

Quickstarts Used

- Tasty Bytes
 - https://quickstarts.snowflake.com/guide/tasty_bytes_introduction/index.html#0
- Snowflake for Data Lake
 - https://quickstarts.snowflake.com/guide/vhol_data_lake/index.html?index=..%2F..index#0
- Snowpark Basics
 - <https://github.com/snowflakecorp/techup-fy24snowday-snowpark>
- Intro to Machine Learning with Snowpark
 - https://quickstarts.snowflake.com/guide/intro_to_machine_learning_with_snowpark_ml_for_python/index.html?index=..%2F..index#0
- Cybersyn Streamlit in Snowflake
 - https://quickstarts.snowflake.com/guide/getting_started_with_streamlit_in_snowflake/index.html?index=..%2F..index#0



One Platform that Makes the Complex Easy



- ▶ Faster Time-to-Market
- ▶ Lower TCO
- ▶ Minimized Security Risk



Building a Data Foundation from the Ground Up



DATATUNE

NASHVILLE
2024

Thank you sponsors!

Gold Sponsor



((CENTRIC))

~~FLAGRANT~~

Couchbase

conExess



Agenda

- Intro to Snowflake
- Our Approach for Today
- Creating Your Snowflake Account
- Account Setup
 - Initial Configuration
 - Setting up Lab 1 - Tasty Bytes
- Basic Data Manipulation
 - Loading data
 - Querying Data
 - Transforming Data
- Advanced Data Types - Semi-Structured



Agenda

- Acquiring Additional Data
 - Marketplace
- Intro to Geospatial
- Monitoring - part 1

BREAK

- Snowpark Python
- Basic Data Transformation & Pipelines
 - CTAS
 - Dynamic Tables



Agenda

- Advanced Analytics
 - Forecasting, Anomaly Detection
- AI for All!
 - Snowflake Cortex
- Integrations
 - External Network Access
- *Data Lakes / Lakehouse*
- *Advanced Data Engineering*
 - *Classification using Python*
 - *ML encoding*
- *Build a Data-Driven AI App w/Streamlit*



SAFE HARBOR AND DISCLAIMERS

Other than statements of historical fact, all information contained in these materials and any accompanying oral commentary (collectively, the “Materials”), including statements regarding (i) Snowflake’s business strategy and plans, (ii) Snowflake’s new or enhanced products, services, and technology offerings, including those that are under development or not generally available, (iii) market growth, trends, and competitive considerations, and (iv) the integration, interoperability, and availability of Snowflake’s products, services, or technology offerings with or on third-party platforms or products, are forward-looking statements. These forward-looking statements are subject to a number of risks, uncertainties and assumptions, including those described under the heading “Risk Factors” and elsewhere in the Annual Reports on Form 10-K and the Quarterly Reports on Form 10-Q that Snowflake files with the Securities and Exchange Commission. In light of these risks, uncertainties, and assumptions, the future events and trends discussed in the Materials may not occur, and actual results could differ materially and adversely from those anticipated or implied in the forward-looking statements. As a result, you should not rely on any forward-looking statements as predictions of future events.

Any future product or roadmap information (collectively, the “Roadmap”) is intended to outline general product direction; is not a commitment, promise, or legal obligation for Snowflake to deliver any future products, features, or functionality; and is not intended to be, and shall not be deemed to be, incorporated into any contract. The actual timing of any product, feature, or functionality that is ultimately made available may be different from what is presented in the Roadmap. The Roadmap information should not be used when making a purchasing decision. In case of conflict between the information contained in the Materials and official Snowflake documentation, official Snowflake documentation should take precedence over these Materials. Further, note that Snowflake has made no determination as to whether separate fees will be charged for any future products, features, and/or functionality which may ultimately be made available. Snowflake may, in its own discretion, choose to charge separate fees for the delivery of any future products, features, and/or functionality which are ultimately made available.

© 2024 Snowflake Inc. All rights reserved. Snowflake, the Snowflake logo, and all other Snowflake product, feature and service names mentioned in the Materials are registered trademarks or trademarks of Snowflake Inc. in the United States and other countries. All other brand names or logos mentioned or used in the Materials are for identification purposes only and may be the trademarks of their respective holder(s). Snowflake may not be associated with, or be sponsored or endorsed by, any such holder(s).

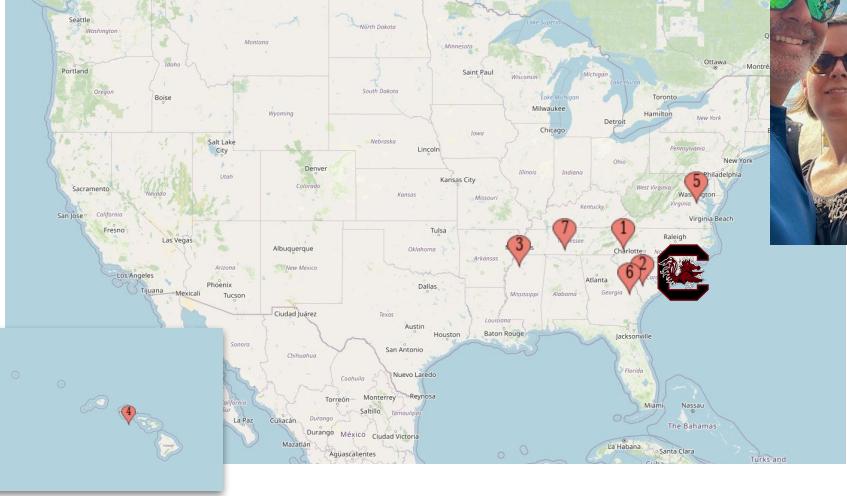


Jeremy Lemmon - About Me

25+ year industry veteran

Broad Perspective

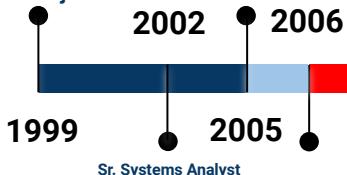
- Applications
- Infrastructure
- Data
- BUSINESS



Sr. Software Engineer /
Technical Project Lead



Chief Engineer



Vice President / COO



Deputy Chief Engineer



2017
Staff Solution Engineer

2022



2019
Global Accounts
Solutions Consultant

2022
Senior Sales Engineer

ORACLE
Senior Sales Engineer



Enough about me.... Let's hear from you

1. Your Name
2. Where you're from
3. What you do
4. Something awesome that you did

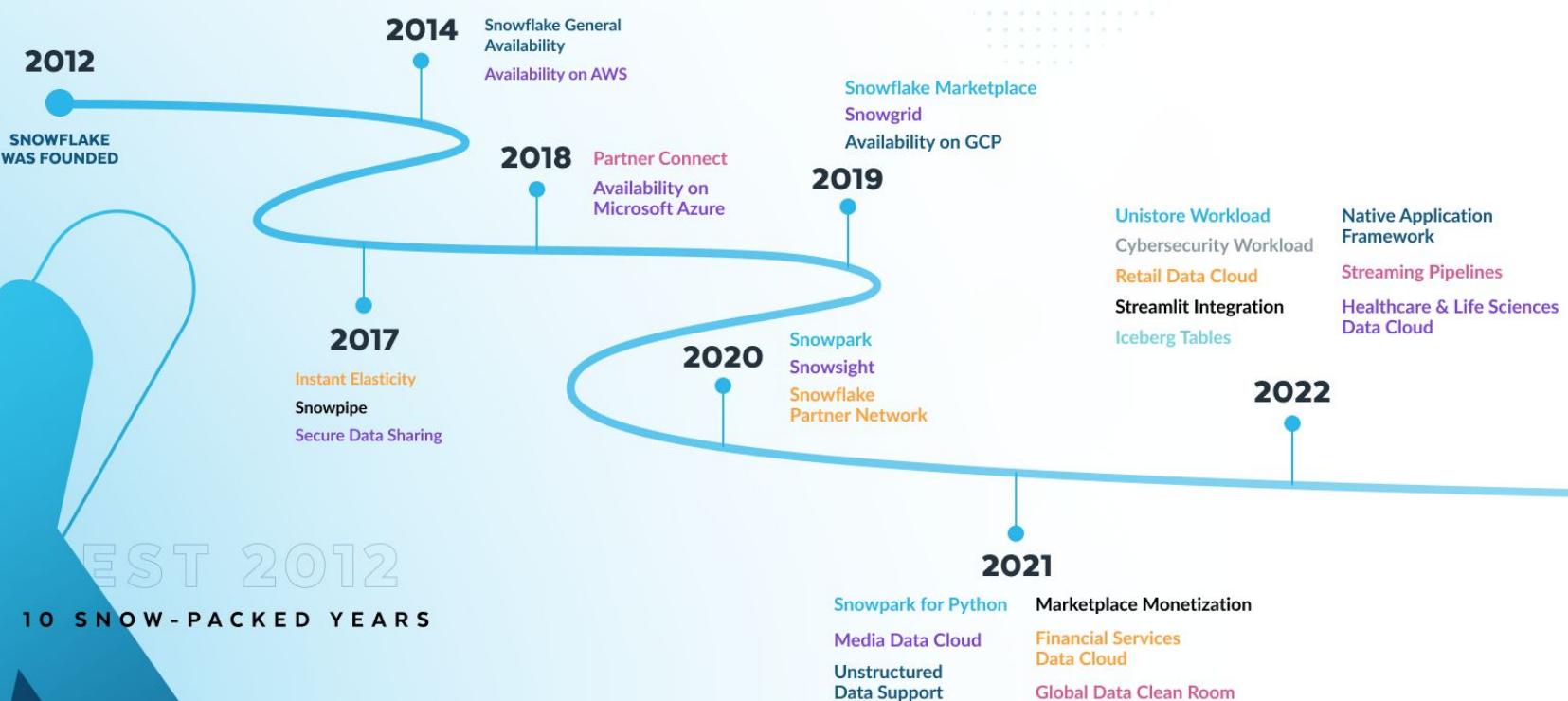


Join at
slido.com
#1643 092





10 YEARS OF PRODUCT INNOVATION



Dates reflect when a product or feature was publicly announced, and does not represent a comprehensive list of announcements.

Goals

- To Help You:
 - Learn Why Snowflake Was Built
 - Understand How Snowflake Works
 - Do the basics in Snowflake
 - Try some advanced use cases
- Give You Resources for Your Next Steps
- Share the Awesomeness of Snowflake



Snowflake 101



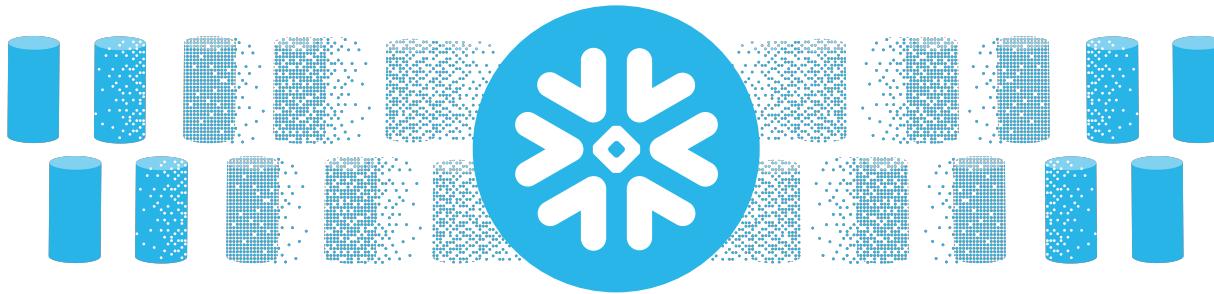
SNOWFLAKE'S MISSION TO MOBILIZE THE WORLD'S DATA



Every Organization Struggles with Silos

Silos equate to complexity, higher costs, and security risks

SNOWFLAKE ELIMINATES SILOS



With no silos, you can:

**Simplify your
data foundation**
for all data and workloads

Accelerate AI
for all users against
enterprise data

Scale with applications
with streamlined development,
deployment, and productization

The Data Cloud Makes This Easy

Snowflake Data Cloud

A global network connecting you to the most relevant content, powered by a single platform

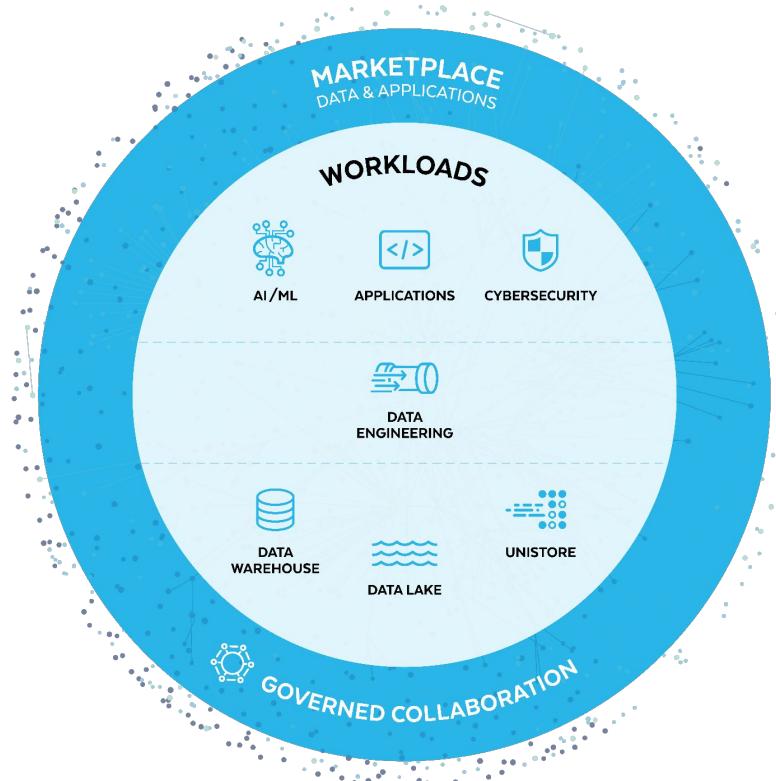
>9.4K

Customers

>2.4K

Marketplace Listings

Figures as of Jan. 31 2024



Simplify the Data Foundation



Requirements for Your Data Foundation

Endless Silos

Silos result in operational complexity and moving data to get value.



Unified Data

Switch to a single platform to access all data, including data that's unstructured, in open formats, and from third-parties.

Hidden Costs

Hidden costs come from multiple services and the manual expertise required to optimize.



Superior TCO

Switch to a fully managed service to efficiently support users and workloads at scale, while reducing time, effort, and expertise required.

Piecemeal Policies

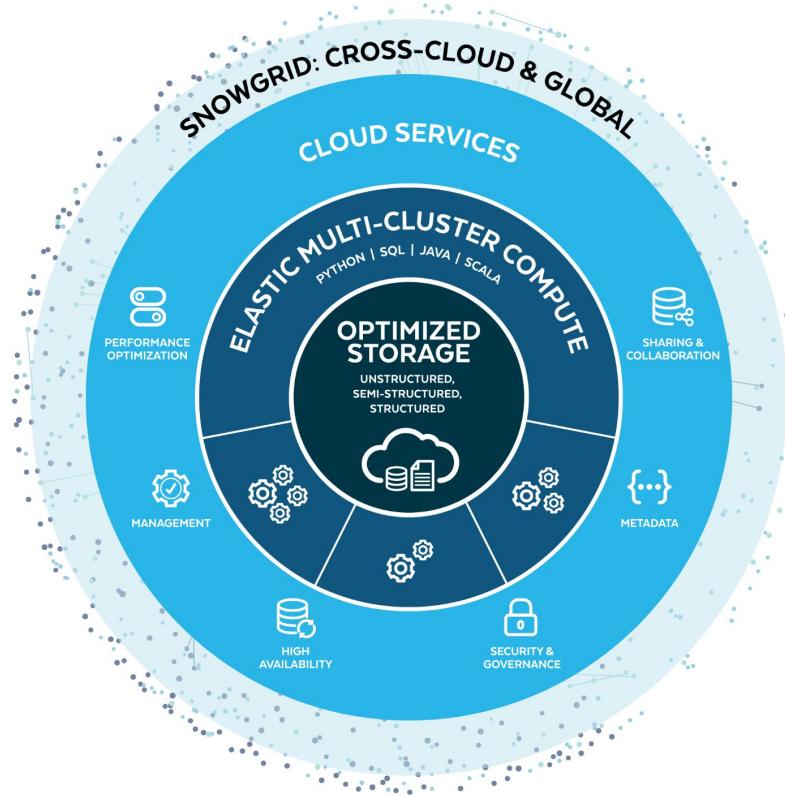
Inconsistent governance policies across systems and users introduce security risk.



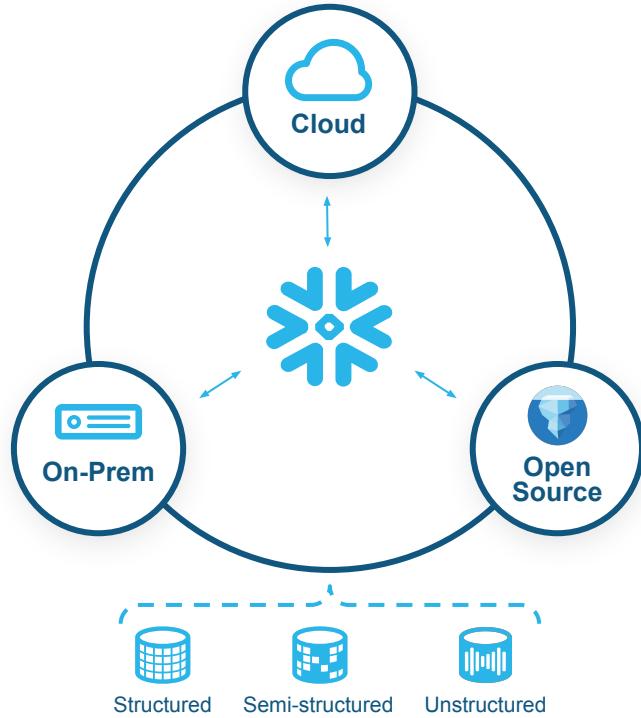
Universal Governance

Switch to a single governance model with comprehensive compliance, security, and privacy controls that are universally enforced.

Snowflake Platform Architecture



Optimized Storage



Un-siloed Access to Your Data

Unstructured, semi-structured, and structured data together with near-infinite scale.

Easily Manage Data at Scale

Fast and efficient access, optimized compression, and secure data - all automated.

Flexibility & Interoperability

Work with data on-premises or in open table formats* to remove lock-in and adapt to any architectural pattern.



One Platform. Any Architectural Pattern.

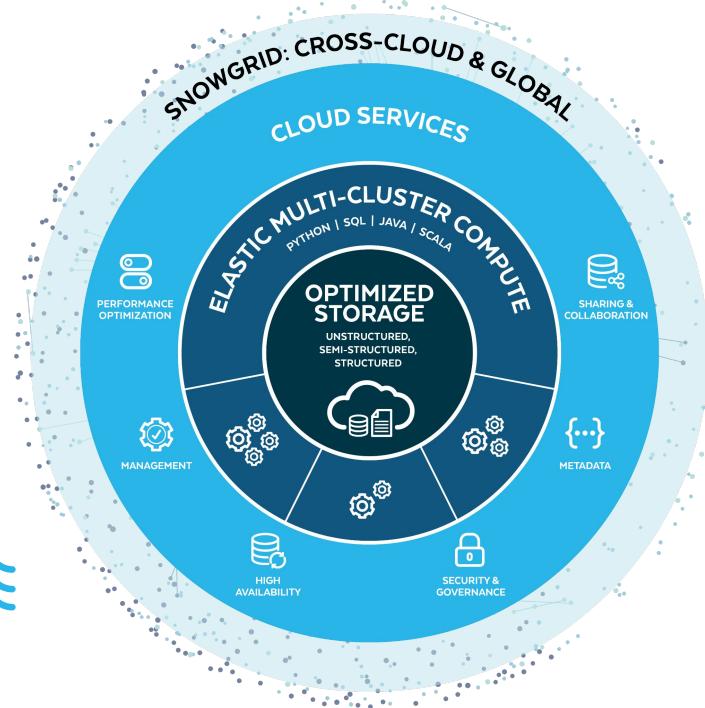
Data Warehouse

Centralized analytics for structured business data



Data Lakehouse

Transactional data lake for unified analytics, AI/ML, collaborative workloads



Data Lake

Unlimited storage for versatile data types and workloads



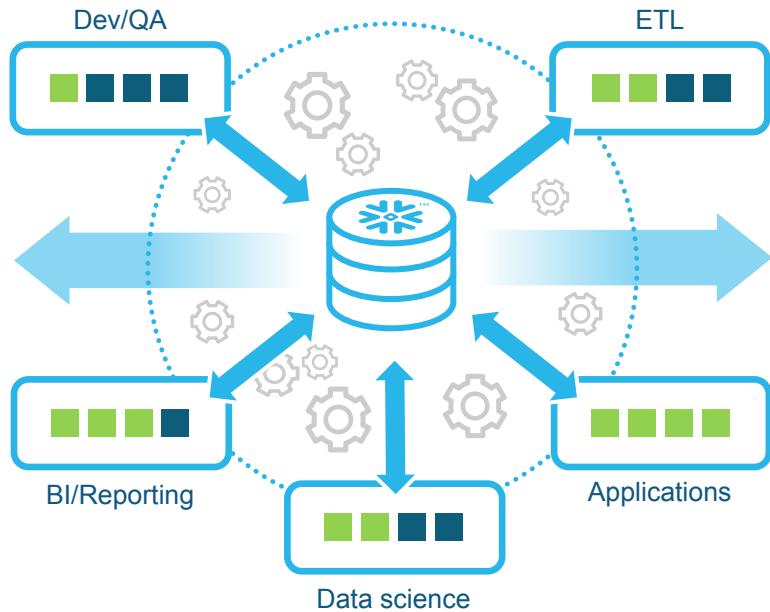
Data Mesh / Fabric

Distributed and governed, domain-oriented collaboration





Elastic Multi-Cluster Compute



One Engine for Every Workload

Power streaming pipelines, analytics, AI, interactive apps, and more through a single engine with flexible compute options including GPUs.

Leading Performance & Concurrency

Fast, reliable performance for virtually all users and jobs with no tuning or contention, delivered through isolated compute.

Familiar Languages

Work in SQL, Python, or Java. Run your preferred libraries with Snowpark. All without moving data.

Flexible Development

Use native development interfaces, bring your own IDE, or leverage popular third-party tools.



Break Down More Silos with Snowpark



What is Snowpark?

Secure deployment and processing of non-SQL code to develop pipelines, machine learning models, apps, and more



Language of Choice

Enable all users to bring their work to a single platform with native Python, Java, Scala support



No Governance Trade-offs

Apply consistent controls trusted by over 500 of the Forbes Global 2000 across all workloads in Snowflake



Faster & Cheaper Pipelines

Migrate Spark pipelines with minimal code change, better price-performance, transparent costs, and less overhead.



Cloud Services



Snowflake Managed

Governance
Controls

AI Models & LLMs

Maintenance
& Tuning

Performance
Enhancements

Networking
& Encryption

Administration &
Availability

Fully Managed Service

Automate costly and complex operations to reduce overhead and improve efficiency. Transparent performance enhancements automatically applied via releases.

Unified, Built-In Governance

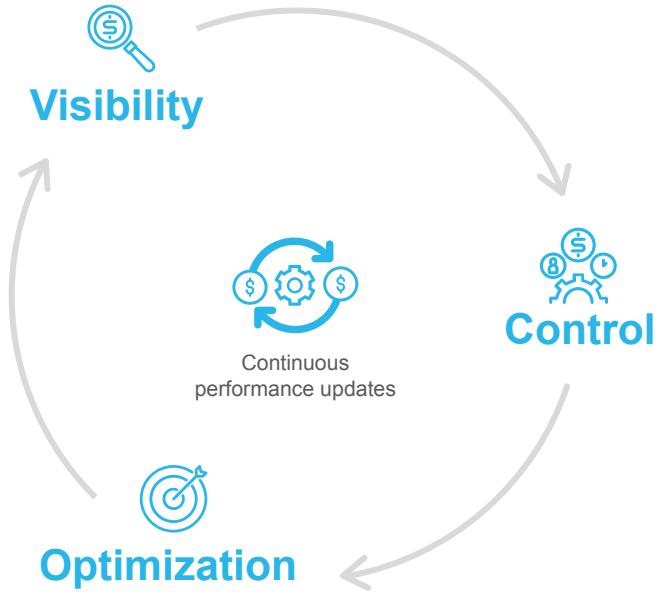
Snowflake Horizon delivers unified compliance, security, privacy, interoperability, and access capabilities, without additional configurations or protocols.

Accessible, Built-In AI

Snowflake Cortex* enables secure access to industry-leading AI models, LLMs, and vector search functionality, with no infrastructure to manage.



Maximizing Cost-Efficiency at Scale



Right-Sized Resources

Avoid over provisioning and manual tuning with the right resources when you need them. Right-size costs with per server, per second pricing.

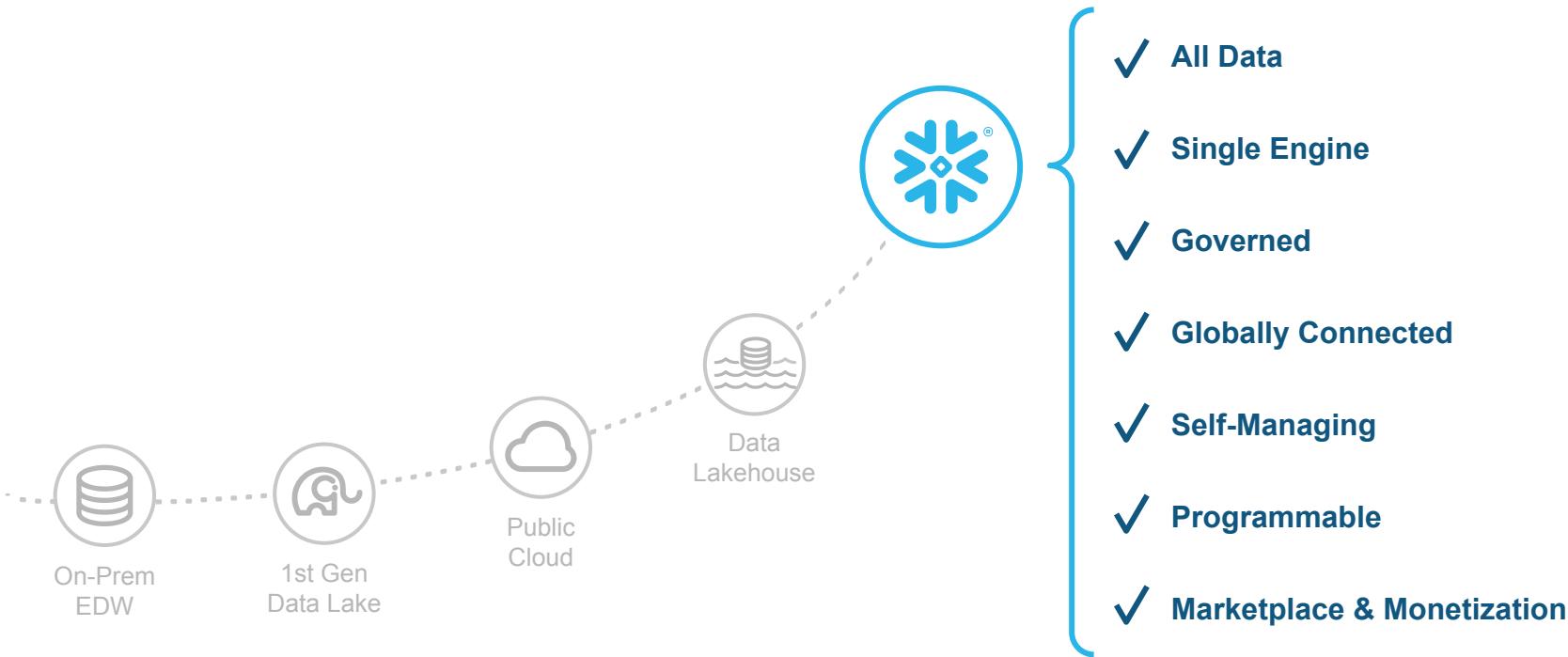
Continuous Improvements

Ongoing, transparent performance enhancements delivered with nearly every release for cost savings with no action required.

Effectively Control & Manage Spend

Understand usage metrics, control spend, and gain optimization insights through the built-in cost management interface to maximize the value of Snowflake.

Innovation Journey to the Data Cloud



Snowflake Platform



It Just Works

Single, unified platform that delivers ease of use and powerful automations



Optimized Performance at Scale

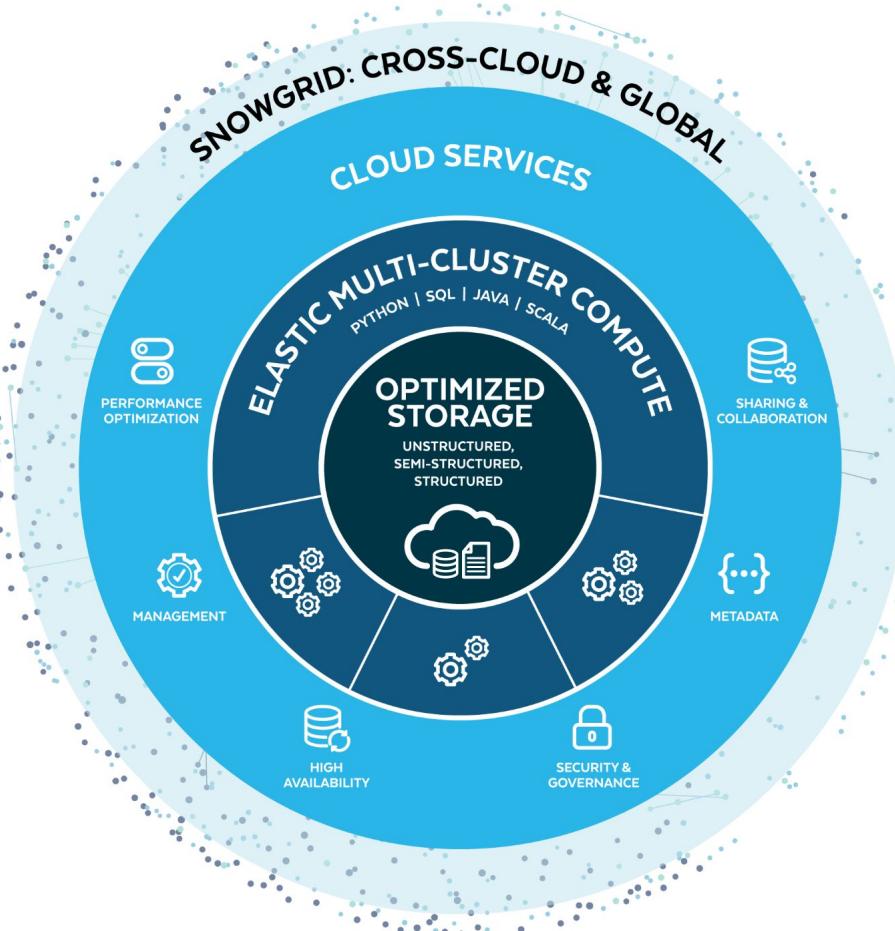
Scale up and down elastically, and scale out to support massive concurrent users, data volumes, and different workloads



Globally Connected

Access data and apps across clouds and regions securely with one consistent experience

SNOWFLAKE PLATFORM ARCHITECTURE



Most Valuable Resources

These + Google are your friends

Snowflake Docs

<https://docs.snowflake.com/>

Some popular topics:

[Loading Data Into Snowflake](#)

[Querying Semi-Structured Data](#)

[Managing Your Snowflake Account](#)

[Managing Security in Snowflake](#)

Snowflake Quickstarts

<https://quickstarts.snowflake.com/>

Snowflake Community

<https://community.snowflake.com/>

Resources

Getting Started: Snowflake Onboarding

Knowledge Base: Articles & Forums

Blogs: Best Practices & How-tos

Newsletter Subscription: What's New?

