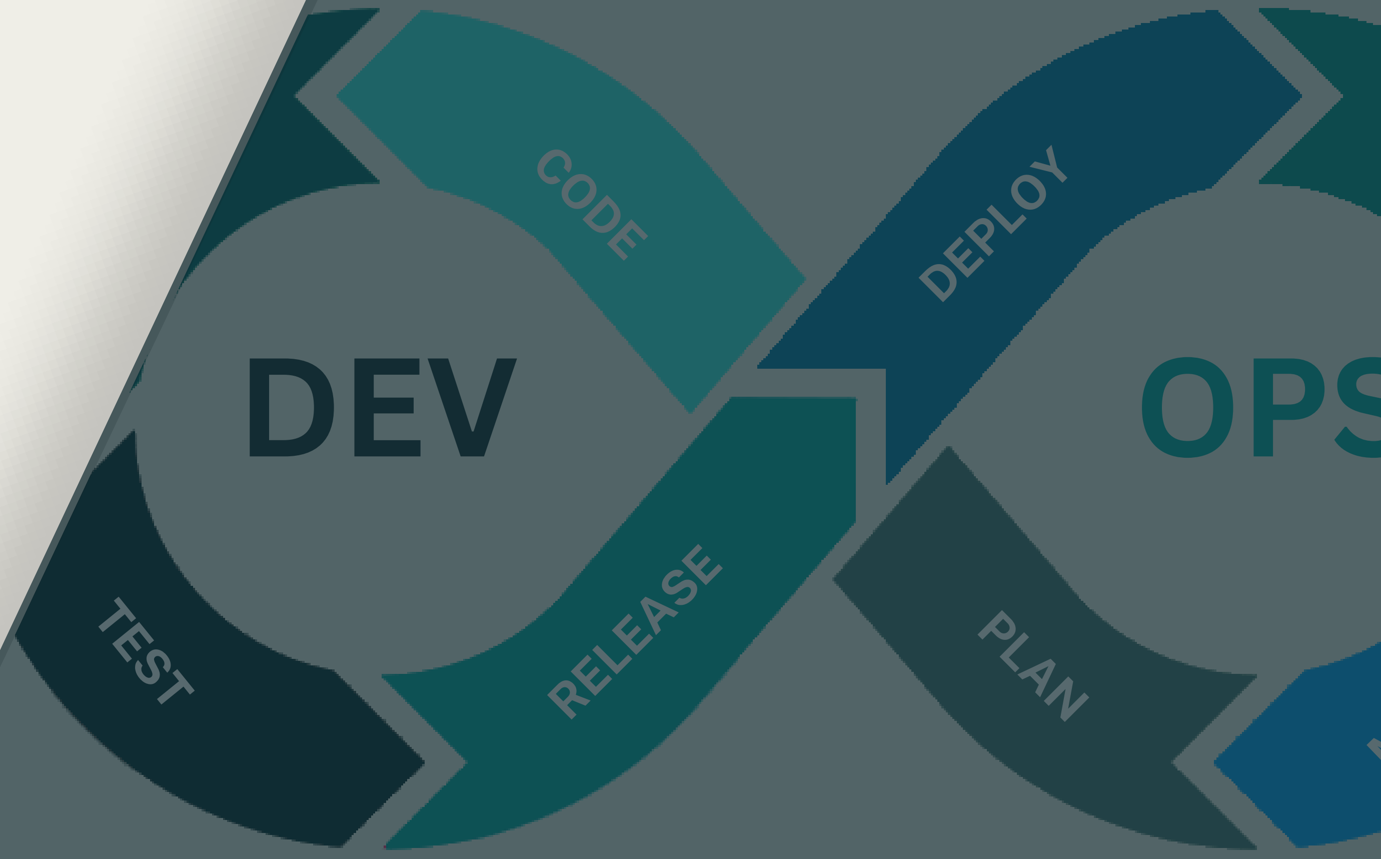
OPSVERSE

Unlocking NextGen Log Analytics With ClickHouse and Kafka

ARUL JEGADISH
COFOUNDER AND CEO, OPSVERSE

OSA CON 23

Do you use logs?



