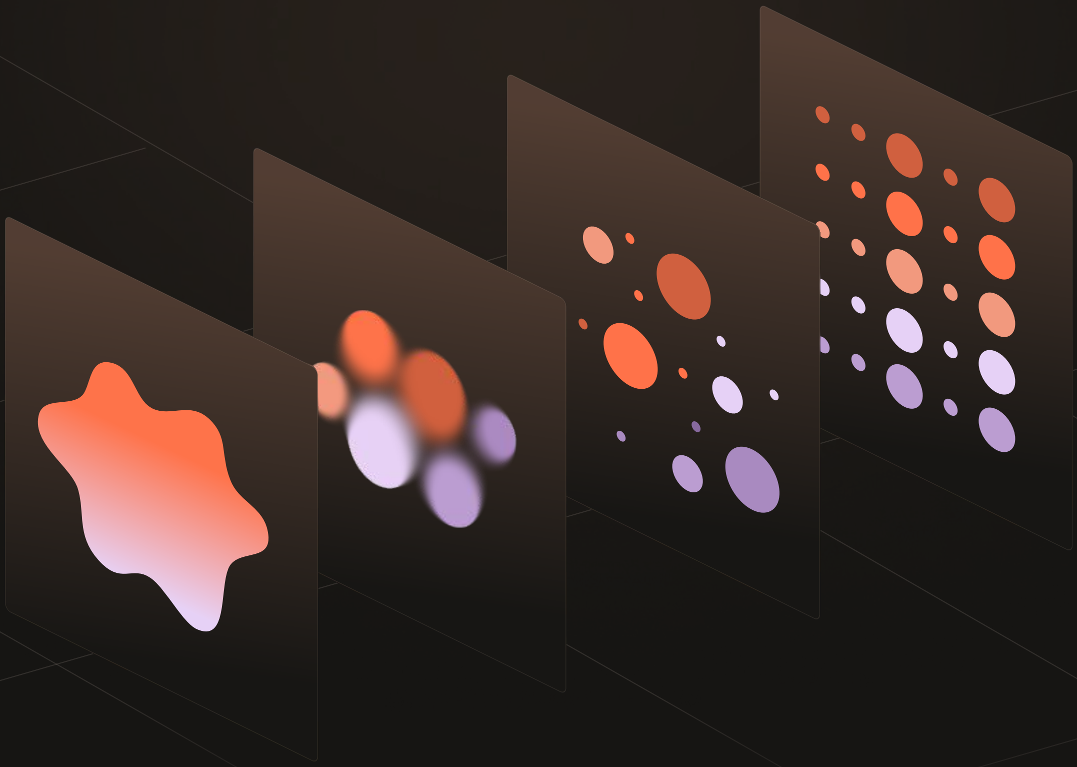
LanceDB

The AI-native Multimodal Lakehouse



By Jonathan Hsieh

in

Introduction

The Multimodal Lakehouse is the next-generation Lakehouse for cutting-edge multimodal AI teams. Unlike the last generation of data lakes, which typically store structured data or unstructured JSON blobs or logs, it was especially optimized for serving and computing over multimodal data, such as documents, video, audio, images, and sensor data.

The Multimodal Lakehouse delivers unparalleled scalability and a superior developer experience to accelerate model development over petabytes of multimodal training datasets, including feature engineering, storage management, data exploration and analysis, and accelerated training.

Key Benefits of The Multimodal Lakehouse

01

Unified Open Source Data Foundation for Multimodal Data

The Multimodal Lakehouse builds on the LanceDB and the Lance Data Format which is backed by a robust open source community and proven in production deployments (ByteDance, RunwayML, MidJourney). It comes with a rich ecosystem of integrations for Python, TypeScript or Java / Scala stacks.

02

Faster time-to-market

Equip AI engineers with seamless, scalable compute access and a frictionless developer experience to accelerate the AI development cycle – from feature engineering to training to evaluation. Before the Multimodal Lakehouse, AI engineers spent 80% of their time on data infrastructure and 20% on creative work. With this new tool, AI engineers can spend 90% of their time focusing on high creativity work.

