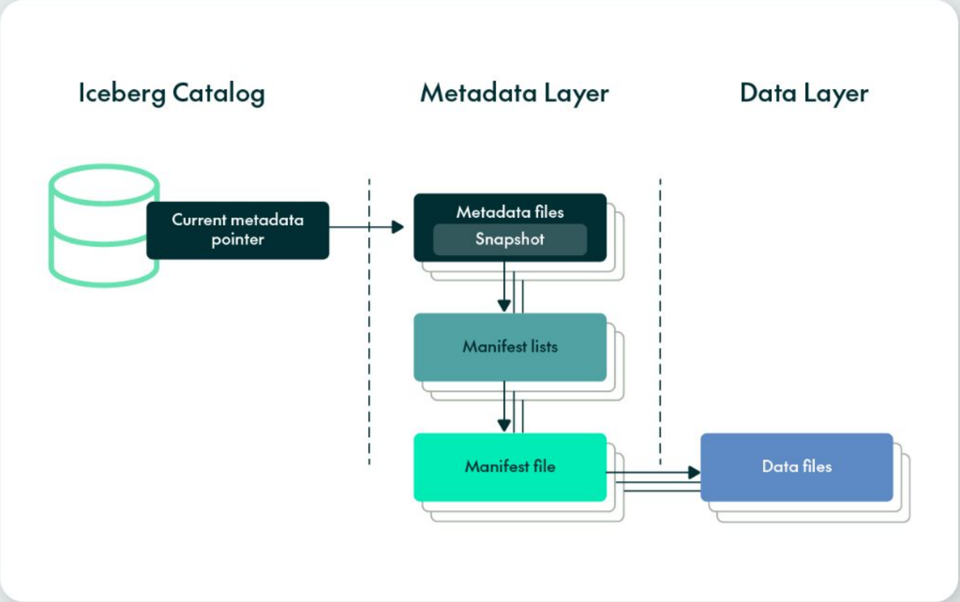
Getting Started - Apache Iceberg -

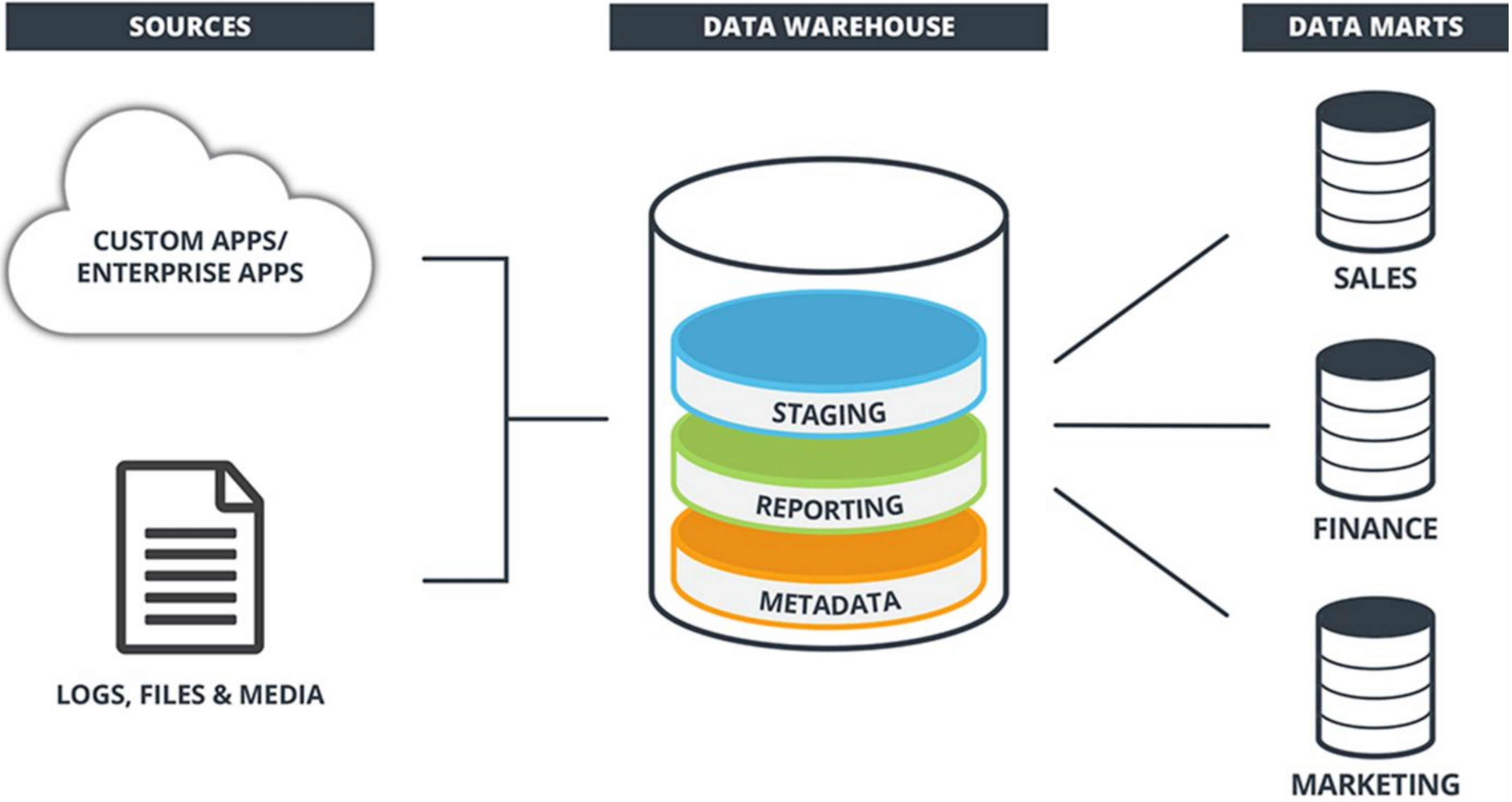
Dr. Firas

Author & Conference speaker

Getting Started - Apache Iceberg

- Combining Strengths
- Key Capabilities
- Benefits and Popularity
- Real-World Applications




SOURCES
DATA WAREHOUSE
DATA MARTS
CUSTOM APPS/
ENTERPRISE APPS
LOGS, FILES & MEDIA
STAGING
REPORTING
METADATA
SALES
FINANCE
MARKETING

Understanding Data Warehouses

■ Introduction to Data Warehouses

Definition and role as a centralized repository optimized for analytics and business intelligence.

■ Centralization and Organization

Goal of having a well-maintained, organized, and centralized data warehouse that stores most of an organization’s data.

■ Challenges with Structuring Data

The complex, messy task of structuring data to fit within a warehouse.

Issues arising from the ETL process: data duplication, delays in data availability, and reduced operational flexibility.

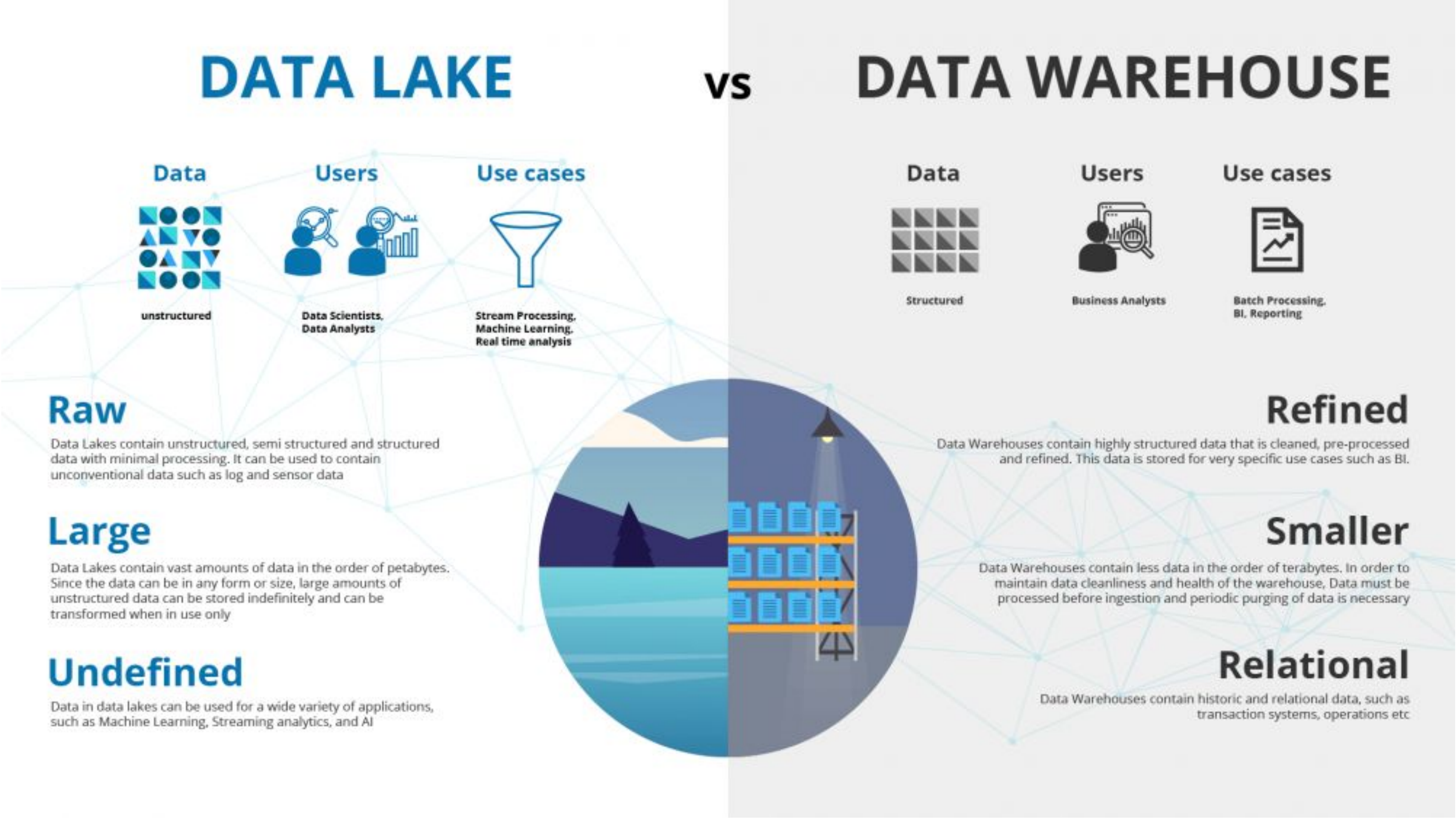
■ Maintenance Costs and Challenges

Ongoing, expensive, and labor-intensive efforts required to maintain a data warehouse.