LOGS IN OBSERVABILITY

- Simplest form of telemetry
- Almost everyone uses them

OPSVERSE

CHALLENGES WITH LOGS

- Verbose
- Unstructured
- Hard to search
- Harder to run analytics

OSA CON 23

EXISTING SOLUTIONS

- **ElasticSearch/OpenSearch**
  - Good for analytics
  - But, complex and costly
- **Loki**
  - Not suitable for analytics
  - Cost effective

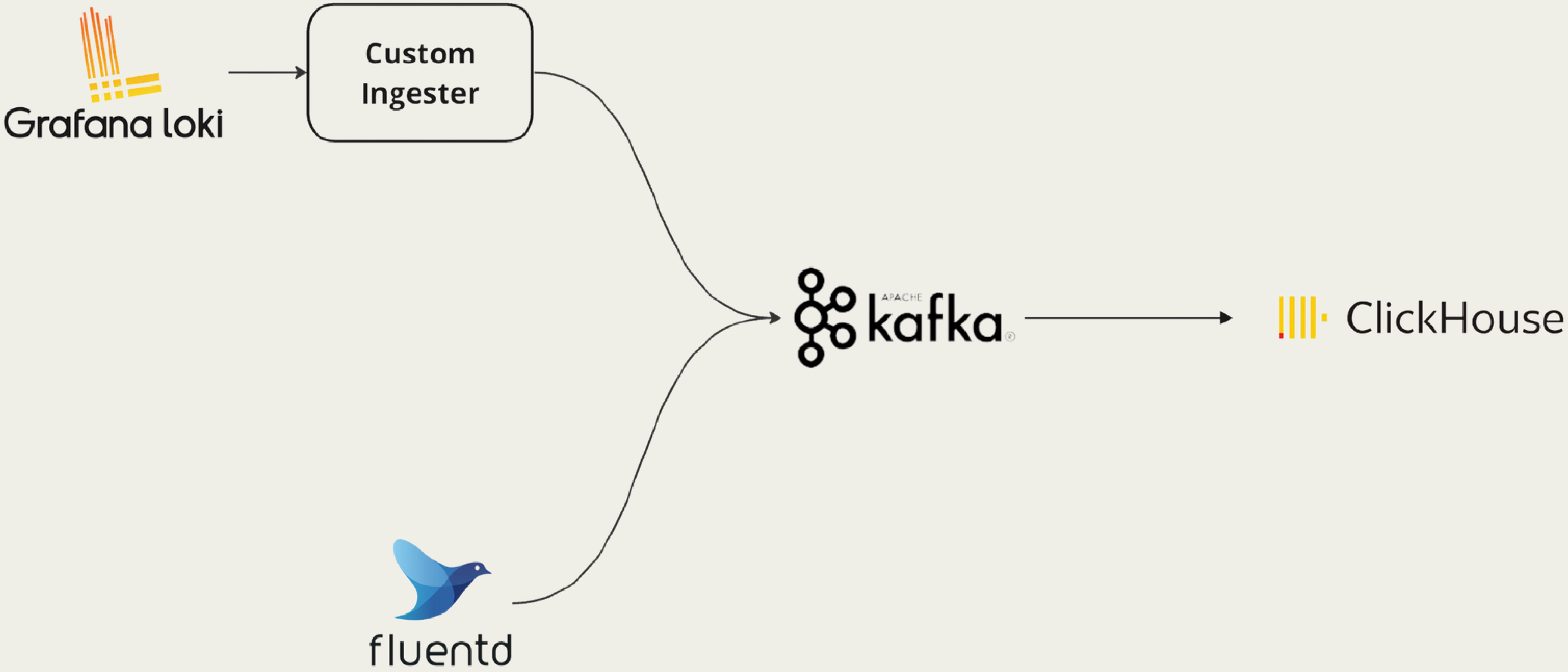
WHAT DO WE NEED?

