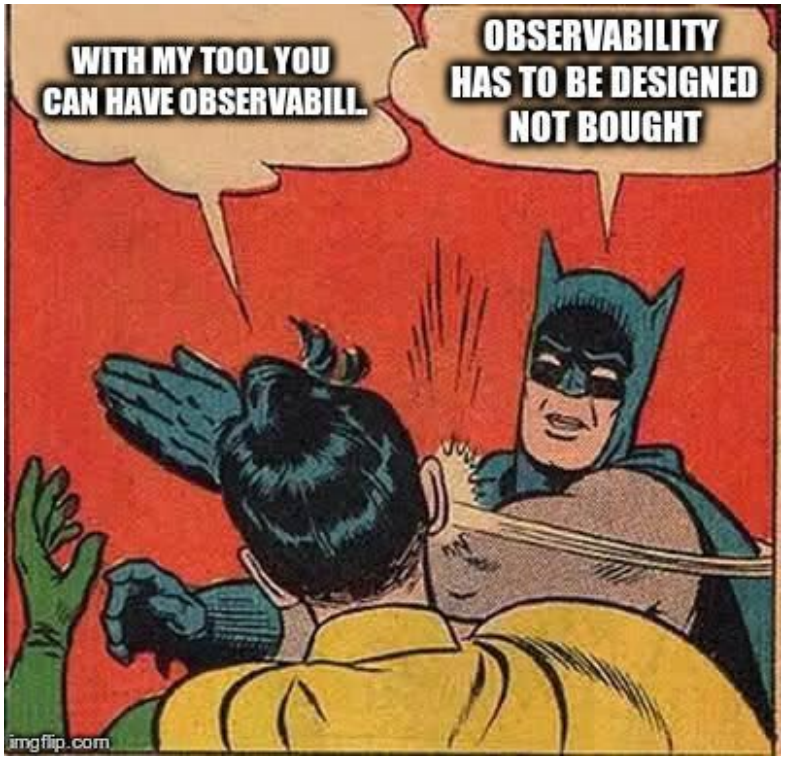

Well Hello There...



End-to-End Open Source Observability
Custom-built on **Your Infrastructure**

Build **observability that adapts to your infrastructure**
Not the other way around.


Other databases
Clickhouse
Incerto
Observability stack

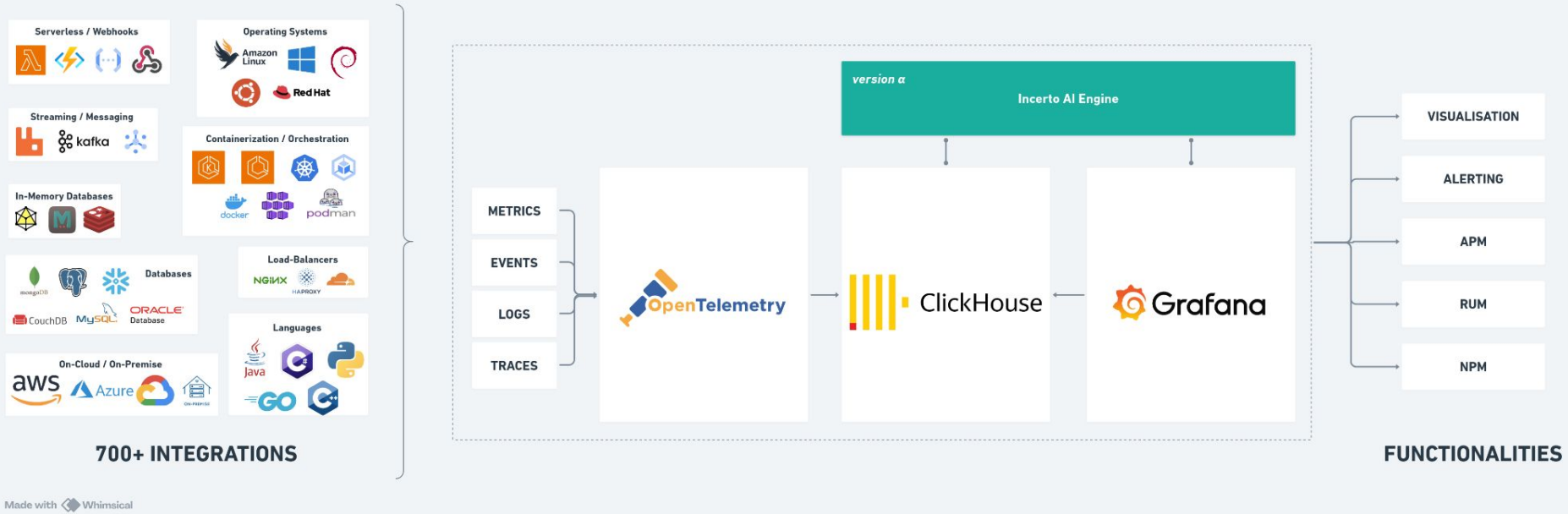
The trifecta of 3 OSS tools

Single backend framework for capturing telemetry data (vender-agnostic)

Telemetry data retention

UI for visualization/alerting

Architecture



What have we achieved with Clickhouse ?

Sub-second Queries : Includes high cardinality groupings, filtering via various keys, aggregation queries & more.

