	[redacted]-group1-c13-data-hot-c-15	39	
Mar 24, 2025 @ 10:42:13.924		1	7	[redacted]-group1-c13-data-hot-c-2	42	
Mar 24, 2025 @ 10:42:11.842		154,975	539	[redacted]-group1-c5-data-hot-c-29	1,716	

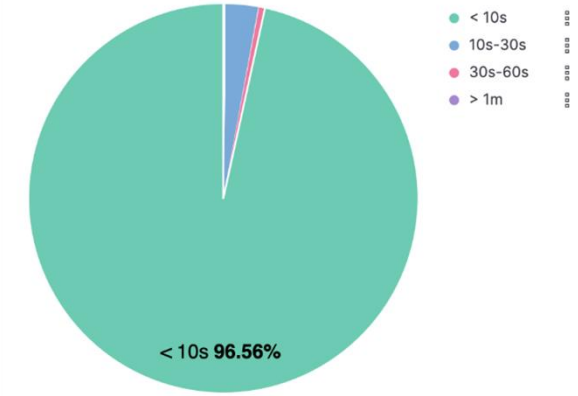
[redacted] - Users x Took Time



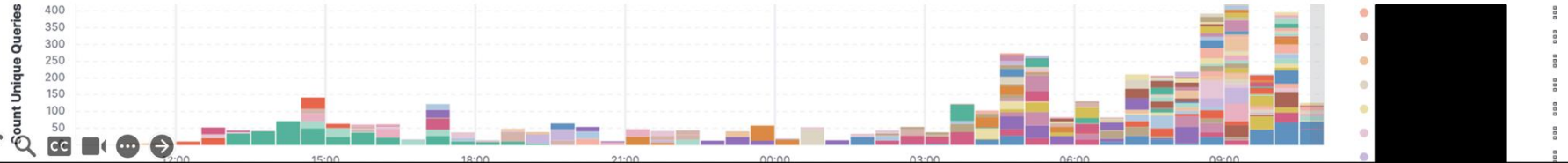
[redacted] - Query Time Distribution



[redacted] - Unique Queries Time Distribution Pie



[redacted] - Unique Queries Per User



Cattle vs. Pets

An Elastic Cluster of Clusters Architecture

Data Cluster of Clusters (CoC)

- Every Data-Cluster has the **same shape** (Disk, CPU, Mem (OS and JVM)) and is agnostic to the incoming data from the LB
- **Consistent Elastic Topology** for each **CoC**
- **Index and Shards Uniformly Defined**
- **No direct access** to Data Clusters for Search and Ingestion
- **Nothing beyond data** running on data-clusters

