



ROKT

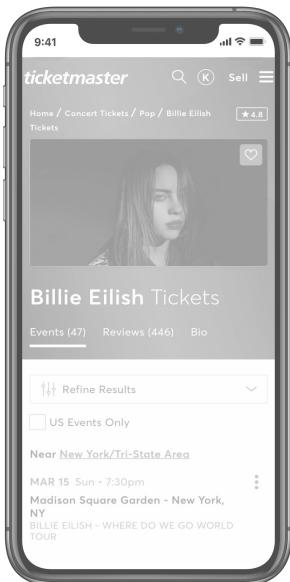
Building the
future of reporting
at Rokt

December 2022

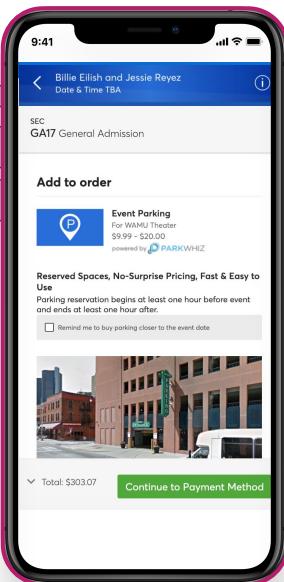
Vadim Semenov

We reach customers as they're actively buying in the Transaction Moment

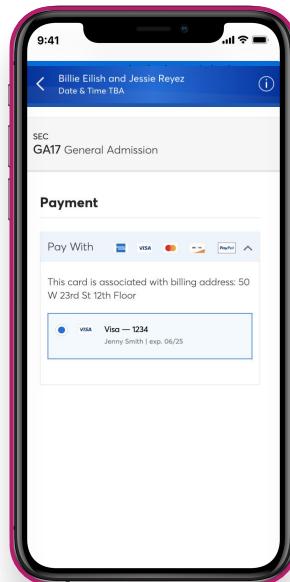
Select



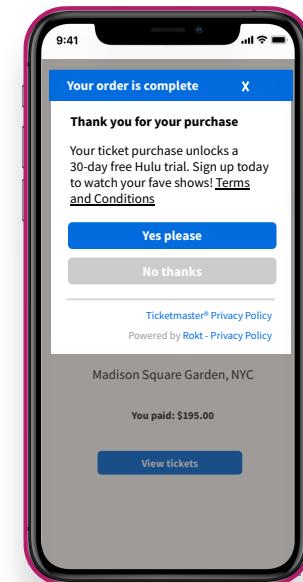
Upsell



Payment



Confirmation



Rokt connects the world's leading brands to e-commerce

Transactions
per year

E-COMMERCE

1B+

THE TRANSACTION MOMENT

MARKETING

14%

Engagement
Rate

WORLD'S BEST E-COMMERCE SITES

ticketmaster®

 **Domino's**

GROUPON

JCPenney

ebay

 **LIVE
NATION**®

axs

 **spirit
airlines**

 **Fanatics**



Hotels.com





WORLD'S BIGGEST BRANDS





GROUPON



Booking.com

hulu




**UNI
QLO**




TOYOTA


DOLLAR SHAVE CLUB

Gillette

Powerful tools to enhance performance



Smart Bidding

Utilize the **Rokt algorithm** to determine the bid price for your campaign at an auction level.



Incrementality

Isolate revenue to compare costs **across marketing channels** and understand the ROI of Rokt.



Custom audiences

Be more granular with targeting to provide the **most relevant and engaging experiences**

