



# **SNOWFLAKE 101**

## **PLATFORM**

# PLATFORM REQUIREMENTS

## FAST FOR ANY WORKLOAD



Run any number or type of job across all users and data volumes quickly and reliably.

## IT JUST WORKS



Replace manual with automated to operate at scale, optimize costs, and minimize downtime.

## CONNECTED TO WHAT MATTERS



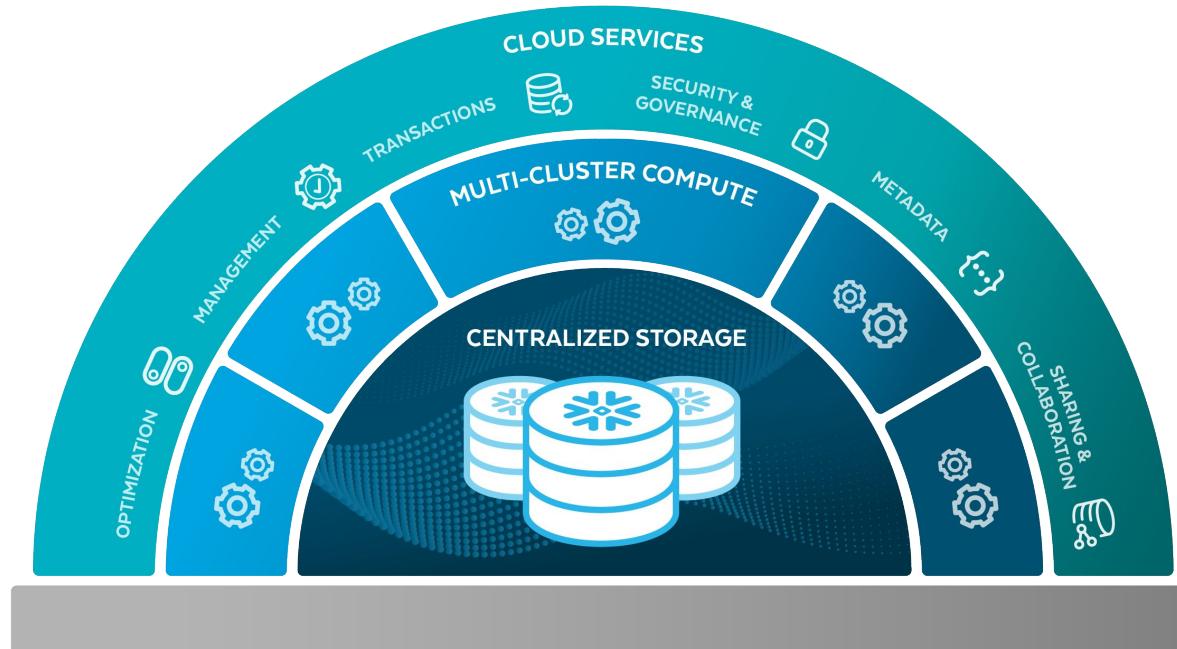
Extend access and collaboration across teams, workloads, clouds, and data, seamlessly and securely.

# SNOWFLAKE PLATFORM

## Under the hood



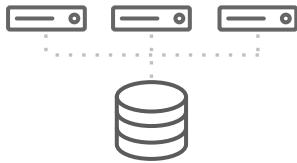
# SNOWFLAKE ARCHITECTURE



# Snowflake Architecture vs. Competition

Multi-cluster, shared data, in the cloud

## Traditional Architectures



### Shared-disk

Additional capacity requires  
forklift upgrade

Reads/Writes at the same  
time cripples the system

Replication requires  
additional hardware



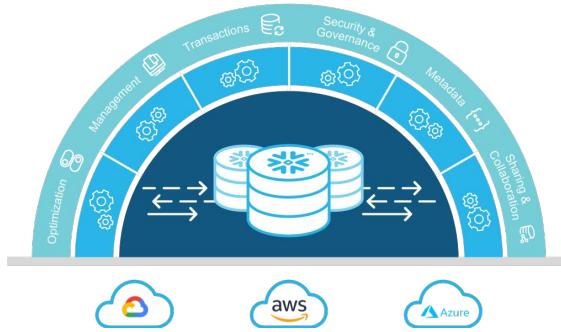
### Shared-nothing

Resizing cluster requires  
redistributing data. Shut down  
requires unloading.

Each cluster requires its own  
copy of data (ex: test/dev, HA)

Admin tasks needed  
to maintain optimal performance

## Snowflake



Best of both Shared-Disk and  
Shared-Nothing architectures

Instant elasticity, plus built-in security, data  
protection, zero-copy cloning and more

