elastic

Elastic SaaS로 Log 의 성능 극대화하기

보다 빠르고 경제적인 로그 관리 및 분석

Philip Choi Sr. Solutions Architect

elastic

Logs

```
64.242.88.11 - - [07/Mar/2023:16:10:02 -0800] "GET /mailman/hsdivision HTTP/1.1" 200 6291
64.242.88.10 - - [07/Mar/2023:16:11:58 -0800] "POST /twiki/bin/TWiki/Wiki HTTP/1.1" 404 7352
74.242.88.10 - - [07/Mar/2023:16:20:55 -0800] "GET /bin/view/DCCAndPostFix HTTP/1.1" 200 5253

2023-01-02T17:24:22Z angoro sshd[23510]: Failed password user joe from 10.0.0.53 port 2006 ssh2
2023-01-02T17:24:22Z angoro clamd[27173]: SelfCheck: Database status OK.
<129>JUN 07 12:54:14: alert  : 1/3/1025: alarm_mgr: 01:03:05:45 Minor ONU

64.242.88.10 - - [07/Mar/2023:16:10:02 -0800] "GET /mailman/hsdivision HTTP/1.1" 200 6291
                                              300] "POST /twiki/bin/TWiki/Wiki HTTP/1.1" 404 7352
                                              300] "GET /bin/view/DCCAndPostFix HTTP/1.1" 200 5253

                                              iled password user luca from 10.0.0.153 port 2006 ssh2
                                              elfCheck: Database status OK.
                                              300] "GET /mailman/hsdivision HTTP/1.1" 200 6291
                                              300] "POST /twiki/bin/TWiki/Wiki HTTP/1.1" 404 7352
                                              300] "GET /bin/view/DCCAndPostFix HTTP/1.1" 200 5253

                                              iled password user ugo from 10.0.0.200 port 2006 ssh2
                                              elfCheck: Database status OK.
                                              300] "GET /mailman/hsdivision HTTP/1.1" 200 6291
                                              300] "POST /twiki/bin/TWiki/Wiki HTTP/1.1" 404 7352
                                              300] "GET /bin/view/DCCAndPostFix HTTP/1.1" 200 5253

                                              iled password user phil from 10.0.0.153 port 2006 ssh2
                                              elfCheck: Database status OK.
```

> 로그는 광범위한 시스템과 응용 프로그램에서 생성되는 **event record**로, 일반적으로 **발생 시기**, **액세스**한 내용, 무엇이 또는 누가 **발생** 시켰는지에 대한 세부 정보와 관련 메타데이터를 포함하고 있습니다

로그는 이런 문제들에 유용하지만...

잠재적인
문제에 대한
조기 경고 제공

무엇이
잘못되었는지
정확히
파악하기

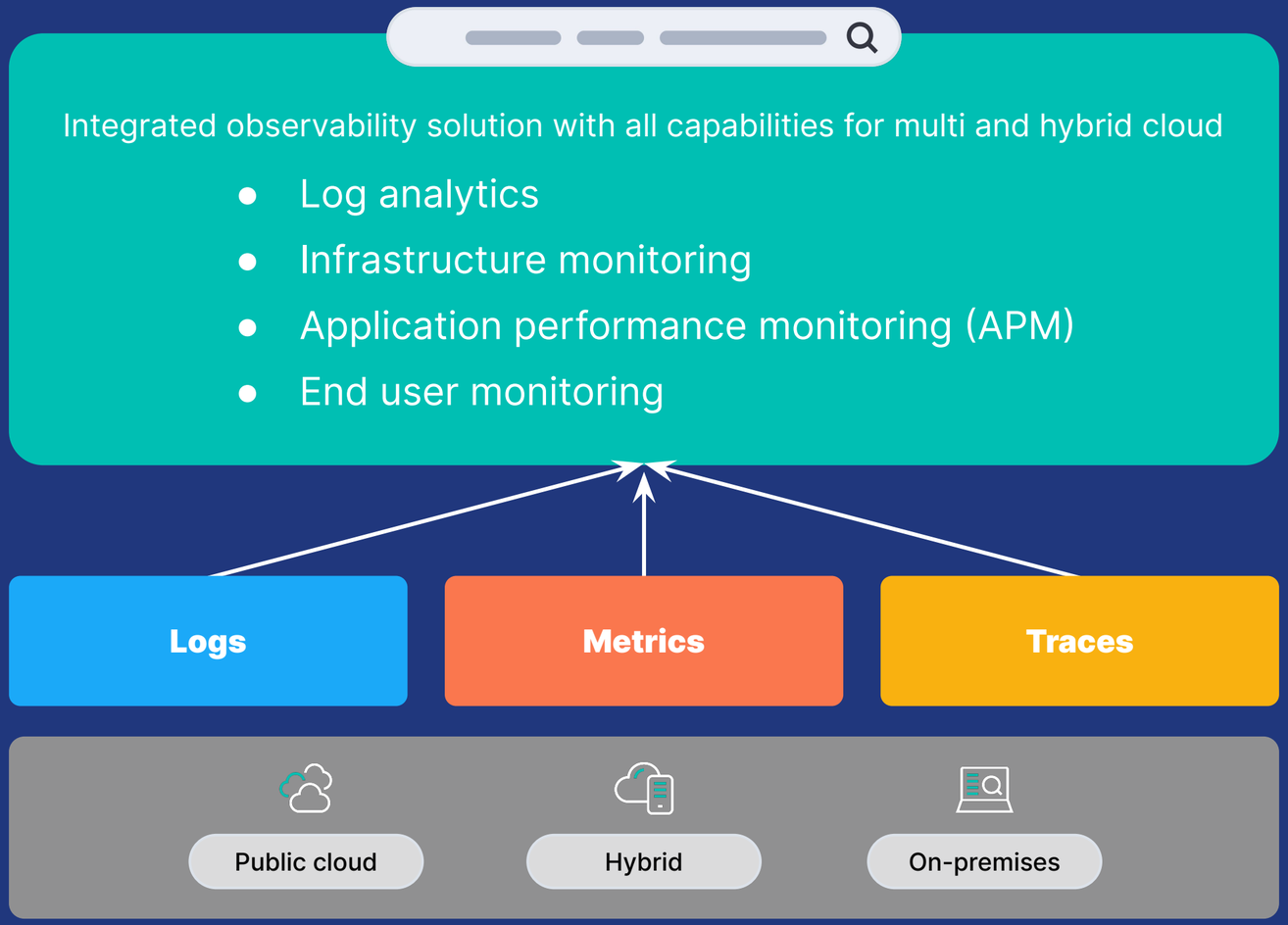
문제 해결
가속화

로그는 매우 급속하게 증가합니다

elastic

Elastic Observability

One platform, single datastore



elastic

5 key challenges of managing logs

Data 수집

천대 이상의 Agent 를 지속적으로 추적 관리하기 어려움

Log 검색 및 집계

종종 느리거나 복잡함...

모래사장에서 바늘찾기

수동으로 로그 찾기 또는 대시보드 보기
장애/응급상황시 가능한 선택이 아님!

