Real-Time Analytics on Data Lakes: Indexing Amazon S3 for up to 125x Faster Queries

Dhruba Borthakur / CTO, Rockset

Nadine Hachouche / Senior Developer Advocate, Rockset

Presented at Data Riders Meetup 04/2021

Slack Community: bit.ly/rockset-community-channel

ROCKSET

Presenter

Dhruba Borthakur
Co-Founder & CTO

Nadine Hachouche
Senior Developer Advocate

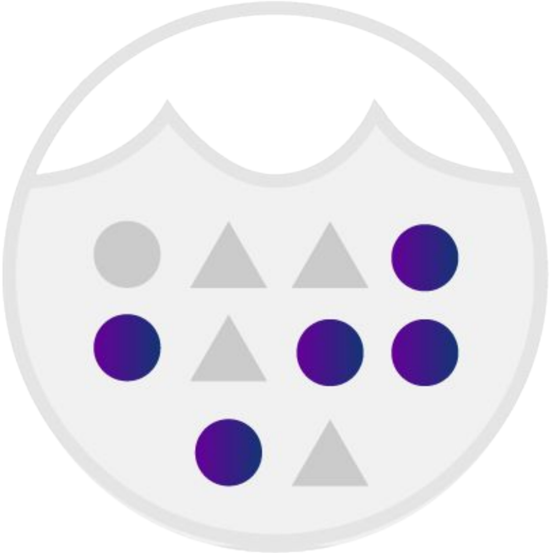
Agenda

1. The state of cloud data lakes
2. Rockset for real-time analytics
3. Considerations for apps on data lakes
   a. Indexing vs scanning
   b. High concurrency
   c. Mutability of data and schema
4. Developer productivity
5. Workshop

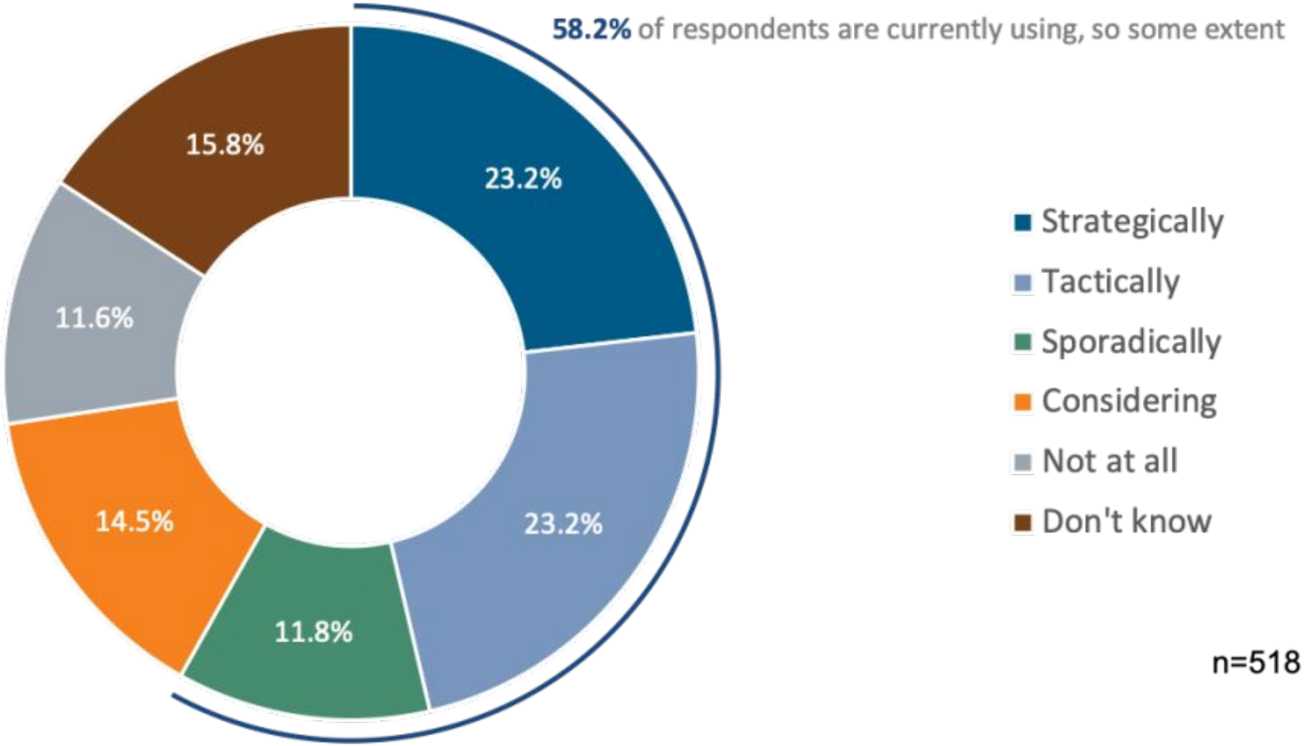
The state of cloud data lakes

Data lake adoption continues to rise

- Store raw data as is
- Bring multiple types of data together
- Scale to massive volumes of data cost effectively
- Run many different kinds of analytics
- Democratize access to data



Most enterprises currently use a data lake



ROCKSET

Source: 451 Research (2019)

Cloud data lakes outpacing on-prem

Is your organization planning to augment or replace its existing BI, analytics, and data warehousing systems with any of the following systems or cloud-based services, solely or in combination? (Please select all that apply.)