Releases: Product Updates

Education & Training: Snowflake University & Self-Learning (including Migration guides, handy scripts)

Snowflake Status -

<https://status.snowflake.com/>



Initial Account Creation & Setup



Trial Account Creation

<https://signup.snowflake.com/> - Includes \$400 of usage credit



The screenshot shows the Snowflake trial sign-up process. On the left, there's a sidebar with a list of benefits and compliance logos. The main form asks for personal and company information. On the right, the user is prompted to choose their Snowflake edition (Enterprise is selected), cloud provider (AWS is selected), and region (US West (Oregon) is selected). A large red arrow points from the text "Be sure to choose:" to the "Enterprise", "AWS", and "US West" options.

START YOUR 30-DAY FREE TRIAL

- Gain immediate access to the Data Cloud
- Enable your most critical data workloads
- Scale instantly, elastically, and near-infinitely across public clouds
- Snowflake is HIPAA, PCI DSS, SOC 1 and SOC 2 Type 2 compliant, and FedRAMP Authorized

Compliance Logos: FCA DSS, HIPAA USA, HIPAA Certified, NIST EDC, FedRAMP

Start your 30-day free Snowflake trial which includes \$400 worth of free usage

Choose your Snowflake edition*

Standard
A strong balance between features, level of support, and cost.

Enterprise
Standard plus 90-day time travel, multi-cluster warehouses, and materialized views.

Business Critical
Enterprise plus enhanced security, data protection, and database failover/fallback.

Choose your cloud provider*

Microsoft Azure Amazon Web Services Google Cloud Platform

United States US West (Oregon)

By clicking the button below you understand that Snowflake will process your personal information in accordance with its [Privacy Notice](#).

Check here to indicate that you have read and agree to the terms of the [Snowflake Self Service On Demand Terms](#).

CONTINUE or [sign in to an existing account](#)

GET STARTED

Be sure to choose:

- Enterprise
- AWS
- US West

Tour of the UI

“Snowsight”

- Data
- Projects
 - Worksheets
- Admin

The screenshot displays the Snowflake UI interface. At the top left is the Snowflake logo. The top right features a photograph of hands playing a piano. Below the logo is a navigation bar with icons for Search, Projects, Data, Data Product, Document AI, Monitoring, and Admin. A search bar is positioned above a dropdown menu. The dropdown menu includes options like "Switch to canary", "Switch Role" (set to ACCOUNTADMIN), and "Account" (set to DEMO460). Other items in the dropdown are "My profile", "Support", "Client download", "Documentation", "Privacy notice", "Classic console", and "Sign Out". A user profile card at the bottom shows "Jeremy Lemmon" and "ACCOUNTADMIN". To the right of the dropdown is a detailed account view for "DEM0460" (JLEMMON - AWS US West (Oregon)). This view includes sections for "SFCOGSOPS" (SNOWHOUSE_AWS_US_WEST_2) and "SFSENORTHAMERICA" (DEMO460). It also lists "Organization: SFSENORTHAMERICA", "Cloud: AWS Amazon Web Services", "Region: US West (Oregon)", "Edition: Enterprise", and "Locator: AOB24649". A "Search PREVIEW" button is located at the top right of the main content area.



Some Info From You



<https://tinyurl.com/3sxy43v3>



Worksheets



Snowflake Worksheet interface showing a query result.

Timestamp: 2024-02-27 4:03pm

Databases Worksheets

Pinned (0)

No pinned objects

Search objects

> SNOWFLAKE

> SNOWFLAKE_SAMPLE_DATA

> INFORMATION_SCHEMA

> TPCDS_SF100TCL

> TPCDS_SF10TCL

> TPCH_SF1

Tables

- CUSTOMER
- LINEITEM
- NATION
- ORDERS
- PART
- PARTSUPP
- REGION
- SUPPLIER

> TPCH_SF10

> TPCH_SF100

> TPCH_SF1000

ACCOUNTADMIN COMPUTE_WH Share

No Database selected Settings Code Versions

```
1 SELECT * FROM SNOWFLAKE_SAMPLE_DATA.TPCH_SF1.LINEITEM
2 LIMIT 100;
```

Results Chart

	L_ORDERKEY	L_PARTKEY	L_SUPPKEY
1	2400001	132304	4818
2	2400001	24513	2020
3	2400001	175232	7750
4	2400001	119658	4681
5	2400001	89532	4549
6	2400002	188783	3820
7	2400002	67505	5024
8	2400002	142916	2917
9	2400002	182905	5424
10	2400002	80484	485
11	2400003	85613	8122
12	2400003	91115	1116
13	2400003	165300	333
14	2400003	186804	4359
15	2400003	2128	9629

Query Details

- Query duration: 1.0s
- Rows: 100
- Query ID: 01b2a3ab-0002-229d-...

Visualizations:

- L_ORDERKEY: Bar chart showing distribution of order keys.
- L_PARTKEY: Bar chart showing distribution of part keys.
- L_SUPPKEY: Bar chart showing distribution of supplier keys.



Your Turn



Make sure you can:

- 1) Log In
- 2) Access Worksheets
- 3) See the SNOWFLAKE_SAMPLE_DATA database
- 4) Run a sample query (e.g., TPC-H data set)
 - a) select count(*) as count_order from lineitem;
 - b) select L_SHIPMODE, count(*) as count_order from lineitem group by L_SHIPMODE;

BONUS:

Generate a query that reports the amount of business that was billed, shipped, and returned using the TPCH data set.

Hint: Snowflake docs are very, very good....



A Note on Fully-Qualified versus “Short Names”

- Queries run in a specific context
 - Role
 - Database
 - Schema
 - Etc.
- If you are querying within the “currently active” context, you can use short names
 - `select count(*) as count_order from lineitem;`
- HOWEVER, you can query from ANY database/schema/table you have access to if you fully qualify
 - `SELECT`
 - `(SELECT count(*) FROM SNOWFLAKE_SAMPLE_DATA.TPCH_SF10.lineitem) as count_order_tpch10`
 - `, (SELECT count(*) FROM SNOWFLAKE_SAMPLE_DATA.TPCH_SF100.lineitem) as count_order_tpch100;`

	... COUNT_ORDER_TPCH10	COUNT_ORDER_TPCH100
1	59986052	600037902



DAY ZERO BEST PRACTICES



SNOWFLAKE DATA PROTECTION

Continuous Data Protection Lifecycle



DATA RETENTION TIME IN DAYS:

- Set to 1 day by default
- Can be set at account level, database and schema level:
 - Set it up for 7 days at the account level until figuring out a more refined approach:

```
use role accountadmin;  
alter account set DATA_RETENTION_TIME_IN_DAYS=7;
```



VIRTUAL WAREHOUSE MANAGEMENT

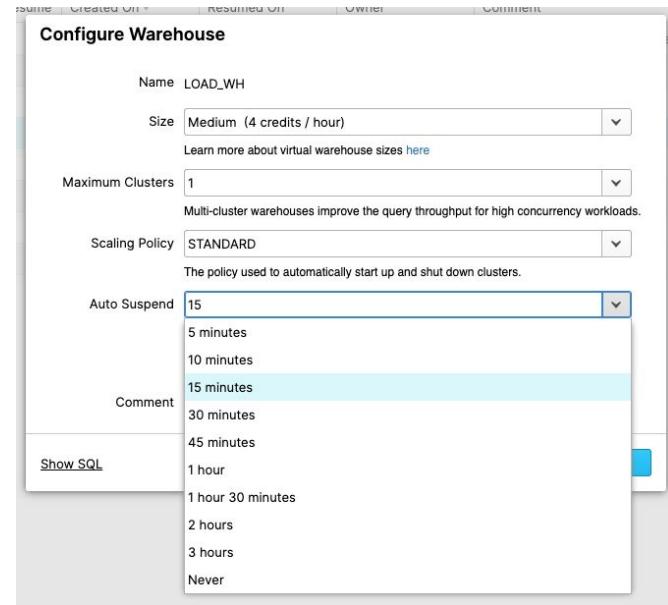
Assign unique warehouses to unique workloads

Start testing query workload with an X-SMALL, and test scaling-up.



Auto-suspend should be set for all warehouse. Time setting should be dependent on caching requirement.

Leverage MCW when it makes sense.



MISCELLANEOUS

Avoid runaway queries by setting STATEMENT_TIMEOUT_IN_SECONDS:

```
/* Set statement timeout for 6 hours */
use role accountadmin;
alter account set STATEMENT_TIMEOUT_IN_SECONDS=21600;
```

Storage costs containment: Make use of short-lived tables in your data pipelines for temporary/intermediate staging tables which do not need to be persisted/saved:

TEMPORARY TABLE: Available for the user owning the session, No fail safe, Time Travel up to 1 day, no fail safe, short lived for the session.

TRANSIENT TABLE: Available for all authorized users, No fail safe, Time Travel up to 1 day, remains till dropped.

Set the default timezone

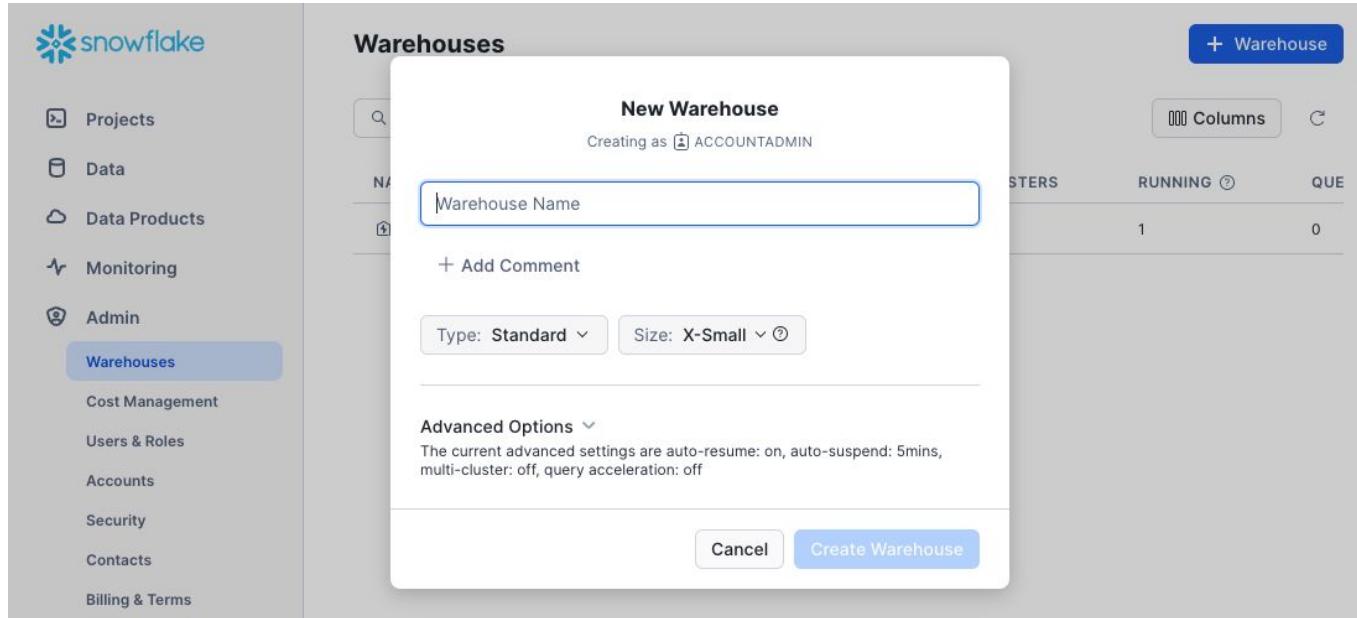
```
use role ACCOUNTADMIN; -- Must have ACCOUNTADMIN to change the setting.
alter account set TIMEZONE = 'America/Chicago';
```



Initial Setup Best Practices

Starting off on the right foot

1. Create a Warehouse in the Snowsight UI



Initial Setup Best Practices

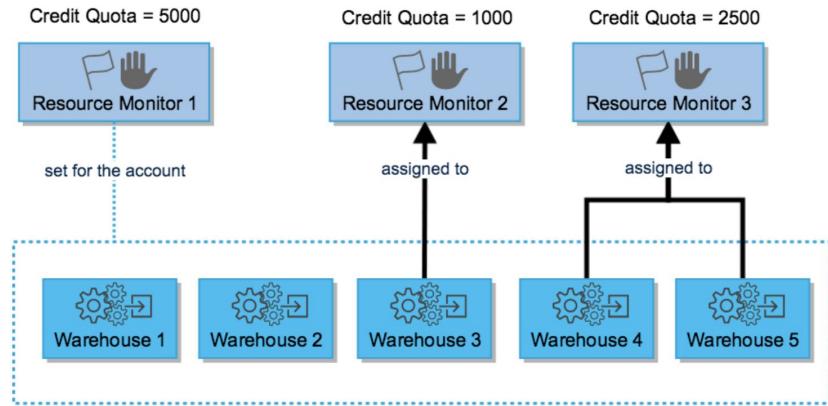
Resource Monitors

Define an initial account level resource monitor as safeguard:

```
USE ROLE ACCOUNTADMIN;
CREATE OR REPLACE RESOURCE MONITOR "ACCT_THRESH_25PCT"
WITH CREDIT_QUOTA = 5000,
frequency = 'NEVER',
start_timestamp = 'IMMEDIATELY'
TRIGGERS
ON 25 PERCENT DO NOTIFY
ON 50 PERCENT DO NOTIFY
ON 75 PERCENT DO NOTIFY;
ALTER ACCOUNT SET RESOURCE_MONITOR = "ACCT_THRESH_25PCT";
```

[Enable Receipt of Email Notifications](#)

Define additional warehouse level RM as required per type of workload/assigned warehouse.



Your Turn

Explore the Cost Management UI



1. Create an account-level Resource Monitor
2. See your usage

The screenshot shows the Snowflake Cost Management UI. On the left, there is a sidebar with navigation links: Projects, Data, Data Products, Monitoring (which is expanded), Admin, Warehouses, Cost Management (which is selected and highlighted in blue), Users & Roles, Accounts, Security, Contacts, and Billing & Terms. A notification at the bottom left says "30 days left in trial". At the bottom center is a blue "Upgrade" button. The main area is titled "Cost Management" and contains a form for "New Resource Monitor". The form fields include:

- Name: Name of resource monitor
- Credit Quota: Number of credits allowed per cycle
- Monitor Type: Select Monitor Type (dropdown menu)
- Schedule: Start Monitoring Immediately, End Monitoring Never, Resets Monthly, Customize (button)
- Actions: Specify an action to perform when the quota is reached. Three options are listed:
 - % Suspend immediately and notify when this % of credit is used. ⓘ
 - % Suspend and notify when this % of credit is used. ⓘ
 - % Notify when this % of credit is used. ⓘ

At the bottom right of the form are "Cancel" and "Create Resource Monitor" buttons.



Your Turn

Set initial controls using SQL commands



- Copy the contents of labs / lab0 / 1-financial_governance.sql into a new worksheet
- Walk through & execute each command

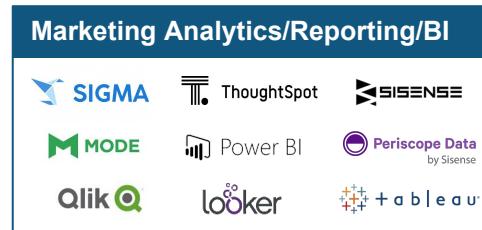
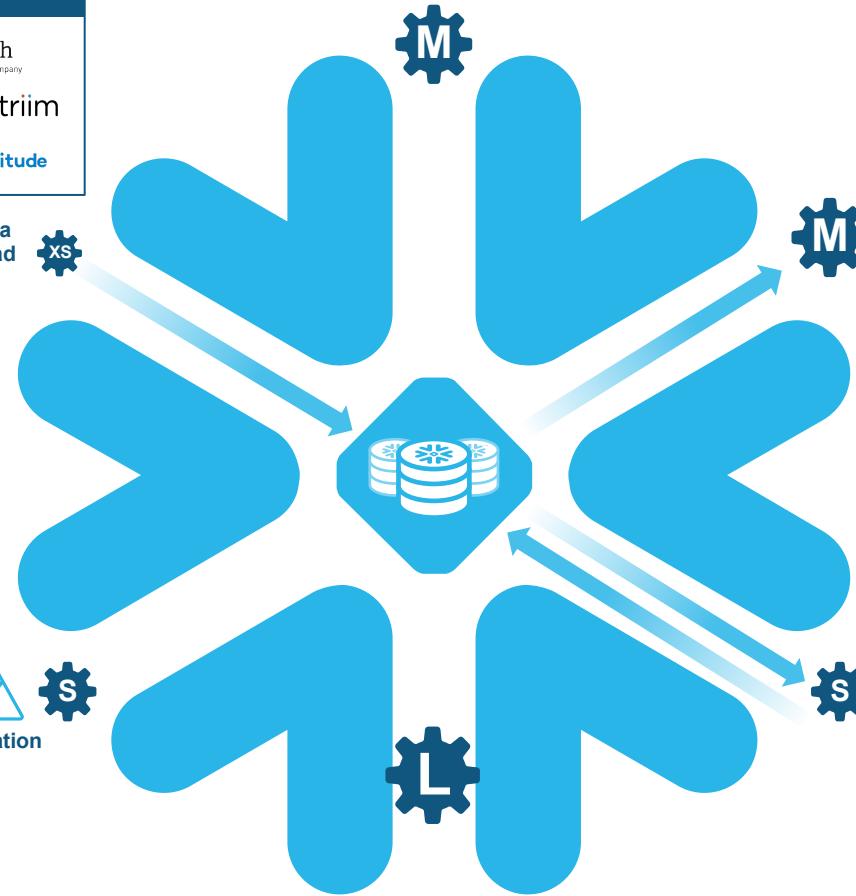
Snowflake UI screenshot showing a worksheet titled "1-Financial_Governance". The user is ACCOUNTADMIN in the TASTY_TEST_WH database. The code in the worksheet sets the default time zone to America/Chicago:

```
No Database selected ▾ Settings ▾ Co  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31 -- Set our Default Time Zone  
32 use role ACCOUNTADMIN; -- Must have ACCOUNTADMIN to change the setting.  
33 alter account set TIMEZONE = 'America/Chicago';  
34 use role SYSADMIN; -- (Best practice: change role when done using ACCOUNTADMIN)  
35
```



SNOWFLAKE ARCHITECTURE DEEP-DIVE





Data Transformation



Intelligent Infrastructure:

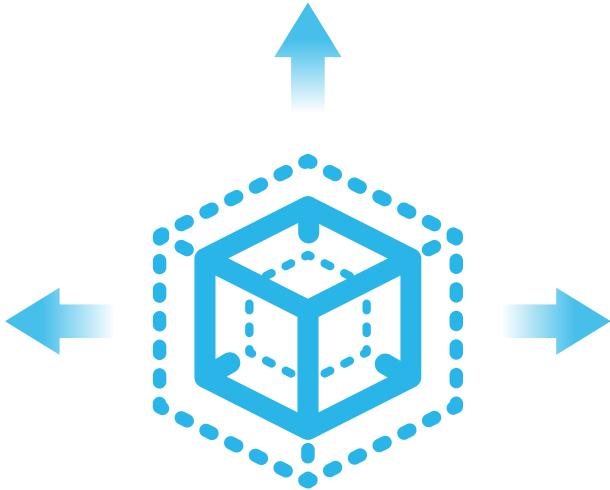
- Logical Model
- Security
- Query Planning & Optimization
- Transactional Control

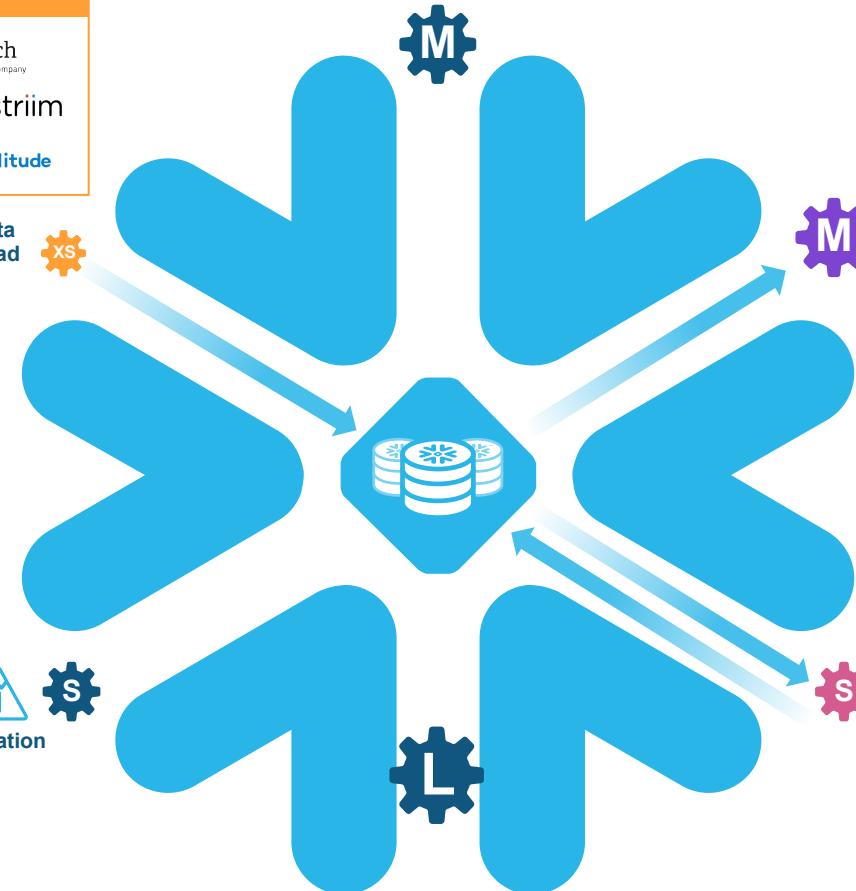


3 Dimensions of Scaling

ACROSS

- Many competing workloads
- Resource contention
- Isolate on separate warehouses





Intelligent Infrastructure:

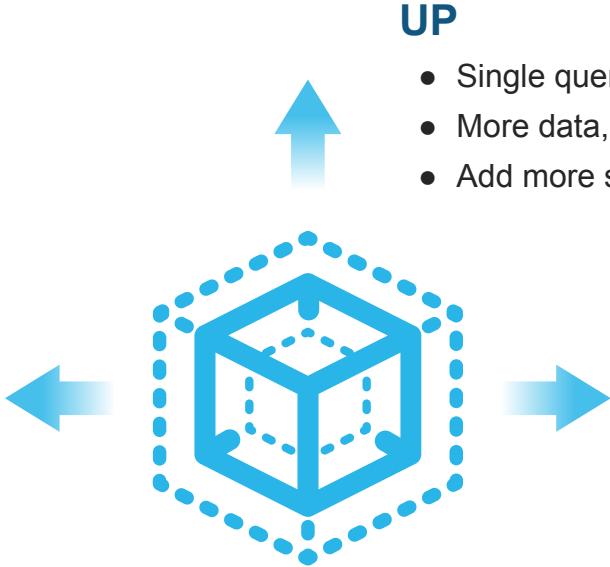
- Logical Model
- Security
- Query Planning & Optimization
- Transactional Control



3 Dimensions of Scaling

ACROSS

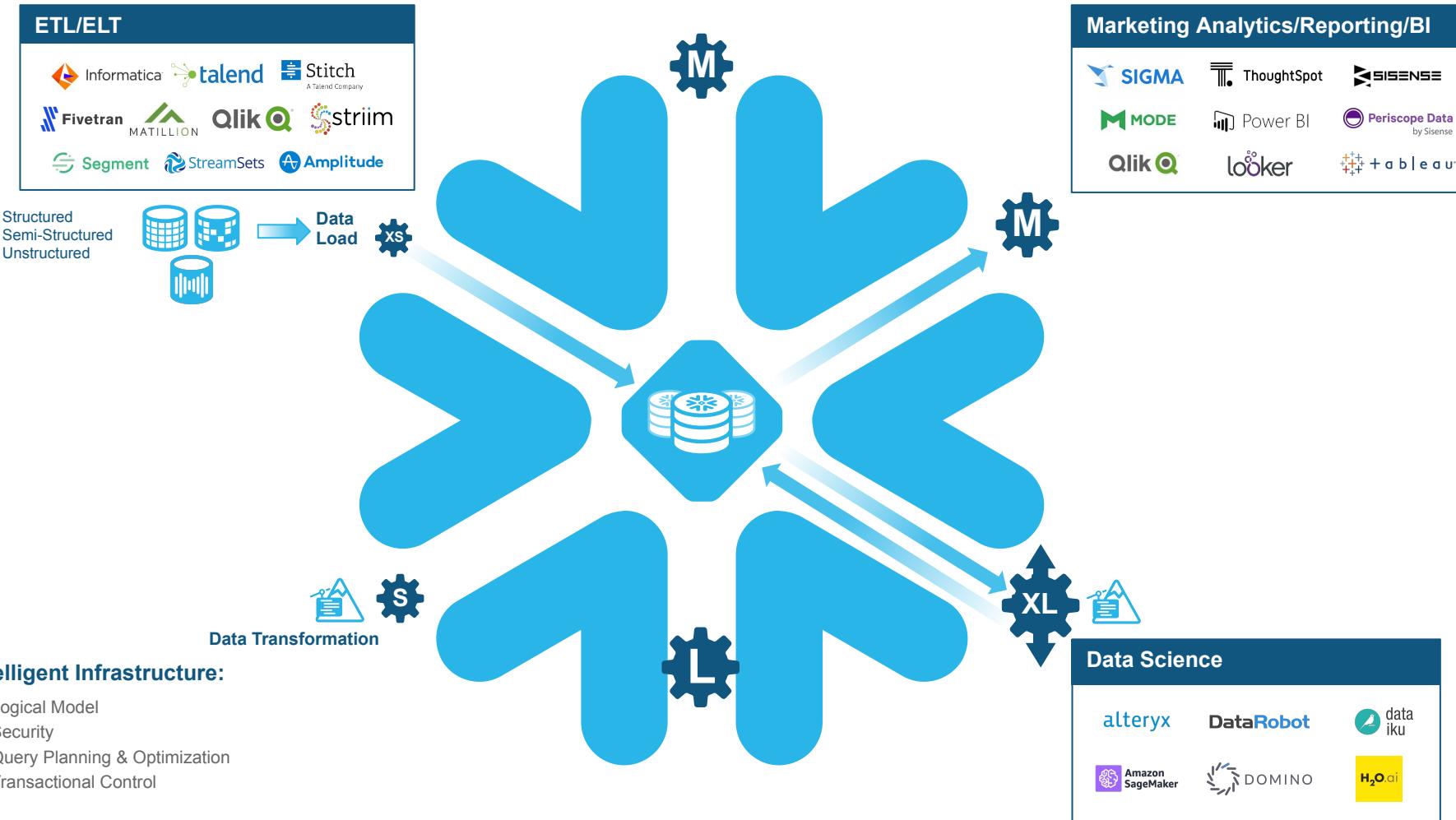
- Many competing workloads
- Resource contention
- Isolate on separate warehouses



UP

- Single query performance
- More data, more complex queries
- Add more servers to the cluster