비용 효율적인 데이터 보관

과거 데이터를 검색 가능하도록 해야 할까요? 혹은 압축해야 할까요? 혹은 둘다?

Data silos

Silo된 solution 으로 silo 문제를 해결시도 — 각 Silo에 접근하고 manually 연관지어 검색하는데 많은 시간 소요

elastic

5 Key Challenges

1. **Data 수집**
2. Log 검색 및 집계
3. 모래사장에서 바늘찾기
4. 비용효율적인 데이터 보관
5. Data silos

elastic

1 Data 수집

수집 Agent 를 지속적으로 관리하기 어려움

Problem:

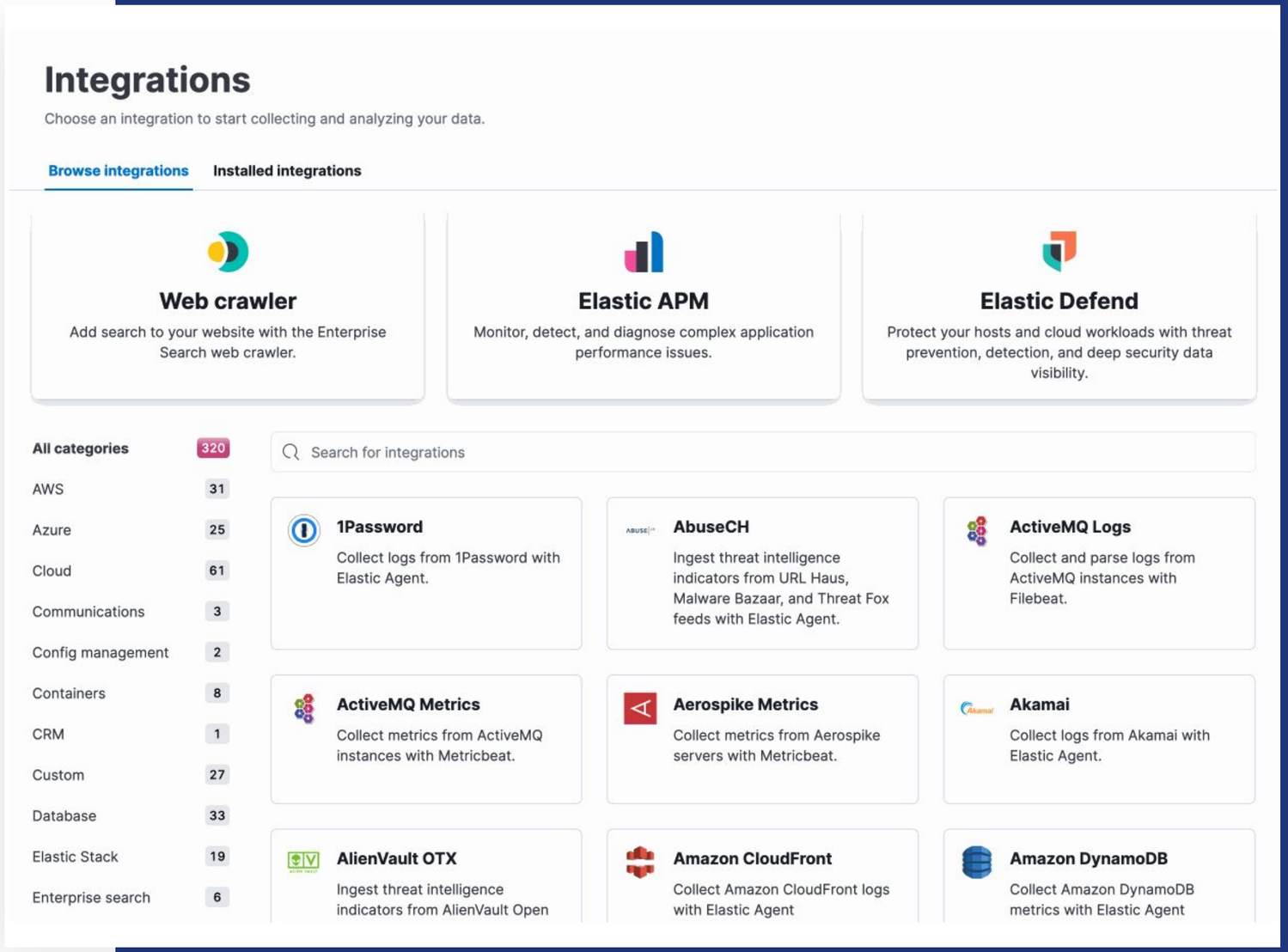
- Data source 로부터 log를 **수집**하는것은 **많은 시간이 소요**되고 때때로 세심한 계획이 선행되어야 함
- 수집 설정 Agent 의 설정을 수정하거나 **최신화** 하기 위해 Host 단에서 수동 작업이 필요
- 경우에 따라 추가적인 integration 설치는 **Vendor 에서 제공**받아야 하는 경우가 있고 추가적인 시간이 소요됨

Solution:

- **Elastic Agent** 는 **하나의 Agent** 로 전체 Infra 에 배포하여 **Log, Metric, Trace** data 를 모두 수집 가능
- **Elastic Fleet** 은 UI 를 통해 agent **중앙관리** 가능
- **추가적인 plugin 불필요, all self service**. 전체 data 에 대한 제어권을 고객이 직접 보유