03

Game-changing cost-performance ratio

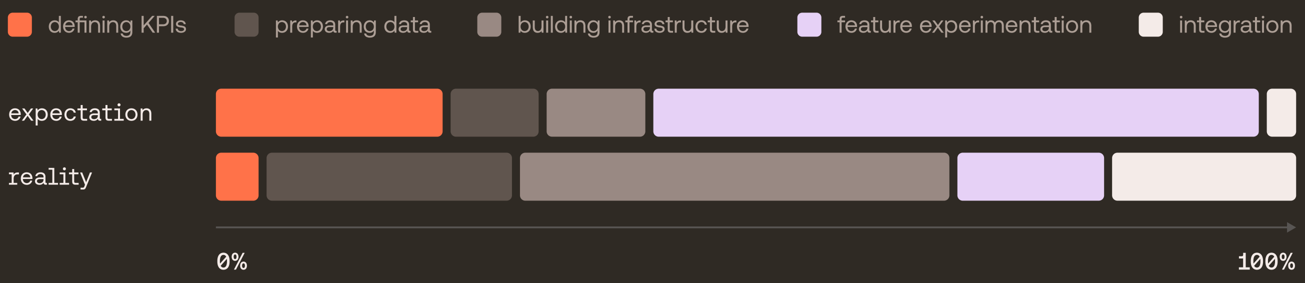
From cloud-native storage format to compute-storage separated serving infrastructure to heavily S3/GCS optimized index, the Multimodal Lakehouse can service low-latency, high-concurrency production traffic directly from inexpensive cloud storage and low-cost interruptable instances (spot/preemptable).

04

Low operational overhead

Stateless serving fleet, declarative pipeline definitions, and single source of truth on cloud object store eliminate the need for one-off ETL for data conversion into different silos and ensure that services are easily recoverable.