Cross-cloud operation



# HOW CAN SNOWFLAKE HELP YOU?





# Functional Architecture



Marketing  
Analytics / Reporting / BI



Data Science





Structured & Semi-Structured

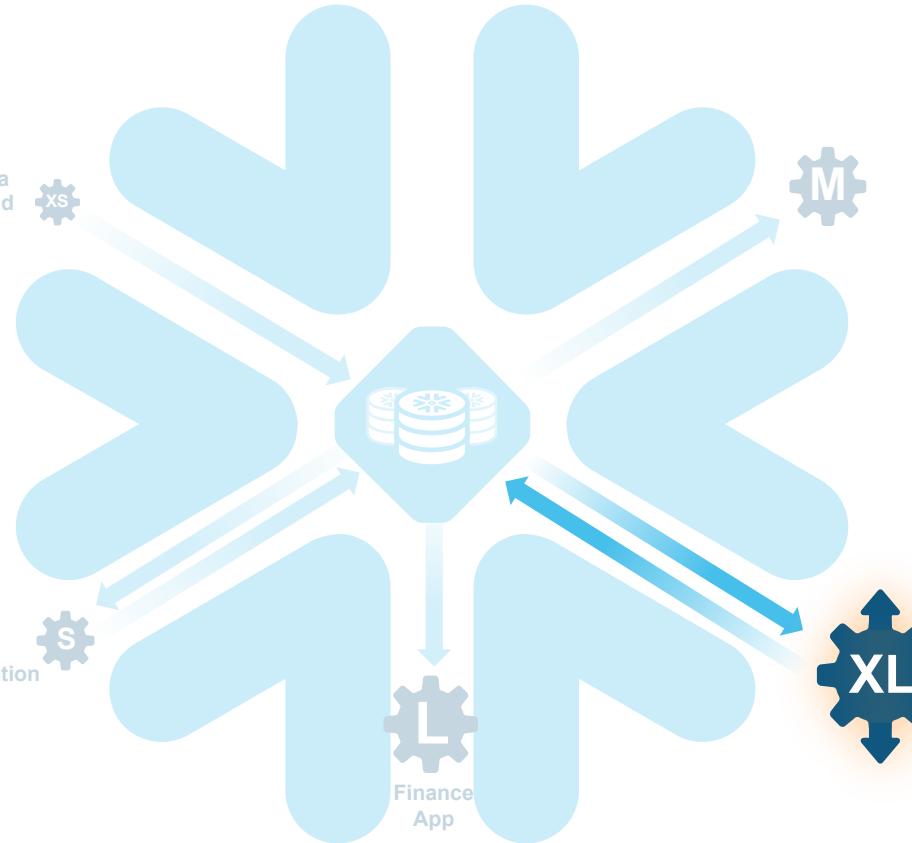


Data Load



# Functional Architecture

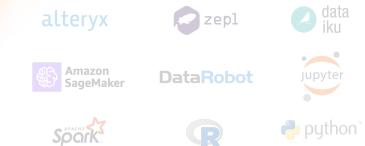
Data Transformation



Marketing  
Analytics / Reporting / BI

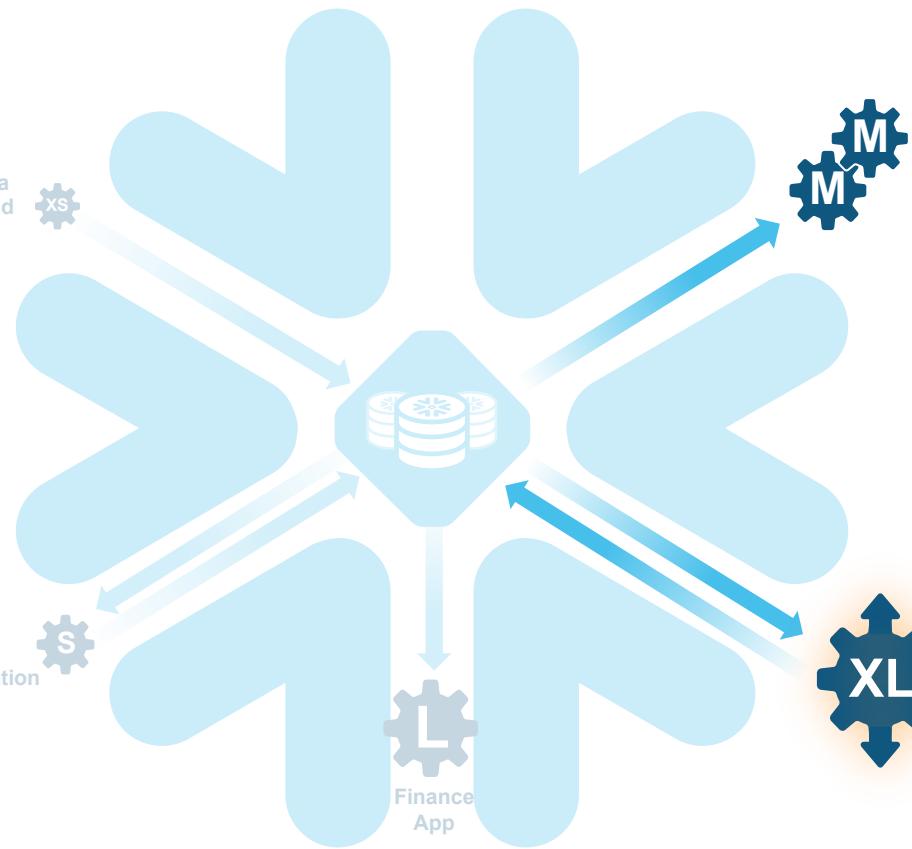


Data Science





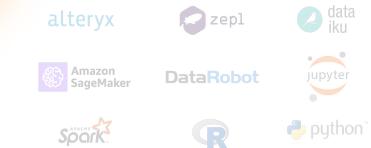
# Functional Architecture



**Marketing**  
Analytics / Reporting / BI



**Data Science**





Structured & Semi-Structured

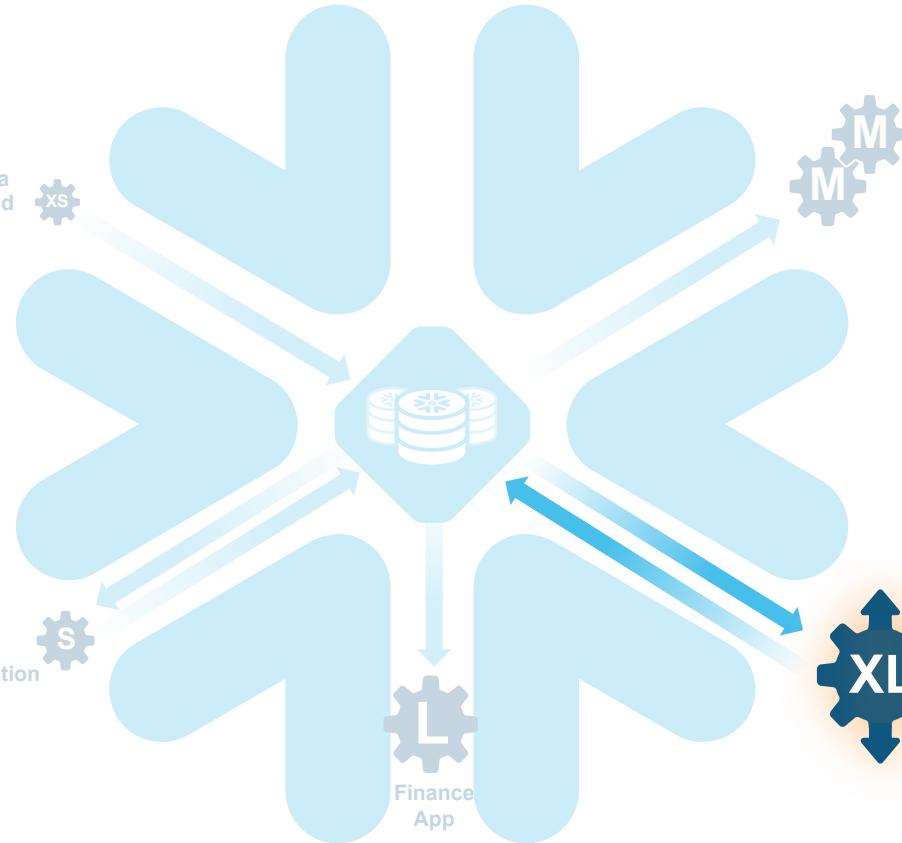


Data Load



# Functional Architecture

Data Transformation



Marketing  
Analytics / Reporting / BI

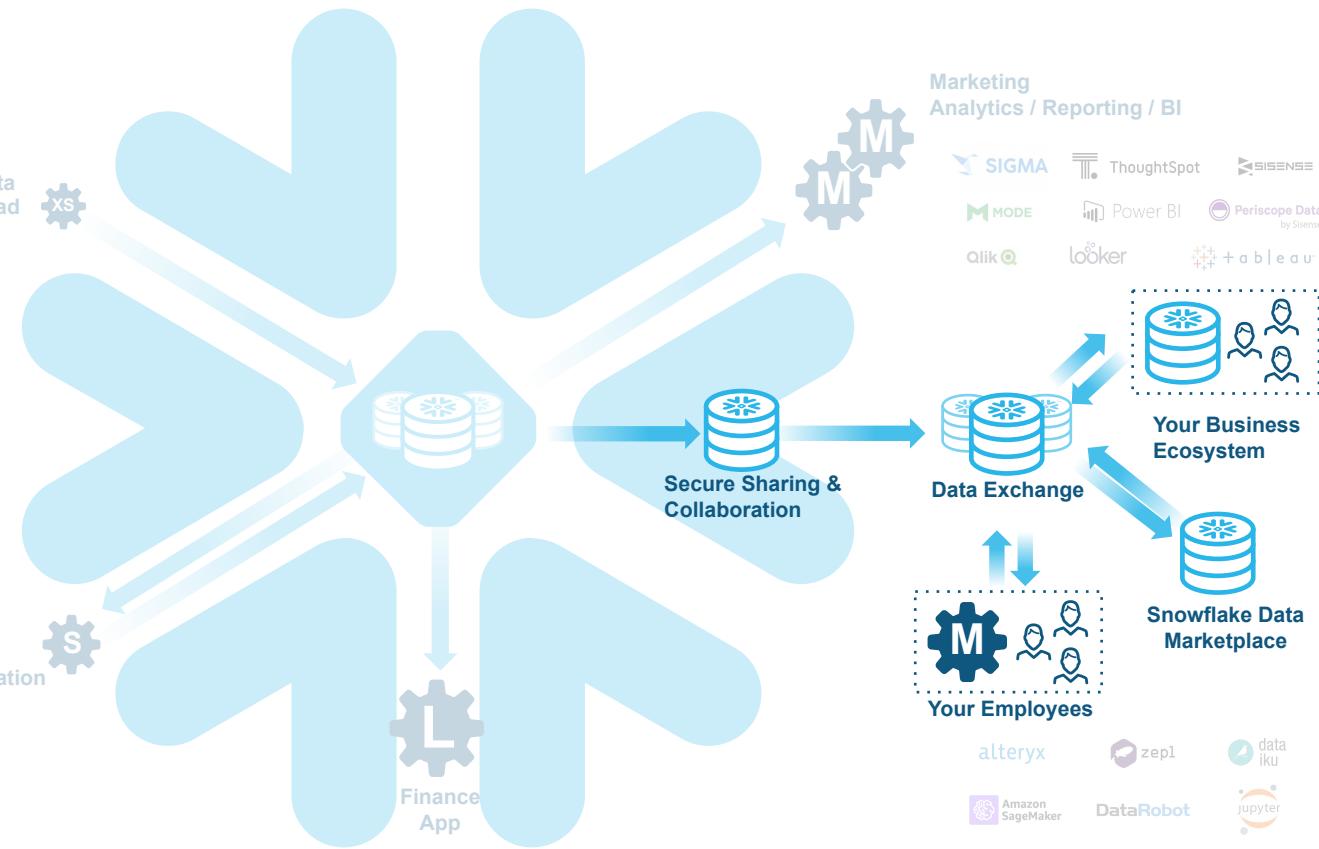


Data Science





# Functional Architecture



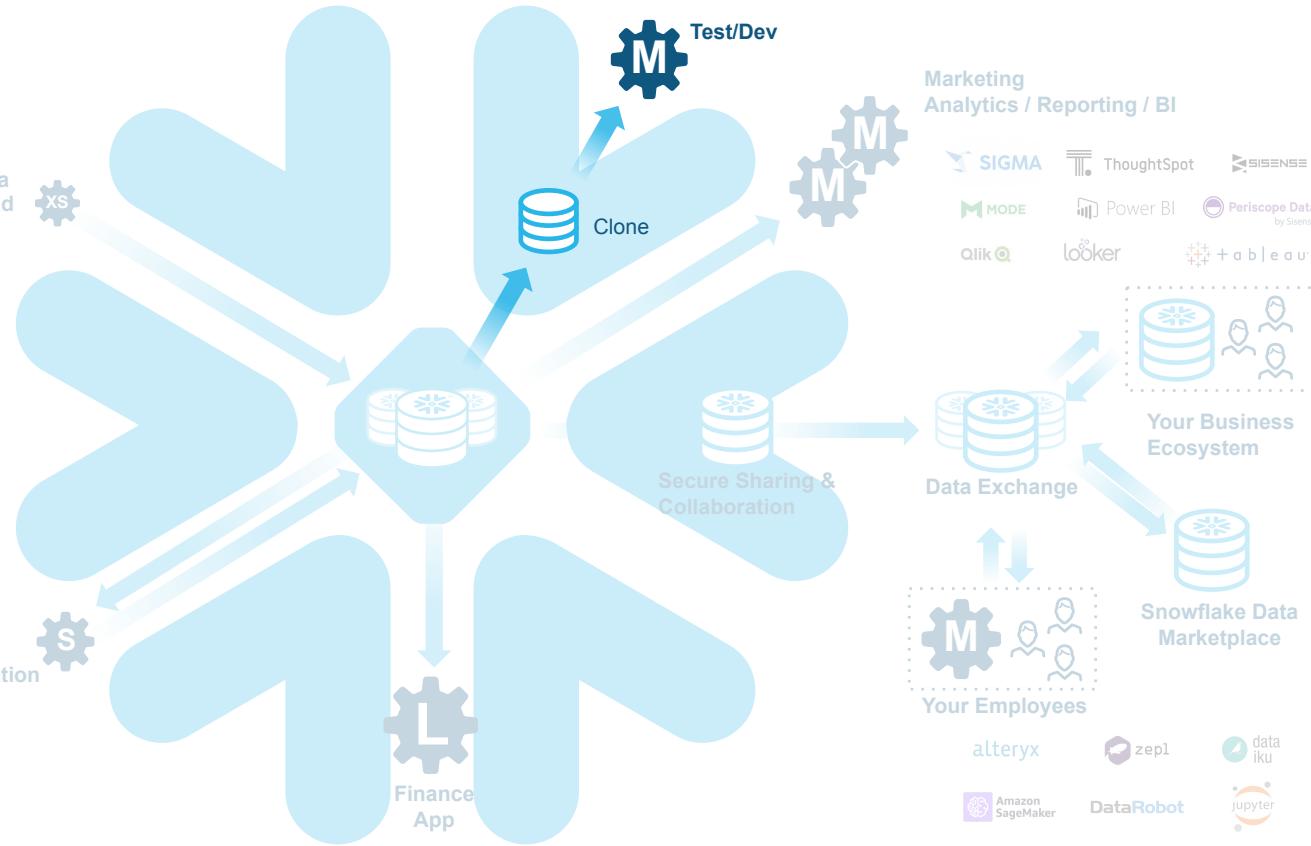


Structured & Semi-Structured



# Functional Architecture

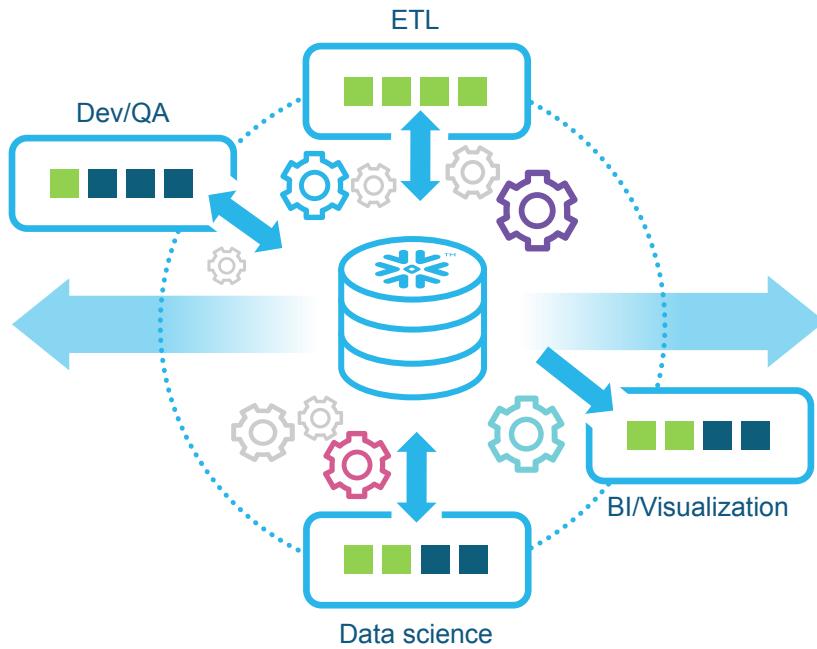
Data Transformation



# ELASTIC PERFORMANCE ENGINE



# ELASTIC PERFORMANCE ENGINE



## One engine for every workload

Simplify your architecture. Power complex pipelines, analytics, data science, interactive applications, and more.