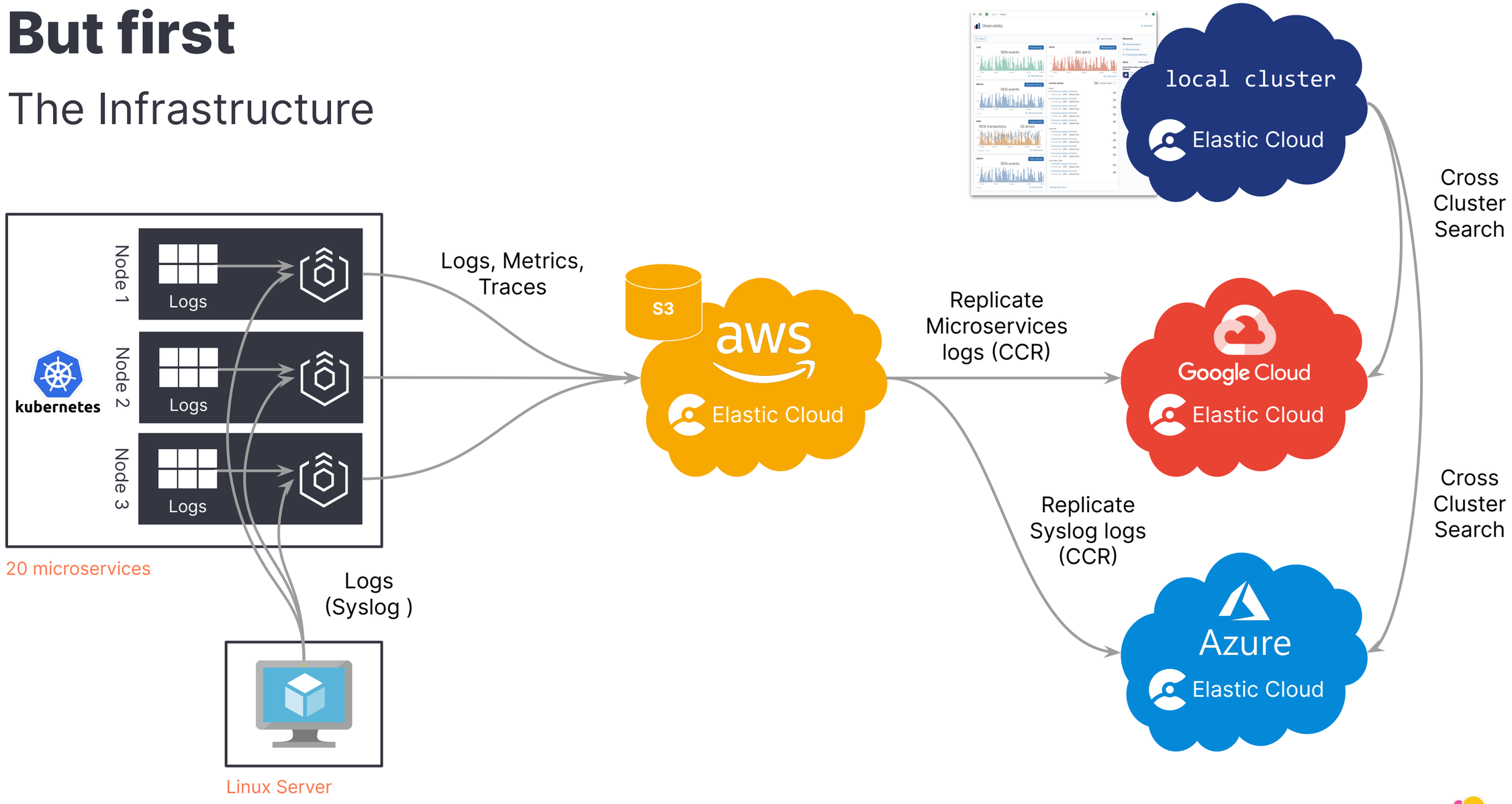
Elastic Agent

하나의 **Agent** 로 **한 곳**에서
수백개의 integration 을 **single click** 으로 관리할 수 있음



But first

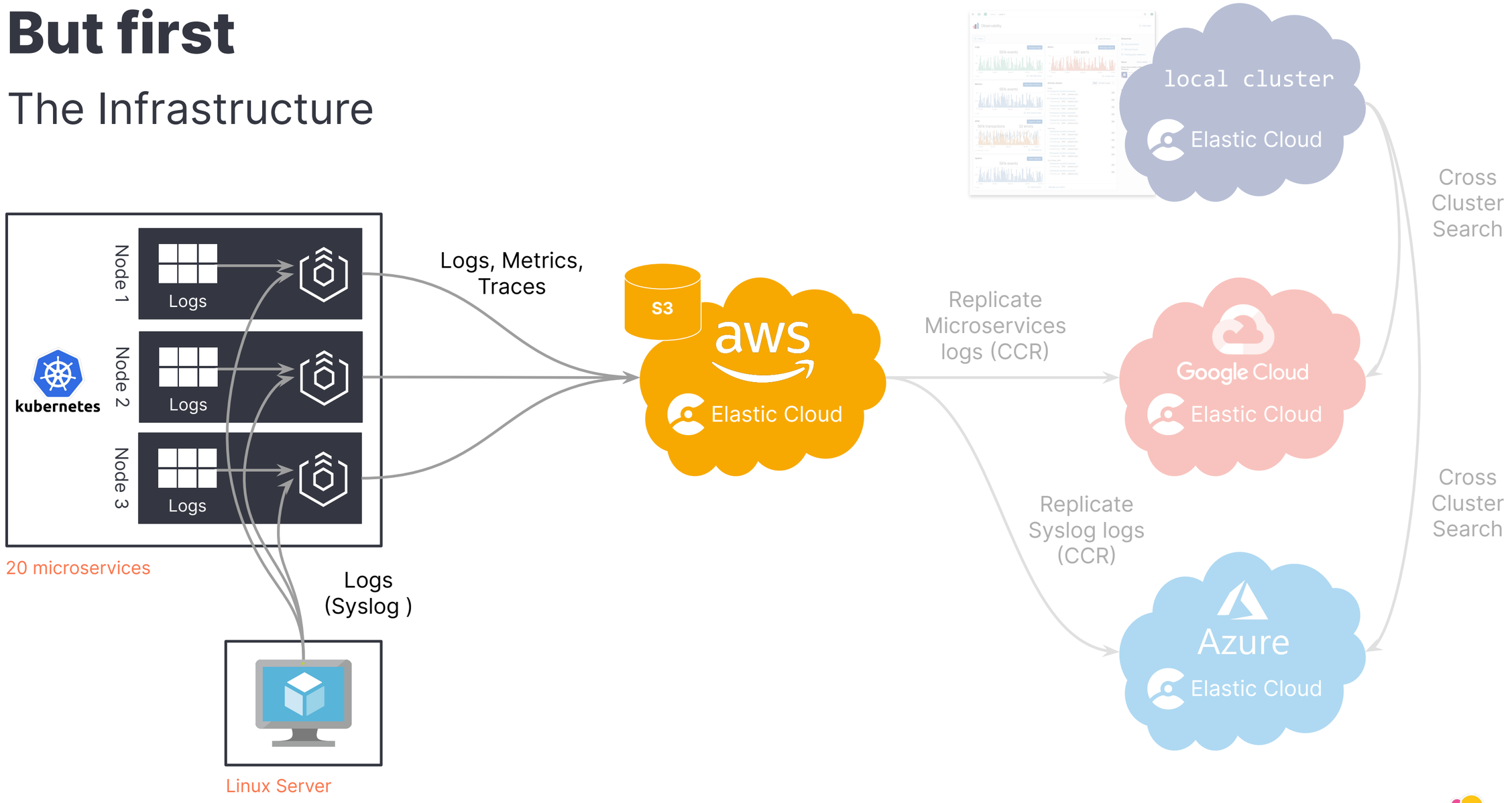
The Infrastructure



elastic

But first

The Infrastructure



elastic

How to onboard data

Kubernetes logs and Syslog

elastic

5 Key Challenges

1. Data 수집
2. **Log 검색 및 집계(aggregating)**
3. 모래사장에서 바늘찾기
4. 비용효율적인 데이터 보관
5. Data silos

elastic

2 Log 검색 및 집계(Aggregate)

Problem:

- **Log 는 검색가능**해야 하지만 모든 query 에 대해 parsing 하는것은 느리고 비용이 많이 소요됨
- event 를 summarize 하기 위해 많은경우 Log 분석에는 **집계 (Aggregate)**가 필요함
- Data 가 변경되거나 새로운 data source 가 추가되면 **unknown format 에 대한 관리**가 필요함

Solution:

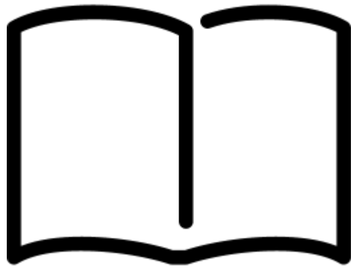
- **Elastic Platform 은 Elasticsearch 를 Core 로 사용합니다** - Full Text 데이터의 독보적인 검색 성능을 제공하는 Full Text 검색 엔진
- data 를 구조화하면 원본 데이터를 parsing 하지 않고도 data 에서 **millisecond 단위의 빠른 집계**를 할 수 있게 됩니다.
- Query 시점 processing 은 기존재하는 데이터에서 **실시간으로 필드를 추출**할 수 있게 해줍니다

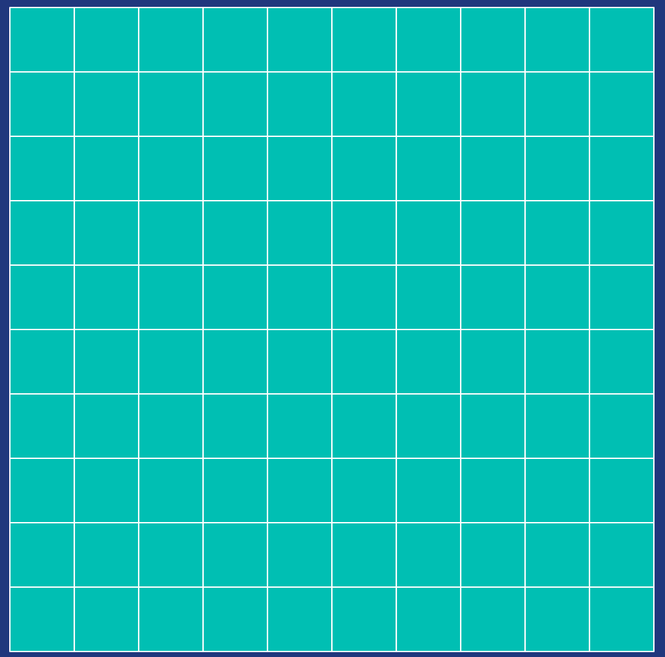
효율적인 데이터 검색

Database 에 대한 책 한권이 앞에 있다고 가정해 보겠습니다

1000 page 입니다

“**data type**” 에 대해 찾으려고 합니다





Individual Pages

효율적인 데이터 검색

Index 는 해당 page 를 즉시 찾을 수 있도록 해줍니다

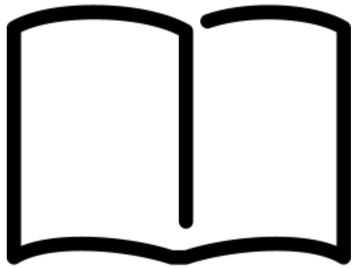
data types
  complex core field types, 93
  core, different indexing of, 80
databases
  in clusters, 11
  ineptness at extracting actionable data, 2
date field, sorting search results by, 112
date histograms, building, 437, 459
date math operations, 186
date ranges, 186
  using now function, no caching of, 193
date type, 88
dates field, sorting on earliest value, 113
date_detection setting, 147
decay functions, 305
decomposed forms (Unicode normalization), 346
deep paging, problems with, 76, 125
default mapping, 149
Default Unicode Collation Element Table (DUCET), 354, 355
default_index analyzer, 212
default_search parameter, 212
DELETE method
  deleting documents, 44
  deleting indices, 132
delete-by-query request, 558
deleted documents, 43, 158
  purging of, 166
denormalization
  and concurrency, 552
  denormalizing data at index time, 548
deployment, 631
  configuration changes, important, 635
  configuration management, 635
  file descriptors and MMap, 645
dictionary stemmers, 363
  dictionary quality and, 363
  Hunspell stemmer, 364
  size and performance, 364
disks, 632
distance
  calculating, 516
  sorting search results by, 520
distance_error_pct (geo-shapes), 537
distinct counts, 458
  optimizing for speed, 461
distributed databases, 1
distributed nature of Elasticsearch, 23
distributed search execution, 121
  fetch phase, 123
  query phase, 122
dis_max (disjunction max) query, 222, 223
  multi_match query wrapped in, 225
  using tie_breaker parameter, 224
doc values, 493
  enabling, 494
  storing geo-points as, 519
docs array
  in request, 54
  in response body, 54
Document Already Exists Exception, 44
document locking, 557
document oriented, 9
document store, Elasticsearch as, 36
documents, 562
  checking whether a document exists, 42
  creating, 43
  creating, indexing, and deleting, 63
  deleting, 44
  in Lucene, 137
  indexing, 10, 38

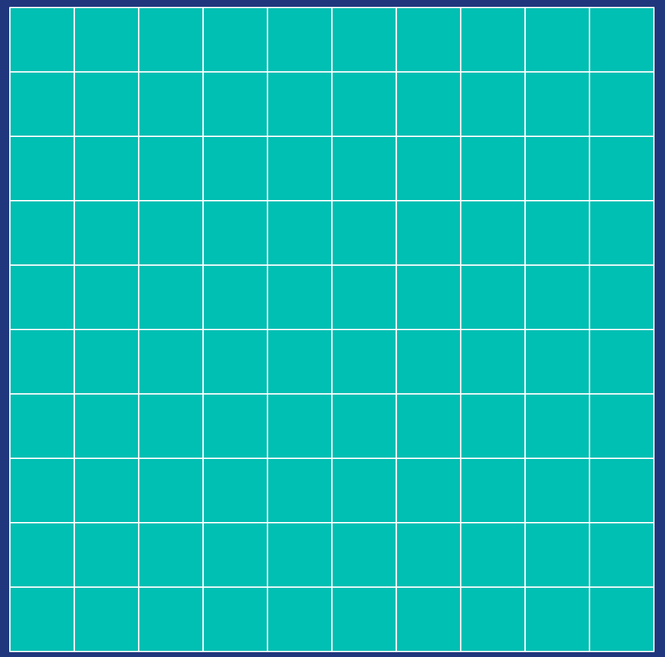
효율적인 데이터 검색

Database 에 대한 책 한권이 앞에 있다고 가정해 보겠습니다

1000 page 입니다

“**data type**” 에 대해 찾고 싶습니다





Individual Pages

Terms Index

효율적인 데이터 검색

전체 raw data를 보는 대신 Elasticsearch는
수집한 data에서 추가적인 정보를 추출하여 향후
검색을 훨씬 더 효율적이게 해줍니다

Data size for different data structures, approximation

전문(Full Text) Search

1,000,000 logs, **raw size 20GB**

해당 정보를 포함한 6 개의 log message

“Error: 123 for user John”

User searches for “**error**”.