The Rokt ecosystem is two sides of the same platform

Rokt **Ecommerce**

Clients = Partners



Product Selection

Review & Upsell

Payment & Shipping

Confirmation Page

Account Management

Rokt **Calendar**

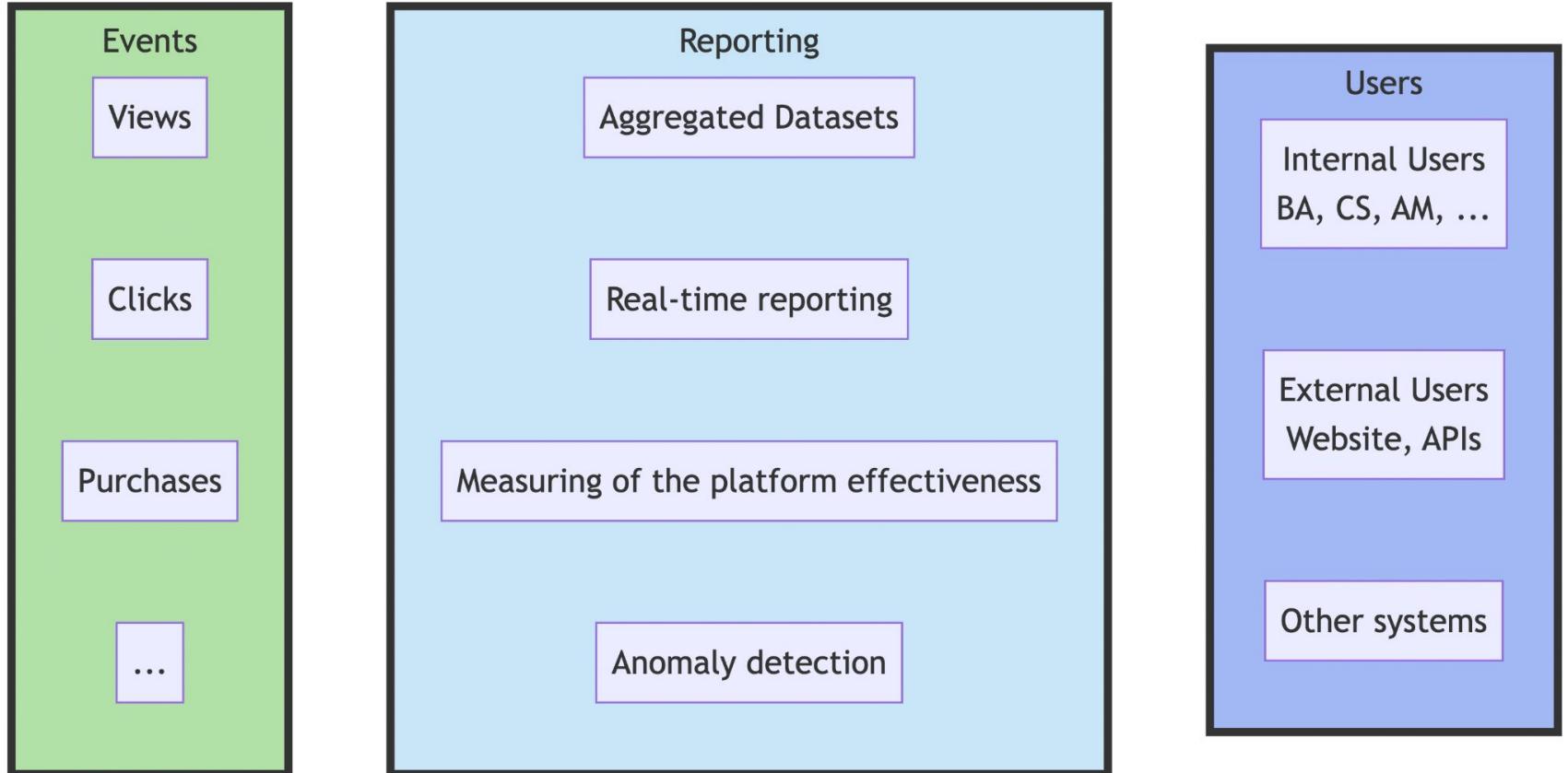
Clients = Partners

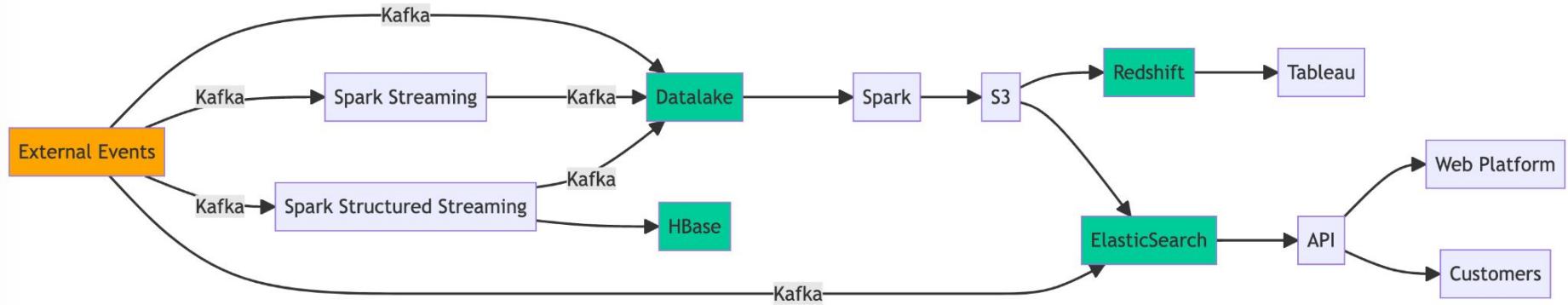
Rokt **Distributed Commerce**

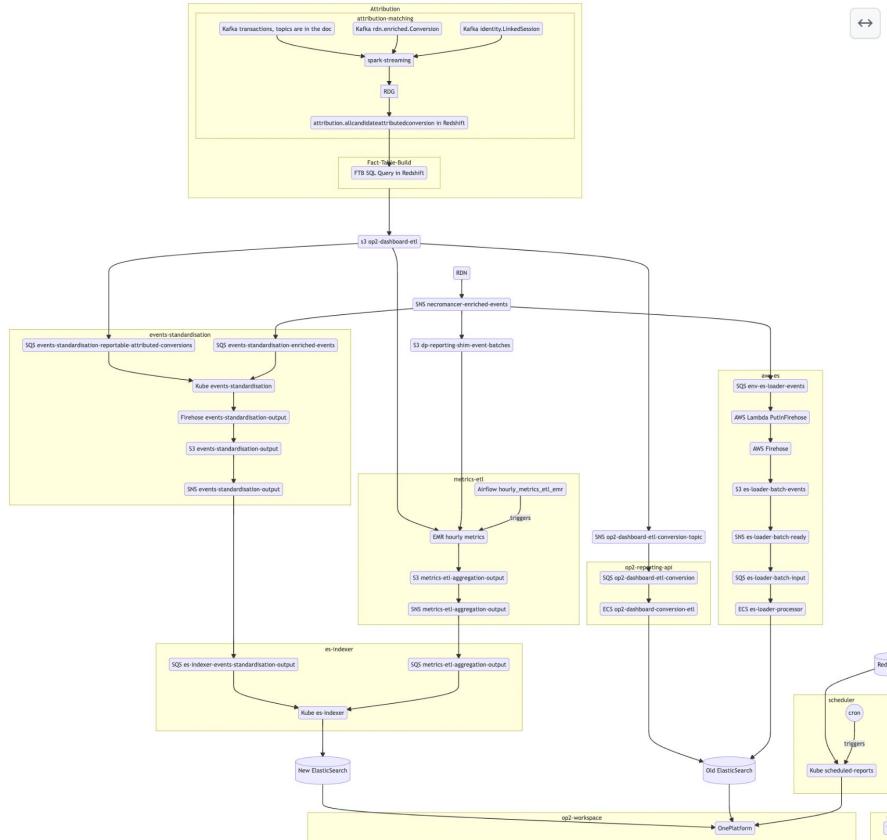
Clients = Providers

Rokt **Ads**

Clients = Advertisers







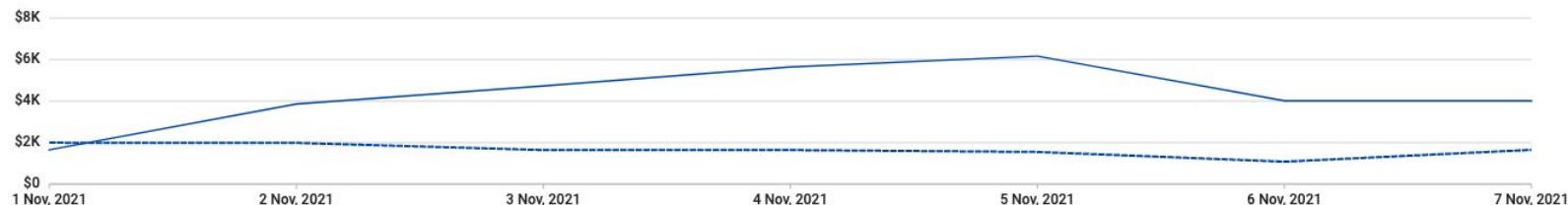
My Company

Search for accounts

**Campaigns**
USD ▼

 Nov 01, 2021 → Nov 07, 2021 UTC+11 Custom 📅

Gross cost ⓘ	Impressions ⓘ	Referrals ⓘ	Click thru acquisitions ⓘ	Referral rate ⓘ	Conversion rate ⓘ	CPR ⓘ	CPA ⓘ
\$30,058.43 ↑ 160.5%	146,450 ↑ 147.5%	16,946 ↑ 160.9%	1,449 ↑ 149.4%	11.57% ↑ 5.4%	8.55% ↓ -4.4%	\$1.77 ↓ -0.2%	\$20.74 ↑ 4.5%

