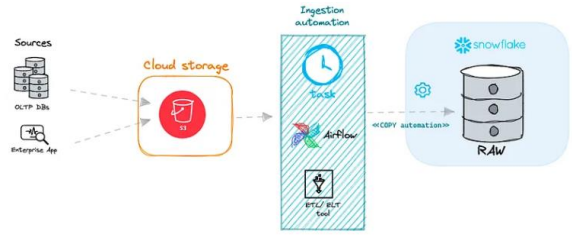
-Snowflake provides a variety of different data ingestion options, each with their own pros, cons and important considerations.

-options for both batch and continuous/real time loading,

-Selecting the right option can have an impact on cost, performance, reliability and ease of maintenance.

-radeoffs like latency, cost, maintenance and required technical expertise.



Batch loading

-data ingestion approach used in all data warehouses,on a fix schedule

- big tradeoff: latency.

-If your data loads once per day, then it can be up over 24 hours stale before the next load happens.

-batch ingestion typically works by taking files from cloud storage (an Amazon S3 bucket, Microsoft Azure Blob storage or Google Cloud Storage), and having some process or orchestration tool automatically load that data into Snowflake on a schedule.

- here are two main approaches to batch data ingestion.

-Batch loading with the COPY command

-This command takes files previously uploaded into a Snowflake internal stage or external stage, and ingests them into the target table. The files could be CSV files, JSON data, or any other supported file format option.

-COPY command runs on one of your own managed virtual warehouse, meaning you are responsible for creating and properly sizing the virtual warehouse

-Cost & efficiency considerations

-COPY INTO command in Snowflake, you pay for each second the virtual warehouse is active

- billed for a minimum of 60 seconds each time they resume,

-The smallest virtual warehouse is able to ingest 8 files in parallel.

-SKILL Required:

-Using the COPY command does not require any additional skills on top of basic SQL knowledge.

- Batch loading with serverless tasks

- Serverless tasks allow Snowflake users to run SQL commands on a defined schedule using Snowflake compute resources instead of their own managed virtual warehouse.

-you only pay per second of compute used, serverless tasks can help with the under utilized compute problem that comes with batch loading on your own virtual warehouse

- The compute costs for serverless tasks are charged at 1.5X the rate of an equivalent sized virtual warehouse managed by you.

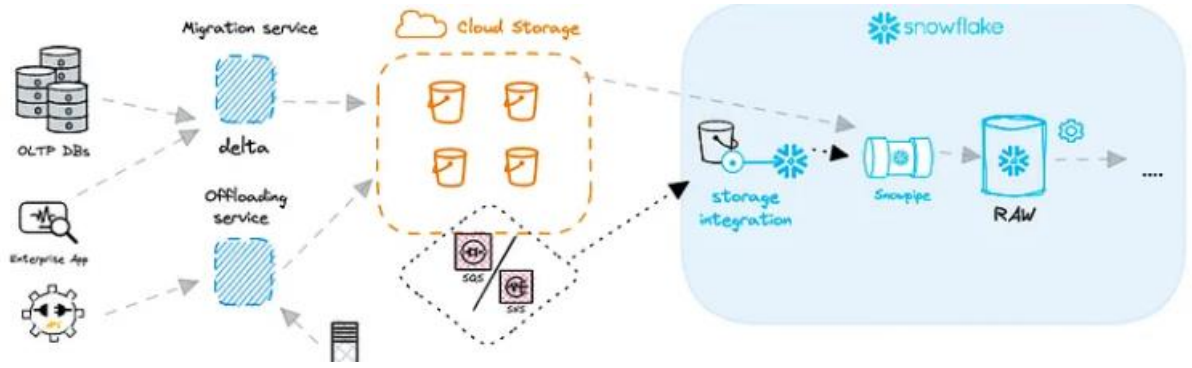
-Continuous loading with Snowpipe

-Unlike batch processing which runs on a fixed schedule, continuous loading processes data based on some event, usually a new file landing in cloud storage.

- Each new file triggers an event notification, which automatically kicks off the loading process

- Cloud based messaging services like AWS SNS or SQS are often used to trigger these notifications.

- With continuous loading, data latency will be much lower as new data is constantly being loaded as soon as it becomes available.



- continuous processing, Snowflake offers a powerful feature called Snowpipe

- Snowpipe is integrated with an event notification service on the cloud provider side (i.e. AWS SNS/SQS). Each Snowpipe object also has an associated COPY command

- Most customers use the “auto-ingest” Snowpipe feature, meaning files get automatically loaded as soon as they arrive.

- REST API for Snowpipe which allows you to choose when a Snowpipe object is triggered.

