

logstore

Saving million+ dollars with Clickhouse -
Zomato's logging platform journey

Anmol Virmani
Palash Goel

zomato

Zomato at a glance

A technology platform that connects customers, restaurant partners and delivery partners serving their multiple needs.



All content provided here is for informational and educational purposes only.

*The business numbers used here correspond to Zomato's latest shareholder's report Q1FY24.

zomato

Why do we need a centralized logging platform?

System Requirements

- Ingestion rate upto: **160 M/min**
- Size: **60 TB/day**
- Queries: **30 k/day**
- Ingestion Lag: **5s**
- P99 query runtime: **Less than 10s**
- Scalable

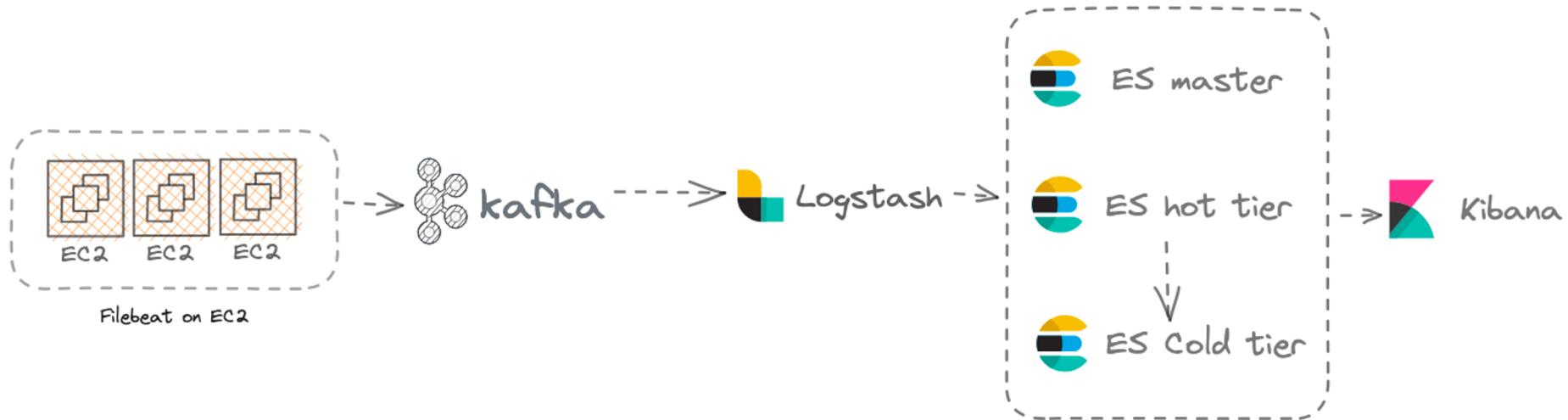


ELK



zomato

Scaling ELK



zomato

Challenges

- Cost ballooned to more than a **million** dollars/year
- Operational complexity
- Requirement for additional paid features



zomato

Alternatives

splunk®>



new relic®



Grafana loki



trino



ClickHouse

zomato

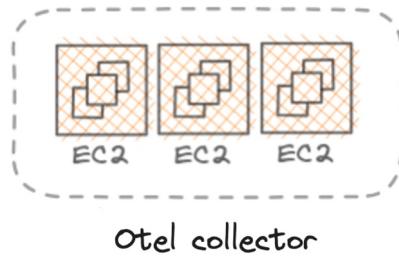
Why ClickHouse

- Continuous background merges
- Horizontal and vertical scaling
- Shared-nothing architecture
- Full-text search capabilities
- Excellent compression
- Mature SQL syntax
- Extensive community support