— Gross cost .. Previous period


2 Campaigns		21 Audiences		15 Creatives			
New Campaign	Status is All but Archived	Add filter for campaigns		Columns	Download	Segment	Graph
<input type="checkbox"/>	Campaign name	Status	Gross cost	Impressions	Referrals	Click thru acquisitions	
Total for 2 Campaigns			\$30,058.43	146,450	16,946	1,449	
<input type="checkbox"/>	Campaign A Website traffic • Ads • Campaign ID	✎ ● Live ▼	\$28,499.74	134,780	15,745	1,449	

Issues with the current setup

Customers have limited abilities to slice&dice data

== difficult to get data that you want

== more requests for custom reporting

== less work we and the BA team can spend on their first-tier tasks

Customers have limited abilities to slice&dice data
== difficult to get data that you want
== we need new UI
== we need a new database that's going to support various queries
with lots of GROUP BYs

- It's difficult to overall bring new dimensions to Elasticsearch
 - Not easy to ingest
 - No joins, so labels must exist somewhere else
- Difficult to query as you have to do JSON, a gateway must exist
 - That leads to data duplication
- Difficulties with deduplication and backfills

Cluster configuration

Data nodes

Availability Zones

3-AZ

Instance type

i3.2xlarge.search

Number of nodes

8

Storage type

Instance (default)

