



7 BEST PRACTICES FOR BUILDING DATA APPLICATIONS ON SNOWFLAKE

Break out of scalability, concurrency, and performance barriers



CHAMPION
GUIDES

EBOOK

TABLE OF CONTENTS

- 3** Begin with the basics
- 4** Invest in a modern architecture
- 5** Discover the Snowflake difference
- 6** Best practice #1: Select strategic virtual warehouse sizes by service or feature
- 8** Best practice #2: Adjust minimum and maximum cluster numbers to match expected workloads
- 9** Best practice #3: Target workloads to the right technologies
- 10** Best practice #4: Reduce SRE/DevOps burden with self-tuning and self-healing
- 11** Best practice #5: Benefit from a range of integrations and partnerships
- 12** Best practice #6: Be strategic with materialized views
- 13** Best practice #7: Use defaults for suspension
- 14** Conclusion
- 15** About Snowflake Inc.

BEGIN WITH THE BASICS

There has never been a better time to build SaaS data applications. International Data Corporation (IDC) predicts that big data and business analytics solutions will generate revenues of \$189 billion in 2019, experiencing double-digit growth through 2022¹. Startups and independent software vendors must meet two basic requirements for their apps to be competitive in this market:

- **THEY MUST INGEST LARGE VOLUMES OF DATA WITH SPEED.**
- **THEY MUST ANALYZE ALL THAT DATA QUICKLY AND EASILY.**

These requirements hold true for all data app types, including business intelligence (BI), Internet of Things (IoT), marketing and sales automation, customer relationship management (CRM), and machine learning, to name a few. It no longer matters if you're building a data app for the financial, retail, insurance, or healthcare

industry or any other industry: The demands on data ingestion and the need for real-time analytics are universal.

Once they have met these fundamental needs, data app builders must demonstrate their product's strong performance for a large number of concurrent users on a global scale. Adding to the challenge, it's imperative for builders to keep their own expenses in line while growing the business and do the best job possible future-proofing their technology investments.

This ebook explains how data apps, and the customers they serve, benefit from development on a cloud-built data platform, and it provides seven best practices around architectural, deployment, and operational settings to ensure you customize and maximize those benefits.

INVEST IN A MODERN ARCHITECTURE

Data app providers must carefully consider their data stack architecture. A massive-scale SaaS application simply cannot handle modern customer demands if it was not built for the cloud.

Yet, many data apps today are built on traditional data stacks, including legacy on-premises and “cloud-washed” data warehouses. These technologies were either created before the cloud existed or shoehorned into the cloud. As a result, they lack the cloud-native attributes that make modern apps successful.

Traditional data stacks have limitations around scalability, concurrency, and performance. Supporting multiple separate workloads is almost impossible in a single-cluster architecture where everyone fights for resources. Even if different clusters are used for concurrent workloads, the risk of data inconsistency is high when data changes. The outcome is inaccurate analytics and unhappy customers. Data app builders who attempt to scale within these architectures require large capital investments, which can be a death knell for a startup with limited financial resources.

The same challenges hold true for open-source databases, where components don't scale and the underlying infrastructure is limited. Rather

than building new features, app developers find themselves constantly rearchitecting to solve intrinsic problems with fragmented open-source technology. These challenges include latency, incomplete data analysis, and overhead requirements around system maintenance, upgrades, and security.

The heart of the problem is that typical massively parallel processing (MPP) data warehouse architectures or cluster-based solutions (such as Hadoop) tie storage and compute resources together. This coupling makes it extremely challenging to support concurrent workloads. Users get frustrated by slow and disruptive scaling, and engineers waste time working around these issues for users.

You can avoid all these scenarios by adopting a modern data stack with unlimited and automatic scalability, concurrency, instant elasticity, and support for semi-structured data. Cloud-built technologies have these strengths built into their core architecture, which enables customers to extract maximum business insights and value from their data.

To keep data app developers focused on new customer features, the underlying technology should also be a fully managed service with a secure data environment. This modern architecture keeps costs under control by providing smarter query execution and a “pay for what you use” model.

DISCOVER THE SNOWFLAKE DIFFERENCE

Snowflake's cloud-built data platform provides the modern stack you need to develop and scale modern data applications. Snowflake is built on and for the cloud and therefore includes fundamental cloud benefits that become clear when you examine its architecture, deployment, and operations.