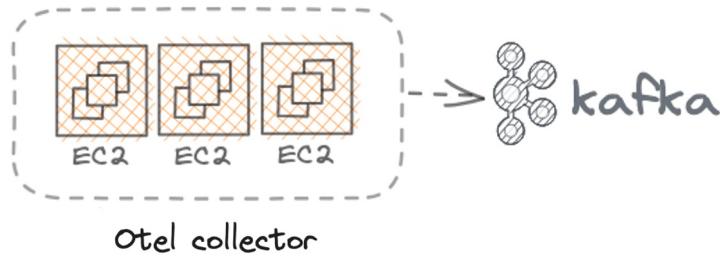


zomato

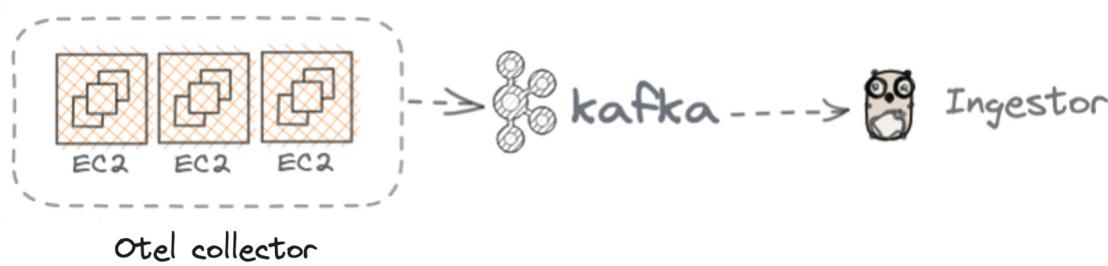
Logstore Architecture



Logstore Architecture



Logstore Architecture



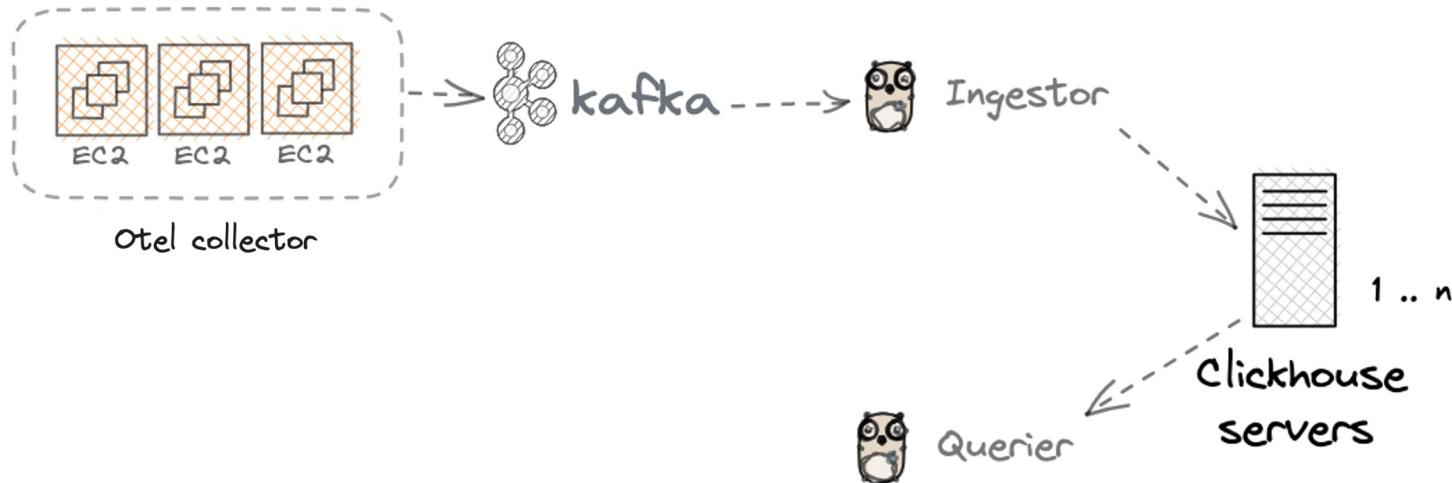
zomato

Logstore Architecture



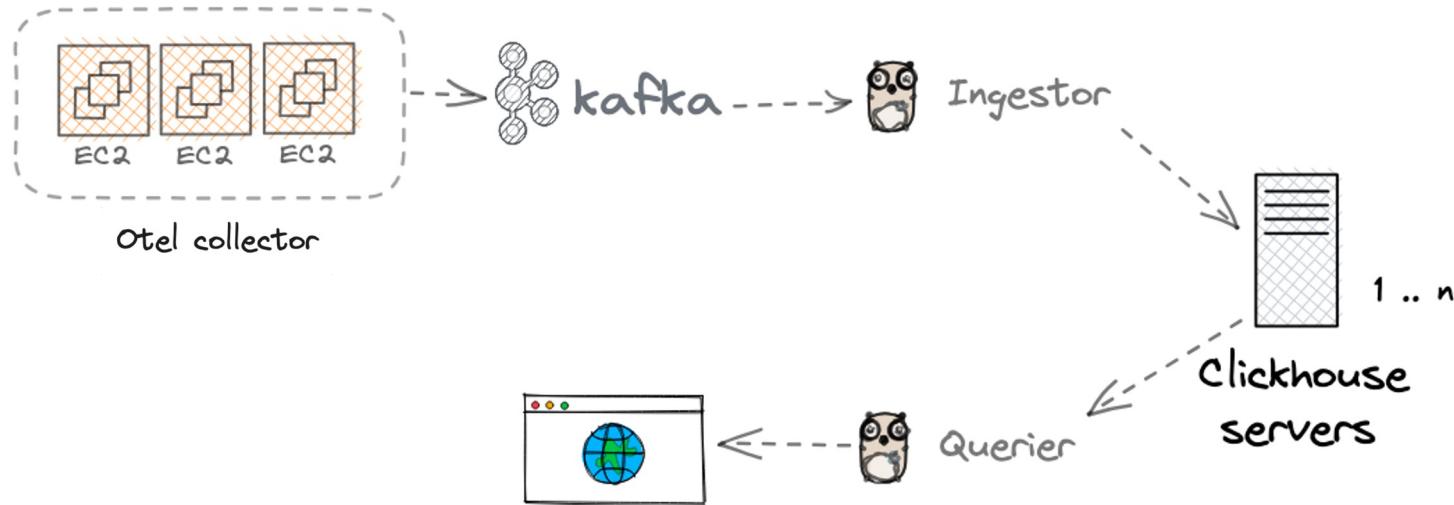