## Leading performance and concurrency

Fast, reliable performance every time with no tuning or contention. Instantly and cost-efficiently scale to any amount of users, jobs, or data.

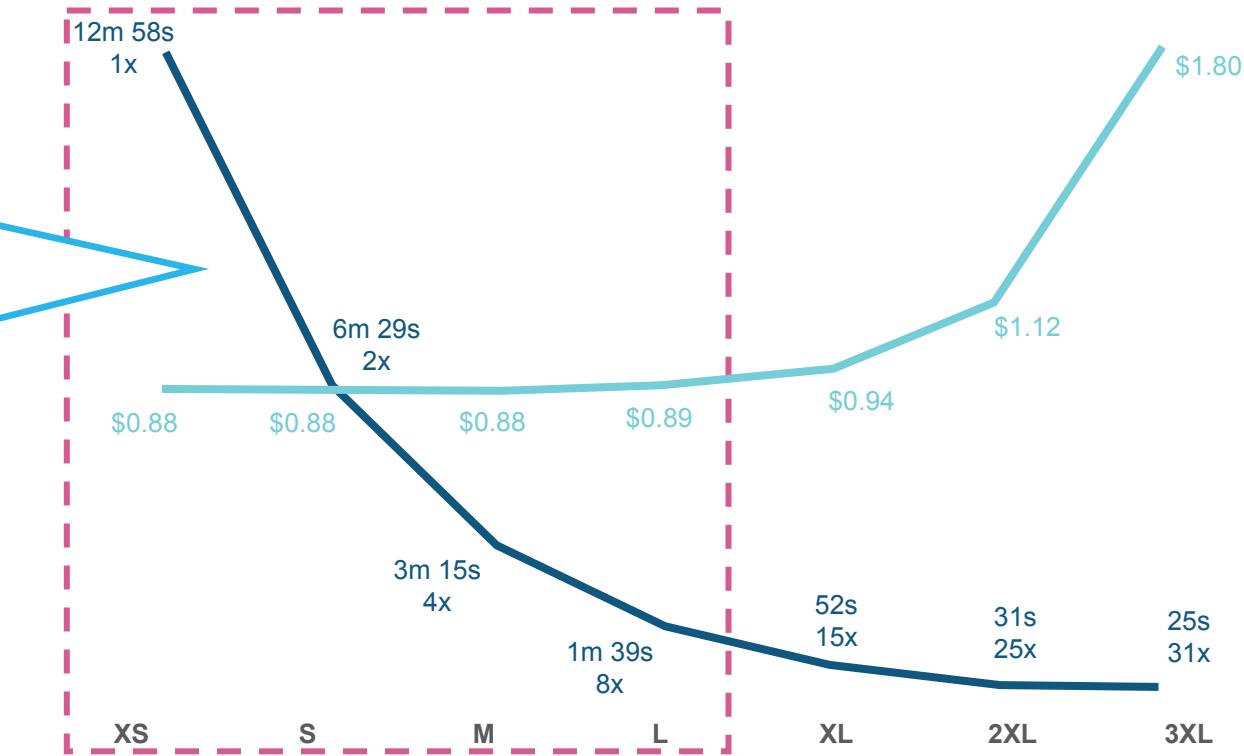
## Support any user or skillset

Get the accessibility of SQL, with the flexibility to support Java, Scala, and more. Run external tools directly for extended capabilities.



# SCALE UP - LOADING 1BN RECORDS

Doubling the number of servers halves the run time  
but you pay per-server, per second of compute  
so you get your answer  
**8x faster for the same cost**



# ALL TOGETHER - SCALE, ELASTICITY, COST



All three examples contain the  
**Same amount of work.**



Using scale up and scale out, total  
**run-time is significantly reduced.**



You pay per server, per second  
so they **all cost the same.**



# INTELLIGENT INFRASTRUCTURE



# INTELLIGENT INFRASTRUCTURE



## Snowflake Managed

MAINTENANCE & TUNING

MULTI-CLUSTER COMPUTE RESOURCES

ADMINISTRATION

NETWORKING & ENCRYPTION

DATA MANAGEMENT

CENTRALIZED STORAGE

### Automated and fully managed for you

Focus on what matters. Fully managed with automations that encrypt data, control access, and eliminate manual maintenance and troubleshooting.

### High availability, high reliability

Automate complex replication and failover cross-clouds and cross-regions. Stay up-and-running no matter what happens.

### Optimized costs for all data

Usage-based model paired with patented compression and fine-grained controls to right-size costs. Continual improvements for new efficiencies.



# ADAPTIVE CACHING



## Metadata

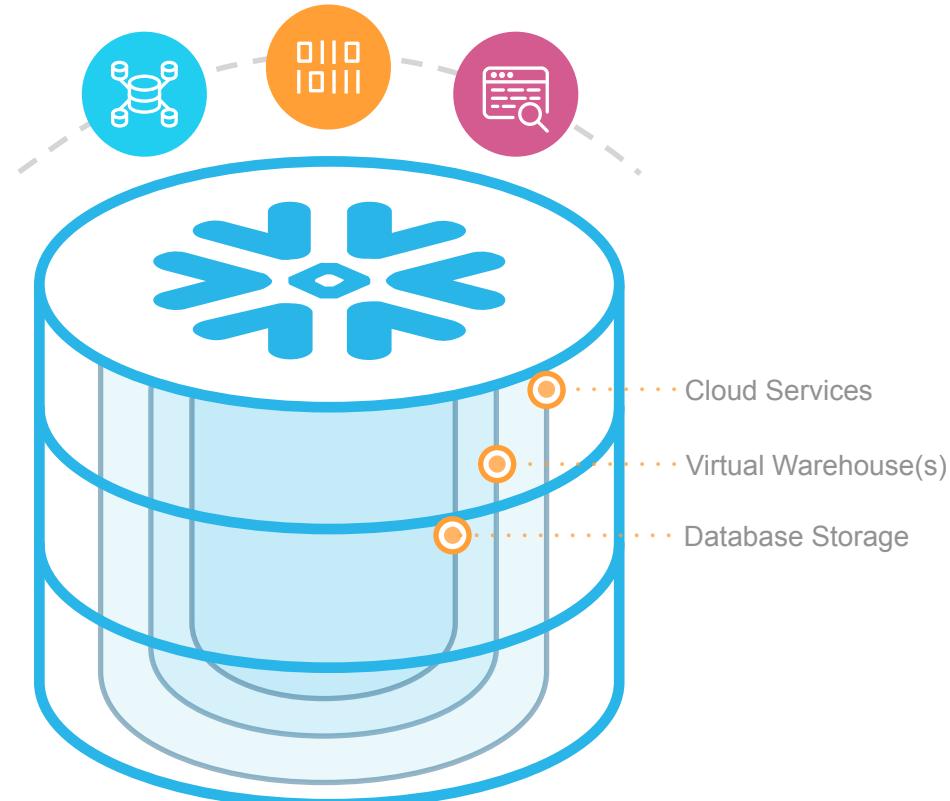
Metadata cached for fast access during query planning

## Data

Active working set transparently cached on virtual warehouse SSD

## Query results

Results sets cached for reuse without requiring compute (e.g., static dashboard queries)



# MICROPARTITIONS

Date	Product	Customer	Amount
Feb 14	Boots	Frank	\$150
Feb 14	Boots	Benoit	\$150
Feb 14	Skis	Thierry	\$300
Feb 14	Snowboard	Mike	\$250
Feb 15	Boots	Chris D	\$150
Feb 15	Skis	Denise	\$600
Feb 15	Snowboard	Shelly	\$250
Feb 16	Boots	Rob	\$150
Feb 16	Skis	Sunny	\$600
Feb 16	Snowboard	Chris K	\$250
Feb 16	Snowboard	Greg	\$750
Feb 16	Snowboard	Matt	\$750

## Profile Overview (Finished)

### Total Execution Time (3.583s)



### Total Statistics

#### IO

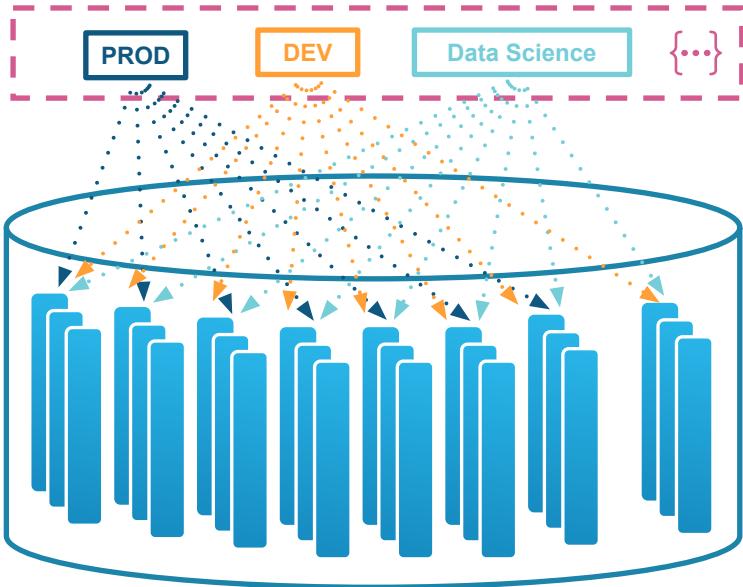
Scan progress	100.00%
Bytes scanned	179.58 MB
Percentage scanned from cache	100.00%
Bytes written to result	25.33 MB