ARCHITECTURE

Snowflake's modern architecture corrects all the flaws inherent in legacy systems to deliver a new type of data platform that enables scalability, elasticity, and concurrency:

SEPARATION OF RESOURCES

When we designed Snowflake, one of the most important decisions we made was to physically separate, but logically integrate, compute and storage. This eliminates the cluster-building efforts that other systems must perform to make separate layers work together.

As a result of this architectural decision, Snowflake provides a multi-cluster, shared data architecture where three main components work together seamlessly.

- **Storage:** All data is stored in a persistent storage layer, which resides in a scalable cloud storage service (such as Amazon S3, Microsoft Azure, and Google Cloud Storage) for maximum data replication, scaling, and availability without customer management.

- **Compute:** Independent compute resources execute data processing tasks, such as loading, transformation, and querying. Snowflake provides "virtual warehouses," or compute clusters, that access databases in the storage layer and execute queries with automatically cached data. Virtual warehouses can be created, resized, and deleted dynamically.
- **Services:** The cloud services layer handles system services such as infrastructure, security, automatic metadata management and resilience, access control, and optimization. Services also coordinate query processing and return results by communicating with client applications via connectors (including JDBC, ODBC, and Kafka clients), and "plug in" services via APIs.

Separating compute and storage means the same data can be used by multiple users simultaneously through multiple compute clusters. Workload contention is eliminated with dedicated and independent compute resources, so slowdowns or disruption to queries never happen.

BEST PRACTICE #1: SELECT STRATEGIC VIRTUAL WAREHOUSE SIZES BY SERVICE OR FEATURE

Dedicate separate Snowflake virtual warehouses (compute clusters) for queries and workloads. This practice helps enable lower compute usage by allocating the right-sized compute resources to specific services, features, or workloads.

For example, rather than use a large virtual warehouse (eight credits per hour), you may discover that a medium-size virtual warehouse (four credits per hour) and a small-size virtual warehouse (two credits per hour) match your application's needs better. This strategy saves two credits per hour without any sacrifice in performance.

And, for those times when a heavy one-time analysis is needed, you can run queries on a separate right-sized warehouse that doesn't impact other queries. If there's a fixed amount of work, it often makes sense to use the biggest warehouse size. Query performance tends to scale linearly, so a large warehouse will end up delivering faster analysis and costs the same as a smaller warehouse that takes more time.

AUTO-SCALING WITH MULTI-TENANCY

With Snowflake, scalability is automatic and without limits, thanks again to a multi-cluster, shared data architecture that separates compute from storage. All users experience seamless, nondisruptive scaling and instant elasticity without the need to redistribute data.

With the Snowflake point-and-click user interface, or with a few short SQL statements, you can create virtual warehouses in seconds. As part of the warehouse creation process, simply tell Snowflake whether you want it to auto-scale and auto-suspend; the Snowflake service then orchestrates the entire cloud infrastructure for you, detecting when scaling is needed and performing the operation automatically. No administrator intervention is required.