Consequences of inadequate maintenance: reduced data accessibility or a completely ineffective system.

■ Evolving Needs and Limitations

Persistent challenges with cost, scalability, and maintenance that prompt the need for innovative solutions like Iceberg.



DATA LAKE vs DATA WAREHOUSE

DATA LAKE

Data

unstructured

Users

Data Scientists,
Data Analysts

Use cases

Stream Processing,
Machine Learning,
Real time analysis

Raw

Data Lakes contain unstructured, semi structured and structured data with minimal processing. It can be used to contain unconventional data such as log and sensor data

Large

Data Lakes contain vast amounts of data in the order of petabytes. Since the data can be in any form or size, large amounts of unstructured data can be stored indefinitely and can be transformed when in use only

Undefined

Data in data lakes can be used for a wide variety of applications, such as Machine Learning, Streaming analytics, and AI

DATA WAREHOUSE

Data

Structured

Users

Business Analysts

Use cases

Batch Processing,
BI, Reporting

Refined

Data Warehouses contain highly structured data that is cleaned, pre-processed and refined. This data is stored for very specific use cases such as BI.

Smaller

Data Warehouses contain less data in the order of terabytes. In order to maintain data cleanliness and health of the warehouse, Data must be processed before ingestion and periodic purging of data is necessary

Relational

Data Warehouses contain historic and relational data, such as transaction systems, operations etc

Understanding Data Lakes

ICEBERG

■ The Concept of a Data Lake

Explanation of data lakes storing data in its native format, avoiding rigorous structuring and massive ETL workloads.

Highlight the cost reduction and simplification of the data management stack.

■ Advantages and Simplification

Discussion of the operational streamlining promised by data lakes.

Transition: While appealing, this simplicity introduces significant challenges.

■ Challenges of Data Lakes

Detailed look at the complexities of extracting information from unstructured data.

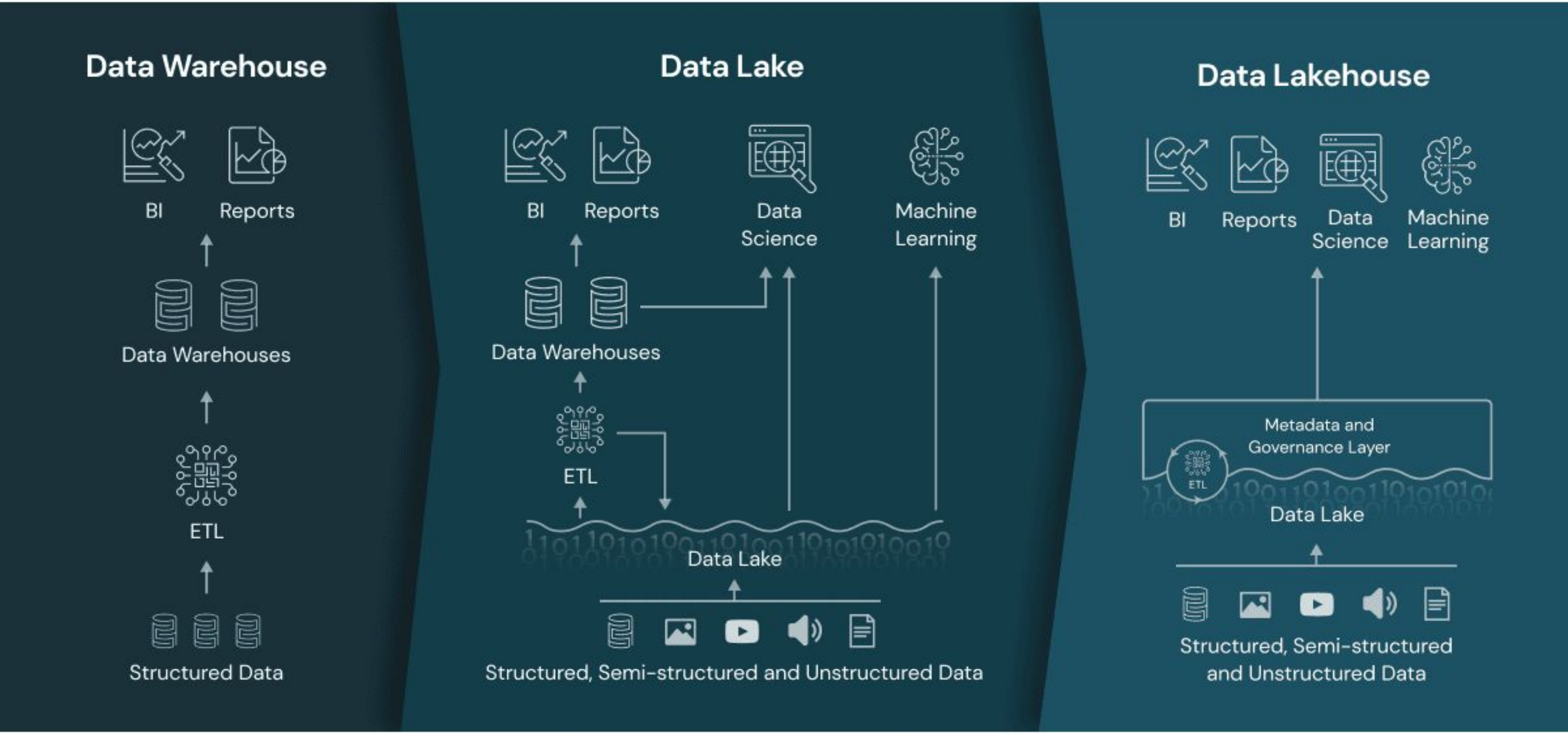
Impact on data scientists and analysts due to advanced requirements for data querying and management.

The evolution of data management challenges over time, leading to potential inefficiencies and data bogs.

■ A Thoughtful Consideration

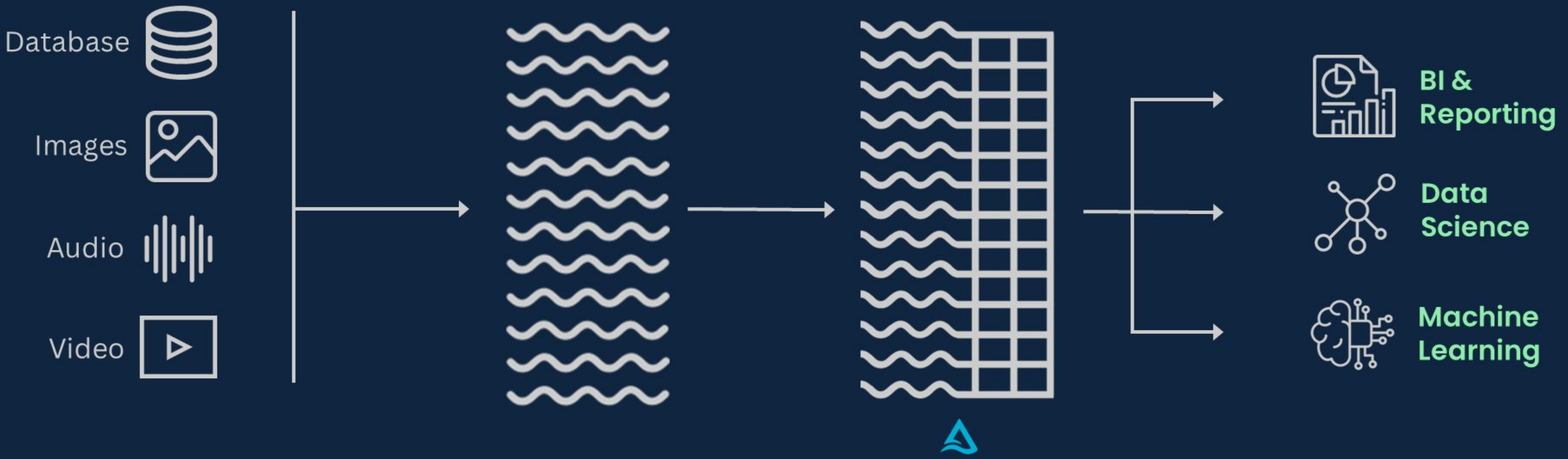
Introduction to the idea of hybrid solutions like data lakehouses.

A proposed solution that blends the flexibility of data lakes with the structured benefits of data warehouses.


Data Warehouse
BI
Reports
Data Warehouses
ETL
Structured Data
Data Lake
BI
Reports
Data Science
Machine Learning
Data Warehouses
ETL
Data Lake
Structured, Semi-structured and Unstructured Data
Data Lakehouse
BI
Reports
Data Science
Machine Learning
Metadata and Governance Layer
ETL
Data Lake
Structured, Semi-structured and Unstructured Data