Dedicated master nodes

Enabled

Yes

Instance type

c5.xlarge.search

Number of nodes

3

Warm and cold data storage

UltraWarm data nodes enabled

One cluster

8x i3.2xlarge.search + 3x c5.xlarge

8x\$0.998+3x\$0.251

~\$6500/month





Apache Pinot, Apache Druid — more real-time focused, no joins*,
limited SQL

Snowflake — more of a data-warehouse, expensive

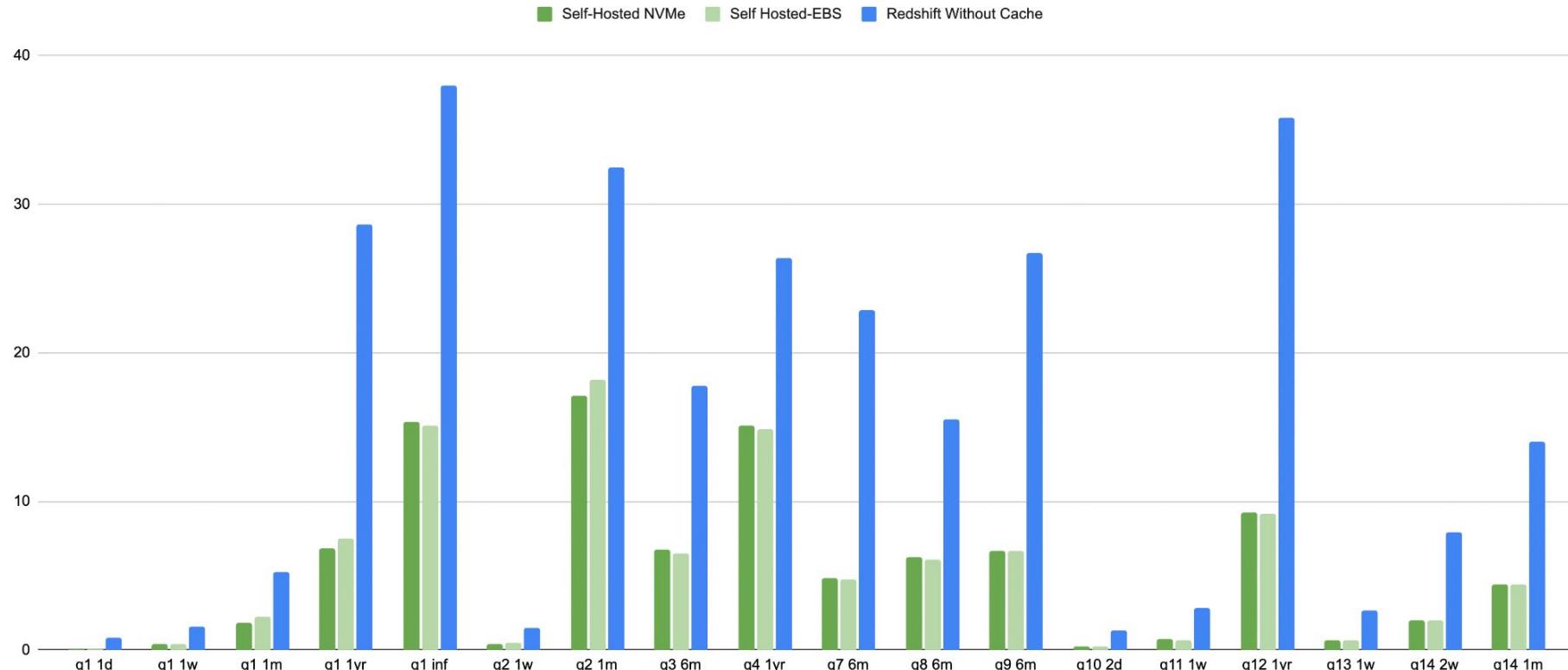
Starrocks — too fresh

Citus — too postgres and microsoft oriented

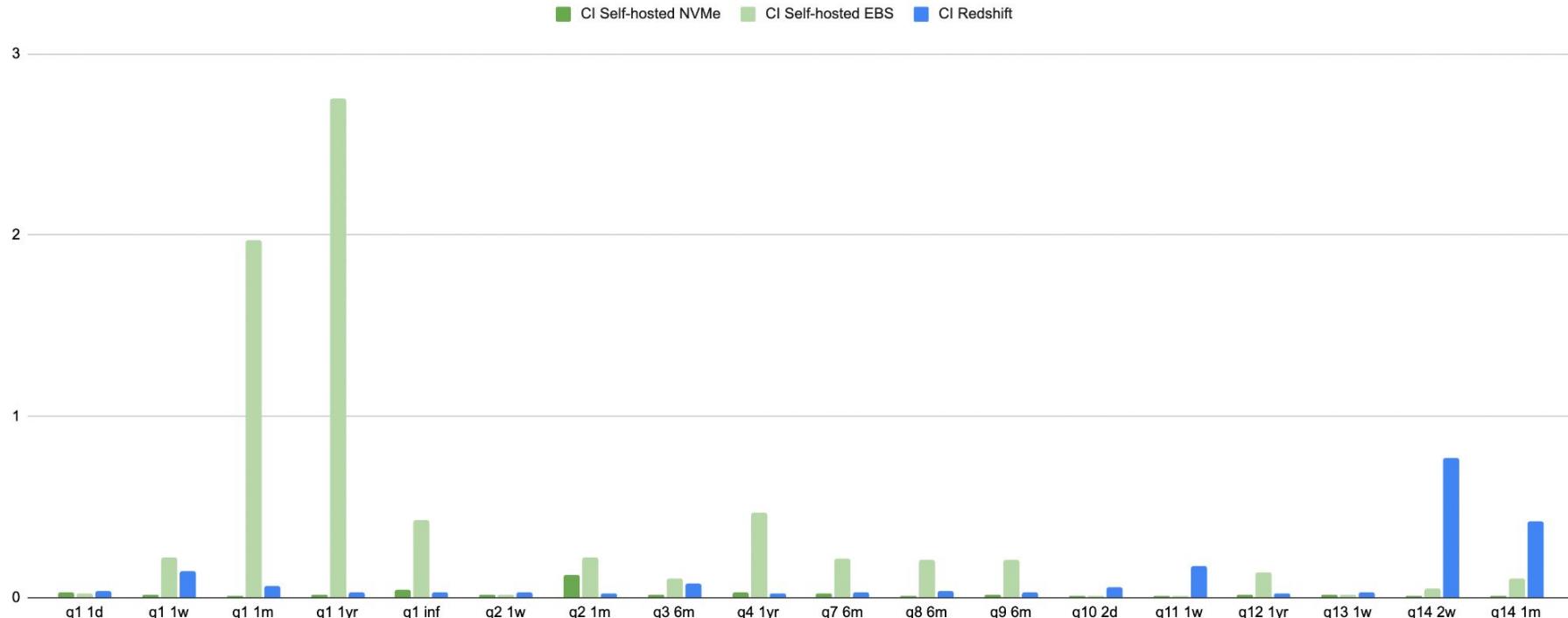


ClickHouse / clickhouse-presentations		
Public		
Code		Issues
Pull requests		Actions
Projects		Security
Insights		
master		8 branches
tags		
		Go to file
		Add file
		Code
 aleksey-milovidov		Add presentations from the 66th ClickHouse Meetup
		✓ aaf1a18f 22 hours ago
		479 commits
 BigDataLondon2022	Add files via upload	2 months ago
 bdtc_2019	Added a presentation from BDTC	3 years ago
 cern	Translate gifs in older presentations	3 months ago
 cpp_russia_2019	yandex/ClickHouse -> ClickHouse/ClickHouse, part 1	3 years ago
 cpp_russia_2020	Fix typos	3 years ago
 cpp_russia_2021_jit_in_clickhouse	Added presentation JIT in ClickHouse	13 months ago
 cpp_russia_2022_clickhouse_performace	Added presentation from C++ Russia 2022	6 months ago
 cpp_siberia_2021	Better markup	2 years ago
 cpp_sprint_2019	yandex/ClickHouse -> ClickHouse/ClickHouse, part 1	3 years ago
 cpp_user_group_2020	Added presentation from C++ UG 2020	2 years ago
 data_at_scale	Translate gifs in older presentations	3 months ago
 database_saturday_2018	Added presentation from Database Saturday	5 years ago
 database_saturday_2018_2	yandex/ClickHouse -> ClickHouse/ClickHouse, part 1	3 years ago
 database_saturday_2019	Added presentation from Database Saturday 2019	3 years ago
 database_saturday_2020	Added presentation from Database Saturday 2020	2 years ago
 dataops_2019	yandex/ClickHouse -> ClickHouse/ClickHouse, part 1	3 years ago
 drafts	yandex/ClickHouse -> ClickHouse/ClickHouse, part 2	3 years ago
 evolution	Translate gifs in older presentations	3 months ago
 github_planet_2021_02_16	Add presentation from GitHub Planet https://youtu.be/vdkdLojmkuhs...	2 years ago
 group_by	Changed tabs to spaces in code [#CLICKHOUSE-3]	6 years ago
 hash_tables	yandex/ClickHouse -> ClickHouse/ClickHouse, part 1	3 years ago
 heisenbug2021	Add Heisenbug 2021 presentation	2 years ago
 highload2016	Added historical presentation from HighLoad 2016	3 years ago
 highload2017	yandex/ClickHouse -> ClickHouse/ClickHouse, part 1	3 years ago
 highload2018	yandex/ClickHouse -> ClickHouse/ClickHouse, part 1	3 years ago
 highload2019	Added presentation from HighLoad++ 2019	3 years ago

Query time (seconds)