Efficient Storage : Upto **20x** compression with S3 storage while juggling TBs of data every month.

QUESTION

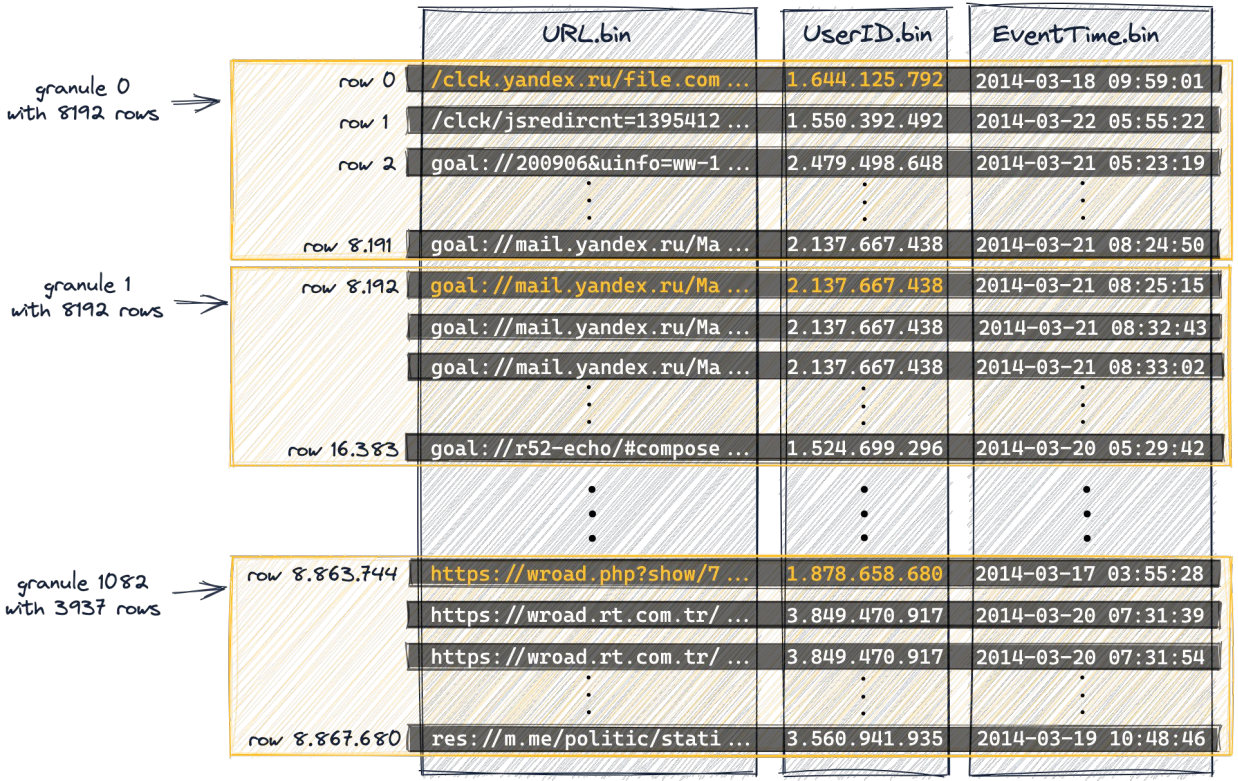
How to configure Clickhouse for fast queries and efficient storage?

ANSWER

Understanding your query patterns, data and ***somethings about Clickhouse***


LETS DIVE IN
imgflip.com

Explain filtering basics in clickhouse



We want this query to run fast!!

```
SELECT
    timestamp, traceId, spanId, symbolName, logBody,
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    traceId = `3b047adbf6964601e01688f31b2f0d7a`
AND
    timestamp BETWEEN t1 and t2
```

Focus on filtering columns

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    traceId = `3b047adbf6964601e01688f31b2f07a`
AND
    timestamp BETWEEN t1 and t2
```

Hierarchy and Cardinality

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    traceId = `3b047adbf6964601e01688f31b2f07a`
AND
    timestamp BETWEEN t1 and t2
```

cardinality = 2

cardinality > billions

cardinality > billions

Hierarchy and Cardinality

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    traceId = `3b047adbf6964601e01688f31b2f07a`
AND
    timestamp BETWEEN t1 and t2
```

cardinality = 2

cardinality > billions

cardinality > billions

Rule1 : Increasing order of cardinality

Hierarchy and Cardinality

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    traceId = `3b047adbf6964601e01688f31b2f07a`
AND
    timestamp BETWEEN t1 and t2
```

cardinality = 2

cardinality > billions

cardinality > billions

Rule1 : Increasing order of cardinality

Rule2 : Atmost 1 high cardinality key

PrimaryKey := (exchId, traceId)

Hierarchy and Cardinality

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    traceId = `3b047adbf6964601e01688f31b2f07a`
AND
    timestamp BETWEEN t1 and t2
```

cardinality = 2

cardinality > billions

cardinality > billions

Rule1 : Increasing order of cardinality

Rule2 : Atmost 1 high cardinality key

PrimaryKey := (exchId, traceId)

No TraceId

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    timestamp BETWEEN t1 and t2
```

cardinality = 2

cardinality > billions

Enters Partition Key

PrimaryKey := (exchId, traceId)