Data Lakehouse

Combining both elements of Data Lakes and Warehouses



SOURCES

Structured +
Unstructured Data

DATA LAKE

Store any file format

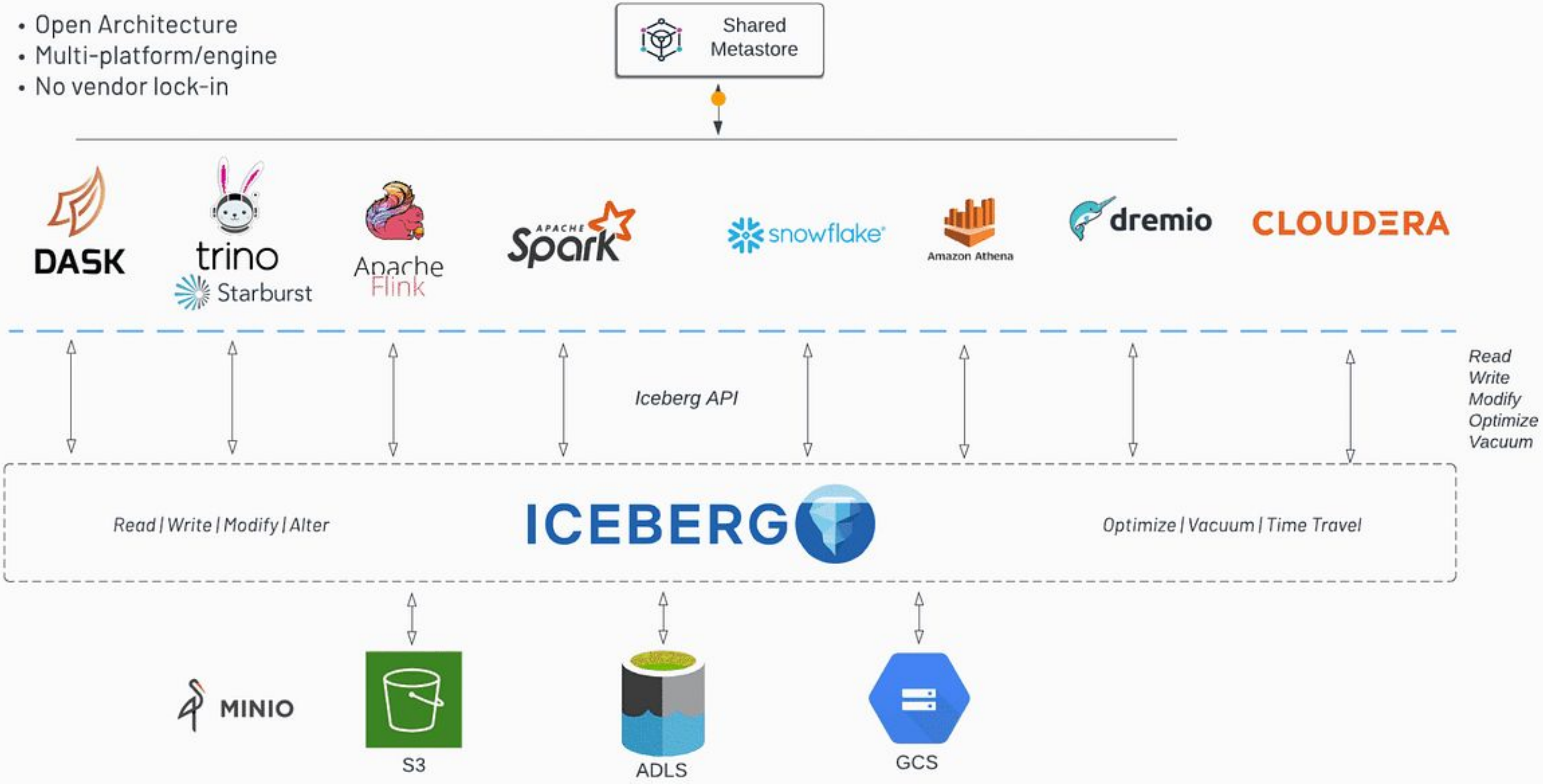
DELTA LAKE

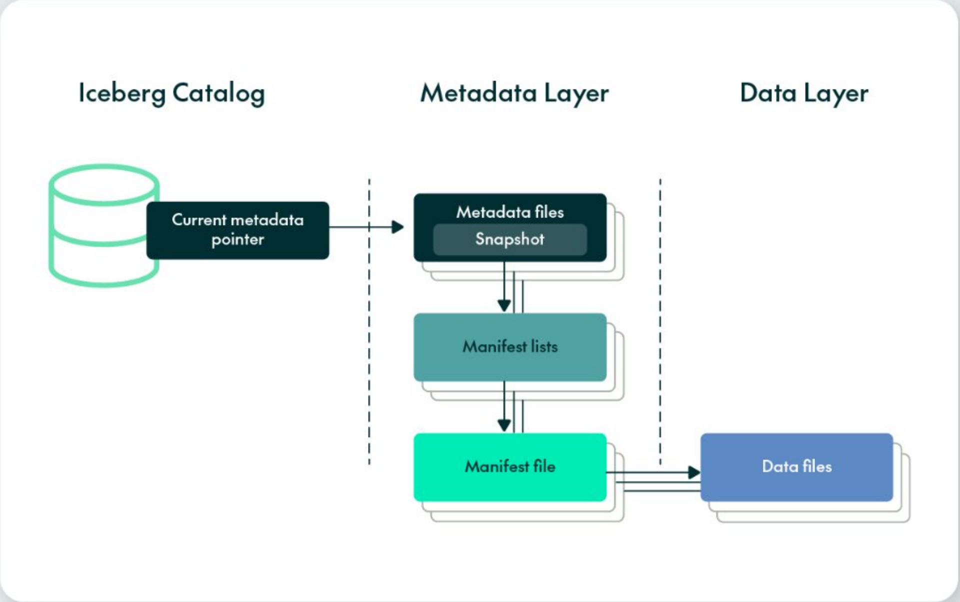
Database-like
features to Data lake

**DATA-DRIVEN
DECISION MAKING**

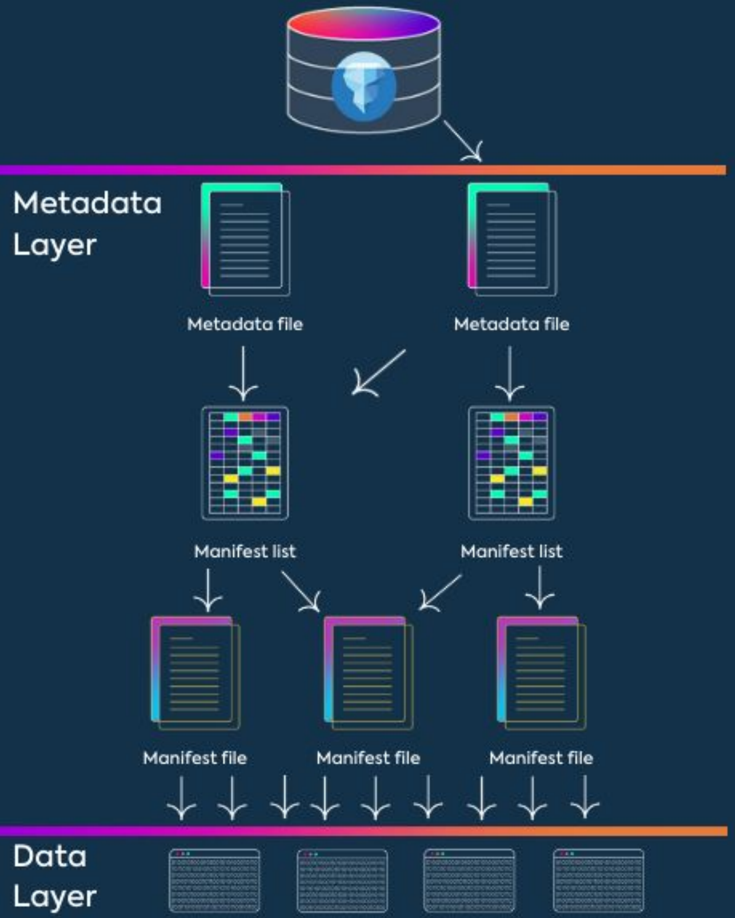
Make informed
business decisions

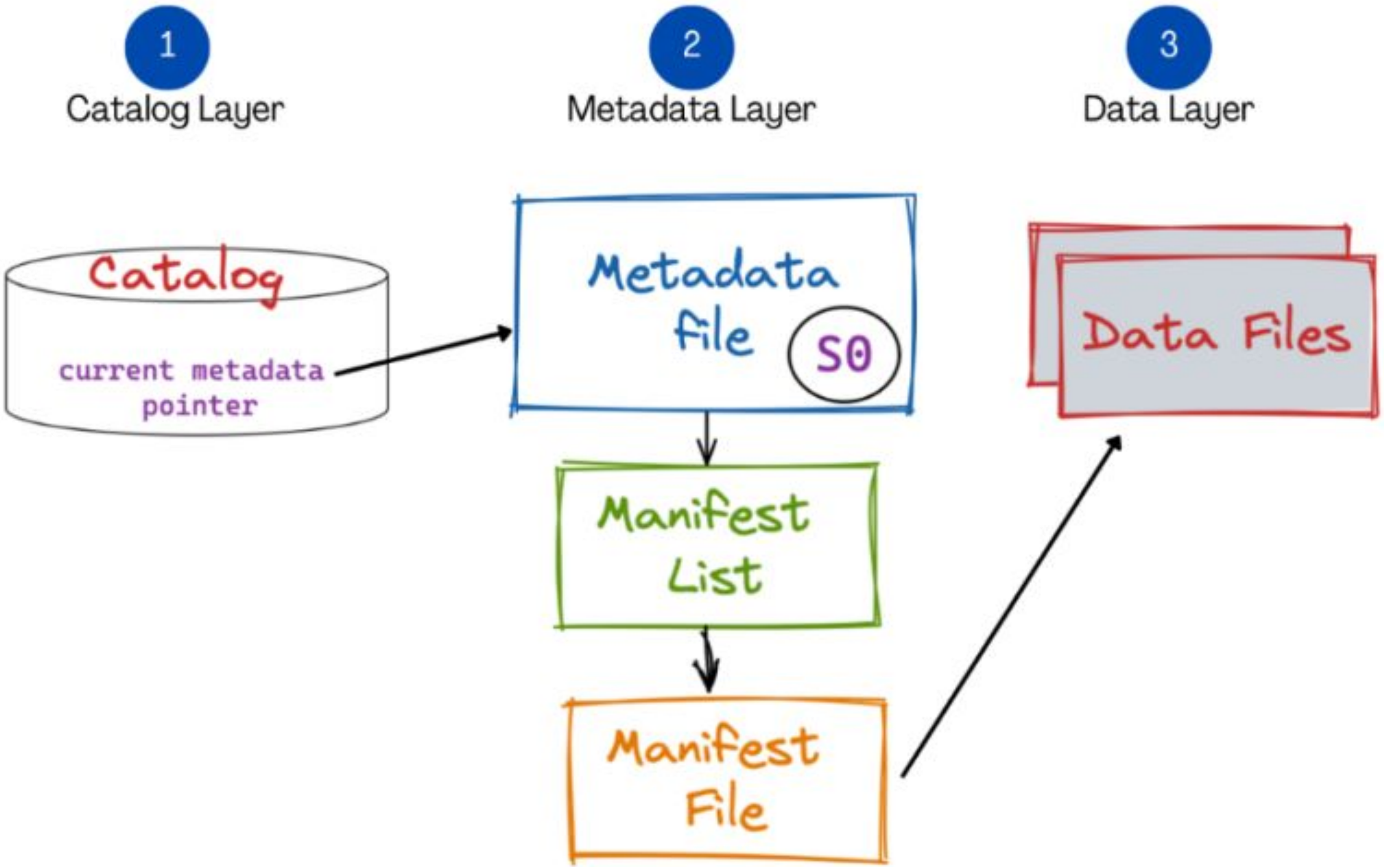
- Open Architecture
- Multi-platform/engine
- No vendor lock-in




Iceberg Catalog
Metadata Layer
Data Layer
Current metadata pointer
Metadata files
Snapshot
Manifest lists
Manifest file
Data files

Apache Iceberg Explained





1 Catalog Layer

2 Metadata Layer

3 Data Layer

current metadata pointer

Understanding Apache Iceberg Core Concepts

■ Introduction to Metadata Management

Overview of Iceberg’s metadata layer handling schemas, partitions, and file locations.

