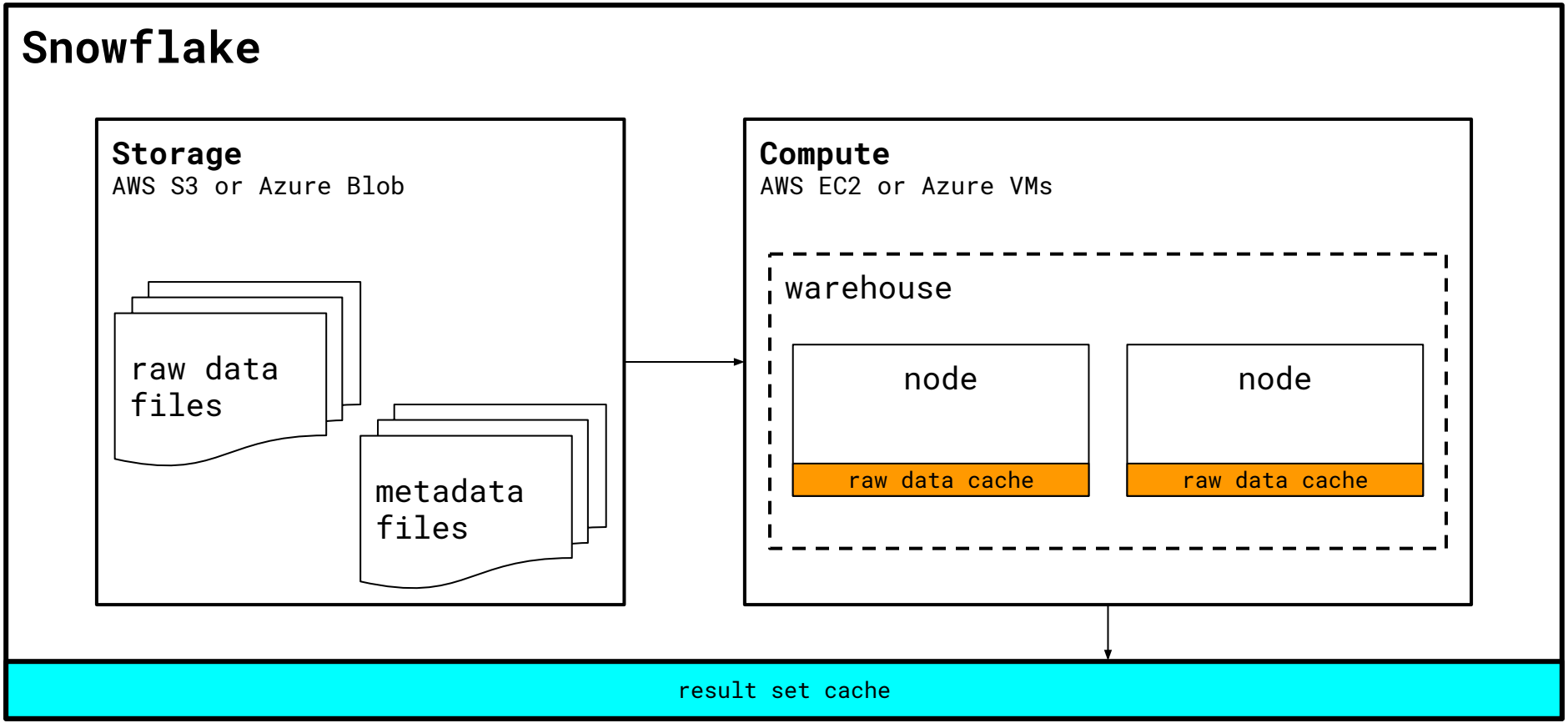
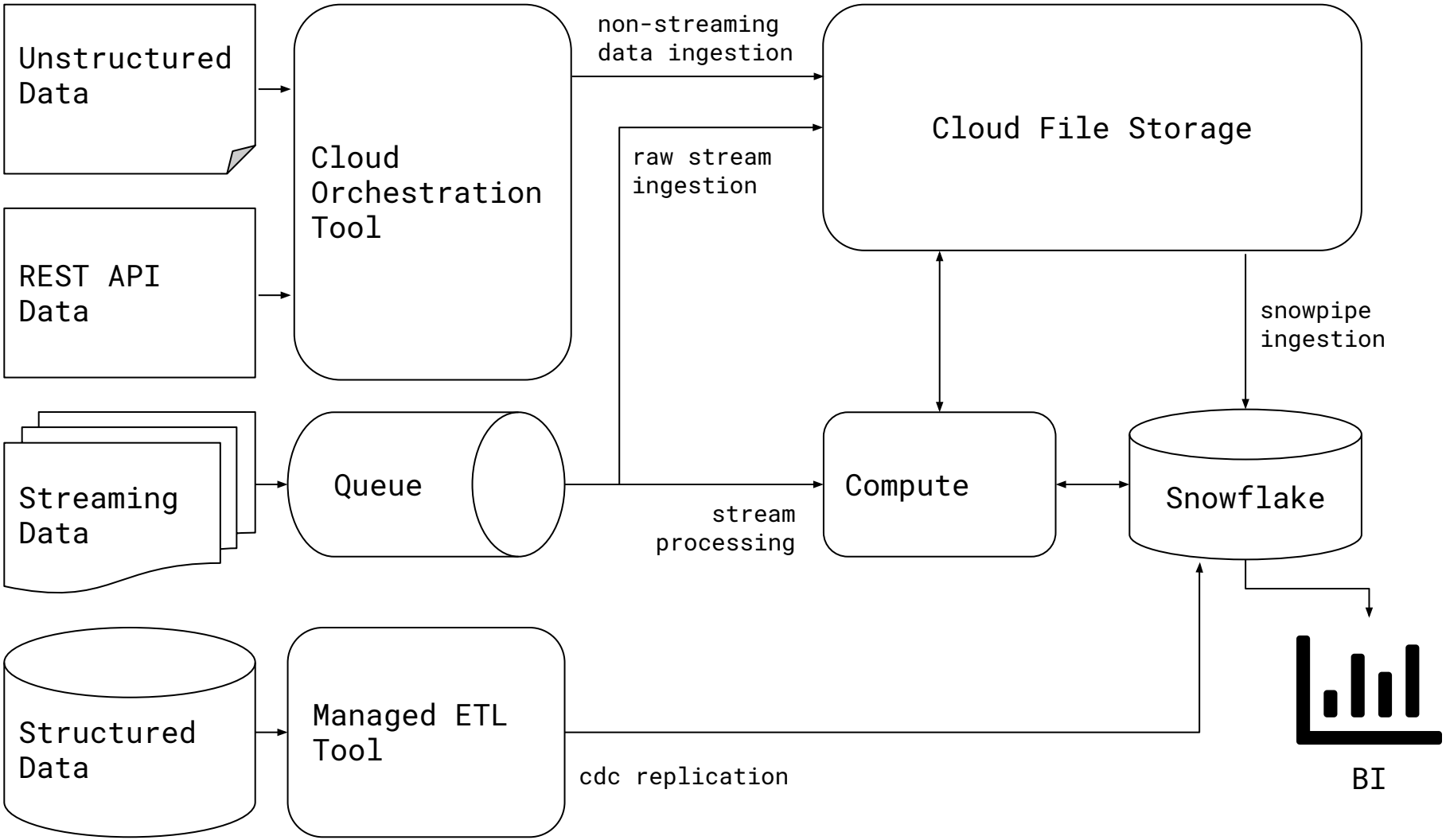