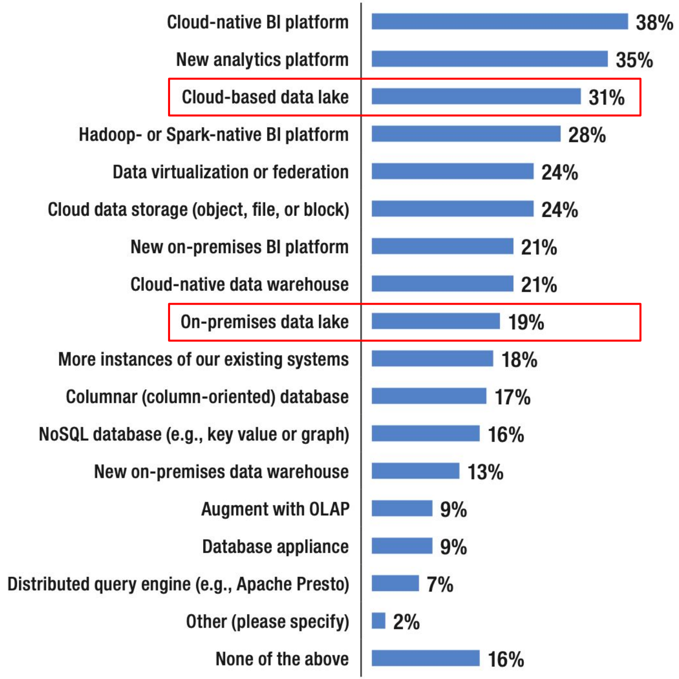


Figure 4. *Based on answers from 232 respondents.*

Source: TDWI (2018)

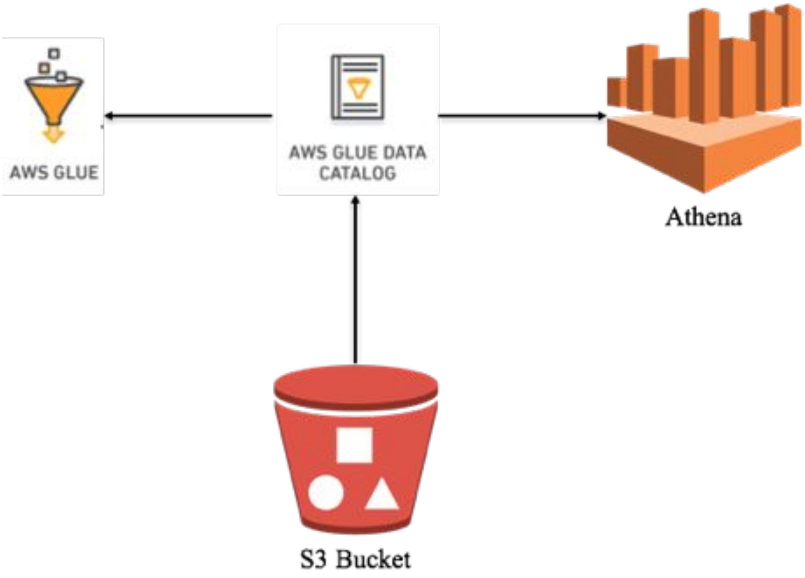
ROCKSET

Amazon S3 is the leading cloud data lake option

- Reliable
- Practically infinite scalability
- Compatible with many other services
- Tens of thousands of data lakes on S3

Athena commonly used for ad-hoc queries on S3

- Standard SQL
- Based on Presto
- Serverless
- Ideal for ad-hoc queries on S3 data lakes



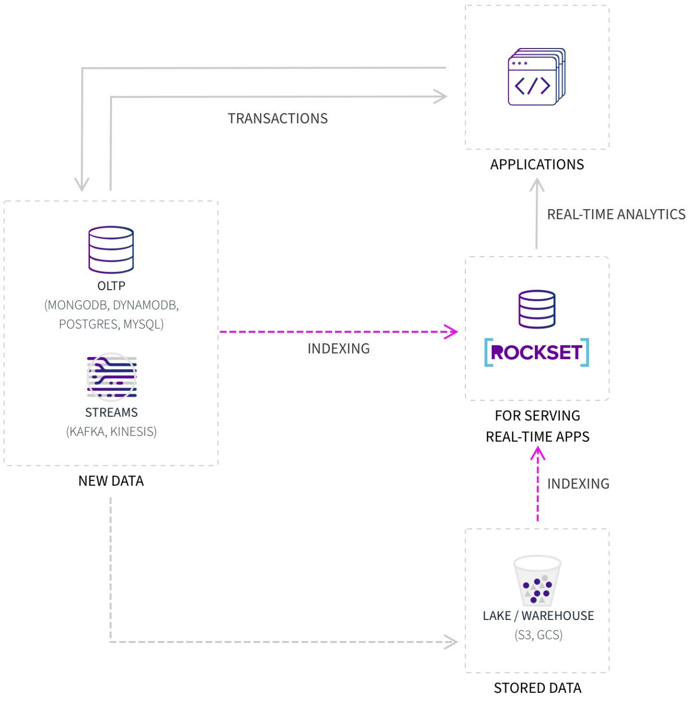
But what about building apps on data in S3?

