Get started > Get started for users > Key concepts and architecture

Snowflake key concepts and architecture

Snowflake is powered by an advanced data platform that is provided to you as a self-managed service. Snowflake's data platform brings together data storage, processing, and analytic solutions that are faster, easier to use, and far more flexible than traditional offerings.

Snowflake combines a completely new SQL query engine with an innovative architecture that is natively designed for the cloud. It offers full enterprise analytic database functionality, and unique features and capabilities.

Data platform as a self-managed service

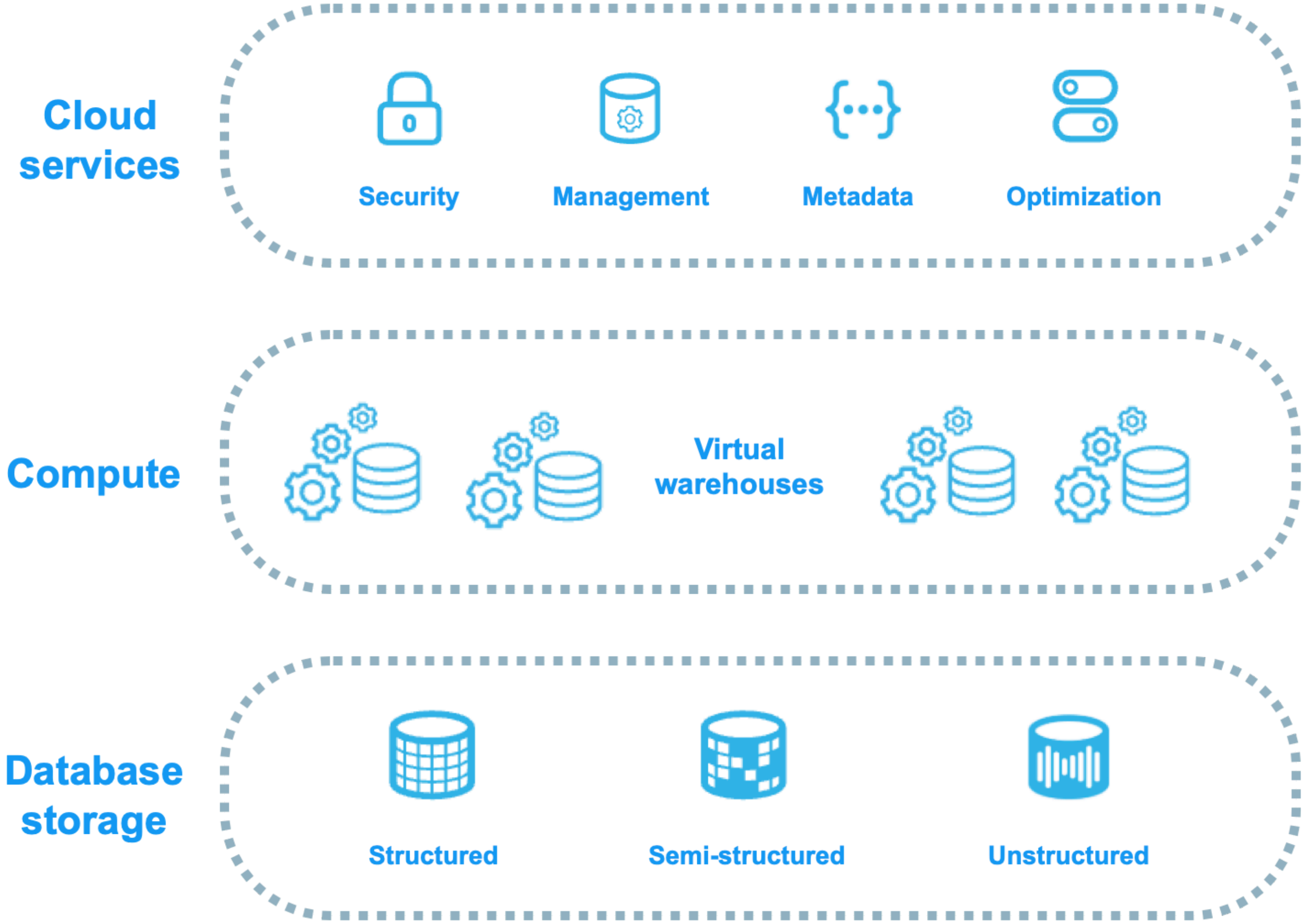
As a *self-managed service*, Snowflake has the following advantages:

- There is no hardware (virtual or physical) for you to select, install, configure, or manage.
- There is virtually no software for you to install, configure, or manage.
- Ongoing maintenance, management, upgrades, and tuning are handled by Snowflake.

Snowflake uses public cloud infrastructure to host virtual compute instances and persistent data storage. Snowflake manages software updates and infrastructure so you don't have to. You can't install and run Snowflake locally or on private cloud infrastructures, whether on-premises or hosted.

