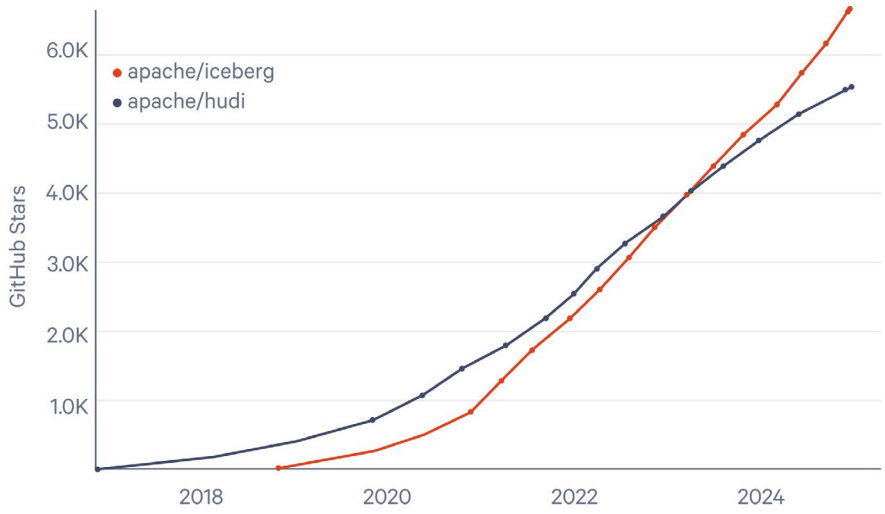
Redpanda

Data Engineering Essentials

Streaming Data into Apache Iceberg™

Introduction

Open-source Apache Iceberg™ is a high-performance, open table format built for large, complex analytic datasets. Due to its performance and having broad support for the standard within the analytics technology industry, Iceberg has quickly become a de facto standard for collecting and analyzing data at scale.



Iceberg is popular with developers, surpassing alternatives like Apache Hudi in GitHub stars

However, streaming data into Iceberg traditionally requires a significant upfront and ongoing data engineering effort.

In this ebook, data engineers will learn why Iceberg makes such a good platform for capturing streaming analytics data, and also learn a new and easier way to stream data directly into Iceberg via Redpanda Iceberg Topics.

By using Redpanda Iceberg Topics, organizations can integrate streaming data into their Iceberg ecosystem and accelerate real-time business innovation, using a small fraction of the data engineering resources traditionally required.

What is Apache Iceberg?

Iceberg is a high-performance, open table format built for large, complex analytic datasets. Originally developed by Netflix to solve data lake scalability limitations, Iceberg defines how datasets are organized, stored, updated, and queried in a distributed storage system like Amazon S3.

Companies like Apple, LinkedIn, Salesforce, and JP Morgan Chase use Iceberg to simplify working with petabytes of analytics data. As data lands in Iceberg tables, it's immediately accessible to various popular analytics engines, such as Apache Spark™, Snowflake, Dremio, Amazon Redshift, and Apache Flink®.

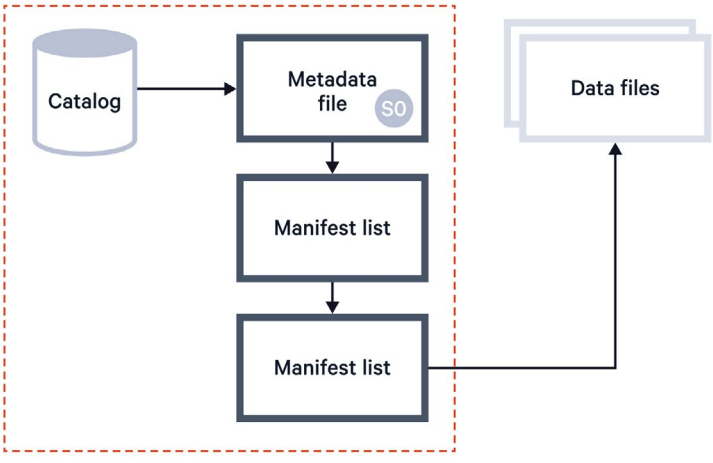
Here's a high-level view of Iceberg's capabilities:

- **Data scalability -** Iceberg is designed to handle petabyte-scale datasets.
- **ACID transactions -** Iceberg keeps data accurate and prevents corruption, even while multiple clients read and update data simultaneously
- **Data processing versatility -** Teams can use preferred tools (SQL, Apache Spark, Trino, Apache Flink, et al) to access the same underlying dataset.
- **Schema evolution -** Tables can be redefined by adding, dropping, and modifying columns without requiring downtime or data rewrites.
- **Time-travel -** Iceberg maintains a versioned history of data changes so you can query data as it was at a specific time. This is useful for auditing and troubleshooting.
- **Open and community-driven -** As an open-source project, Iceberg is vendor-neutral and has a sizeable user community and ecosystem of compatible technologies.

