Data Intelligence Platform - Data

INTRODUCTION TO DATABRICKS

Kevin Barlow
Data Analytics Practitioner

datacamp

Why do organizations care about data management?

Protection and security

Confidence in data

Kinds of data

Structured

- Most common and understood
- Typical rows and columns
- **Examples:**
  - database tables
  - .csv
  - Parquet
  - Delta

id	name	occupation	location
1	Kevin	Data Scientist	California
2	Tom	Architect	Arizona
3	Sally	Lawyer	Texas
4	Tina	Surgeon	Florida
5	Joe	Engineer	New York

Kinds of data

Semi-structured

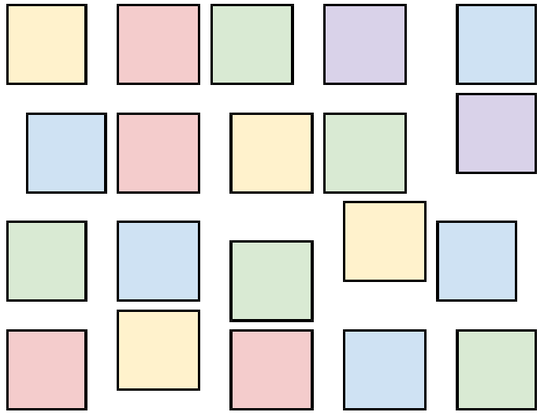
- Common with web-based devices
- Some structure, but more flexible in content
- **Examples:**
  - JSON
  - XML
  - HTML

```
{
  "people": [{
      "id": 1,
      "name": "Kevin",
      "occupation": "Data Scientist",
      "location": "California"},
    {
      "id": 2,
      "name": "Tom",
      "occupation": "Architect",
      "location": "Arizona"}]
}
```

Kinds of data

Unstructured

- Common with smart devices, cameras, etc.
- Little structure, information-rich
- **Examples:**
  - JPEG
  - PNG
  - MP4
  - PDF
  - DOC