- Occasional queries → continuous, highly concurrent queries
- Growing need for apps that activate data in S3
  - A/B experiments on behavioral data
  - Personalization and customer 360 on marketing data
  - IoT apps on sensor data
- Need a real-time analytics solution built for low-latency, high-concurrency queries

Rockset for real-time analytics

Real-time analytics on data lakes with Rockset

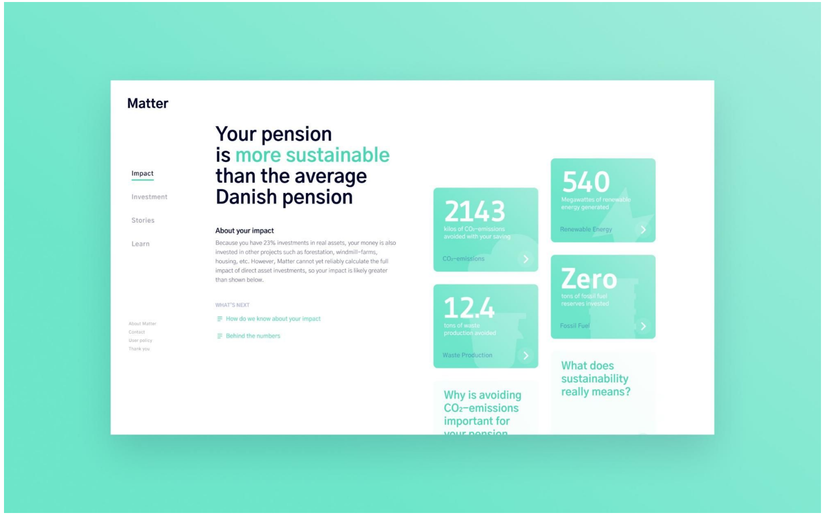
- Rockset: real-time indexing database
- Built-in connector to S3
- Continuous, schemaless ingest
- Purpose built to serve apps



ROCKSET

Matter delivers AI-powered insights for sustainable investing

- NLP pipeline on newsfeeds stores results in S3 data lake
- High-volume queries for 1000s of asset positions in investment portfolios
- Hours ⟶ sub-second query latency moving from Athena to Rockset



Your pension is more sustainable than the average Danish pension

About your impact

Because you have 23% investments in real assets, your money is also invested in other projects such as forestation, windmill-farms, housing, etc. However, Matter cannot yet reliably calculate the full impact of direct asset investments, so your impact is likely greater than shown below.

WHAT'S NEXT

How do we know about your impact

Behind the numbers

2143 kilos of CO_2-emissions avoided with your saving — CO_2-emissions

540 Megawattes of renewable energy generated — Renewable Energy

12.4 tons of waste production avoided — Waste Production

Zero tons of fossil fuel reserves invested — Fossil Fuel

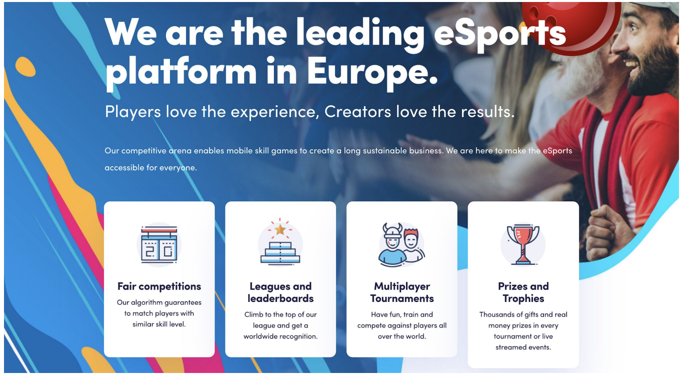
Why is avoiding CO_2-emissions important for your pension

What does sustainability really means?

https://rockset.com/blog/matter-uses-rockset-to-bring-ai-powered-sustainable-insights-to-investors/

eGoGames: esports platform for mobile games

- Understand what users are doing in real time
- Analyze user acquisition and retention data in S3 with transactional data in DynamoDB
- Takes too long to centralize data in S3 to query with Athena
- Rockset allows eGoGames to query across data sources within seconds of data being produced



We are the leading eSports platform in Europe.

Players love the experience, Creators love the results.

Our competitive arena enables mobile skill games to create a long sustainable business. We are here to make the eSports accessible for everyone.

Fair competitions
Our algorithm guarantees to match players with similar skill level.

Leagues and leaderboards
Climb to the top of our league and get a worldwide recognition.

Multiplayer Tournaments
Have fun, train and compete against players all over the world.

Prizes and Trophies
Thousands of gifts and real money prizes in every tournament or live streamed events.

Ponemos los eSports al alcance de todos.

https://rockset.com/blog/egogames-rockset-real-time-analytics-on-gaming-data/

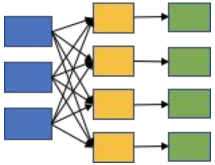
Considerations for apps on data lakes

Query Latency

Athena	Rockset
Designed for ad-hoc queries	Designed for real-time applications ● high selectivity on large data sets ● continuous queries
Stores data in columnar format	Stores data in ● rowstore ● columnstore ● inverted index Data sorted by *_event_time* for fast time-range queries
Every query is parallelize and scan	Queries are served from indexes ● up to 125x faster than Athena

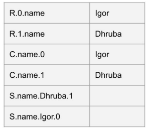
Optimize Query Latency by Converged Indexing