Cost effective yet scalable analytics on logs

SOLUTION

APACHE kafka + ClickHouse

HIGH LEVEL DESIGN

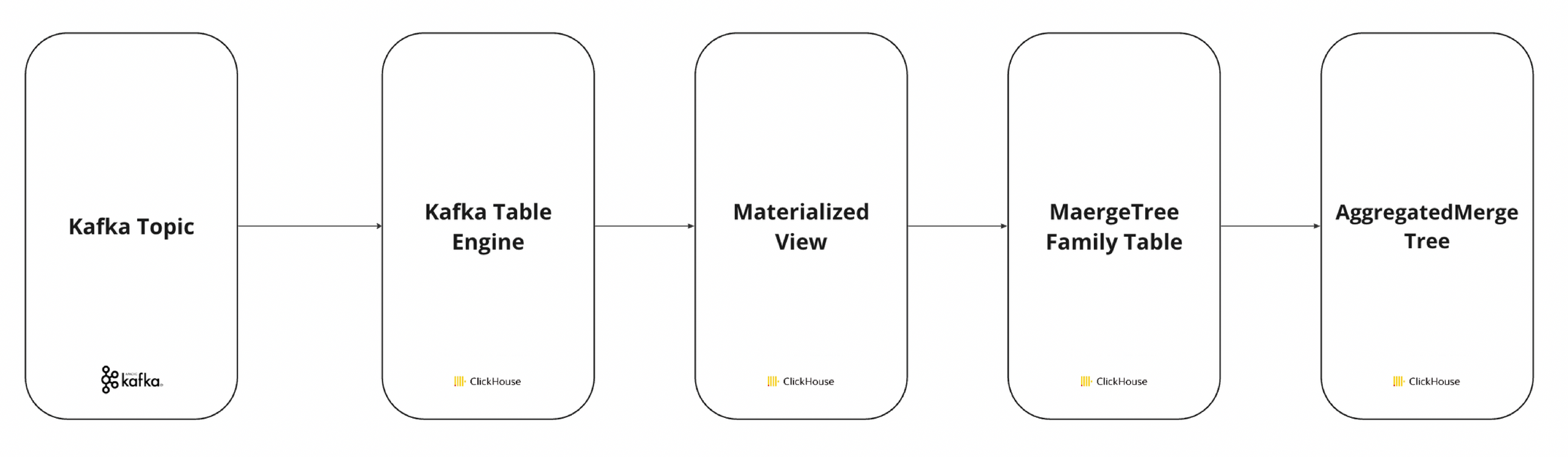


CLICKHOUSE FEATURES

- JSON fields
- Kafka table engine
- Materalized View
- AggregatedMergeTree engine