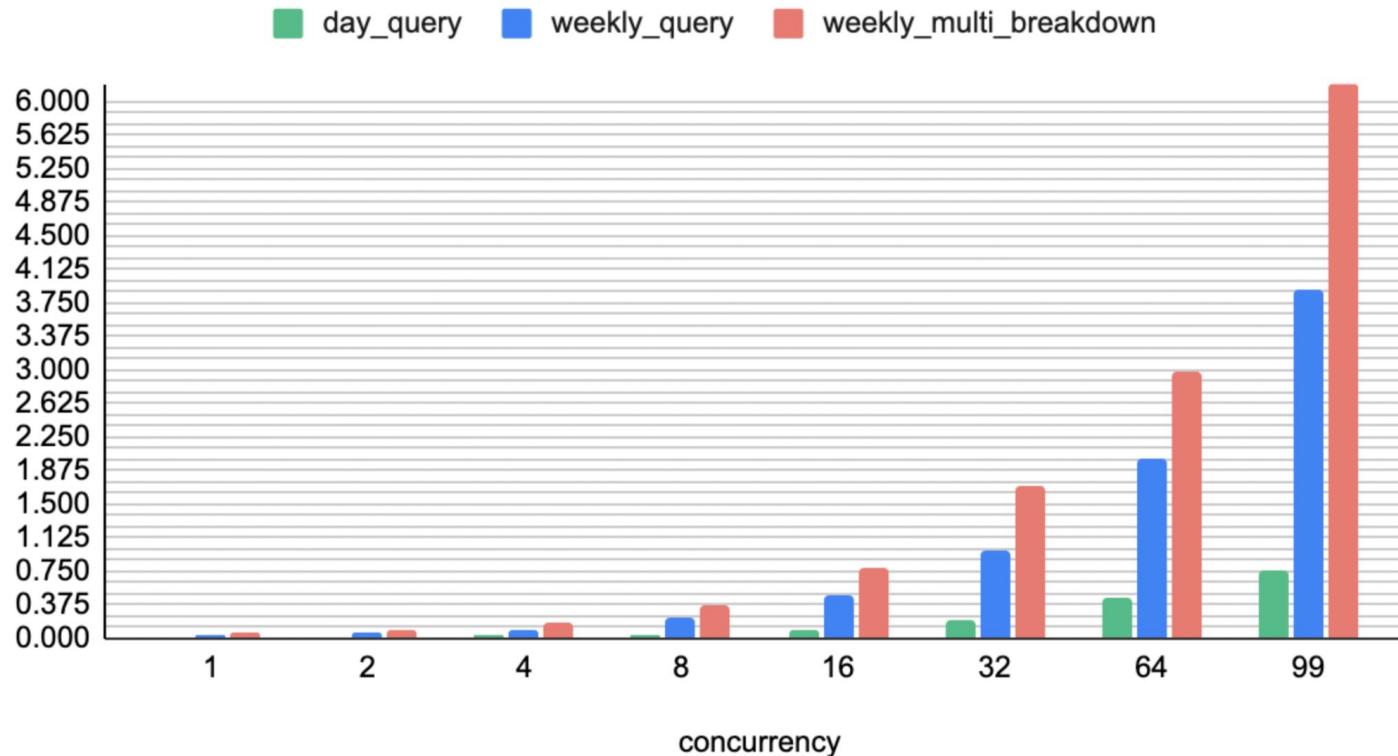


Confidence Interval (seconds)