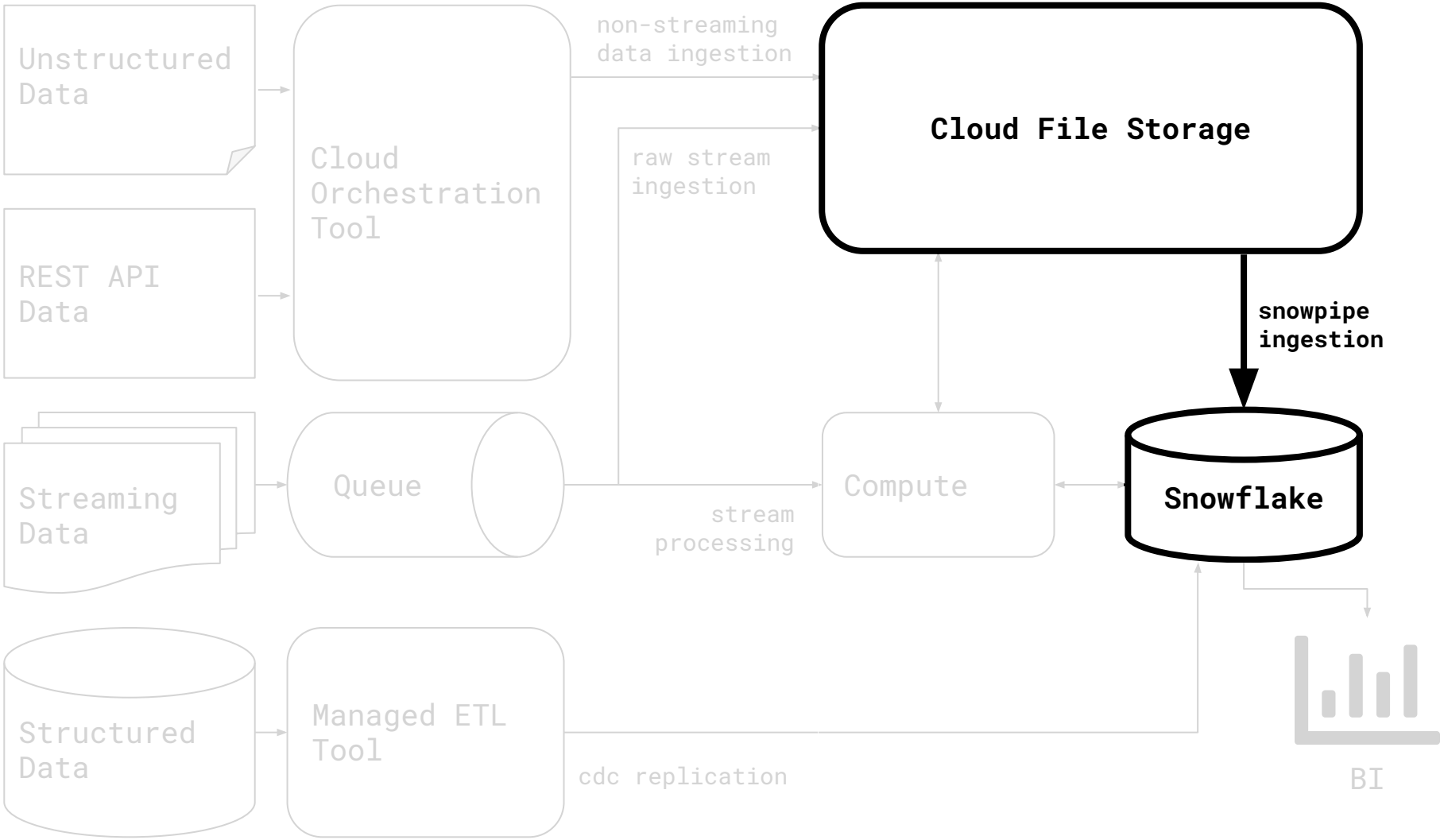
Ingestion of Streaming Sources into Snowflake using Snowpipe