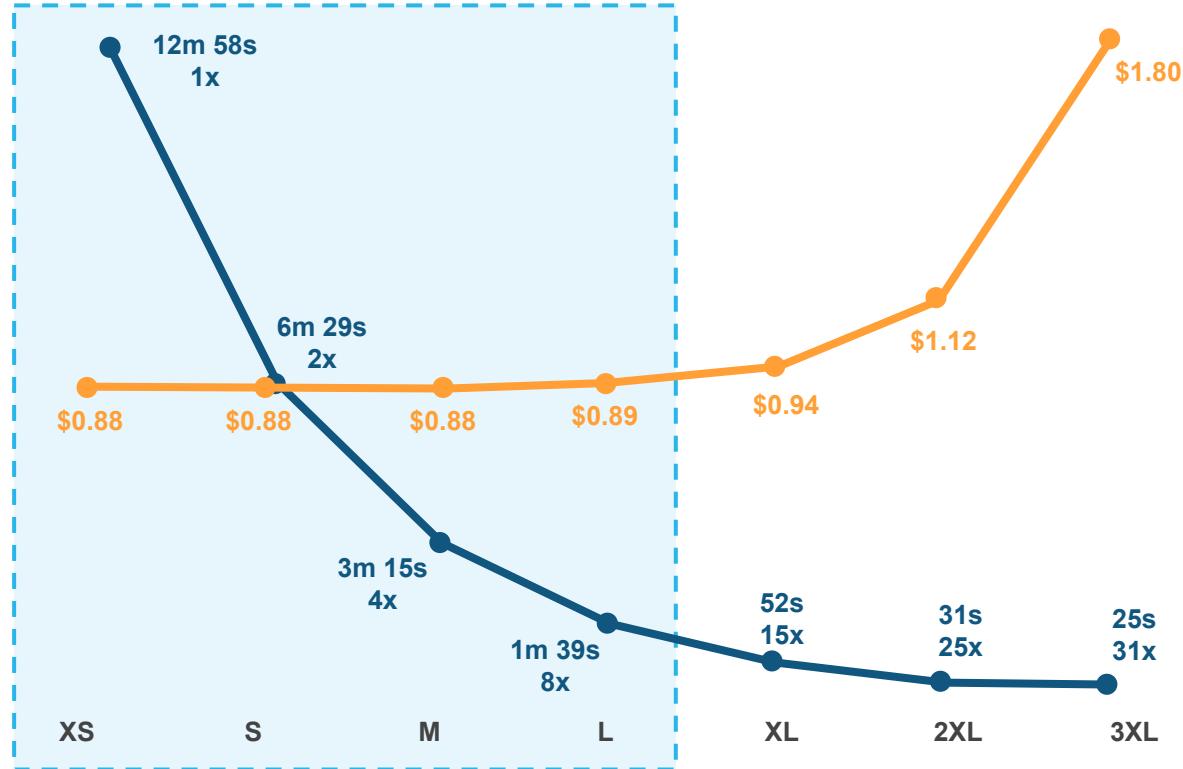




Scale Up – Loading 1BN Records

- 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

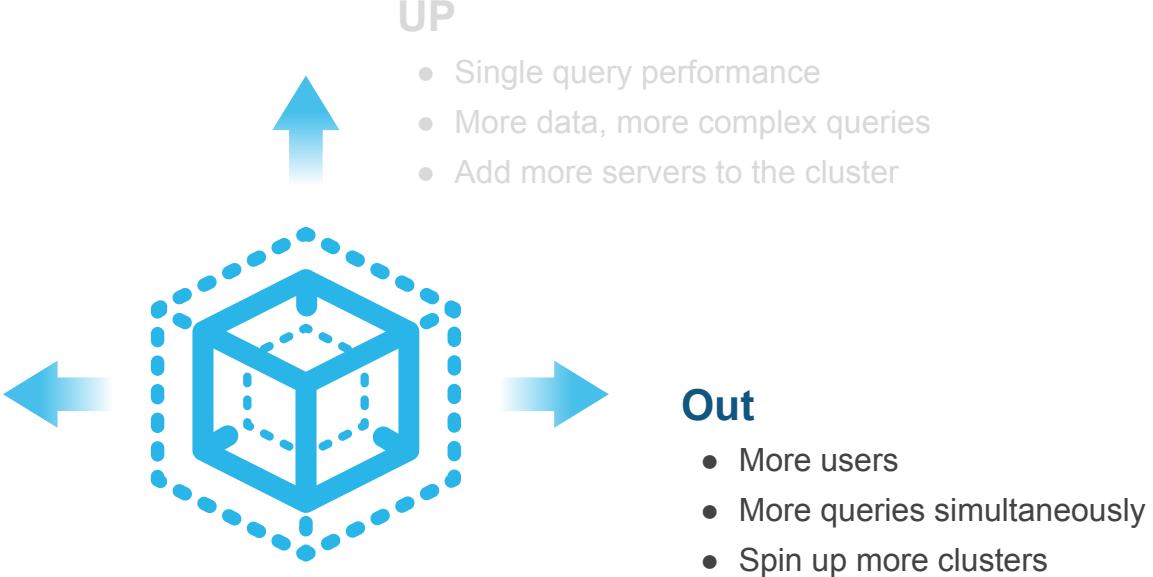
— Cost
— Secs



3 Dimensions of Scaling

ACROSS

- Many competing workloads
- Resource contention
- Isolate on separate warehouses



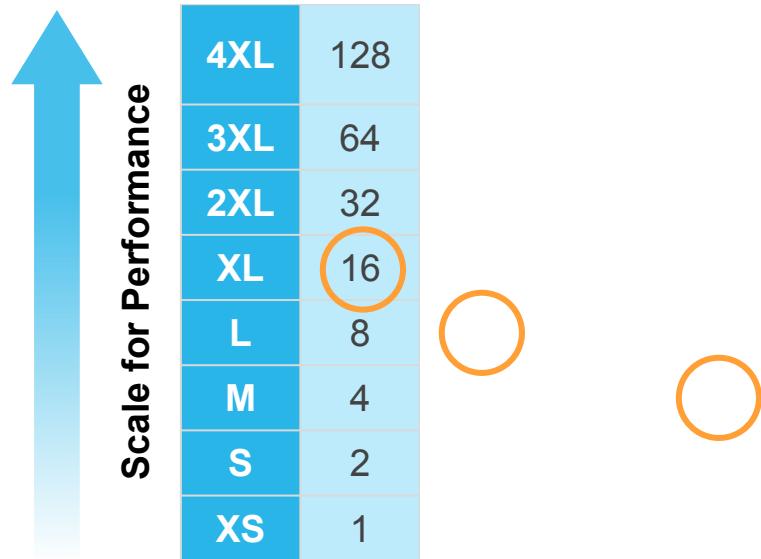


Intelligent Infrastructure:

- Logical Model
- Security
- Query Planning & Optimization
- Transactional Control



Scale Out: Multi-cluster Warehouses



Scale Out - Multi-cluster Warehouses



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.**

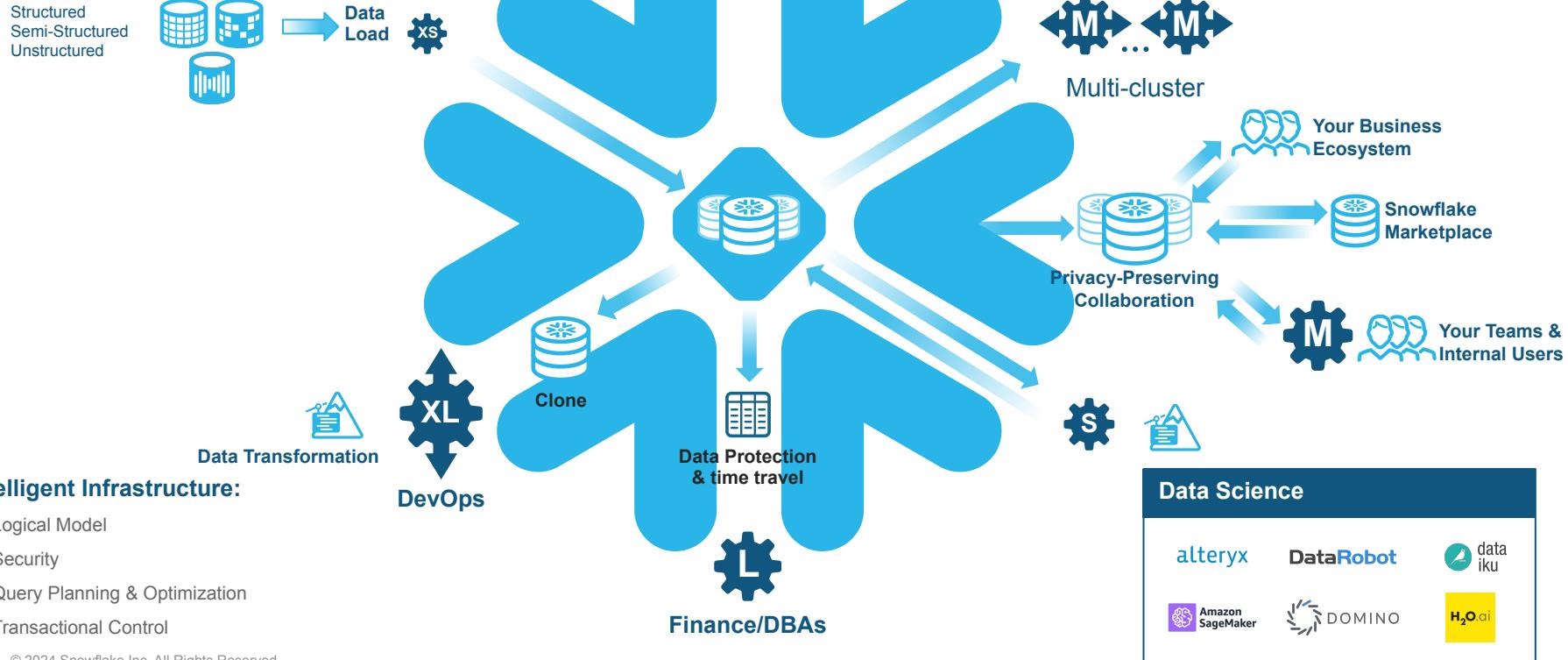


You pay per-server, per-second so
THEY ALL COST THE SAME.



Time





Intelligent Infrastructure:

- Logical Model
- Security
- Query Planning & Optimization
- Transactional Control

Your Turn - Tasty Bytes



Who Is Tasty Bytes?

ABOUT US: Global food truck network, localized menu options, 15 countries, 30 major cities, and 15 core brands.

OUR MISSION

We serve to give people unique food options with high quality items in a safe, convenient and cost effective way. We ensure that the ingredients used are of the highest quality from mostly local food vendors to make sure our success has a positive impact on community partners.



LOCATIONS SERVED

- USA: San Mateo, Denver, Seattle, Boston, New York City
- Canada: Toronto, Vancouver, Montreal
- United Kingdom: London, Manchester
- France: Paris, Nice
- Poland: Warsaw, Krakow
- India: Mumbai, Delhi
- Japan: Tokyo
- South Korea: Seoul
- Australia: Sydney, Melbourne



OUR VISION

To become the largest food truck network in the world by 2027 that has sustainable profitability with a zero carbon footprint future that our team, customers, and communities are proud of supporting.



CURRENT STATE & FUTURE GOALS



Business Overview

TASTY BYTES - ORDER PROCESS

ORDERING SERVICE

navigating through the menu options available to the customer. Once ready, they select the items they want to order.



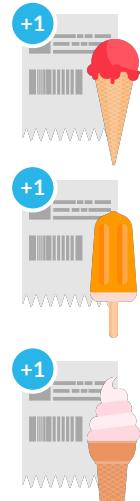
PAYMENT PROCESSING

customers can now pay with their saved selected payment method as part of their profile.



ORDER FULFILLMENT

Orders are placed in a queue, driver can prioritize each order.



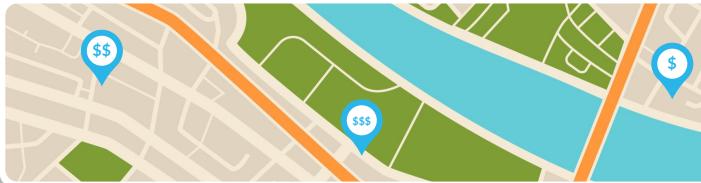
CUSTOMER SENTIMENT

voice of customer is captured through mobile app survey capturing the customer's experience and happiness with Tasty Bytes which is used to improve truck operation in the future.



GEO LOCATION SERVICES

sales data is captured through online order or POS for each truck at a given location. We can then determine where trucks should park in the future to maximize sales based on previous history, weather, and time of day (AM/PM)

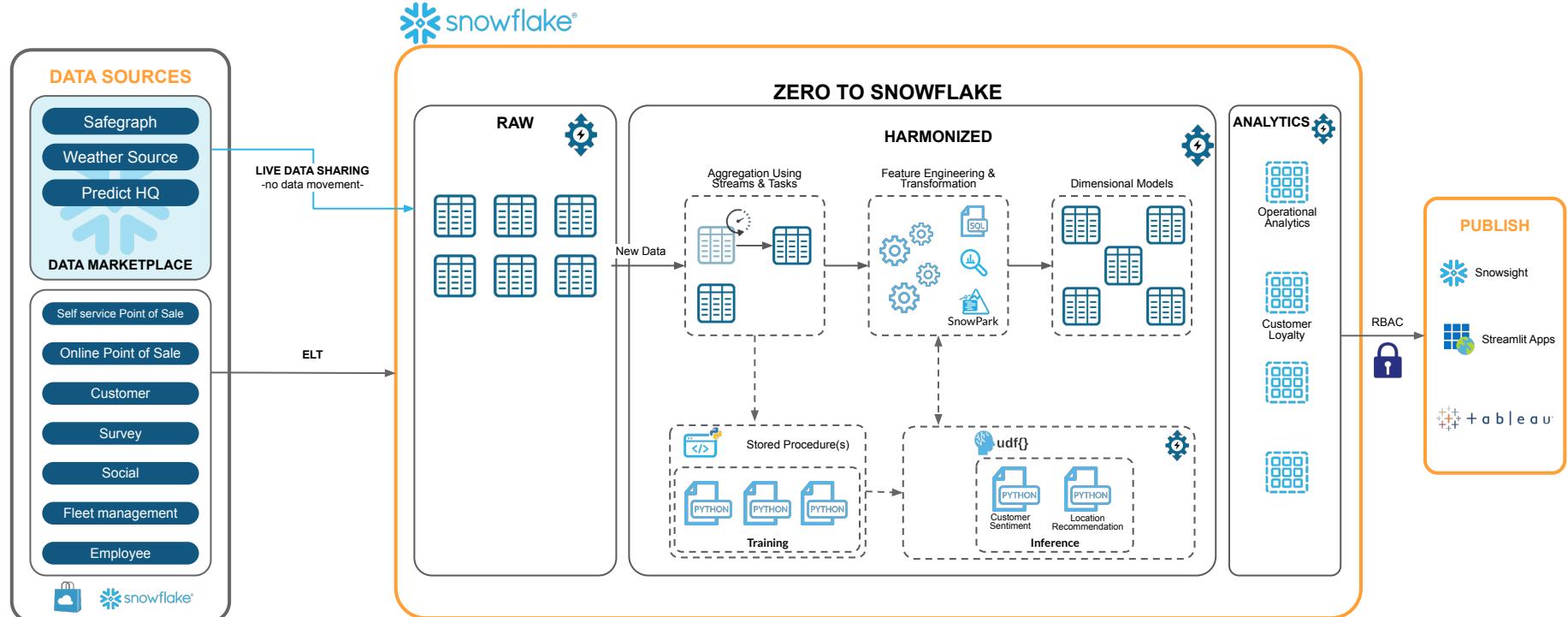


DRIVER APP

Truck Owner/ Operators have access to a complete 360 view of the order process via a specialized app



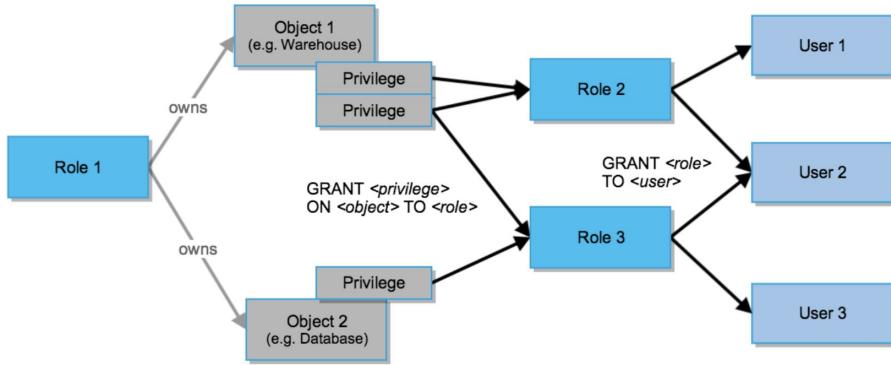
Reference Architecture



ROLE-BASED ACCESS CONTROL



ACCESS CONTROL



Granular control over access to objects

Covers who can access what objects, what operations can be performed on those objects, and who can create or alter access control policies

Two modes:

- Role-based Access Control (RBAC)
- Discretionary Access Control (DAC)

All object access authorizations occur through granted role grants

Independent of authentication

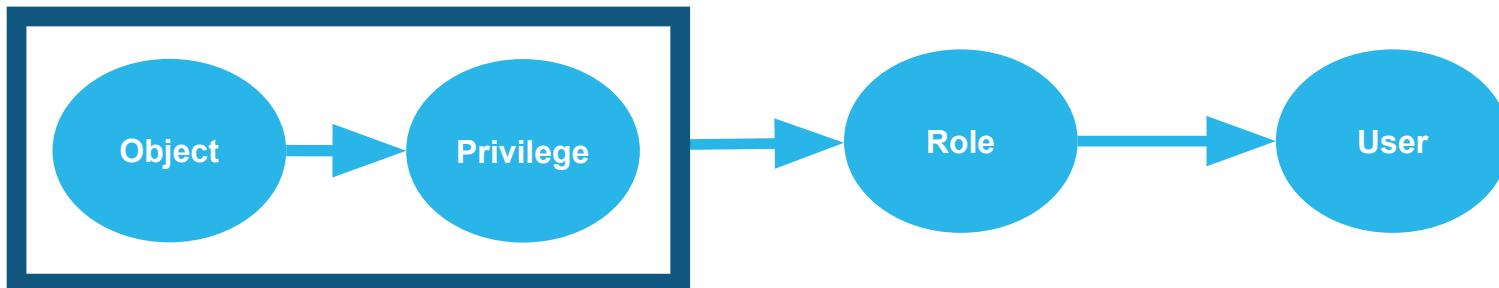
ACCESS CONTROL KEY CONCEPTS

Securable Object: An entity to which access can be granted

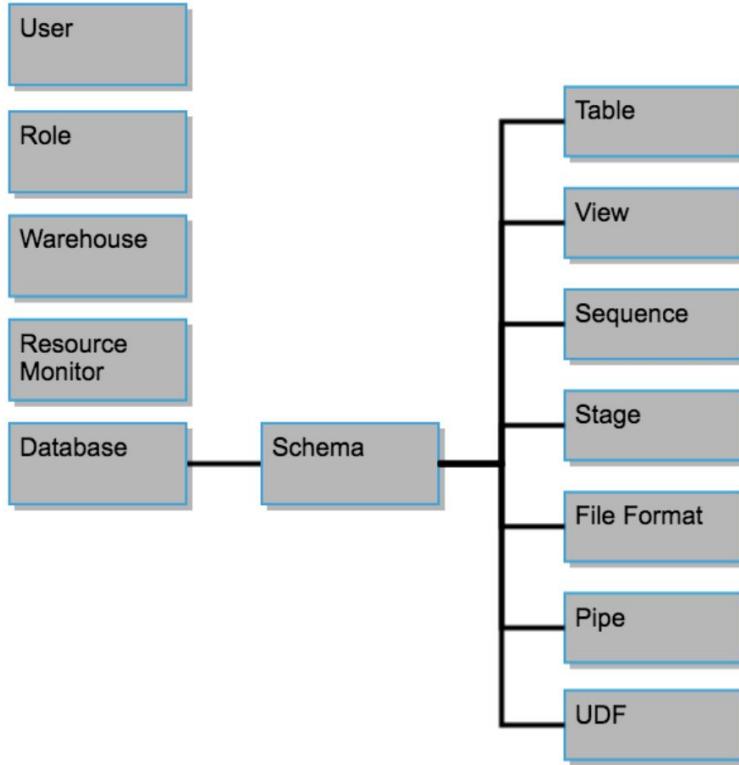
Privilege: A defined level of access to an object

Role: An entity to which privileges can be granted

User: A user identity associated with a person or program



SECURABLE OBJECTS



Securable objects reside within a logical hierarchy of containers.

The top-most container is the customer account.

Every securable object is owned by a single role, which is typically the role used to create the object.

When a Role is assigned to a User, the user is given shared control over the object.

The Object Owner can grant/revoke privileges or transfer ownership to another role.

Example privileges:

- *Create warehouse*
- *List tables in a schema*
- *Add data to a table*

Next Lab (5 minutes)

Tasty Bytes Setup



1. Create Initial Structures
2. Set up Access Control



Go to lab0 → tb_introduction.sql

- Run commands individually, or in bulk



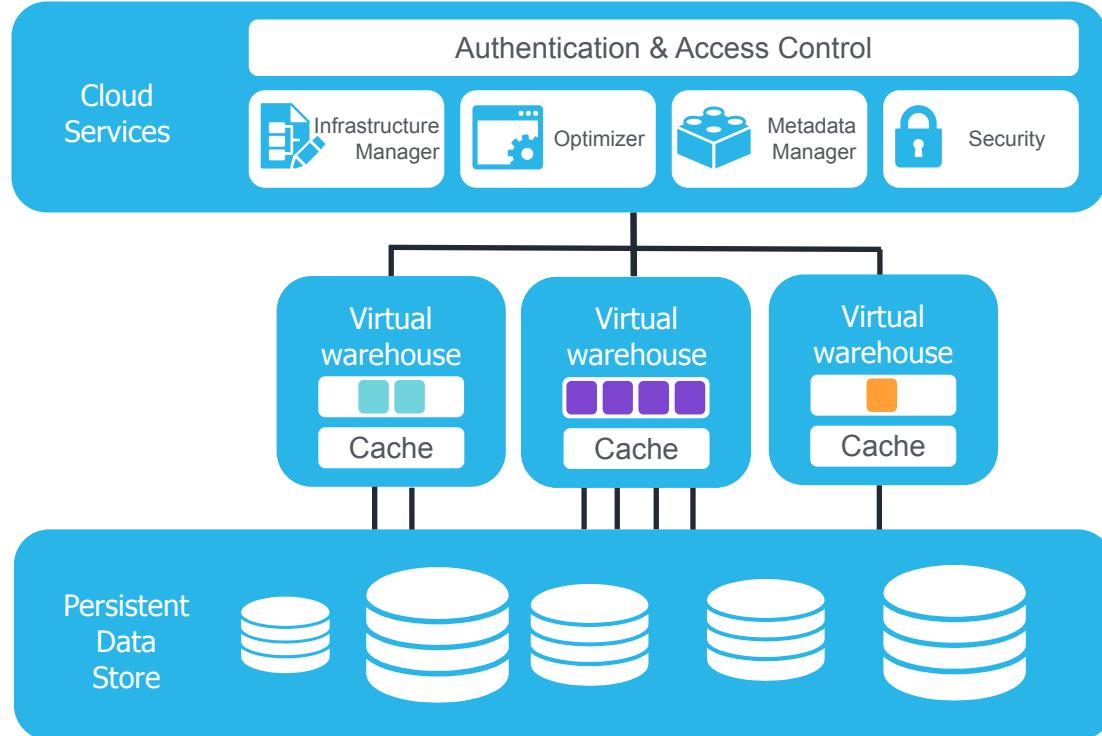
- Best Practice
 - Use SECURITYADMIN for creating/managing ROLES
 - Use SYSADMIN (or delegate) for creating/managing DB structure
 - ACCOUNTADMIN required for certain permissions
- GRANT can be on current objects (GRANT ... ON ...) or can include FUTURE objects



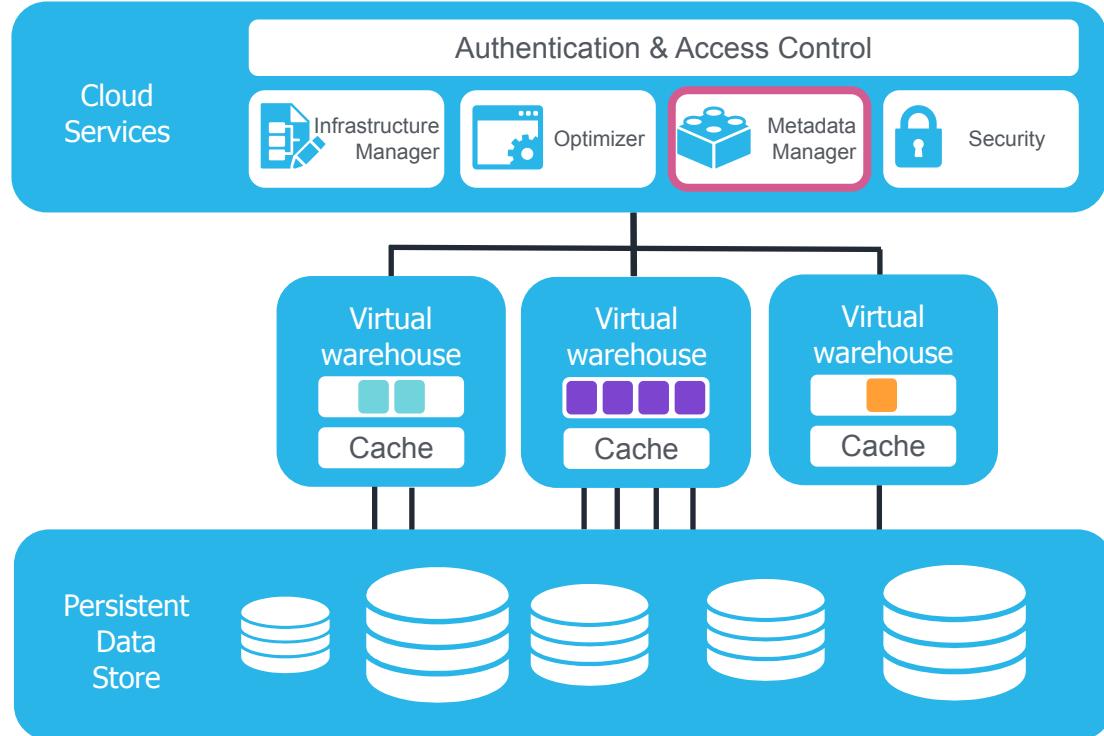
LOADING DATA INTO SNOWFLAKE



DATA STORAGE & DATA LOADING



DATA STORAGE & DATA LOADING

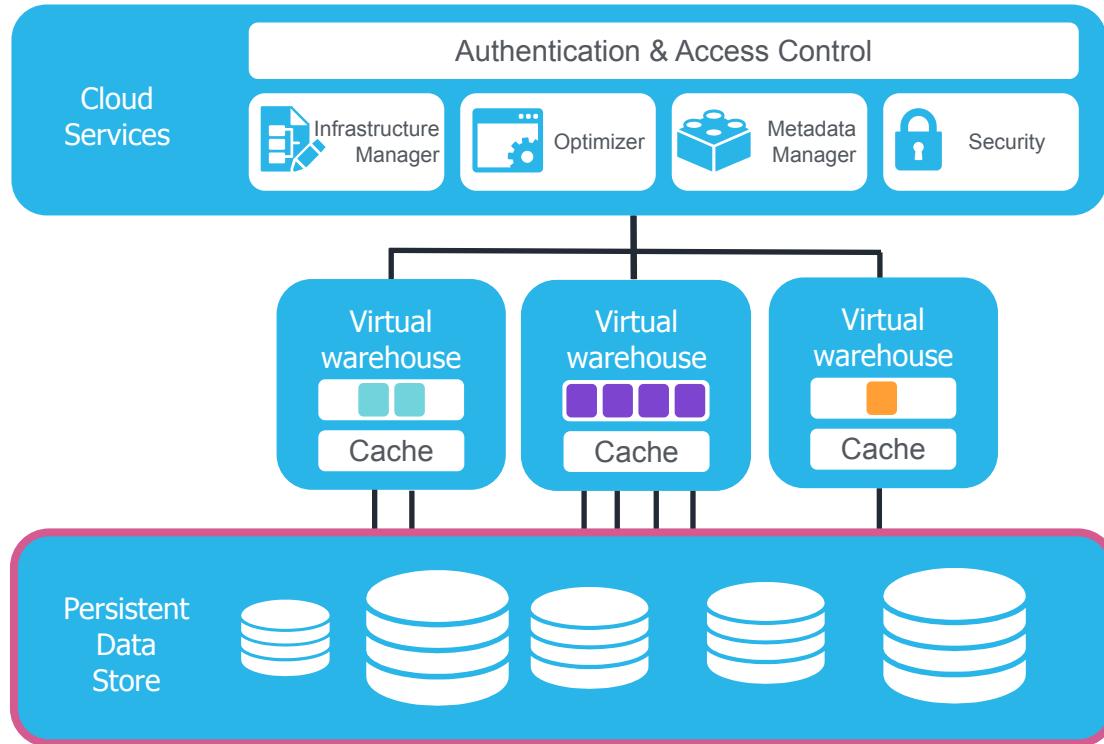


Metadata

Information about data stored in Snowflake
Stored & managed in cloud services layer
Stored in proprietary, fast-access, fault-tolerant, scalable datastore



DATA STORAGE & DATA LOADING



Metadata

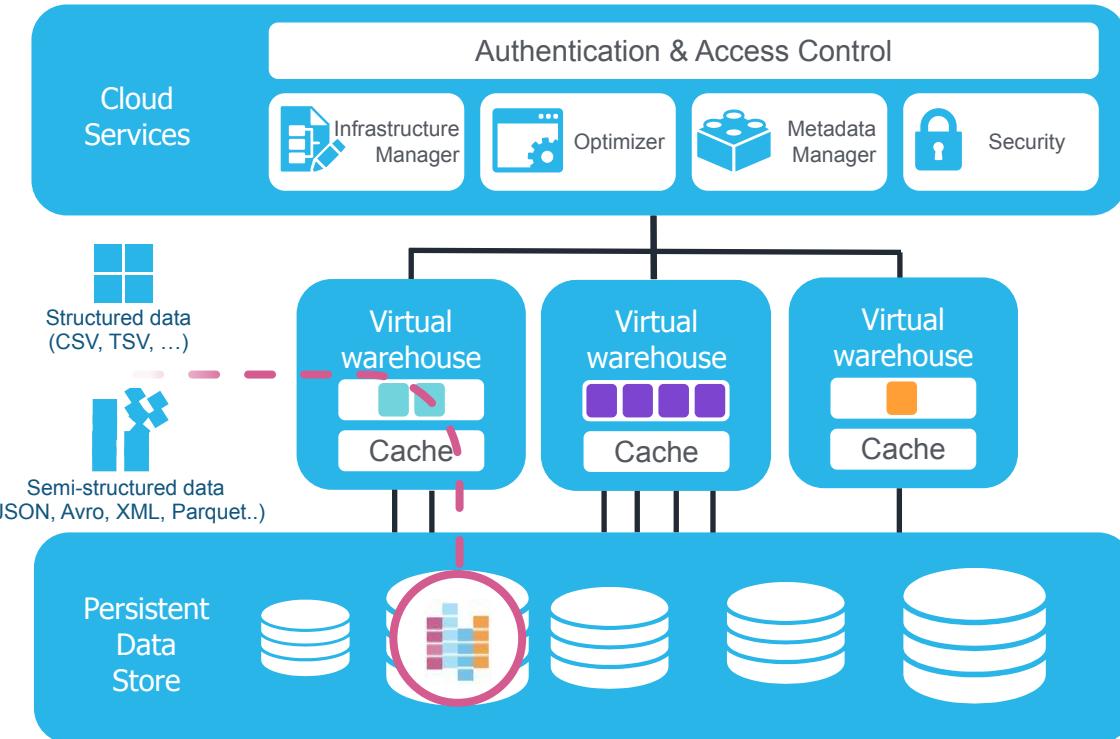
Information about data stored in Snowflake
Stored & managed in cloud services layer
Stored in proprietary, fast-access, fault-tolerant, scalable datastore

Database Data

Actual data in Snowflake databases & tables
Stored in Snowflake-managed cloud storage
Optimized, proprietary file format
Automatically compressed & encrypted