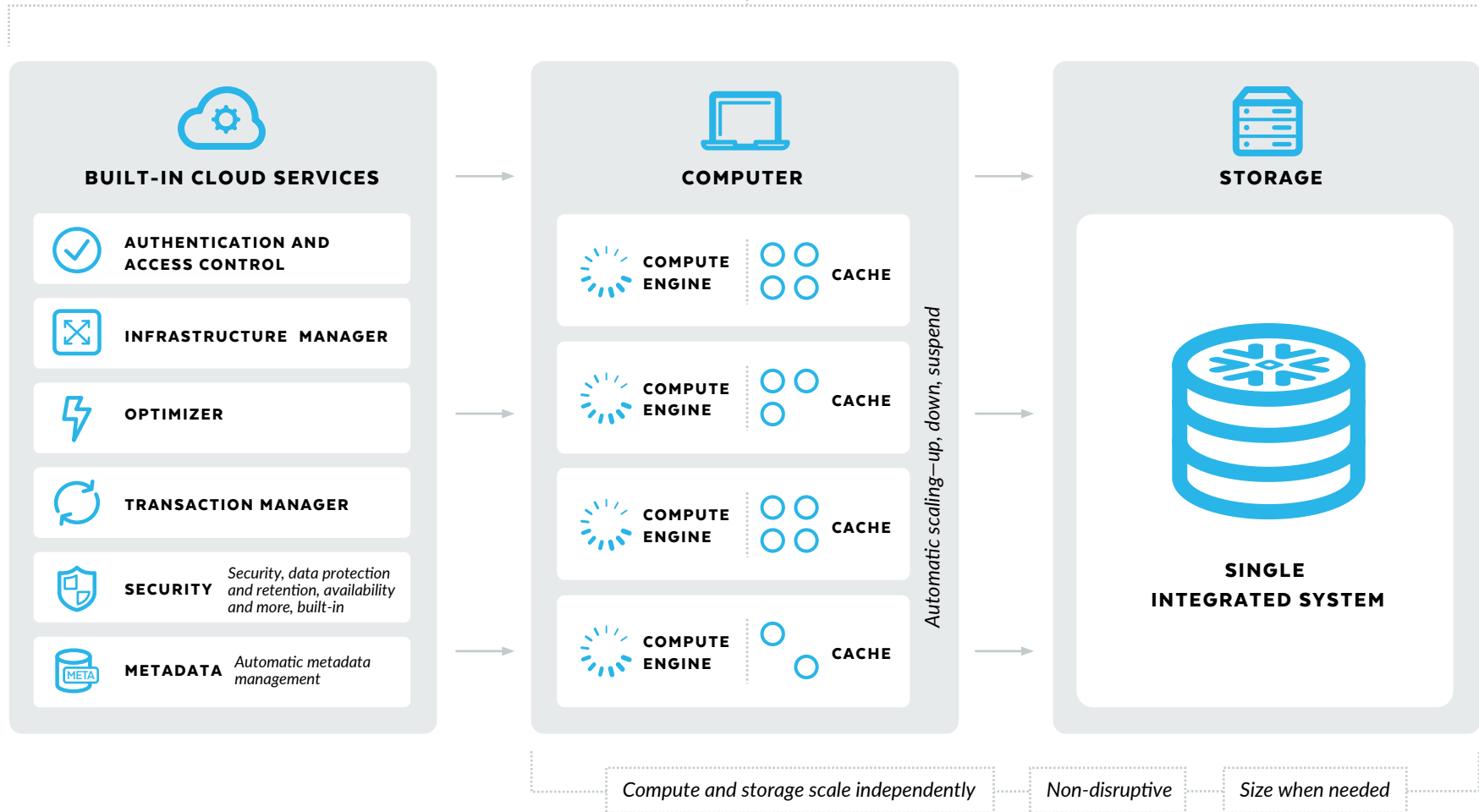
That means data app providers have the ability to instantly and infinitely scale compute up, down, and out to any and all workloads. Provisioning is immediate for on-demand performance. Snowflake enables any number of virtual warehouses (compute engines) to be spun up to support any number of customers—hundreds or thousands. Each customer can be assigned its own virtual warehouse and have complete control over the size and amount of compute resources. It takes less than a minute to launch a Snowflake virtual warehouse, and they can be set to scale automatically based on workload demand.

The same holds true with storage: Resources can be scaled to virtually any capacity at any time, with the added benefit of not incurring charges for unnecessary compute resources.

SNOWFLAKE IS A HIGHLY SCALABLE, MULTI-TENANT ENVIRONMENT



BUILT FOR THE CLOUD



Snowflake is a data platform powered by a unique auto-scaling architecture that delivers high performance.

BEST PRACTICE #2: ADJUST MINIMUM AND MAXIMUM CLUSTER NUMBERS TO MATCH EXPECTED WORKLOADS

Select a warehouse size (small, medium, large) that provides adequate performance for each individual query that runs, keeping in mind that a given warehouse size can run individual queries twice as fast as the size below it, and each additional cluster allows the warehouse to run more queries in parallel to increase concurrency. Then, to maximize performance and minimize costs, it's important to adjust a virtual warehouse's minimum and maximum number of clusters based on the corresponding concurrent throughput you expect for the workload.

Keep in mind that, as workloads subside, one cluster at a time shuts off so you pay only for the resources needed in any given moment. This Snowflake strategy provides consistent performance regardless of the number of queries.



BEST PRACTICE #3: TARGET WORKLOADS TO THE RIGHT SERVICES

If you're ingesting multiple types of data from multiple sources, it's important to recognize what your data needs and set up your architecture to support separate workloads that are targeted at the technologies that make the most sense. For example, you might want to process some streaming data in near real time and take action on it, while other data types might not need immediate attention and instead should be sent straight to storage for future complex analytical segmentation.