Explanation of metadata and manifest files stored in JSON format.

■ Schema Evolution

Definition and significance of schema evolution in adapting to changing data needs.

Example of adding a new column to employee data and how Iceberg updates metadata without affecting existing data.

■ Partitioning Strategies

Introduction to partitioning as a method for dividing data into manageable subsets for faster querying.

Description of different partitioning strategies:

Range partitioning (e.g., dates, numeric values), Hash partitioning (applying a hash function), Truncate partitioning (e.g., truncating zip codes), List partitioning (e.g., categorizing by company names)

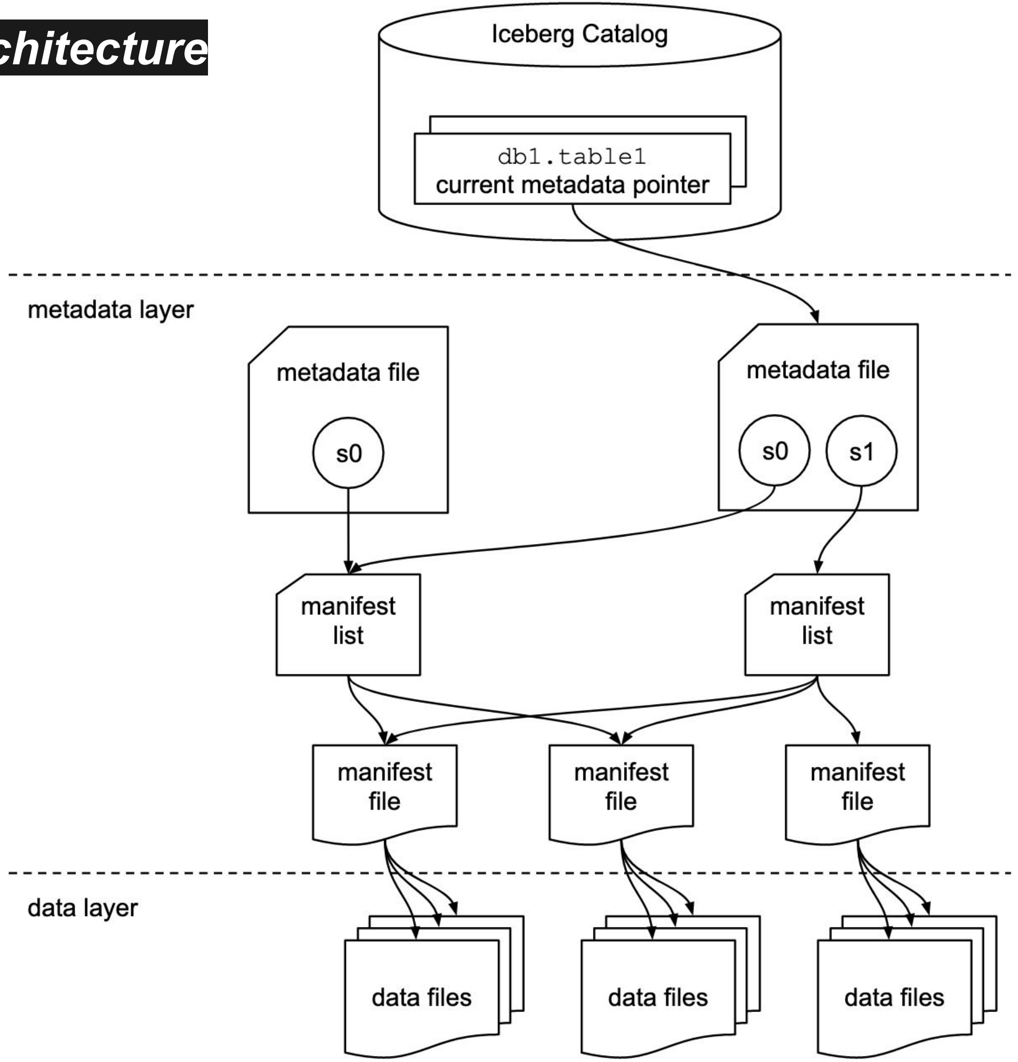
■ Snapshots and Their Importance

Explanation of how each data change creates a new snapshot with updated manifest files.

The role of snapshots in enabling historical data access and rollback capabilities.

Benefits of snapshot-based querying for maintaining data integrity and performing audits.

Iceberg Architecture



ICEBERG

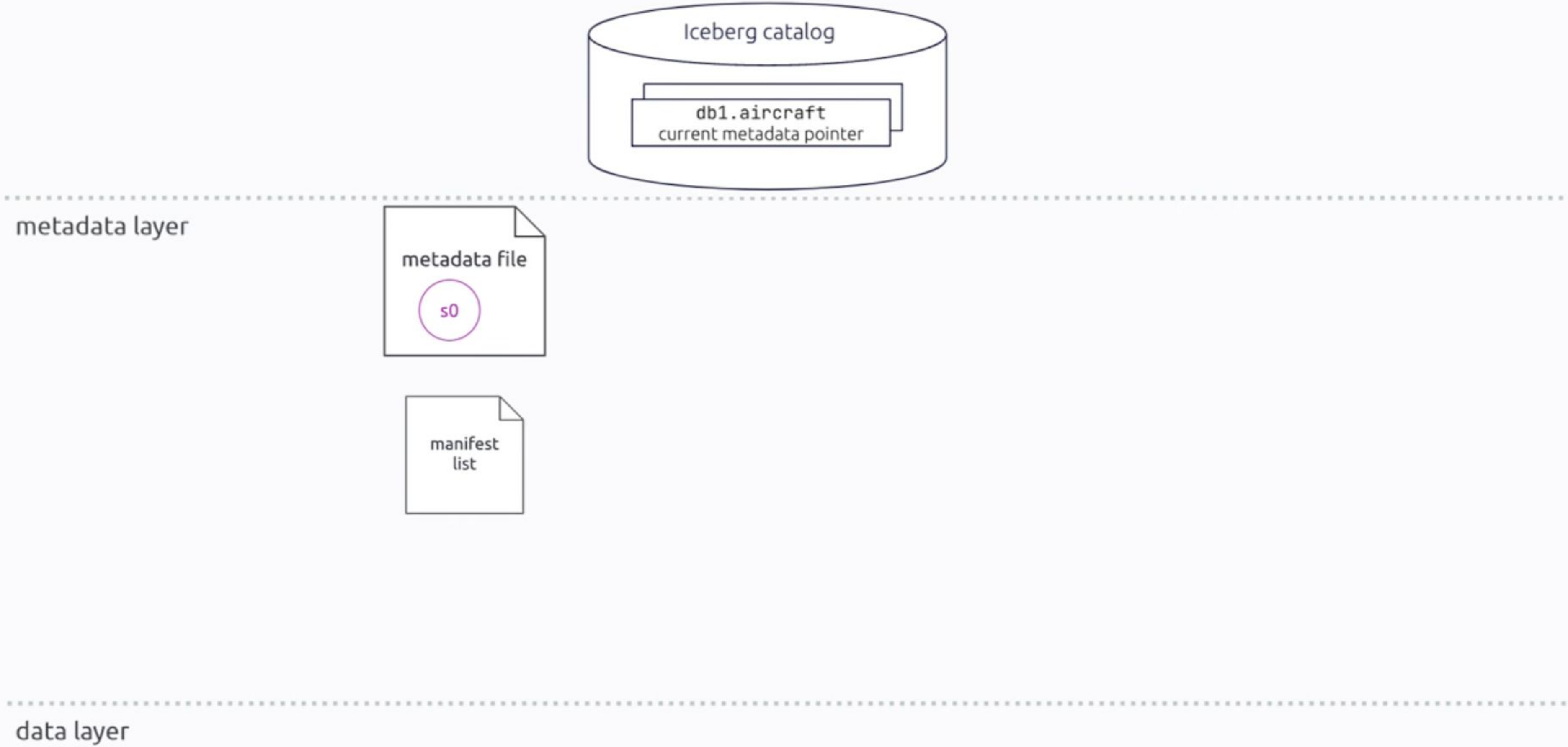
Improved query performance

Efficient metadata management

Support for ACID transactions

Large-scale analytics in modern cloud environments

```sql
CREATE TABLE aircraft (
  tail_number varchar(15),
  description varchar(150),
  class varchar(50),
  year integer
)
WITH
  (type = 'iceberg');
```