Atlanta Snowflake Cloud Data Warehouse User Group
November 6th 2019

Meetup Github: http://bit.ly/32j9QkD


Snowflake
Storage
AWS S3 or Azure Blob
raw data files
metadata files
Compute
AWS EC2 or Azure VMs
warehouse
node
raw data cache
node
raw data cache
result set cache


Unstructured Data
REST API Data
Cloud Orchestration Tool
non-streaming data ingestion
raw stream ingestion
Cloud File Storage
snowpipe ingestion
Streaming Data
Queue
stream processing
Compute
Snowflake
Structured Data
Managed ETL Tool
cdc replication
BI


Unstructured Data
REST API Data
Streaming Data
Structured Data
Cloud Orchestration Tool
Queue
Managed ETL Tool
non-streaming data ingestion
raw stream ingestion
stream processing
cdc replication
Cloud File Storage
snowpipe ingestion
Compute
Snowflake
BI

Data Acquisition Considerations - 3Vs

Volume

Volume		
Large: 1. Bulk Loading File based 2. Push products eg., Attunity HVR 3. Spark	Medium: 1. ETL tools eg.,Talend, Pentaho 2. Bulk Loading File based 3. SAAS offerings eg., ADF, Glue, Alooma, Fivetran, Stitch	Small: 1. Python connectors 2. ETL tools eg.,Talend, Pentaho 3. SAAS offerings eg., ADF, Glue, Alooma, Fivetran, Stitch 4. Serverless Eg., Lambda, Azure Functions

Data Acquisition Considerations - 3Vs

Variety and Velocity

Variety		
RDBMS: 1. ETL tools eg.,Talend, Pentaho 2. Bulk Loading File based 3. Push products eg., Attunity HVR 4. Spark	JSON, XML etc: 1. SAAS offerings eg., ADF, Glue, Alooma, Fivetran, Stitch 2. Spark, Kafka based	Security
Velocity		
Batch: 1. ETL tools eg.,Talend, Pentaho 2. Bulk Loading File based 3. Push products eg., Attunity HVR 4. Spark	Realtime: 1. Serverless eg., 2. Confluent, Databricks 3. SAAS offerings eg., ADF, Glue, Alooma, Fivetran, Stitch	CDC: 1. Build your own on ELT patterns with SnowSQL