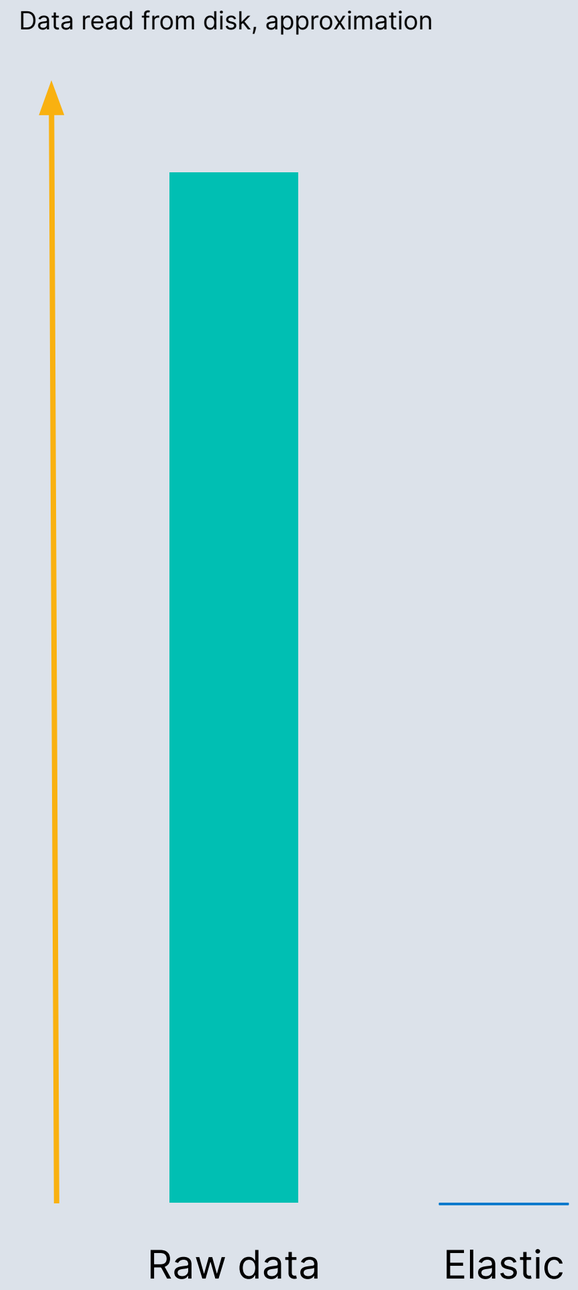
Elastic은 단어가 나타나는 문서 ID만 조회하면 되지만 Schema on Read 기반의 tool 들은 20GB의 data 전체를 parsing해야 합니다.

Error: 1, 63, 212, 7561, 12885, 756322

...

...

...



로그 검색 및 집계

전문(Full text) 검색과 몇분만에 custom dashboard 만들기

elastic

5 Key Challenges

1. Data 수집
2. 로그 검색 및 집계
3. **모래사장에서 바늘찾기**
4. 비용효율적인 데이터 보관
5. Data silos

elastic

3 Finding the needle in the haystack

Problem:

- 모든 데이터를 수동으로 살펴보는 것은 현실적으로 적용 가능한 방법이 아닙니다. **Machine Learning 이 필요**합니다.
- 단순히 데이터를 저장하는 것만으로는 부족합니다. 데이터에서 **가치를 도출**해야 합니다
- 조사는 **효율적**이고 **신속**하게 수행되어야 합니다
- Data 는 자동적으로 **집계**되어야 합니다

Solution:

- **Elastic Machine Learning -** 즉시 적용 가능한 job 을 제공합니다
- 로그에서 파생된 **Business Metric**에 대한 경고를 만들 수 있습니다(예시: “결제가 정상 완료되었습니다” 절반 감소)
- 애플리케이션에서 발생한 특정 event 에 일정 형식의 로그가 연관있음을 알 수 있는 **Log Correlation** 을 제공합니다
- **AIOPs -** 수십억건의 Log 에서도 Log Spike 를 감지하고 Category 를 자동으로 분류하여 줍니다

Elastic Observability

통합적인 AIOps with Machine Learning

10년 이상의 개발 경험, 업계 선두 기술

- 시계열 기반, 이상탐지
- Categorizations
- Correlations
- 유연하고 쉽게 cutomzing 가능한 ML modeling

모래사장에서 바늘찾기

AIOPS 로 Explain Log Spike 및 연관관계 찾기

elastic

5 Key Challenges

1. Data 수집
2. 로그 검색 및 집계
3. 모래사장에서 바늘찾기
4. **비용 효율적인 데이터 보관**
5. Data silos

elastic

비용 효율적인 데이터 보관

Problem:

- Compliance 또는 Business 적인 이유로 **수년간의 Data 를 보관**해야 함
- **모든곳에 빠른 SSD 와 고성능 CPU 를 탑재하는 것은 경제적으로 적용할 수 없음**
- 상대적으로 **저렴한 Storage** 가 요구되지만 통상적으로 Local SSD 대비 **현저히 느림**
- 다른 솔루션들은 데이터를 **복원**하거나 사용 가능하게 하기 위해 다른 수동적인 단계들이 필요한 경우가 있어서, **중요한 조사과정이 느리게** 되거나 **비용을 상승**시키는요소가 됩니다.

Solution:

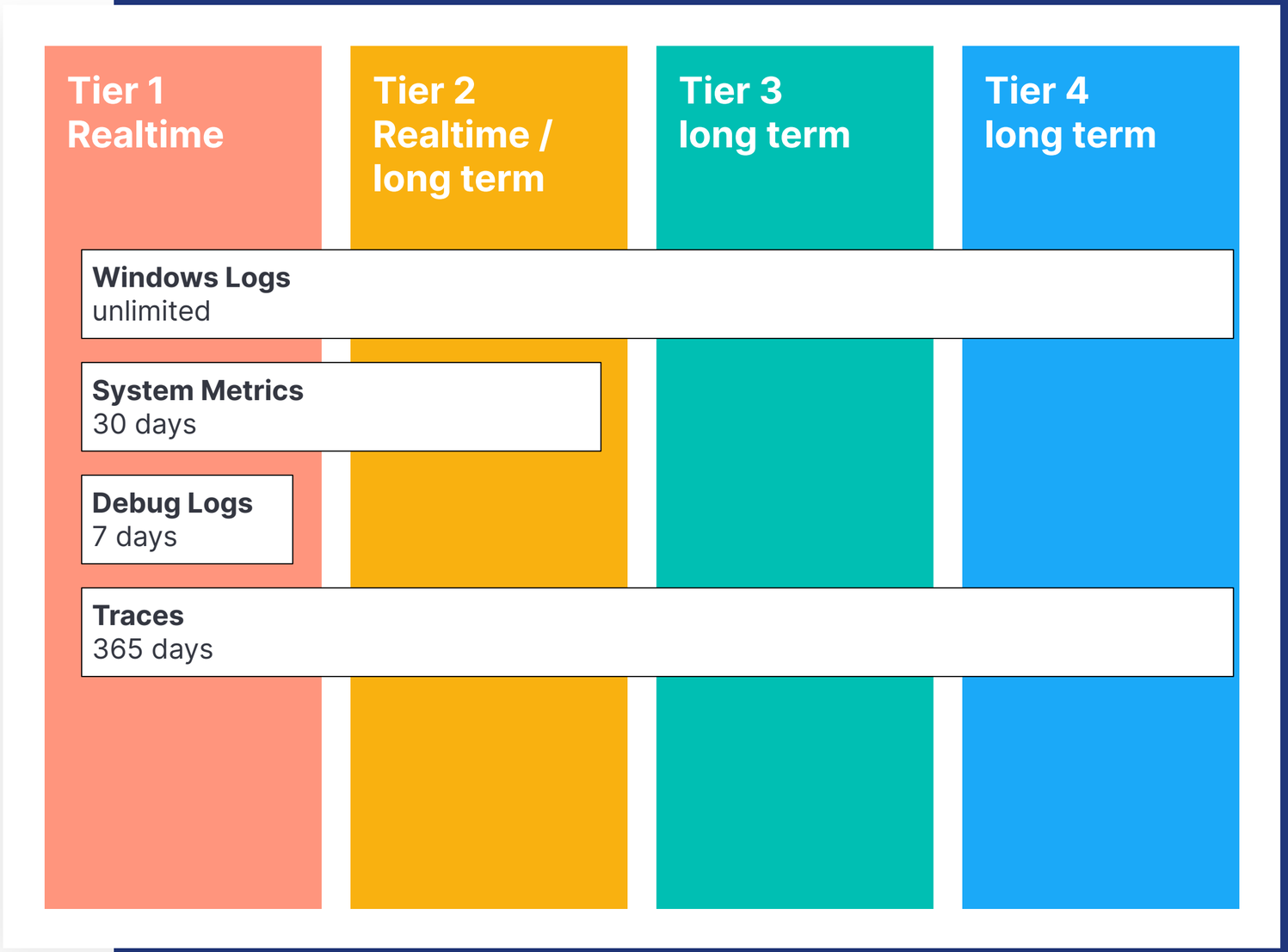
- **Elastic Data Tiering** 은 Hardware 의 **최대한의 가치**를 이끌어낼 수 있습니다
- **저비용 장기보관**을 위해 S3 등 비용효율적인 blob storage 를 지원합니다.
- 별도의 복원과정 없이 전체 데이터에 대한 접근을 동일 Kibana 안에서 지원하여 **Seamless User Experience** 를 제공할 수 있습니다
- **사전 정의된 Data 구조 기반으로 좋은 performance 를 제공합니다 - 데이터의 일부분만 load**