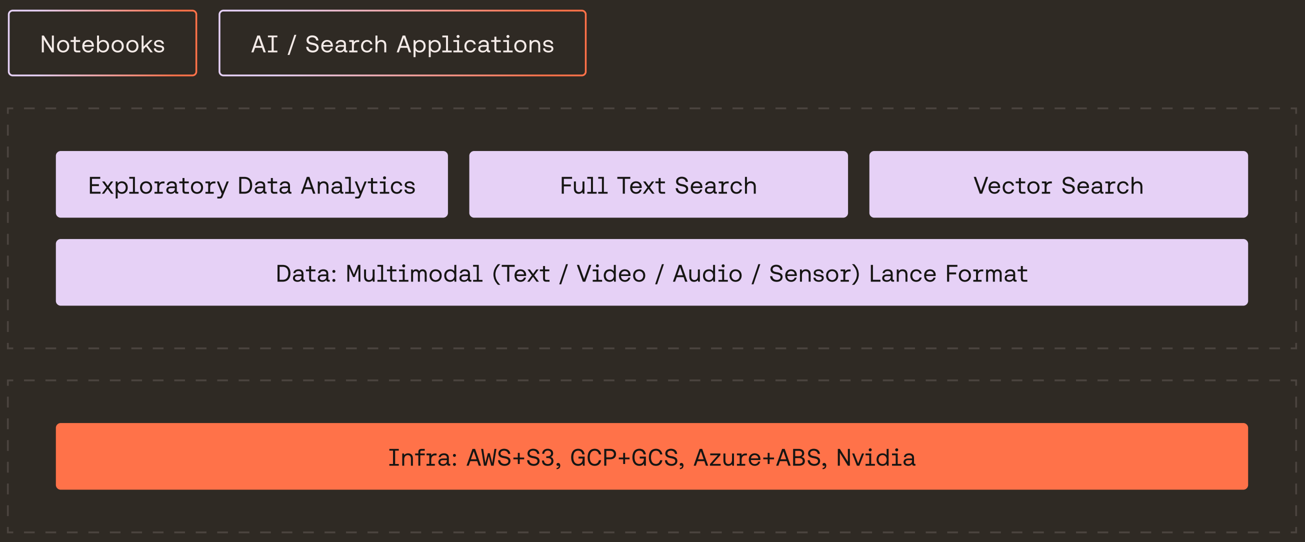
Feature creation effort allocation



Key Components of The Multimodal Lakehouse

The Multimodal Lakehouse, built on the foundations of LanceDB and the Lance format, boosts AI engineers' productivity and streamlines feature engineering and experimentation. It simplifies compute scaling for data processing and eliminates the need for multiple systems during exploratory data analysis, all while maintaining high priority for critical workloads like training.

The AI-Native Multimodal Lakehouse


Notebooks
AI / Search Applications
Exploratory Data Analytics
Full Text Search
Vector Search
Data: Multimodal (Text / Video / Audio / Sensor) Lance Format
Infra: AWS+S3, GCP+GCS, Azure+ABS, Nvidia

Data Management: Uniform Open Source Data Foundation

Built on the open-source Lance format, the Multimodal Lakehouse offers distinct features that are designed for a multimodal AI feature data layer:

- A modern columnar format with uncompromised random read performance
- Optimized layout for the mix of tiny fields and wide blob data
- Zero-copy schema evolution for feature management

It offers rich integrations into the Python, Java, TypeScript, and Rust ecosystems with an optimized I/O path to access data with variable modalities and sizes.

Other popular formats treat blob data as second class citizens. Lance and the Lakehouse on top of it are optimized for exploratory data analysis required for data curation to feed fine-tuning and model creation workloads.

	LANCE	WEB DATASET	ICEBERG / DELTALAKE	PARQUET
Random Access	✅ Fast	⚠️ Local Only	⚠️ Slow	⚠️ Slow
Fast Scan	✅ Yes	✅ Yes	✅ Yes	✅ Yes
Schema Evolution	✅ Inexpensive (Zero-copy)	❌ No Support	⚠️ Expensive (Copy)	⚠️ Expensive (Copy)
Multimodal	✅ Yes	✅ Yes	✅ Yes	✅ Yes
Search	✅ Yes	❌ No Support	⚠️ Slow	⚠️ Slow
Analytics	✅ Fast	❌ No Support	✅ Fast	✅ Fast

Compute: Feature Engineering

The Multimodal Lakehouse empowers AI engineers to execute feature engineering at scale and without having to dive deeply into distributed data processing infrastructure. The platform significantly simplifies feature generation and increases efficiency:

	THE MULTIMODAL LAKEHOUSE	RAY DATA	DATAFLOW, SPARK
Scale (Cores)	1-100K	1-5K (Actors)	1-100K
Data Processing Checkpoint	✅ Built in	❌ No Support	❌ No Support
Preemption	✅ Built in	❌ No Support	❌ No Support
Ingest Data	✅ Easy	✅ Easy	✅ Easy
Data and Schema Evolution	✅ Easy	✅ Easy	✅ Easy

By declaratively defining features using Python user defined functions (UDFs), the Multimodal Lakehouse orchestrates feature computation across a large-scale distributed environment, supporting filtering, checkpointing, preemption, and autoscaling.

```
import torch
import pyarrow as pa

@udf(version="blip", memory=16 * 1024**3, num_cpus=4, cuda=True)
def caption_udf(image:bytes) -> string:
    try:
        import the multimodal lakehouse

        # Load images and prepare for inputs
        image_stream = io.BytesIO(image)
        pil_image = Image.open(image_stream).convert("RGB")

        answer = generate_caption(pil_image)
        return answer
    except:
        raise ValueException("problem in UDF") from e

tbl.add_columns({"caption_blip": caption_udf})

tbl.backfill("caption_blip", batch_size=10)
```

The Multimodal Lakehouse provides key capabilities that eliminate 90% of the boilerplate infrastructure needed to manage data processing and feature development at scale.