PartitionKey := toDate(timestamp)

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    timestamp BETWEEN t1 and t2
```

cardinality = 2

cardinality > billions

Enters Min Max Skip Indices

PrimaryKey := (exchId, traceId)

PartitionKey := toDate(timestamp)

addIndex := timestamp(type=Minmax, Granularity=3)

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    timestamp BETWEEN t1 and t2
```

cardinality = 2

cardinality > billions

Random High Cardinality

PrimaryKey := (exchId, traceId)

PartitionKey := toDate(timestamp)

addIndex := timestamp(type=Minmax, Granularity=3)

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    timestamp BETWEEN t1 and t2
AND
    spanId = fa1152363fac23bf
```

cardinality = 2

cardinality > billions

cardinality > billions

Random High Cardinality

```
SELECT
    timestamp,traceId,spanId,symbol,logBody
FROM
    brokerage.orders_logs
WHERE
    exchID = NSE
AND
    timestamp BETWEEN t1 and t2
AND
    spanId = fa1152363fac23bf
```

cardinality = 2

cardinality > billions

cardinality > billions

PrimaryKey := (exchId, traceId)

PartitionKey := toDate(timestamp)

addIndex := timestamp(type=Minmax, Granularity=3)

addIndex := spanId(type=bloom_filter, p=0.01)


GO ON.
TELL ME MORE

Quick Recap

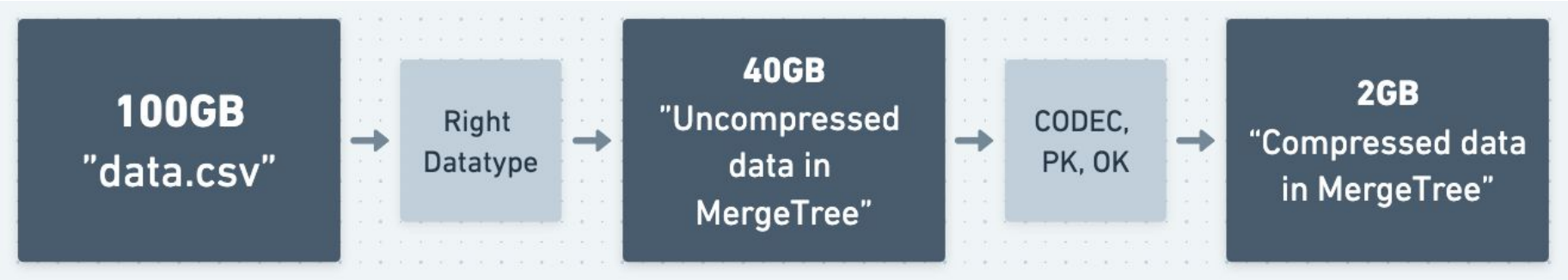
Queries: List down queries in order of priority. Note down the filtering columns.

Data: Know types of columns, number of unique values, range. And for the noted columns only, understand the hierarchy – through grouping, frequencies etc.

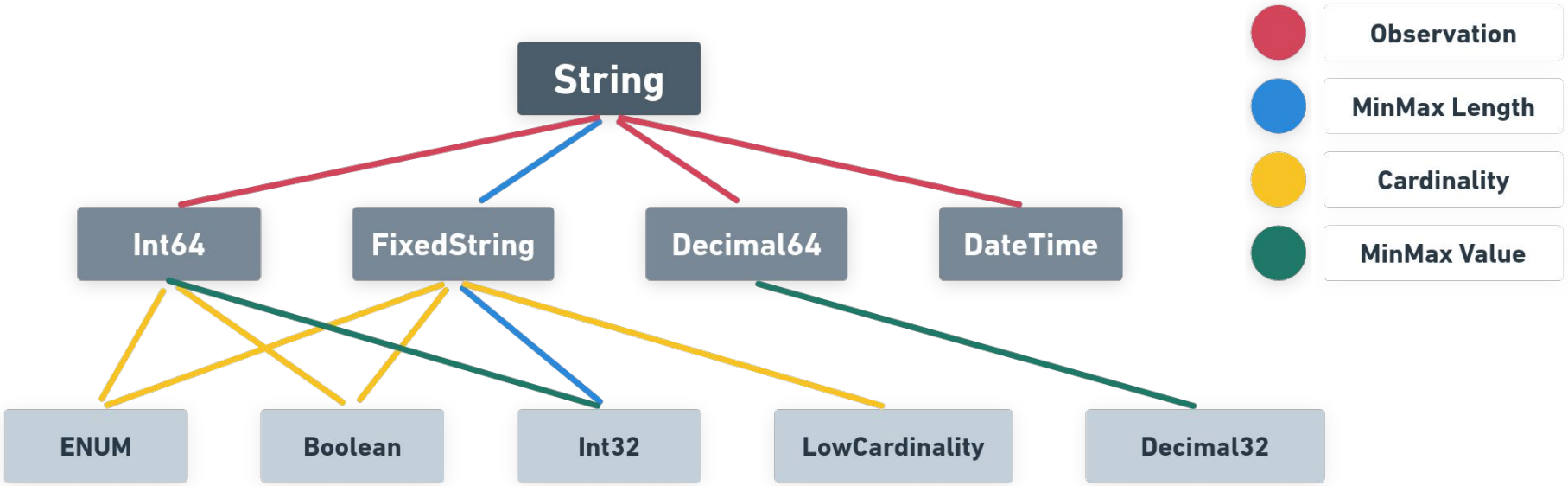
“Somethings”: Primary keys, Ordering keys, Skip indexes, Materialized views and Projections.

Storage Optimisations

How to go from 100GB to 2GB ?



How to get that **60%**?



Use ZSTD(1)

Use Right Primary Key

Use Right Ordering Key

Top Brokerage firm of India

Constraints