#### Pruning

Partitions scanned	13
Partitions total	73



# ZERO-COPY CLONING



**The Metadata layer keeps track of every micro-partition file in every customer database.**

**Creating a DEV environment usually means copying the PROD database**

Limited to subset of full Prod

Up to 2x storage requirement

Periodic refreshes

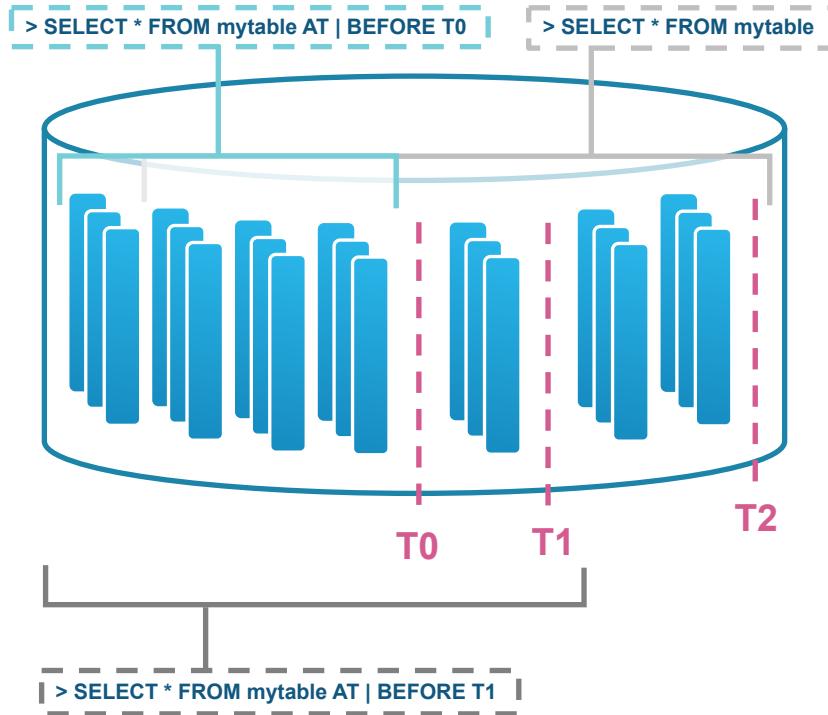
## Snowflake Zero-Copy Clones

Simply “point” to the same files

Consumes zero additional storage

Changes to either DB are isolated

# TIME TRAVEL



**T0 – Initial state of database**

T1 – update myTable set  
colX = Y where...

**T2 – ELT job loads new data**

**Previous versions of data  
automatically retained**

AT | BEFORE [ timestamp | statement | offset ]

CLONE AT | BEFORE to recreate a prior version

UNDROP recovers from accidental deletion

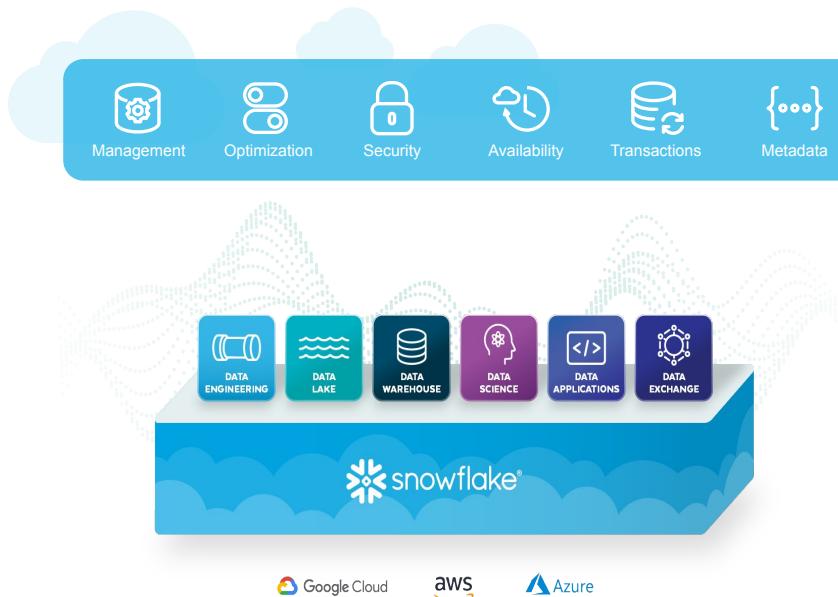
**Accessed via SQL extensions**

AT | BEFORE [ timestamp | statement | offset ]

CLONE AT | BEFORE to recreate a prior version

UNDROP recovers from accidental deletion

# SNOWFLAKE: SECURITY AT MULTIPLE LEVELS



Orchestrated security built into the fabric of the platform, automated controls in place for all functions, constant monitoring and analysis to detect and mitigate threats quickly

You have security features built in to protect the data you load in and use in Snowflake

Snowflake uses the industry standard “shared responsibility” model

Snowflake uses many sophisticated mechanisms to keep the platform safe and stable

# PROTECTING YOUR DATA IN SNOWFLAKE

## End-to-End Encryption

Always-encrypted client communications, plus integration with cloud provider private networking



## Fully Encrypted Storage

Data at rest is always encrypted while handled by the Snowflake drivers and systems



## Strong Authentication

Built in multi-factor, integration with your federated SSO, easy user management



## Full Auditing

Track every login, every transaction, every data transfer, and export to your security tools



## Role-Based Access Control

All objects, actions, and even compute usage can be controlled with roles



## Recovery

We give you options to ensure your data can be recovered in case of an accident or worse



[Snowflake Security Product Documentation](#)



# DATABASE REPLICATION & FAILOVER

## 1 Cross-Cloud & Cross-Region Replication

Business Continuity & Disaster Recovery  
Secure Data Sharing across regions/clouds  
Data Portability for Account Migrations

## 2 Zero Performance Impact on Primary

Asynchronous Replication

## 3 Reduced Data Loss

Incremental Refreshes

## 4 Instant Recovery

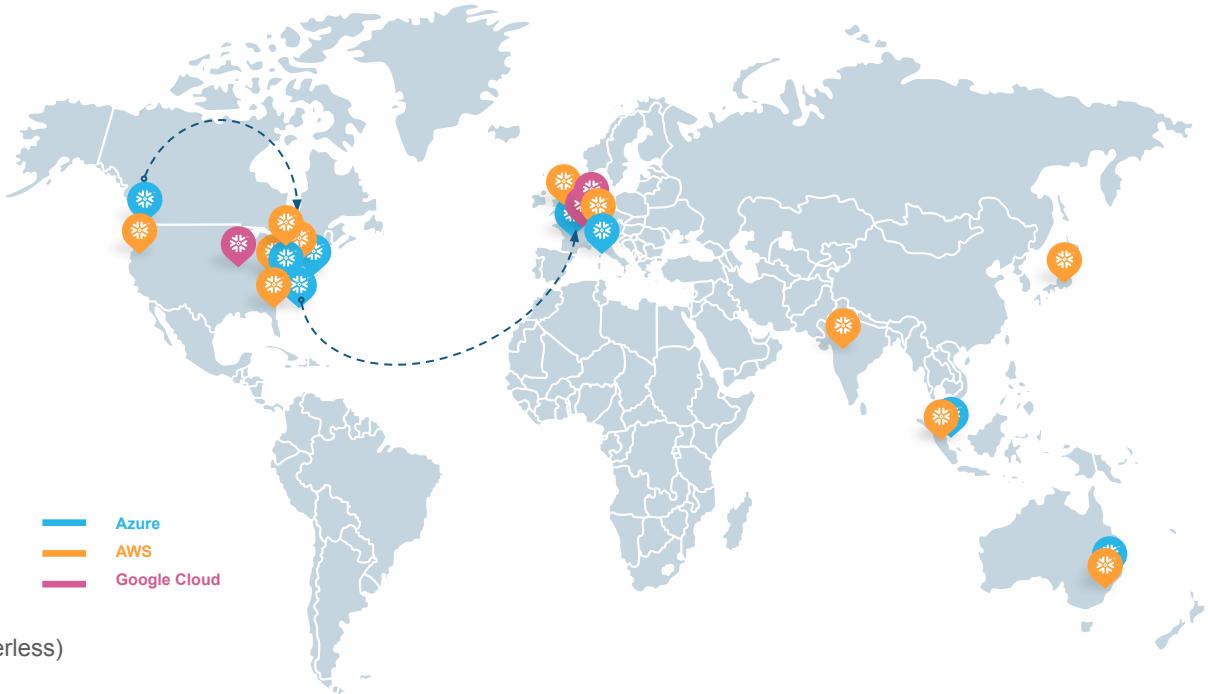
Read: Readable Secondary Databases  
Write: Database Failover