- Traditional approach: Parallelize and scan



event data → MapReduce → reports

- Real-time event analytics: Parallelize and index

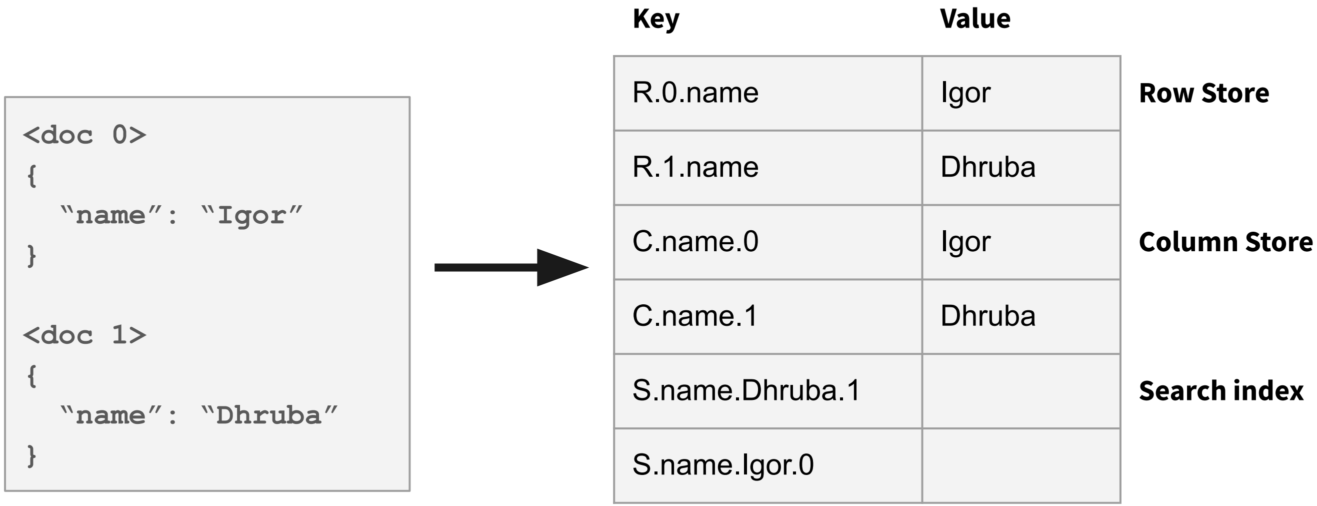


R.0.name	Igor
R.1.name	Dhruba
C.name.0	Igor
C.name.1	Dhruba
S.name.Dhruba.1	
S.name.Igor.0	

event data → Converged Indexing → applications and APIs

Converged Indexing

- Columnar and search indexes in the same system
- Built on top of key-value store abstraction
- Each document maps to many key-value pairs



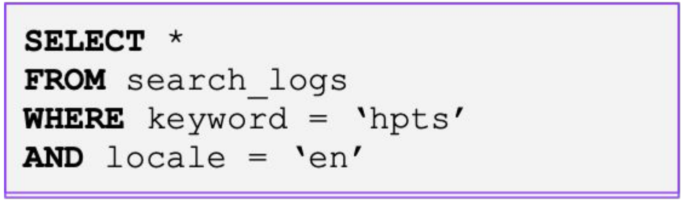
```
<doc 0>
{
  "name": "Igor"
}

<doc 1>
{
  "name": "Dhruba"
}
```

Key	Value	
R.0.name	Igor	**Row Store**
R.1.name	Dhruba	
C.name.0	Igor	**Column Store**
C.name.1	Dhruba	
S.name.Dhruba.1		**Search index**
S.name.Igor.0		

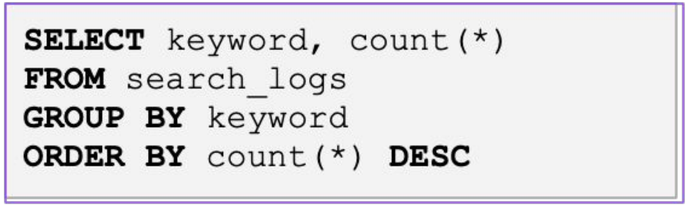
Query Optimizer

- Low latency for both highly selective queries and large scans
- Optimizer picks between
  - inverted index (Index Filter operator)
  - columnar format (Column Scan operator)
  - inverted index (Index Scan operator)



```
SELECT *
FROM search_logs
WHERE keyword = ‘hpts’
AND locale = ‘en’
```

Inverted index
(for highly selective queries)