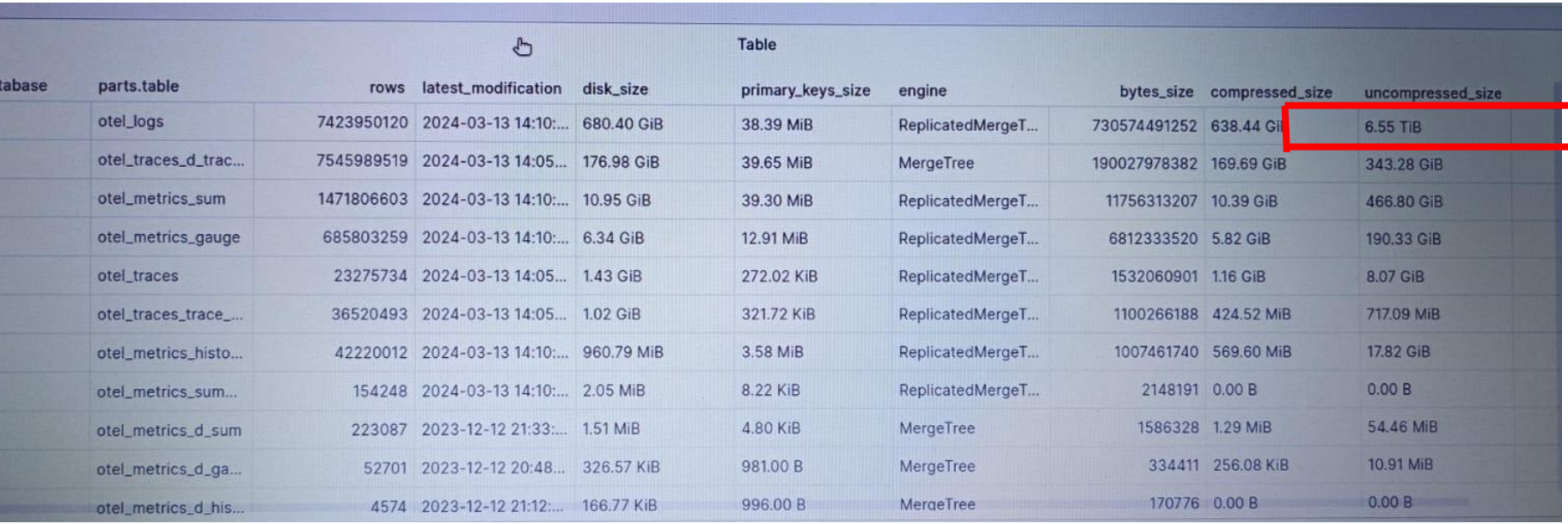
- **~300GB** logs data/day
- **40TB** telemetry data/month
- Data retention for **7+** years
- **SEBI** compliance specific needs w.r.t auditing
- High **$$$** monthly costs with other vendors

Result

“2 SHARD 2 REPLICA”
CLICKHOUSE CLUSTER

4 x (16vCPU 32GB RAM)
(c7a.4xlarge/c7g.4xlarge)

Stage Clickhouse DB snapshot



Table

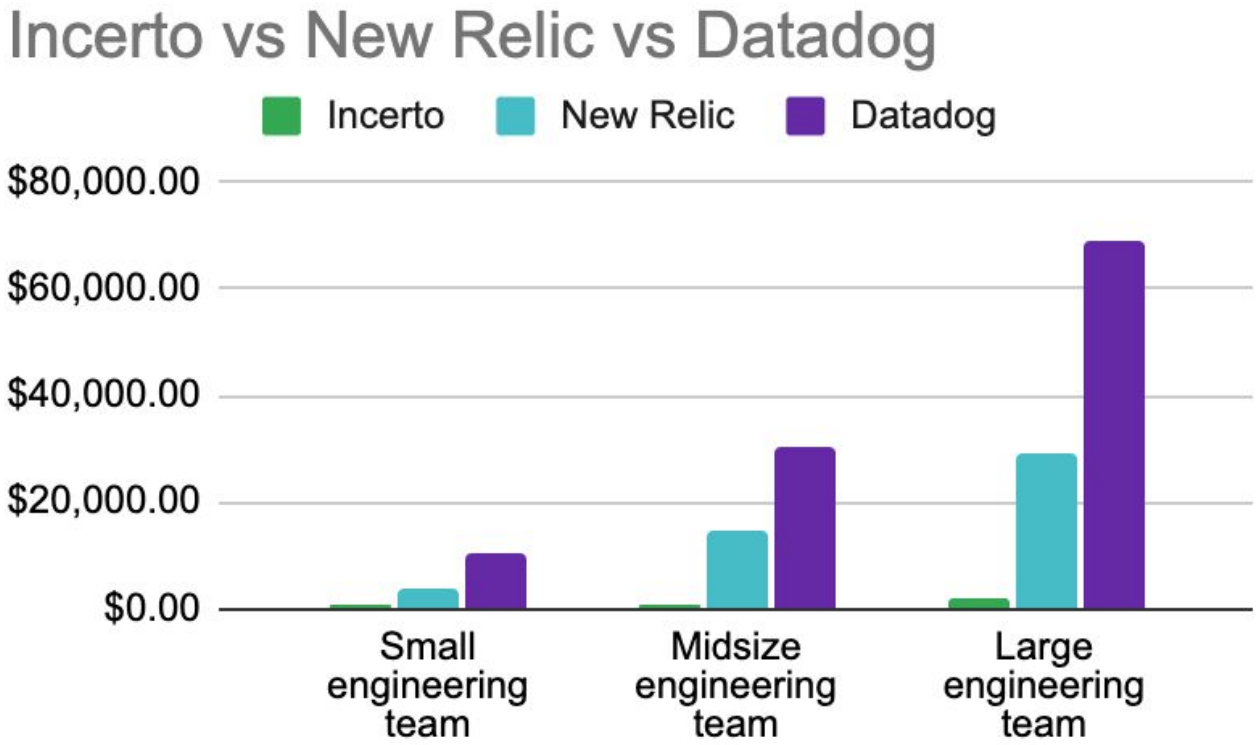
tabase	parts.table	rows	latest_modification	disk_size	primary_keys_size	engine	bytes_size	compressed_size	uncompressed_size
	otel_logs	7423950120	2024-03-13 14:10:...	680.40 GiB	38.39 MiB	ReplicatedMergeT...	730574491252	638.44 GiB	6.55 TiB
	otel_traces_d_trac...	7545989519	2024-03-13 14:05...	176.98 GiB	39.65 MiB	MergeTree	190027978382	169.69 GiB	343.28 GiB
	otel_metrics_sum	1471806603	2024-03-13 14:10:...	10.95 GiB	39.30 MiB	ReplicatedMergeT...	11756313207	10.39 GiB	466.80 GiB
	otel_metrics_gauge	685803259	2024-03-13 14:10:...	6.34 GiB	12.91 MiB	ReplicatedMergeT...	6812333520	5.82 GiB	190.33 GiB
	otel_traces	23275734	2024-03-13 14:05...	1.43 GiB	272.02 KiB	ReplicatedMergeT...	1532060901	1.16 GiB	8.07 GiB
	otel_traces_trace_...	36520493	2024-03-13 14:05...	1.02 GiB	321.72 KiB	ReplicatedMergeT...	1100266188	424.52 MiB	717.09 MiB
	otel_metrics_histo...	42220012	2024-03-13 14:10:...	960.79 MiB	3.58 MiB	ReplicatedMergeT...	1007461740	569.60 MiB	17.82 GiB
	otel_metrics_sum...	154248	2024-03-13 14:10:...	2.05 MiB	8.22 KiB	ReplicatedMergeT...	2148191	0.00 B	0.00 B
	otel_metrics_d_sum	223087	2023-12-12 21:33:...	1.51 MiB	4.80 KiB	MergeTree	1586328	1.29 MiB	54.46 MiB
	otel_metrics_d_ga...	52701	2023-12-12 20:48...	326.57 KiB	981.00 B	MergeTree	334411	256.08 KiB	10.91 MiB
	otel_metrics_d_his...	4574	2023-12-12 21:12:...	166.77 KiB	996.00 B	MergeTree	170776	0.00 B	0.00 B

Retention Cost per GB per month

	A	B	C	D	E
1	*Service*	1GB	**10GB**	100GB	1TB