zomato

Logstore Architecture



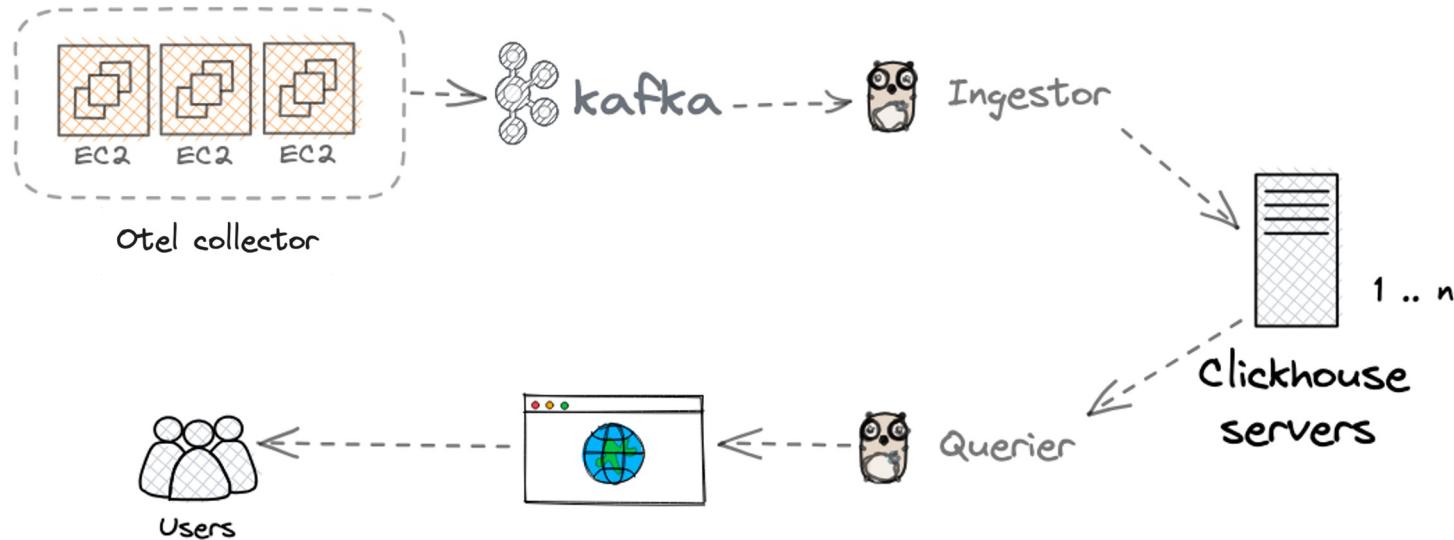
zomato

Logstore Architecture



zomato

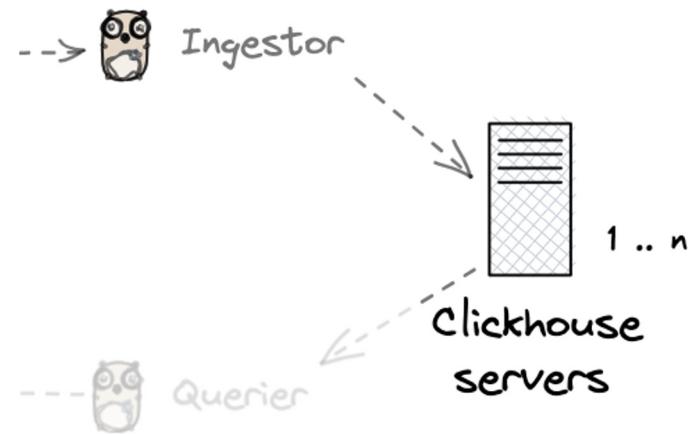
Logstore Architecture



zomato

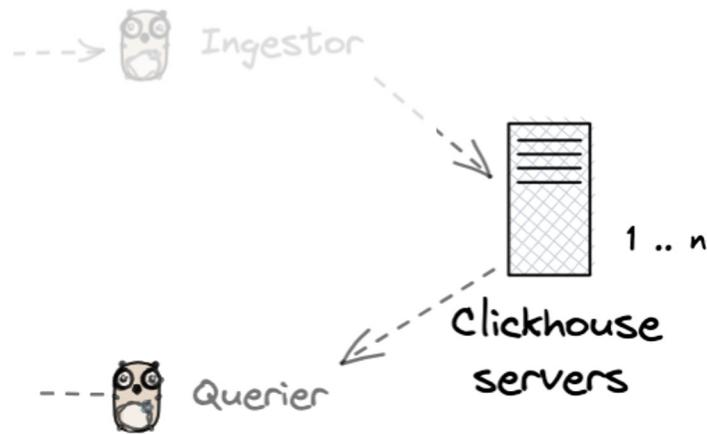
Optimizations/Features

- Segregating Ingestion
- Custom DSL