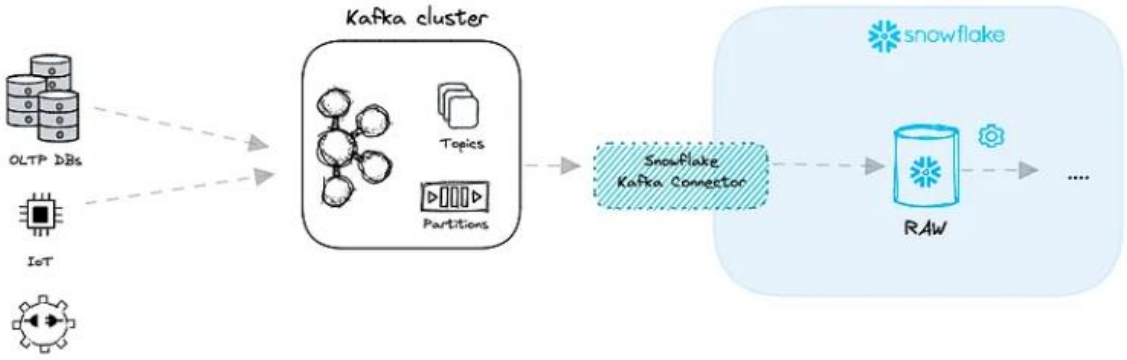
-Snowpipe is serverless feature, meaning you do not have to worry about which virtual warehouse to use or whether it is sized correctly.

-Snowpipe charges an overhead fee of 0.06 credits / 1000 files processed

- Snowflake recommends aiming for your files to be 100–250MB compressed.

-If your upstream applications frequently send data with small file sizes, you will want to consider implementing a process that will aggregate the files into larger sized batches. A common service used for this purpose is Amazon's Kinesis Firehose

-Real time loading with Kafka



- credit scoring, fraud analysis, or even user facing analytics.

-Delivering data with low latency is typically accomplished by using a message broker like Apache Kafka.

- Snowflake becomes a data consumer.

-"Snowpipe mode", which combines Kafka with the traditional Snowpipe methods we discussed above.

- "Snowpipe Streaming" mode, a new offering released by Snowflake in 2023.

-Kafka Connector — Snowpipe mode

- The Snowpipe mode for Kafka connector uses a combination of micro-batching files and Snowpipes. Kafka messages are flushed into temporary files and ingested via Snowpipe

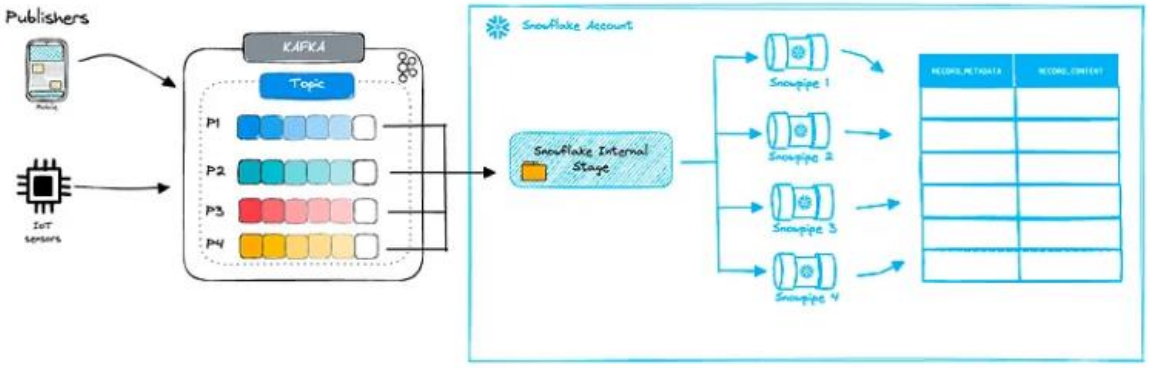
-Maintaining this optimal file sizing can become challenging considering the other factors that come into play:

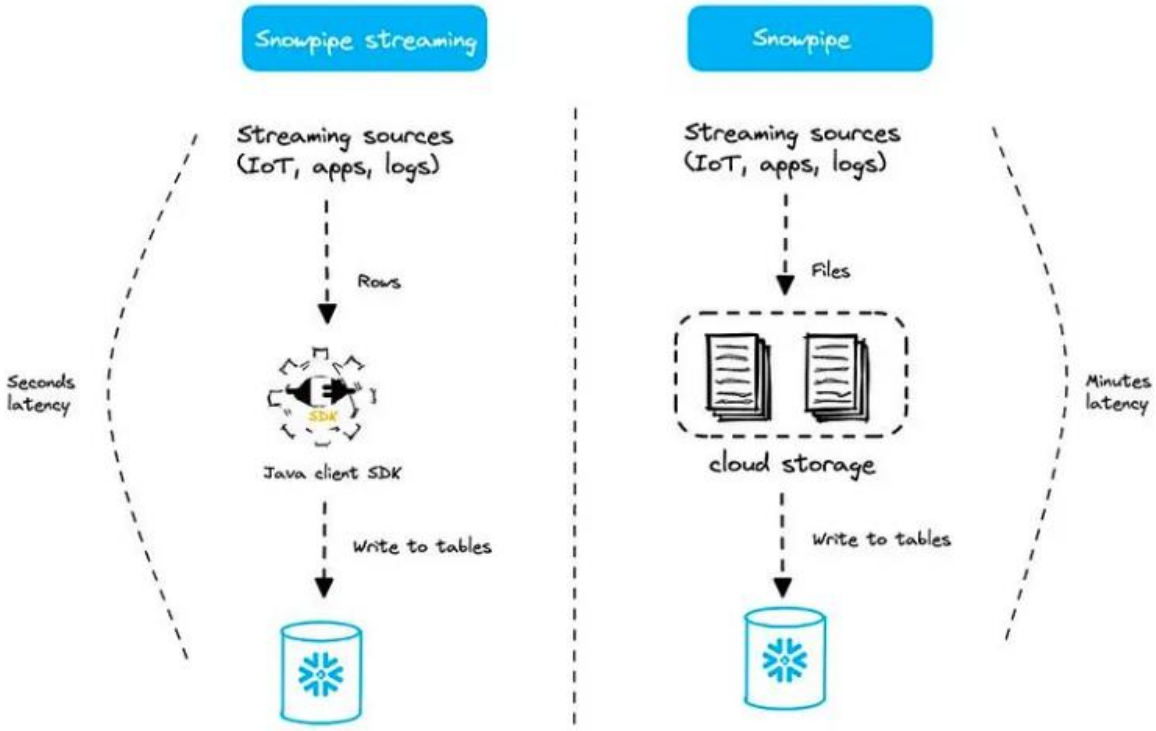
-how frequently your source generates data and how fast you need to have its data loaded into Snowflake

- he flush rate: how frequently you flush the data into the files, configurable in the Kafka connector options

-the partition count in your Kafka cluster

- Snowpipe Streaming





-With Snowpipe Streaming, there is no stage, no files and no snowpipe objects (which I recognize is confusing, since the name includes “Snowpipe”). With Snowpipe Streaming, data is loaded row by row instead of using files.

-Snowpipe at 1 credit per compute-hour (Snowpipe is 1.25).

- If a user creates 100 clients, then this overhead management fee would cost $22,000 / year assuming an credit price of $2.5/credit (100*0.01*24*365*2.5).

-he Snowpipe Streaming API is part of the Java SDK, which means you must be familiar with Java in order to use the feature.

-How to choose a data loading option

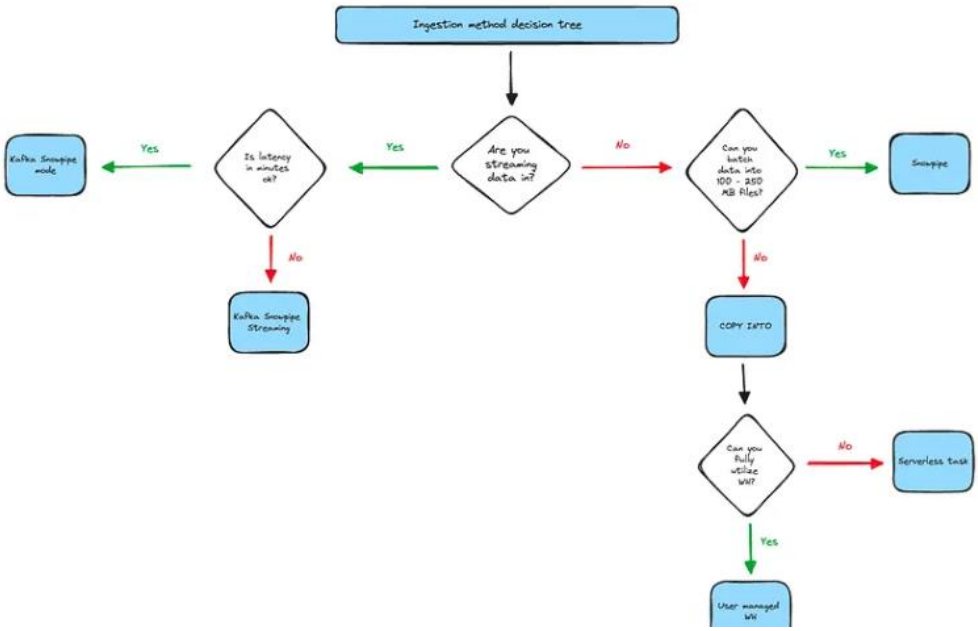
-What latency does your use case require?

- Are you looking for the most cost efficient option?

- What technical expertise does your team have and how will the loading option fit into your existing stack?

-What system is producing the data and files you need to load, and do you have control over that?

Method	Option	Lowest Possible Latency	Required skills
BATCH	COPY command	minutes	SQL
BATCH	Serverless task	minutes	SQL
CONTINOUS	Snowpipe	minutes	SQL + cloud
REAL TIME	Kafka Snowpipe mode	minutes	SQL + cloud + Kafka
REAL TIME	Kafka Snowpipe streaming	seconds	SQL + cloud + Java



https://medium.com/snowflake/an-overview-of-data-loading-options-in-snowflake-712e6ae4a6d2