WHITEPAPER

Choosing the Right Data Warehousing Solution

Microsoft Fabric

snowflake

Databricks

contactus@intmvens.com

Executive Summary

This whitepaper helps CXOs evaluate three leading modern data platforms **Microsoft Fabric**, **Snowflake**, and **Databricks**—for analytics, reporting, and AI/ML use cases. It provides detailed platform overviews, an in-depth technical comparison, scenario-based recommendations (100 GB / 100 reports and 1 TB / 200 reports), archival and large-report strategies, guidance on complex compute, and an assessment of AI/ML readiness. Secondary options such as Google BigQuery, Amazon Redshift, and Teradata are summarized as alternatives.

Introduction

Selecting the right data platform is a strategic decision that affects agility, cost, time-to-insight, and the organization's ability to scale analytics and AI initiatives. Choosing between vendor ecosystems (Azure vs multi-cloud vs open-source), data patterns (structured vs unstructured), and workload types (BI, real-time analytics, ML) is central to success.

This document emphasizes operational considerations CXOs need to prioritize: total cost of ownership (TCO), vendor lock-in risk, time-to-value, governance, and future-proofing for AI/ML workloads.

Platform Overviews

Microsoft Fabric

Microsoft Fabric is a unified SaaS analytics platform that brings together OneLake (a logical data lake), managed warehouses and lakehouses, Power BI, ETL/dataflows, and real-time eventing into a single analytics experience.

Architecture & Core Components

- **OneLake**: Logical single-store for files and tables with an enterprise catalog.
- **Managed Warehouses / Lakehouses**: SQL-optimized warehouses for BI and lakehouses for open-format storage and analytics.
- **Power BI Native Integration**: Semantic layer, Direct Lake, dataset acceleration, and model governance.
- **Data Integration**: Built-in pipelines and connectors for common enterprise sources.
- **Real-time & Eventing**: Data Activator + streaming ingestion to support alerting and near-real-time dashboards.

Operational Characteristics

- **Storage**: Abstracted onto ADLS Gen2; supports columnar formats and lifecycle policies.
- **Compute**: Capacity SKU model (F / P) where Microsoft manages compute allocation and scaling within capacity limits.
- **Governance**: Tight Purview integration for cataloging, lineage, and policy enforcement.
- **Security**: Azure AD, RBAC, encryption (including customer-managed keys), and enterprise compliance through Azure.

Strengths & Typical Fit

- Fastest time-to-value for organizations already invested in Microsoft 365 and Power BI.
- Simplifies governance and user access for enterprise BI initiatives.

Limitations

- Azure-bound; limited multi-cloud portability. Capacity management requires planning for peak concurrency.

Snowflake

Snowflake is a cloud-native, multi-cloud data platform built around a separation of storage and compute. It focuses on high-performance SQL analytics, secure data sharing, and ease of operations.

Architecture & Core Components

- **Storage Layer**: Durable object storage in the chosen cloud provider with proprietary micro-partitioning and automatic clustering.
- **Compute Layer (Virtual Warehouses)**: Independent clusters that run queries and can be auto-scaled or suspended to control cost.
- **Control Plane**: Manages metadata, metadata caches, and the coordination between compute and storage.
- **Data Sharing / Marketplace**: Secure, zero-copy data sharing and a marketplace for data products and third-party apps.

Operational Characteristics

- **Storage**: Billed separately; optimized for compression and pruning.

- **Compute**: Credit-based, per-second billing with resource monitors to manage budgets.
- **Governance**: Object-level privileges, row/column-level security patterns, dynamic data masking.
- **Security & Compliance**: Broad compliance, encryption, and enterprise key management options.

Strengths & Typical Fit

- Strong multi-cloud support and excellent concurrency for BI workloads.
- Low operational burden for analytics teams focused on SQL and data sharing across units or partners.

Limitations

- Less native ML/AI tooling; typically integrated with Databricks, SageMaker, or custom ML stacks.

Databricks

Databricks is a Lakehouse platform built on Apache Spark with Delta Lake for ACID transactions over data lakes. It is optimized for large-scale data engineering, streaming, and production ML workloads.

Architecture & Core Components

- **Delta Lake**: Transactional layer over object storage providing ACID guarantees.
- **Databricks Runtime**: Optimized Spark runtime with performance and connector enhancements.
- **Notebooks & Collaboration**: Interactive notebooks and collaborative workspaces with multi-language support (Python/Scala/SQL/R).
- **MLOps**: MLflow, Model Registry, Feature Store, and integrated serving for model lifecycle management.

Operational Characteristics

- **Storage**: Uses cloud object storage (S3/ADLS/GCS); data layout and compaction are important for performance.
- **Compute**: Cluster-based Spark compute with autoscaling and GPU support for deep learning.
- **Governance**: Unity Catalog for centralized governance, lineage, and access control.

Strengths & Typical Fit

- Best-in-class for ML/AI, streaming, and complex transformations at scale.
- Flexibility for multi-language engineering teams and open-source ecosystems.