Snowflake architecture

Snowflake's architecture is a hybrid of traditional shared-disk and shared-nothing database architectures. Similar to shared-disk architectures, Snowflake uses a central data repository for persisted data that is accessible from all compute nodes in the platform. But similar to shared-nothing architectures, Snowflake processes queries using massively parallel processing (MPP) compute clusters, where each node in the cluster stores a portion of the entire data set locally. This hybrid architecture, which is shown in the following diagram, offers the data management simplicity of a shared-disk architecture, but with the performance and scale-out benefits of a shared-nothing architecture:



Snowflake's unique architecture has the following key layers:

- Database storage
- Compute
- Cloud services

Database storage

Snowflake supports the following kinds of data:

- *Structured data* — such as rows and columns in a table — follows a strict tabular schema.
- *Semi-structured data* — such as a JSON file or an XML file — has a flexible schema.
- *Unstructured data* — such as a document, image, or audio file — has no inherent schema.

Snowflake supports several types of tables for data storage, including the following table types:

- Snowflake tables

- Apache Iceberg™ tables
- Hybrid tables

Snowflake tables

When data is loaded into a Snowflake table, Snowflake reorganizes that data into its internally optimized, compressed, columnar format. Snowflake stores this optimized data in cloud storage. Snowflake tables are ideal for data warehouses.

Snowflake manages all aspects of how this data is stored — including the organization, file size, structure, compression, metadata, and statistics. All data in Snowflake tables is automatically divided into *micro-partitions*, which are contiguous units of storage. Micro-partitions improve efficiency and provide other benefits.

You can use Snowflake tables to store structured and semi-structured data. You can also use the FILE data type for unstructured data.

For more information about Snowflake tables, see Understanding Snowflake Table Structures.

Apache Iceberg™ tables

Apache Iceberg™ tables for Snowflake combine the performance and query semantics of typical Snowflake tables with external cloud storage that you manage. They are ideal for existing data lakes and data lakehouses that you can't, or choose not to, store in Snowflake.

Iceberg tables store their data and metadata files in an external cloud storage location; for example, Amazon S3, Google Cloud Storage, or Microsoft Azure Storage. The external storage isn't part of Snowflake.

You can use Iceberg tables to store structured and semi-structured data.

For more information, see Apache Iceberg™ tables.

Hybrid tables

Hybrid tables are optimized for low latency and high throughput by using index-based random reads and writes. Hybrid tables support row locking and enforce unique and referential integrity constraints, which are critical for transactional workloads. You can use a hybrid table along with other Snowflake tables and features for Unistore workloads that bring transactional and analytical data together in a single platform.

You can use hybrid tables to store structured and semi-structured data.

For more information, see Hybrid tables.

Compute

A *virtual warehouse* is a cluster of compute resources in Snowflake. Virtual warehouses process SQL statements and, using Snowpark, run code in languages, such as Java, Python, and Scala. With Snowpark Connect for Spark, you can also run Apache Spark™ workloads on virtual warehouses.

Each virtual warehouse is an independent compute cluster that doesn't share compute resources with other virtual warehouses. As a result, each virtual warehouse has no effect on the performance of other virtual warehouses.

For more information, see Virtual warehouses.

Cloud services

The cloud services layer is a collection of services that coordinate activities across Snowflake. These services tie together all of the different components of Snowflake in order to process user requests, from sign-in to query dispatch. The cloud services layer also runs on compute instances that are provisioned by Snowflake from the cloud provider.

Services managed in this layer include the following:

- Security, authentication, and access control
- Snowflake Horizon Catalog
- Infrastructure management with cloud platforms
- Metadata management, including the SNOWFLAKE database and the Snowflake Information Schema
- Query parsing and optimization
- Regulatory compliance

Integrated features for your workloads

Instead of moving data to different systems so that different teams can complete specific operations and tasks, you can bring all of your workloads directly to their data with an integrated set of features.

These features support the following broad areas of data integration and development:

- Data engineering
- Analytics
- AI and ML
- Applications and collaboration

Data engineering

Snowflake separates storage and compute, which simplifies some traditional challenges of data engineering, such as infrastructure management and performance tuning. Data engineers can focus on implementing pipelines that ingest, transform, and deliver data.

Snowflake provides several ways to ingest data, including the following options:

- COPY INTO <table> command — Loads data from files to a table.
- Snowpipe — Loads data from files as soon as they are available in a stage.
- Snowpipe Streaming — Loads row-level data continuously and with low latency, using the Snowflake SDKs or a REST API, directly into Snowflake tables and Snowflake-managed Iceberg tables, instead of loading data from files.
- Openflow connectors — Ingest data from specific sources by using connectors built on Apache NiFi, such as Microsoft Sharepoint and Google Drive.
- Snowflake Connectors — Connect from external applications and systems and stream data into Snowflake.