2	Datadog 😭	$5.080	$50.805	$508.047	$5,080.469
3	Honeycomb	$2.539	$25.391	$253.906	$2,539.063
4	Logz	$1.170	$11.700	$117.000	$1,170.000
5	Coralogix	$1.150	$11.500	$115.000	$1,150.000
6	CW logs	$0.530	$5.300	$53.000	$530.000
7	Highlight	$0.50	$5.00	$50.00	$500.00
8	Grafana	$0.50	$5.00	$50.00	$500.00
9	HyperDX	$0.40	$4.00	$40.00	$400.00
10	Signoz	$0.30	$3.00	$30.00	$300.00
11	NewRelic 😭	$0.300	$3.000	$30.000	$300.000
12	BetterStack	$0.250	$2.500	$25.000	$250.000
13	Axiom	$0.150	$1.500	$15.000	$150.000
14	Clickhouse 😎	$0.047	$0.471	$4.710	$47.100
15	S3 Standard	$0.023	$0.230	$2.300	$23.000

Ref: https://bit.kevinslin.com/i/140460591/volume-based-fixed-cost-comparison

Incerto’s stack pricing



traceID All | spanID All | SeverityText All | ServiceName All | SpanTypes All | AttributeValue All

Process | Host Metrics | Anomaly | Otel Metrics | Latency | Clickhouse Query Analysis

Traces & Logs

Traces

service-A: GET /service-A/items 262.21ms

Give feedback | Trace ID | Export

2024-03-16 12:01:38.442 GET 200 /check

Span Filters | 100 spans | Prev | Next

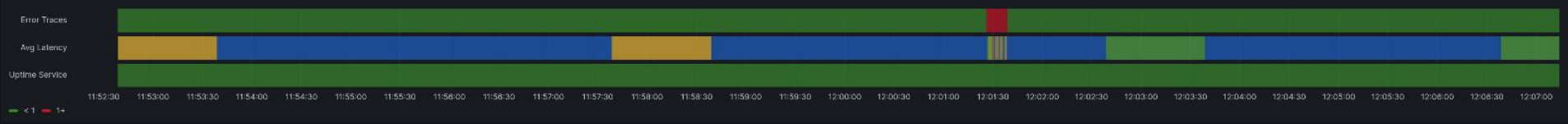
GET /check http send (800.09µs)	800.09µs
GET /check http send (2.44ms)	2.44ms
service-A GET (17.19ms)	17.19ms
service-C GET /service-C/items (13.2ms)	13.2ms
GET (2.99ms)	2.99ms
go-service /region (68.09µs)	68.09µs
getRegion (1.61µs)	1.61µs
service-C GET (2.42ms)	2.42ms
go-service /version (55.3µs)	55.3µs
getVersion (1.25µs)	1.25µs
service-C SELECT (1.71ms)	1.71ms
GET /service-C/items http send (519.81µs)	519.81µs
GET /service-C/items http send (212.61µs)	212.61µs
service-A GET /service-A/items http send (761.31µs)	761.31µs
GET /service-A/items http send (279.3µs)	279.3µs
GET /service-A/items (39.25ms)	39.25ms
GET (6.47ms)	6.47ms
service-B GET /check (2.4ms)	2.4ms
GET /check http send (731.33µs)	731.33µs
GET /check http send (259.97µs)	259.97µs

All Logs