OPSVERSE

DATA FLOW

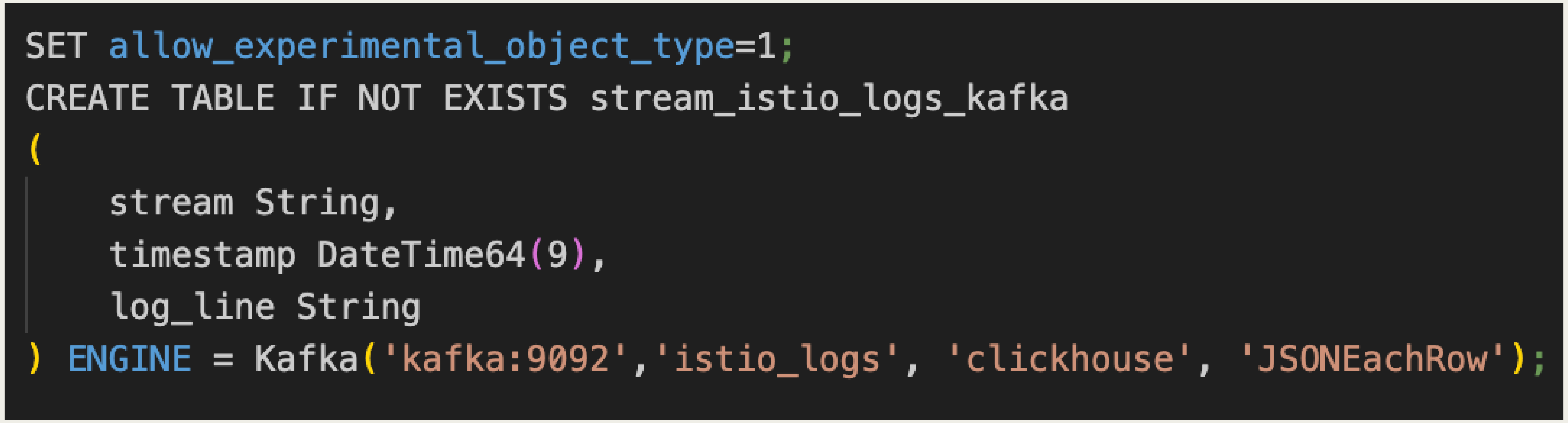


Unlocking NextGen Log Analytics With ClickHouse and Kafka

OSA CON 23

DATA MODEL

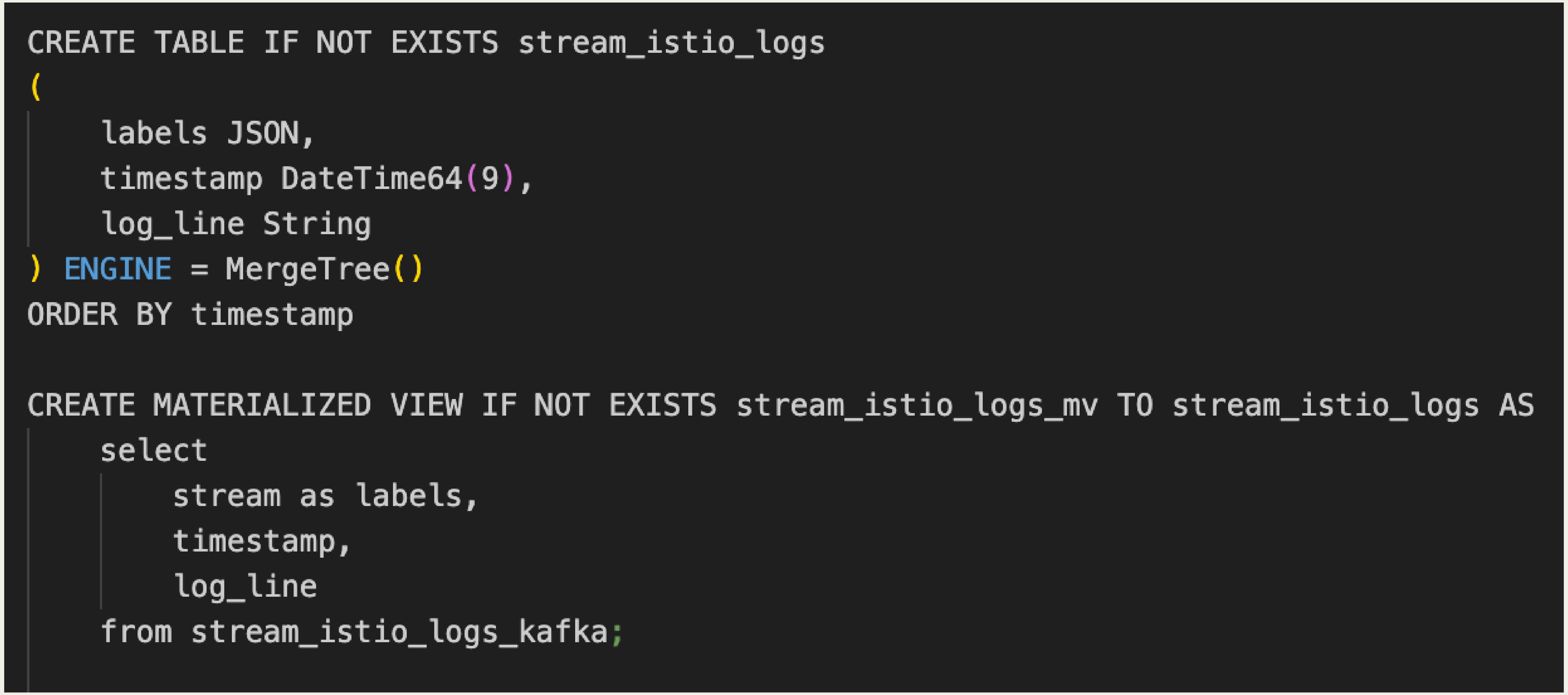
Kafka Engine Table



```
SET allow_experimental_object_type=1;
CREATE TABLE IF NOT EXISTS stream_istio_logs_kafka
(
    stream String,
    timestamp DateTime64(9),
    log_line String
) ENGINE = Kafka('kafka:9092','istio_logs', 'clickhouse', 'JSONEachRow');
```

