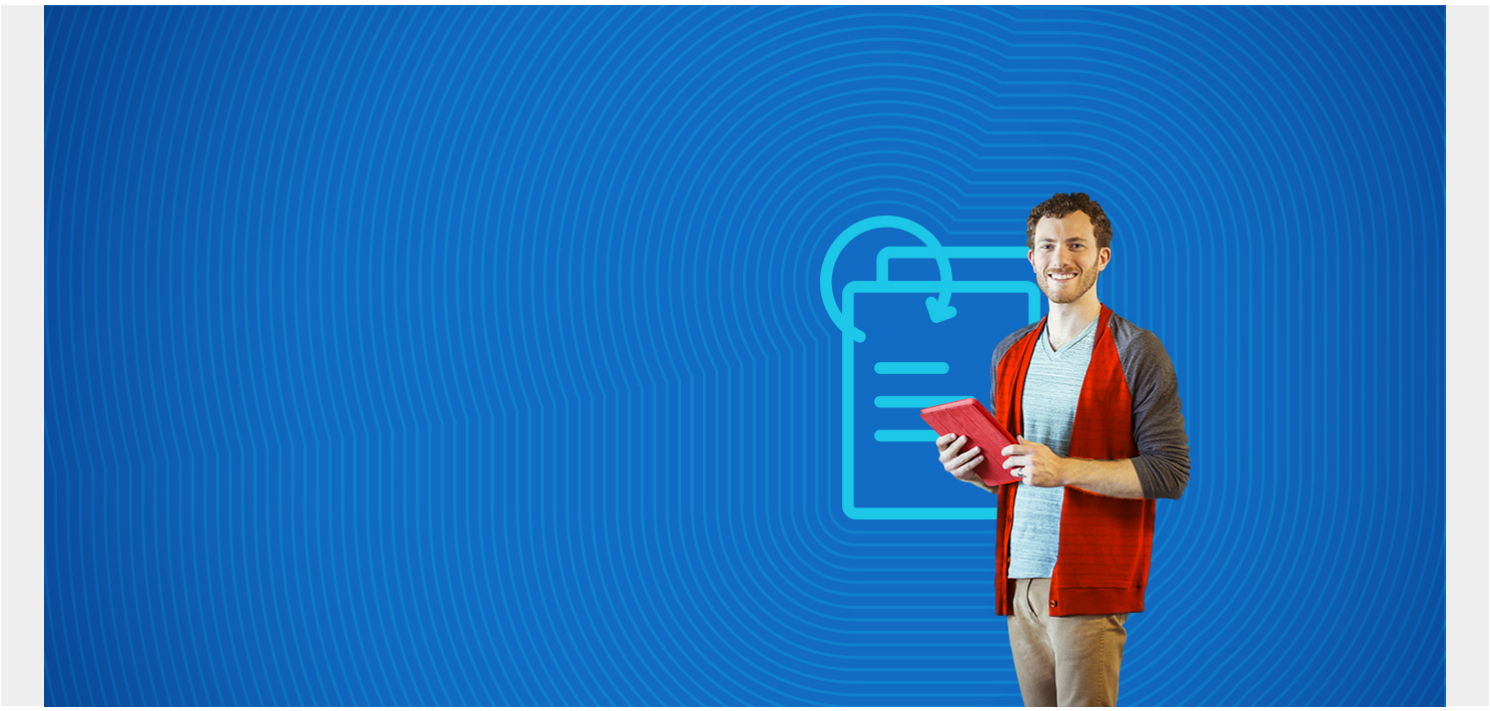
SNOWFLAKE 101: INTRO TO THE SNOWFLAKE DATA CLOUD



With data's consistent rise in volume and velocity, organizations seek solutions to process big data and any related challenges. One of the first decisions that organizations take? Adopting a cloud-based model that offers flexibility, scalability, and high performance.

Snowflake is one cloud-based data warehouse platform that is gaining popularity thanks to its numerous features and efficiency.

In this article, we delve into Snowflake's architecture, key features, and the purpose it solves.

(This article is part of our Snowflake Guide. Use the right-hand menu to navigate.)

What is Snowflake?

Snowflake is a SaaS-based data warehouse (DWH) platform that runs over an AWS or MS Azure cloud infrastructure. (You might hear this called data warehouse as a service.)