```
2024-03-16 12:01:38.702 http://localhost:8003 'POST /service-C/items HTTP/1.1' 200 58
2024-03-16 12:01:38.694 Add item called
2024-03-16 12:01:38.691 Starting new HTTP connection (1): localhost:8003
2024-03-16 12:01:38.688 http://localhost:8002 'GET /check?role=admin HTTP/1.1' 200 16
2024-03-16 12:01:38.685 Authorization API Called
2024-03-16 12:01:38.683 Starting new HTTP connection (1): localhost:8002
2024-03-16 12:01:38.665 Error Price can't be less than 100
2024-03-16 12:01:38.653 http://localhost:8003 'POST /service-C/items HTTP/1.1' 200 58
2024-03-16 12:01:38.644 Add item called
2024-03-16 12:01:38.641 Starting new HTTP connection (1): localhost:8003
2024-03-16 12:01:38.638 http://localhost:8002 'GET /check?role=admin HTTP/1.1' 200 16
2024-03-16 12:01:38.635 Authorization API Called
2024-03-16 12:01:38.632 Starting new HTTP connection (1): localhost:8002
2024-03-16 12:01:38.615 Error Price can't be less than 100
```

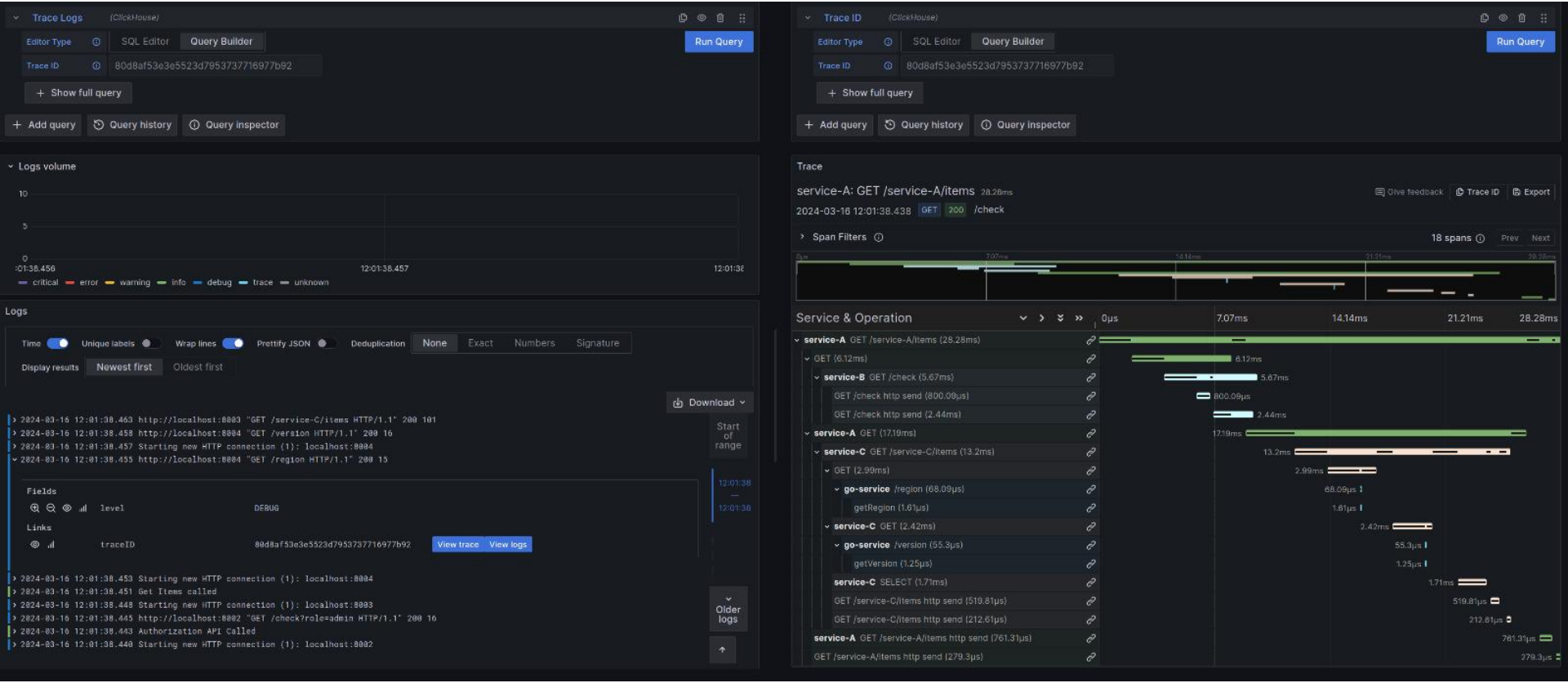
Service Graph

Availability



Service Info

ServiceName	errorTraces	errorLogs	ResourceInfo	SpanInfo	LogInfo	ResourceInfoLog
service-A	265	143	{ "host.name": ["example-host"], "o	{ "http.flavor": ["1.1"], "http.host	{ "otelServiceName": ["service-A"],	{ "host.name": ["example-host"], "o
service-B	202	67	{ "host.name": ["example-host"], "o	{ "http.flavor": ["1.1"], "http.host	{ "otelServiceName": ["service-B"],	{ "host.name": ["example-host"], "o
service-C	119	45	{ "host.name": ["example-host"], "o	{ "http.flavor": ["1.1"], "http.host	{ "otelServiceName": ["service-C"],	{ "host.name": ["example-host"], "o
go-service	43	0	{ "host.name": ["example-host"], "o	{ "http.method": ["GET"], "http.rout	{}	


Trace Logs
(ClickHouse)
Editor Type
SQL Editor
Query Builder
Run Query
Trace ID
80d8af53e3e5523d7953737716977b92
Show full query
Add query
Query history
Query inspector
Logs volume
10
5
0
12:01:38.456
12:01:38.457
critical
error
warning
info
debug
trace
unknown