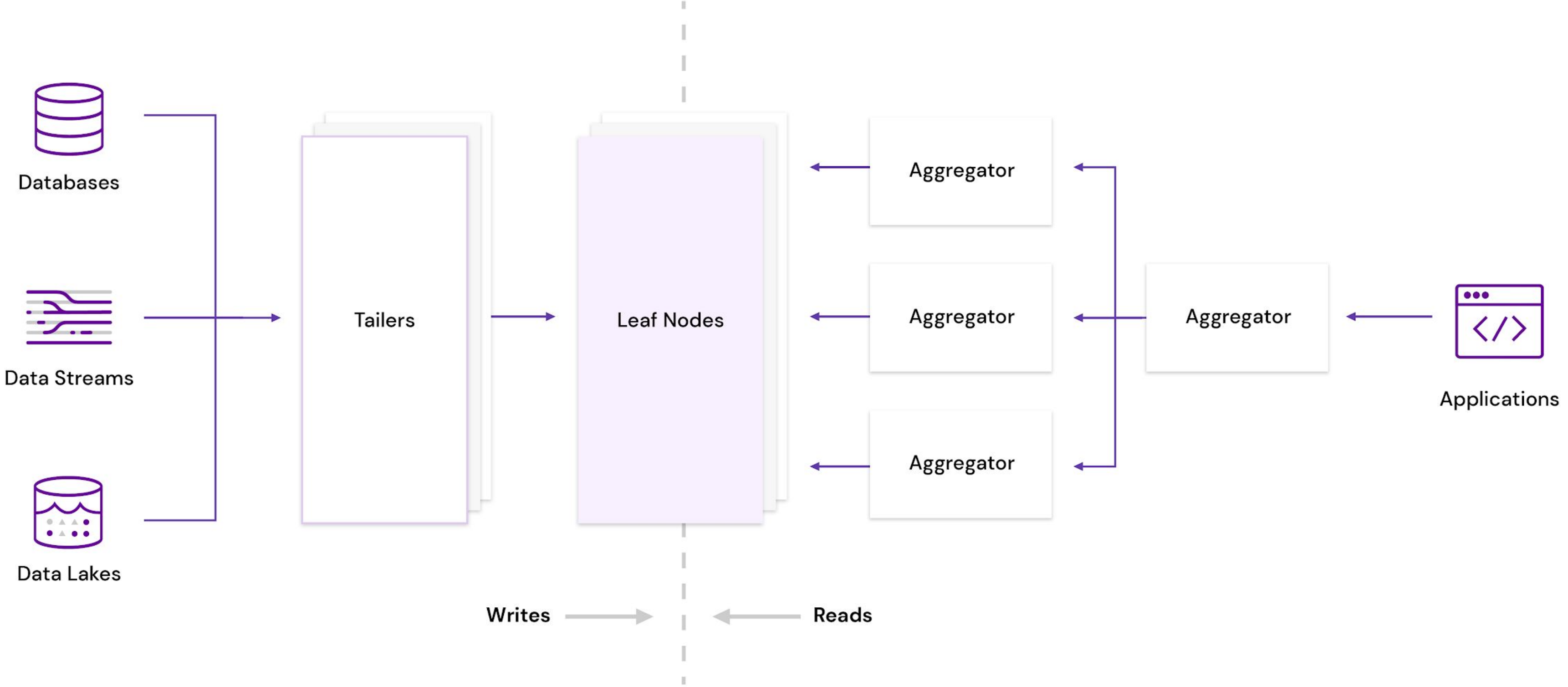
```
SELECT keyword, count(*)
FROM search_logs
GROUP BY keyword
ORDER BY count(*) DESC
```

Columnar store
(for large scans)

Concurrency

Athena	Rockset
Used for low-concurrency use cases ● ad-hoc, interactive queries ● data science ● BI	Used to power high-concurrency use cases ● real-time analytic applications ● large numbers of users ● spiky usage
Executes 5 concurrent queries ● queues any additional queries	Supports 1000s of QPS

Rockset Uses an ALT Architecture

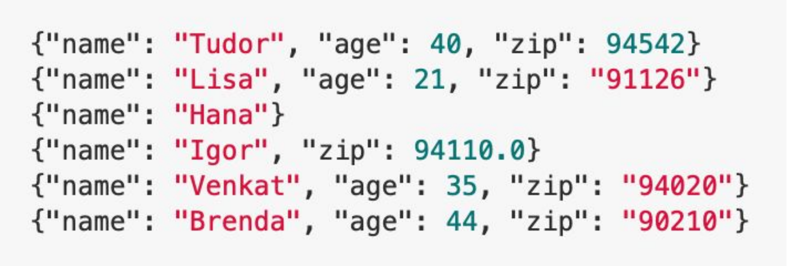


Mutability of data and schema

Athena	Rockset
Requires table creation with a schema	Automatically generates schema based on exact fields and types in the data
Schema changes result in delays in querying data	Optimized for data latency ● raw data is immediately queryable without requiring a schema
Only supports inserts but not updates	All documents are mutable

Strong Dynamic Typing

- Fields are dynamically typed

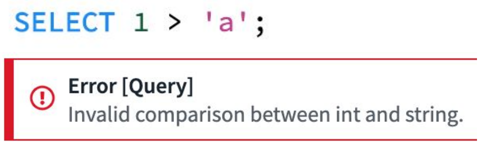


```
{"name": "Tudor", "age": 40, "zip": 94542}
{"name": "Lisa", "age": 21, "zip": "91126"}
{"name": "Hana"}
{"name": "Igor", "zip": 94110.0}
{"name": "Venkat", "age": 35, "zip": "94020"}
{"name": "Brenda", "age": 44, "zip": "90210"}
```

Strong Dynamic Typing

- Fields are dynamically typed

- Queries are strongly typed



```
SELECT 1 > 'a';
```

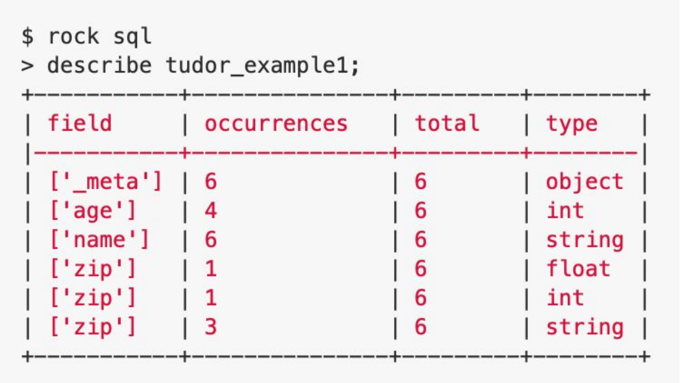
Error [Query]
Invalid comparison between int and string.