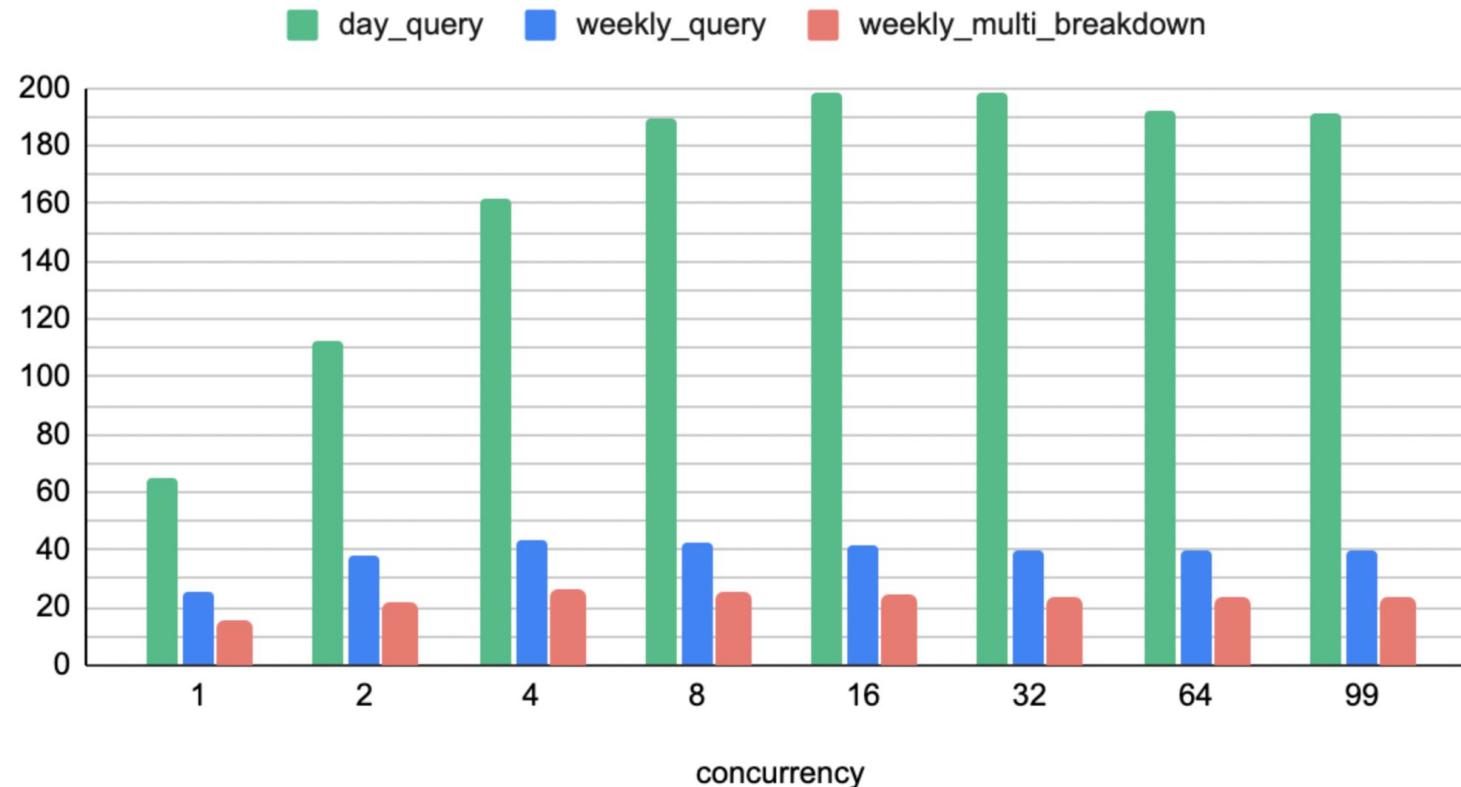


duration	1st run	2nd run	3rd run	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
q1 1d	0.139	0.078	0.077	0.132	0.077	0.084	0.133	0.069	0.071	0.135	0.076	0.096	0.157	0.083	0.08	0.091	0.097	0.07	0.09	0.076	0.076
q1 1w	0.405	0.393	0.398	0.416	0.396	0.386	1.531	0.397	0.4	0.396	0.399	0.402	0.422	0.399	0.391	0.382	0.393	0.384	0.394	0.39	0.387
q1 1m	1.78	1.766	1.77	1.793	1.808	1.84	11.708	1.766	1.77	1.781	1.786	1.811	1.765	1.803	1.952	1.765	1.751	1.764	1.762	1.754	1.758
q1 1yr	6.625	6.683	6.67	6.767	6.777	7.361	20.657	6.862	6.621	7.324	6.766	6.638	7.353	7.344	7.305	6.619	6.589	6.594	6.669	6.792	6.734
q1 inf	14.705	14.725	14.678	14.964	14.79	14.855	14.883	15.046	14.699	14.827	14.793	14.935	15.05	15.916	15.68	15.148	16.446	15.806	14.87	15.044	14.919
q2 1w	0.464	0.448	0.454	0.479	0.467	0.468	0.479	0.462	0.481	0.484	0.443	0.447	0.496	0.464	0.459	0.475	0.445	0.453	0.482	0.452	0.453
q2 1m	17.793	17.763	18.281	18.242	17.947	18.145	18.393	18.486	18.415	17.954	18.2	18.018	18.114	18.165	18.009	18.175	18.243	18.507	18.304	18.708	17.856
q3 6m	6.513	6.537	6.974	6.63	6.607	6.558	6.473	6.486	6.695	6.485	6.497	6.459	6.479	6.448	6.451	6.553	6.48	6.625	6.485	6.473	6.492
q4 1yr	16.337	15.965	15.427	14.896	14.575	14.612	14.625	14.544	14.504	14.507	14.495	14.715	14.483	14.489	14.771	14.646	14.828	15.476	14.652	14.565	14.534
q7 6m	4.867	4.788	4.754	4.982	4.693	4.608	4.595	4.58	4.577	4.611	4.599	4.595	4.662	4.61	4.614	5.272	5.243	5.202	4.605	4.565	4.559
q8 6m	6.135	6.084	6.059	5.964	6.031	6.101	5.953	6.701	6.726	5.968	5.96	5.94	6.032	5.913	5.911	6.309	6.129	6.12	5.92	5.908	5.924
q9 6m	6.518	6.517	6.502	6.732	6.717	6.604	7.349	7.008	6.661	6.504	6.485	6.886	6.492	6.466	6.481	6.577	6.955	6.767	6.482	6.471	6.476
q10 2d	0.22	0.198	0.207	0.226	0.213	0.207	0.225	0.216	0.219	0.219	0.228	0.212	0.225	0.202	0.211	0.241	0.217	0.207	0.214	0.213	0.228
q11 1w	0.687	0.691	0.692	0.691	0.691	0.687	0.681	0.691	0.686	0.705	0.708	0.699	0.711	0.695	0.694	0.708	0.696	0.709	0.696	0.686	0.686
q12 1yr	9.077	9.102	9.102	9.612	9.321	9.31	9.088	9.051	9.098	9.071	9.461	9.19	9.083	9.059	9.021	9.286	9.07	9.045	9.088	9.021	9.103
q13 1w	0.663	0.654	0.671	0.66	0.698	0.669	0.711	0.682	0.691	0.698	0.685	0.682	0.69	0.68	0.672	0.702	0.673	0.67	0.705	0.691	0.672
q14 2w	1.972	1.977	1.95	2.03	2.138	2.059	1.968	1.957	1.98	1.967	2.071	2.128	1.974	1.961	1.98	1.998	1.963	1.973	1.959	1.971	1.936
q14 1m	4.342	4.336	4.341	4.603	4.601	4.534	4.461	4.449	4.373	4.38	4.57	4.596	4.328	4.346	4.316	4.406	4.336	4.331	4.695	4.374	4.393

Latency(p90)[seconds] vs. concurrency



QPS vs concurrency



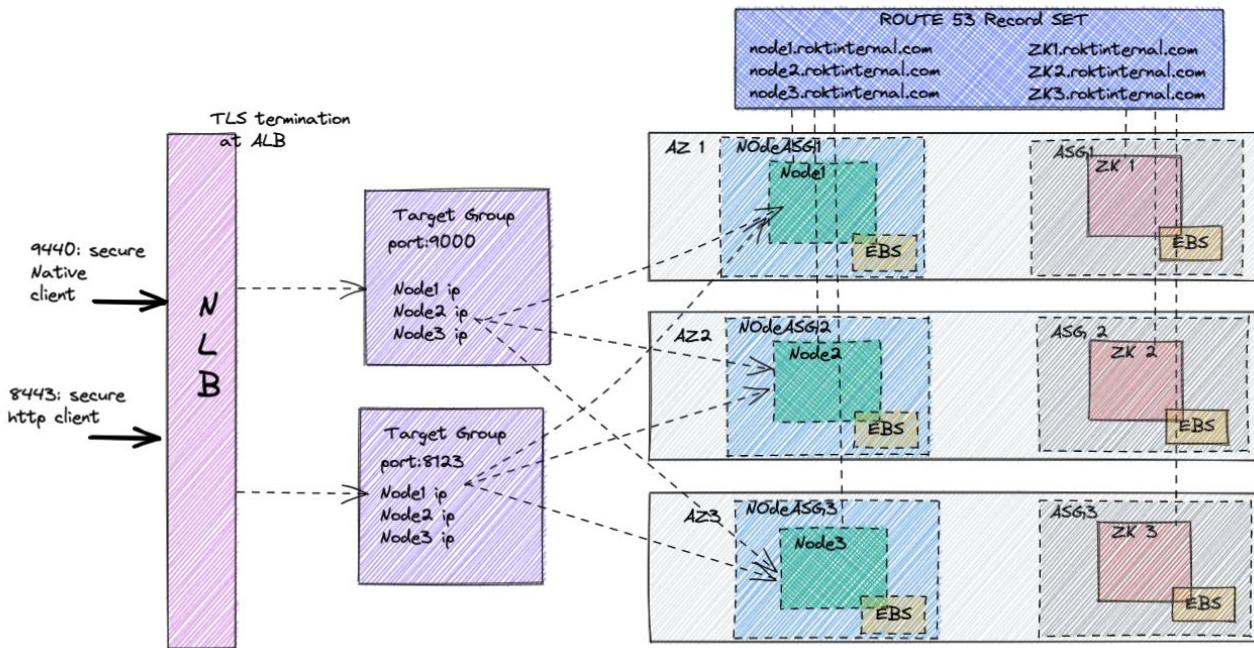
S3 size of test data (disjoint): 484GB, parquet, gzip

```
SELECT
    table,
    formatReadableSize(sum(bytes)) as size,
    min(min_date) as min_date,
    max(max_date) as max_date
FROM system.parts
WHERE active
GROUP BY table
```

~450GB for ~6.5B rows and we didn't use any compression methods (~70bytes per record),