Logs
Time
Unique labels
Wrap lines
Prettify JSON
Deduplication
None
Exact
Numbers
Signature
Display results
Newest first
Oldest first
Download
2024-03-16 12:01:38.463 http://localhost:8003 "GET /service-C/items HTTP/1.1" 200 101
2024-03-16 12:01:38.458 http://localhost:8004 "GET /version HTTP/1.1" 200 16
2024-03-16 12:01:38.457 Starting new HTTP connection (1): localhost:8004
2024-03-16 12:01:38.455 http://localhost:8004 "GET /region HTTP/1.1" 200 15
Fields
level
DEBUG
Links
traceID
80d8af53e3e5523d7953737716977b92
View trace
View logs
Start of range
12:01:38
12:01:38
2024-03-16 12:01:38.453 Starting new HTTP connection (1): localhost:8004
2024-03-16 12:01:38.451 Get Items called
2024-03-16 12:01:38.448 Starting new HTTP connection (1): localhost:8003
2024-03-16 12:01:38.445 http://localhost:8002 "GET /check?role=admin HTTP/1.1" 200 16
2024-03-16 12:01:38.443 Authorization API Called
2024-03-16 12:01:38.440 Starting new HTTP connection (1): localhost:8002
Older logs
Trace ID
(ClickHouse)
Editor Type
SQL Editor
Query Builder
Run Query
Trace ID
80d8af53e3e5523d7953737716977b92
Show full query
Add query
Query history
Query inspector
Trace
service-A: GET /service-A/items 28.28ms
Give feedback
Trace ID
Export
2024-03-16 12:01:38.438
GET
200
/check
Span Filters
18 spans
Prev
Next
Service & Operation
0µs
7.07ms
14.14ms
21.21ms
28.28ms
service-A GET /service-A/items (28.28ms)
GET (6.12ms)
6.12ms
service-B GET /check (5.67ms)
5.67ms
GET /check http send (800.09µs)
800.09µs
GET /check http send (2.44ms)
2.44ms
service-A GET (17.19ms)
17.19ms
service-C GET /service-C/items (13.2ms)
13.2ms
GET (2.99ms)
2.99ms
go-service /region (68.09µs)
68.09µs
getRegion (1.61µs)
1.61µs
service-C GET (2.42ms)
2.42ms
go-service /version (55.3µs)
55.3µs
getVersion (1.25µs)
1.25µs
service-C SELECT (1.71ms)
1.71ms
GET /service-C/items http send (519.81µs)
519.81µs
GET /service-C/items http send (212.61µs)
212.61µs
service-A GET /service-A/items http send (761.31µs)