- Sorting and Bloom Filters
- Schema-agnostic
- Codecs and compression
- Data-tiering



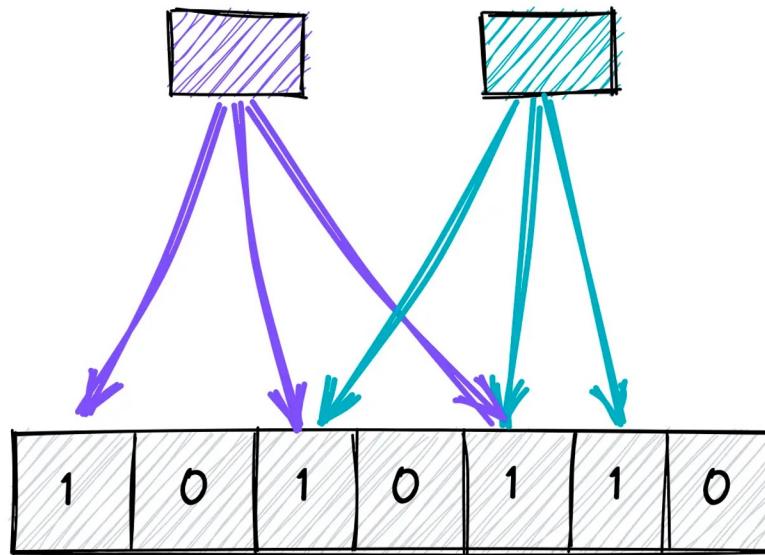
Optimizations/Features

- Segregating Ingestion
- Custom DSL
- Sorting and Bloom Filters
- Schema-agnostic
- Codecs and compression
- Data-tiering