비용효율적인 데이터 보관
with data tiers

비용 효율적인 데이터 보관

Data 종류별로 사용자 정의 가능

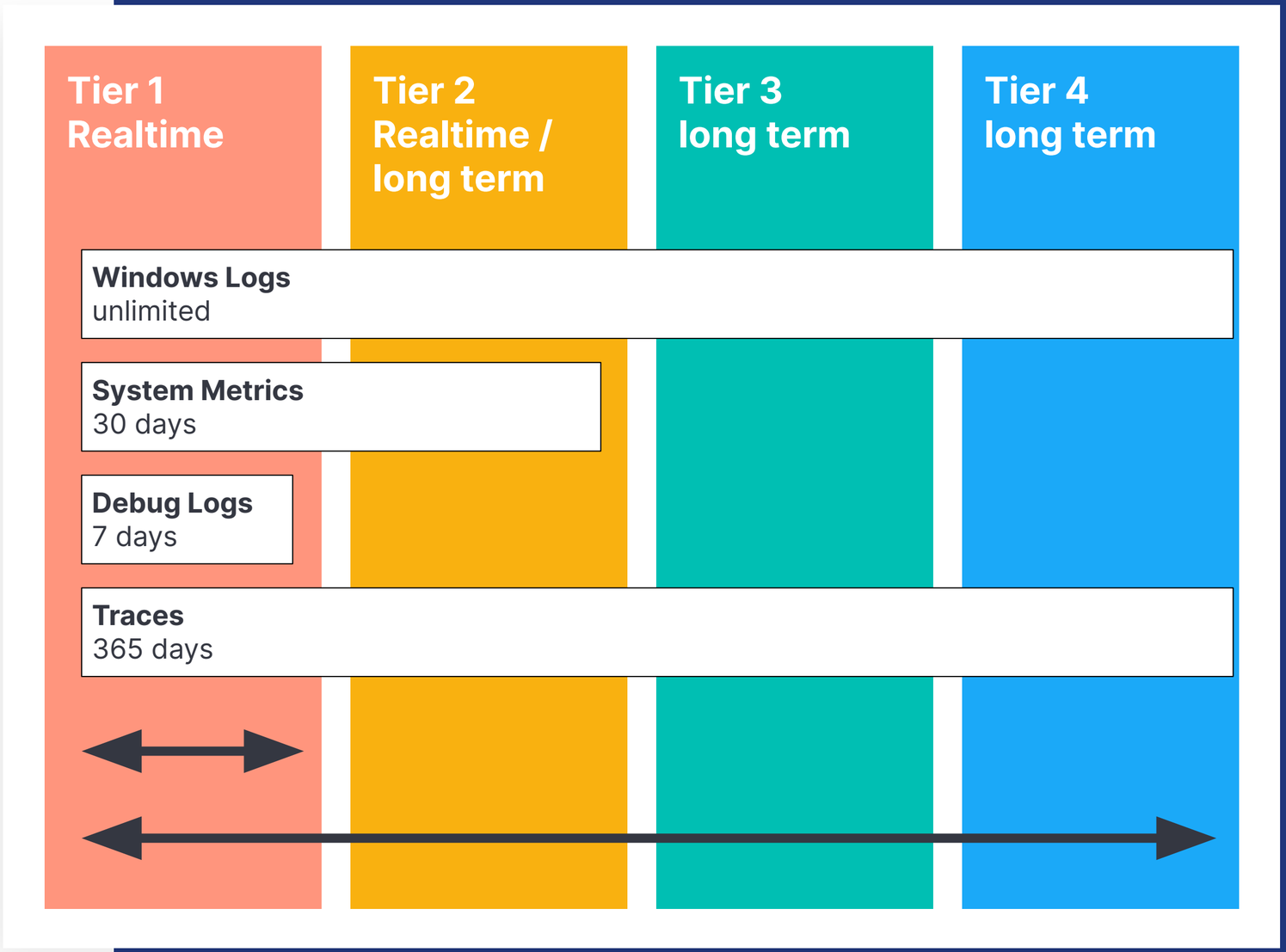
- 데이터를 계층 이동시 소스별로 사용자 정의 가능
- 보관주기 무제한



끊김없는 검색

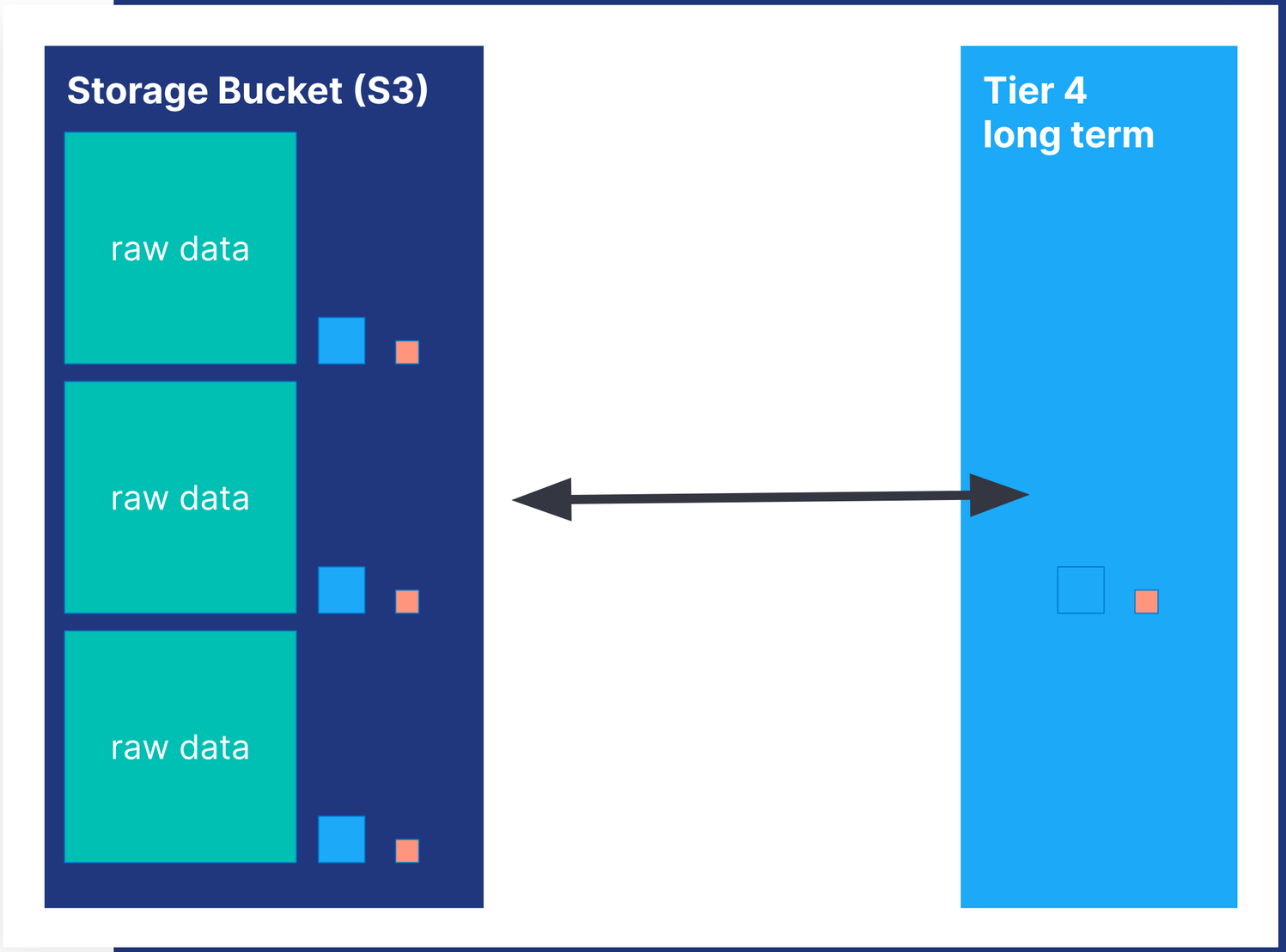
별도 수동작업 불필요