The anatomy of Apache Iceberg

The Iceberg architecture comprises three layers:

- **Data files** contain raw data, often compressed, queryable Parquet format residing on cloud storage like AWS S3.
- **Metadata files** support query execution, versioning, and access. They contain metadata about data files, such as record count, partition membership, schema, partition information, and file paths. Whenever a change is made to the table, a new metadata file is created and becomes the latest version of the metadata in the catalog.
- **Catalog** makes tables discoverable by clients. It contains the current metadata pointer for all tables.



Anatomy of Apache Iceberg

Apache Iceberg use cases

Iceberg can process petabytes of data, which makes it ideal for large-scale data engineering use cases, such as:

- Multi-tenant data lakes
- Data warehousing on top of data lakes
- ETL pipelines
- Historical analysis
- Log analytics
- Machine learning feature stores
- Real-time analytics

When is Apache Iceberg not a good fit?

Iceberg is a versatile platform for handling large-scale data analytics, but there are scenarios where it might not be the best choice.

Consider alternatives if the following apply:

- **Small data sets and simple queries -** For smaller data sets or simple analytics, the technical complexity of Iceberg might be overkill. Consider alternatives such as JSON, CSV, or Parquet.
- **Read-only workloads -** Iceberg excels at handling updates and deletions. If your workloads are primarily read-only, these advanced features may go unused, and simpler formats like raw Parquet might be sufficient.

Streaming data into Apache Iceberg

Streaming data into Iceberg facilitates real-time business innovation

Iceberg is an ideal destination for streaming data to enable real-time and historical analytics use cases. Structured and unstructured data, such as financial trading data, point-of-sale transactions, website clicks, system logs, vehicle GPS data, factory equipment readings (IoT), and other information, can be streamed into Iceberg tables. Once stored, this information is immediately available to various downstream analytics systems supporting the Iceberg format.

Applications like Snowflake, Spark, and Flink can query and process data streamed into Iceberg tables to support high-stakes use cases, such as:

- Real-time monitoring and alerting
- AI inference, model training, and feature engineering
- Historical analysis and auditing
- Anomaly detection

The challenges of streaming data into Iceberg

Historically, streaming data into Iceberg requires a substantial amount of data engineering work and infrastructure cost to create and run an often-brittle series of jobs that:

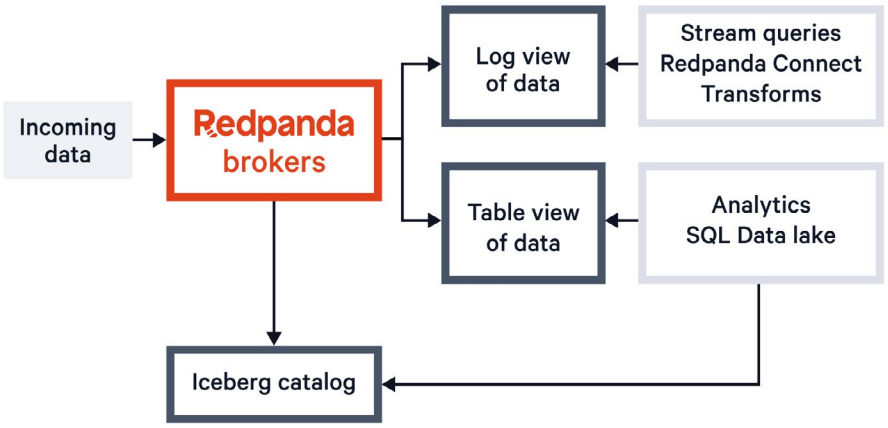
a. Output the topic data (JSON) into a queryable format such as Parquet
b. Update Iceberg metadata and catalog with schema information
c. Cost money to host and run
d. Require modifications every time the topic schema changes
e. Must be repeated for every topic you need to stream into an Iceberg lakehouse

Introducing Redpanda Iceberg Topics

Redpanda greatly simplifies storing topic data from the Redpanda streaming platform in the Iceberg open table format. The ability to insert messages directly into Iceberg tables is now a native capability of Redpanda brokers, making your streaming data immediately available in Iceberg for downstream analytical systems without creating and maintaining topic-to-Iceberg ETL pipelines.