Strong Dynamic Typing

- Fields are dynamically typed

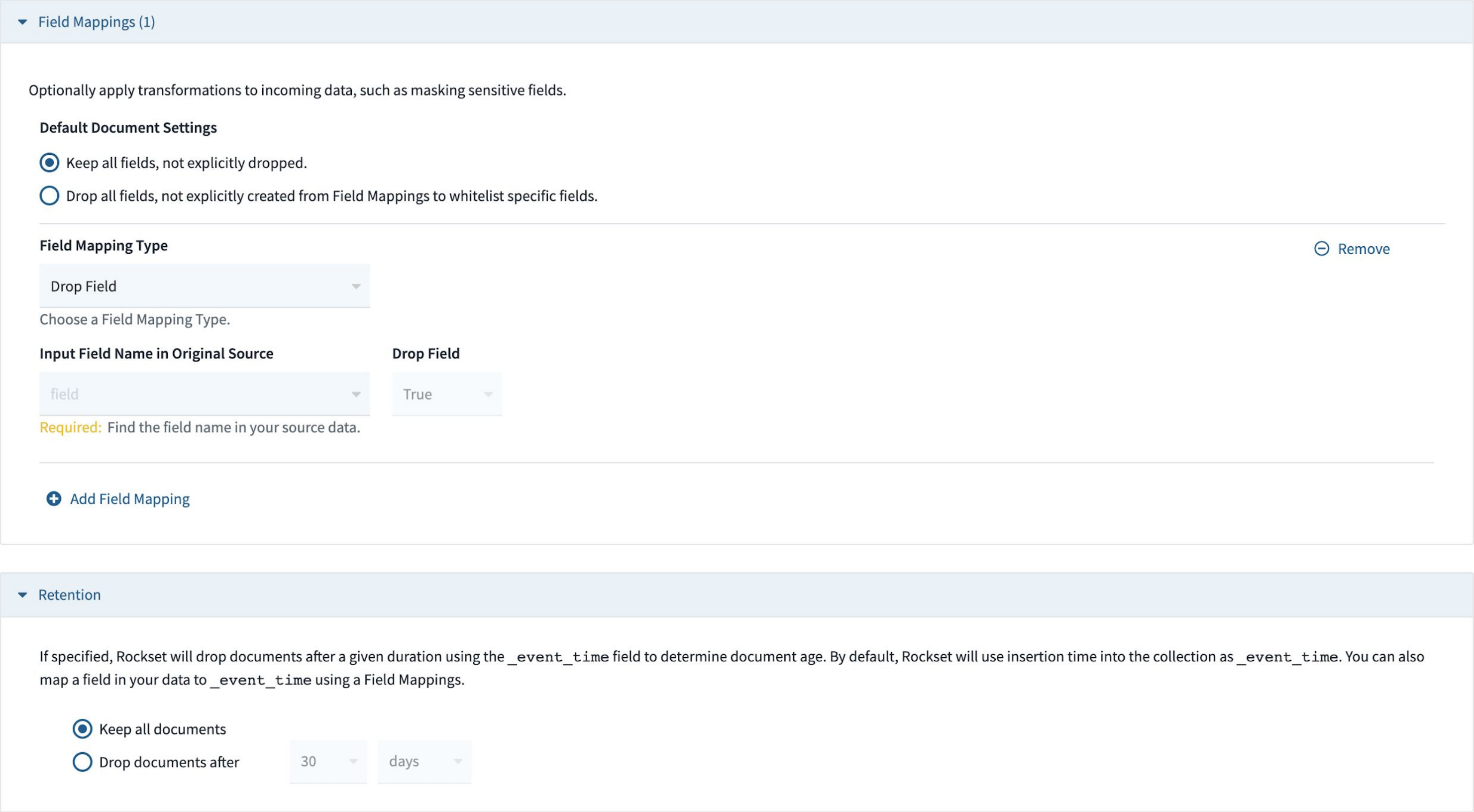
- Queries are strongly typed

- Smart schemas



```
$ rock sql
> describe tudor_example1;
+-----------+---------------+---------+--------+
| field     | occurrences   | total   | type   |
|-----------+---------------+---------+--------|
| ['_meta'] | 6             | 6       | object |
| ['age']   | 4             | 6       | int    |
| ['name']  | 6             | 6       | string |
| ['zip']   | 1             | 6       | float  |
| ['zip']   | 1             | 6       | int    |
| ['zip']   | 3             | 6       | string |
+-----------+---------------+---------+--------+
```

Set field mappings and retention as needed



▾ Field Mappings (1)

Optionally apply transformations to incoming data, such as masking sensitive fields.

Default Document Settings

● Keep all fields, not explicitly dropped.

○ Drop all fields, not explicitly created from Field Mappings to whitelist specific fields.

Field Mapping Type

⊖ Remove

Drop Field

Choose a Field Mapping Type.

Input Field Name in Original Source

field

Required: Find the field name in your source data.