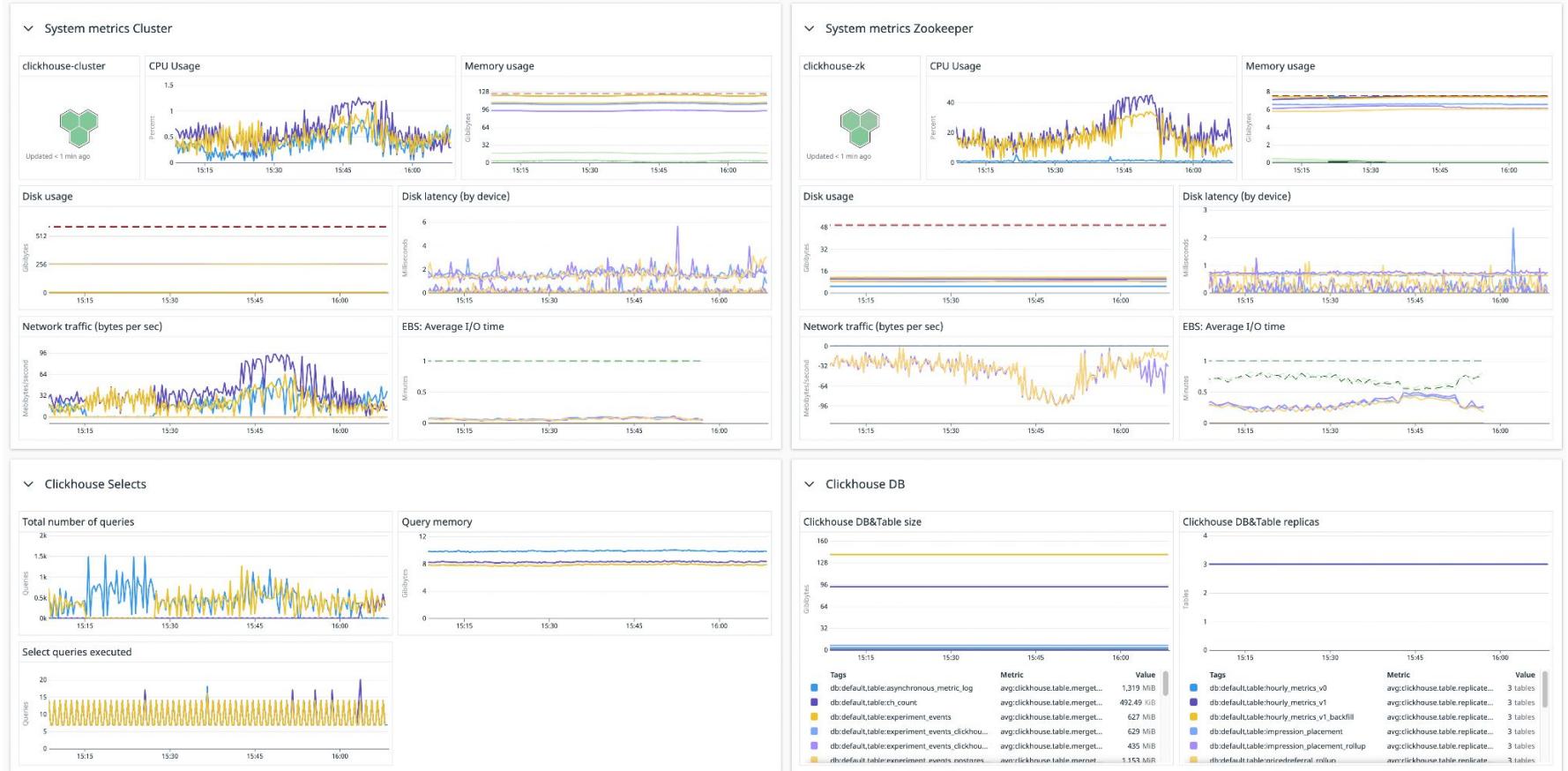
, in Elasticsearch an index for 325,087,446 documents uses ~400GB across 3 replicas, so ~133GB on one machine.

Which means ~409bytes per record or 5.8x more than in Clickhouse!



3x c6i.16xlarge (64, 128GB, 25Gbps)
3x m5.large (2, 8GB)
6x gp3 600GB 125MB/s 7000IOPS

$3 \times (\$2.72 + \$0.096) * 24 * 30 + 6 \times \$102 \approx \$6700/\text{month}$

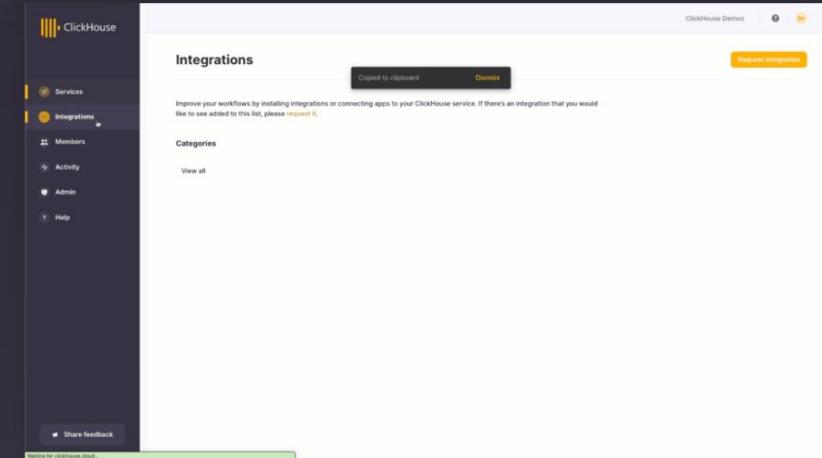


Complexities:

- Zookeeper
- Replication
- Sharding
- Upgrades
- Resizing
- Backups

More insights. Less operations.

Introducing ClickHouse as a service, built by the creators and maintainers of the fastest OLAP database on earth.

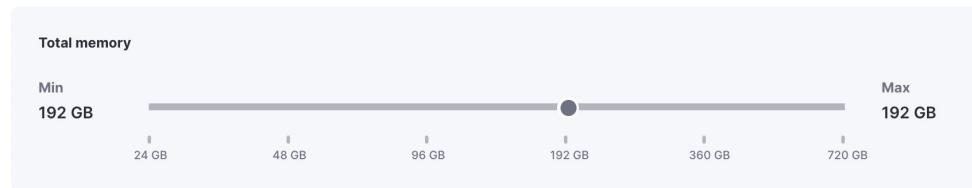
[Start free trial](#)[Cloud providers](#)[Coming soon](#)

Advanced scaling

X

By default, ClickHouse Cloud automatically scales compute capacity to accommodate the needs of your workload. The default behaviour is to scale down your compute instances after a period of inactivity and vertically scale available memory until it reaches the maximum vertical limit.

In situations where you would like to reserve capacity or set a lower vertical scaling threshold, you can change the default settings below.



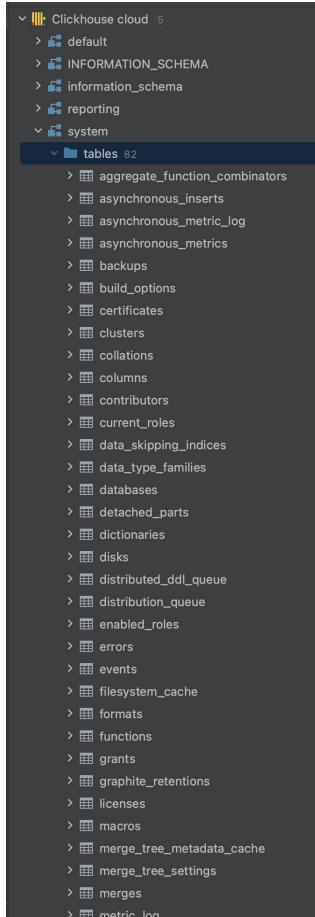
Pause service after a period of inactivity

Pausing services when inactive saves on costs. Services usually take 20-30 seconds to restart after pausing. [Read more](#)

Close

Reset to default

Update advanced scaling



Access to system-wide tables

Query details: 156492851

⟳ Terminate query Copy link Open in query editor

Rewritten queries (1)

Amazon Redshift rewrote this query to optimize it.