- 전체 데이터에 대한 동일 UX
- 복원 불필요



효율적인 과거 데이터 검색

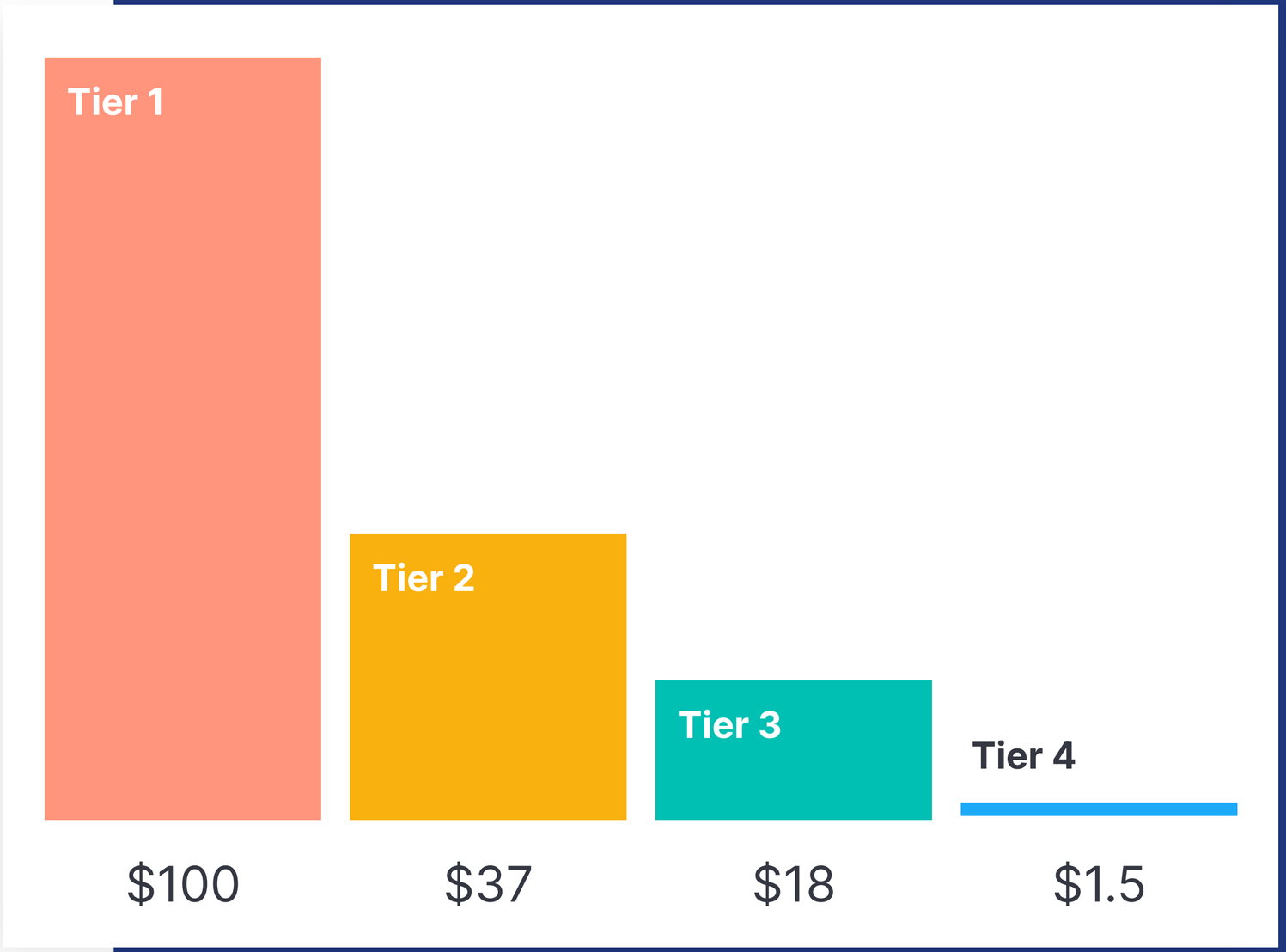
- 필요 data 만 loading
- Cached locally
- 복원작업 혹은 수동 개입 불필요
- 다른 접근 방식에 비해 빠른 쿼리 성능
- Hardware 비용 감소
- object storage API 비용 감소
- data 전송 비용 감소