Optimizations/Features

- Segregating Ingestion
- Custom DSL
- Sorting and Bloom Filters
- Schema-agnostic
- Codecs and compression
- Data-tiering



Optimizations/Features

- Segregating Ingestion
- Custom DSL
- Sorting and Bloom Filters
- Schema-agnostic
- Codecs and compression
- Data-tiering

```
create table if not exists foo_service (
    ts          Datetime,
    env         Lowcardinality(String),
    container_id Lowcardinality(String),
    trace_id    FixedString,
    msg         String,
    offset      UInt64 codec(DoubleDelta,zstd)
    _others     Map<LowCardinality(String), String>
    INDEX foo_service_msg_index
    TYPE tokenbf_v1(212062,3,0)
    GRANULARITY 1
)
Engine = MergeTree
Partition by toYYYYMMDD(ts)
order by (env, trace_id, ts)
```

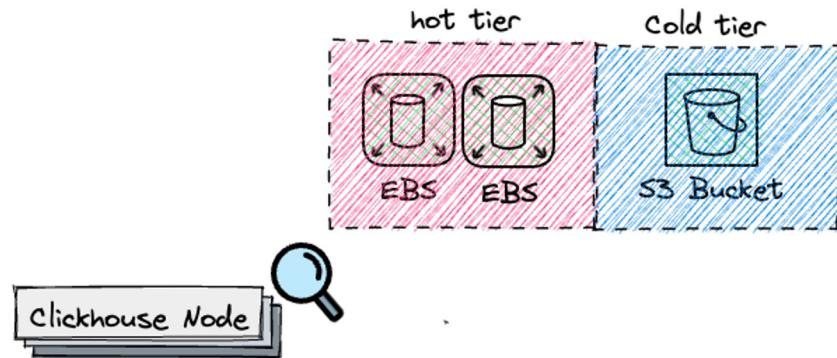
Optimizations/Features

- Segregating Ingestion
- Custom DSL
- Sorting and Bloom Filters
- Schema-agnostic
- Codecs and compression
- Data-tiering

```
create table if not exists foo_service (
    ts          Datetime,
    env         Lowcardinality(String),
    container_id Lowcardinality(String),
    trace_id    FixedString,
    msg         String,
    offset      UInt64 codec(DoubleDelta,zstd)
    _others     Map<LowCardinality(String), String>
    INDEX foo_service_msg_index
    TYPE tokenbf_v1(212062,3,0)
    GRANULARITY 1
)
Engine = MergeTree
Partition by toYYYYMMDD(ts)
order by (env, trace_id, ts)
```

Optimizations/Features

- Segregating Ingestion
- Custom DSL
- Sorting and Bloom Filters
- Schema-agnostic
- Codecs and compression
- Data-tiering



production

dummy_service_3

now-3h to now

Available Fields

Search fields...

Add filter

100,001,212 hits

Timestamp per hour Hours Jul 11, 11:11:55 to Jul 11, 14:11:55

Time →

Download

Timestamp channel msg

- ▶ Jul 11 2023, 13:11:01 billing message received for order in NEW state
- ▶ Jul 11 2023, 13:11:02 PRINTING order print request in for orderId 2020393209
- ▼ Jul 11 2023, 13:11:03 billing enqueued in worker for relay