Rather than configuring a separate component to funnel data to Iceberg, the control over what data lands in your Iceberg data lake is left in the hands of the topic owner. Every Iceberg-enabled topic creates a single Iceberg table. All your data for that topic, no matter what partition it's written to, appears in a single view.

To start writing streaming data to Iceberg, applications can simply set a topic property, and data begins flowing from the topic into your Iceberg data lake.



Streaming data to Iceberg with Redpanda

How Redpanda Iceberg Topics work:

1. Configure your Redpanda cluster and topic to run in Iceberg mode.
2. As usual, the topic can be viewed as a log through the Kafka API.
3. In addition, data is also fed directly into Iceberg tables.
4. Redpanda will create a schema in Iceberg that matches the schema defined for the topic (or just use a simple key-value schema).

Now that you know the basics, it's time to roll up your sleeves and see it in action.

Stream data to Apache Iceberg in three easy steps

Let's say you already have a Redpanda Enterprise cluster configured for Tiered Storage. It has a topic named `ClickEvent` for streaming website click events to an Iceberg table. `ClickEvent` topic messages have the following schema:

```
{
    "type" : "record",
    "namespace" : "com.redpanda.examples.avro",
    "name" : "ClickEvent",
    "fields" : [
       { "name": "user_id", "type" : "int" },
       { "name": "event_type", "type" : "string" },
       { "name": "ts", "type": "string" }
    ]
 }
```

With Iceberg Topics, you can set this topic to stream into Iceberg in three steps:

Step 1. Update the Redpanda cluster configurations to integrate with the Iceberg REST catalog.

```
rpk cluster config edit
<update the following configurations…>
iceberg_enabled: true
iceberg_rest_catalog_type: rest
iceberg_rest_catalog_endpoint: http://catalog-service:8181
iceberg_rest_catalog_client_id: <rest-connection-user>
iceberg_rest_catalog_client_secret: <rest-connection-secret>
iceberg_rest_catalog_prefix: ‘my_warehouse’
```

Step 2. Set the topic to Iceberg mode.

```
rpk topic alter-config clickEvent --set redpanda.iceberg.mode=value_
schema_id_prefix
```

Step 3. Specify that the ClickEvent schema will define the Iceberg table schema.

```
rpk registry schema create ClickEvent-value --schema path/to/schema.
avsc --type avro
```

Now, just configure your Iceberg processor (e.g., Spark) to connect to the same catalog.

```
spark.sql.catalog.streaming = org.apache.iceberg.spark.SparkCatalog
spark.sql.catalog.streaming.type = rest
spark.sql.catalog.streaming.uri = http://catalog-service:8181
Spark.sql.catalog.streaming.warehouse = ‘my_warehouse’
```

That’s it! No ETL jobs to code and host. No connector/sink to run.

Best of all, it integrates valuable streaming data into your analytics ecosystem and reduces the burden on your data engineering team. Win-win.

Conclusion

Apache Iceberg is an ideal destination for streaming data to enable real-time and historical analytics use cases. Streaming data into Iceberg usually requires a substantial amount of data engineering work and infrastructure cost to create and run an often-brittle series of ETL jobs.

However, with native support for Iceberg as a target, Redpanda clusters and topics can now be configured to stream data directly into Iceberg tables. It eliminates the overhead of creating topic-to-Iceberg ETL jobs, allowing you to integrate valuable streaming data into your analytics ecosystem in a fraction of the time it used to take.

To get started with Redpanda, sign up for a free trial of Redpanda Enterprise and dig into the documentation. If you want more guidance or have questions before getting started, get in touch and our team will lead the way.

If, however, you're still on the fence, here are a few more resources:

Redpanda | Customer Sucess Stories
Redpanda | Tech Talks and Masterclasses
Blog | When to choose Redpanda vs. Apache Kafka?
Blog | Redpanda vs. Kafka with KRaft: Performance comparison
Report | Redpanda vs. Kafka: a practical guide for data leaders
Report | Redpanda vs. Confluent: A Performance and TCO Benchmark
Slack | Ask our team in the Redpanda Community

Ready to learn more?

Chat with our team for a demo and expert advice on all things streaming data.

CHAT WITH US

Connect with us