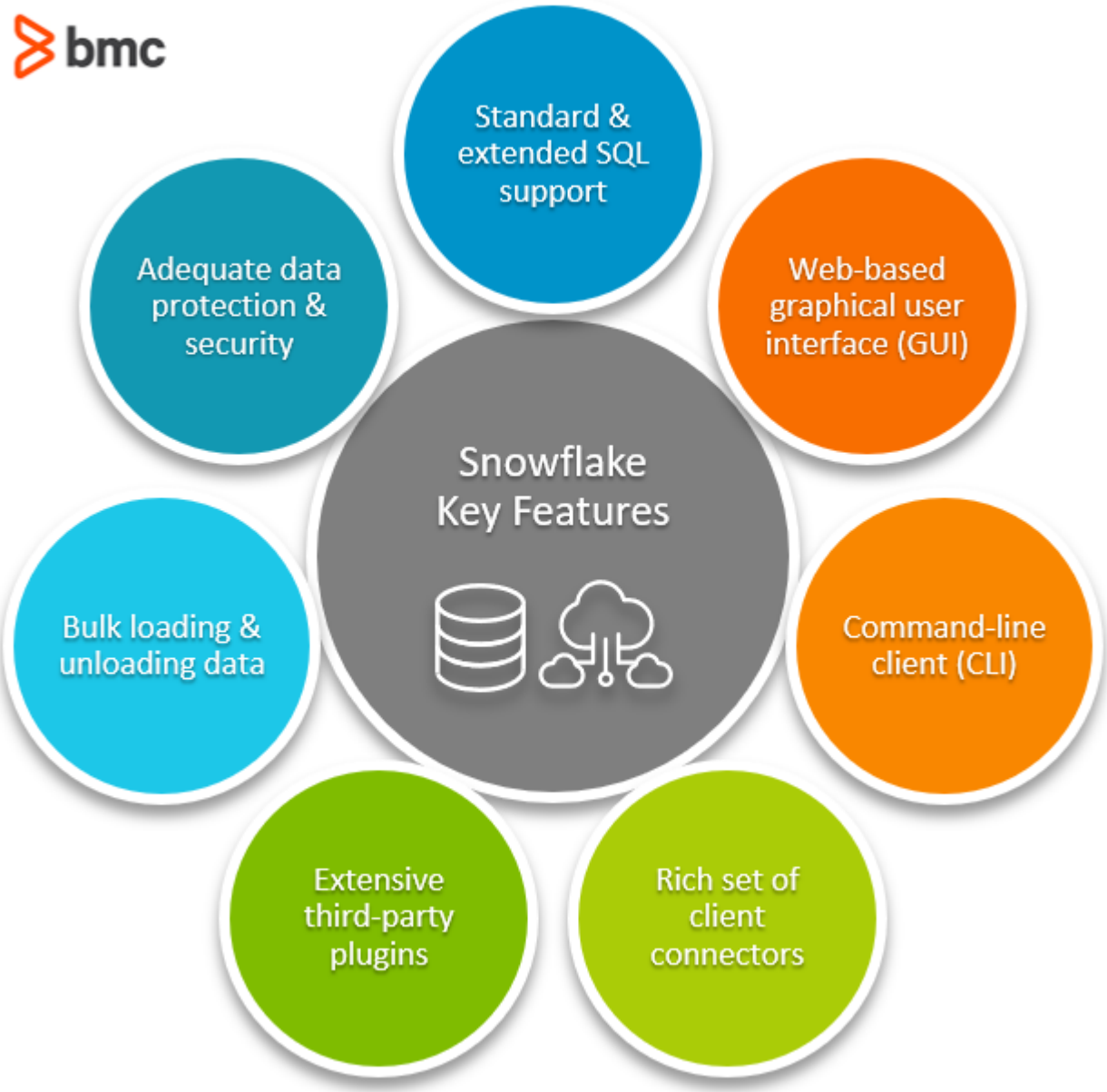
Unlike other warehouse solutions, Snowflake utilizes an enhanced ANSI-compliant SQL engine that is designed to work solely on the cloud.

Fundamentally, Snowflake's core architecture enables it to run on the public cloud, using virtual compute instances and efficient storage buckets, making it a highly scalable and cost-efficient solution to process enormous amounts of big data.

(Understand the differences between data warehouses & databases.)