Iceberg catalog

db1.aircraft
current metadata pointer

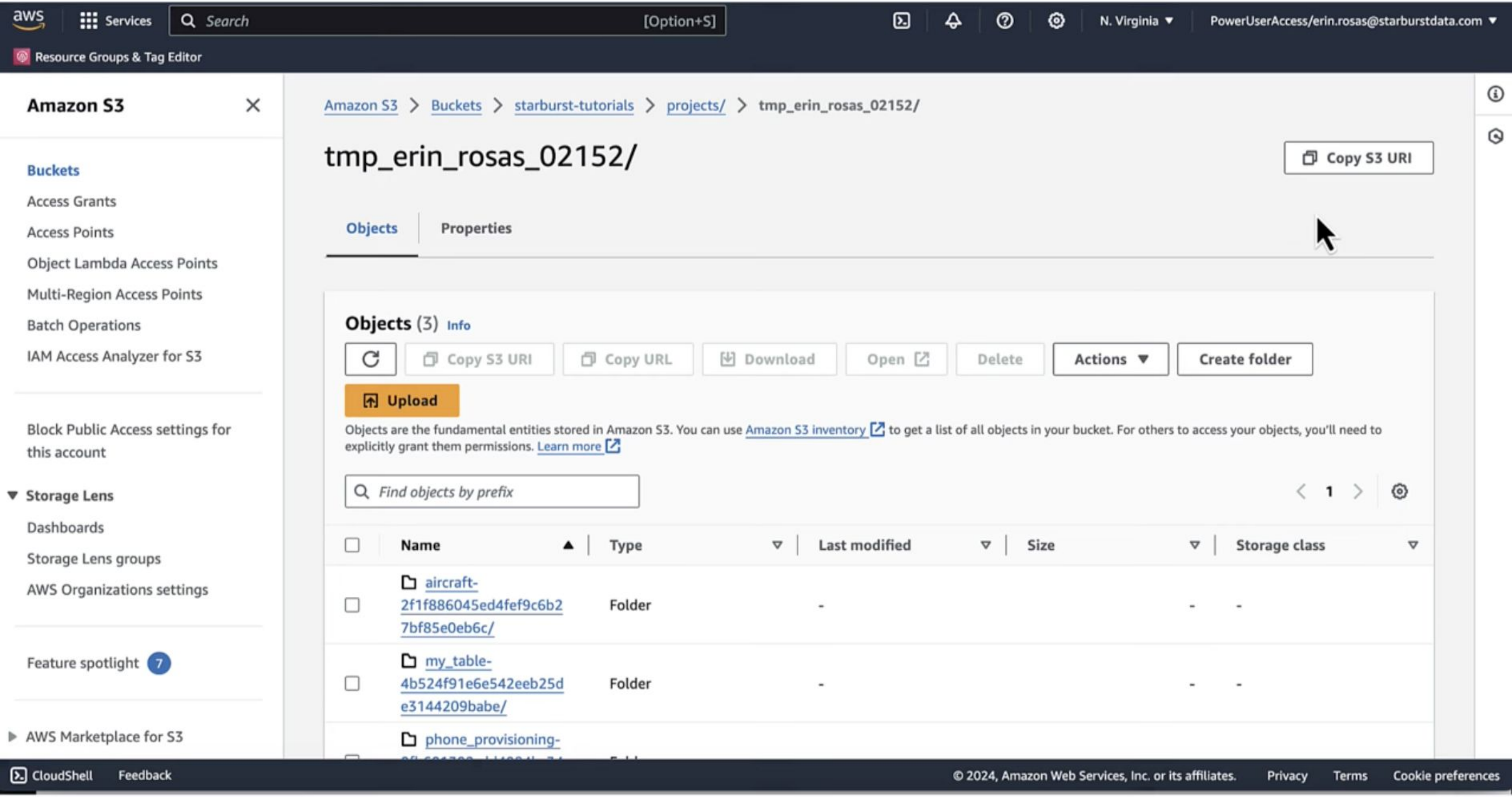
metadata layer

metadata file

s0

manifest list

data layer

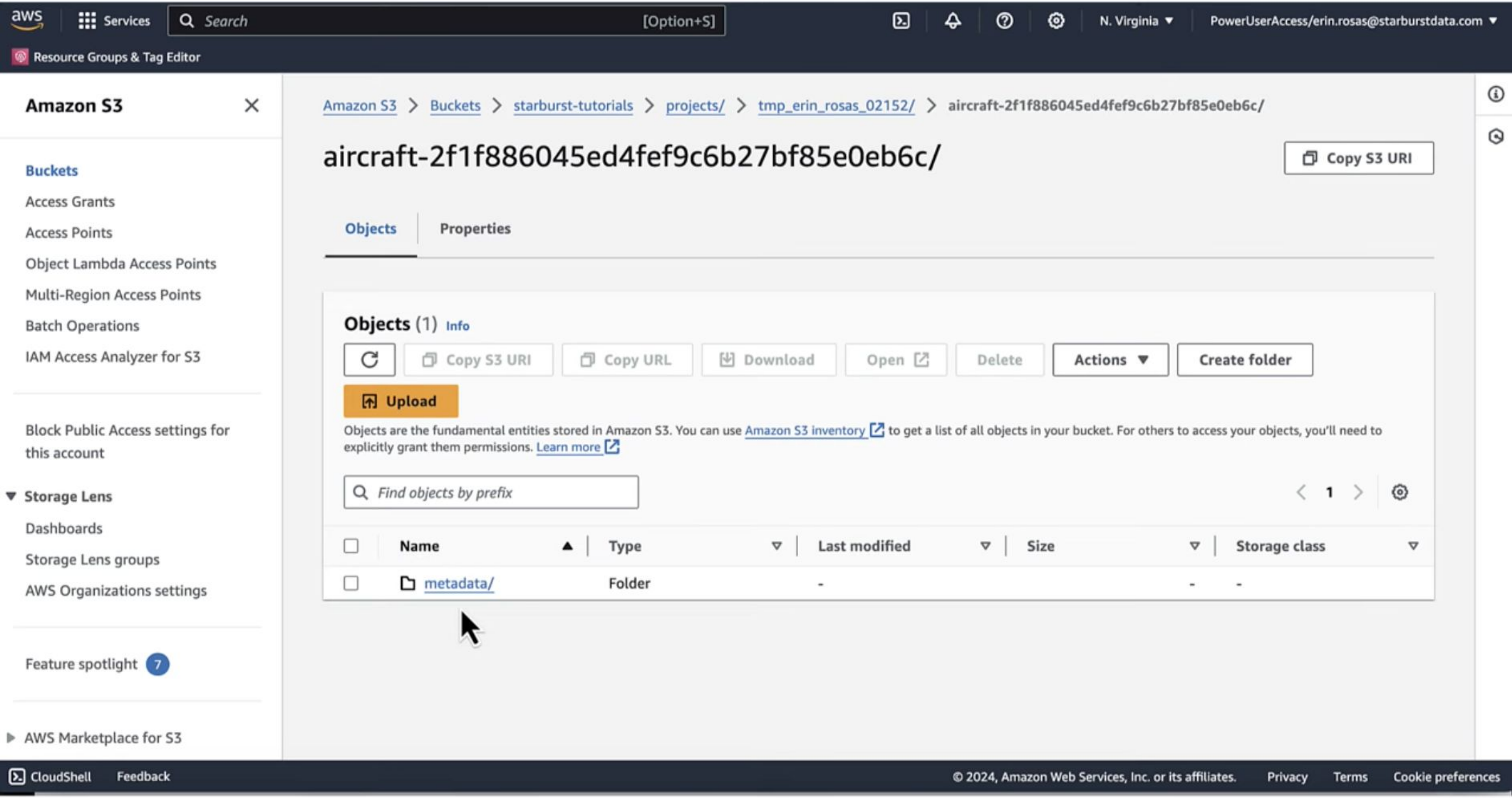

aws
Services
Search
[Option+S]
N. Virginia
PowerUserAccess/erin.rosas@starburstdata.com
Resource Groups & Tag Editor
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
AWS Marketplace for S3
Amazon S3 > Buckets > starburst-tutorials > projects/ > tmp_erin_rosas_02152/
Copy S3 URI
Objects
Properties
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

tmp_erin_rosas_02152/

Objects (3) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/	Folder	-	-	-
□	my_table-4b524f91e6e542eeb25de3144209babe/	Folder	-	-	-
□	phone_provisioning-				


aws
Services
Search
[Option+S]
N. Virginia ▼
PowerUserAccess/erin.rosas@starburstdata.com ▼
Resource Groups & Tag Editor
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
AWS Marketplace for S3
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

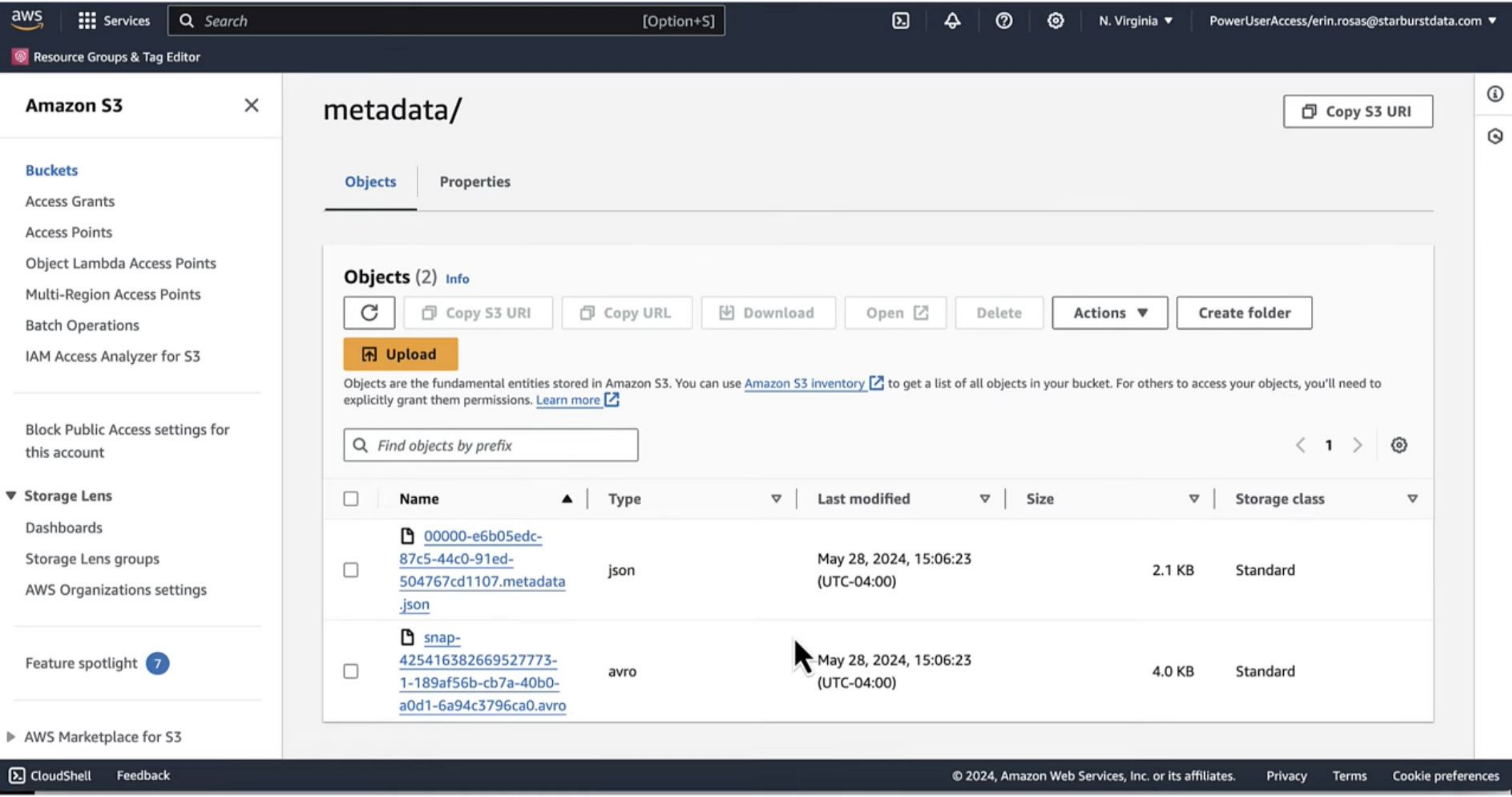
Amazon S3 > Buckets > starburst-tutorials > projects/ > tmp_erin_rosas_02152/ > aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/

aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/

Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	metadata/	Folder	-	-	-



metadata/

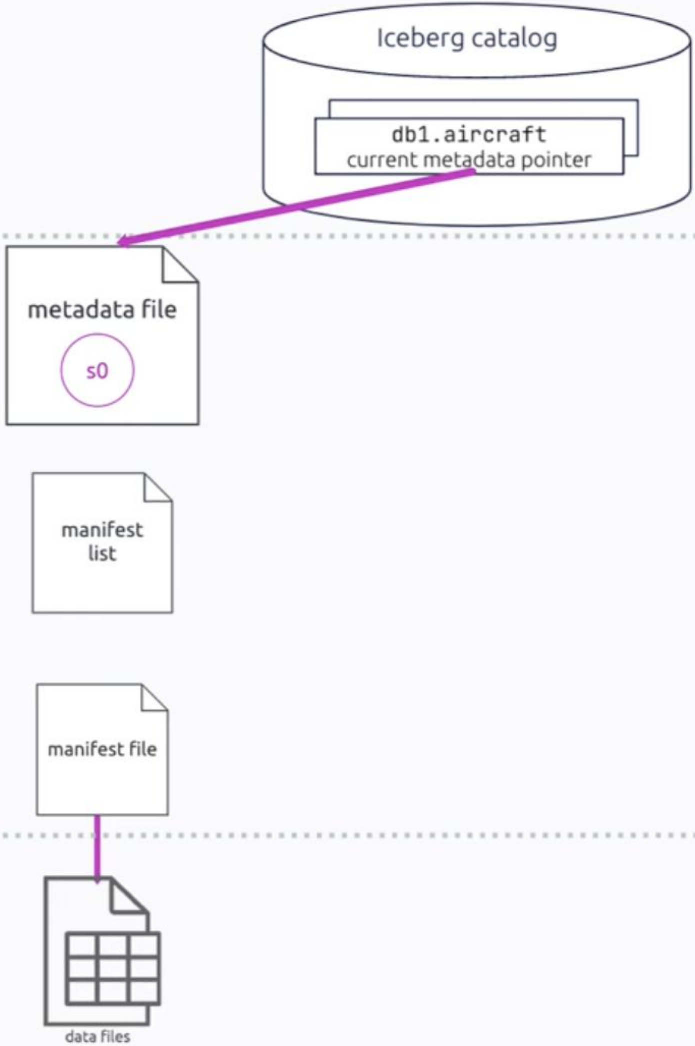
Objects (2) Info

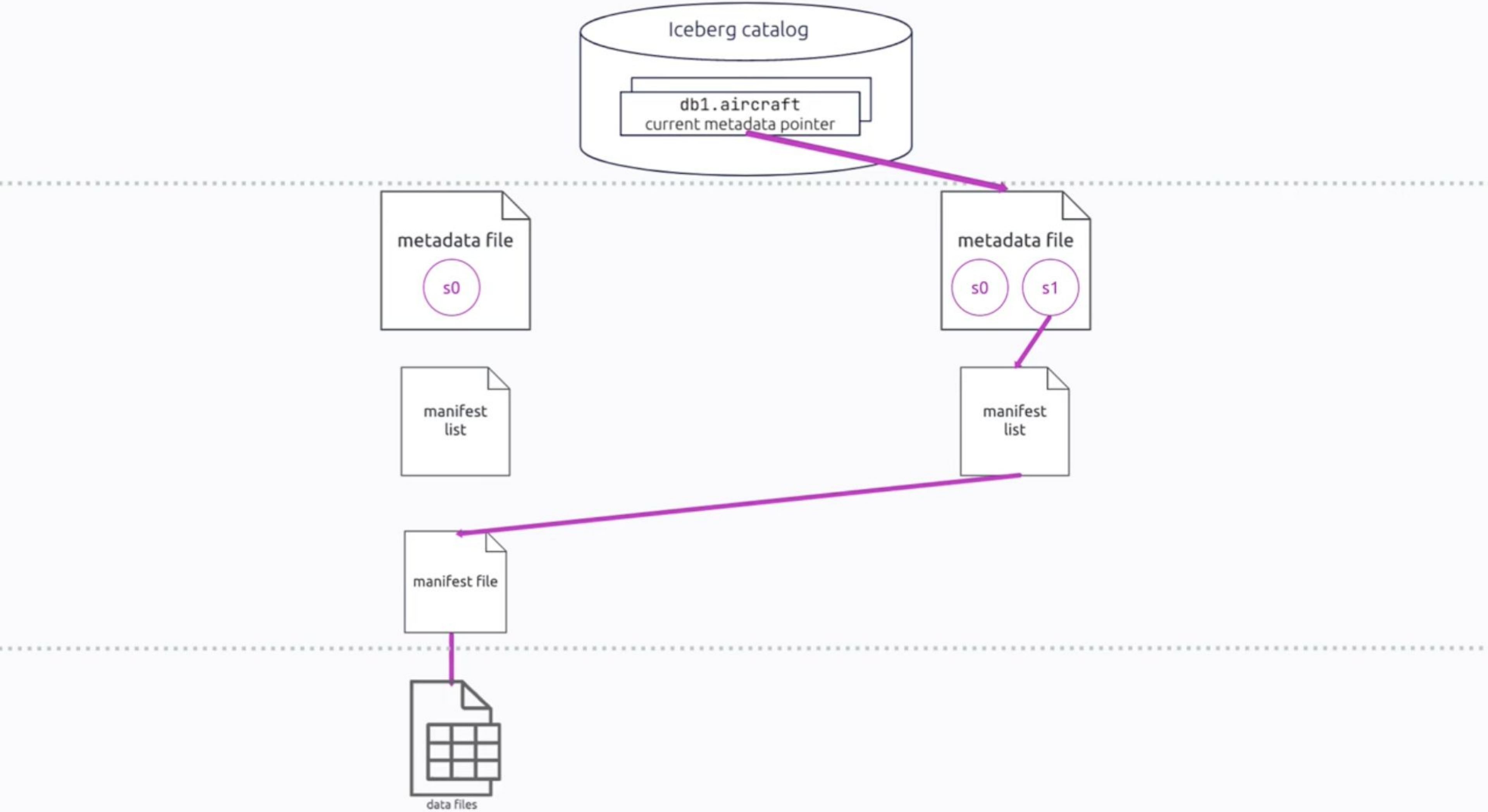
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	00000-e6b05edc-87c5-44c0-91ed-504767cd1107.metadata.json	json	May 28, 2024, 15:06:23 (UTC-04:00)	2.1 KB	Standard
□	snap-425416382669527773-1-189af56b-cb7a-40b0-a0d1-6a94c3796ca0.avro	avro	May 28, 2024, 15:06:23 (UTC-04:00)	4.0 KB	Standard

© 2024, Amazon Web Services, Inc. or its affiliates.

```sql
INSERT INTO
  aircraft (tail_number, description, class, year)
VALUES
  ('N535NA', 'NASA', 'Helicopter', 1969),
  ('N611TV', 'COOL', 'Jet', 1983);
```


Iceberg catalog
db1.aircraft
current metadata pointer
metadata file
s0
manifest list
manifest file
data files