DATA STORAGE & DATA LOADING



Metadata

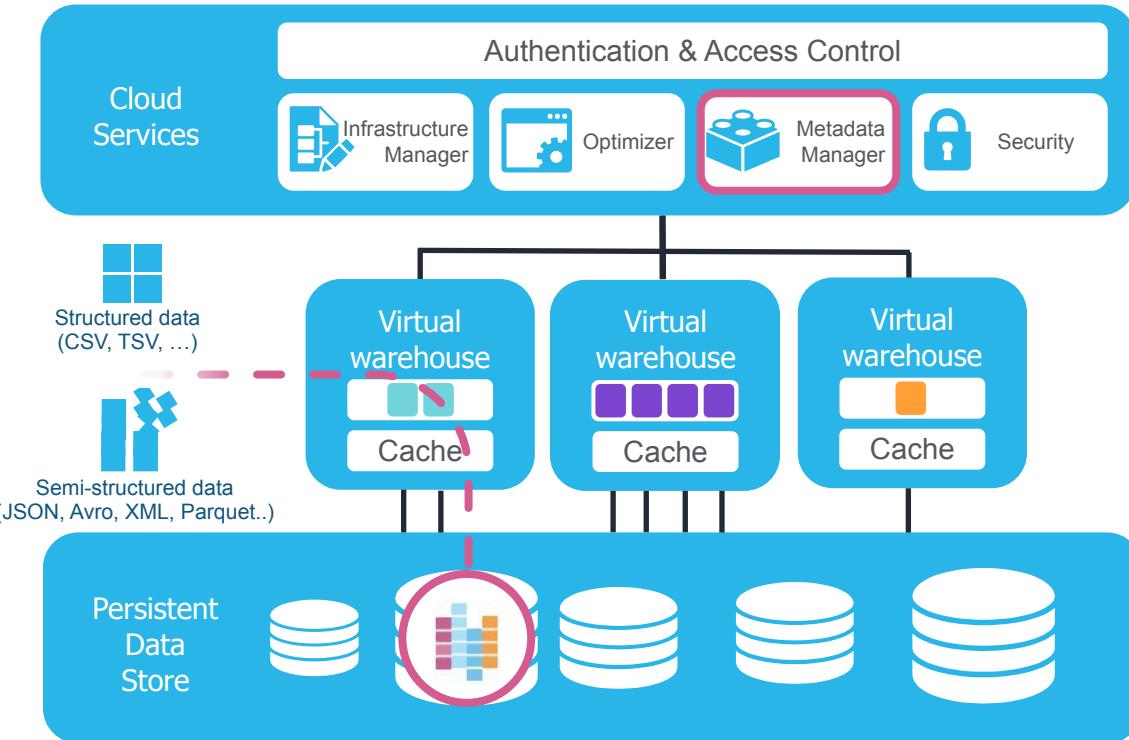
Information about data stored in Snowflake
Stored & managed in cloud services layer
Stored in proprietary, fast-access, fault-tolerant, scalable data store

Database Data

Actual data in Snowflake databases & tables
Stored in Snowflake-managed cloud storage
Optimized, proprietary file format
Automatically compressed & encrypted



DATA STORAGE & DATA LOADING



Metadata Created

Information about data stored in Snowflake

Stored & managed in cloud services layer

Stored in proprietary, fast-access, fault-tolerant, scalable data store

Database Data

Actual data in Snowflake databases & tables

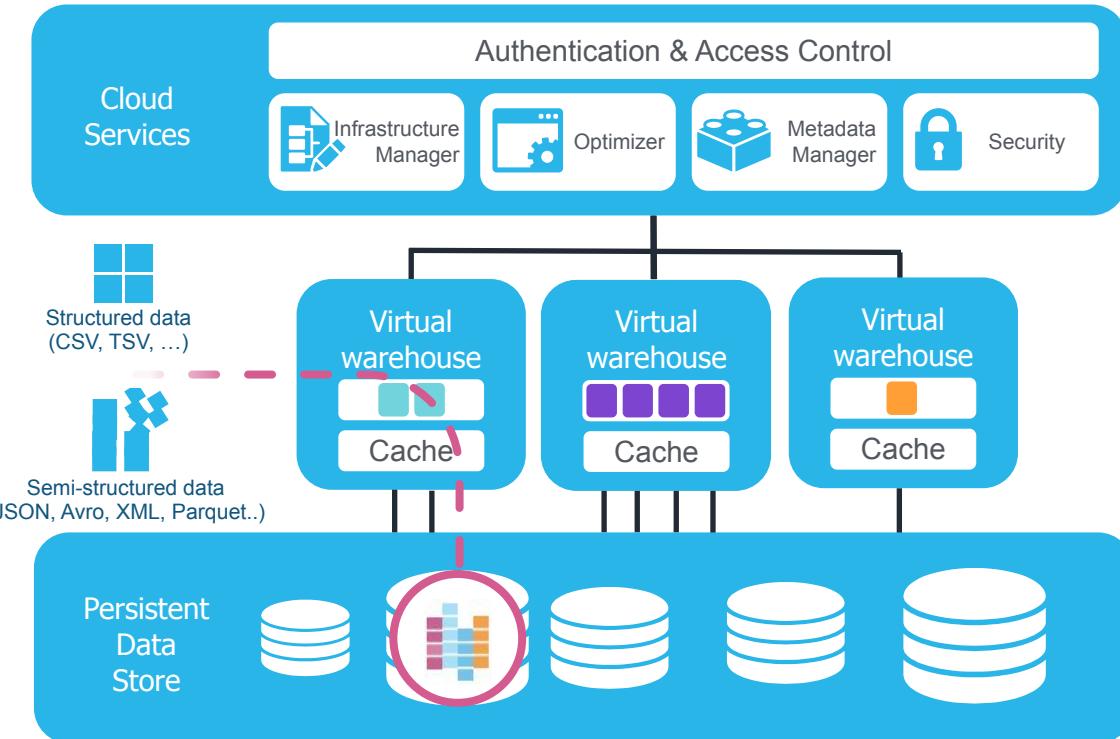
Stored in Snowflake-managed cloud storage

Optimized, proprietary file format

Automatically compressed & encrypted



DATA STORAGE & DATA LOADING



Metadata Created

Information about data stored in Snowflake
Stored & managed in cloud services layer
Stored in proprietary, fast-access, fault-tolerant,
scalable data store

Database Data Stored

Actual data in Snowflake databases & tables
Stored in Snowflake-managed cloud storage
Optimized, proprietary file format
Automatically compressed & encrypted



HOW TO LOAD DATA INTO SNOWFLAKE

An important concept in Snowflake is called a **STAGE**:

- Stages are named objects used for loading and unloading data
 - Different types for Snowflake Stage objects:
 - Internal stage managed by Snowflake
 - External (AW3 S3) stage
 - External (Microsoft Azure) stage
- Stage options:
- Explicit file format
 - Compression scheme
 - Credentials
 - Error management
 - Validation Practices

Detailed guide:

<https://docs.snowflake.com/en/sql-reference/sql/create-stage.html>



HOW TO LOAD DATA INTO SNOWFLAKE

COPY/PUT Commands:

- Designed for batch loads
- Needs to be scheduled
- Needs a warehouse
- Micro-batching up to a point
- Easy to hit concurrency limits

Snowpipe:

- Designed for continuous loads
- No scheduling
- No warehouse needed
- Server-less billing model
- Designed for high concurrency



HOW TO LOAD DATA INTO SNOWFLAKE

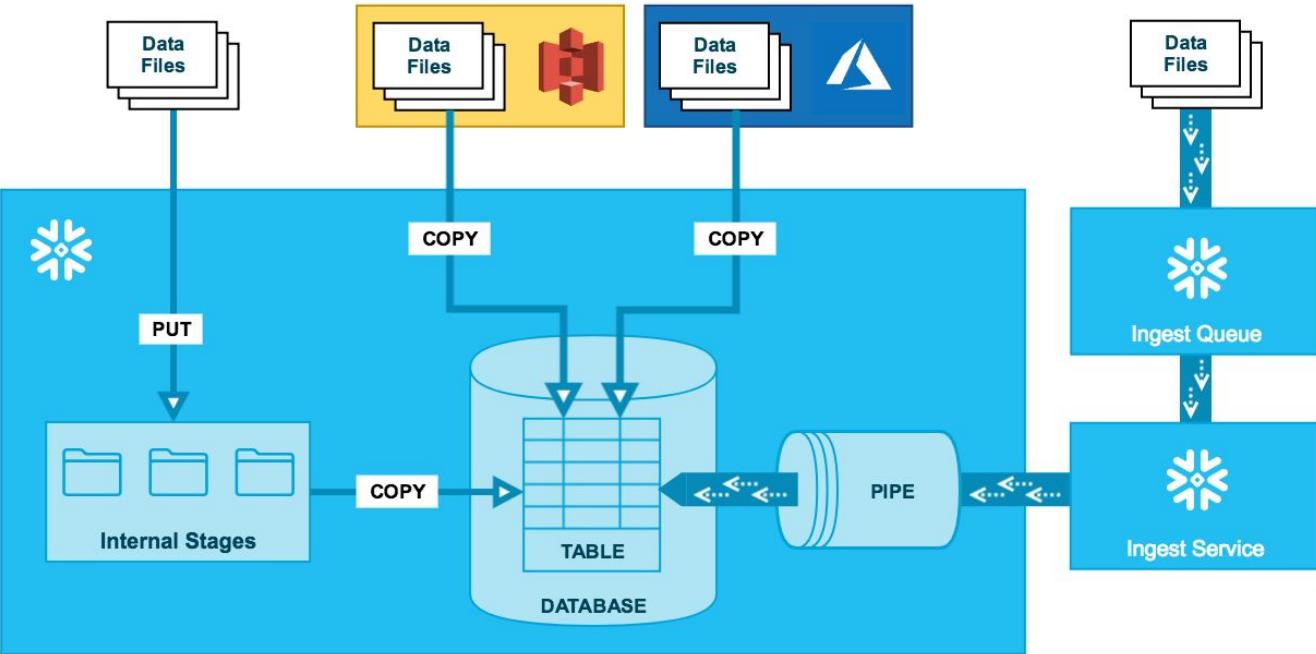
Bulk Loading (using COPY)
* Virtual Warehouse Required *

Continuous Loading (using Snowpipe)
* No Virtual Warehouse Required *

Local File System

External Stages

Data Feed



Batch/Bulk Loading:

- From local files
- From existing cloud landing zones

Continuous Loading:

- Snowpipe
- Serverless model (no user-managed warehouse needed)

HOW TO LOAD DATA INTO SNOWFLAKE

Batch/Bulk loading from cloud storage (S3 or Azure Blob Store):

Create a Snowflake Stage to associate with the Cloud storage file path:

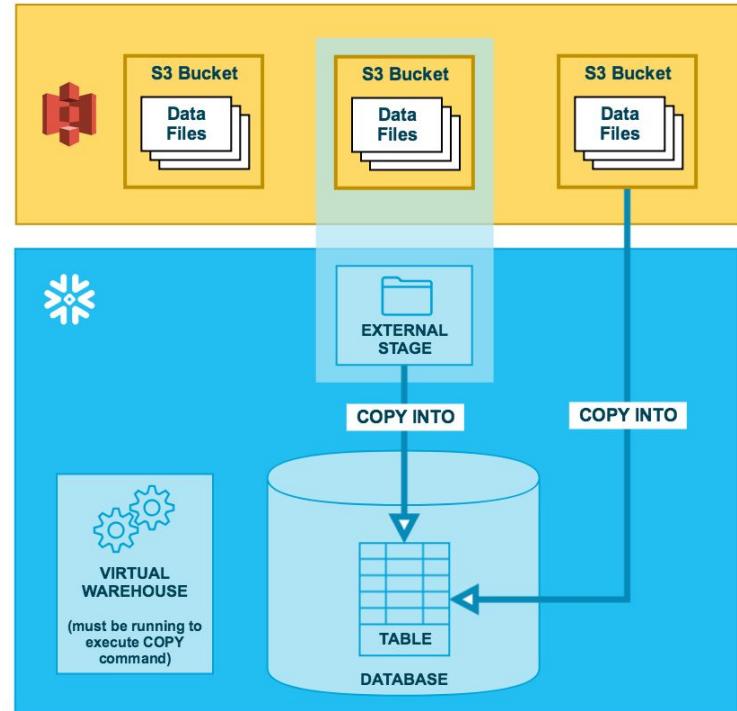
```
CREATE STAGE stage_name
url='s3://load/encrypted_files/'
CREDENTIALS='...'
FILE_FORMAT='...';
```

Use the COPY command to load data into the Snowflake table from the data file(s) in the stage:

```
COPY INTO mytable
FROM @stage_name/folder/folder;
```

Or, you can use the COPY command to load data into a Snowflake table without a stage, like so:

```
COPY INTO mytable
FROM 's3://mybucket/data/files' ;
```



HOW TO LOAD DATA INTO SNOWFLAKE

Batch/Bulk loading from the local filesystem (SnowSQL):

Create a Snowflake Stage:

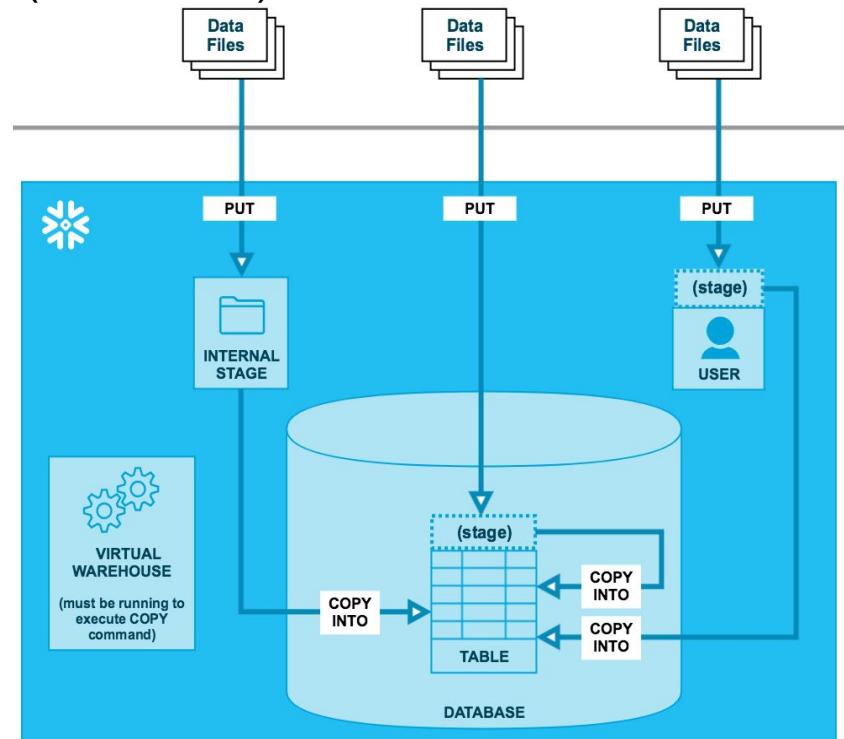
```
CREATE STAGE stage_name  
FIELD_DELIMITER='|'  
FILE_FORMAT='...';
```

Use the PUT command to load data into the Snowflake stage from the data files:

```
put file://c:\temp\contacts*.csv  
@my_stage;
```

Then use the COPY command to load data into a Snowflake table from the stage:

```
COPY INTO mytable  
FROM @my_stage;
```



HOW TO LOAD DATA INTO SNOWFLAKE

Snowflake's Web UI provides a wizard-style interface for batch/bulk loads:

The screenshot illustrates the Snowflake Web UI for loading data into a table. It consists of three main panels:

- Left Panel:** Shows the file selection interface for "MANUAL_LOAD". A CSV file named "Metering.csv" (3.9KB, last modified 11 seconds ago) is selected. A context menu is open over the file, with the "Load into table" option highlighted.
- Middle Panel:** Displays the "Load Data into Table" step for the "Metering" table. It shows the file name "Metering.csv" and the target schema "COMPUTE_WH".
- Right Panel:** Provides detailed configuration for the load. It includes:
 - Select or create a database and schema:** Set to "COMPUTE_WH".
 - File format:** Set to "Delimited Files (CSV or TSV)".
 - Header:** Set to "First line contains header...".
 - Field delimiter:** Set to "Comma (default)".
 - Trim space:** Set to "True (default)".
 - Field optionally enclosed by:** Set to "Double quotes (default)".
 - Replace invalid characters:** Set to "True (default)".
 - Date format:** Set to "Auto* (default)".
 - Time format:** Set to "Auto* (default)".

Table Schema (Right Panel):

DATA TYPE	COLUMN NAME	COLUMN DATA
VARCHAR	SERVICE_TYPE	WAREHOUSE_METERING, WAREHOUSE...
TIMESTAMP_NTZ	START_TIME	2023-07-26T08:00:00-07:00, 2023-07...
TIMESTAMP_NTZ	END_TIME	2023-07-26T09:00:00-07:00, 2023-07...
NUMBER	ENTITY_ID	0, 0, 1, 0, 1
VARCHAR	NAME	CLOUD_SERVICES_ONLY, CLOUD_SERVI...
NUMBER	CREDITS_USED_CC	0, 0, 0.189444444, 0, 0.4525
NUMBER	Minutes_Billed	11.37, 27.15

At the bottom right are buttons for "Cancel", "Back", and "Load".



HOW TO LOAD DATA INTO SNOWFLAKE

Data Loading Considerations: File Formats

- Columnar file formats can be ingested with flexible options
- Leverage file formats and compressions with broad set of options: CSV (text delimited),
 - Format: CSV (text delimited), TSV, JSON, Avro, ORC, Parquet
 - Compression: ZLIB (GZIP, DEFLAT, RAW_DEFLAT), BZ2, Snappy (for Avro)
- File format load performance (2XL):

Source Format	Target Layout	Load Time (sec)	TB/Hr (uncompressed)
CSV (Gzipped)	Structured	1104	15.4
Parquet (Snappy comp)	Semi-structured	3518	4.8
Parquet (Snappy comp)	Structured	3095	5.4
ORC (Snappy comp)	Semi-structured	3845	4.4
ORC (Snappy comp)	Structured	2820	6.0

Load performance for CSV is fast, but not fast enough to convert everything to CSV!

- Factors that might affect load rates:
 - Location of your S3 buckets
 - Number and types of columns
 - Gzip Compression efficiency



Next Lab (10 minutes)

Tasty Bytes Setup



1. **NOT NEEDED IF
USING
tb_introduction.sql**



HOW TO LOAD DATA INTO SNOWFLAKE

Data Loading Considerations: File Location

- Organize data in logical paths using identifier such as application, along with the file create date

Example: /mybucket/application/2018/09/05/

- Run concurrent COPY statements to take advantage of parallel operations
- In each COPY command, narrow the path to the most granular level for best performance

For example, if files have different suffixes, include matching part of the file names in the path
FILES:

mydata01.csv, mydata02.csv, mydata03.csv, ...

COMMAND:

```
copy into t1 from @%t1/united_states/california/los_angeles/2016/06/01/11/mydata;
```

- Listing multiple files may provide the best ingestion performance COMMAND:

```
copy into...FILES=('/2018/09/05/custdata', '/2018/09/06/custdata', '...')
```



HOW TO LOAD DATA INTO SNOWFLAKE

Data Loading Considerations: File Size

- File size and the number of files are crucial for data loading performance
- Split large files before loading them into Snowflake to optimize performance
 - Recommended file size range: 100MB to 250MB
 - Have a sufficient number of files to take advantage of the full parallelism of the virtual warehouse; for example, for an XL warehouse, 128 files can be processed in parallel by the available 128 threads.

Warehouse Size	# of COREs	# of Threads
XS	4	8
S	8	16
M	16	32
L	32	64
XL	64	128

- Limitations
 - Semi-structured data size limitations: VARIANT type has a upper limit of 16MB (compressed data)
 - Parquet limitations: Max file size is 1G, smaller may be even better, for better parallelism of ingestion job when using a larger warehouse



HOW TO LOAD DATA INTO SNOWFLAKE

Data Loading Considerations: Transformations

- COPY command supports column reordering, column omission, and casts using a SELECT statement
 - NOT SUPPORTED: Joins, filters, aggregations
 - Can include SEQUENCE columns, current_timestamp(), or other column functions during data load

```
copy into home_sales(city, zip, sale_date, price)
  from (select substr(t.$2,4), t.$1, t.$5, t.$4 from @mystage t)
file_format = (format_name = mycsvformat);
```

```
-- Convert the staged CSV column data to the specified data types before loading it
copy into casttb(col1, col2, col3)
from (
  select to_binary(t.$1, 'utf-8'),to_decimal(t.$2, '99.9', 9, 5),to_timestamp_ntz(t.$3)
  from @~/datafile.csv.gz t
)
file_format = (type = csv);
```



HOW TO LOAD DATA INTO SNOWFLAKE

Data Loading Considerations: Monitoring

- Monitor the status of each COPY command run on the History tab page of the Snowflake UI
- Use the LOAD_HISTORY Information Schema view to retrieve the history of data loaded into tables using the COPY command

```
select * from information_schema.load_history
  where schema_name=current_schema() and
    table_name='MYTABLE' and
    last_load_time > 'Fri, 01 Apr 2016 16:00:00 -0800';
```

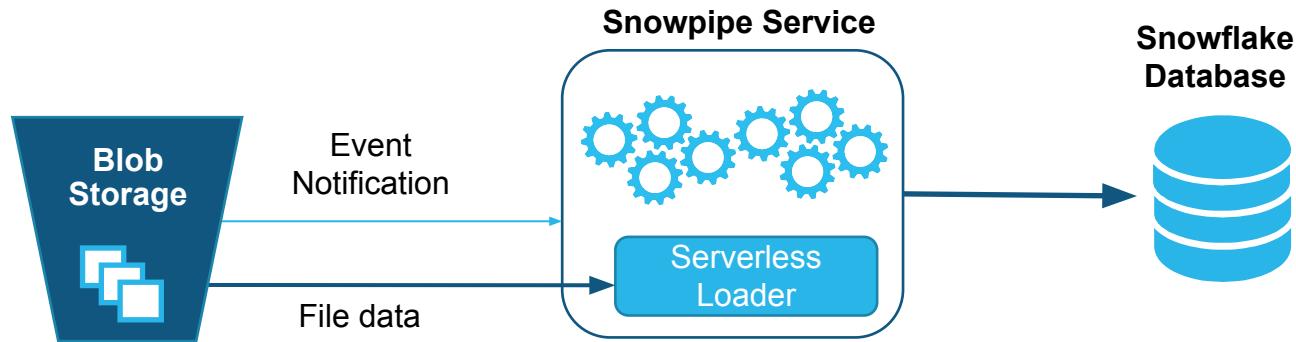
- From the previous example:

TABLE_NAME	LAST_LOAD_TIME	STATUS	ROW_COUNT	ROW_PARSED	FIRST_ERROR_MESSAGE
TEST	2018-11-20 07:44:22.868 -0800	LOAD_FAILED	0	3	Date '1,2,3' is not recognized



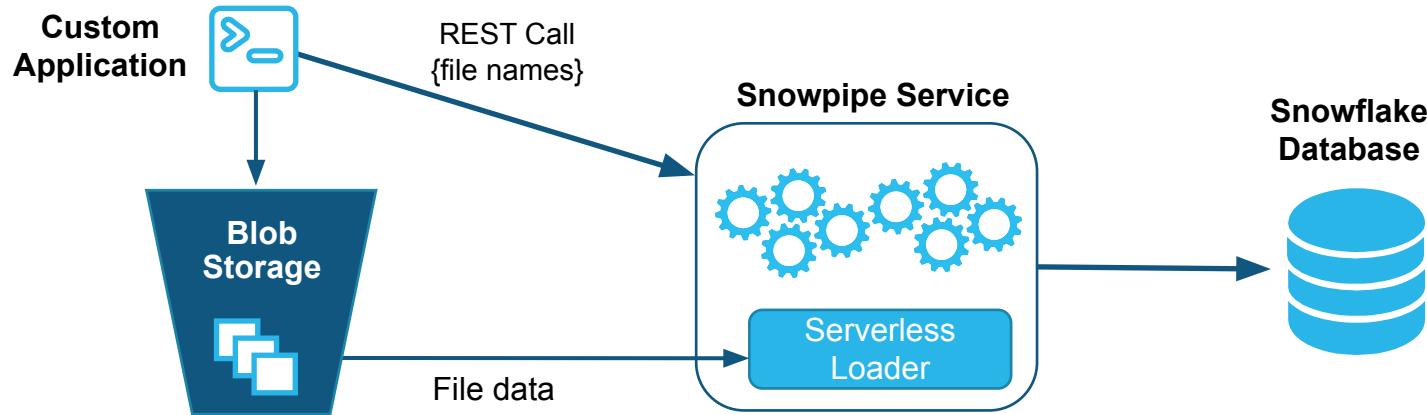
HOW TO LOAD DATA INTO SNOWFLAKE

Snowpipe loading from cloud storage (S3 or Azure Blob Store):



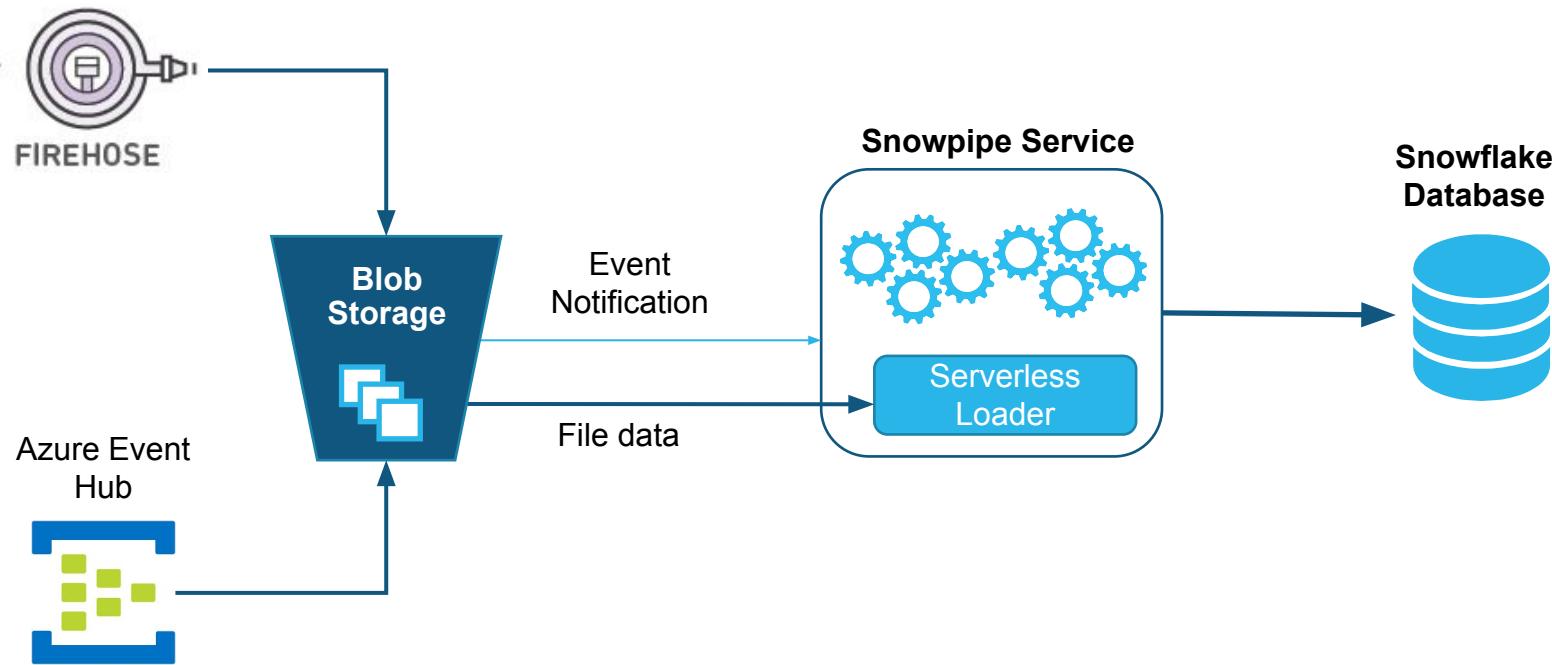
HOW TO LOAD DATA INTO SNOWFLAKE

Snowpipe loading from cloud storage (S3 or Azure Blob Store):



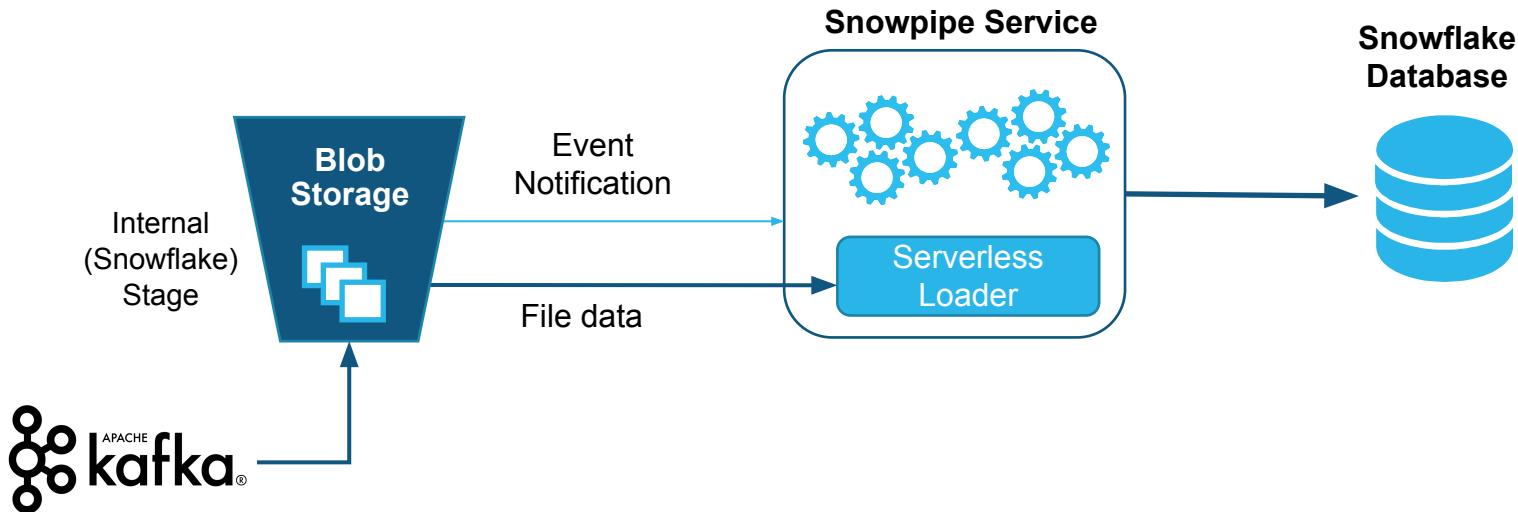
HOW TO LOAD DATA INTO SNOWFLAKE

Snowpipe loading from Amazon Kinesis Firehose or Azure Event Hub:



HOW TO LOAD DATA INTO SNOWFLAKE

Snowpipe loading from Kafka:



HOW TO LOAD DATA INTO SNOWFLAKE

IMPORTANT NOTE:

- Load frequency is approximately once per minute
- Although older files are typically loaded first, there is no guarantee that files are loaded in the same order they are staged, since multiple processes pull files from the queue
- Multiple smaller files are recommended over larger files since individual file processing is single-threaded



HOW TO LOAD DATA INTO SNOWFLAKE

SNOWPIPE DDL

Pipe Management:

- CREATE PIPE
- ALTER PIPE
- DROP PIPE
- DESCRIBE PIPE
- SHOW PIPES

Access Control:

- GRANT <privileges> ... TO ROLE
- REVOKE <privileges> ... FROM ROLE
- SHOW GRANTS



HOW TO LOAD DATA INTO SNOWFLAKE

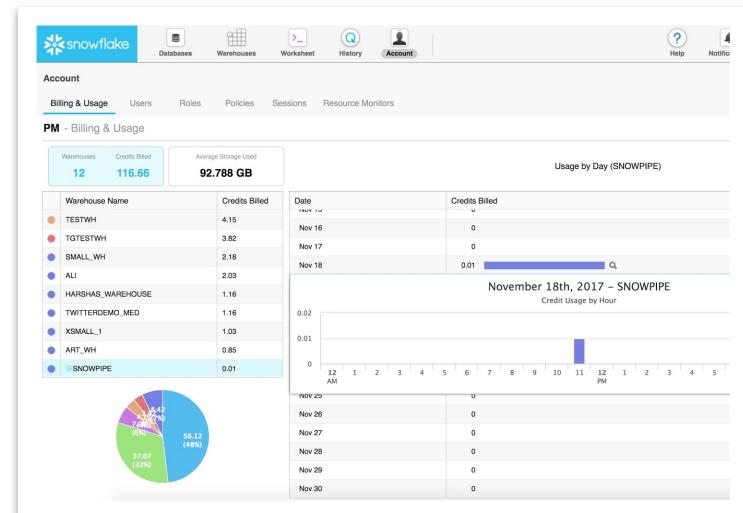
SNOWPIPE PRICING

Utilization-based serverless billing model:

- No warehouse to manage for load
- Per core, per second granularity
- Charged as Snowflake credits

Components of the Snowpipe charges:

- Loading work (idle times not charged)
- Plus 0.06 credits per 1000 queued files



HOW TO LOAD DATA INTO SNOWFLAKE

SNOWFLAKE DATA INTEGRATION PARTNER PRODUCTS (PARTIAL LIST):



Your Turn - Tasty Bytes



Zero To Snowflake



1. Financial
Governance



2. Transformation



3. Semi -
Structured Data



4. Data
Governance



5. Collaboration



6. Geospatial

Transformation

1. Cloning
2. Result Set Cache
3. Table (ADD COLUMN, UPDATE)
4. Time-Travel
5. Table SWAP, DROP and UNDROP

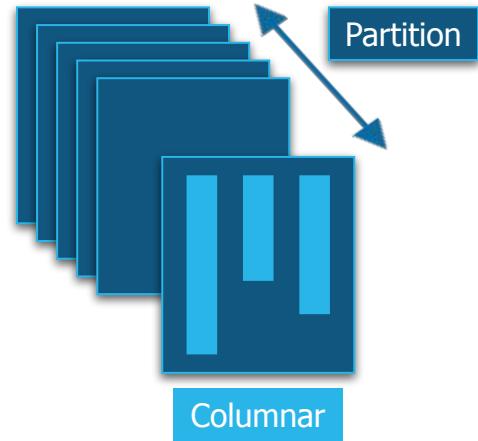
Benefits

1. Consume zero additional storage for cloned database
2. No additional compute when using result set cache
3. Reduce data risks with Time-Travel



SNOWFLAKE'S AUTOMATIC MICRO-PARTITIONING

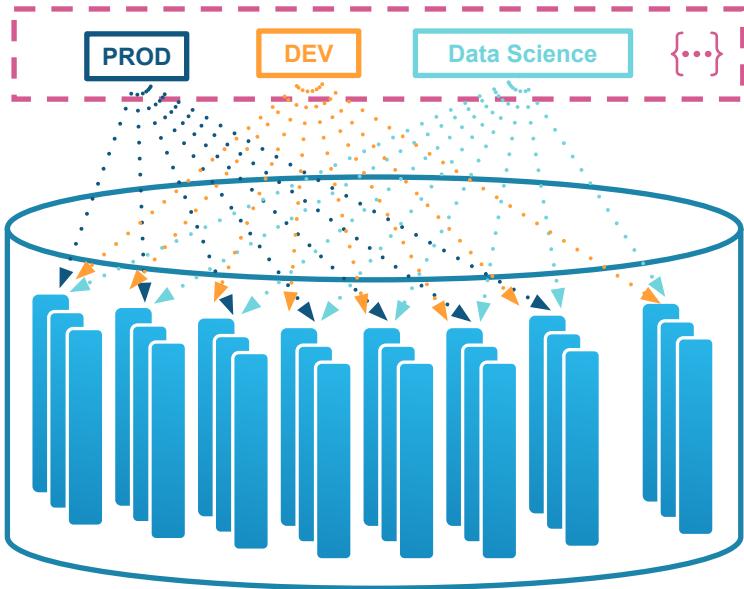
- Data is **automatically** partitioned in Snowflake on natural ingestion order
 - Doesn't require a partition definition up front
 - Partitions are defined based on size (few megabytes) → no skew
 - Natural ingestion order maintains correlations between columns
- Enable horizontal and vertical **pruning**
 - First by partition and then by column
 - True columnar storage within partitions
- **Micro**: Partitions are kept small automatically (few 10s MBs)
 - Enables very fine-grained pruning
 - System designed to handle millions of partitions
- Metadata structure maintains **data distribution** for each column and each partition



Partitions are the unit of pruning and DML operations



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



Next Lab - Transformation

Tasty Bytes



1. Zero Copy Clone



```
28 USE ROLE tasty_dev;
29
30 CREATE OR REPLACE TABLE frostbyte_tasty_bytes.raw_pos.truck_dev
31     CLONE frostbyte_tasty_bytes.raw_pos.truck;
```



Adaptive Caching

Metadata

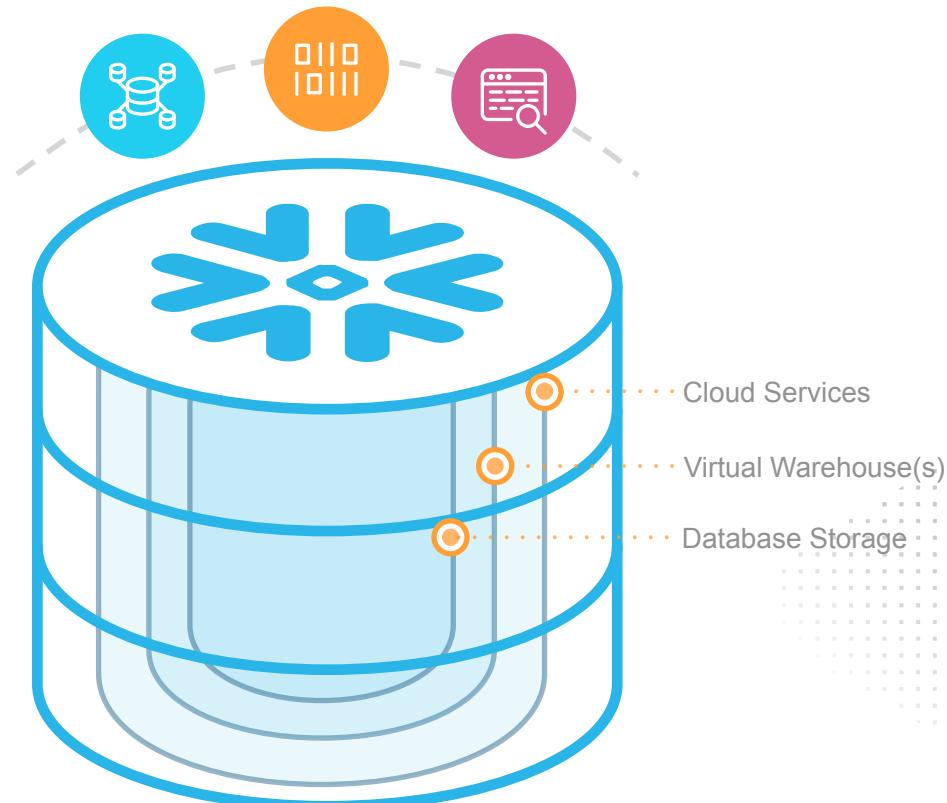
Metadata cached for fast access during query planning

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)



Next Lab - Cache

Tasty Bytes



```
43 -- Section 4: Step 1 - Querying our Cloned Table
44 USE WAREHOUSE tasty_dev_wh;
45
46
47 SELECT
48     t.truck_id,
49     t.year,
50     t.make,
51     t.model
52 FROM frostbyte_tasty_bytes.raw_pos.truck_dev t
53 ORDER BY t.truck_id;
54
55 -- Section 4: Step 2 - Re-Running our Query
56 SELECT
57     t.truck_id,
58     t.year,
59     t.make,
60     t.model
61 FROM frostbyte_tasty_bytes.raw_pos.truck_dev t
62 ORDER BY t.truck_id;
63
64
```

Results

Chart

TRUCK_ID	YEAR	MAKE	MODEL
1	2009	Ford_	Step Van
2	2015	Ford_	Step Van
3	2004	Freightliner	MT45 Utilimaster
4	1997	Chevrolet	P30

Query Details

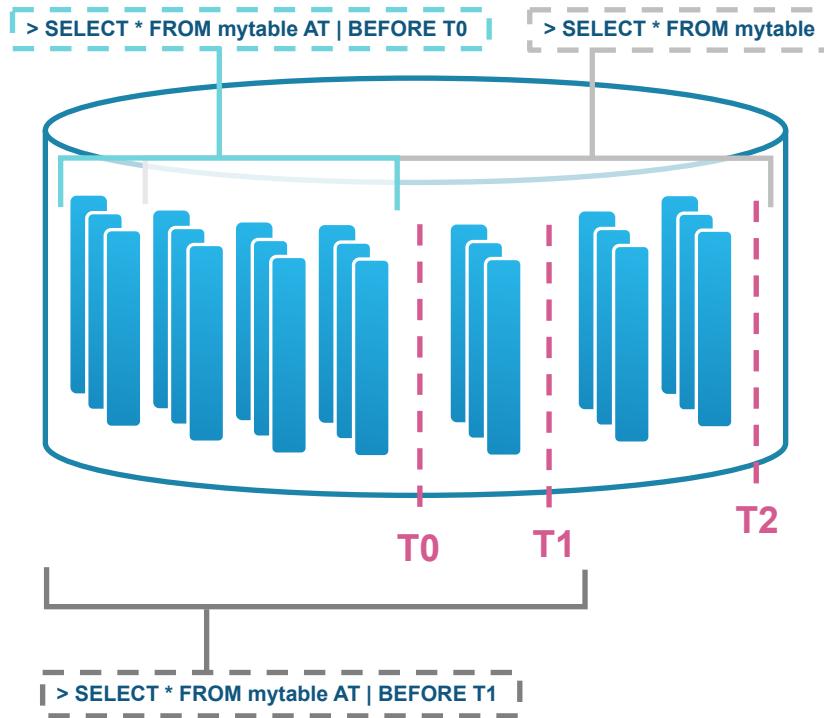
Query duration 33ms

Rows 450

Query ID 01b2aa69-0002-238d-...



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



Lab - Time Travel

Tasty Bytes `tb_zts_transformation.sql`



Snowflake makes it very easy to recover from common mistakes!



- Start at line 65, run commands sequentially through line 112.
 - What do you notice? What's the problem with the query at line 103-104?
 - How are we going to recover?
- Use next statements (up to line 155) to FIND the bad query and FIX the problem
 - No downtime!

```
-- Section 7: Step 3 - Leveraging Time-Travel to Revert our Table
CREATE OR REPLACE TABLE frostbyte_tasty_bytes.raw_pos.truck_dev
AS
SELECT * FROM frostbyte_tasty_bytes.raw_pos.truck_dev
BEFORE(STATEMENT => $query_id);
```

STOP at Line 155

Lab - Transformation

Tasty Bytes



1. SWAP (~line 175)
2. UNDROP



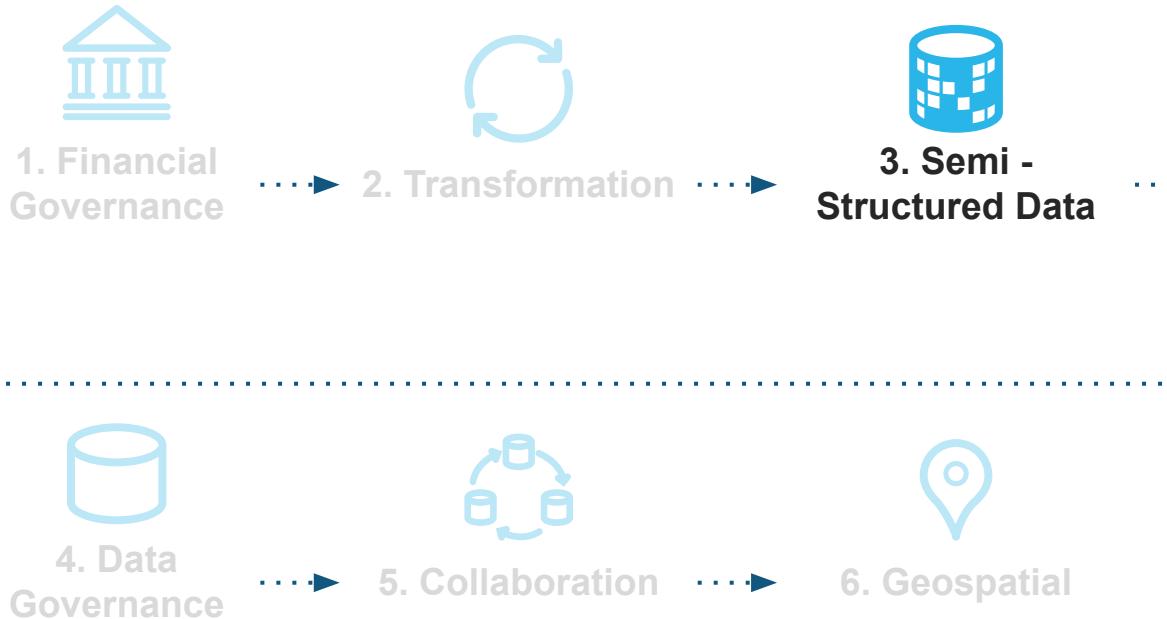
- SWAP is really useful for development flows
 - Make a change in a clone, run tests, swap if successful
- UNDROP is a lifesaver!





© 2024 Snowflake Inc. All Rights Reserved

Zero To Snowflake



Semi-Structured Data

1. Semi- Structured Data VARIANT
2. Dot Notation, Flattening and Arrays
3. View Creation

Benefits

1. Unsiloed access to all your data with near-infinite scale
2. Enable data analysts to quickly move from data to action

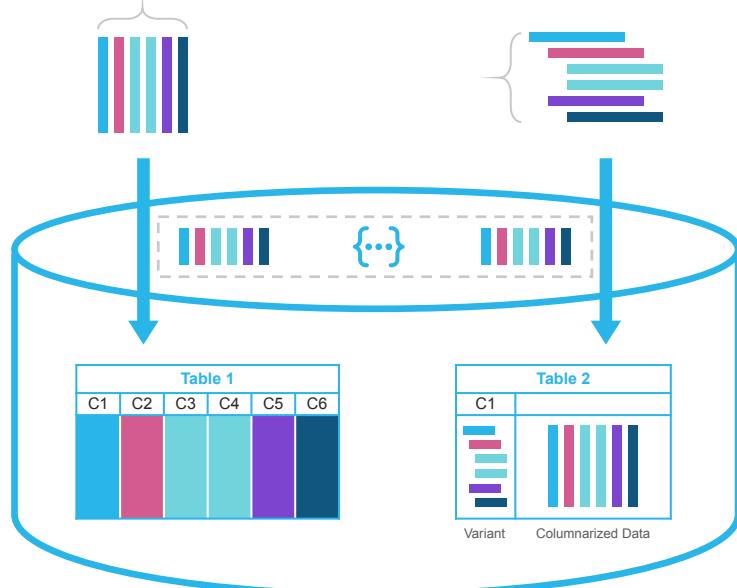


Semi-Structured Data

Native support for JSON, XML, Avro, Parquet, ORC

Structured data

(Delimited or fixed width files)



Semi-structured data

(Delimited or fixed width files)

Structured formats (CSV, TSV, ...)

- Strongly-typed “columns”
- Typically map 1:1 to table columns
- Ingestion process generates important metadata

Semi-structured formats (JSON, XML, ...)

- Traditional DBs require complex transforms to “flatten” data into structures.
- Snowflake has a VARIANT data type
- Stores original document as-is
- During ingestion, data is columnized and metadata collected
- SQL syntax is a simple dot notation



WHAT IS JSON?

- JSON (JavaScript object notation) is a lightweight, plain-text, data interchange format
 - Snowflake accepts wide range of JSON and JSON-like inputs that can be interpreted consistently
- JSON data is a hierarchical collection of name/value pairs grouped into objects and arrays
 - Colons : separate names and values in name/value pairs
 - Curly braces {} denote objects
 - Square brackets [] denote arrays
 - Commas , separate entities in objects and arrays
- Supported data types for values are:
 - A number (integer or floating point)
 - A string (in double quotes)
 - A Boolean (true or false)
 - An array (in square brackets)
 - An object (in curly braces)
 - Null
 - Rules for JSON keys are different from Snowflake SQL Identifiers. Please refer to <http://json.org>



Lab - Semi-Structured - 20 minutes

Tasty Bytes `tb_zts_semi_structured.sql`



Snowflake eats semi-structured data for lunch!

The screenshot shows the Snowflake documentation website at docs.snowflake.com/en/sql-reference/functions-semistructured. The page is titled "DOCUMENTATION". The navigation bar includes "ed", "Guides", "Developer", "Reference" (which is underlined), "Releases", "Tutorials", and "Status". The main content area is titled "Array/Object Creation and Manipulation" and lists various functions:

Function
ARRAY_AGG
ARRAY_APPEND
ARRAY_CAT
ARRAY_COMPACT
ARRAY_CONSTRUCT
ARRAY_CONSTRUCT_COMPACT
ARRAY_CONTAINS
ARRAY_DISTINCT
ARRAY_EXCEPT
ARRAY_FLATTEN
ARRAY_GENERATE_RANGE
ARRAY_INSERT
ARRAY_INTERSECTION

-
- Walk through the lab material
 - HIGHLIGHTS
 - “Dot notation” (lines 41+)
 - Lateral Flatten
 - ARRAY functions



TRAVERSING JSON DATA

- To access any of the first-level elements, use : between the variant column name and element like this: <column_name>:<element>
- To access elements in a JSON object other than first-level, you can use either:
 - Dot notation (for example: payload:phoneNumber.number)
 - Generic example of traversing a path using dot notation -
<column_name>:<level1_element>.<level2_element>.<level3_element>
 - Bracket notation (for example: payload['age'])
- Query output of <column_name>:<element> is enclosed in double quotes and the datatype is VARIANT
- Explicitly cast the values to desired data types using :: notation (for example: payload:fullName::varchar)
- Column name is case-insensitive but element names are case-sensitive
- To retrieve a specific instance of a child element in a repeating array, use array reference or flatten the array
 - To get the age of first child of element named children, use this expression: payload:children[0].age



USING FLATTEN FUNCTION

- Parse an array using FLATTEN function.
- FLATTEN function explodes VARIANT, OBJECT or ARRAY elements into multiple rows (one row per element)
- FLATTEN function can be used to convert semi-structured data to a relational representation
- Use either table function or lateral join to be able to access elements returned by table expression
 - Example of Lateral Join to explode the top-level elements

```
select v.*
```

```
from stg_json_nested_array, lateral flatten(payload) v;
```

- Example of table function to explode the top-level elements

```
select v.*
```

```
from stg_json_nested_array, table(flatten(payload)) v;
```

- Use FLATTEN function with recursive=> true to explode elements at all levels

- Example of Lateral join to explode elements at all levels

```
select v.*
```

```
from stg_json_nested_array, lateral flatten(payload, recursive=>true) v;
```



LATERAL FLATTEN

```
select
    v.seq, v.key, v.path, v.index, v.value
from stg_json_nested_array s
, lateral flatten(s.payload) v
where v.seq = 1;
```

Results Data Preview ◀ Open History

✓ [Query ID](#) [SQL](#) 74ms 7 rows

Filter result... Columns ▾

Row	SEQ	KEY	PATH	INDEX	VALUE
1	1	age	age	NULL	22
2	1	children	children	NULL	[{"age": "6", "gender": "Female", "name": "..."}]
3	1	citiesLived	citiesLived	NULL	[{"place": "Seattle", "yearsLived": ["1995..."]}]
4	1	fullName	fullName	NULL	"John Doe"
5	1	gender	gender	NULL	"Male"
6	1	kind	kind	NULL	"person"
7	1	phoneNumber	phoneNumber	NULL	{"areaCode": "206", "number": "1234567"}



LATERAL FLATTEN (recursive=>true)

```
select v.seq, v.key, v.path, v.index, v.value  
from stg_json_nested_array s  
, lateral flatten(s.payload, recursive=>true) v  
where v.seq = 1;
```

Results Data Preview [← Open History](#)

✓ [Query ID](#) [SQL](#) 122ms  25 rows

Filter result... [Download](#) [Copy](#) Columns ▾ [X](#)

Row	SEQ	KEY	PATH	INDEX	VALUE
1		1 age	age	NULL	22
2		1 children	children	NULL	[{"age": "6", "gender": "Female", "name": "..."}]
3		1 NULL	children[0]	0	{"age": "6", "gender": "Female", "name": "J..."} ...
4		1 age	children[0].age	NULL	"6"
5		1 gender	children[0].gender	NULL	"Female"
6		1 name	children[0].name	NULL	"Jane"
7		1 NULL	children[1]	1	{"age": "15", "gender": "Male", "name": "Jo..."} ...
8		1 age	children[1].age	NULL	"15"
9		1 gender	children[1].gender	NULL	"Male"
10		1 name	children[1].name	NULL	"John"
11		1 citiesLived	citiesLived	NULL	[{"place": "Seattle", "yearsLived": ["1995..."]}]

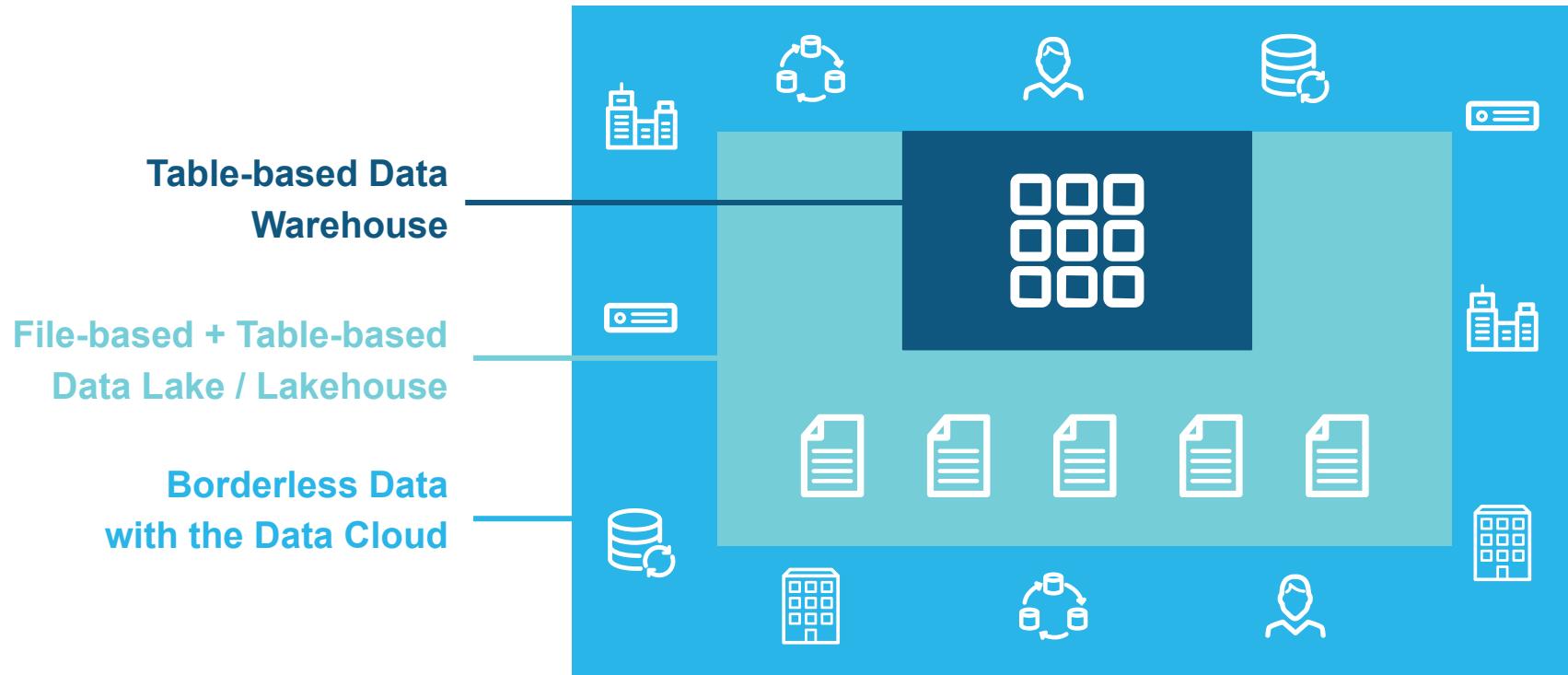


Snowflake Collaboration



© 2024 Snowflake Inc. All Rights Reserved

Evolution of Data Management



Benefits of Snowflake Collaboration

Across Cloud & Region with Snowgrid

Delivers direct access to live, ready-to-query data across clouds and regions with auto-fulfillment and no ETL

More than Data

Snowflake enables customers to collaborate with data, data services and applications including built-in usage based monetization.

Robust Data Governance

Achieve privacy-preserving collaboration with targeted discovery, revocable access and custom event logging.



Snowflake Regions



AWS



Azure



GCP



Benefits of Snowflake Collaboration

Across Cloud & Region with Snowgrid

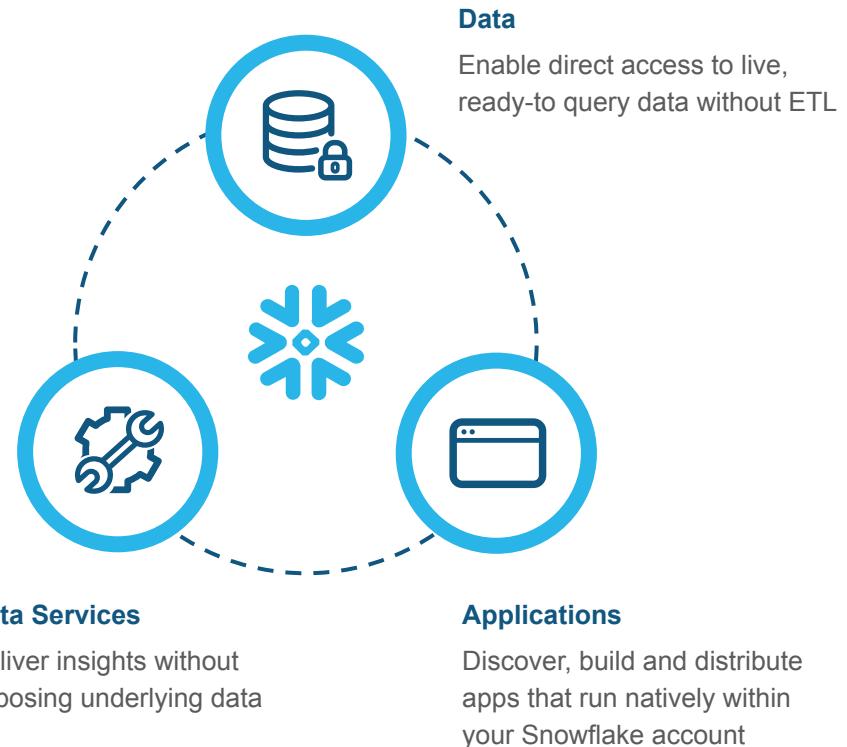
Delivers direct access to live, ready-to-query data across clouds and regions with auto-fulfillment and no ETL

More than Data

Snowflake enables customers to collaborate with data, data services and applications including built-in usage based monetization.

Robust Data Governance

Achieve privacy-preserving collaboration with targeted discovery, revocable access and custom event logging.



Benefits of Snowflake Collaboration

Across Cloud & Region with Snowgrid

Delivers direct access to live, ready-to-query data across clouds and regions with auto-fulfillment and no ETL

More than Data

Snowflake enables customers to collaborate with data, data services and applications including built-in usage based monetization.

Robust Data Governance

Achieve privacy-preserving collaboration with targeted discovery, revocable access and custom event logging.