Managing new features and data feature evolution is dramatically simplified. The Lance table format enables efficient zero-copy column addition and backfill using UDFs. For newly added data, the Multimodal Lakehouse jobs can be triggered externally or automatically.

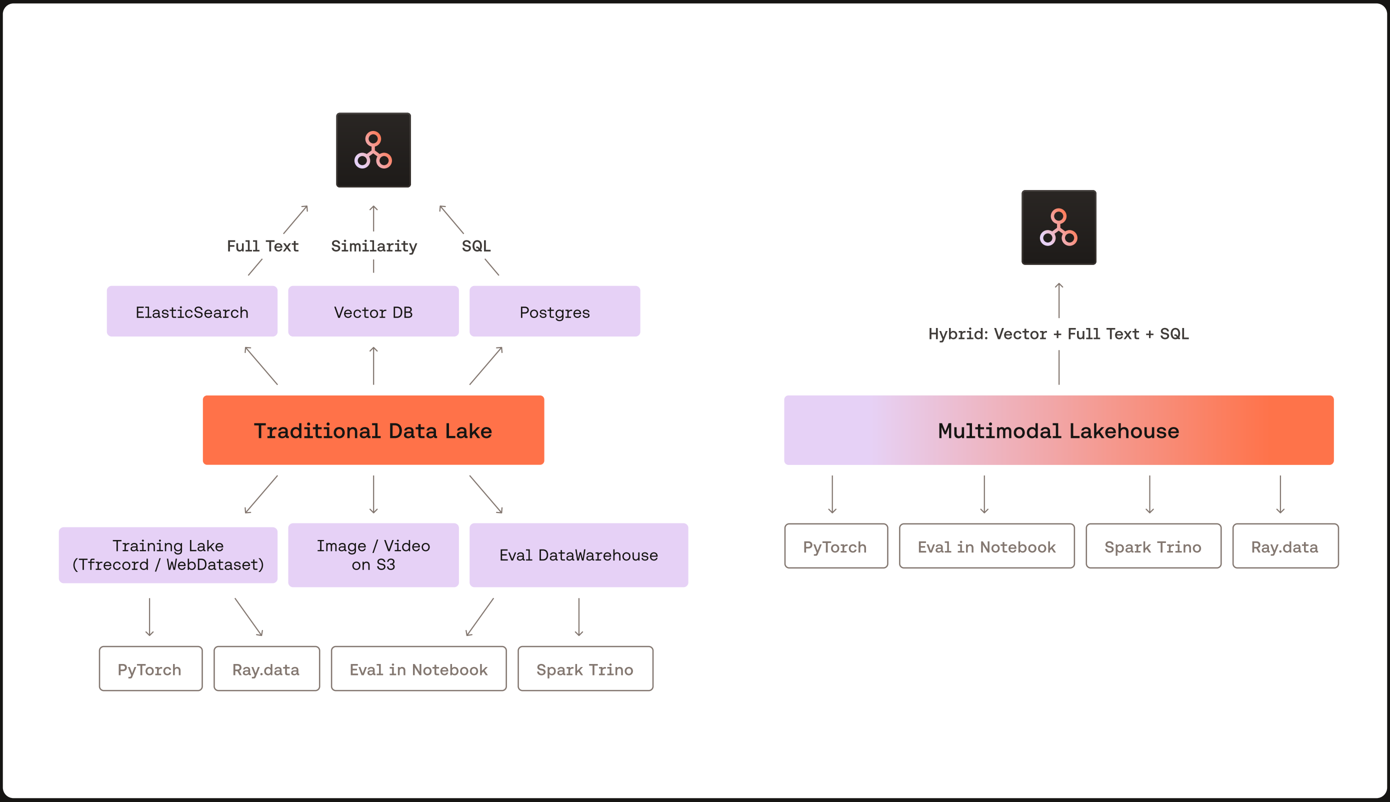
To scale to petabytes of data, the Multimodal Lakehouse dispatches jobs and can run on 100k's of cores and schedule GPUs using backend compute engines such as Ray, or runs on an existing Kubernetes cluster. These can autoscale so that your jobs are done quickly and so that you only pay for resources while they are in use.