Start time	▲ Query	▼ Status	▼ Duration	▼ Executed on	▼ Query type
Nov 29th, 2022 07:23:25 AM 3 hours ago	722811974165169.1682142	Running...	3 sec		Parent query

Connecting to Kafka directly

[rokt-clickhouse / DDL / anomaly_detection / prod / kafka_transaction.sql](#)

70 lines (70 loc) · 2.98 KB

```
1  DROP VIEW IF EXISTS anomaly_detection.v_transaction ON CLUSTER reporting_cluster SYNC;
2  DROP TABLE IF EXISTS anomaly_detection.transaction_rollup ON CLUSTER reporting_cluster SYNC;
3  DROP TABLE IF EXISTS anomaly_detection.kafka_transaction ON CLUSTER reporting_cluster SYNC;
4  CREATE TABLE IF NOT EXISTS anomaly_detection.kafka_transaction ON CLUSTER reporting_cluster
5  (
6      id String,
7      type String,
8      timestamp UInt64,
9      source String,
10     target String,
11     i Nullable(String),
12     Nullable(String),
13     Nullable(String),
14     Nullable(String),
15     Nullable(String),
16     Nullable(String),
17     Nullable(String),
18     Nullable(Boolean),
19     Nullable(UInt64),
20     Nullable(String),
21     Nullable(String),
22     Nullable(String),
23     Nullable(String)
24 ) ENGINE = Kafka() SETTINGS
25     kafka_broker_list = 'kafka-bootstrap-int',
26     kafka_topic_list = 'rdn.enriched.Transac',
27     kafka_group_name = 'admc-clickhouse',
28     kafka_format = 'AvroConfluent',
29     kafka_num_consumers = 1,
30     kafka_thread_per_consumer = 2,
31     format_avro_schema_registry_url = 'https
32 ;
```

```

52 ;
53 CREATE TABLE IF NOT EXISTS anomaly_detection.transaction_rollup ON CLUSTER reporting_cluster (
54     eventTime          DateTime('UTC')      comment 'dimension',
55     partnerId         Nullable(String)    comment 'dimension',
56     pageId            Nullable(String)    comment 'dimension',
57     placementId       Nullable(String)    comment 'dimension',
58     pageInstanceId    Nullable(String)    comment 'dimension',
59     pageType          Nullable(String)    comment 'dimension',
60     instanceGuid      String           comment 'dimension',
61     quantity           UInt32          comment 'metric'
62 ) ENGINE = ReplicatedSummingMergeTree()
63         PRIMARY KEY (eventTime,partnerId)
64         ORDER BY (eventTime, partnerId, pageId, placementId, pageInstanceId, pageType, instanceGuid)
65         PARTITION BY toYYYYMM(eventTime)
66         TTL eventTime + INTERVAL 6 MONTH
67 SETTINGS
68     allow_nullable_key = 1
69 ;
70 CREATE MATERIALIZED VIEW anomaly_detection.v_transaction TO anomaly_detection.transaction_rollup AS
71 SELECT
72     toStartOfMinute(fromUnixTimestamp(intDiv(eventTime, 1000))) AS eventTime,
73     partnerid AS partnerId,
74     pageid AS pageId,
75     placementid AS placementId,
76     pageInstanceId AS pageInstanceId,
77     pagetype AS pageType,
78     instanceGuid AS instanceGuid,
79     count() as quantity
80 FROM
81     anomaly_detection.kafka_transaction
82 GROUP BY
83     eventTime,
84     partnerId,
85     pageId,
86     placementId,
87     pageInstanceId,
88     pageType,
89     instanceGuid
90 ;

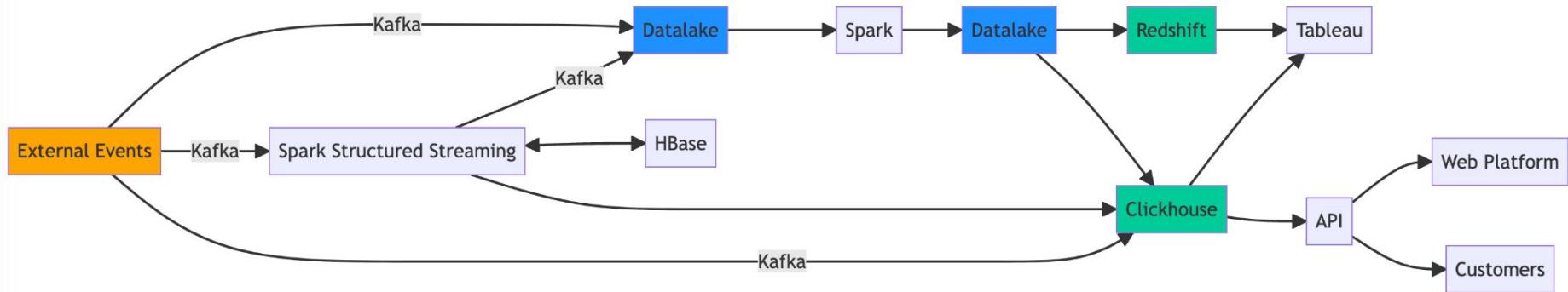
```

No JOINS in ES == the API has to do them, get data from ES, get IDs for each dimension, get labels from the cache or by querying other APIs, then join back

Clickhouse:

```
CREATE VIEW reporting.jdbc_dict_source_account_v0 AS
SELECT accountid, name as "name", verticalid, subverticalid
FROM jdbc('redshift-server','select accountid, name, verticalid, subverticalid from public.account')

CREATE DICTIONARY reporting.jdbc_dict_account_v0
(
    `accountid` UInt64,
    `name` String,
    `verticalid` String,
    `subverticalid` String
)
PRIMARY KEY accountid
SOURCE(CLICKHOUSE(TABLE 'jdbc_dict_source_account_v0'))
LIFETIME(MIN 300 MAX 6000)
LAYOUT(complex_key_hashed());
```



ROKT

ROKT IS HIRING!

vadim.semenov@rokt.com