Iceberg catalog

db1.aircraft

current metadata pointer

metadata file

s0

metadata file

s0

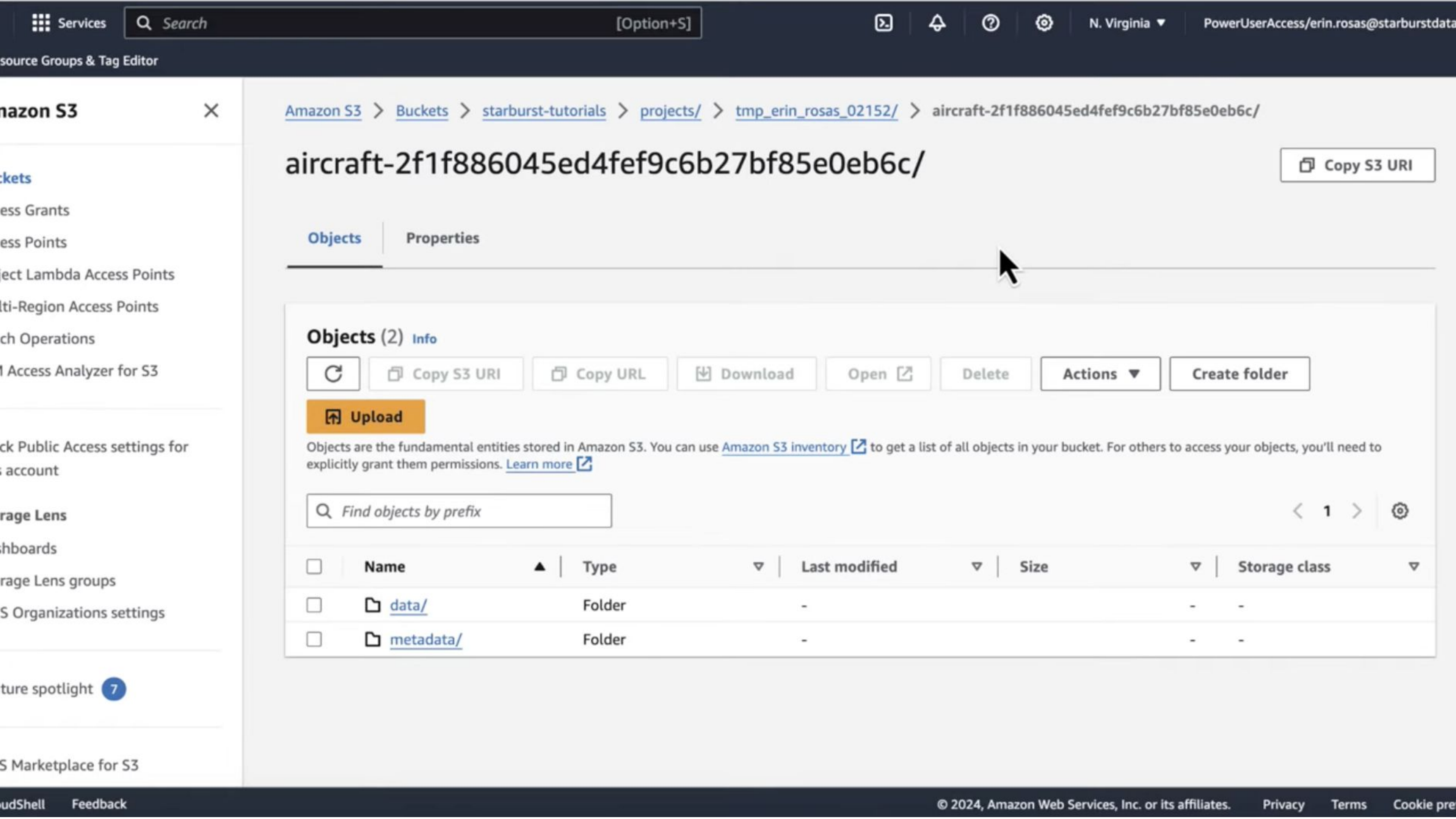
s1

manifest list

manifest list

manifest file

data files



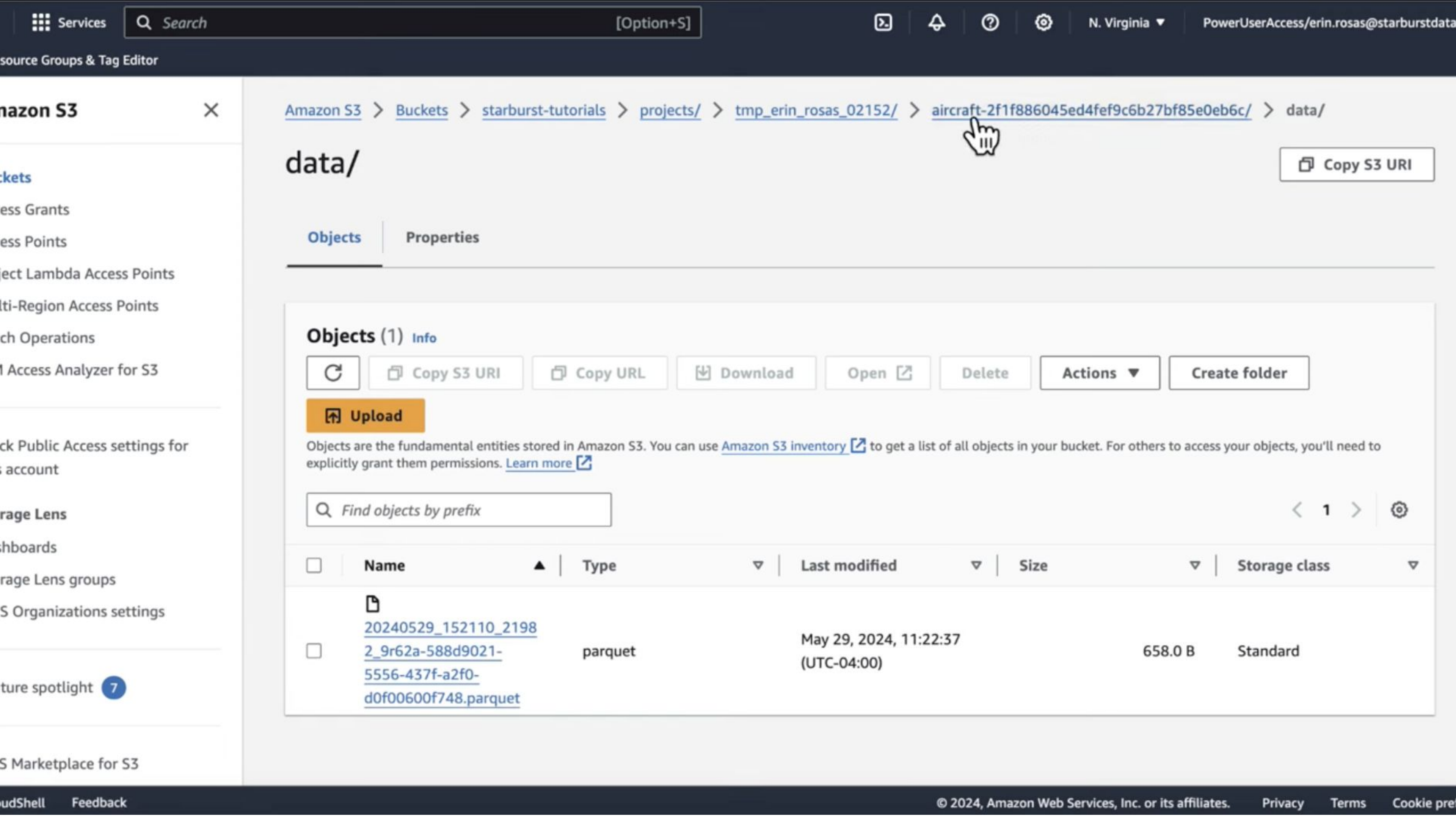
Amazon S3 > Buckets > starburst-tutorials > projects/ > tmp_erin_rosas_02152/ > aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/

aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	data/	Folder	-	-	-
□	metadata/	Folder	-	-	-



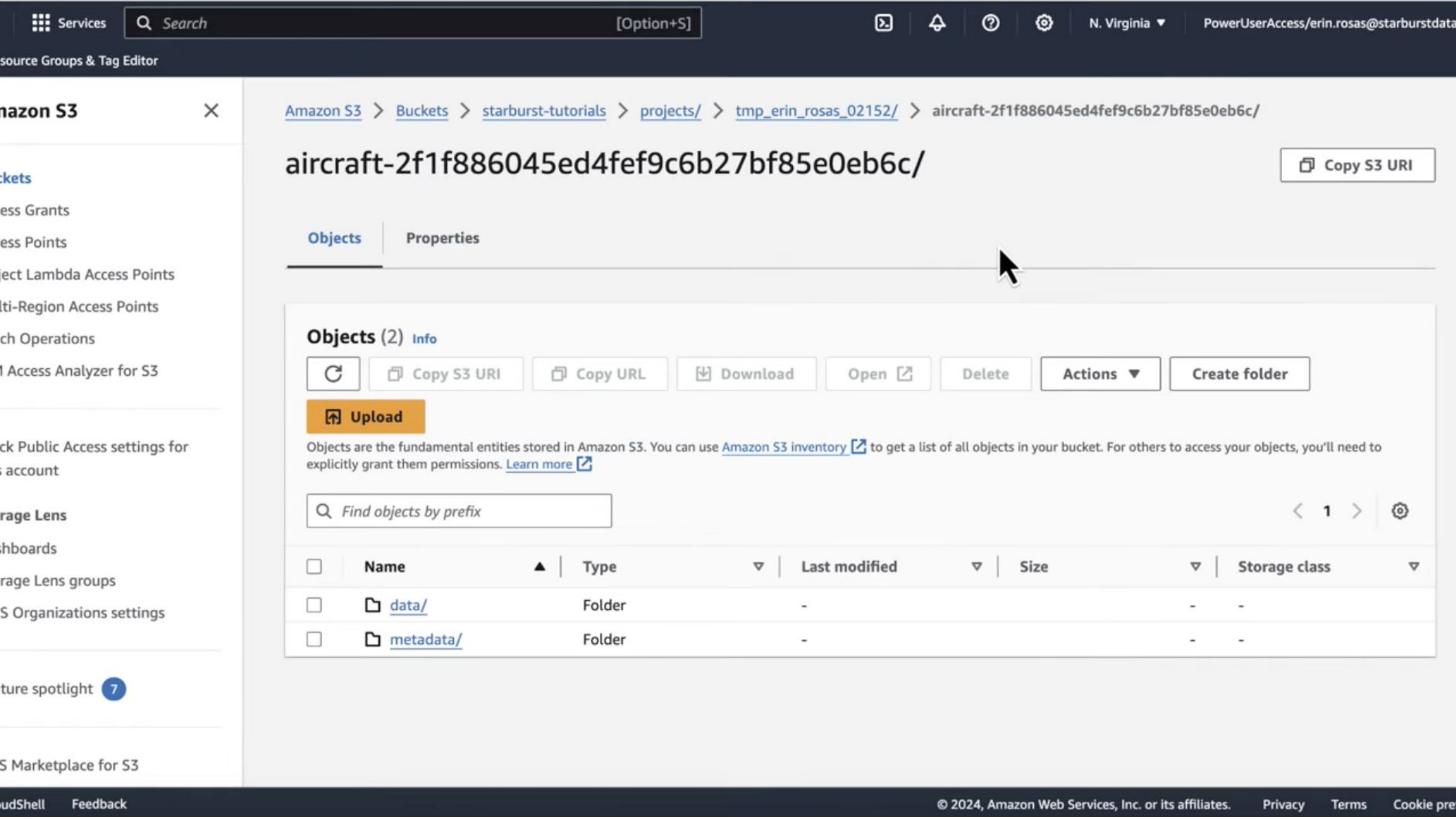
Amazon S3 > Buckets > starburst-tutorials > projects/ > tmp_erin_rosas_02152/ > aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/ > data/

data/

Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	20240529_152110_21982_9r62a-588d9021-5556-437f-a2f0-d0f00600f748.parquet	parquet	May 29, 2024, 11:22:37 (UTC-04:00)	658.0 B	Standard