Instead of having to time or schedule feature experimentation jobs behind time-critical production tasks, jobs can be launched with low-priority allowing automatic preemption and executed when your expensive committed-use resources (e.g. GPUs) are underutilized.

With the checkpointing feature, you don't have to worry about jobs losing progress. Instead of the expensive cost and time wasted having to rerun entire jobs, they can be paused and resumed when higher-priority training jobs preempt or if there are failures in processing.

Query: Exploratory Data Analysis and Search at Scale

The Multimodal Lakehouse empowers AI engineers to execute feature engineering at scale and without having to dive deeply into distributed data processing infrastructure. The Multimodal Lakehouse compute significantly simplifies feature generation and increases efficiency:



The Multimodal Lakehouse uses LanceDB as its query engine, to enable exploratory data analysis at ascale.

With its unparalleled search and random access capabilities are powered by a rich set of secondary indexes, including BTree, NGram, Full Text Search, and Vector Indices. The Multimodal Lakehouse eliminates the need to maintain multiple copies of the same data, each for one purpose and eliminates the need for expensive ETL to transform multimodal data into for different types of data exploration and evaluation.

The Compute-Storage-Separation Architecture, combined with a distributed cache service, allows users to scale the system for the traffic instead of the volume of data. LanceDB has been deployed in some of the most challenging production environments and serving business-critical applications.

LanceDB Performance in Production

Max QPS	20K+
Highest Throughput	10+ GB/s vector search traffic
Max Internal I/O	5M+ IOPS from cache fleet
Max Number of Rows	10 Billion Rows
Largest Dataset	O(10) Petabytes
Number of Tables Managed	O(10M) tables
GPU Indexing	3 Billion Vectors under 3 hours

Integration: Accelerate Training

The Multimodal Lakehouse provides additional value by optimizing the I/O path of training. By combining several technical advantages, it offers a best-in-class training data loader.

01

Fast random access

huffling and filtering during training are fast and inexpensive due to fast random access, which simplifies preparing data for training runs.

02

Large blob storage and API

Large blobs of data are stored in an efficient layout to reduce metadata. A Python File API is provided to open a File object on each blob without reading the blob in memory.

03

Named Views for Training

AI engineers can use SQL to express the training dataset as named views from joining the raw data and many other sources. Users can even customize the level of materialization of those views. The Multimodal Lakehouse PyTorch / Jax data loader can directly scan over those named views, so AI engineers can self-serve data management.