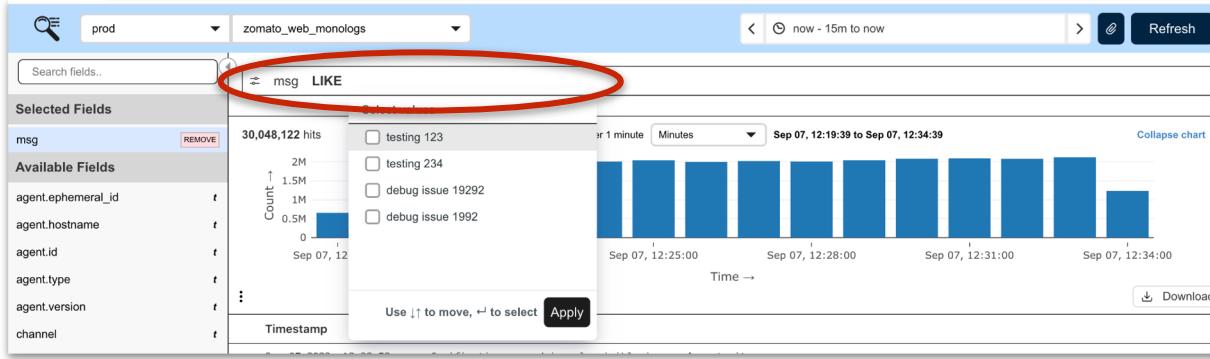
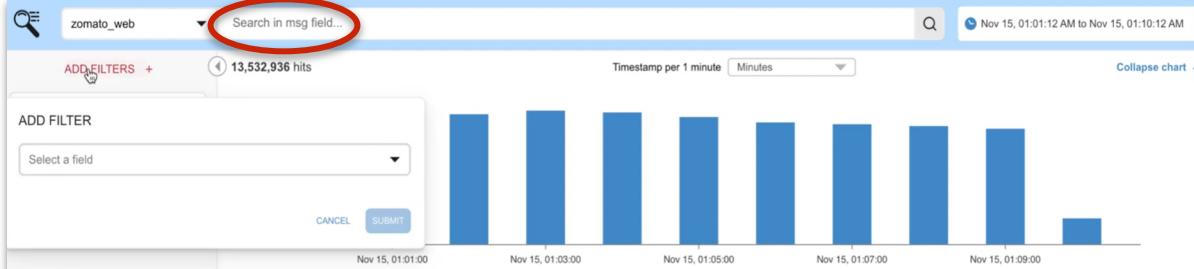
TABLE JSON

agent	-
agent.ephemeral_id	dummy-id-1
agent.hostname	9d2c49731dad
agent.id	wdjwhdu-28282-2dbkwjdbk
agent.type	filebeat
agent.version	7.1.1
channel	billing
container.id	jeibh3hbivfph34bivbn3rvicun3rvoosnzs
container.image.name	dummy-container-image-1
container.labels.env	prod
container.labels.environment	prod
container.labels.log_type	application

zomato

Frontend application

- Changed developer's behaviour to filter logs based on values in dedicated fields.



zomato

Frontend application

- Encouraged structured logging pattern

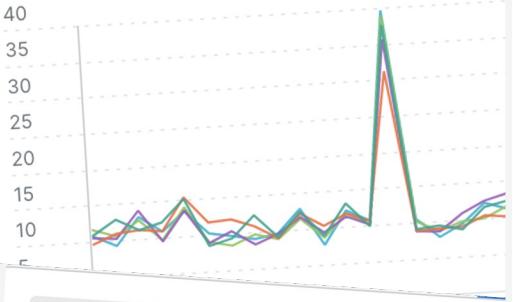
```
{  
  "ts": "2023-10-17T10:39:46-00:00",  
  "severity": "info",  
  "msg": "job failed",  
  "attributes": {  
    "job_id": "123",  
    "reason": "out of memory"  
  },  
  "resource": {  
    "service_name": "foo",  
    "environment": "prod",  
    "container_name": "statsd-sidecar"  
  }  
}
```