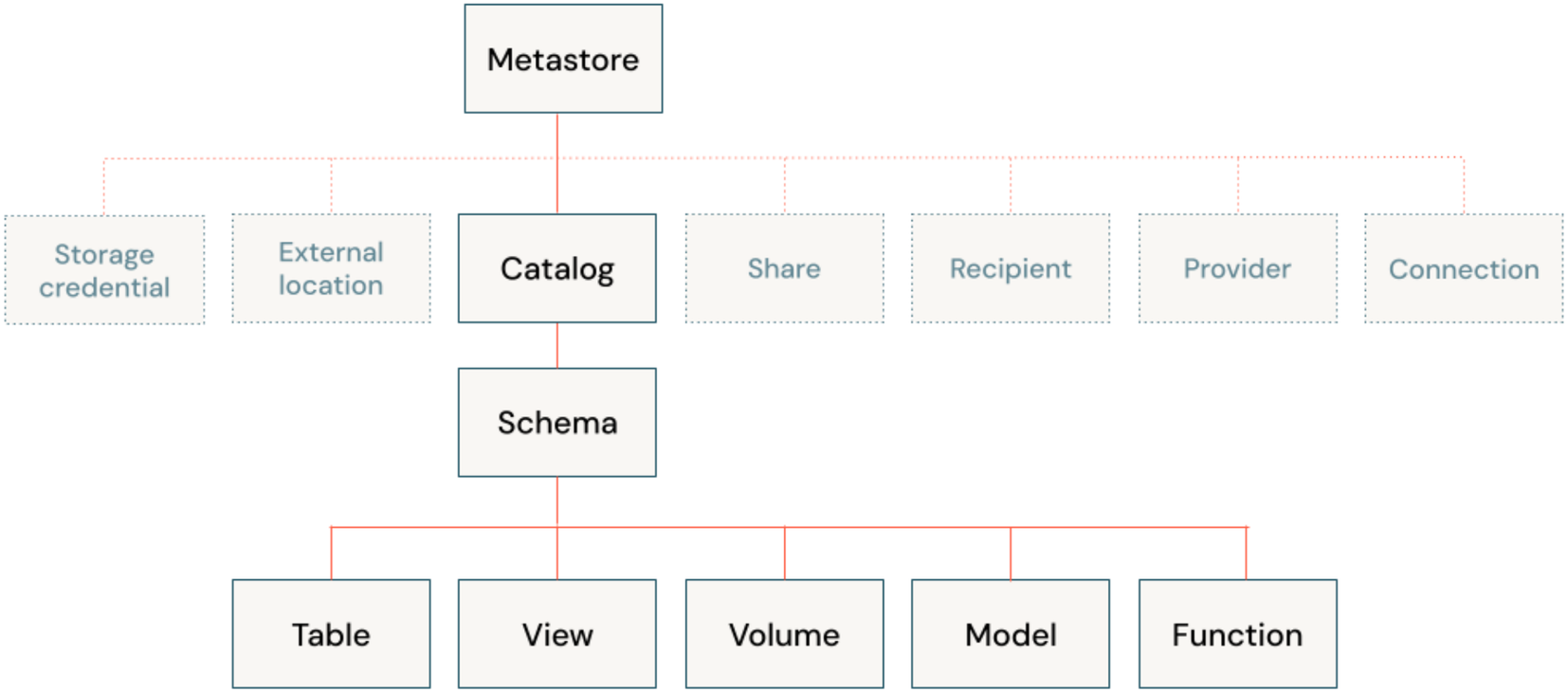


Delta

delta.io

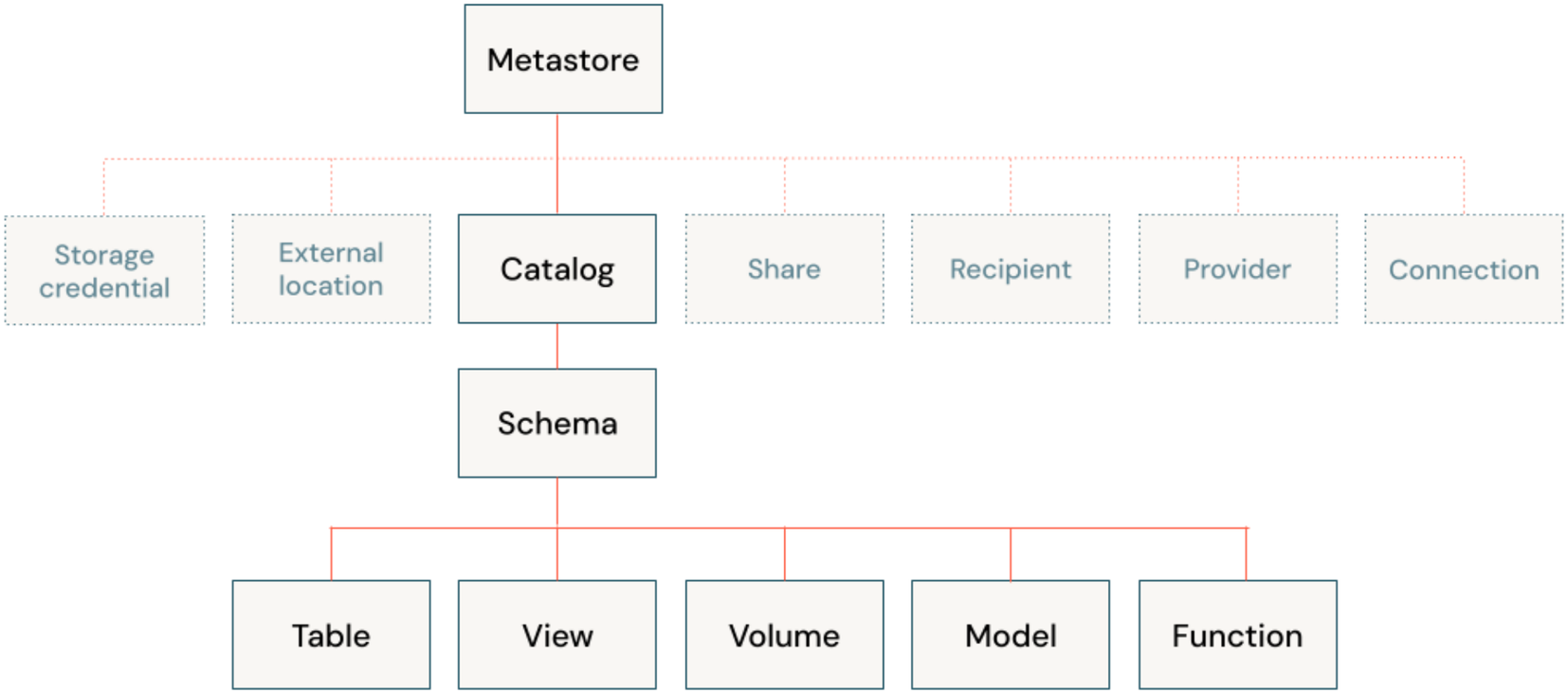
- Open-source storage format
- Collection of parquet tables
- JSON transaction log
- Fully ACID compliant
- Batch and streaming datasets