Key features of Snowflake

When compared to legacy DWH technologies, Snowflake offers a number of features, including:



Standard & extended SQL support

As a SQL-based data warehouse, it supports the specified data-defined language and data manipulation language DML commands used by SQL. It also provides advanced DML commands for multi-table operations such as INSERT, MERGE, and MULTI-MERGE.

With Snowflake, users can:

- Set up temporary and transient tables for short-term data
- Use analytical and statistical aggregate functions and lateral views
- Create user-defined functions (UDFs) to extend functionality in both SQL and JavaScript

(Compare SQL & no-SQL data storage.)

Web-based graphical user interface (GUI)

Snowflake provides a web interface for users to interact with the data cloud. With the web GUI, users can:

- Manage their account and other general settings
- Monitor resources and system usage
- Query data

Command-line client (CLI)

Snowflake provides a Python-based CLI called SnowSQL for connecting to the DWH. It is a separate downloadable and installable terminal tool for executing all queries, including data definition and data manipulation queries for loading and unloading data.

(Get started with our Python introduction.)

Rich set of client connectors

Snowflake provides a wide range of connectors and drivers that users can use to connect to their data cloud. Some of these client connectors include:

- **Python Connector**, a programming interface for writing Python apps that connect to Snowflake
- **NodeJS driver**
- **ODBC driver** for C/C++ development
- **JBDC driver** for Java programming

Extensive third-party plugins

In addition to the programmatic interfaces mentioned above, several other big data tools integrate with Snowflake. These tools range from business intelligence tools to data integration, machine learning, security, and governance software.

Bulk loading & unloading data

Snowflake allows data loading in different formats and from various data sources - as long as the data uses a supported character encoding. Users can load data from:

- Compressed files
- AWS S3 data sources
- Local files
- Flat data files like CSV and TSV
- Data files in Avro, JSON, ORC, Parquet, and XML formats

Additionally, with Snowpipe, users can continuously load data in batches from within Snowflake stages, AWS S3, or Azure storage.

Adequate data protection & security implementation

With Snowflake, users can:

- Set regions for data storage to comply with regulatory guidelines
- Adjust their security levels based on requirements

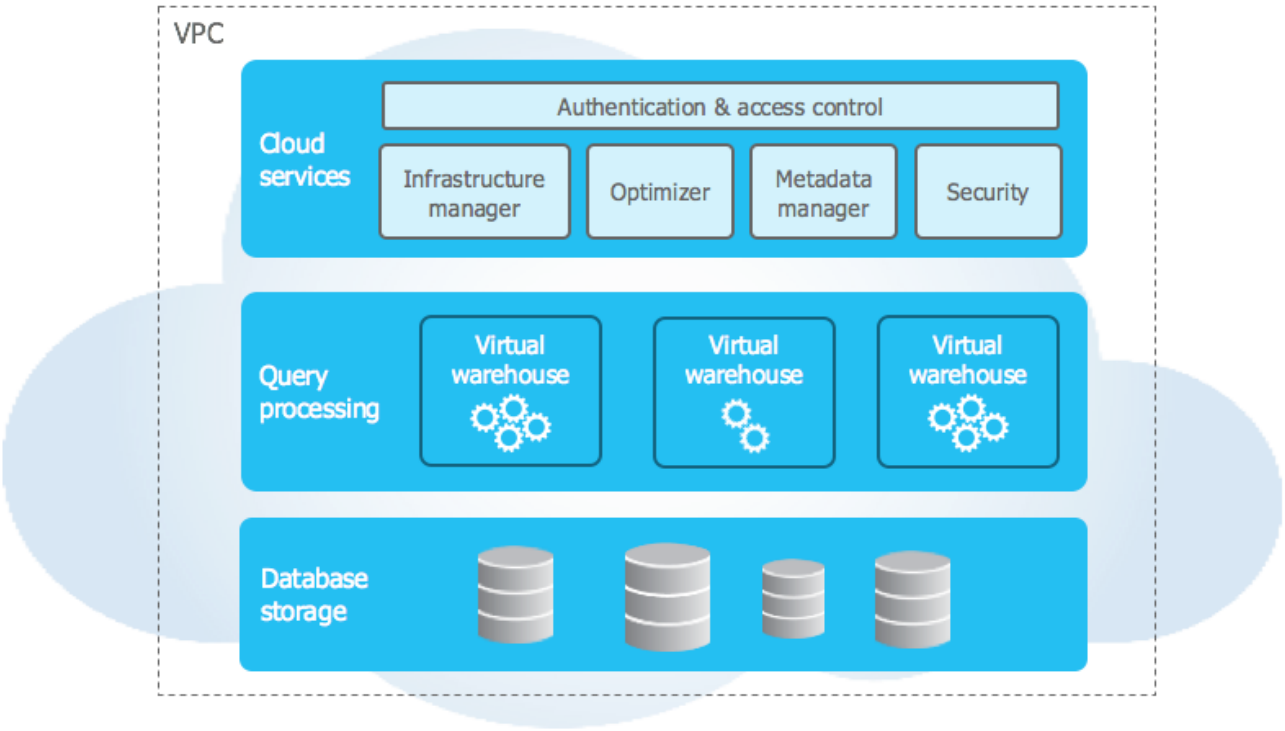
Snowflake also automatically encrypts data. Object-level access control offers granular control on who can access what.