## 5 Secure

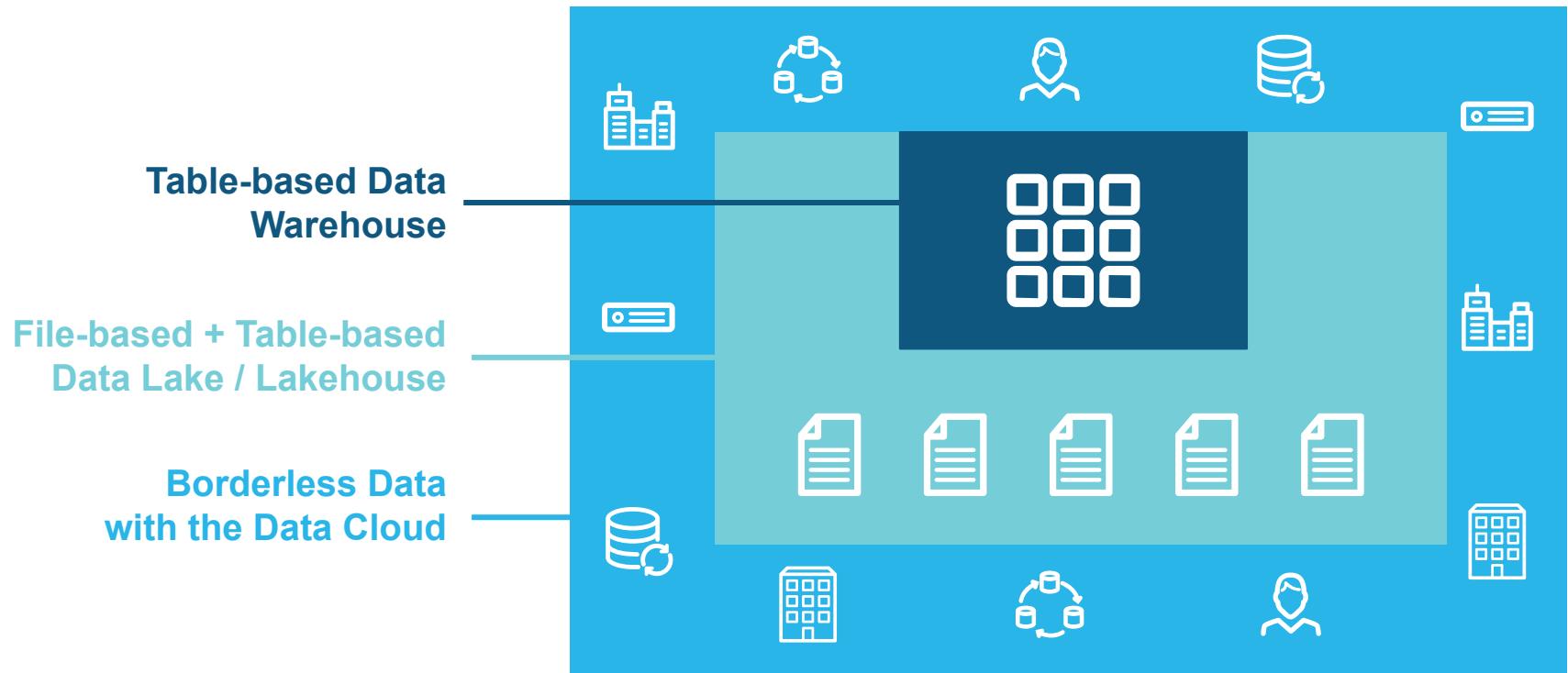
Data Encrypted at-rest & in-transit  
Tri-secret secure compatible

## 6 Cost Effective

Replication Costs: Data Transfer & Compute (serverless)  
Control which databases to replicate



# EVOLUTION OF DATA MANAGEMENT



# SNOWGRID



# SNOWGRID



Snowflake Regions



AWS



Azure



GCP

## Maintain global business continuity

Eliminate disruptions, deliver better experiences, and comply with changing regulations through unique cross-cloud, cross-region connectivity.

## Share data with no ETL or silos

Remove the barriers to data, regardless of cloud, region, workload, or organizational domains. Get instant access and distribution through a single copy of data.

## Cross-cloud governance controls

Simplify governance at scale with flexible policies that follow the data for consistent enforcement across users and workloads.

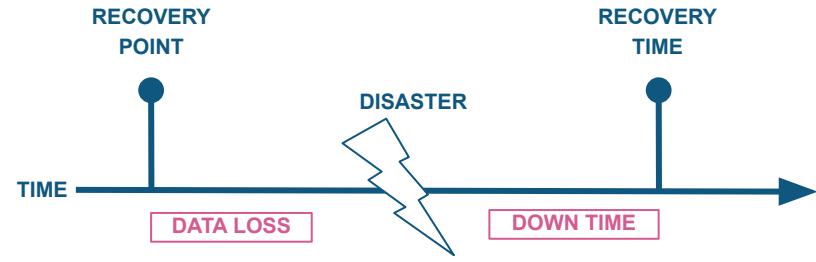
## Tap into the extended ecosystem

Enrich insights with a network of third-party data. Discover and run new functions for extended workflows.



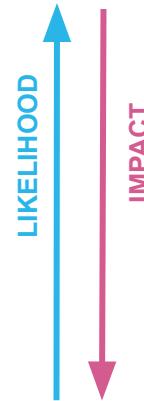
# WHAT DO WE MEAN BY BUSINESS CONTINUITY?

- 1 Identify critical applications
- 2 Analyze business impact
- 3 Create plan & objectives



# PREPARING FOR DIFFERENT TYPES OF FAILURES

Types of Failures	
<b>Customer Error</b>	Accidental data corruption or deletion due to human error
<b>Single Instance Failure</b>	Virtual Machine fails due to hardware or system error
<b>Zone Failure</b>	Loss of service availability in a single zone
<b>Region Failure</b>	Loss of service availability in multiple zones in a region
<b>Multi-Region Failure</b>	Loss of service availability in multiple regions of a cloud provider



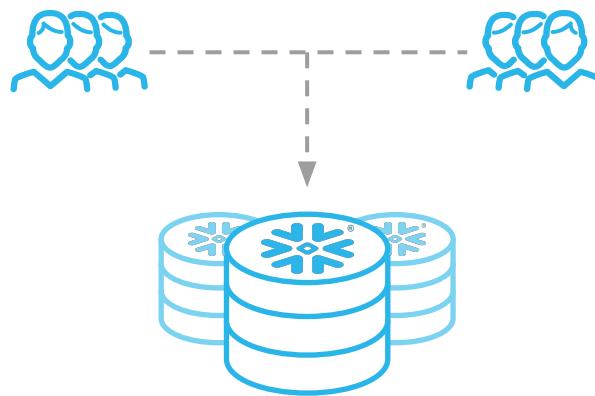
# SNOWFLAKE ENABLES BUSINESS CONTINUITY

Failure	Mitigation
<b>Customer Error</b>	<b>Snowflake Features</b> Time Travel Fail-safe
<b>Single Instance Failure</b>	<b>Snowflake Built-in Redundancy</b> Triple-redundancy for critical services Automatic retries for failed parts of a query
<b>Zone Failure</b>	<b>Snowflake Built-in Redundancy</b> Using Availability Zones on AWS, Azure, GCP Using Availability Sets on Azure
<b>Region Failure</b>	<b>Snowflake Features</b> Cross-Region Database Replication Cross-Region Database Failover
<b>Multi-Region Failure</b>	<b>Snowflake Features</b> Cross-Cloud Database Replication Cross-Cloud Database Failover



# SNOWGRID UNLOCKS DATA SHARING

## Secure Data Sharing



Live, ready-to-query data

## Differentiators

- ✓ Live, ready-to-query data
- ✓ No ETL
- ✓ Share personalized data
- ✓ Share business logic
- ✓ Cross-cloud and cross-region
- ✓ Governed, revocable access



# SHARE AND COLLABORATE IN THE DATA CLOUD

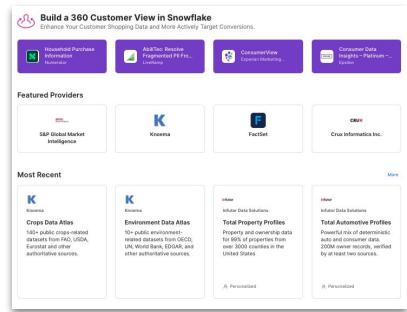
## SECURE DATA SHARING FOR EVERY SCENARIO

### DISCOVER AND BE DISCOVERED IN THE DATA CLOUD

Access data and services from 150+ providers

### SNOWFLAKE DATA MARKETPLACE

Market and deliver your products to customers



### SHARE ACROSS YOUR BUSINESS ECOSYSTEM

#### DIRECT SHARE

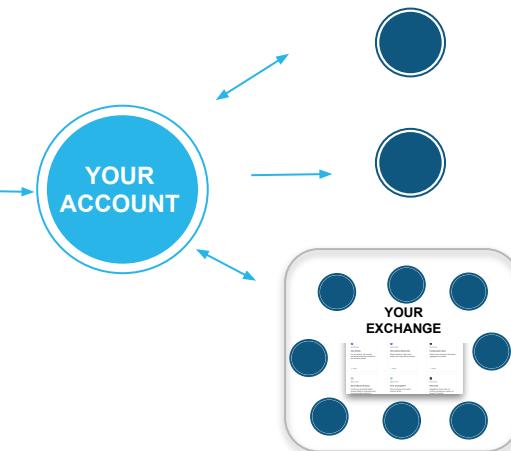
Share with other Snowflake customers

#### READER ACCOUNTS

Share with companies not yet on Snowflake

#### DATA EXCHANGE

Administer group sharing and data discovery across business units



Powered by Secure Data Sharing: Live, ready-to-query data, cross-cloud and cross-region. No ETL.

# EXTENSIBILITY



# SNOWPARK



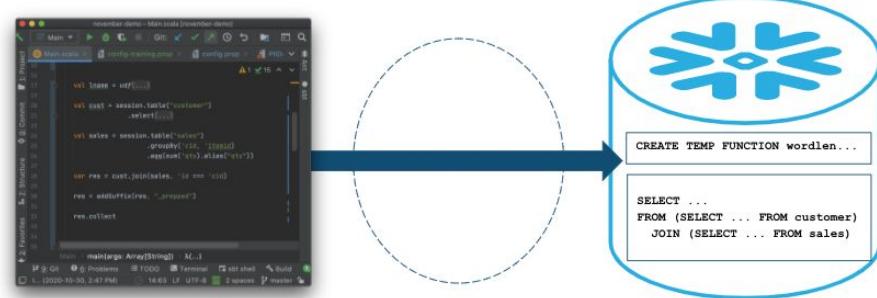
A new developer experience that allows you to write Snowflake code in your preferred way, and execute it directly within Snowflake

## Example Use Cases:

- Data transformation
- Data preparation and feature engineering
- ML Scoring / Inference to operationalize ML models in data pipelines
- ELT systems
- Data apps

## Allows coders to:

- Write in your language with your preferred tool
- Easily complete and debug data pipelines with familiar constructs such as DataFrames, and bring in third-party libraries.
- Eliminate the need to have other processing systems, and run directly on Snowflake.

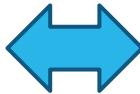


*Snowpark pushes all of its operations directly to Snowflake without Spark or any other intermediary.*

*\*Support for Scala is in Public Preview. Plan to add other languages in future.*

# SNOWPARK DATAFRAME

Dataframe



Query

*functional languages*

Java

Scala

Python

SQL

*declarative language*

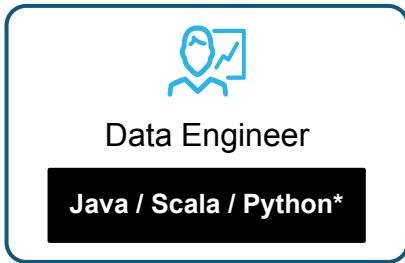
```
val df =  
    session.table("products")  
    .filter('id === 1)
```

```
select * from products  
where id = 1;
```

- Easy to debug/concise syntax
- More efficiency with less coding required
- No bindings with external processing engine (e.g. Spark) required



# WHY SNOWPARK?



```
val df =  
    session.table("products") .  
    filter(col("id") === 1)
```

```
select * from products  
where id = 1;
```



**Snowpark:** Native Dataframes  
for Snowflake



# JAVA FUNCTIONS

Transform and augment your data using custom logic running right next to your data, with no need to manage a separate service.