By building these capabilities into your architecture early on, you accelerate the ability to manage your data and derive fast insights exactly where they are needed. With Snowflake, every piece of data can be sent to two places, which makes it easy to immediately process and store it, and it's easy to handle unstructured data without forcing schemas on customers.

SUPPORT FOR SEMI-STRUCTURED AND STREAMING DATA

Traditional data stacks fall short due to singular support for structured data, limited processing capabilities, and inadequate memory. Snowflake's architecture shines by enabling the consolidation of all data into one platform.

- **Semi-structured data:** For semi-structured data types such as JSON, Avro, Parquet, and XML, Snowflake's patented VARIANT data type loads, transforms, and integrates semi-structured data natively alongside structured data. Other platforms require multiple data stores and query grids, but Snowflake makes it easy to query semi-structured data immediately in a fully relational manner.
- **Streaming data:** Streaming data from sources such as IoT devices, mobile devices, and advertising technology is necessary if you want to perform complete data analysis. That's why Snowflake built a service called Snowpipe to provide continuous loading for streaming data and serverless computing for data loading. Traditional systems often rely on latent tactics such as batch loading; Snowflake automatically loads data into target tables through a programmatic REST API or within a minute of receiving AWS S3 event notifications.

With Snowflake, analytics always run against a complete data set, which represents a single source of truth. New data that's already in Amazon S3, Azure Data Lake Storage, or Google Cloud Platform can be loaded in mere minutes, and customers can execute complex queries, including joins, without performing any pretransformations. That means data app providers can deliver real-time insights to customers using all of their data—without any additional work or wasted effort.

SELF-SERVICE

Traditional infrastructures have a dependency: Requests for data access and resource scaling must go through the data team, which is responsible for orchestrating all resources of the data platform. Time to value slows down because the data team becomes a natural bottleneck. This situation is especially frustrating for data scientists, integration developers, business analysts, data stewards, executives, and the finance and sales teams, all of whom need fast access to data and resources for real-time decision-making and collaboration opportunities.

In contrast, Snowflake provides a data platform where self-service is enabled at any scale. The complexities and workflows that create bottlenecks are completely removed to provide higher value for organizations and faster time to value.

BEST PRACTICE #4: REDUCE SRE/DEVOPS BURDEN WITH SELF-TUNING AND SELF-HEALING

When traditional systems break down due to software or hardware defects or intensive workloads, manual intervention is needed, which in turn requires Site Reliability Engineering (SRE) or DevOps teams to intervene. Administrators have to be prepared to analyze workloads and tweak any of the hundreds or thousands of controls that might be available.

In contrast, Snowflake builds high availability and the ability to self-tune and self-heal into every layer of the system. A great example is the way that Snowflake handles data. Any time it's written to a table, that data is synchronously written to highly durable cloud storage in three different data centers. If a compute cluster in one data center starts losing machines, or if the entire data center goes down, Snowflake instantly provisions a cluster in another data center that has access to all your data.

DEPLOYMENT

Many data app providers adopt a continuous integration/continuous delivery (CI/CD) pipeline model that enables apps to be improved and code to be deployed to customers on a regular basis. That's why Snowflake provides data recovery features to ensure deployments always go smoothly, even when they don't.

DATA RECOVERY

Traditional data warehouses struggle with CI/CD because it requires backups before a release and complex table isolation and ETL data loading during a release. And that's assuming all goes well: If a database needs to be rolled back after a bad release, then DevOps is looking at another costly and time-consuming process.

To ensure all releases and data are backed up properly, Snowflake provides continuous data protection (CDP) through two built-in features that eliminate the need for traditional backup scripts and processes:

- **Time Travel:** Snowflake provides a fully updatable relational database and uses data modification language (DML) operations to capture updates or deletion of data rows. These changes are written internally to a new storage object that automatically retains the previous storage object.

As such, all data and data objects are fully recoverable during a retention period (the length of which is determined by a service agreement), which ensures that accidental errors or bad releases can be rolled back with ease.

- **Fail-Safe:** After a retention period has passed, Snowflake has a built-in "fail-safe" seven-day period where data can still be recovered upon request. After this extended time has passed, an automated process deletes the data.



BEST PRACTICE #5: BENEFIT FROM A RANGE OF INTEGRATIONS AND PARTNERSHIPS

Snowflake's ecosystem is designed for flexibility, openness, and extensibility without exposure to security threats. That means you benefit from partnerships and seamless integrations with vendors of your choice, including vendors for BI tools, data science technologies, marketing and sales automation software, CRM solutions, and machine learning.