761.31µs
GET /service-A/items http send (279.3µs)
279.3µs


Dynamic TimeSeries Grouping
Service
service-A
service-B
service-C
go-service
Host
example-host
HTTP Target
errorTraces
http://127.0.0.1:8002/check?role=user
http://localhost:8002/check?role=user
http://127.0.0.1:8001/service-A/items
http://127.0.0.1:8001/service-A/item
http://127.0.0.1:8002/service-B/error
http://127.0.0.1:8003/service-C/error
http://localhost:8004/region
http://localhost:8004/version
HTTP Method
GET
errorTraces
POST
CONNECT
Host Port
errorTraces
8001
8002
8004
8003
Panel Title
Panel Title

ServiceName All | SeverityText All | AttributeValue All | FirstSpan All | SpanName All | GroupInterval_secs 300 | traceID ALL | Traces and Logs

Panel Title

service-A: GET / 1d 1h

Give feedback | Trace ID | Export

2024-03-15 11:17:23.299 GET 200 /

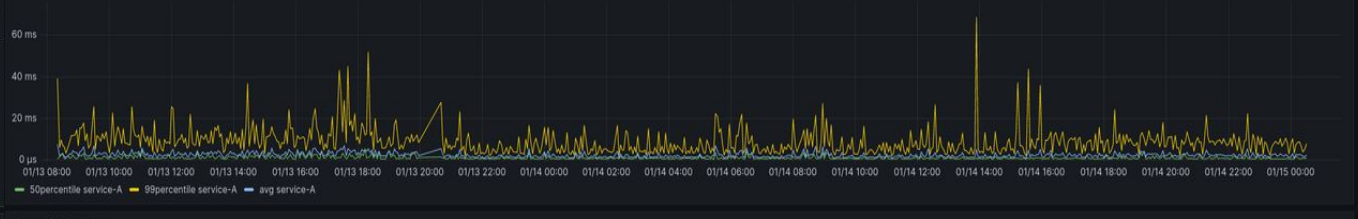
Span Filters

158 spans | Prev | Next

0µs | 6h 12m | 12h 24m | 18h 36m | 1d 1h

Service & Operation	3h	3h	3h	3h	4h
	7m	23m	38m	54m	9m
service-A GET / (35.06ms)					
service-B GET / (27.68ms)					
GET / http send (13.61ms)					
service-C GET / (28.3ms)					
service-A GET /service-A/items (53.33ms)					
GET (43.99ms)					

Grouped Latencies

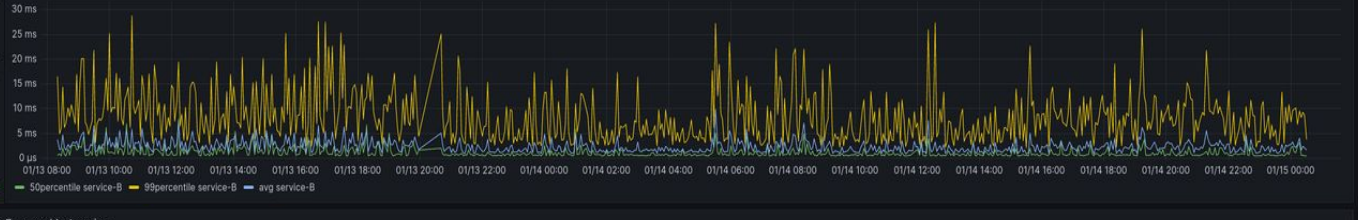


Slow APIs

SpanName	max(Duration)
GET /service-A/items	115455685
POST /service-A/items	78798503
GET	72389129
POST	48266492

‹ 1 2 3 › 1 - 4 of 10 rows

Grouped Latencies

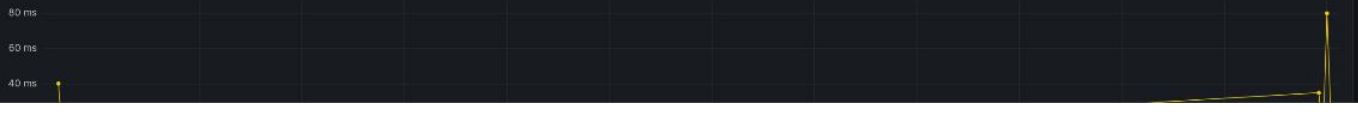


Slow APIs

SpanName	max(Duration)
GET /	54373894
GET /check	18795241
GET / http send	16566745
GET	6411140

‹ 1 2 › 1 - 4 of 8 rows

Grouped Latencies

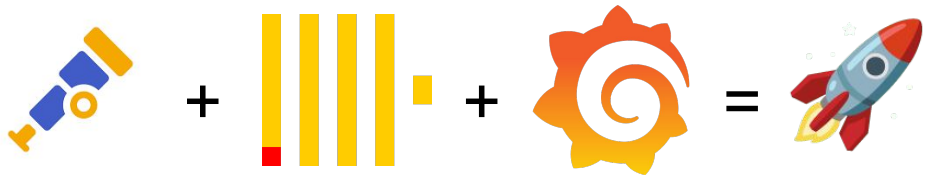


Slow APIs

SpanName	max(Duration)
GET /service-C/items	80197457
GET /	35115597
SELECT	30472162

The key thing
we want you take home today is

You can ABSOLUTELY build your own observability



Or you can just come to us 😎

Shiva Pundir

Co-Founder

Incerto Technologies