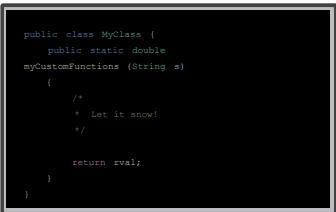
## Example Scenarios:

- ML Scoring
- Apply custom code
- Use third-party libraries

## Benefits:

- Developers can build functionality into Snowflake using the popular Java language and libraries.
- Users can access this functionality as if it were built into Snowflake.
- Administrators can rest easy: data never leaves Snowflake.

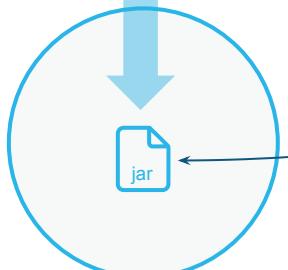
### 1. Build with your tools



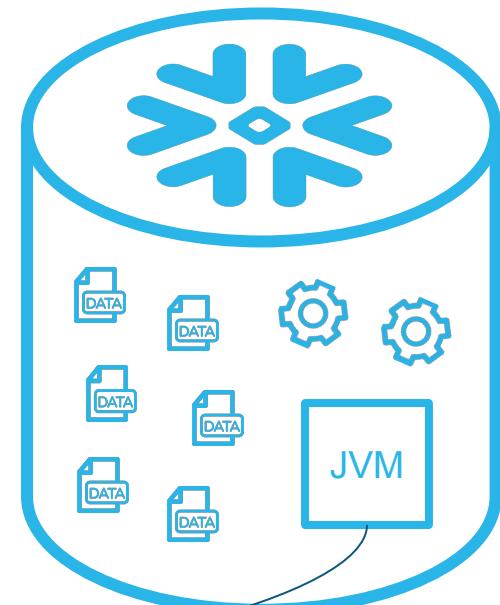
```
public class MyClass {
    public static double myCustomFunctions (String s)
    {
        /*
         * Let it snow!
         */

        return rval;
    }
}
```

### 2. Deploy .jar to Snowflake stage



### 3. Bind and use in Snowflake



# EXTERNAL FUNCTIONS

## Bind SQL Functions to Implementations Outside Snowflake

- V1: Batched scalar functions
- Mediated via API Gateway
- Usable wherever functions are used

## Example Scenarios:

- Custom Lambda
- ML Scoring
- Geocoding

## ACCOUNTADMIN in Control of Security

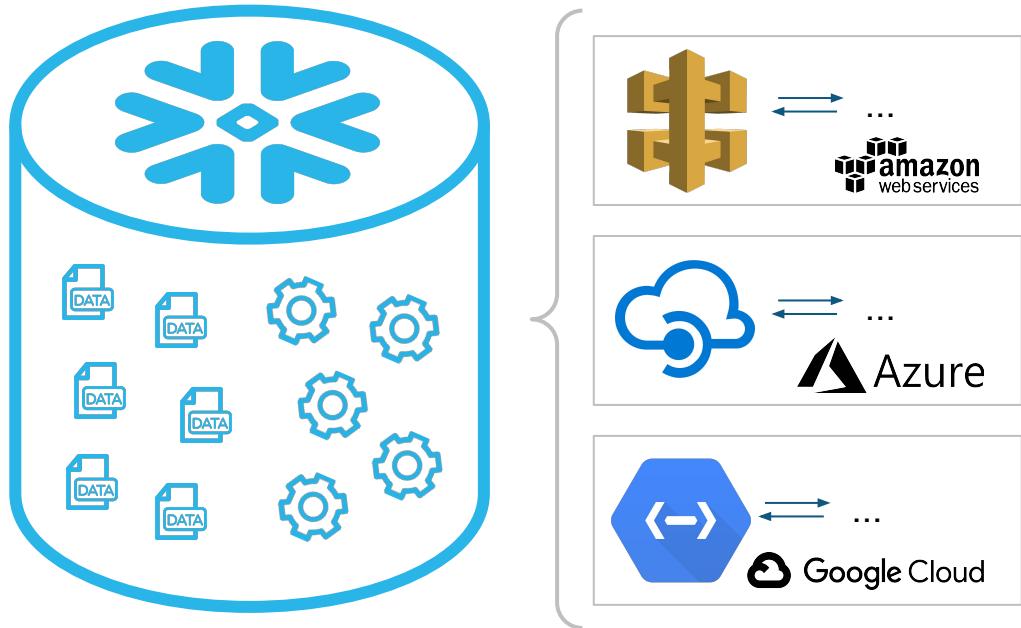
- Must explicitly whitelist endpoints

## GA:

- AWS, Azure

## Public Preview:

- Private Endpoint
- GCP



# ECOSYSTEM

## Client drivers

Feature	Description	Current State	Target
Arrow Format for Result Sets - Go	Improve fetch performance. (JDBC, Python Done)	Public Preview	Q3 2021
PUT/GET - node.js, .NET	Add PUT/GET support for these drivers	NA	Q3 2021
Bulk Insert Optimization - Python, .NET, node.js	Bulk array binding via automated streaming PUT + COPY	NA	Q3 2021
Python parallel result fetch	Support parallel/distributed fetching of result chunks	Private Preview	Q3 2021

**Supported Clients: JDBC, ODBC, Python, SnowSQL, .NET, node.js, Golang, PHP**



# SQL API

## What is the SQL API?

A new REST interface for submitting SQL statements.

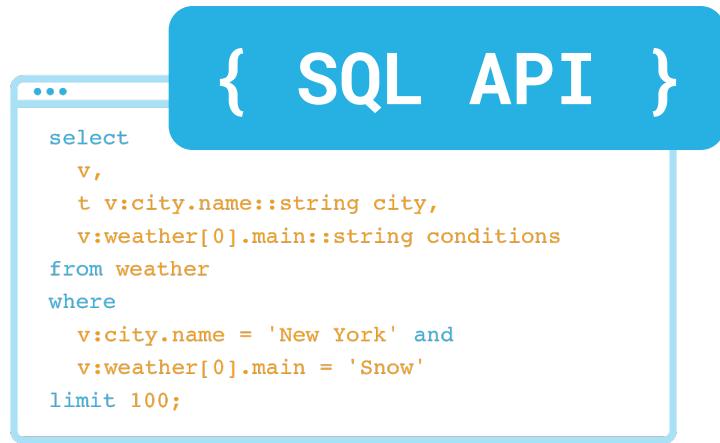
Supports all standard DDL, DML, and Queries

## Why SQL API?

- Drivers cannot always be loaded/used
- Developer preference
- Ease of migration

## With the SQL API, Developers can:

- Build custom REST based applications
- Easily migrate existing applications built for APIs
- Integrate with applications that provide a REST Interface (ServiceNow, PowerApps, etc)
- Integrate with resource constrained environments



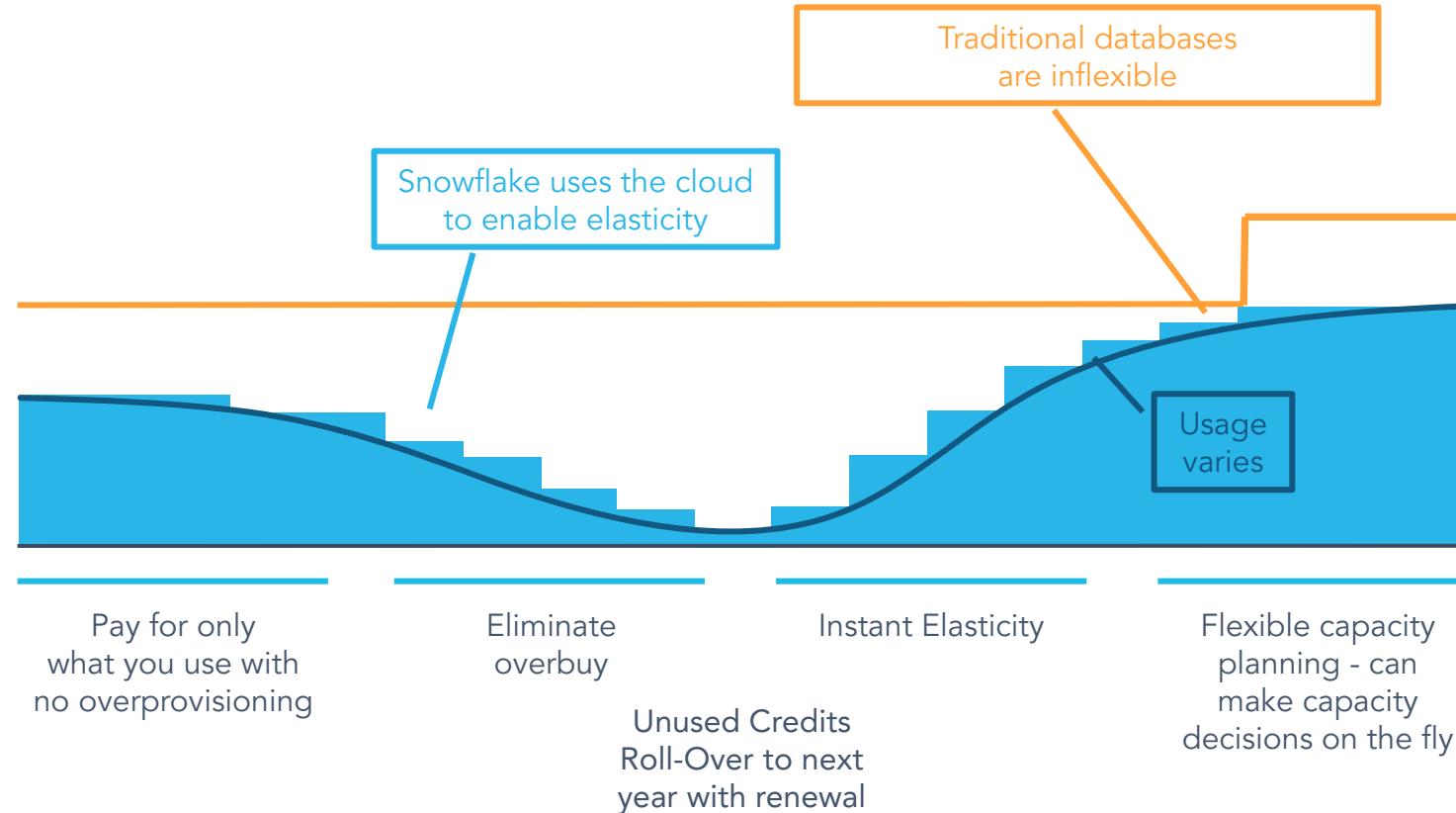
```
... { SQL API }
select
    v,
    t v:city.name::string city,
    v:weather[0].main::string conditions
from weather
where
    v:city.name = 'New York' and
    v:weather[0].main = 'Snow'
limit 100;
```

