Making Iceberg Easy with DuckDB-Iceberg

Tom Ebergen / tom@duckdblabs.com / @Tmonster

What is DuckDB?

DuckDB is easy to use

In-process

Portable (Zero dependencies)

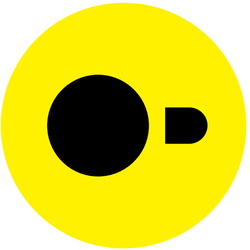
Feature-rich

Open-source under the MIT license

FireShip Youtube video

Current Iceberg Ecosystem

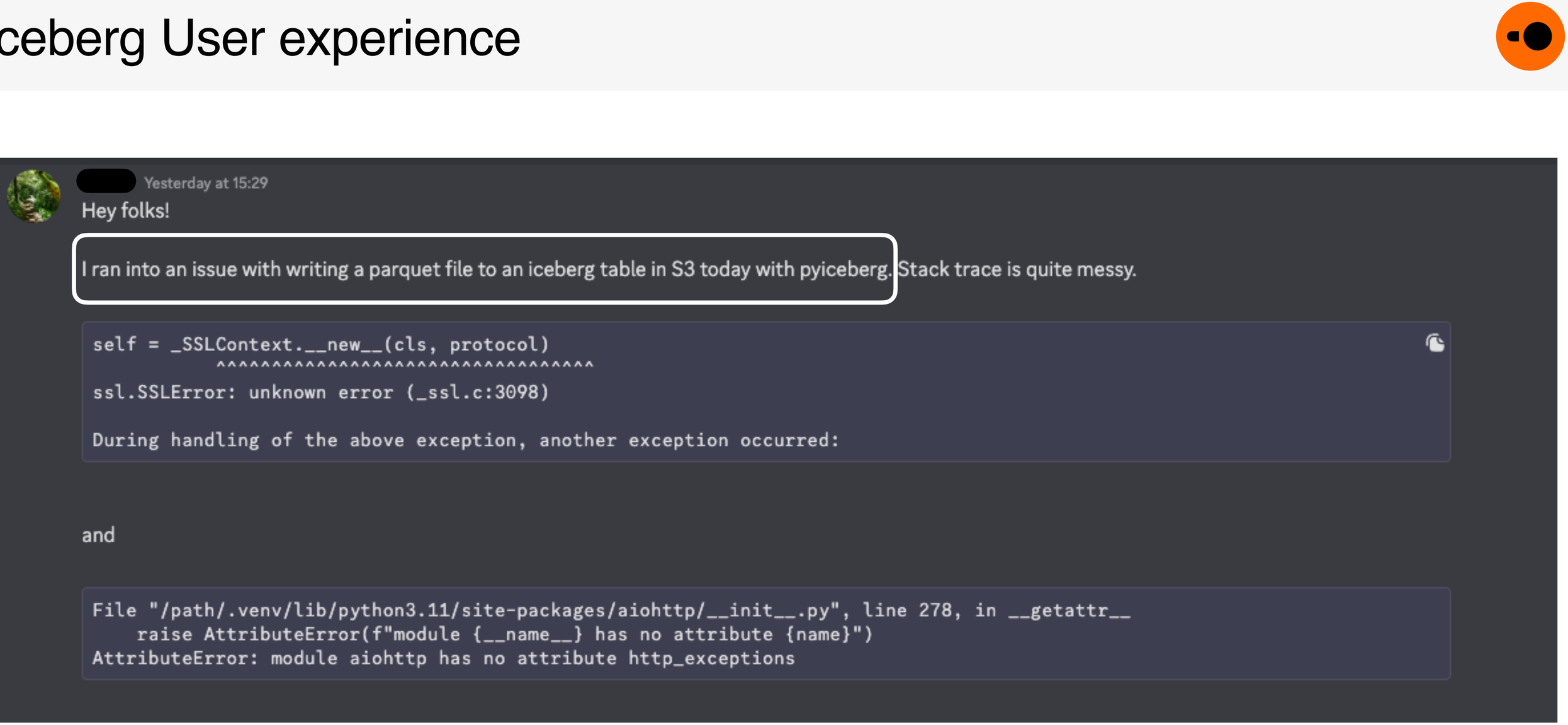
Engines



Catalogs

Good Support != Good User Experience

Iceberg User experience



[redacted] Yesterday at 15:29

Hey folks!