04

Distributed Cache Fleet

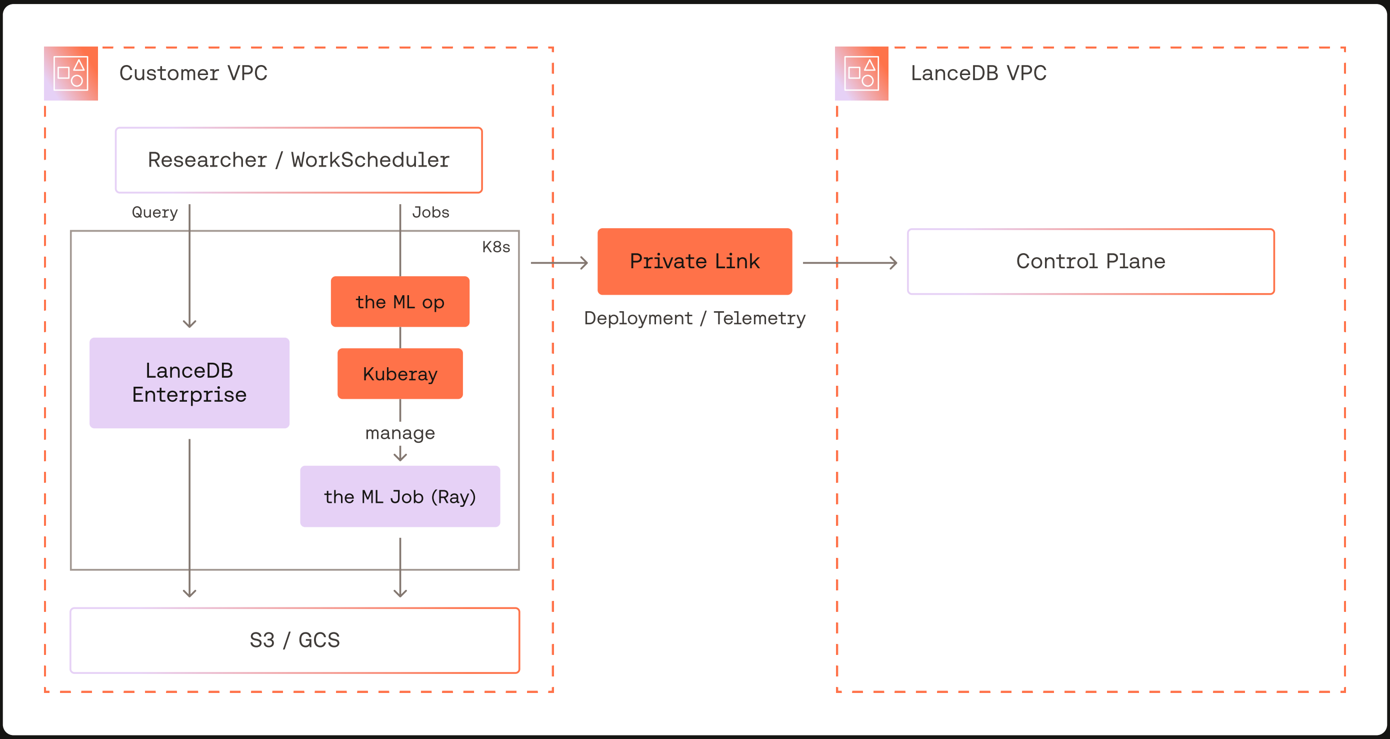
Originally built for LanceDB Enterprise, the Multimodal Lakehouse's cache fleet can serve 5M IOPS from NVMe SSDs, significantly improving throughput and reducing S3/GCS API cost during training.

Security

Security is paramount in a data lake. Access control, encryption, and auditing are crucial for protecting sensitive data. The Multimodal Lakehouse supports BYOC deployments (bring your own cloud) so you can control policies on how your data is protected. The tool relies on cloud storage (Amazon S3, Google GCS, Azure ABS) and inherits the access control, encryption, and auditing capabilities of those systems.

The Multimodal Lakehouse deploys helper services, which communicate workload telemetry data to LanceDB's control plane via:

- Private Service Connect (PSC) on GCP, or
- Private Link, a.k.a. VPC Interface service, on AWS, or
- User-owned auditable network proxy, that usually has two NICs, one in user VPC and one in LanceDB VPC



Trusted by Leading AI and Infrastructure Companies

Some leading AI companies have used the Lance format and LanceDB, including Databricks, ByteDance, RunwayML, Midjourney, Character AI, Harvey, Hex, Luma AI, UBS, Bosch, WeRide, and many more.

Conclusion

The Multimodal Lakehouse is a next-generation Data Lake designed to revolutionize how AI teams work with multimodal datasets. By leveraging the open-source Lance data format, it offers a unified, scalable solution for managing multimodal data like text, video, and images. With its superior performance, cost-effectiveness, and streamlined workflows, the tool empowers teams to accelerate their AI development cycles and bring innovations to market faster. Trusted by leading AI companies, the Multimodal Lakehouse is paving the way for the future of AI data management.

References

- LanceDB Enterprise Doc
- https://blog.lancedb.com/designing-a-table-format-for-ml-workloads/
- Lance v2: A columnar container format for modern data