# PRICING METHODOLOGY





# Only pay for what you use



## Snowflake Pricing Explained

- WHAT IS A SNOWFLAKE CREDIT? – Think: '*prepaid token*'

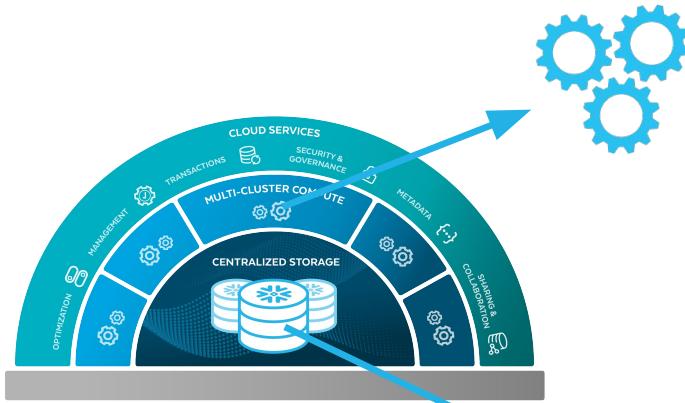
Snowflake Credits are used to pay for the processing time of a Virtual Warehouse. Snowflake Credits, a unit of measure, are consumed only when a Virtual Warehouse is running. When a warehouse is not running (auto-suspend), it does not consume any Snowflake Credits. Snowflake Credits are consumed at different rates based on the size of the running warehouse.

- VIRTUAL WAREHOUSE SIZES

Snowflake supports a wide range of Virtual Warehouse sizes: X-Small, Small, Medium, Large, X-Large, 2X-Large, 3X-Large and 4X-Large (and can go up to 5XL and 6XL). The size of the Virtual Warehouse determines how fast queries will run. The different sizes of Virtual Warehouses consume Snowflake Credits at the following rates for each hour the warehouse is operational, billed by the second with a one minute minimum:

VIRTUAL WAREHOUSE TYPES								
	XS	S	M	L	XL	2XL	3XL	4XL
Nodes	1	2	4	8	16	32	64	128

# SNOWFLAKE PRICING



## COMPUTE

Virtual Warehouse Credits Per Hour							
XS	S	M	L	XL	2XL	3XL	4XL
1	2	4	8	16	32	64	128

\*Compute credits are billed per second of use, after 1 minute

## STORAGE

Database Storage
AZURE/AWS - \$23 per compressed TB/Month
GCP - \$20 per compressed TB/Month