I ran into an issue with writing a parquet file to an iceberg table in S3 today with pyiceberg. Stack trace is quite messy.

```
self = _SSLContext.__new__(cls, protocol)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ssl.SSLError: unknown error (_ssl.c:3098)

During handling of the above exception, another exception occurred:
```

and

```
File "/path/.venv/lib/python3.11/site-packages/aiohttp/__init__.py", line 278, in __getattr__
    raise AttributeError(f"module {__name__} has no attribute {name}")
AttributeError: module aiohttp has no attribute http_exceptions
```

Getting Started

Apache Iceberg with Spark

- We know,
  - The Rest Catalog URI
  - Some Authentication token that we need

1. ?
2. ?
3. ?
4. ?

Apache Iceberg with Spark

- We know,
  - The Rest Catalog URI
  - Some Authentication token that we need

1. Install Java
2. Know what spark iceberg runtime you want to use
3. Create a python virtual environment
   1. Install the correct version of spark according to your spark iceberg runtime that is also compatible with your java version
4. Write python code

Apache Iceberg with Spark

```
import pyspark
from pyspark.conf import SparkConf
from pyspark.sql import SparkSession

SPARK_VERSION = pyspark.__version__
SPARK_MINOR_VERSION = '.'.join(SPARK_VERSION.split('.')[:2])
ICEBERG_VERSION = "1.7.0"

config = {
    "spark.sql.defaultCatalog": "demo",
    "spark.sql.catalog.demo": "org.apache.iceberg.spark.SparkCatalog",
    "spark.sql.catalog.demo.type": "rest",
    "spark.sql.catalog.demo.uri": "http://localhost:8181/catalog/",
    "spark.sql.catalog.demo.warehouse": "demo",
    "spark.sql.extensions": "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
    "spark.jars.packages": f"""org.apache.iceberg:iceberg-spark-runtime-{SPARK_MINOR_VERSION}_2.12:
{ICEBERG_VERSION},org.apache.iceberg:iceberg-azure-bundle:{ICEBERG_VERSION},org.apache.iceberg:iceberg-aws-
bundle:{ICEBERG_VERSION},org.apache.iceberg:iceberg-gcp-bundle:{ICEBERG_VERSION}"""
}

spark_config = SparkConf().setMaster('local').setAppName("Iceberg-REST")
for k, v in config.items():
    spark_config = spark_config.set(k, v)

spark = SparkSession.builder.config(conf=spark_config).getOrCreate()

spark.sql("USE demo")
```