Limitations

- Higher operational complexity for BI-only teams; costs can escalate without cluster discipline.

Other Platforms (brief)

- **Google BigQuery**: Serverless, highly elastic, pay-per-query; excellent for ad-hoc SQL analytics and quick scaling.
- **Amazon Redshift (including Serverless)**: AWS-native with strong integration into the AWS ecosystem and incremental improvements for concurrency and serverless management.
- **Teradata / Vertica / Firebolt**: Purpose-built analytic databases that excel in specific latency and concurrency scenarios for enterprise deployments.

High-Level Comparison (executive view)

Requirement	Fabric (Azure)	Snowflake (Multi-cloud)	Databricks (Lakehouse)
Time-to-value (BI)	Very High	High	Medium
Multi-cloud portability	Low	Very High	High
AI/ML readiness	Medium	Medium	Very High
Governance & compliance	Very High	High	High
Predictable cost for BI	Medium	Medium/High	Low (unless optimized)
Required operational skill	Low	Medium	High

Technical Feature Comparison

The table below compares technical capabilities and operational behaviors that matter for deployment, cost, and long-term platform strategy.

Dimension	Microsoft Fabric	Snowflake	Databricks
Architecture Model	Unified SaaS (OneLake + managed compute) where Microsoft controls orchestration and capacity.	Multi-cluster shared data architecture with clear separation of compute/storage and a managed control plane.	Lakehouse on object storage with Delta Lake transactional layer and Spark compute orchestration.
Storage Engine & Formats	Open columnar formats (Parquet/Delta-like) on ADLS Gen2 beneath OneLake; encourages open formats.	Proprietary micro-partitioned storage optimized for pruning and compression; object store back-end.	Delta Lake (Parquet + transaction logs), supports ACID and open access to files.
Compute Model & Elasticity	Capacity SKUs; elasticity managed within purchased capacity limits. Easy for predictable BI workloads.	Virtual warehouses — independent, right-sized compute for each workload with per-second billing. Scales well for concurrency.	Cluster-based Spark with autoscaling; strong for distributed compute and GPU workloads but requires operational control.
Query Engine & Performance	SQL engine tuned for semantic models and integrations with Power BI; performance enhancements for Direct Lake queries.	Highly optimized Snowflake SQL engine; result caching, pruning, automatic micro-partition maintenance accelerate queries.	Spark SQL with Databricks runtime optimizations (AQE, caching, Tungsten improvements). Best for complex transformations.
Transactions & Consistency	ACID support for lakehouse operations via managed layers; suitable for transactional-ish analytics.	Strong consistency for SQL workloads; Time Travel and cloning for point-in-time recovery.	Delta Lake provides ACID transactions, upserts, and concurrent read/write guarantees.
Streaming & Real-time	Native streaming & Data Activator for eventing and alerts;	Snowpipe provides near-real-time ingestion; streaming analytics are typically	First-class streaming via Spark Structured Streaming and Auto

Dimension	Microsoft Fabric	Snowflake	Databricks
	suitable for near-real-time dashboards.	architected via partner tools.	Loader; low-latency pipelines supported.
Metadata, Catalog & Lineage	Centralized metadata, Purview integration (lineage, catalog, policies).	Central metadata/catalog; tagging and governance features; integrates with external catalog tools.	Unity Catalog centralizes metadata, lineage, and fine-grained access control across workspaces.
Governance & Security	Strong enterprise governance via Purview; Azure-native IAM, conditional access, encryption with CMK support.	Enterprise-grade security, RBAC, dynamic masking, BYOK, audit logs; robust compliance posture.	Cloud IAM integration, role-based controls via Unity Catalog, encryption at rest/in transit; enterprise compliance options.
Data Sharing & Ecosystem	Optimized for in-tenant sharing and Power BI distribution; external sharing via Azure services.	Native secure data sharing with zero-copy sharing and marketplace for data providers.	Delta Sharing protocol and third-party integrations for cross-platform data exchange.
Developer Experience	Low-code for analysts (Power Query) and code-first options (notebooks); strong Power BI UX.	SQL-first with Snowpark for modern applications; broad partner ecosystem.	Notebook-first collaborative development, multi-language, strong CI/CD and SDKs for production ML.
Observability & Ops	Integrated monitoring; Azure Monitor + tenant telemetry; platform-managed operational tasks.	Admin UI, resource monitors, query profiling; integrates with external observability stacks.	Rich telemetry for clusters/jobs; requires active ops for cost & performance management.
Pricing Model & Cost Controls	Capacity SKUs (predictable for steady workloads); needs capacity planning for peaks.	Consumption-based credits per second; resource monitors and auto-suspend control costs.	Consumption (DBUs + cloud compute); cost discipline needed around cluster

Dimension	Microsoft Fabric	Snowflake	Databricks
			lifecycles and GPU usage.
Backup, DR & Retention	OneLake lifecycle policies, versioning in lakehouse layers; relies on Azure backup mechanisms for DR.	Time Travel, Fail-safe, and external archival recommended for long-retention.	Delta time travel and object-store lifecycle; recommended to combine with cloud-region replication for DR.
AI/ML & MLOps	Integrates with Azure ML and Synapse ML; suitable for moderate ML workload productionization.	Snowpark + external model tooling; good for SQL-centric ML workflows and inference.	Comprehensive MLOps stack (MLflow, Feature Store, Model Registry), GPU-enabled training and distributed serving.