DATA MODEL

MergeTree Table and Materialized View

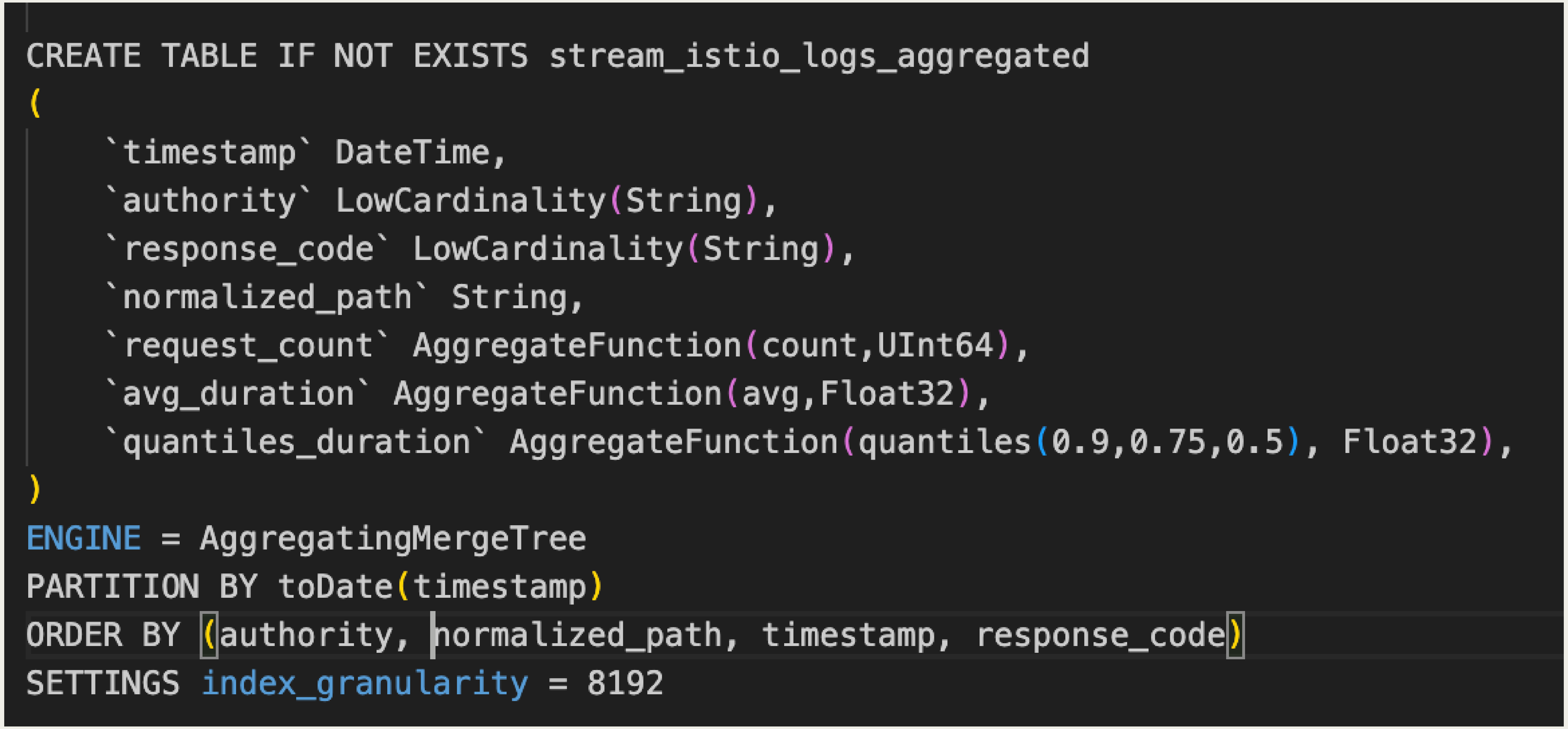


```
CREATE TABLE IF NOT EXISTS stream_istio_logs
(
    labels JSON,
    timestamp DateTime64(9),
    log_line String
) ENGINE = MergeTree()
ORDER BY timestamp

CREATE MATERIALIZED VIEW IF NOT EXISTS stream_istio_logs_mv TO stream_istio_logs AS
    select
        stream as labels,
        timestamp,
        log_line
    from stream_istio_logs_kafka;
```