Pray you don’t get an error

```
>>> spark.sql("create table default.spark_table (a int, b int)")
DataFrame[]
>>> spark.sql("insert into default.spark_table values (0, 0), (1, 1)")
25/09/05 10:45:29 ERROR Utils: Aborting task                                (0 + 1) / 1]
software.amazon.awssdk.core.exception.SdkClientException: Received an UnknownHostException when attempting to interact with a
service. See cause for the exact endpoint that is failing to resolve. If this is happening on an endpoint that previously
worked, there may be a network connectivity issue or your DNS cache could be storing endpoints for too long.
            at software.amazon.awssdk.core.exception.SdkClientException$BuilderImpl.build(SdkClientException.java:111)
            at  software.amazon.awssdk.awscore.interceptor.HelpfulUnknownHostExceptionInterceptor.modifyException
                (HelpfulUnknownHostExceptionInterceptor.java:59)
            at software.amazon.awssdk.core.interceptor.ExecutionInterceptorChain.modifyException(ExecutionInterceptorChain
                .java:181)
            at software.amazon.awssdk.core.internal.http.pipeline.stages.utils.ExceptionReportingUtils.runModifyException
                (ExceptionReportingUtils.java:54)
            at software.amazon.awssdk.core.internal.http.pipeline.stages.utils.ExceptionReportingUtils.reportFailureTo
                Interceptors(ExceptionReportingUtils.java:38)
            at software.amazon.awssdk.core.internal.http.pipeline.stages.ExecutionFailureExceptionReportingStage.
                execute(ExecutionFailureExceptionReportingStage.java:39)
            at software.amazon.awssdk.core.internal.http.pipeline.stages.ExecutionFailureExceptionReportingStage.
                execute(ExecutionFailureExceptionReportingStage.java:26)
            at software.amazon.awssdk.core.internal.http.AmazonSyncHttpClient$RequestExecutionBuilderImpl.
                execute(AmazonSyncHttpClient.java:210)
            at software.amazon.awssdk.core.internal.handler.BaseSyncClientHandler.invoke(BaseSyncClientHandler.java:103)
            at software.amazon.awssdk.core.internal.handler.BaseSyncClientHandler.doExecute(BaseSyncClientHandler.java:173)
            at software.amazon.awssdk.core.internal.handler.BaseSyncClientHandler.lambda$execute$1
                (BaseSyncClientHandler.java:80)
            at software.amazon.awssdk.core.internal.handler.BaseSyncClientHandler.measureApiCallSuccess(
                BaseSyncClientHandler.java:182)
            at software.amazon.awssdk.core.internal.handler.BaseSyncClientHandler.execute(BaseSyncClientHandler.java:74)
            at software.amazon.awssdk.core.client.handler.SdkSyncClientHandler.execute(SdkSyncClientHandler.java:45)
            at software.amazon.awssdk.awscore.client.handler.AwsSyncClientHandler.execute(AwsSyncClientHandler.java:53)
            at software.amazon.awssdk.services.s3.DefaultS3Client.putObject(DefaultS3Client.java:10595)
```

Getting Started

Getting started with Iceberg and DuckDB

- We know,
  - The Rest Catalog URI
  - Some Authentication token that we need

1. ???
2. ???
3. ???

Extra information / References

Getting started with Iceberg and DuckDB

- We know,
  - The Rest Catalog URI
  - Some Authentication token that we need

1. Install and Run DuckDB
2. Attach a catalog & Insert data
3. Enjoy your analytics

Extra information / References

Getting started with Iceberg and DuckDB

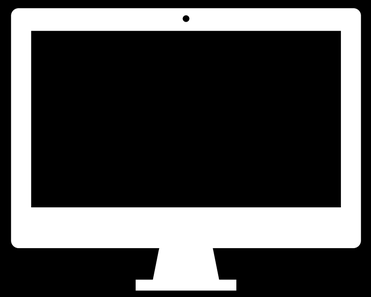
```
$ curl https://install.duckdb.org | sh
D: ATTACH '<warehouse name>' AS my_datalake (
      TYPE ICEBERG,
      CLIENT_ID '***',
      CLIENT_SECRET '***',
      ENDPOINT 'https://<endpoint>'
   );

D: -- optional if schema doesn't exist
D: CREATE SCHEMA my_datalake.default;
D: CREATE TABLE my_datalake.default.csv_table AS
         SELECT * FROM 'path/to/file.csv';
```

Getting started with Iceberg and DuckDB

```
D ATTACH '' AS my_datalake (
      TYPE ICEBERG,
      CLIENT_ID 'admin',
      CLIENT_SECRET 'password',
      ENDPOINT 'http://127.0.0.1:8181'
  );
D SELECT * FROM my_datalake.default.customer;

IO Error:
Could not establish connection error for HTTP GET to 'http://warehouse.127.0.0.1:9000/
default/customer/metadata/snap-1524418100976860318-1-896e0f5d-a7ff-4584-
bb4c-26ae0f37c0fb.avro'
```