Unity Catalog



[1] https://docs.databricks.com/en/data-governance/unity-catalog/index.html#the-unity-catalog-object-model

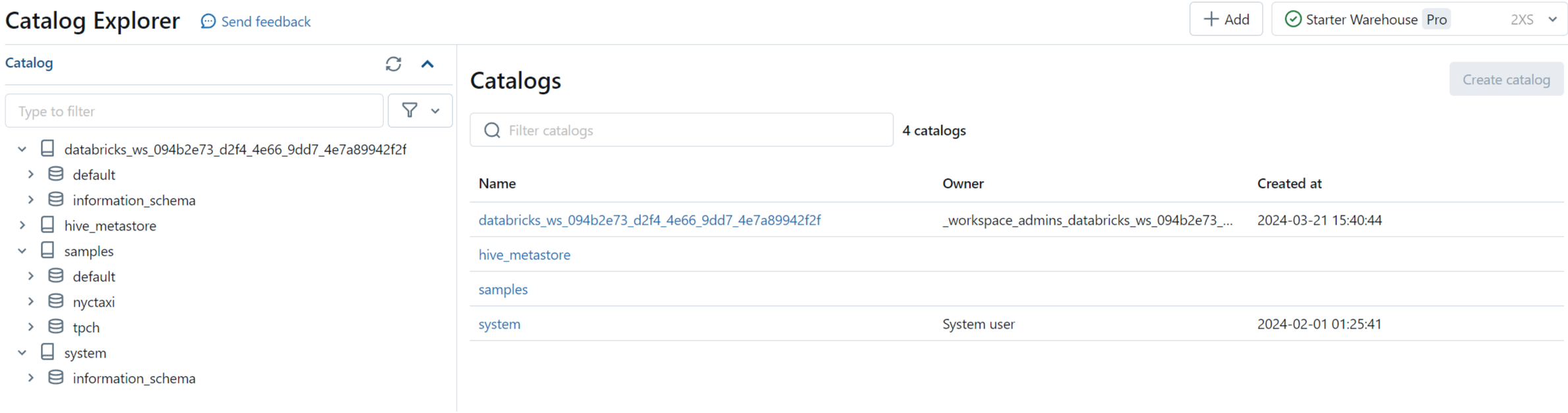
Unity Catalog




```
GRANT, SHOW, REVOKE, USE ...
```

Catalog Explorer

- Single location to explore all data assets
- UI to discover data
- Manage Unity Catalog permissions
- View data lineage and related assets



Let's practice!

INTRODUCTION TO DATABRICKS

Managing Data Catalogs

INTRODUCTION TO DATABRICKS

Kevin Barlow
Data Practitioner

datacamp

Let's practice!

INTRODUCTION TO DATABRICKS

Data Intelligence Platform - Compute

INTRODUCTION TO DATABRICKS

Kevin Barlow
Data Practitioner

Why do organizations care about compute?