Drop Field

True

⊕ Add Field Mapping

▾ Retention

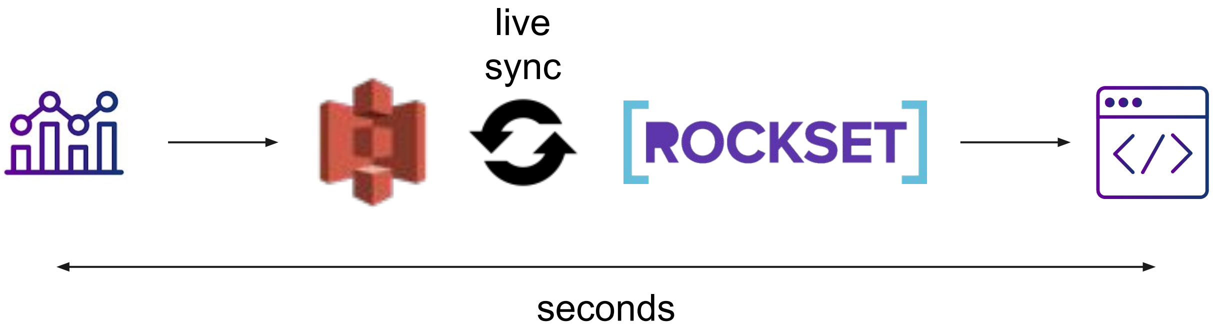
If specified, Rockset will drop documents after a given duration using the `_event_time` field to determine document age. By default, Rockset will use insertion time into the collection as `_event_time`. You can also map a field in your data to `_event_time` using a Field Mappings.

● Keep all documents

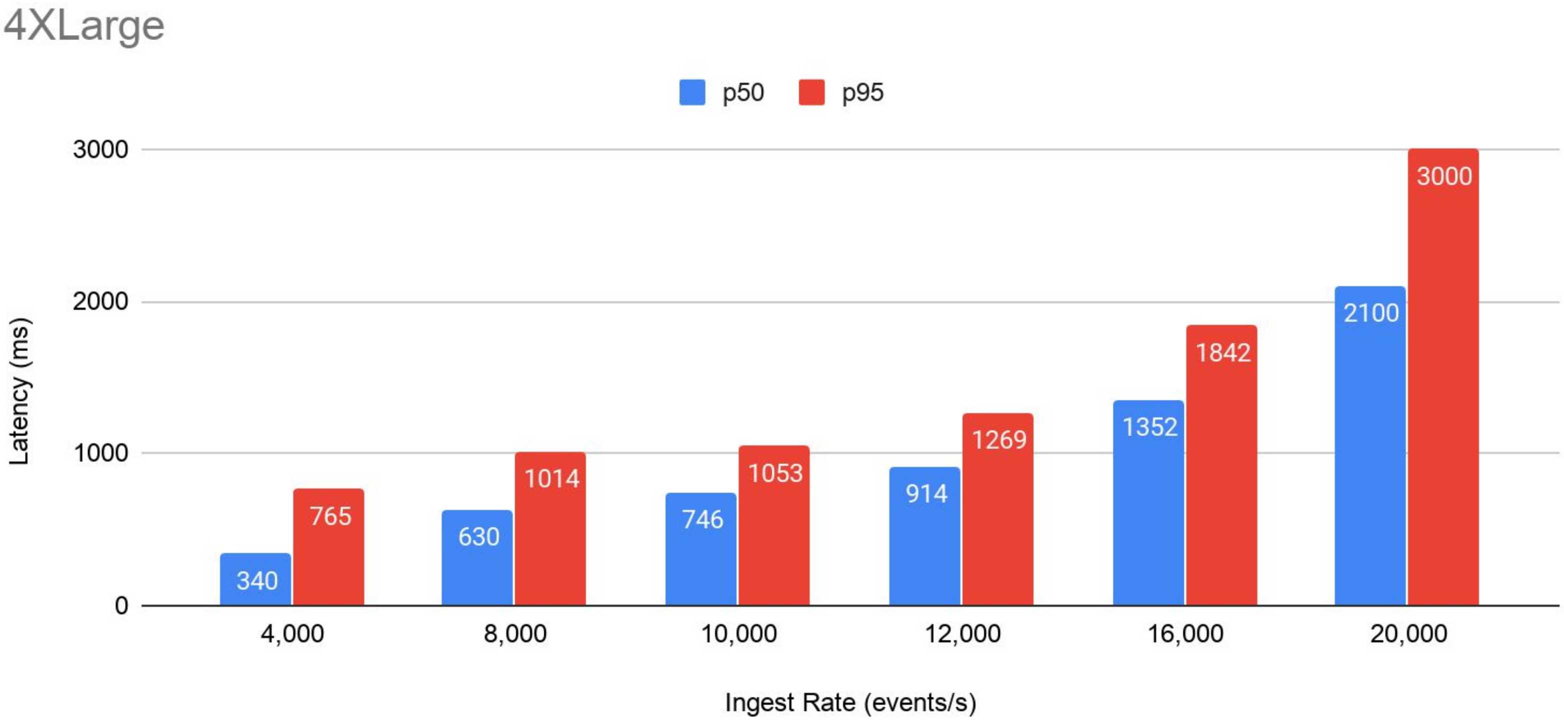
○ Drop documents after 30 days

Data Latency

- Fast ingestion
    - New data is visible in query results within a minute or less
    - Continuous Live sync of new data from S3 to Rockset