Discovery

- Single Account
- Account Group
- Cloud Region(s)
- Public Marketplace

Audit

- Listing Views & Events
- Jobs Run by Consumer
- Object & Columns Accessed
- Custom Event Logging



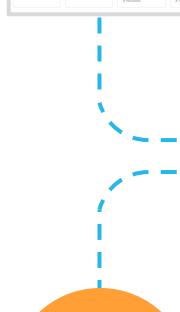
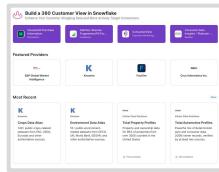
Access

- Row Access Policies
- Dynamic Data Masking
- Conditional Masking
- Query Constraints*

Collaboration in the Data Cloud

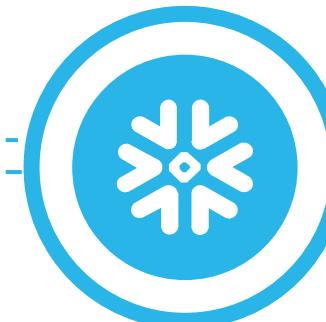
Governed, privacy-preserving collaboration for every scenario

Discover and Monetize via
Snowflake Marketplace

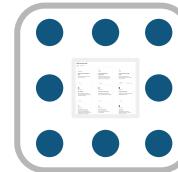


Managed
or Referral
Accounts

Share with Companies
Not Yet on Snowflake



Share Across Your
Business Ecosystem

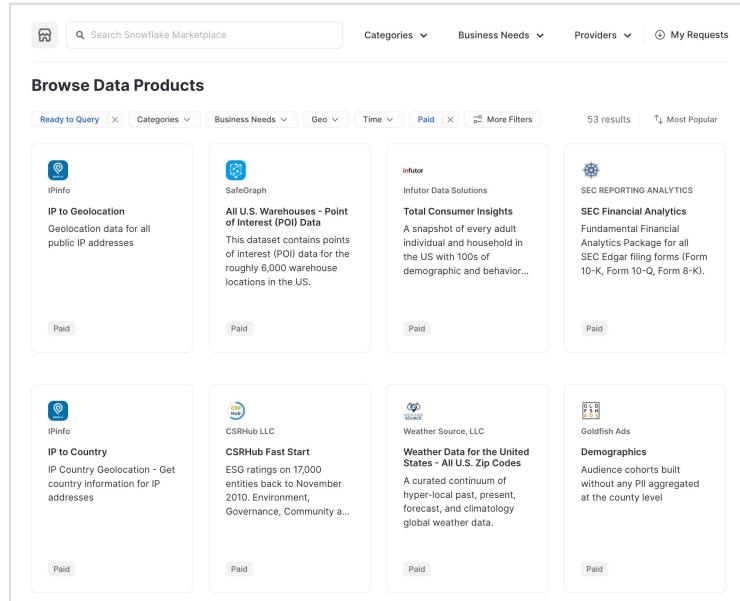


Analyze Data
without Exposing It via
Global Data Clean Rooms

Live, ready-to-query data, services and apps cross-cloud and cross-region. No ETL.

Discover & Monetize via Snowflake Marketplace

SNOWFLAKE MARKETPLACE



Discover data, services and apps from 400+ providers across 18+ categories*



Access the most current data available and receive real-time updates automatically



Reduce data integration costs with direct access to live, ready-to-query data; No ETL

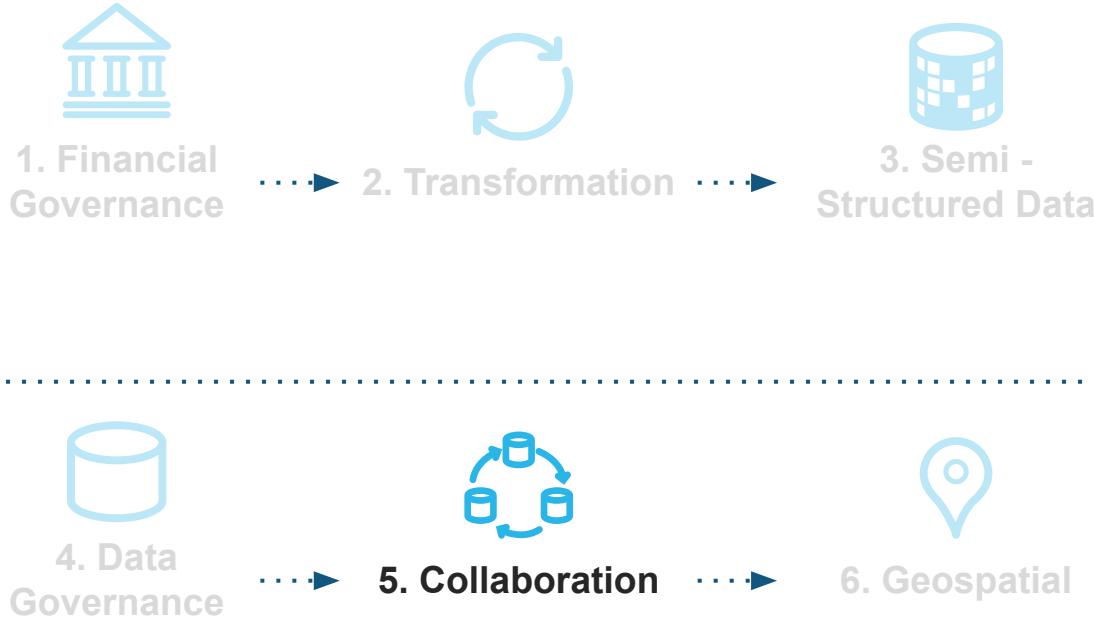


Monetize your products with built-in usage based pricing and visibility into customer adoption

* as of Apr 31, 2023

Live, ready-to-query data, services and apps cross-cloud and cross-region. No ETL.

Zero To Snowflake



Collaboration

Part 1 - Public, Free Marketplace Listing

1. Acquire Free Listing - Weathersource
2. SQL Functions - Length and Temperature conversions
3. Harmonizing Orders and Weather - Views

Part 2 - Personalized, Marketplace Listing

1. Explore Personalized - Safegraph POI Listing
2. Analyze Top Selling Locations + Common Table Expression

Benefits

1. Collaboration across teams, workloads, clouds and data, seamlessly and securely
2. Tap into the Snowflake Marketplace to discover third party data, data services or applications
3. Reduce data integration costs with direct access to live, ready-to-query data; No ETL



Lab - Collaboration

Tasty Bytes `tb_zts_collaboration.sql`



The best ETL is no ETL!

The screenshot shows the Snowflake Marketplace interface. On the left, a sidebar menu includes 'Projects', 'Data', 'Data Products', 'Marketplace' (which is selected and highlighted in blue), 'Installed Apps', and 'App Packages'. The main content area displays search results for 'frostbyte' in the 'Data Products' category. A card for 'SafeGraph: frostbyte' is visible, along with a larger card for 'Weather Source LLC: frostbyte'. This card provides a brief description: 'Weather Source provides actionable insights for a wide range of use cases across industries.' To the right, a detailed description for 'Weather Source LLC: frostbyte' is shown under the heading 'Get it for Free'. It highlights features like hourly data updates, global data at postal code level, historical and forecast data from 2019 to present, and real-time forecasts. Below this is an 'Options' section where the user can set the database name to 'FROSTBYTE_WEATHERSOURCE' and grant access to the 'PUBLIC' role. A large blue 'Get' button is at the bottom.

Get it for Free

Ready to Query
No Storage Cost

Weather Source LLC: frostbyte

- ✓ Data updated hourly
- ✓ Data for global at postal code level
- ✓ Data after 01-01-2019 with daily interval
- ✓ Historical Actuals (2019 - present) in daily & hour intervals
- ✓ Historical Forecast (2019 - present) in daily & hour intervals
- ✓ Real-time Forecast in hourly & daily format with d...

Options

Database name

FROSTBYTE_WEATHERSOURCE

A new database will be created. Consumes no storage.

Which roles, in addition to ACCOUNTADMIN, can access this database?

PUBLIC

Get



Lab - Collaboration

Tasty Bytes tb_zts_collaboration.sql



Lab - Collaboration

User Defined Functions

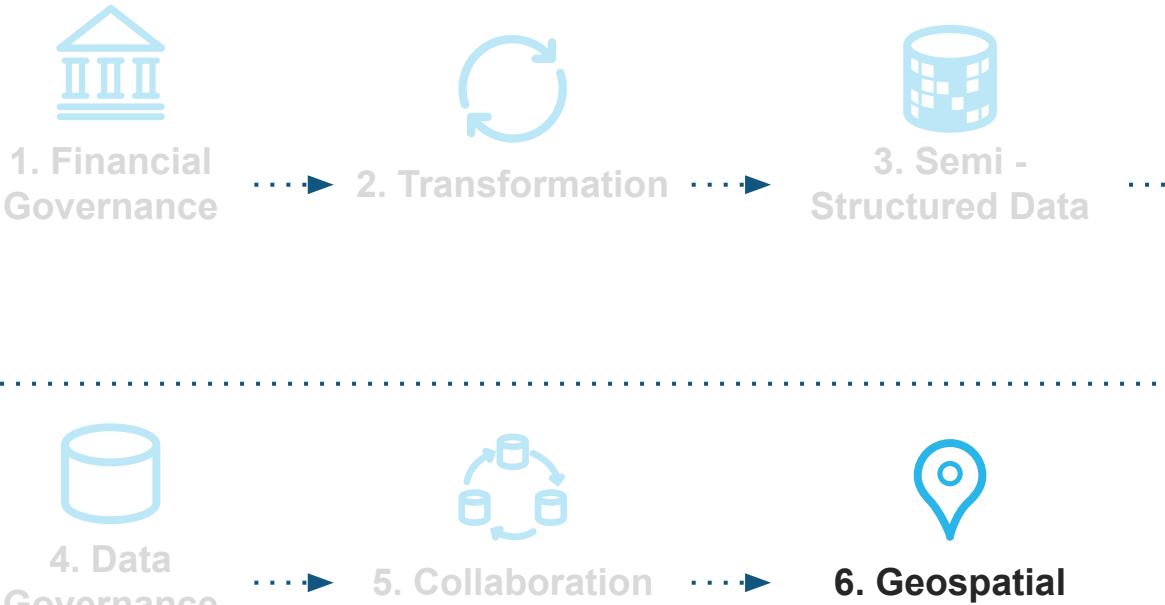


- Embed business logic
- Multiple languages (but usually SQL or Python)

```
155 CREATE OR REPLACE FUNCTION frostbyte_tasty_bytes.analytics.fahrenheit_to_celsius(temp_f NUMBER(35,4))
156 RETURNS NUMBER(35,4)
157 AS
158 $$
159     (temp_f - 32) * (5/9)
160 $$;
161
162     --> create the SQL function that translates Inches to Millimeter
163 CREATE OR REPLACE FUNCTION frostbyte_tasty_bytes.analytics.inch_to_millimeter(inch NUMBER(35,4))
164 RETURNS NUMBER(35,4)
165     AS
166 $$
167     inch * 25.4
168 $$;
```



Zero To Snowflake



Geospatial

1. Create a Geographic Point from Safegraph Latitude and Longitude Data
2. Calculate Distance between our Top Selling Locations
3. Collect Coordinates and find a Bounding Polygon, its area, and Center Point of our Top Selling Locations
4. Derive Locations furthest away from our Top Selling Hub

Benefits

1. Execute geospatial intense calculations at scale directly in Snowflake
2. Integrate geospatial calculations with BI and ETL tools
3. Utilize native industry standard geospatial data types



Lab - Geospatial

Tasty Bytes `tb_zts_geospatial.sql`



< frostbyte X

Search for "frostbyte"

Data Products

 SafeGraph: frostbyte

 Weather Source LLC: frostbyte

This data
but is pr

SafeGraph's Places data provides detailed information about physical places. Our data is provided in a format that empowers the user. With POI data for countries around the world, users can gain insights about any location that a person can visit aside from private residences. C...

Browse Data Products

 SafeGraph - SafeGraph: frostbyte X

Get it for Free

Ready to Query
No Storage Cost

SafeGraph: frostbyte

- ✓ Static data
- ✓ Data for global at latitude and longitude level

Options

Database name FROSTBYTE_SAFEGRAPH
A new database will be created. Consumes no storage.

Which roles, in addition to ACCOUNTADMIN, can access this database? PUBLIC

Get



Lab - Geospatial

Tasty Bytes `tb_zts_geospatial.sql`



Highlights:

- Snowflake has a LOT of built-in geospatial functions
 - <https://docs.snowflake.com/en/sql-reference/functions-geospatial>
- In particular, ST_DISTANCE

```
164 | SELECT
165 |     a.location_id,
166 |     b.location_id,
167 |     ROUND(ST_DISTANCE(a.geo_point, b.geo_point)/1609,2) AS geography_distance_miles,
168 |     ROUND(ST_DISTANCE(a.geo_point, b.geo_point)/1000,2) AS geography_distance_kilometers
```



Compliance & Resiliency



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](#)



© 2024 Snowflake Inc. All Rights Reserved

Snowflake Enables Business Continuity

Possible failures and how Snowflake mitigates these

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



Snowflake Horizon

Built-in governance solution with a unified set of capabilities



Compliance

Protect & audit your data with compliance, business continuity, data quality monitoring, & lineage.

Security

Secure your environment with continuous risk monitoring & protections, RBAC, & granular authorization policies.

Privacy

Unlock the value of your sensitive data with advanced privacy policies & data clean rooms.

Interoperability

Integrate with other Apache Iceberg- compatible catalogs & engines, & with data catalog & data governance partners.

Access

Classify, share, discover & take immediate action on data, apps and more across regions & clouds.



A Little Fun.... then a break



Streamlit for Your Food Truck

We will do more later.... But why not have dessert first?!



Projects

Worksheets

Streamlit

Dashboards

App Packages

Welcome to Streamlit Apps!

To let your team create Streamlit Apps, you need to grant some privileges! [Here is an example script.](#)

Streamlit Apps

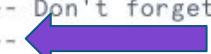
```
-- If you want to create a new database for Streamlit Apps, run  
CREATE DATABASE STREAMLIT_APPS;  
-- If you want to create a specific schema under the database, run  
CREATE SCHEMA APPS; -- Or, you can use the PUBLIC schema that was automatically created with the database.
```

```
-- If you want all roles to create Streamlit apps in the PUBLIC schema, run  
GRANT USAGE ON DATABASE STREAMLIT_APPS TO ROLE PUBLIC;  
GRANT USAGE ON SCHEMA STREAMLIT_APPS.PUBLIC TO ROLE PUBLIC;  
GRANT CREATE STREAMLIT ON SCHEMA STREAMLIT_APPS.PUBLIC TO ROLE PUBLIC;  
GRANT CREATE STAGE ON SCHEMA STREAMLIT_APPS.PUBLIC TO ROLE PUBLIC;
```

```
-- Don't forget to grant USAGE on a warehouse.  
-- ON WAREHOUSE <WAREHOUSE_NAME> TO ROLE PUBLIC;|
```



© 2024 Snowflake Inc. All Rights Reserved



Streamlit for Your Food Truck

We will do more later.... But why not have dessert first?!



Create Streamlit App

App will run with rights of ACCOUNTADMIN

App title

My Locations

App location

STREAMLIT_APPS

APPS

App warehouse

TASTY_DATA_APP_WH

Cancel

Create

Streamlit for Your Food Truck

We will do more later.... But why not have dessert first?!



Create Streamlit App

App will run with rights of ACCOUNTADMIN

App title

App location

App warehouse

Cancel

Create

Streamlit for Your Food Truck

Anywhere after Line 17



```
15 # Get the current credentials
16 session = get_active_session()
17
18
19
20 my_locations_df = session.table("FROSTBYTE_TASTY_BYTES.ANALYTICS.TOP_10_TRUCKS")
21 st.map(my_locations_df.to_pandas())
22 st.dataframe(my_locations_df.to_pandas())
23
24
```

▶ Run

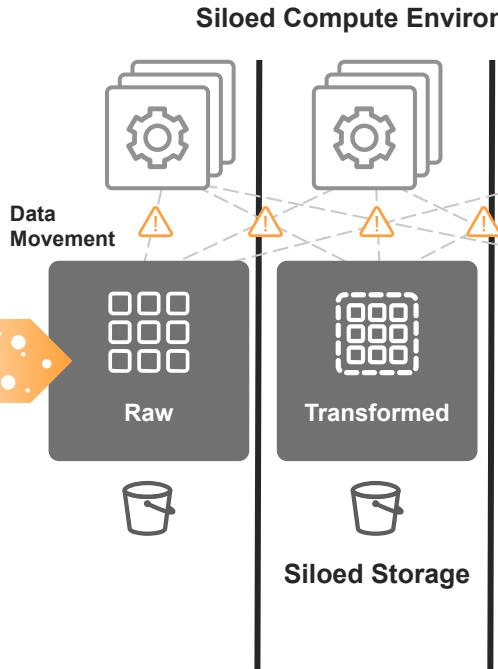


© 2024 Snowflake Inc. All Rights Reserved

Some Data Pipeline Features



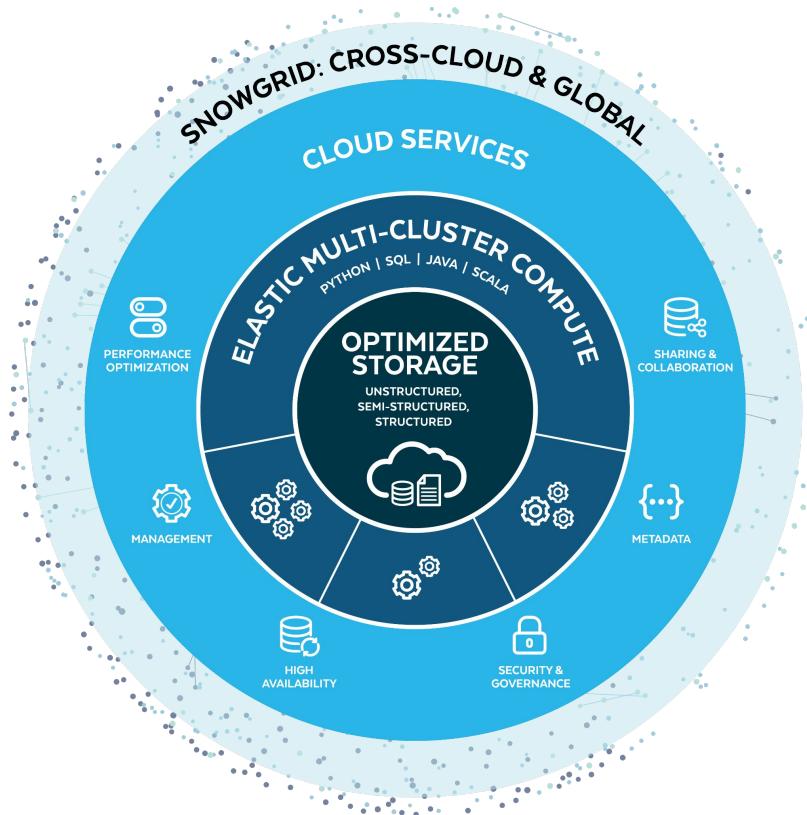
Challenges with Traditional Data Engineering



CHALLENGES

- ✖ Customers often run separate processing clusters for different languages
- ✖ Complex capacity management, resource sizing and configurations
- ✖ Data ends up siloed and difficult to share
- ✖ Loose governance control and security loopholes

Streamlined Architecture with Snowflake



TECHNICAL REQUIREMENTS

- Single platform with native support for different languages
- Automatic capacity management and resource sizing
- Workload isolation and high concurrency
- Consistent governance and security policies

Simple Options for Any Pipeline Use Case

Easy to Batch, Micro-Batch, or Stream Data

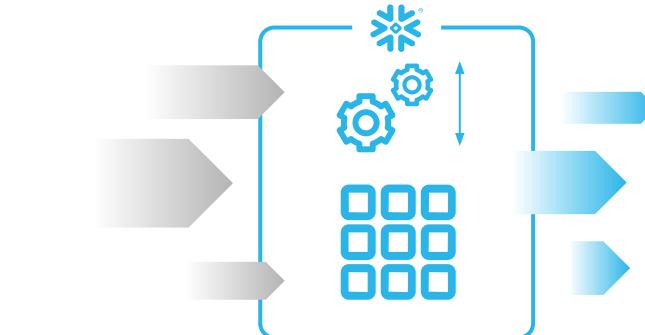
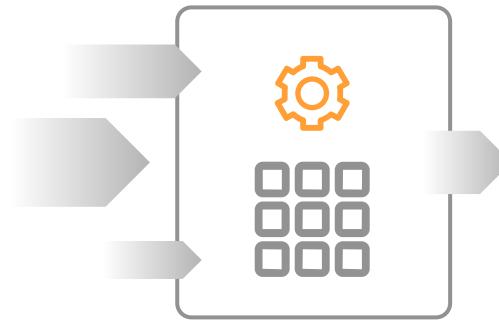
Easy to use capabilities to batch load data into Snowflake, or stream it into Snowflake in seconds

Automatic Adjustments = Lower TCO

Snowflake automatically adjusts to fluctuations in workloads so jobs of any size run without manual configuration. Latency is your tuning knob

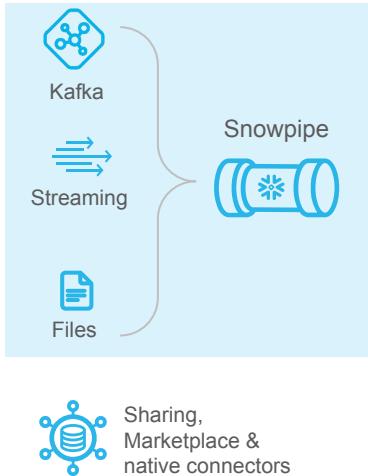
Declarative Pipelines as Easy as SELECT

Dynamic Tables transform data incrementally, only operating on data that has changed since the last refresh, which makes high volume and complex pipelines significantly more cost effective



Key Capabilities

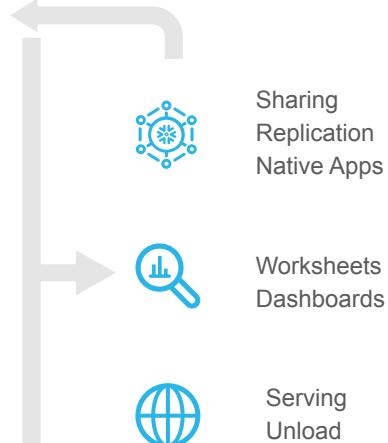
INGESTION



TRANSFORMATION



DELIVERY



OBSERVABILITY



Alerts, Notifications



Logs and events



Visual DAG viewer in the
Snowflake UI

CREATE TABLE AS Select (CTAS) & MERGE

Like a comfortable pair of jeans...

```
CREATE TABLE mytable_copy (b) AS SELECT * FROM mytable;
```

 Copy

```
DESC TABLE mytable_copy;
```

name	type	kind	null?	default	primary key	unique key	check
B	NUMBER(38,0)	COLUMN	Y	NULL	N	N	NULL

```
CREATE TABLE mytable_copy2 AS SELECT b+1 AS c FROM mytable_copy;
```

```
DESC TABLE mytable_copy2;
```

```
MERGE INTO t1 USING t2 ON t1.t1Key = t2.t2Key
    WHEN MATCHED AND t2.marked = 1 THEN DELETE
    WHEN MATCHED AND t2.isNewStatus = 1 THEN UPDATE SET val = t2.newVal, status =
    WHEN MATCHED THEN UPDATE SET val = t2.newVal
    WHEN NOT MATCHED THEN INSERT (val, status) VALUES (t2.newVal, t2.newStatus);
```

<https://docs.snowflake.com/en/sql-reference/sql/create-table>

<https://docs.snowflake.com/en/sql-reference/sql/merge>

Dynamic Tables

Description

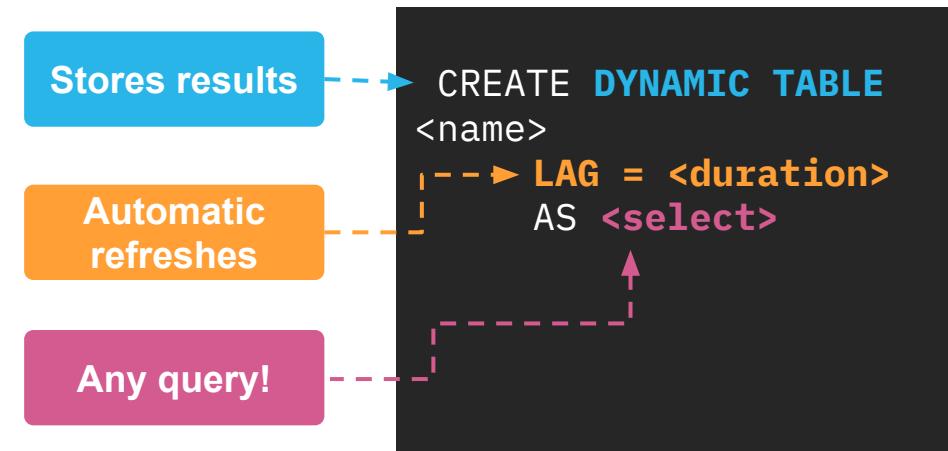
Declarative approach to transformations and simple data pipeline creation

Value

Automate incremental data refresh with low latency using easy-to-use declarative pipelines. Helps with streaming pipelines that require many small, incremental updates

Functionality

Join and aggregate across multiple source objects and incrementally update results as sources change



Some (Not All) Monitoring Features

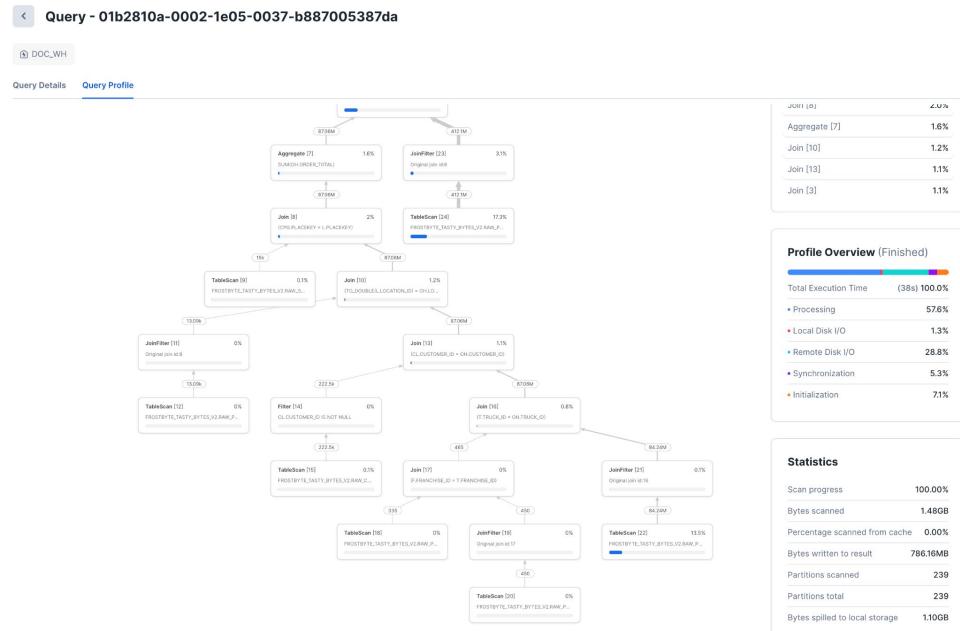


Snowflake Monitoring

Overall Platform

- Login activity
- Query activity
- Cost/Usage
- Tasks
- Copy / Data Loading

Individual Query Troubleshooting



INFORMATION SCHEMA

- Read Only system-defined views and table functions that provide extensive metadata and usage information about your account
- Delivered as a schema within each database
- **Zero latency**
- Results are limited:
 - Current state only
 - No guaranteed consistency with respect to concurrent DDL
 - Only objects for which the current role has been granted access privileges are returned
 - Limited to database called by INFORMATION_SCHEMA schema for database object centric views/table functions
- Cloud Services intensive

ACCOUNT USAGE

- Read Only views that provide extensive metadata and usage information about your account
- Delivered via the SNOWFLAKE database share
- **45-180 min latency**
- Retention: 1 year
- All concepts of INFORMATION_SCHEMA + STORAGE_USAGE views
- By default, only available to ACCOUNTADMIN



Usage Monitoring

Example: Identify credit consumption by warehouse for the past 7 days

```
SELECT
    START_TIME,
    WAREHOUSE_NAME,
    SUM(CREDITS_USED_COMPUTE) AS CREDITS_USED_COMPUTE_SUM
FROM ACCOUNT_USAGE.WAREHOUSE_METERING_HISTORY
WHERE START_TIME >= DATEADD(DAY, -7,
CURRENT_TIMESTAMP()) // Past 7 days
GROUP BY 1
ORDER BY 2 DESC;
```

Credits by Warehouse



Credits by Warehouse

xxxxxxxxxx_WH	2,244.88
xxxxxxxxxxx_WH	1,971.79
xxxxxx_WH	1,568.79
xxxxxx_WH	1,328.34
xxxxxxxxxxxxxx_WH	1,103.8
xxxxx_WH	1,059.84
xxxxxxxxxxxxxxx_WH	696.75
xxxxxxxxxxxxxx_WH	385.61
xxxxxxxxxxxxxx_WH	179.25
xxxxxx_WH	119.36
xxxxxxxxxxxxxxx_WH	112.5
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..._WH	110.86
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..._WH	48.02
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..._WH	41.22
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..._WH	40.68
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..._WH	37.49
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..._WH	32.13

Performance

Example: Identify scale **UP** opportunities - warehouses with high amount of spilling and could benefit from a larger warehouse size

```
SELECT QUERY_ID  
,USER_NAME  
,WAREHOUSE_NAME  
,WAREHOUSE_SIZE  
,BYTES_SCANNED  
,BYTES_SPILLED_TO_REMOTE_STORAGE  
,BYTES_SPILLED_TO_REMOTE_STORAGE / BYTES_SCANNED AS  
SPILLING_READ_RATIO  
  
FROM SNOWFLAKE.ACCOUNT_USAGE.QUERY_HISTORY  
  
WHERE BYTES_SPILLED_TO_REMOTE_STORAGE > BYTES_SCANNED  
* 5 -- Each byte read was spilled 5x on average  
  
ORDER BY SPILLING_READ_RATIO DESC;
```

Example: Identify scale **OUT** opportunities - warehouses with high amount of queuing and could benefit from a multi-cluster strategy

```
SELECT TO_DATE(START_TIME) as DATE  
,WAREHOUSE_NAME  
,SUM(AVG_RUNNING) AS SUM_RUNNING  
,SUM(AVG_QUEUED_LOAD) AS SUM_QUEUED  
FROM SNOWFLAKE.ACCOUNT_USAGE.WAREHOUSE_LOAD_HISTORY  
WHERE TO_DATE(START_TIME) >=  
DATEADD(month,-1,CURRENT_TIMESTAMP())  
GROUP BY 1,2  
HAVING SUM(AVG_QUEUED_LOAD) >0;
```



Monitoring / Optimization

The screenshot shows a web browser displaying the quickstarts.snowflake.com/?cat=resource-optimization page. The page has a blue header with the Snowflake logo and a search bar. Below the header, the title "GETTING STARTED WITH SNOWFLAKE" is displayed in large white text. A sub-header below it says "Follow along with our tutorials to get you up and running with the Snowflake Data Cloud." There are three buttons: "SNOWFLAKE QUICKSTARTS ON GITHUB", "VIRTUAL HANDS-ON LABS", and "FREE TRIAL". A black banner at the bottom of the blue section says "SNOWPARK DAY: Learn AI/ML Workflows in Snowflake. Register now for free" with a link icon. Below this, there are four cards arranged in two rows. The top row contains three cards: "Resource Optimization: Billing Metrics", "Resource Optimization: Performance", and "Resource Optimization: Setup & Configuration". The bottom row contains one card: "Resource Optimization: Usage Monitoring". Each card has a "START" button at the bottom. The top row cards were last updated on Jan 19, 2024. The bottom row card was last updated on Jan 19, 2024. The page also includes navigation links for "A-Z", "RECENT", and "DURATION", and dropdown menus for "Choose a language" and "resource-optimization".