Apache Spark

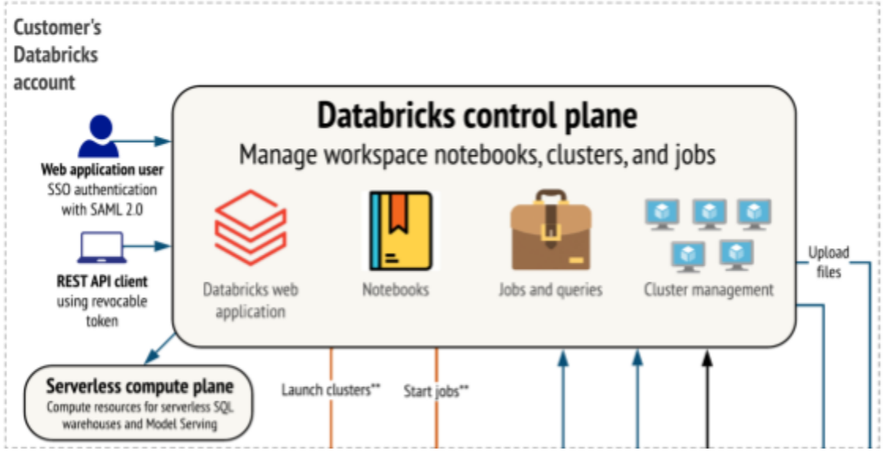
- Created by Databricks co-founders
- Open source framework
- Highly efficient distributed computing
- APIs for Python, SQL, Scala, R
- Great for all use cases:
  - data engineering to machine learning and business intelligence

Check out some of the **Apache Spark courses** on DataCamp!

Cluster Types

Classic

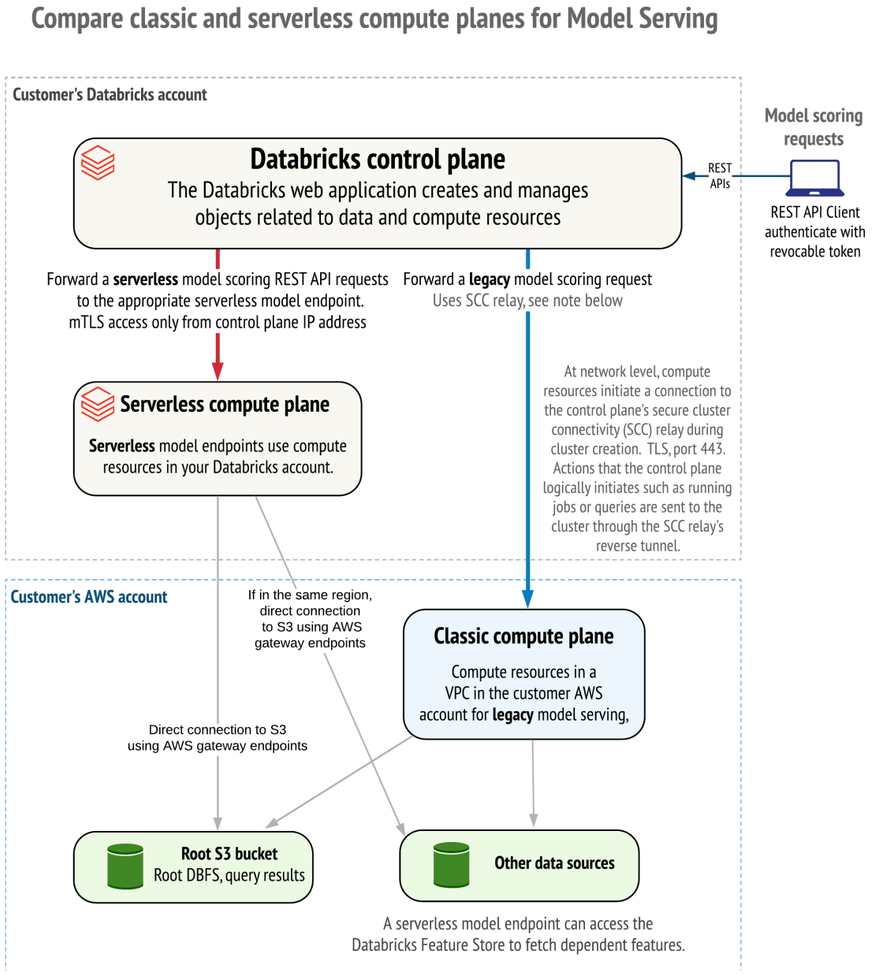
- Compute resources (virtual machines) are created in the Compute Plane
- Databricks provides configuration to your cloud
- ***Pros***: compute and security in your environment, leverage pre-existing compute pools, etc.
- ***Cons***: slow startup time



