

When selecting a data platform vendor, assessing their expertise in Databricks versus Snowflake is often a pivotal decision. Both platforms have revolutionized modern data architectures, but they cater to different paradigms — the lakehouse and the cloud data warehouse respectively. Adding in the broader ecosystem, such as Azure's Microsoft Fabric and Synapse Analytics, the lines get even more nuanced. This post will walk through how to rigorously compare Databricks skills versus Snowflake skills in a vendor, focusing on delivery depth, cloud implementation experience (Azure & AWS), governance, data lineage, and semantic modeling capabilities.

Understanding the Ecosystem: Lakehouse vs Warehouse vs Data Lake

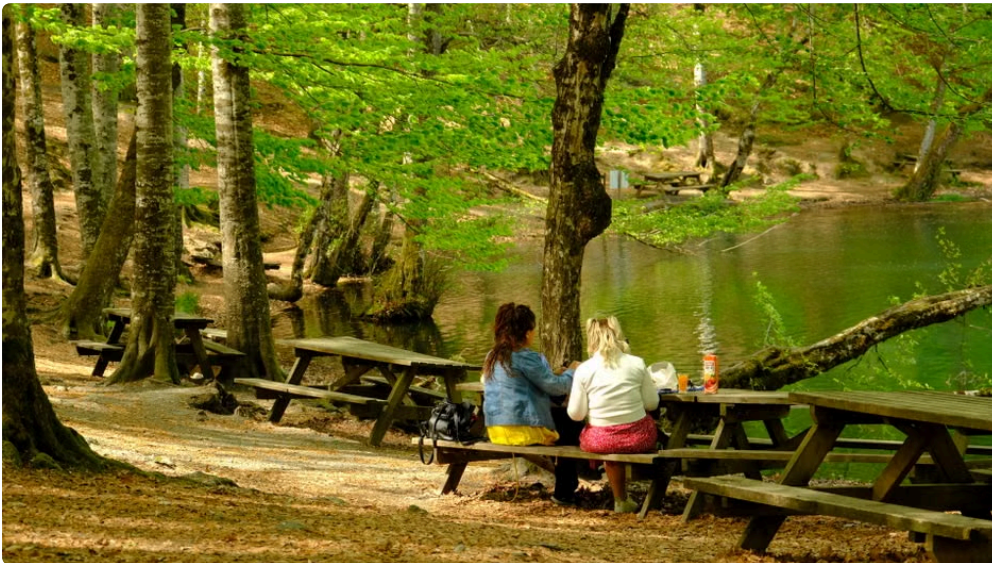
Before diving into skills comparison, foundational clarity on the platforms is critical. The core of vendor competency lies not only in platform-specific features but in how they architect data solutions around these foundational concepts.

What is a Data Lake?

A data lake is a centralized, vast repository designed to store raw and varied data formats in their native forms — structured, semi-structured, and unstructured. It allows users to store data at scale cheaply but often suffers from lack of governance, schema enforcement, and agility for analytics.

What is a Data Warehouse?

A cloud data warehouse like Snowflake is a highly optimized, schema-centric environment designed primarily for performant SQL analytics over structured data. It enforces schema-on-write, manages metadata, and provides built-in governance and performance tuning features.



What is a Lakehouse?

The lakehouse combines the best of data lakes and warehouses. Databricks pioneered this concept with its Delta Lake technology: a transactional storage layer on top of a data lake, which enables ACID transactions, schema enforcement, and BI-friendly performance without copying or extracting data to a separate warehouse.

Platform	Data Paradigm	Schema Enforcement	Key Strength	Typical Use Case
Databricks Lakehouse	(Data Lake + Warehouse features)	Schema-on-read & write with Delta Lake ACID	Unified analytics, machine learning, streaming	Data science, streaming ETL, machine learning pipelines
Snowflake Cloud Data Warehouse (DWH)	Strict schema-			

on-write Simple SQL analytics, performance tuned warehouse BI dashboards, structured data reporting Azure Synapse & Fabric Hybrid (Warehouse + Lakes + Semantic Layer) Depends on workload (SQL Pools enforce schema) Enterprise data integration & analytics Enterprise-scale analytics, data integration

Comparing Databricks Expertise vs Snowflake Expertise in a Vendor

How does one validate a vendor's claimed expertise in Databricks and Snowflake? This section covers critical dimensions you need to probe, backed by real-world expectations and red flags to watch for.

1. Depth of Delivery Experience

- **Databricks:** Look for vendors who have implemented end-to-end pipelines involving *Delta Lake* for streaming and batch ETL, constructed ML pipelines within Databricks' collaborative notebooks, and integrated Databricks with Azure Active Directory or AWS IAM for governance. Delivery depth manifests in use of CI/CD pipelines for Databricks notebooks, infrastructure-as-code (IaC) for cluster management, and automated data quality testing embedded into pipelines. Vendors should also show competency in Spark SQL and Python/Scala optimizations.
- **Snowflake:** Expertise is shown by vendors who have created complex multi-warehouse sizing and auto-scaling strategies, designed zero-copy cloning for development/test cycles, and implemented row-level security, masking policies for governance. Demonstrated Snowflake skills include optimizing query performance with clustering keys, managing Time Travel and Fail-safe periods, and implementing continuous data ingestion using Snowpipe or third-party ETL tools. CI/CD and Terraform-based infrastructure automation around Snowflake objects are key indicators of maturity.

2. Cloud Implementation: Azure and AWS Experience

Despite [suffolknewsherald](#) being cloud-agnostic platforms, both Databricks and Snowflake differ in how they are implemented on Azure versus AWS, critical when comparing vendor skills for your stack.