Cattle vs. Pets

An Elastic Cluster of Clusters Architecture

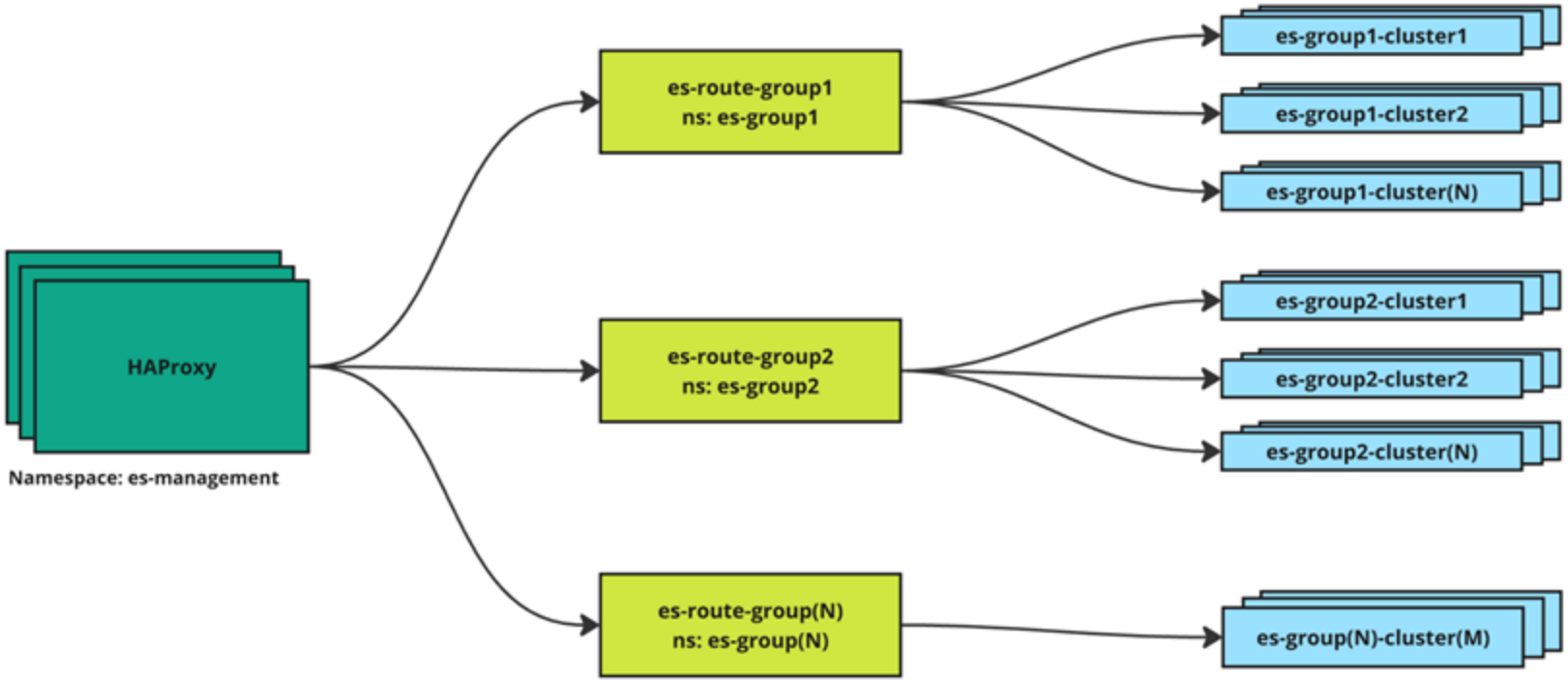
Data Cluster of Clusters (CoC)

- Routing to Cluster Groups (CoC)
- Namespace structure matching CoC routing
- Agnostic (standard) naming conventions
- Dedicated namespace for the entry point cluster (es-management)

```
groups_to_cluster_map:
  group1:
    default: true
    namespace: "es-group1"
    description: "meehhh"
    clusters:
    {{ range $i := until 17 }}
    - es-group1-c{{ i }}
    {{ end }}
  group2:
    namespace: "es-group2"
    description: "mooooh"
    paths: ["foo"]
    clusters: ["es-group2-c1"]
    best_effort: true
```

Cattle vs. Pets

An Elastic Cluster of Clusters Architecture



Outcomes

Outcomes

- **Map-reduce style** cluster architecture (hierarchical in the future?)
- **Standardized cluster sizes** and configurations – reducing engineering ops
- Consistent internal naming across all regions
- **Better code maintenance** and **surge of new automation tooling**
- **Enabled more experimentation** and performance analysis on a single cluster, allowing results to be applied across all others (traffic mirroring and load test in future?)
- **Convenience for upgrading** underlying cloud infrastructure

Outcomes

- **Reduced cluster build** and **migration time** from **14 weeks to ~2-5 days**
- Allowed us to:
  - Achieve up to **57% reduction** in compute costs through **pod packing**, with faster rollout and analysis across all clusters
  - Achieve up to 4x on query/indexing performance (latency)
  - Identify IO bottlenecks with projected storage savings of 50-82% by moving from single SSDs to HDDs with "Cattle RAID 0" (ongoing)

Meet the Team

Leonardo Santos

Sr. Distributed Systems Engineer

Mark Levins

Principal Distributed Systems Engineer

Valerio Aputini

Principal Distributed Systems Engineer

Mislav Bobesic

Sr. Distributed Systems Engineer

Stefano Cilloni

Sr. Distributed Systems Engineer

Szabolcs Szallar

Distributed Systems Engineer

Namrata Namrata

Sr. Distributed Systems Engineer

Nathan Ford

Distributed Systems Engineer

Thank you!

—

Q&A