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

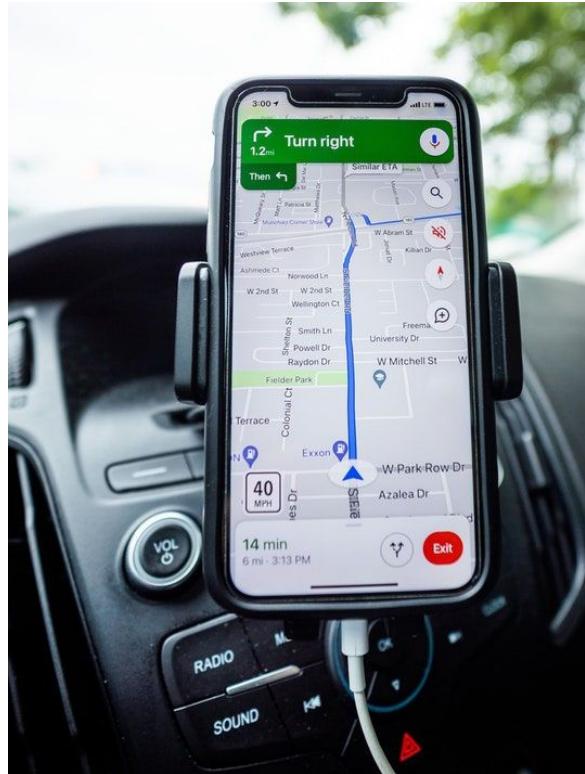


Dynamic Tables

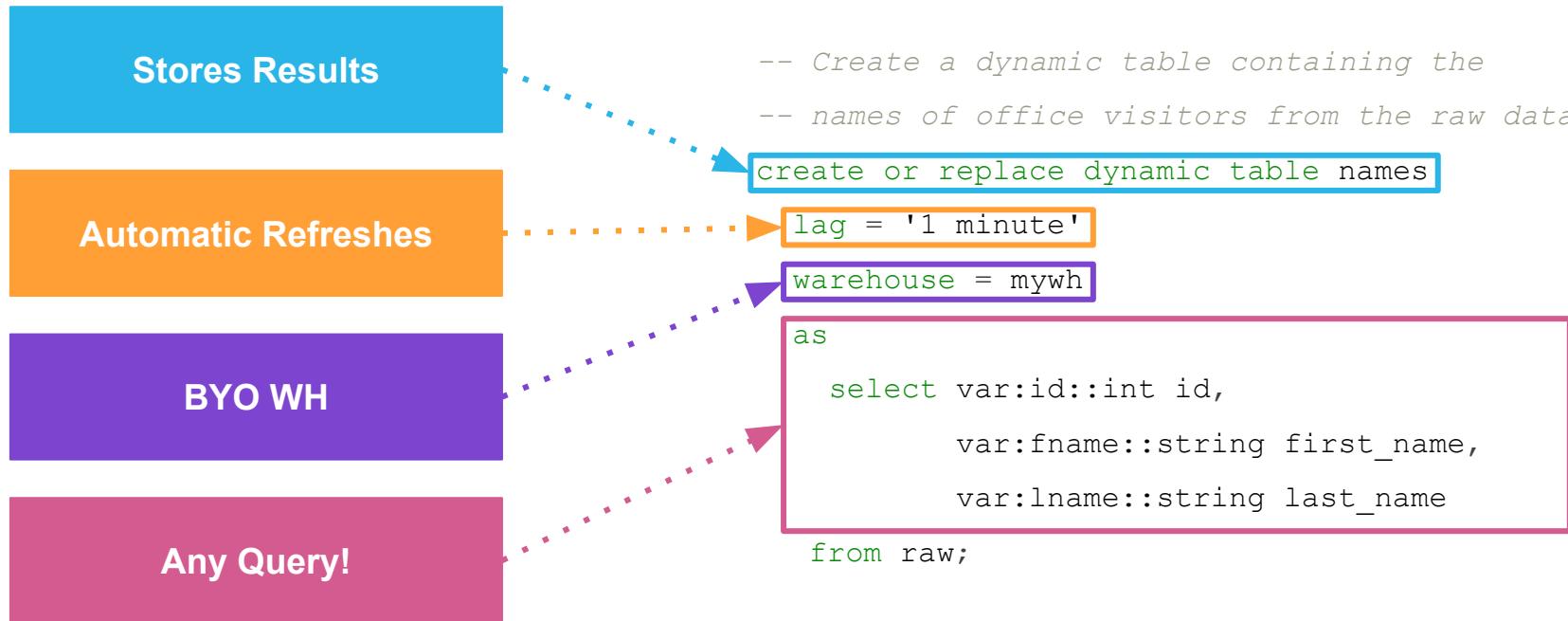


What are Dynamic Tables?

- New table type
 - Declarative
 - Describes end point
- Continuous automatic refresh
 - Better performance
 - Lower cost



How do Dynamic Tables work?



Benefits



Snowflake Managed

Leave the complex pipeline management to Snowflake



Value Add

Make data engineers more productive allowing them to spend more time on value add tasks



Scalability

Modern data stacks need to be scalable and cost effective

Your Turn

Creating / Using a Dynamic Table



- Open Labs → dynamic-tables
- Follow the instructions at
 - https://quickstarts.snowflake.com/guide/getting_started_with_dynamic_tables/
- Lab generally follows this flow
 - 0-setup.sql includes all the initial setup you'll need
 - Start the Quickstart at step 3

X Getting Started with Snowflake Dynamic Tables

- 1 Overview
- 2 Sample data setup
- 3 Build data pipeline using Dynamic Tables
- 4 Use case: Using Snowpark UDTF in Dynamic table

3. Build data pipeline using Dynamic Tables

With Dynamic Tables, customers provide a query and Snowflake automatically materializes the results of that query. That means, instead of creating a separate target table and writing code to transform source data and update the data in that table, you can define the target table as a Dynamic Table, specifying the query that performs the transformation and just forget about the scheduling and orchestration. The user specifies a minimum acceptable freshness in the result (target lag), and Snowflake automatically tries to meet that target, further enhancing the flexibility and control data engineers can have over their pipelines without the normally



Your Turn

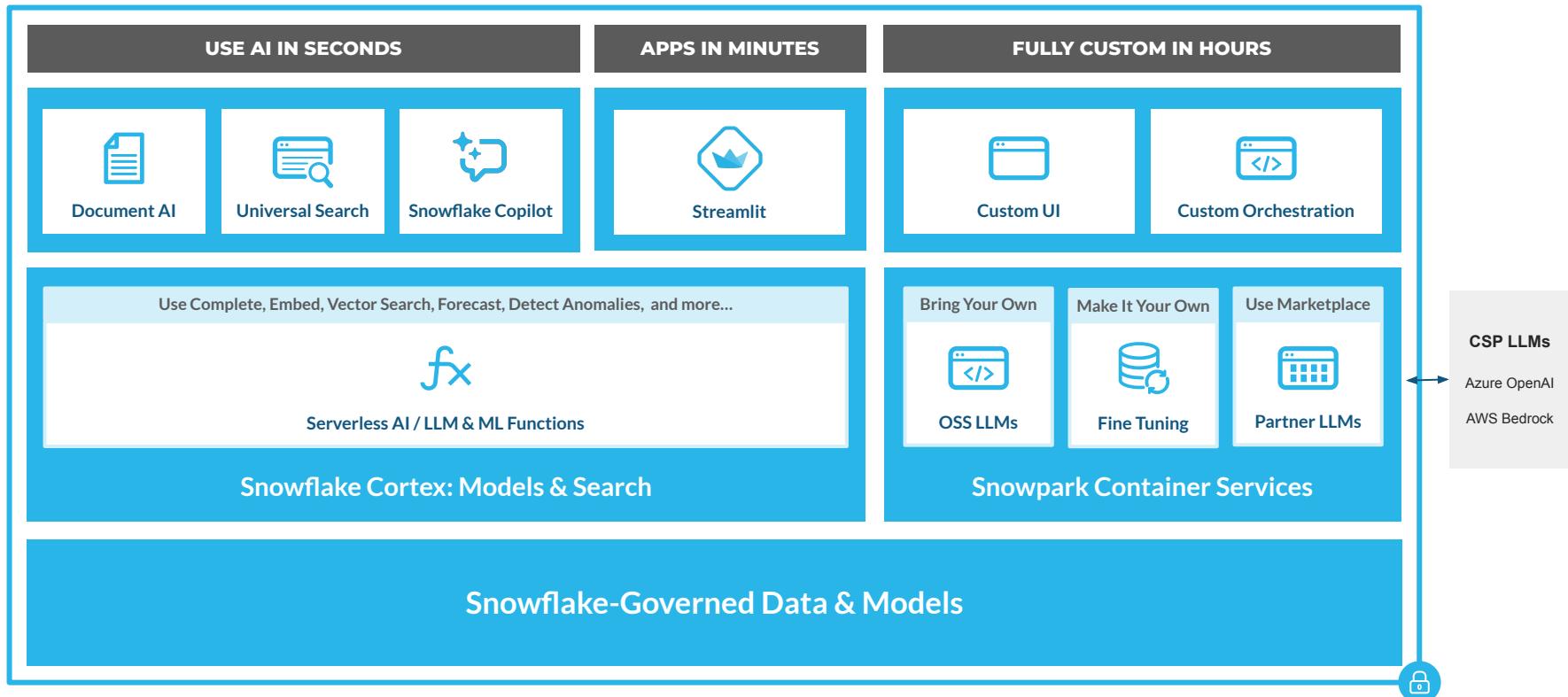
Creating / Using a Dynamic Table



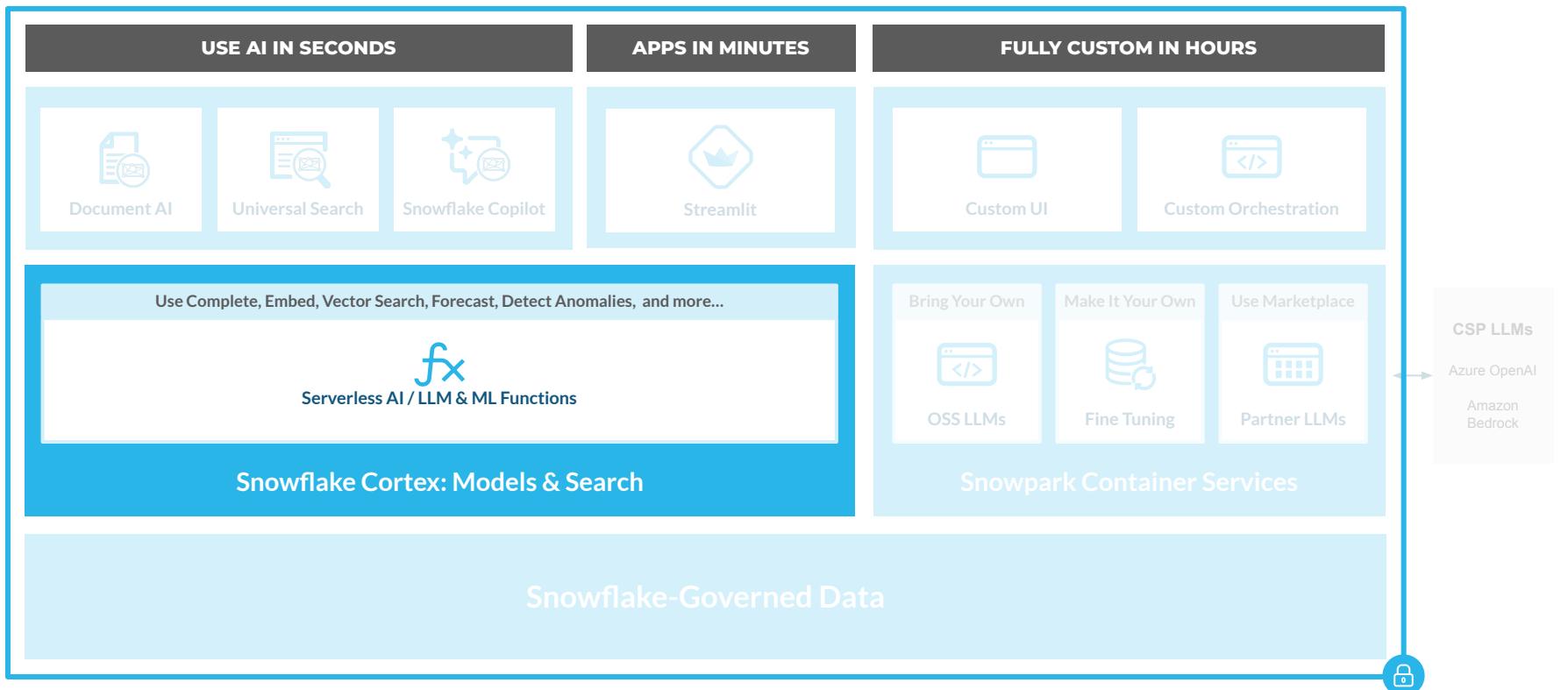
Advanced Analytics (ML Functions)



Snowflake for AI & ML: Platform Details



Snowflake for Gen AI & LLMs: Platform Details



Anomaly detection and forecasting introduction

Snowflake has released **built-in time-series forecasting and anomaly detection models** that can be invoked from SQL directly.

The models leverage machine learning and are optimized for **accuracy** and **speed**. They are built by Snowflake and support predictions where one variable changes over time (univariate) and can include predictions over multiple time series at once.



Easy to Use

- One-line SQL invocation
- No external tools needed
- No data egress
- No need to choose ML model
- No model hyperparameter tuning required



Accurate

Competitive in its accuracy vs. most popular out-of-the-box forecasting models like Auto ARIMA and Prophet*



Fast & Scalable

- Handles hundreds of thousands of rows in seconds
- Scales to process millions of time series simultaneously



Robust

- Automatically identifies and adjusts for missing data
- Automatically identifies multiple seasonality and holiday patterns

What are Snowflake Cortex ML-Based Functions?

What are they

SQL functions that perform predictive analytics using machine learning under the hood

Why use them

Generate predictions and insights on your structured data – with the power of ML but without its complexity

How to use them

Use SQL function in Snowsight or wherever you access your Snowflake data

ML-Based Functions

- FORECAST() GA
- ANOMALY_DETECTION() GA
- TOP_INSIGHTS() PuPr
- CLASSIFICATION() PrPr



Private Preview



Public Preview



General Availability



Forecasting user experience

To train a model and reuse it for multiple forecasts:

- 1 Input your target variable and optionally, exogenous features, a partition ID (for multi-series forecasts).
- 2 Create (train) a model by call a SNOWFLAKE.ML.FORECAST.
- 3 Prediction (forecasting) is done by calling forecast() on your trained model and specifying the number of predictions you want to generate. Optionally, you can customize the size of your prediction interval.

```
// Train a forecast model. Input table "input" has  
// input.date and input.sales columns  
CREATE SNOWFLAKE.ML.FORECAST my_forecast(  
    input_data => SYSTEM$REFERENCE('TABLE', 'input'),  
    timestamp_colname => 'date',  
    target_colname => 'sales'  
);  
  
// Forecast 10 future data points  
CALL my_forecast!forecast(10,  
    {'prediction_interval': 0.8});
```

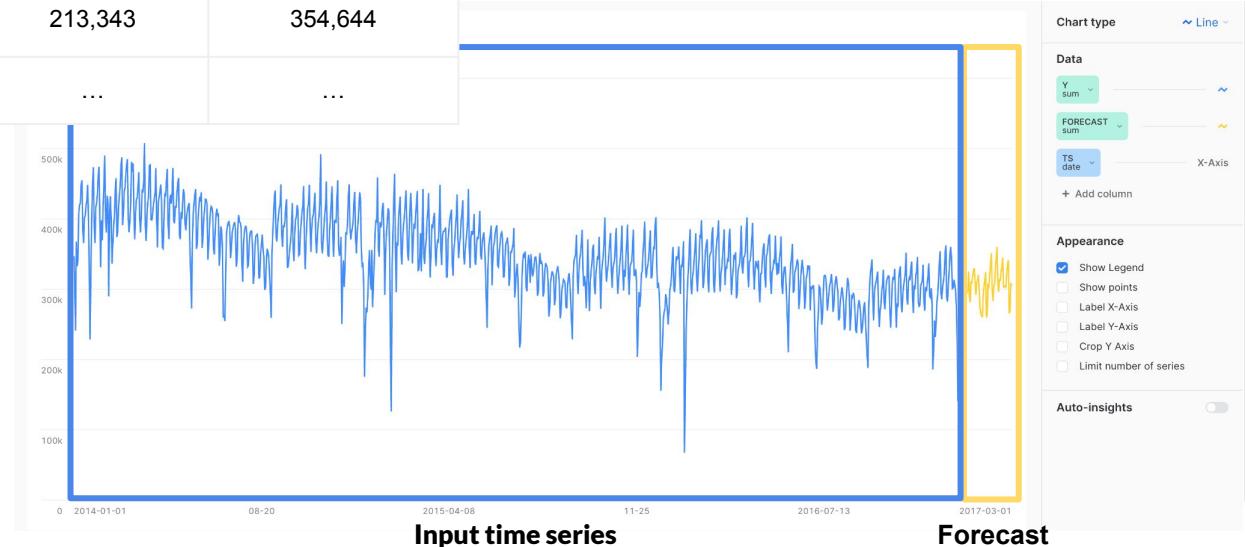


Example: Time series forecast output table

When you call your forecast function for a single time series, you'll get output like the below, with a forecast and an upper and lower bound around the likely range for that forecast.

TS	FORECAST	LOWER_BOUND	UPPER_BOUND
2022-05-10 00:00:00	257,123	186,504	327,741
2022-05-11 00:00:00	283,994	213,343	354,644
...

Note: By default, the lower and upper bounds represent the range in which 95% of observations are expected to fall. You can adjust this using the `prediction_interval` parameter.



Forecasting user experience - advanced

This function provides options for building on a simple single-series forecast:

- 1 You can use a column in your data table to identify series that should forecasted separately. (e.g., `storeA_jacket` vs. `storeB_jacket`)

Your output will look like this:

SERIES	TS	FORECAST	LOWER_BOUND	UPPER_BOUND
[1, jacket]	2020-01-06 00:00:00.000	7	3	11
[1, jacket]	2020-01-07 00:00:00.000	8	2	15
[2, umbrella]	2020-01-06 00:00:00.000	3	1	5

```
// Train a forecast model for multiple series.  
CREATE SNOWFLAKE.ML.FORECAST my_forecast(  
    input_data => SYSTEM$REFERENCE('TABLE', 'input'),  
    1 series_colname => 'store_item',  
    timestamp_colname => 'date',  
    target_colname => 'sales'  
);  
  
call my_forecast!forecast(2);
```



Forecasting user experience - advanced

This function provides options for building on a simple single-series forecast:

- 2 You can feed exogenous variables into your model.

If you do this, be sure to provide those same features when you're calling your model to generate predictions!

Note: When you call your model with exogenous variables, the model infers how many predictions to make from the number of observations in your **forecast** dataset.

```
// Create a view with your exogenous variables.  
create or replace view v2 as select  
    date, sales, precipitation, holiday 2  
from sales_data;  
  
// Train your model.  
create snowflake.ml.forecast inst2(  
    input_data => SYSTEM$REFERENCE('VIEW', 'v2'),  
    timestamp_colname => 'date',  
    target_colname => 'sales');  
  
// Create a forecast dataset with your exogenous features.  
create or replace view v2_forecast as select  
    date, precipitation, holiday  
from future_features;  
  
// Create your forecast.  
call inst2!forecast(input_data => SYSTEM$REFERENCE('VIEW',  
    'v2_forecast'), timestamp_colname =>'date');
```



Anomaly detection user experience

To detect outliers on a time series, use Snowflake's anomaly detection functions:

- 1 Input your target variable and optionally, exogenous features, a prediction interval size, a set of labels for already-identified outliers, and a partition ID (for multi-series forecasts).
- 2 Create your anomaly detector using the `anomaly detection` function. Then generate outlier labels by calling `detect_anomalies`.
- 3 The output table returns prediction intervals, an estimated value for each timestamp, a label `is_anomaly`, and the distance between the observed value and the estimated value.

```
// Create a view and anomaly detection object
create or replace view training_view as select date, sales
    from historical_sales_data
    where store_id=1 and item='jacket';

create or replace SNOWFLAKE.ML.ANOMALY_DETECTION model1(
    input_data => SYSTEM$REFERENCE('VIEW', 'training_view'),
    timestamp_colname => 'date',
    target_colname => 'sales',
    label_colname => '');

// Call your anomaly detector on new data
create or replace view new_data as select
    date, sales
    from new_data
    where store_id=1 and item='jacket';

call model1!detect_anomalies(input_data =>
    SYSTEM$REFERENCE('VIEW', 'new_data'),
    timestamp_colname =>'date',
    target_colname => 'sales');
```

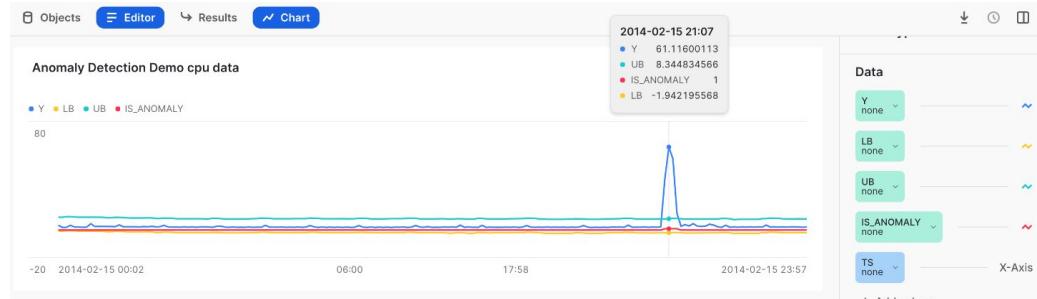


Single time series output example

- 1 In this example, one anomaly is detected (`is_anomaly` = 1).

TS	Y	FORECAST	LOWER_BOUND	UPPER_BOUND	IS_ANOMALY
2020-01-06 0:00:00	7	9.12	6.19	12.06	0
2020-01-07 0:00:00	20	11.72	8.79	14.66	1

- 2 You can also visualize your data in our web UI, Snowsight, to see where your outliers fall.



Anomaly detection user experience - advanced

This function provides options for building on a simple single-series detection:

- 1 You can use a column in your data table to identify series that should be monitored separately for anomalies. (e.g., `storeA_jacket` vs. `storeB_jacket`)
- 2 You can provide the model with hand-labeled anomalies in your training data – to ensure future anomalies are identified as accurately as possible.
- 3 You can determine the size of your prediction interval. A smaller prediction interval corresponds to a stricter definition of “normal” – and increases the number of anomalies you’ll identify.

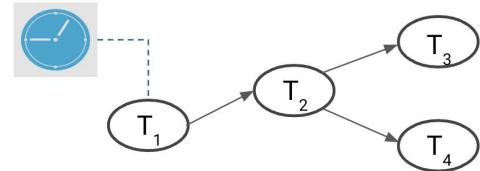
```
// Use the `series_colname` parameter to partition your data  
// into series that should be monitored separately for anomalies  
// Use `label_colname` parameter to train your model with  
manually-identified anomalies  
CREATE OR REPLACE SNOWFLAKE.ML.ANOMALY_DETECTION model(  
    INPUT_DATA => SYSTEM$REFERENCE('VIEW', 'training_view'),  
    1 SERIES_COLNAME => 'store_item',  
    TIMESTAMP_COLNAME => 'date',  
    TARGET_COLNAME => 'sales',  
    2 LABEL_COLNAME => 'label');  
  
// Use the `config_object` parameter to define the size of your  
prediction interval  
CALL model_trained_with_labeled_data!DETECT_ANOMALIES(  
    INPUT_DATA => SYSTEM$REFERENCE('VIEW', 'new_data'),  
    TIMESTAMP_COLNAME => 'date',  
    TARGET_COLNAME => 'sales',  
    3 CONFIG_OBJECT => {'prediction_interval':0.99});
```



Automate your detections with tasks

```
// Use Snowflake Tasks to train your model and detect anomalies regularly

CREATE OR REPLACE TASK ad_model_training_task
    WAREHOUSE = ML_ANOMALY_WH
    SCHEDULE = '24 HOUR'
    AS
        EXECUTE IMMEDIATE
    $$
    BEGIN
        CREATE OR REPLACE SNOWFLAKE.ML.ANOMALY_DETECTION
            login_detector_hourly(
                INPUT_DATA => SYSTEM$REFERENCE('VIEW', 'v1_train'),
                TIMESTAMP_COLNAME => 'date',
                TARGET_COLNAME => 'login_count',
                LABEL_COLNAME => '');
    END;
$$;
```

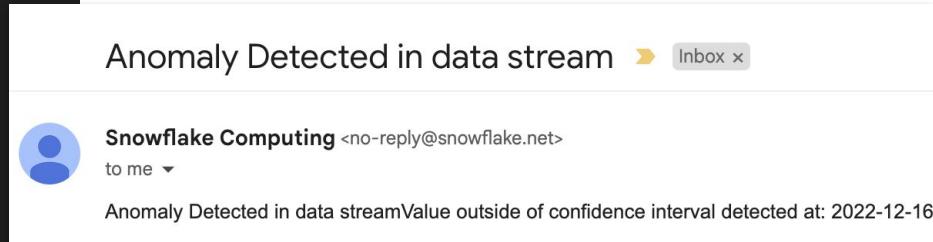


Tasks can also be chained!



Receive email alerts for detected anomalies

```
// Use Snowflake Alerts to detect and alert on near real time  
anomalies  
  
CREATE OR REPLACE ALERT user_login_alert  
    WAREHOUSE = ML_ANOMALY_WH  
    SCHEDULE = '1 HOUR'  
    IF (EXISTS (CALL extract_anomalies()))  
    THEN  
        CALL SYSTEM$SEND_EMAIL(  
            'login_alert',  
            'username@snowflake.com',  
            'Anomalous Login Data Detected',  
            concat(  
                'Value outside of prediction interval  
                detected in the most recent run at ',  
                current_timestamp(1)  
            )  
        );
```



Your Turn

Detecting Anomalies & Forecasting Sales



- Open `labs/ml-functions/anomaly_forecast.sql`
- Find anomalies in total sales using the `ORDER_HEADER` table
- Create a forecast for the next 21 days
- Visualize the forecast

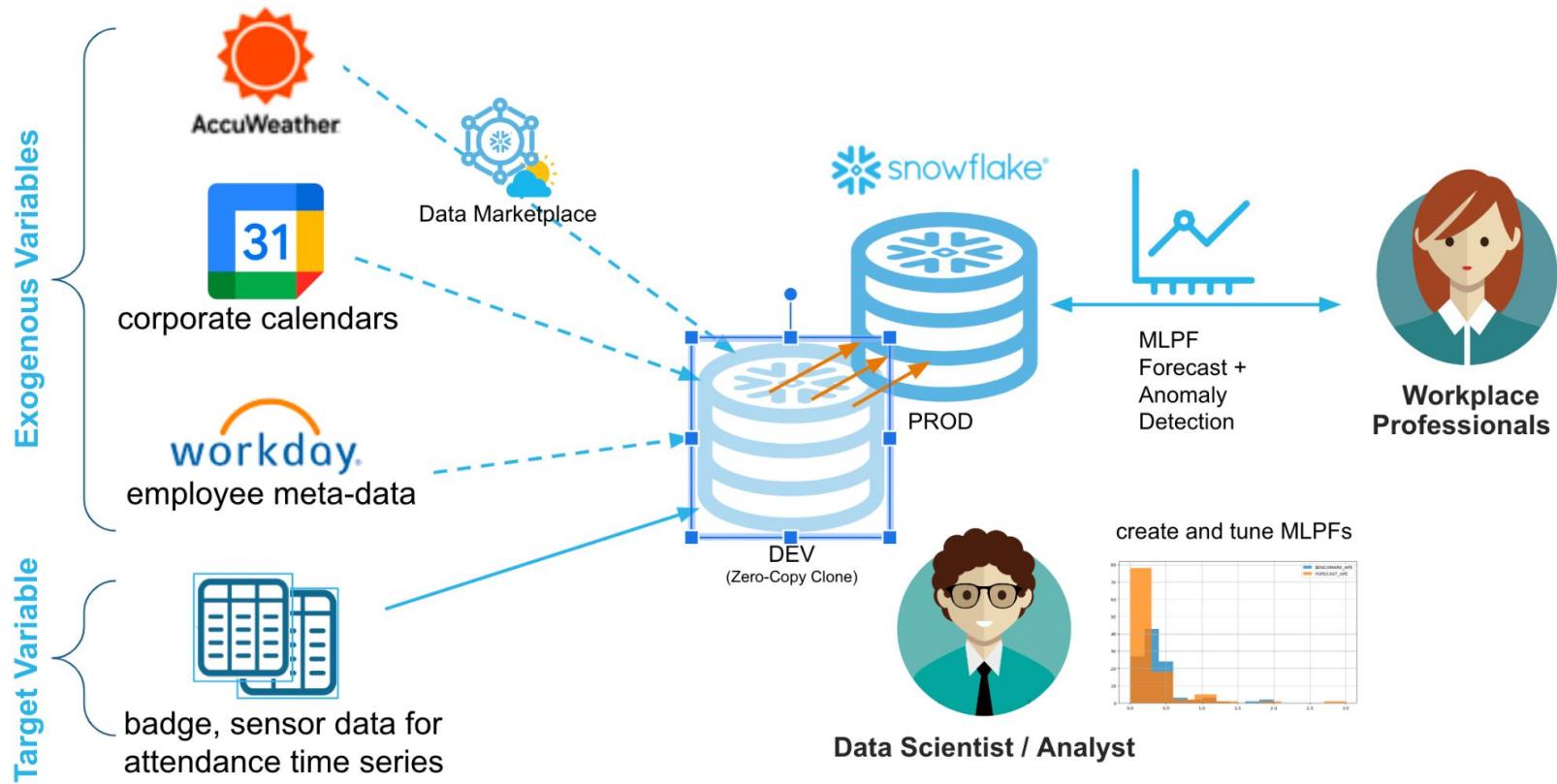


Your Turn

Forecasting & Anomalies



Using Cortex to Forecast Workplace Attendance



<https://medium.com/snowflake/super-charging-sql-analysts-with-coffee-and-ml-e28bdac94f6e>