Snowpipe Ingestion Components

1. Cloud Storage (AWS, Azure, GCP)

2. AWS - S3, IAM, SQS Notifications

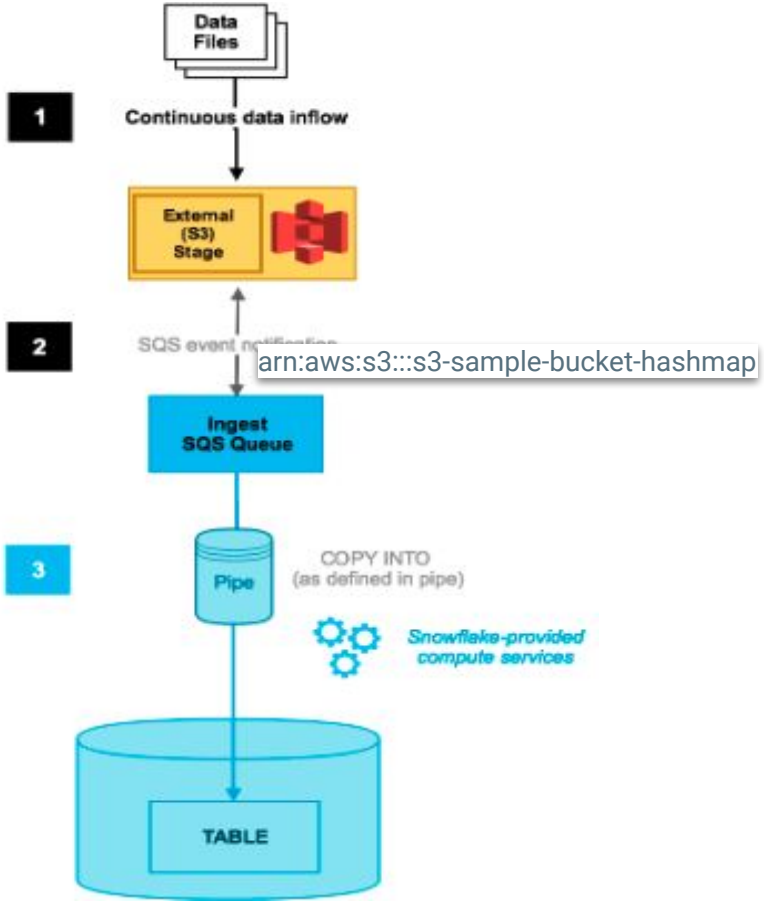
3. Snowflake Objects - Warehouse (Compute), Database (Storage), Schema, Fileformat, Stage, Table, Pipe

Why and How - Snowpipe Ingestion

Why

1. Cost: Cheaper
2. Velocity: Near real time
3. Setup: Fast

How



Ingestion Source

{"exchange":"binance","base":"ethereum","quote":"bitcoin","direction":"buy","price":0.021254,"volume":0.019,"timestamp":1572018897798,"priceUsd":176.7050820281409}
{"exchange":"binance","base":"stellar","quote":"bitcoin","direction":"buy","price":0.00000755,"volume":57,"timestamp":1572018897799,"priceUsd":0.062770460586829}

JSON structure - Key value pair


```
{
  "base": "bitcoin",
  "direction": "sell",
  "exchange": "binance",
  "price": 9243.52,
  "priceUsd": 9276.044929863805,
  "quote": "tether",
  "timestamp": 1572537834970,
  "volume": 0.007547
}
```


Github:http://bit.ly/2qoSUfx

Json converted to a structured form

↓ Row	BASE	DIRECTION	EXCHANGE	PRICE	PRICEUSD	QUOTE	TIMESTAMP	VOLUME	CURRENT_TIME
1,000	iostoken	buy	binance	0	0	ethereum	2019-11-05 22:...	85	2019-11-05 14:1...
999	litecoin	sell	binance	63	63	tether	2019-11-05 22:...	1	2019-11-05 14:1...

Workshop Steps to build Continuous Ingestion:

1. Build named file format and stage in Snowflake.
2. Create a staging table to hold raw ingestion data along with an **INGESTION_TIME** field.
3. Create a snowpipe in the same schema as the staging table.
4. Configure Queue notifications for new files arriving in the stage.
5. After confirming new files are arriving, manually run a copy command to load existing files from the stage if any

Things to know about Snowpipe:

- Loads data from files as soon as they’re available in a stage (auto loading). The load may take up to a minute to complete.
- Load history stored in metadata of pipe
- Meant for frequently staged data
- No guarantees that files are loaded in the order it is staged
- Can contain only COPY command
- Requires SQS (AWS) configuration
- Can be invoked via REST, if auto ingestion is not needed

Staging Tips:

- Keep file sizes to between 10 and 100 MB for best results
- Watch out for JSON files that are greater than 16MB uncompressed (the max variant column size is 16MB)
- PIPE refresh commands only grab data less than 1 week old
- SNOWPIPES and MATERIALIZED VIEWS are each billed in a single line item. It’s difficult to attribute costs to individual pipes or views