Snowflake architecture

Snowflake follows a hybrid of **shared-disk** and **shared-nothing** database architecture. It consists of:

- A central repository that persists data
- Compute nodes within the data warehouse can access that base disk storage

For executing queries, Snowflake uses distributed Massively Parallel Processing (MPP) cluster nodes, each having its own local storage for storing portions of data locally, CPU, and memory.



(Source)

Snowflake's framework is typically segregated across three layers. All of these layers are independent of each other and can be scaled, configured, and managed individually. These layers include:

- Storage layer
- Compute layer
- Cloud services layer

Storage layer

The layer at which the central repository lies. Any data loaded into the system undergoes partitioning and reorganization into Snowflake's compressed, internally optimized columnar format, encryption using AES 256, and subsequently stored in cloud storage. Snowflake automatically does

the partitioning but provides settings for users to configure partition parameters.

Data stored in this layer is central, and all nodes in the cluster can access it. Snowflake manages all aspects of data storage, thereby allowing users to only interact with the underlying data through SQL queries.

Compute layer

The compute layer handles the execution of queries. It does this using virtual warehouses—that are independent MPP compute clusters with multiple compute nodes.

Snowflake assigns these compute nodes from a chosen cloud provider to each user. These clusters are autonomous—having their own CPU, memory, and local storage—where the performance of one does not affect the others.

Cloud services layer

Snowflake provides a collection of services for administering and managing a Snowflake data cloud. This layer is where several activities happen:

- Access control
- Authentication
- Infrastructure management
- Metadata management
- Query parsing
- Optimization

Why use Snowflake?

There are plenty of reasons organizations opt for Snowflake. Here are the top reasons:

- **Hybrid architecture offers users the best of both worlds.** Users pay separately for the underlying central repository and as much compute power as they require.
- **SQL-based for fast learning.** A SQL-based implementation ensures developers do not have to go through a steep learning curve to understand new technology.
- **Data first.** Supports data cloning and secure data sharing.
- **No infrastructure configuration.** Snowflake does not require any infrastructure configuration –instead, Snowflake does it automatically once you've chosen the preferred cloud service provider.

Getting started with Snowflake

Ready to get started? Snowflake currently offers a 30-day free trial to new users. Once you get access, you can:

- View our Snowflake Guide menu, to the right, for several tutorials.
- Refer to the official documentation

Snowflake is cloud native

Cloud-native services are the new normal.

Snowflake is one DWH service that has been built specifically for the cloud that allows organizations to handle enormous big data storage and processing by allowing to scale compute and storage independently. For faster query execution and improved performance, Snowflake allows users to scale up with additional data warehouses by offering extra compute resources, as required.

While offering enhanced DWH features, Snowflake helps to cut-down costs of provisioning infrastructure and the redundant efforts of managing it, allowing organizations to focus on generating efficient analytics—the whole purpose of data.

Related reading

- BMC Machine Learning & Big Data Blog
- Using Stored Procedures in Snowflake
- MySQL vs MongoDB: Comparing Databases, part of our MongoDB Guide
- DataOps Explained: Understand how DataOps leverages analytics to drive actionable business insights
- Data Architecture Explained: Components, Standards & Changing Architectures
- Data Ethics for Companies