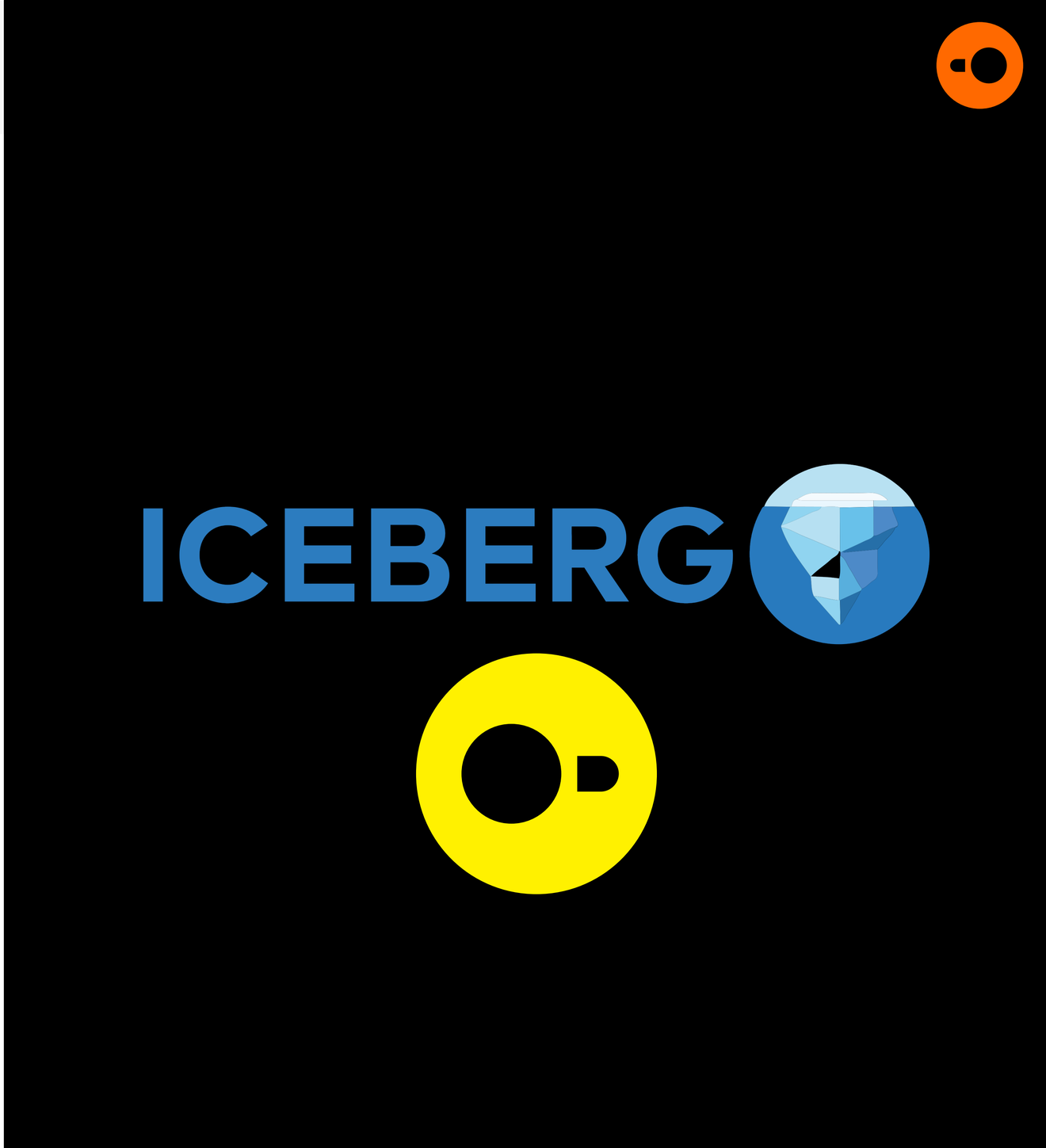


Demo

v1.4.0

DuckDB-Iceberg v1.4.0

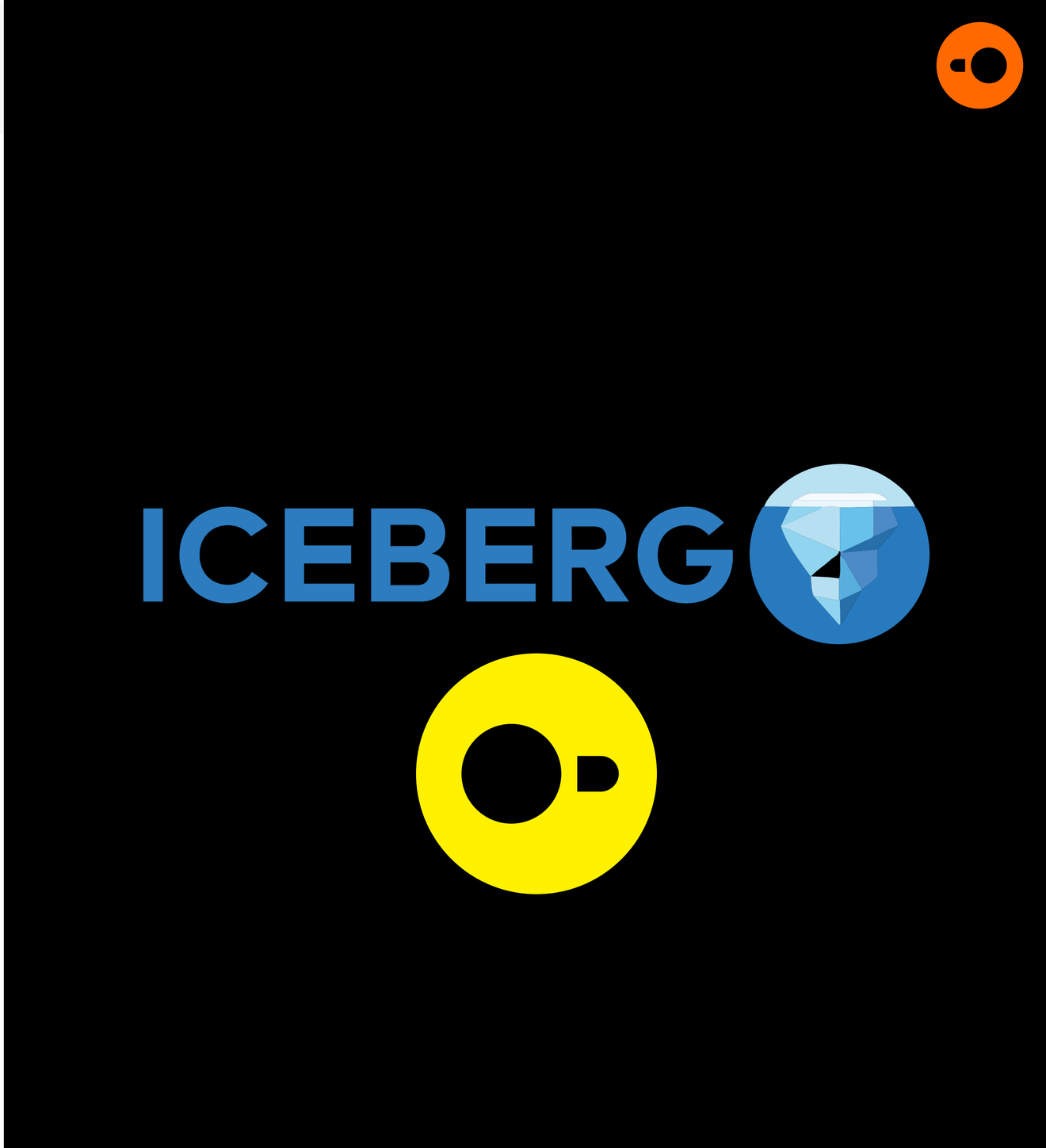
- Read and insert to v2 Iceberg tables*
- Read v3 Iceberg tables
- Connect to IRC catalogs backed by S3, GCS, or R2.
- Wasm support

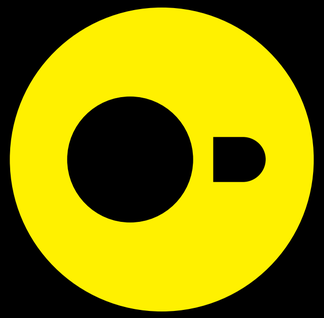


* With no sort order or partitioning

DuckDB-Iceberg v1.4++

- Update & Delete for v2 tables
  - planned for v1.4.1
- Insert/Delete/Update support for v3 tables.
- Compaction methods
- Support for Azure
- More Metadata query support
- Support Partitioning and Sorting Tables
- Writing without an IRC connection.





DuckDB-Iceberg

Repo: https://github.com/duckdb/duckdb-iceberg
GitHub: @Tmonster
Email: tom@duckdblabs.com