Scenario-Based Recommendations

Scenario A — 100 GB Data + 100 Reports

Requirements: Primarily BI workloads, daily refresh, 20–50 concurrent users, moderate transformations.

- **Microsoft Fabric**: Best if organization uses Power BI; low ops and quick deployment. Use PPU or small dedicated capacity for consistent concurrency.
- **Snowflake**: Strong alternative for multi-cloud shops; use a small warehouse per workload and auto-suspend to control costs.
- **Databricks**: Consider only if heavy transformations or streaming/ML features are required.

Recommendation: Choose Fabric for Microsoft-first shops; Snowflake for cross-cloud SQL-centric teams.

Scenario B — 1 TB Data + 200 Reports

Requirements: Larger datasets, more concurrency, hourly refreshes, some analytic aggregates pre-computed.

- **Microsoft Fabric**: Works with partitioning and higher capacity SKUs; consider dataset design to minimize full model refreshes.

- **Snowflake**: Excellent at this scale; separate warehouses for ETL and BI workloads with caching and materialized views.
- **Databricks**: Strong when ETL and ML pipelines are complex; use Delta Lake + Databricks jobs for transformations and serve aggregated tables to BI.

Recommendation: Snowflake for easier scaling & concurrency; Databricks for heavy engineering+ML use cases. Fabric is viable when Power BI integration and governance are primary concerns.

Archival & Handling Very Large Reports (>10 GB)

Key patterns:

- **Cold storage**: Move historical data to object storage (ADLS/S3/GCS) and keep only active data in the warehouse.
- **Partitioning & Aggregation**: Pre-aggregate or partition large tables to reduce scan sizes for reporting.
- **Incremental & Direct Lake Patterns**: Use Direct Lake (Fabric) or external tables (Snowflake/Databricks) to query large datasets without fully importing them into a dataset.
- **Materialized Views & Result Caching**: Use materialized views, result caching, or summary tables so reports do not re-run expensive queries frequently.

For reports that exceed 10 GB in output size, consider exporting as paginated reports or precomputed files (Parquet/ORC) for download rather than interactive query results.

Complex Compute & AI/ML Readiness

Considerations for CXOs:

- **Model Lifecycle Management**: Databricks provides the most complete native MLOps stack; Snowflake and Fabric rely on integrations for full MLOps.
- **Distributed Training & GPUs**: Databricks supports distributed GPU training; Snowflake and Fabric integrate with external GPU-enabled services.
- **Feature Stores & Model Serving**: Databricks includes a Feature Store and Model Registry. For Snowflake, use Snowpark + external registries. Fabric integrates with Azure ML for model serving.

Recommendation: If AI/ML productionization and experimentation velocity are strategic priorities, Databricks should be a core component of the platform strategy;

Snowflake can serve feature and model scoring needs alongside Databricks or other training platforms.

Migration & Implementation Notes

- **Data Movement**: For non-Azure migrations, stage data in cloud object stores and use the platform's bulk load utilities.
- **Semantic Layer & Reporting**: Design a semantic layer (Power BI datasets / Snowflake logical models) to reduce dataset proliferation and improve governance.
- **Cost Management**: Implement resource monitors, auto-suspend policies, lifecycle rules, and tagging to attribute costs and prevent surprises.
- **Governance**: Centralize metadata and lineage using Purview (Fabric), Unity Catalog (Databricks), or an enterprise data catalog for Snowflake.

Decision Framework

Ask these questions before choosing a platform:

1. Are you primarily Microsoft/Power BI users? If yes, Fabric accelerates outcomes.
2. Do you require multi-cloud portability and secure data sharing? Snowflake is purpose-built.
3. Is AI/ML a strategic priority with high experimentation velocity? Databricks leads here.
4. What is your operational tolerance for platform complexity and cost-management discipline?

Use a weighted decision matrix (Ecosystem Fit, TCO, Time-to-Value, Governance, AI-readiness) to quantify choices.

Conclusion

There is no one-size-fits-all. For BI-first, Microsoft Fabric offers a rapid, governed path when Power BI and Azure are core to the business. Snowflake offers the most straightforward path to multi-cloud analytics and secure data sharing. Databricks is the choice for organizations that prioritize large-scale data engineering, streaming, and production-grade ML.

References & Further Reading

- Microsoft Fabric: https://learn.microsoft.com/en-us/fabric/
- Snowflake: https://www.snowflake.com/
- Databricks: https://www.databricks.com/
- Google BigQuery: https://cloud.google.com/bigquery
- Amazon Redshift: https://aws.amazon.com/redshift/