Snowflake also provides several ways to transform data, including the following options:

- Dynamic tables — Define tables that automatically refresh based on target freshness and a query that performs data transformations.
- Streams and tasks — Capture changes made to base objects with streams and define tasks to perform data transformations.
- Snowpark — Perform more complex transformations by using programming languages, such as Python, Java, and Scala.

- dbt — Use an open-source data transformation tool and framework to define, test, and deploy SQL transformations.

In addition, SnowConvert AI can ingest and transform data, and Snowpark Migration Accelerator can convert code from various platforms to Snowflake.

For more information, see Overview of data loading.

Analytics

With Snowflake, you can scale workloads dynamically based on demand, access different types of data — including structured, semi-structured, and unstructured — and share data easily. These features let you analyze data stored in Snowflake to extract meaningful insights, patterns, and trends for analytical use cases, such as business intelligence or predictive modeling.

Snowflake provides several ways to analyze data, including the following options:

- System functions and SQL constructs — Perform calculations and statistical analysis with the following Snowflake system functions and SQL constructs:
  - Aggregate functions — Summarize data by performing calculations on a set of related rows and returning a single value.
  - Window functions — Perform calculations on a set of related rows in partitions for rolling operations on subsets of the rows in each partition, such as calculating running totals or moving averages.
  - Common table expressions (CTEs) — Improve the readability and reusability of complex queries, which might perform multiple steps of data transformation.
- Cortex AI Functions — Run unstructured analytics on text and images with large language models (LLMs) from OpenAI, Anthropic, Meta, Mistral AI, and DeepSeek.
- Semantic views — Store semantic business concepts directly in the database to define business metrics and model business entities and their relationships.

AI and ML

Snowflake simplifies the use of artificial intelligence (AI) and machine learning (ML) capabilities so you can perform AI and ML feature engineering, training, and inference with your Snowflake data.

Models can access your most up-to-date data in a secure environment. With Snowflake, you can avoid the cost and complexity of moving your data to a separate platform for AI and ML tasks.

Snowflake offers AI and ML capabilities in two broad suites of features:

- Snowflake Cortex — AI features that use LLMs to understand unstructured data, answer freeform questions, and provide intelligent assistance. Cortex AI functions can automate routine tasks, such as simple summaries and quick translations.
- Snowflake ML — Features that you can use to build your own models. ML functions give you automated predictions and insights into your data by using ML. Snowflake ML is a unified environment for ML development.

For more information, see Snowflake AI and ML.

Applications and collaboration

Snowflake offers many ways to build applications and share them with your teams, partners, and customers. When you use Snowflake to share data, you control access to the data, and avoid the challenges of keeping it synchronized in different places.

The following list shows some of the tools and services you can use to build, deploy, and manage applications in Snowflake:

- Streamlit — Use an open-source Python library to create and share custom web apps with an interactive user interface (UI) for ML and data science.
- Snowpark Container Services — Deploy, manage, and scale containerized applications from directly inside Snowflake.
- Snowflake Native App Framework — Build applications that expand the capabilities of other Snowflake features by sharing data and related business logic with other Snowflake accounts. The business logic of an application might include a Streamlit app, stored procedures, and functions written by using Snowpark API, JavaScript, and SQL. A Snowflake Native App can also run container workloads with Snowpark Container Services.

Snowflake includes support for the following kinds of collaboration:

- Secure Data Sharing — Share selected objects in a database in your account with other Snowflake accounts.

- Listings — Provide data and other information to other Snowflake users, or access data and other information shared by Snowflake providers. You can explore, access, and provide listings to consumers privately and on the Snowflake Marketplace.
- Data Clean Rooms — Define what analyses can be run against the shared data, which allows the consumer to gather insights from the data without having unrestricted access to it.

Snowgrid

Snowgrid is Snowflake's cross-region, cross-cloud technology layer. With Snowgrid, you can achieve the following goals:

- Connect a data ecosystem across different cloud regions and providers — such as, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud — by using listings and other collaboration features.
- Apply consistent security and governance policies across clouds and regions.
- Enable disaster recovery and business continuity capabilities across regions by using replication.

For more information, see Snowgrid.

Connecting to Snowflake

Snowflake supports multiple ways for you to connect to the service:

- Snowsight, a web-based UI that you can use to access all aspects of managing and using Snowflake can be accessed.
- Command-line clients that you can also use to access all aspects of managing and using Snowflake; for example, Snowflake CLI.
- Native APIs that you can use to create and manage Snowflake resources programmatically; for example, Snowflake Python APIs and Snowflake REST APIs.
- Drivers that other applications can use to connect to Snowflake; for example, JDBC and ODBC.
- Native connectors that you can use to develop applications for connecting to Snowflake; for example, Apache Kafka and Apache Spark.
- Third-party technologies that you can use to connect applications to Snowflake; for example, extract, transform, load (ETL) tools such as Informatica, and business intelligence (BI) tools

such as ThoughtSpot.

For more information, see Sign in to Snowflake.