- **Databricks on Azure:** Vendors should know how to leverage Azure Databricks alongside Azure Synapse Analytics and Microsoft Fabric – integrating Lakehouse data with semantic models in Purview and Power BI.

Real-world experience includes configuring Databricks clusters with Azure virtual networks (VNETs), managing workspace security with Azure AD, and optimizing cost by leveraging Azure Blob Storage.

- **Databricks on AWS:** Look for mastery of configuring Databricks on AWS with Amazon S3 as the underlying data lake, implementing IAM roles for security, and integrating with Glue Data Catalog for metadata management. Experience in integrating Databricks with other AWS analytics services like Athena or Redshift Spectrum enhances capabilities.
- **Snowflake on Azure:** Validate knowledge of Snowflake's integration with Azure Blob Storage and managed identity authentication, alongside usage patterns combining Snowflake with Azure Data Factory and Microsoft Purview governance.
- **Snowflake on AWS:** Deep familiarity with Snowflake's use of S3 storage, IAM roles, Glue integration, and AWS KMS customer-managed keys reflects maturity.

3. Governance, Lineage, and Semantic Layer Proficiency

A red flag I always watch for is a "data platform expert" who glosses over governance, lineage, and semantic modeling as afterthoughts. In reality, these are foundational to trustworthy data delivery and user trust.

- **Data Governance:** Vendors should demonstrate how they implement role-based access control (RBAC), data masking policies, and dynamic data security. For Databricks, this means experience with Unity Catalog for fine-grained access control. Snowflake experts should understand object-level privileges, external tokenization, and masking policies.
- **Lineage:** Lineage is often neglected, yet a must-have. Vendors should clearly articulate how they track data provenance — at minimum leveraging features like Databricks' Unity Catalog lineage tracking or Snowflake's access history combined with integrated third-party data catalog tools such as Microsoft Purview or Collibra. Asking "Where does lineage live and who owns it?" is a vital test.
- **Semantic Modeling and Metadata Management:** Ideally, vendors build semantic models (business logic definitions, KPIs, calculated columns) in a re-usable, governed manner rather than ad hoc transformations buried downstream. On Azure, integration with Microsoft Fabric's semantic layers or Power BI datasets shows mastery. In the Databricks world, leveraging Delta Live Tables for clean, reusable datasets and ML feature stores is a signal of next-level maturity. Vendors overly focused on just raw ETL pipelines without semantic considerations are likely to produce brittle architectures.

How to Validate a Vendor's Stack and Skills

Now, how do you confirm your vendor's Databricks and Snowflake skills beyond marketing speak? Here are actionable steps for stack validation:

1. **Review Past Delivery Artifacts:** Ask the vendor for code samples, notebook walkthroughs, pipeline architecture diagrams, and CI/CD pipeline snapshots. Does their work show proper use of IaC (Terraform, ARM templates), unit tests, integration tests, and automated deployments? Lack of CI/CD is a red flag for lakehouse projects.
2. **Lineage and Governance Demo:** Request a live demo or screenshots showing how lineage is tracked, how governance policies are implemented, and how users are authorized. If they dodge semantic layer questions, beware.
3. **Reference Checks Focused on Depth:** Go beyond the pilot or POC and ask about vendor support after go-live. Were there incidents related to data quality or performance? What processes did the vendor implement to remediate or prevent these?

4. **Cloud-Specific Knowledge Tests:** Since both Azure and AWS implementations have subtle differences, probe cloud-specific questions: How do you manage secrets for Databricks on AWS? How is cost optimization achieved for Snowflake on Azure? These reveal real expertise.
5. **Stack Integration Scenarios:** Challenge the vendor on integrating these platforms with your broader ecosystem, whether Azure Synapse, Microsoft Fabric, Power BI, or third-party tools. Vendors dependent solely on “standard” use cases may show limited creativity or capability.

Key Differences in Skill Sets to Watch For

Skill Area Databricks Expertise Snowflake Expertise Data Engineering Spark-based ETL, Delta Lake optimization, streaming pipelines, ML pipelines, notebook development SQL-centric ELT, Snowpipe ingestion, micro-partitioning, performance tuning with clustering keys Cloud Infrastructure Cluster setup with autoscaling, workspace security, integration with AD/IAM, cost management Multi-warehouse sizing, role management, resource monitors, external functions integration Data Governance Unity Catalog for fine-grained access, lineage tracking in notebooks, governance via Delta Lake RBAC on objects, masking policies, row-level security, integration with external catalogs like AWS Glue Semantic Layer Delta Live Tables, Feature Stores, notebook-based reusable data artifacts Materialized views, secure views, BI layer integration in Power BI or Looker CI/CD and IaC Pipelines with Databricks CLI, Terraform/Azure DevOps/CloudFormation integration for cluster & job management Terraform for Snowflake objects, deployment pipelines for SQL scripts and UDFs

Final Thoughts

Choosing suppliers with proven **databricks expertise** versus **snowflake expertise** requires thorough stack validation and clear understanding of your current and future analytics needs. Vendors proficient in both can offer hybrid architectures combining the flexibility of a lakehouse with the robustness of a cloud data warehouse, but you need to discern true mastery from surface-level claims.

Key prerequisites for any vendor include deep delivery experience beyond pilots, demonstrable governance and lineage capabilities, solid cloud-specific implementation experience (Azure and AWS), and a commitment to CI/CD and infrastructure-as-code best practices. Without these, what looks like a shiny “lakehouse” or “AI-ready” solution could quickly become a governance and support nightmare.

Always ask, “Where does the semantic layer live?” and “Who owns data quality tests?” If the answer is vague or missing, you are often investing in tech over trust — and that’s a red flag for any enterprise-scale data modernization.