비용효율적인 데이터 보관

계층별 데이터 저장 비용

- 지속적인 비용 감소
- 저렴한 장기 storage
- 계층간 유연한 data 이동



*approximation, data from https://cloud.elastic.co/pricing

비용효율적인 데이터 보관

5년치 데이터에 대한 seamless search

elastic

Demo: Cost Effective Data Retention

6

Tier 4 nodes

5

Years of data

1PB

Searchable data

1.7

Trillion Documents

elastic

5 Key Challenges

1. Data 수집
2. 로그 검색 및 집계
3. 모래사장에서 바늘찾기
4. 비용효율적인 데이터 보관
5. **Data silos**

elastic

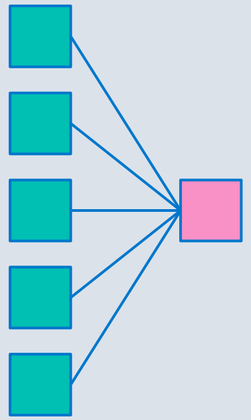
5 Data ownership 을 유지하면서 Silo 해체하기

Problem:

- **Observability Data**는 많은 경우 **Silo에 저장**되며, 이는 단일 시스템에서 모든 것을 사용할 수 없음을 의미합니다
- **Log** 는 **Metric** 이나 **Trace** 와 서로 **다른곳에 저장되어 있습니다**
- 동일데이터 유형이라도 **지역적으로 다른곳**에 있어 상호 접근이 안되거나, **data correlation이 복잡하고 느립니다**
- **장애 발생시 data 접근이 불가합니다**

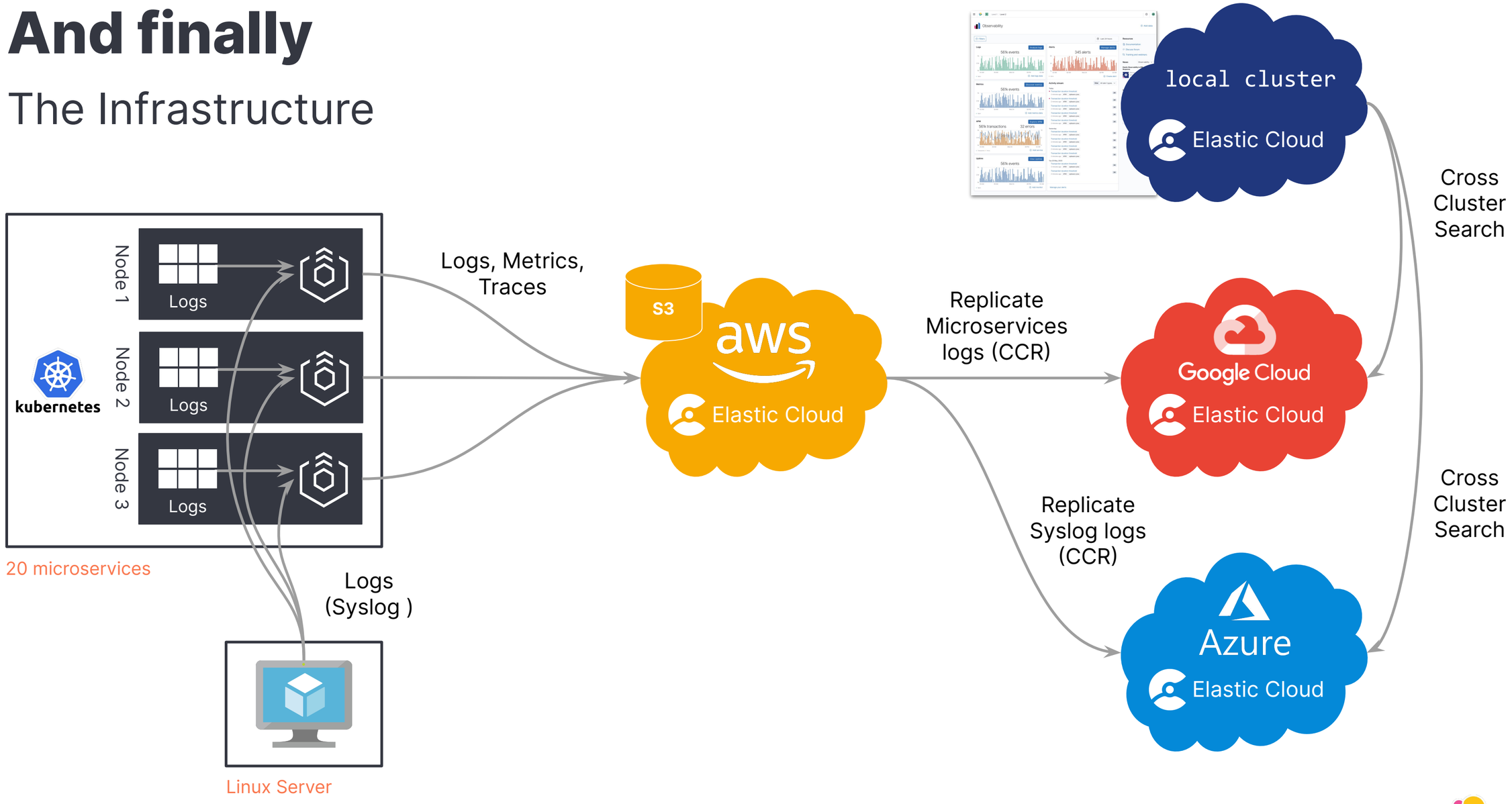
Solution:

- **Elastic Platform -** Cross Cluster Search and Cross Cluster Replication
- **Elastic Observability,** 전체 data 에 대한 단일 platform
- **Cross Cluster Search** 다른지역, 여러 CSP 간에도 user 에게 동일한 UX 를 제공합니다
- **Cross Cluster Replication** 여러 region 에 걸쳐 중요 data 에 대해 HA 를 구성할 수 있습니다



And finally

The Infrastructure



elastic

Data Silos

Cross-cluster Replication and Cross-cluster Search를 사용하면 data ownership 을 유지하면서 Silo 문제를 해결할 수 있습니다

elastic

5 key ~~challenges~~ takeaways when managing logs

Data 수집

Elastic Agent는 다양한 유형의 로그, 지표 및 추적을 수집하는 확장 가능한 솔루션

Log 검색 및 집계

빠른 결과, 쉬운 집계, 유연한 런타임 필드

모래사장에서 바늘찾기

AIOPs 기능, Root Cause 분석 및 business metric 에 대한 Anomaly Detection

비용효율적인 데이터 보관

Data tiering, seamless UX, 복원 불필요, 빠른 검색

Data silos

Cross Cluster Search 와 Cross Cluster Replication 로 data ownership 을 유지하면서 log 배포

elastic

감사합니다

elastic