Another area you might use vendors for is security analytics. Snowflake enables you to connect to existing SIEM software, fraud products, and threat intelligence feeds. In addition, Snowflake provides built-in security options from leading BI partners, including Tableau, Looker, Sigma Computing, and Periscope Data, so you can create a wide range of user interfaces, visualizations, and reports that align to your needs, processes, and workflows.

OPERATIONS

Snowflake is a fully managed service that requires near-zero maintenance and provides complete security. The burden to build and maintain complex data infrastructures is removed when there's nothing to manage or optimize, and no downtime is required for software updates. Data app engineers can focus all their energies on building new features rather than on managing the data stack.

COMPLETE SECURITY

Every aspect of Snowflake's cloud data platform is designed to deliver end-to-end data security and protect data against current and evolving security threats. Snowflake follows best-in-class standards and practices, leverages NIST 800-53 and the CIS Critical Security Controls, and provides always-on secure data environments with ACID compliance, including multi-statement transaction support.

Snowflake protects against security threats and ensures consistency and data integrity by providing:

- Encryption of data stored in the cloud
- Automatic protection against accidental or malicious loss of data
- Fine-grained, role-based access control for data and actions
- Isolation of query processing and data storage

Snowflake is SOC II Type 2 certified, HIPAA and PCI DSS compliant, and FedRAMP Ready. In addition, Snowflake integrates with existing security information and event management (SIEM) software, as well as with anti-fraud products, BI tools, ticketing systems, and data science technologies.



BEST PRACTICE #6: BE STRATEGIC WITH MATERIALIZED VIEWS

Another way to save money is to be strategic about creating materialized views and measure the performance benefits you get before deciding whether to keep them. If it's inexpensive to run a query on the base table, or if a big query is run infrequently, the cost of materialized view maintenance will not be worth it. The only candidates recommended for materialized views are extremely expensive aggregations, projections, and selections that must be run frequently.

PER-SECOND PRICING

Snowflake provides a clear pricing model with only two items to consider:

- **Storage** is charged per terabyte, compressed, per month.
- **Compute** is based on how many processing units are consumed to run queries or perform a service. Compute charges are billed on actual usage, per second, and only active clusters accrue charges.

BEST PRACTICE #7: USE DEFAULTS FOR SUSPENSION

A traditional data warehouse must run at full capacity 24/7, regardless of whether it's being used. In contrast, Snowflake is built to match usage. Simply enable all virtual warehouses to automatically be suspended when they are idle and automatically resume when they are queried. That means you won't pay for these virtual warehouses when your users aren't running queries.

There are no added usage quotas or hidden price premiums. You simply pay for what you use. With Snowflake's cloud-built architecture, data app providers can start small and increase usage as needed with auto-scaling. Instant elasticity enables Snowflake's cloud-built data platform to scale up, down, and out for complete efficiency and affordability.

Snowflake also includes integrated usage tracking by time or by accumulated usage, which allows you to easily administer cost allocations and chargebacks. It also lets you stay on top of usage for all virtual warehouses and monitor hot spots throughout any period.



CONCLUSION

The best way to differentiate your data application is to provide customers with a highly performant service that analyzes all data together and delivers insights with speed and agility. By future-proofing your data stack with a cloud-built data platform, you deliver remarkable customer experiences while guaranteeing the right framework and support for your own organic growth. You'll achieve all that while not having to worry about planning for and performing any of the menial, time- and cost-heavy tasks that were once required to scale your systems, product, and business.



ABOUT SNOWFLAKE

Snowflake's cloud data platform shatters the barriers that have prevented organizations of all sizes from unleashing the true value from their data. More than 2,000 customers deploy Snowflake to advance their businesses beyond what was once possible by deriving all the insights from all their data by all their business users. Snowflake equips organizations with a single, integrated platform that offers the only data warehouse built for the cloud; instant, secure, and governed access to their entire network of data; and a core architecture to enable many types of data workloads, including a single platform for developing modern data applications. Snowflake: Data without limits. Find out more at [snowflake.com](https://www.snowflake.com)



© 2020 Snowflake. All rights reserved.

CITATIONS

¹ <https://www.idc.com/getdoc.jsp?containerId=prUS44998419>