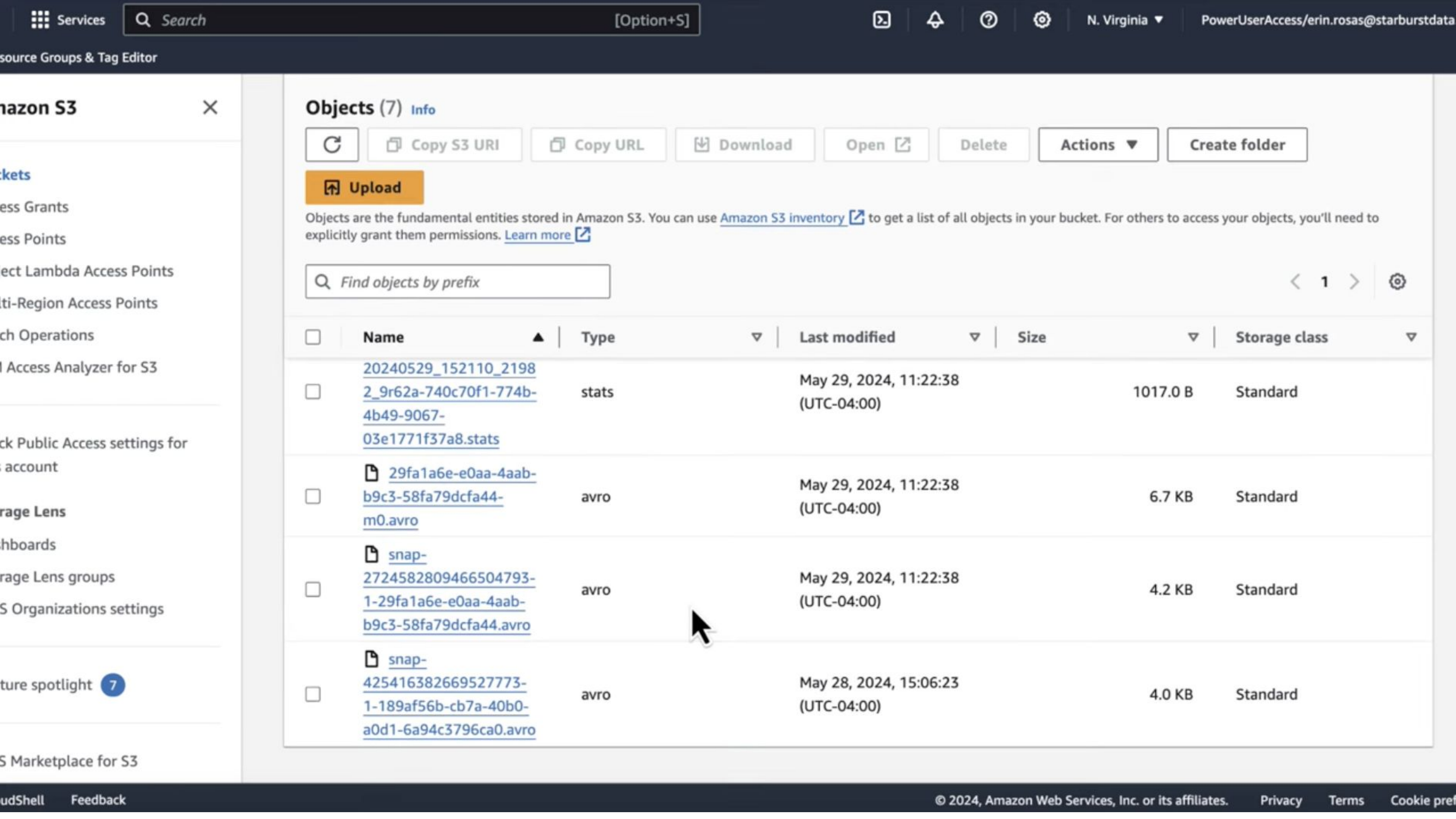
Amazon S3 > Buckets > starburst-tutorials > projects/ > tmp_erin_rosas_02152/ > aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/

aircraft-2f1f886045ed4fef9c6b27bf85e0eb6c/

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	data/	Folder	-	-	-
□	metadata/	Folder	-	-	-

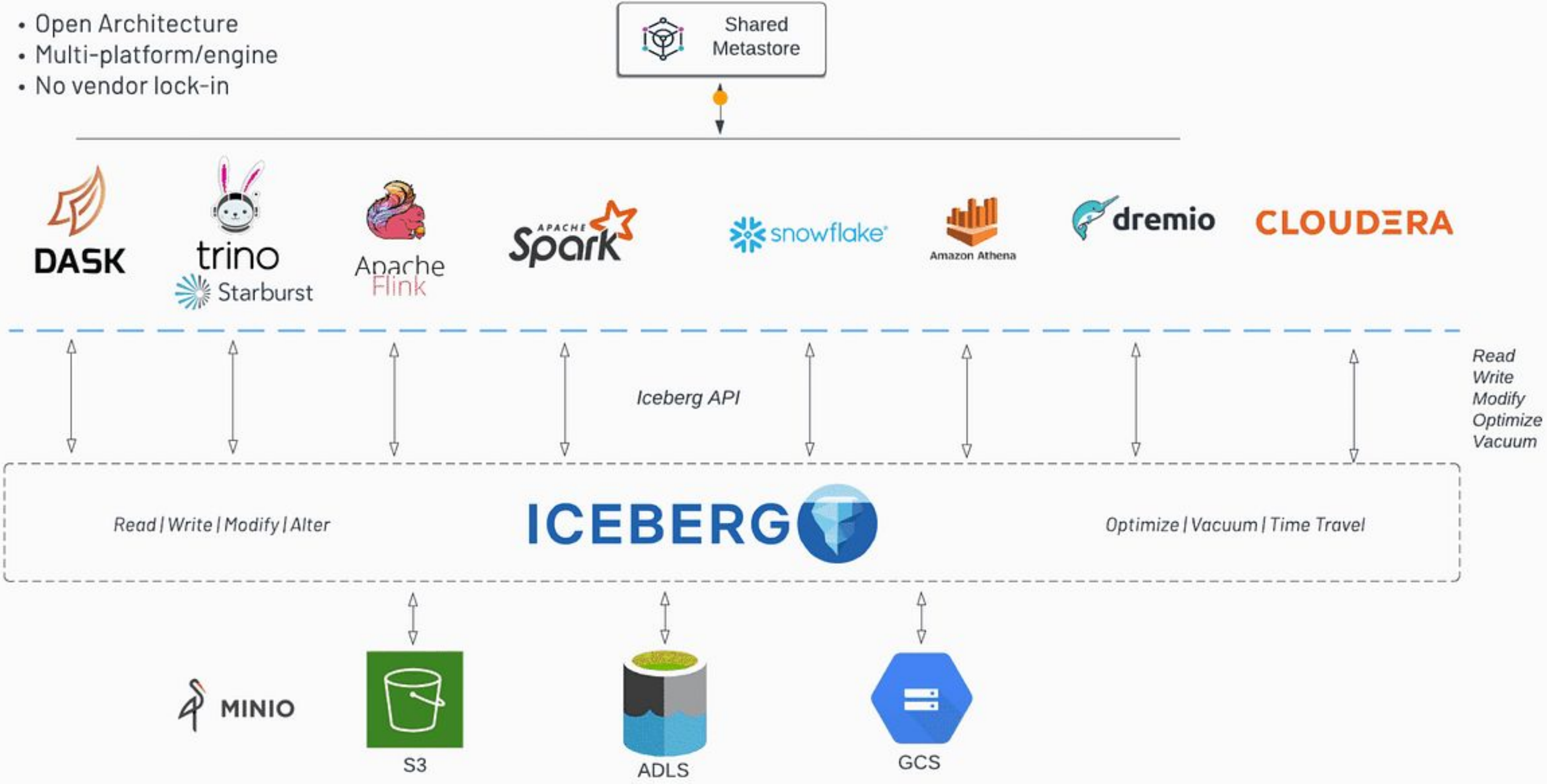


Objects (7) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	20240529_152110_21982_9r62a-740c70f1-774b-4b49-9067-03e1771f37a8.stats	stats	May 29, 2024, 11:22:38 (UTC-04:00)	1017.0 B	Standard
□	29fa1a6e-e0aa-4aab-b9c3-58fa79dcfa44-m0.avro	avro	May 29, 2024, 11:22:38 (UTC-04:00)	6.7 KB	Standard
□	snap-2724582809466504793-1-29fa1a6e-e0aa-4aab-b9c3-58fa79dcfa44.avro	avro	May 29, 2024, 11:22:38 (UTC-04:00)	4.2 KB	Standard
□	snap-425416382669527773-1-189af56b-cb7a-40b0-a0d1-6a94c3796ca0.avro	avro	May 28, 2024, 15:06:23 (UTC-04:00)	4.0 KB	Standard

- Open Architecture
- Multi-platform/engine
- No vendor lock-in



Shared Metastore

Iceberg API

Read
Write
Modify
Optimize
Vacuum

Read | Write | Modify | Alter

ICEBERG

Optimize | Vacuum | Time Travel

Apache Iceberg Integration and Compatibility

ICEBERG

■ Integration with Apache Spark

Capability to use Spark APIs for reading and writing data to Iceberg tables.

Two key catalogs in Spark :

org.apache.iceberg.spark.SparkCatalog: For external catalog services like Hive or Hadoop

org.apache.iceberg.spark.SparkSessionCatalog: Manages both Iceberg and non-Iceberg tables

■ Apache Flink Integration

Ideal for streaming data processing

Enables direct data streaming from various sources into Iceberg tables

Simplifies real-time data analytics

■ Integration with Presto and Trino

Known for fast data processing capabilities

Suitable for massive data querying and analysis

Dependency on external catalogs like Hive Metastore or AWS Glue for table management

Data Lake Compatibility

■ Apache Iceberg and Amazon S3 Integration

Description of Amazon S3 as a cloud storage service

Role of S3 in data lake architectures

Integration process using AWS Glue as the catalog service

Benefits: Enhanced querying capability and data consistency

■ Google Cloud Storage Compatibility

Advantages of Google Cloud for data lakes: Scalability and flexibility

Integration details: Using Iceberg with Google Cloud Storage

Querying options: Google’s BigQuery and standard SQL languages

■ Azure Blob Storage and Iceberg Integration

Overview of Azure Blob Storage: Designed for massive unstructured data

Benefits of integrating Iceberg with Azure

Outcome: Improved data access speed and reliability

Practical Exercise

- https://www.docker.com/

```
Terminal : docker version
docker info
clear
docker pull hello-world
docker images
docker +tab
docker run hello-world
docker ps
docker ps -a
```

Practical Exercise

- https://iceberg.apache.org/docs/nightly/

```
docker-compose up notebook
docker-compose up dremio
docker-compose up minio
docker-compose up nessie
```

http://127.0.0.1:8888/tree

http://127.0.0.1:9001/

http://127.0.0.1:9047/

Practical Exercise

- **localhost:9047**

Set the name of the source to “nessie”

Set the endpoint URL to “http://nessie:19120/api/v2”

Set the authentication to “none”

Navigate to the storage tab, by clicking on “storage” on the left

For your access key, set “admin”

For your secret key, set “password”

Set root path to “/warehouse”

Set the following connection properties:

“fs.s3a.path.style.access” to true

“fs.s3a.endpoint” to “minio:9000”

“dremio.s3.compat” to “true”

Uncheck “encrypt connection” (since our local Nessie instance is running on http)

Thank You

Dr. Firas

Author & Conference speaker