1-sec data latency when writing 1B documents/day

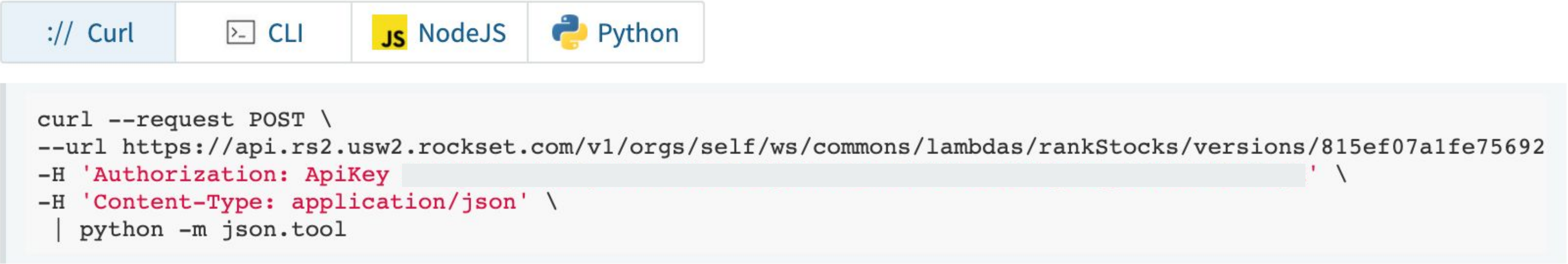


ROCKSET

Developer productivity

Query Lambdas

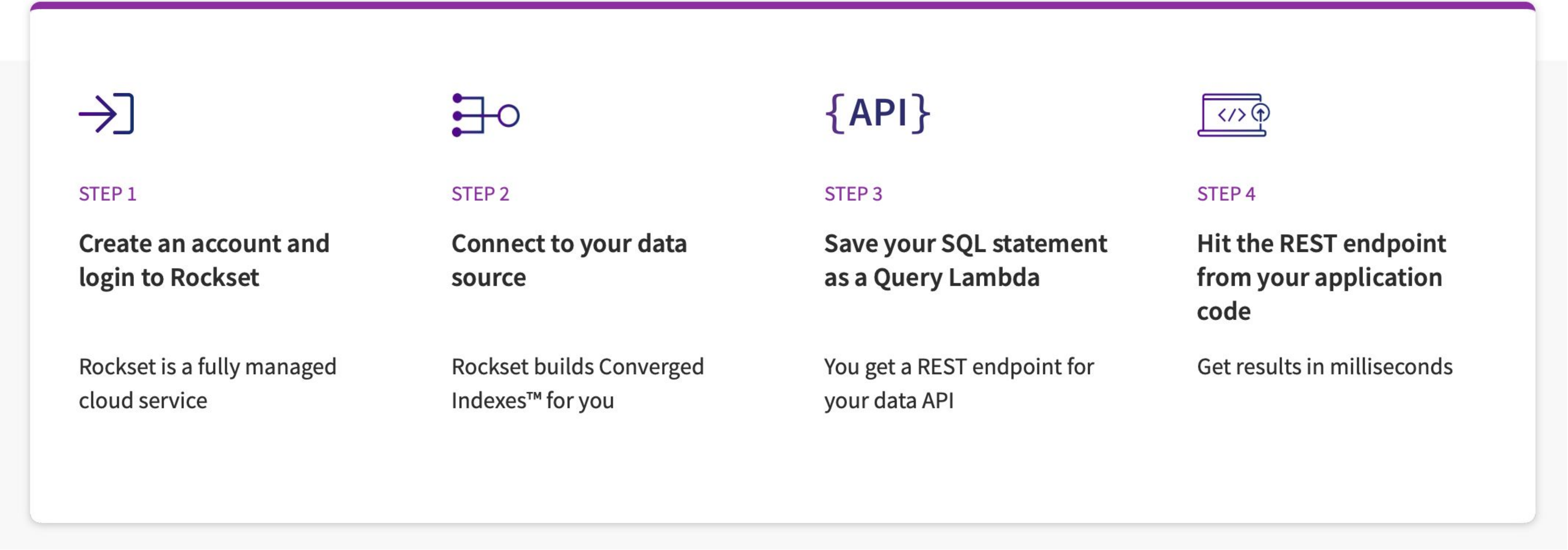
- Named, parameterized SQL queries stored in Rockset
- Executed from dedicated REST endpoint
- Organize by versions and tags
- Create data APIs used by multiple application developers
- Avoid having SQL in application code



```
curl --request POST \
--url https://api.rs2.usw2.rockset.com/v1/orgs/self/ws/commons/lambdas/rankStocks/versions/815ef07a1fe75692
-H 'Authorization: ApiKey [redacted]' \
-H 'Content-Type: application/json' \
  | python -m json.tool
```

Workshop

Using Rockset to Build Real-time Analytics



STEP 1

Create an account and login to Rockset

Rockset is a fully managed cloud service

STEP 2

Connect to your data source

Rockset builds Converged Indexes™ for you

STEP 3

Save your SQL statement as a Query Lambda

You get a REST endpoint for your data API

STEP 4

Hit the REST endpoint from your application code

Get results in milliseconds

Indexing S3 Using Rockset

- Low-latency, high-concurrency queries for serving applications
- Increase developer velocity and reduce time to market
- Complete solution for real-time analytics on data lakes

> “While you can build or buy solutions individually that might provide more ingest options, better absolute performance, lower marginal costs or higher scalability potential, I have yet to find anything that comes remotely close to Rockset on all of these areas at the same time, in a setting where time to market is a highly valuable metric.”
>
> - Matter CTO Alex Harrington

Get started with **$300** in free credits

https://console.rockset.com/create

Thank you

Dhruba Borthakur / **dhruba@rockset.com**

Join the community channel after the talk at
bit.ly/rockset-community-channel