Cluster Types

Serverless

- Compute resources (virtual machines) are created in the Control Plane
- Databricks provides access to your users
- ***Pros***: Fast startup time, the latest and greatest feature, the fastest performance, Databricks improves performance over time
- ***Cons(?)***: compute not in your environment



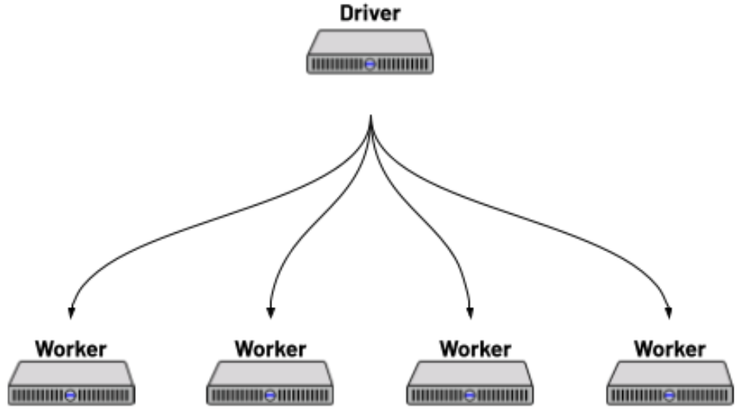
Single-node vs. Multi-node

Single-node

- Cluster with just a Driver Node
- Can still run Spark
- Can also run single-node frameworks (i.e., pandas)
- Great for smaller datasets

Multi-node

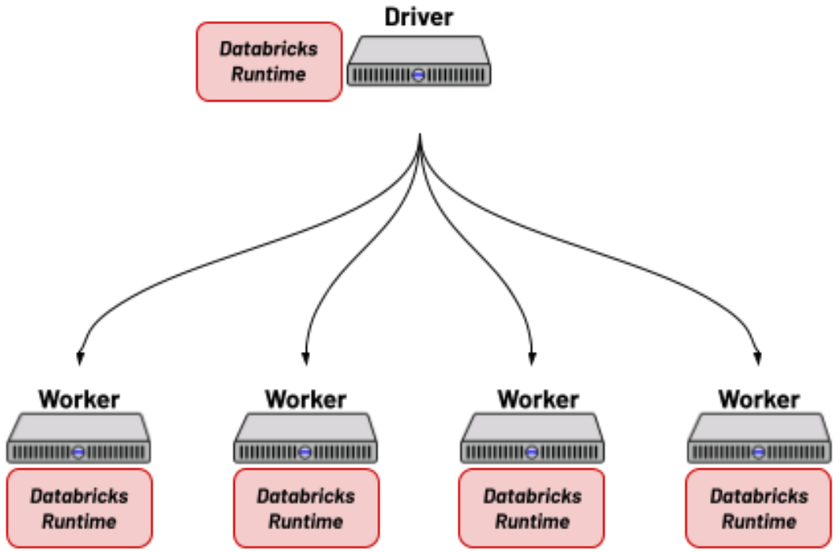
- Cluster with a Driver Node and one or more Worker Nodes
- Spark can distribute work across multiple nodes
- Great for larger datasets



Databricks Runtime

- Installed on every Databricks cluster
  - Optimized version of Apache Spark
  - Photon for faster SQL queries
  - Common libraries (e.g., pandas, dplyr, sci-kit learn)
  - Logic to connect with Databricks services

General recommendation: Use the most recent Long Term Support (LTS) version of the Runtime



Let's practice!

INTRODUCTION TO DATABRICKS