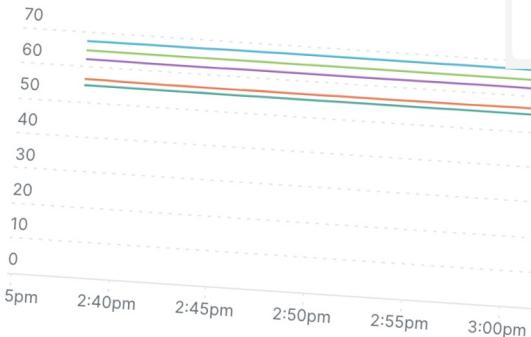
zomato

Things we monitor

CPU (%) ▾



Disk used % ▾



Transmit bytes ▾

160 M
140 M
120 M
100 M
80 M
60 M
40 M
20 M

Receive bytes ▾

400 M
350 M
300 M
250 M
200 M
150 M
100 M
50 M
0

Read bytes ▾

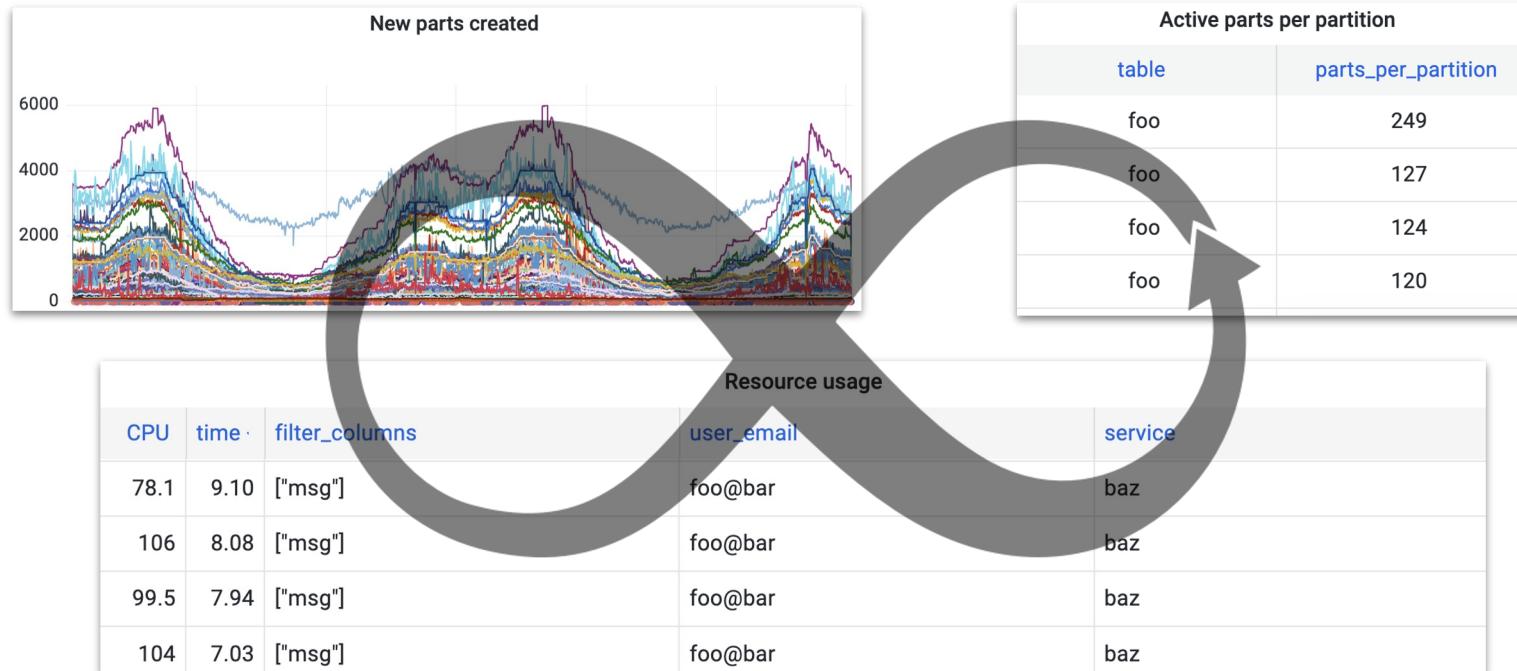
120 M
100 M
80 M
60 M
40 M
20 M
0

Write bytes ▾

160 M
140 M
120 M
100 M
80 M
60 M
40 M
20 M
0

zomato

Things we monitor



zomato

Impact?



Auditing capabilities



Stats and monitoring



Ingestion time < 5s



P99 Query time < 10s



zomato

Next?

- Distributed Tracing
- Compressed log processor (CLP)
- Query suggestions
- Autoscale clickhouse
- Remove Kafka
- Open-source Logstore



zomato

Embracing Clickhouse

- Metrics Platform (Beta)
- Iceberg with clickhouse
- chDB



zomato

Questions?

zomato

Keen to uncover our Tech Prowess?



<https://blog.zomato.com/category/technology>

zomato

Thank you

Email your resumes to techrecruitment@zomato.com
and let's innovate towards "better food for more people"



zomato