# Quick Starts

<https://quickstarts.snowflake.com/?cat=resource+optimization>





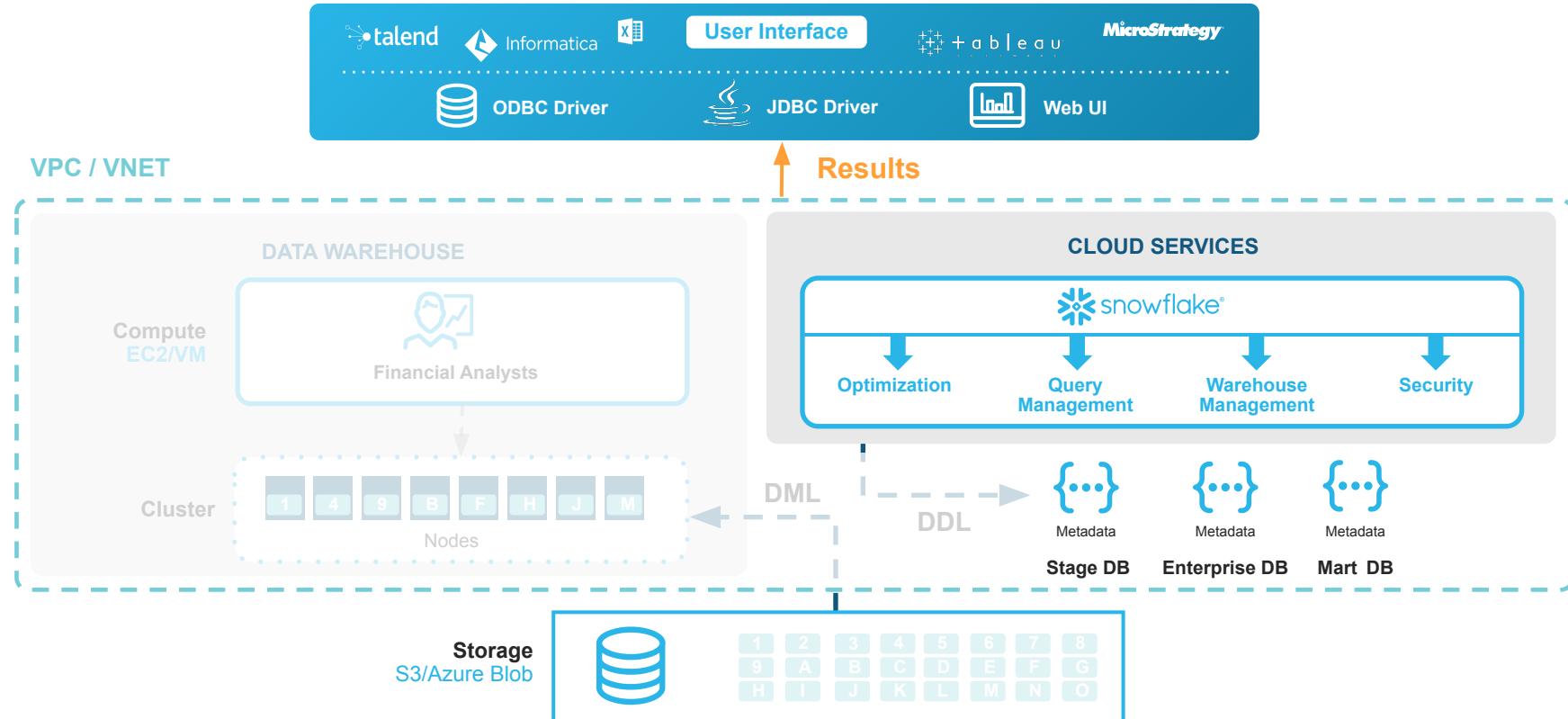
THANK YOU



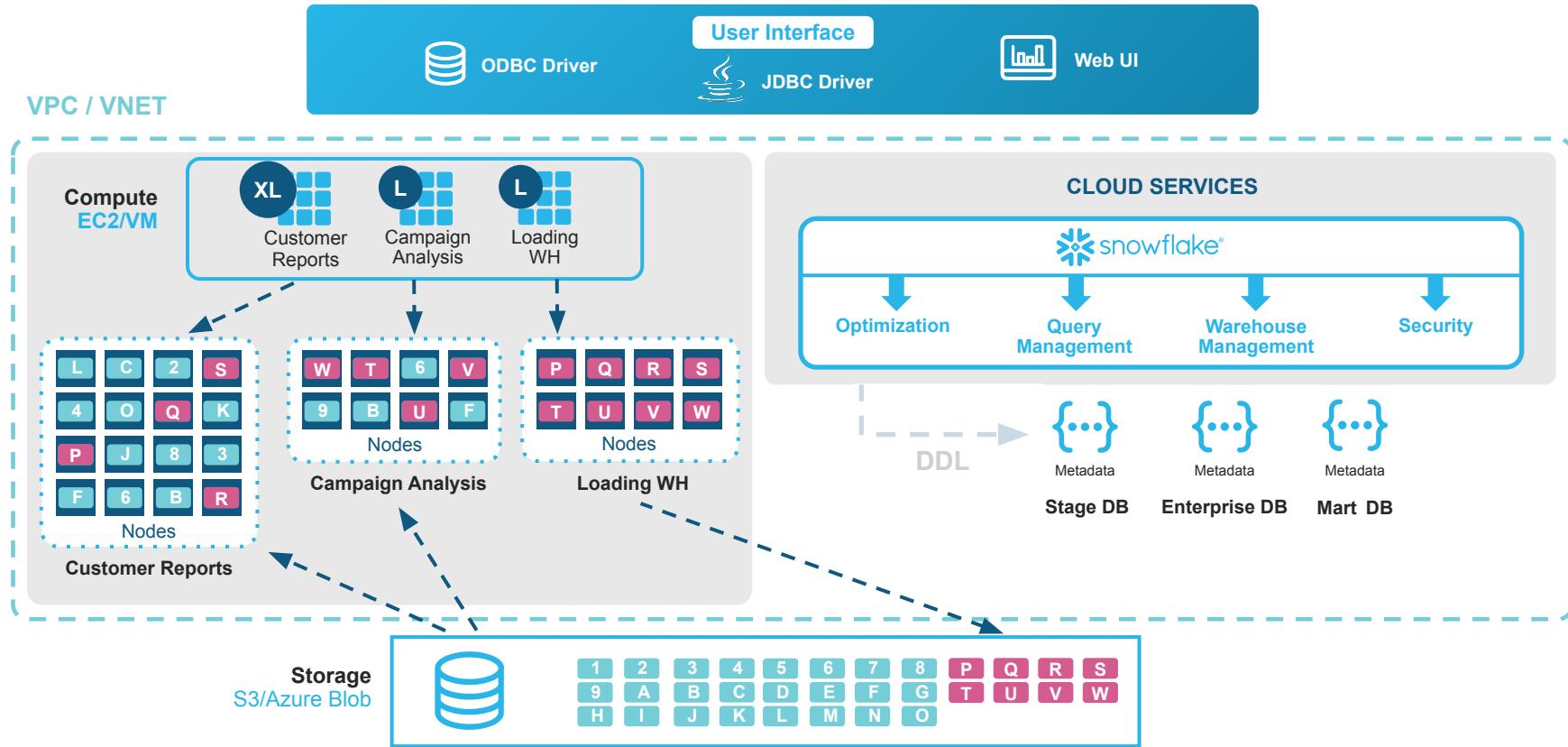
# ADDITIONAL FEATURE SLIDES



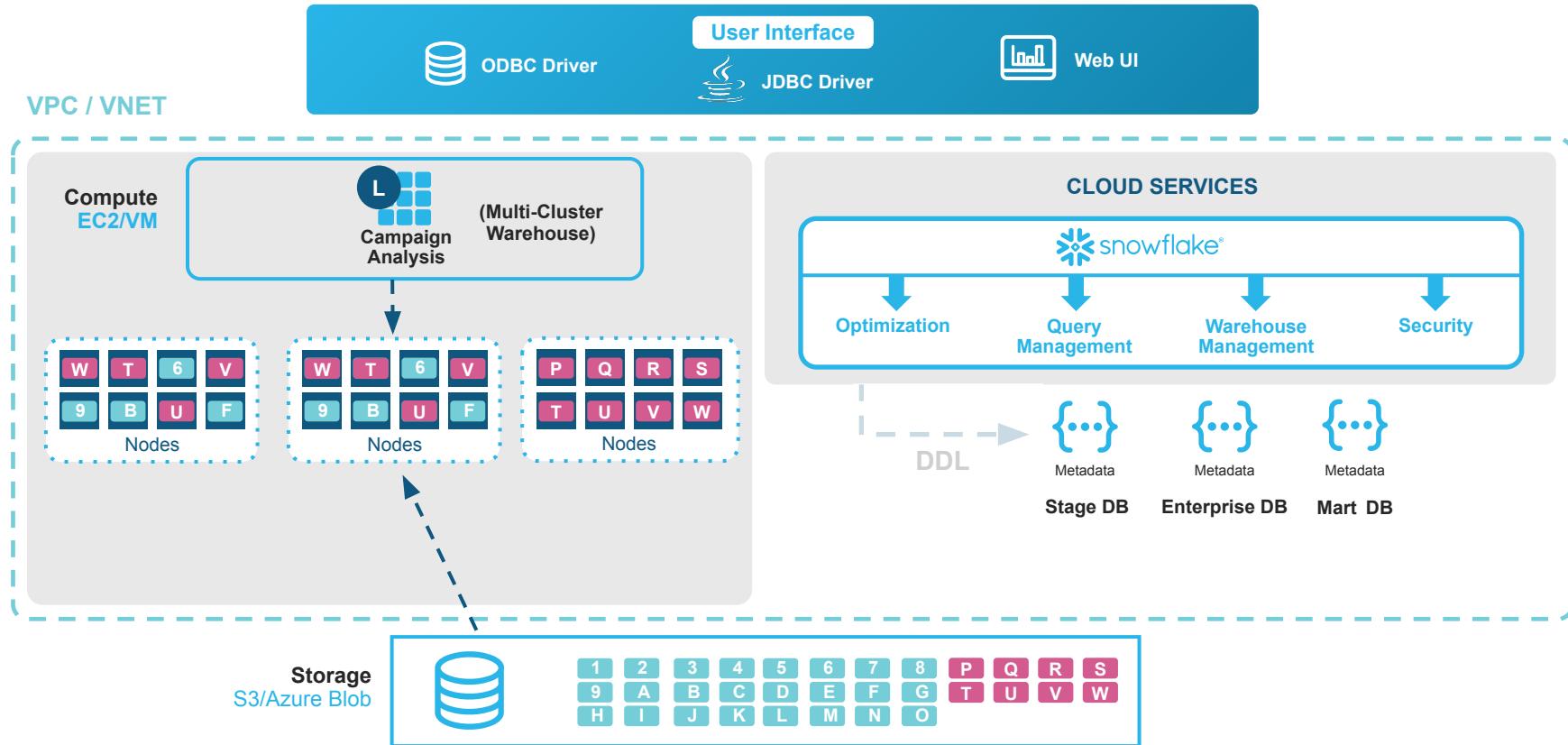
# SNOWFLAKE ARCHITECTURE



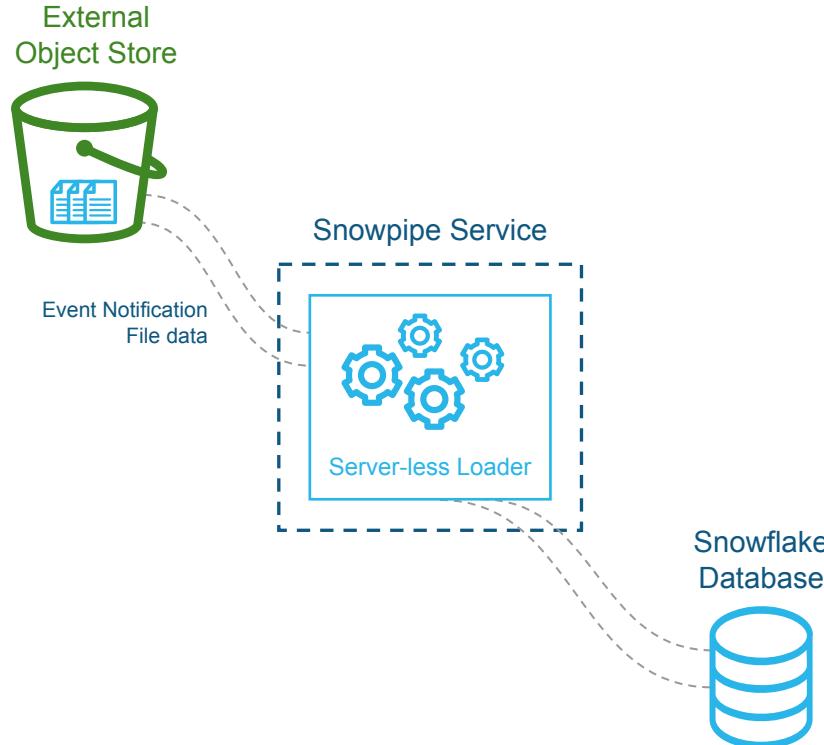
# SNOWFLAKE ARCHITECTURE



# SNOWFLAKE ARCHITECTURE



# SNOWPIPE



Account

Billing & Usage   Reader Accounts

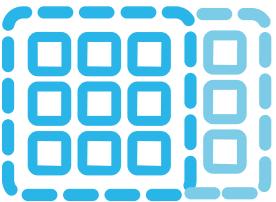
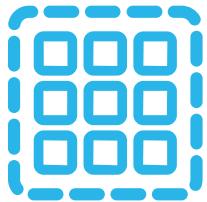
- Billing & Usage

Warehouses	Credits Used	Average Storage Used
8	29.47	47.927 GB

	Warehouse Name	Credits Used
●	LOAD_WH	17.58
●	PLAYWH	10.20
●	BI_MEDIUM_WH	1.64
●	XSMALL	0.03
●	AUTOMATIC_CLUSTERING	0.01
●	<b>SNOWPIPE</b>	<b>0.01</b>
●	CLOUD_SERVICES_ONLY	0.00
●	DEMO_WH	0.00



# STREAMS



```
CREATE STREAM  
s_sales_str ON TABLE  
l_sales;
```

l_sales		
key	number	number
cust_id	number	number
amount		

↓

l_sales		
key	number	number
cust_id	number	number
amount		
METADATA\$ACTION		
METADATA\$ISUPDATE		
METADATA\$ROWID		

Adding a stream to a table appends three metadata columns that can be queried.

These columns track the CDC records and their type: appends, deletes, or both (updates = inserts + deletes).

Little additional storage is required, as the stream is a logical pointer to the table's existing Time Travel micro-partitions.

# TASKS

```
create or replace task t_sales_task
  warehouse = xsmall_vvh
  schedule = '1 minute' as
  insert into f_sales (
    select key, cust_id, amount
    from s_sales_stream
  where metadata$action = 'INSERT');
```

# ACTIONS

## Stream:

Identifies Inserts, Updates, Deletes

## Task:

Executes on a

- **Schedule** (CRON)
- **Interval** (Minutes/seconds)
- **Predecessor** - child task runs after a parent task completes; can daisy chain

Single SQL statement or a stored procedure call

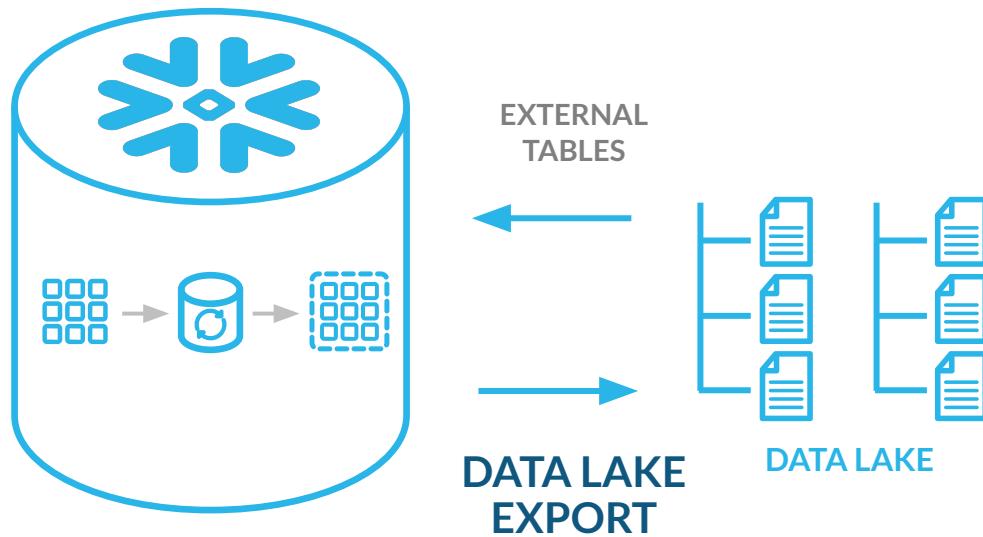
Can process a stream

Check for one or more streams to be populated before executing



# DATA LAKE EXPORT

Export data from Snowflake table to files on data lake in partitioned folders



```
COPY INTO @processed_data from  
processed_table  
partition by('dt='|| date_col)  
file format = (type = parquet)  
max_file_size = 512000000;
```

- Export large parquet files with 128MB row groups
- Export with hive style partitioning