DATA MODEL

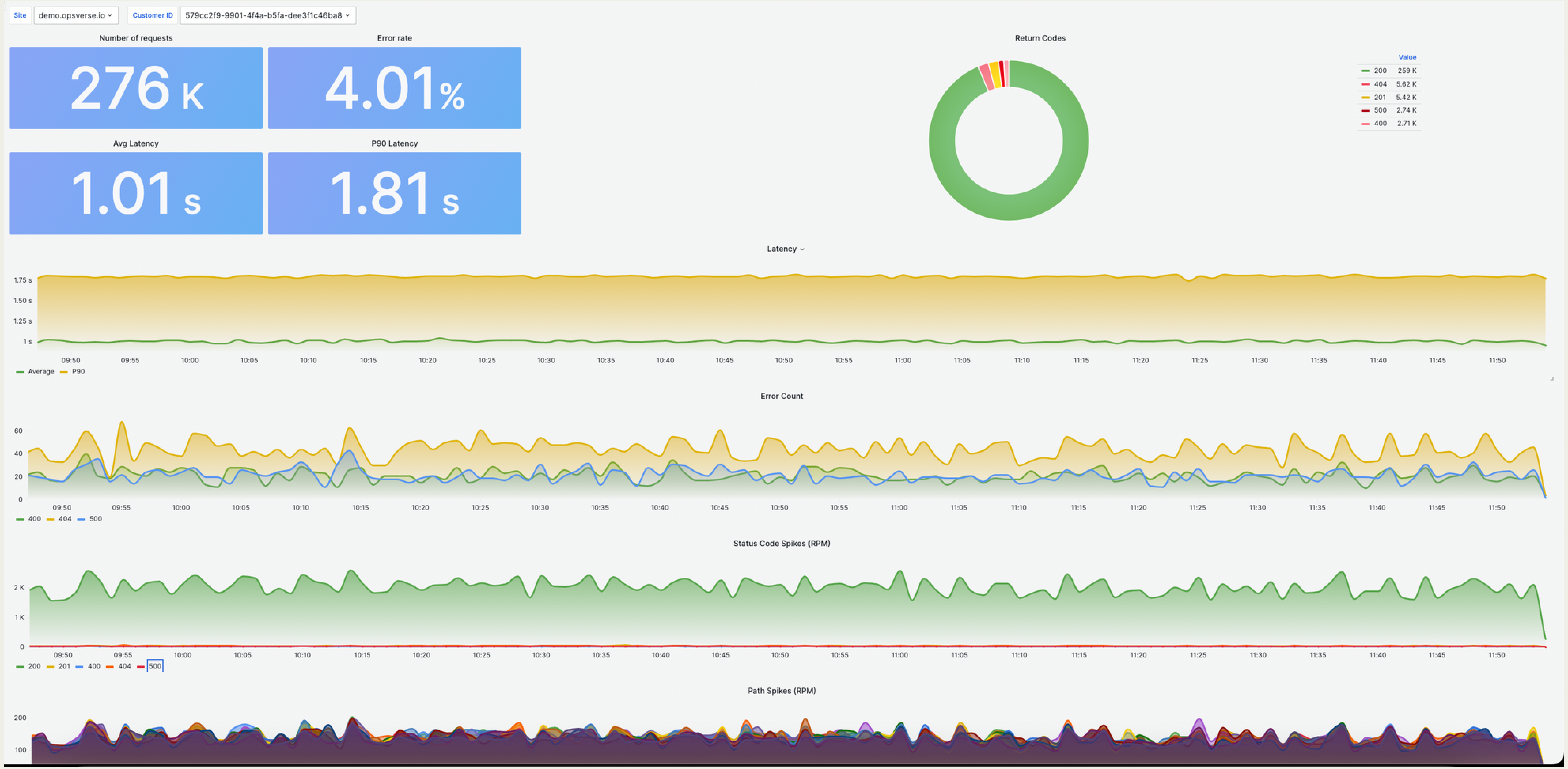
AggregatedMergeTree Table



```
CREATE TABLE IF NOT EXISTS stream_istio_logs_aggregated
(
    `timestamp` DateTime,
    `authority` LowCardinality(String),
    `response_code` LowCardinality(String),
    `normalized_path` String,
    `request_count` AggregateFunction(count,UInt64),
    `avg_duration` AggregateFunction(avg,Float32),
    `quantiles_duration` AggregateFunction(quantiles(0.9,0.75,0.5), Float32),
)
ENGINE = AggregatingMergeTree
PARTITION BY toDate(timestamp)
ORDER BY (authority, normalized_path, timestamp, response_code)
SETTINGS index_granularity = 8192
```

RESULTS

Istio Logs



Site: demo.opsverse.io

Customer ID: 579cc2f9-9901-4f4a-b5fa-dee3f1c46ba8

Number of requests: 276 K

Error rate: 4.01%

Avg Latency: 1.01 s

P90 Latency: 1.81 s

Return Codes

	Value
200	259 K
404	5.62 K
201	5.42 K
500	2.74 K
400	2.71 K

CONCLUSION

- Logs are everywhere
- We need a cost effective, yet scalable way to analyze them
- ClickHouse and kafka together offer a solution!

OPSVERSE

Thank you!

@arul-jegadish
arul@opsverse.io