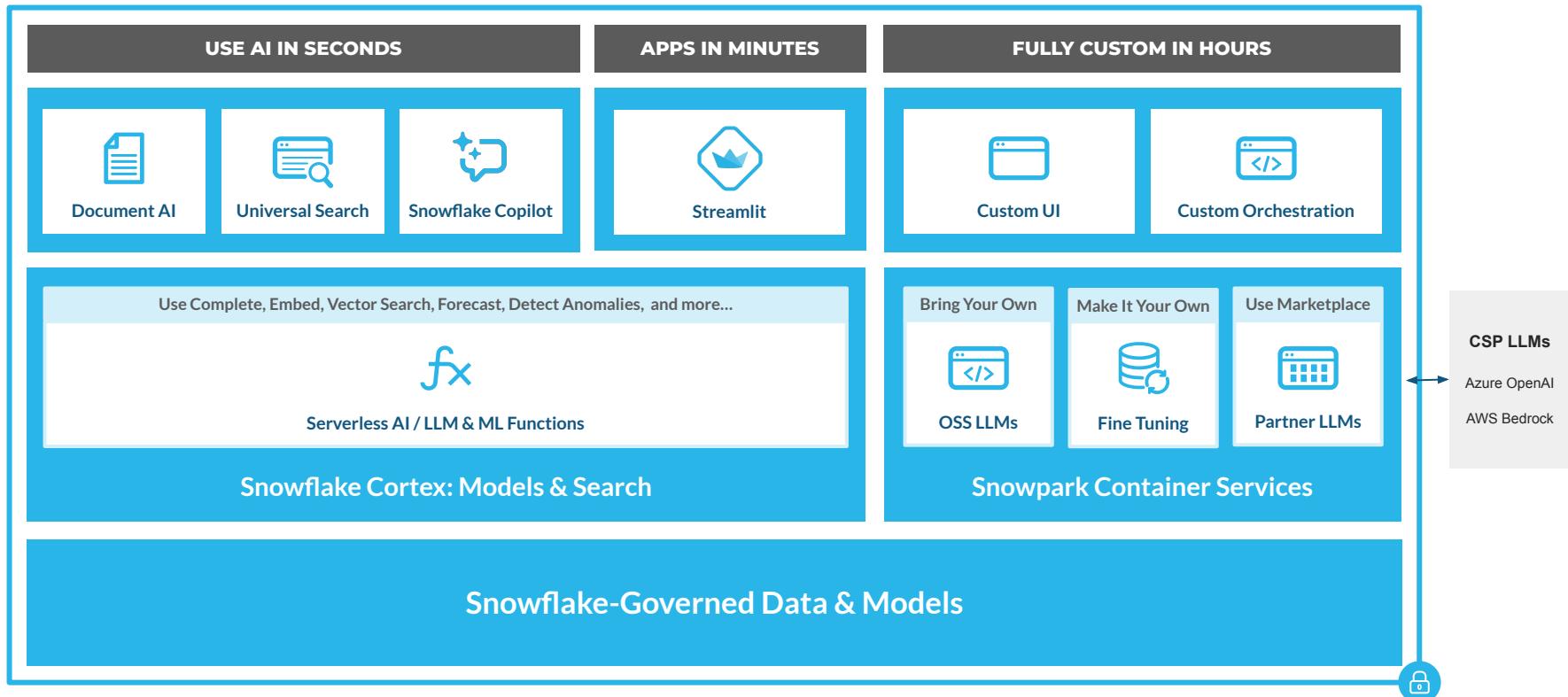


AI / ML

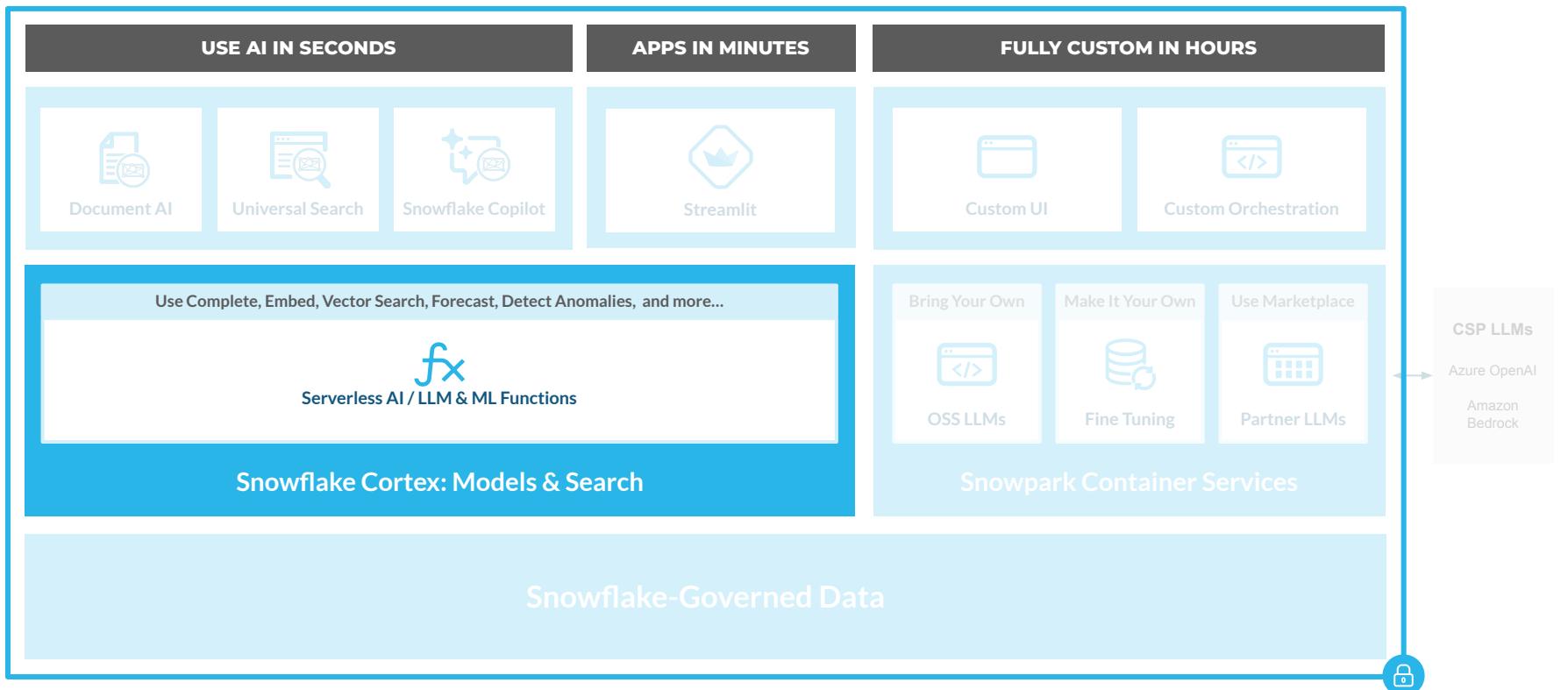


© 2024 Snowflake Inc. All Rights Reserved

Snowflake for AI & ML: Platform Details



Snowflake for Gen AI & LLMs: Platform Details



Snowflake Cortex: Specialized Functions

What are they

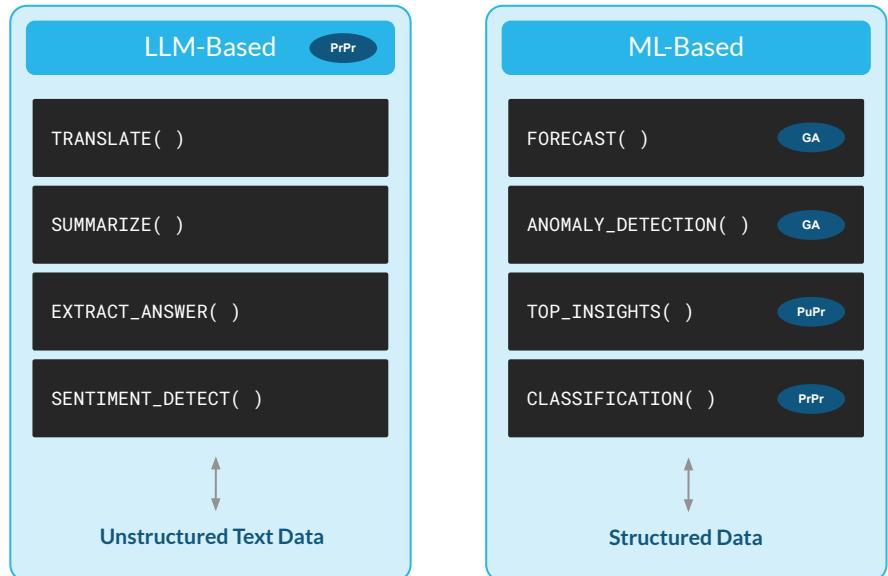
Serverless SQL / Python functions that are well suited for unstructured text analysis and predictive analytics

Why use them

Receive specific outputs from unstructured text data in a cost-effective way using smaller, task-tuned language models, or generate predictions and insights on your structured data

How to use them

Use SQL / Python function in Snowsight or expose as a custom app via Streamlit



PR Private Preview

PU Public Preview

GA General Availability



Snowflake Cortex: Generalized Functions

What Is It

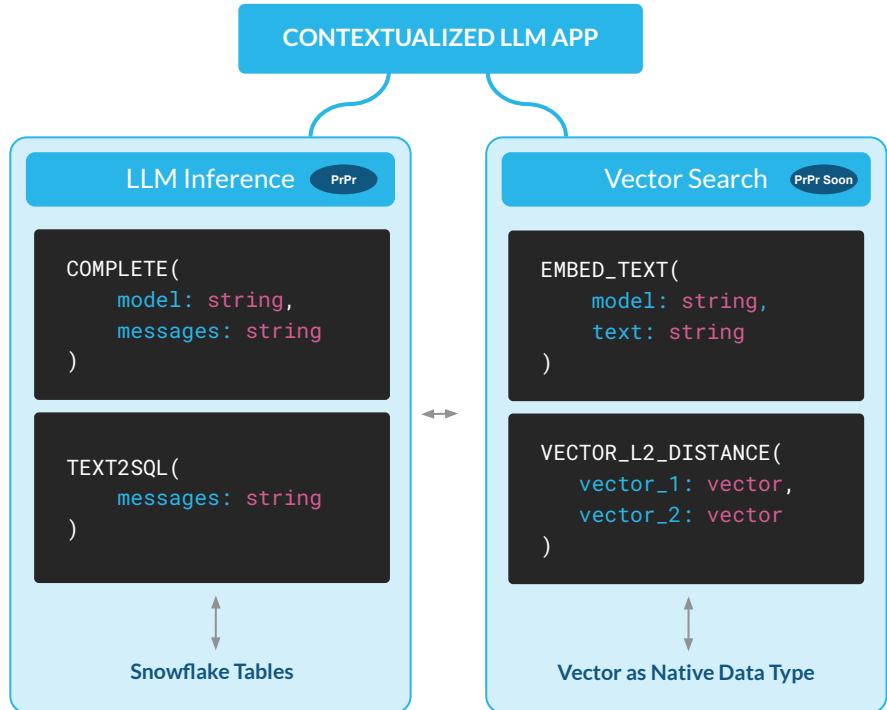
Serverless SQL / Python functions that run inference on conversational LLMs and execute vector search functionality

Why Use It

Quickly build contextually enriched applications using RAG by combining cutting-edge LLMs such as text2SQL and Llama 2 with vector embedding and similarity functions

How To Use It

Use SQL / Python function in Snowsight or expose as a custom app via Streamlit

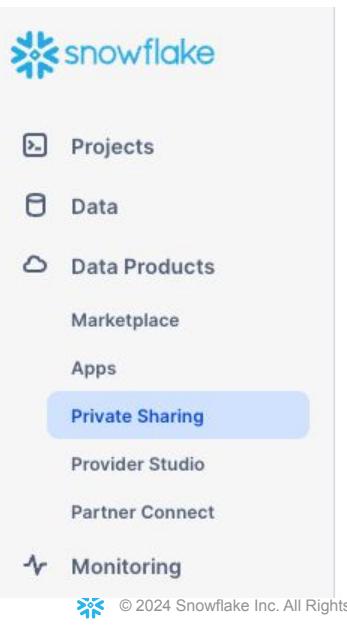


Your Turn

Using Snowflake Cortex



- Access Sample Call Transcripts from a Private Share



The image shows the left sidebar of the Snowflake web interface. It features the Snowflake logo at the top, followed by a vertical list of navigation items: Projects, Data, Data Products, Marketplace, Apps, Private Sharing (which is highlighted with a blue background), Provider Studio, Partner Connect, and Monitoring. At the bottom, there is a copyright notice: © 2024 Snowflake Inc. All Rights Reserved.

Shared With You Shared By Your Account

Search

Privately Shared Listings

Datatune Call Transcripts
SFSENORTHAMERICA.DEMO460

Sample call transcripts that need analysis to provide the best experience for our customers.

Published Just now

Direct Access

SFSENORTHAMERICA.DEMO460 - Datatune Call Transcripts

Shared Privately with You

Ready to Query

No Storage Cost

Datatune Call Transcripts
Sample call transcripts that need analysis to provide the best experience for our customers.

Options

Database name

Datatune_Call_Transcripts

A new database will be created. Consumes no storage.

Which roles, in addition to ACCOUNTADMIN, can access this database?

PUBLIC

Get

Your Turn

Using Snowflake Cortex



Task 1:

- Create a new table called “AUGMENTED_TRANSCRIPTS” that adds the following columns to the CALL_TRANSCRIPTS table:
 - **ENGLISH_TRANSCRIPT** that contains an English translation of the transcript
 - **CALL_SENTIMENT** that contains a sentiment score for the call
 - **CALL_SUMMARY** that summarizes the call



Your Turn

Creating / Using a Dynamic Table



Your Turn

Using Snowflake Cortex



Task 2:

- Write queries that do the following:
 - Find the products mentioned in the TRANSCRIPTS table
- Use a Large Language Model (LLM) to answer a question about ski gear
 - *Do you get the same answer if you use different models?*



Your Turn



15:00



Your Turn

Creating / Using a Dynamic Table



Snowflake Cortex

What Is It

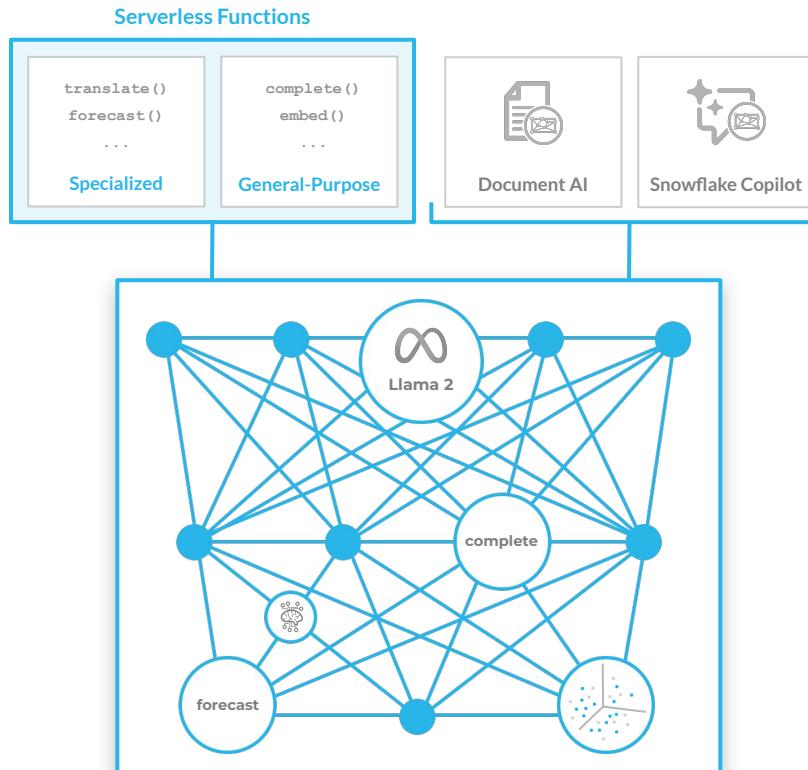
An intelligent, fully managed service that hosts and serves industry-leading AI and ML models, LLMs and vector functions

Why Use It

Quickly and securely analyze your data and build AI applications contextualized with your enterprise data

How To Use It

Access the power of Snowflake Cortex via serverless SQL / Python functions or as part of LLM-powered experiences such as Document AI and Snowflake Copilot



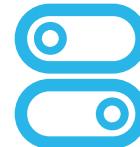
Snowflake Cortex

Puts LLMs and AI models in the hand of every user to get value securely from enterprise data



Easy To Use

Use AI in everyday analytics or build LLM applications without infrastructure management



Flexible

Access industry-leading AI models, LLMs and vector search functionality via SQL / Python functions



Cost Effective

Compute optimized for inference and search to run where your data is secure and governed



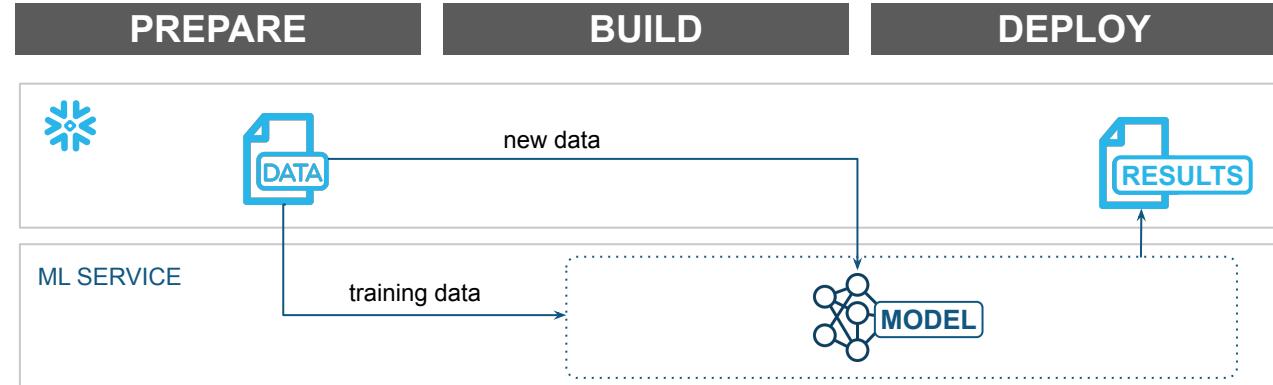
External Access / Integrations



UDFs vs External Functions

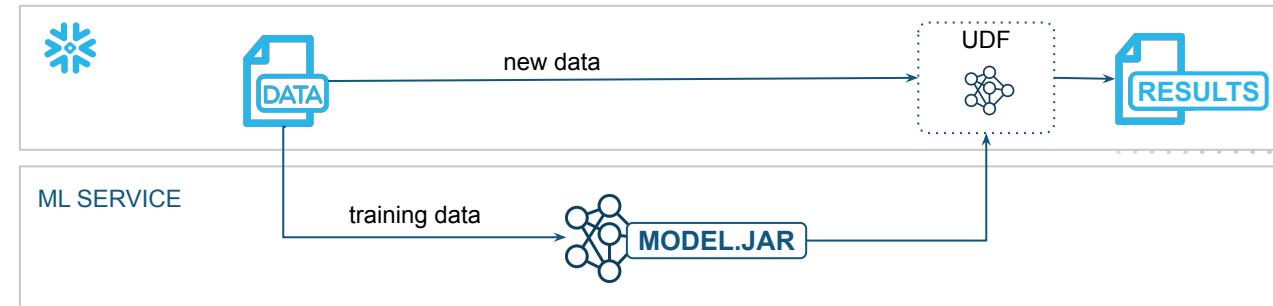
EXTERNAL FUNC

Data continuously travels to externally hosted model



UDFs

Model packaged (e.g. as a .jar file) and runs where the data lives



External Functions

Bind SQL Functions to Implementations

Outside Snowflake

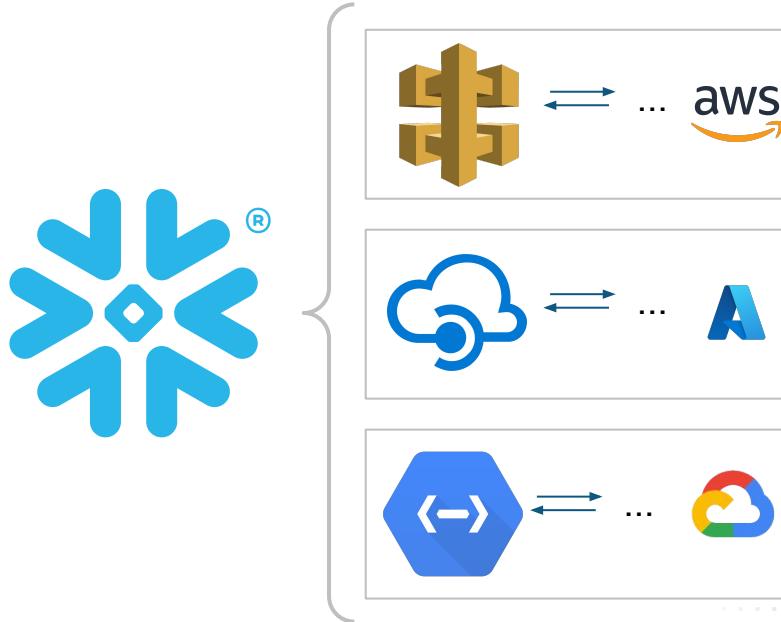
- V1: Batched scalar functions
- Mediated via API Gateway
- Usable wherever functions are used

Example Scenarios:

- Custom Lambda
- ML Scoring
- Geocoding

ACCOUNTADMIN in Control of Security

- Must explicitly whitelist endpoints



External Network Access

<https://docs.snowflake.com/en/developer-guide/external-network-access/external-network-access-examples>

External network access overview

FEATURE — GENERALLY AVAILABLE

External access is not supported in the Gov region.

You can create secure access to specific network locations external to Snowflake, then use that access from within the handler code for user-defined functions (UDFs) and stored procedures. You can enable this access through an external access integration.

With an external access integration, you can:

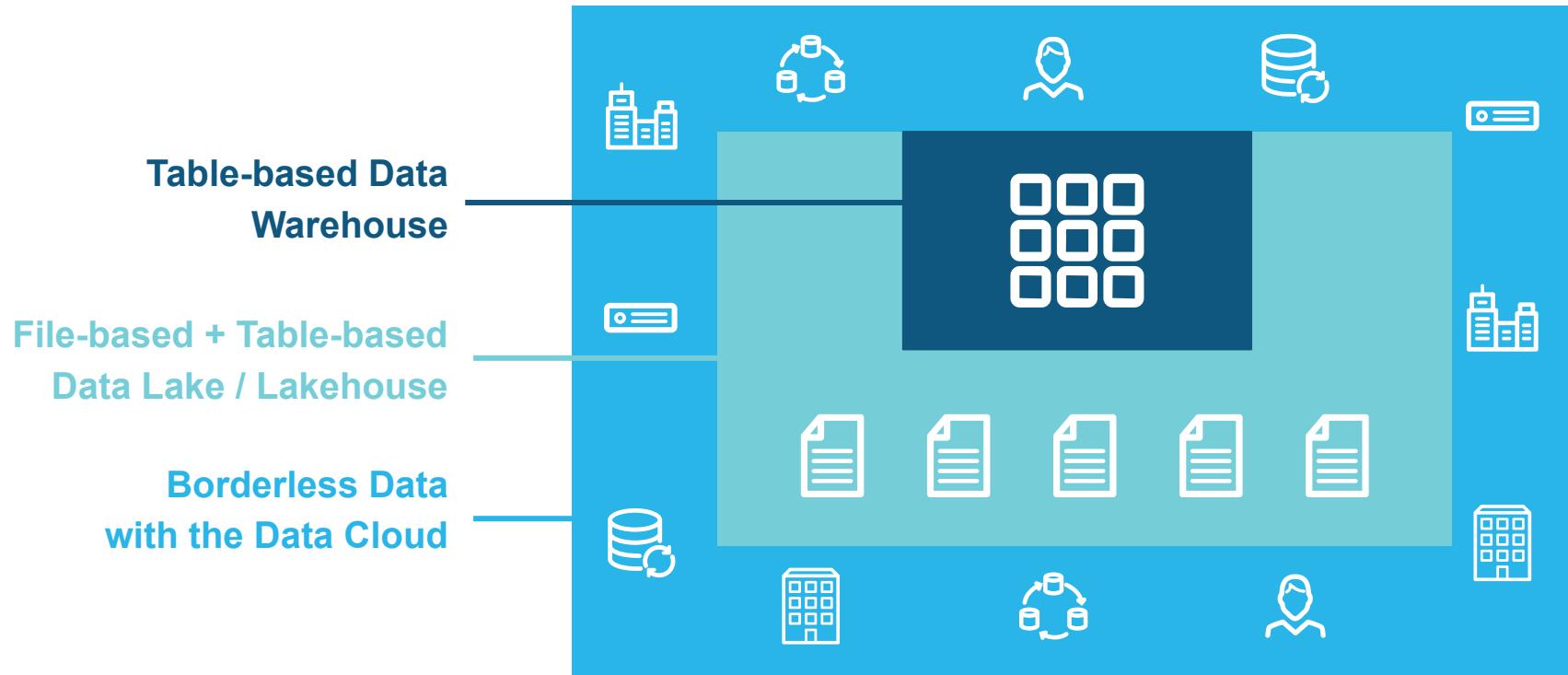
- Write UDF and procedure handlers that access external locations.
- Allow or block access to locations on a network external to Snowflake.
- Use secrets that represent stored credentials, rather than using literal values, within handler code to authenticate with external network locations.
- Specify which secrets are allowed for use with external network locations.



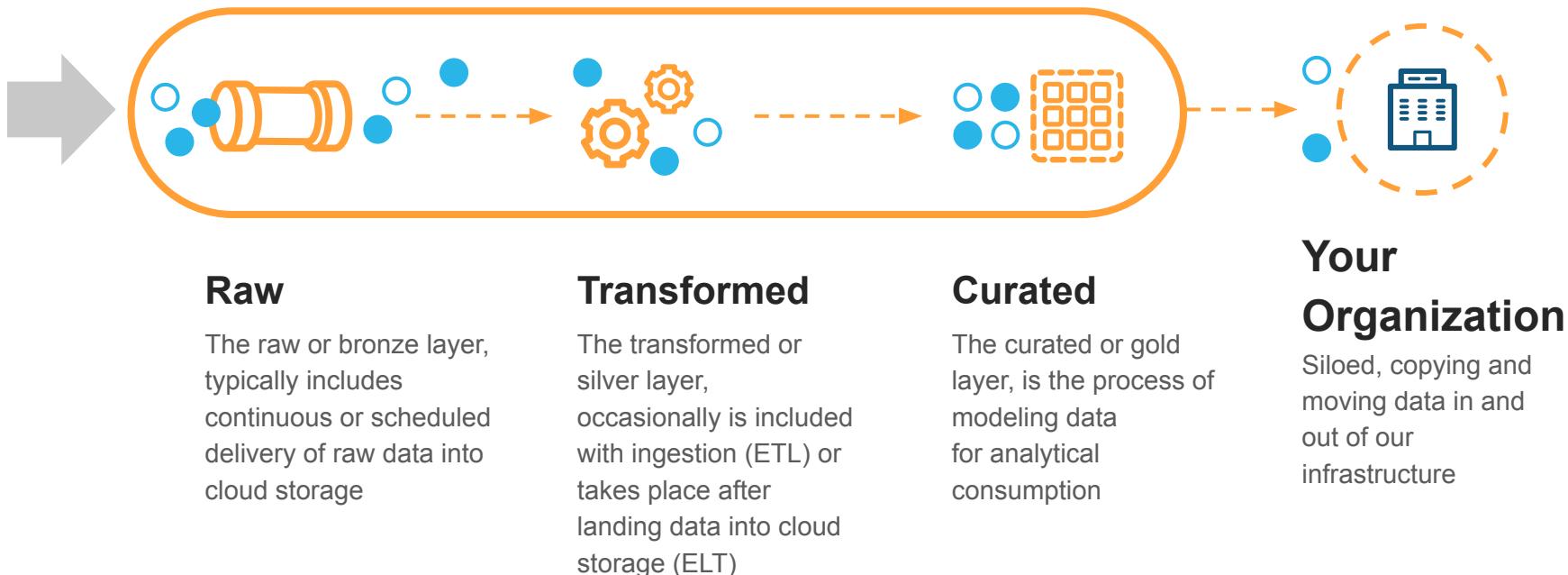
Data Lake



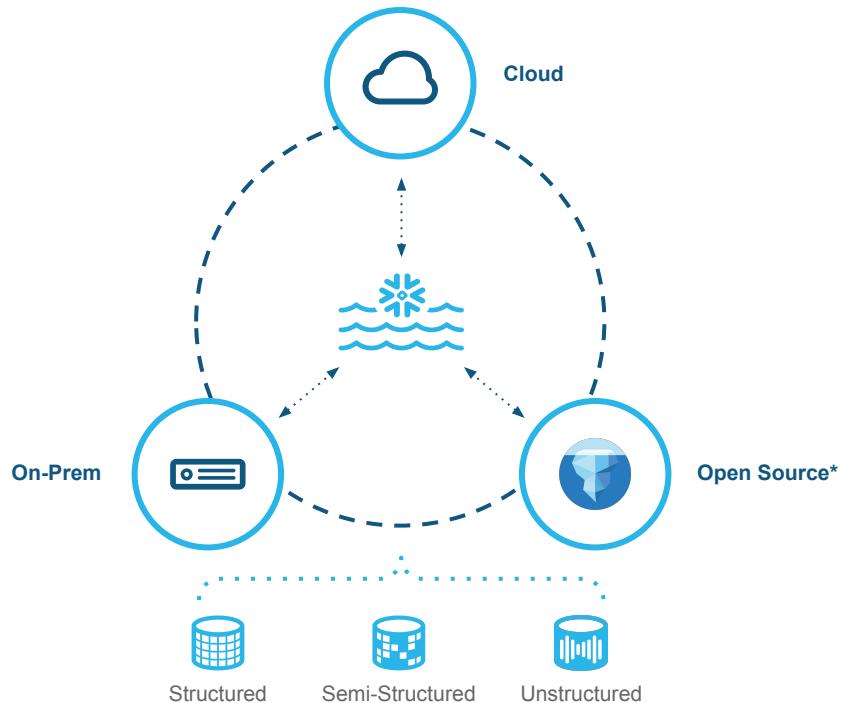
Evolution of Data Management



Data Pipelines Today

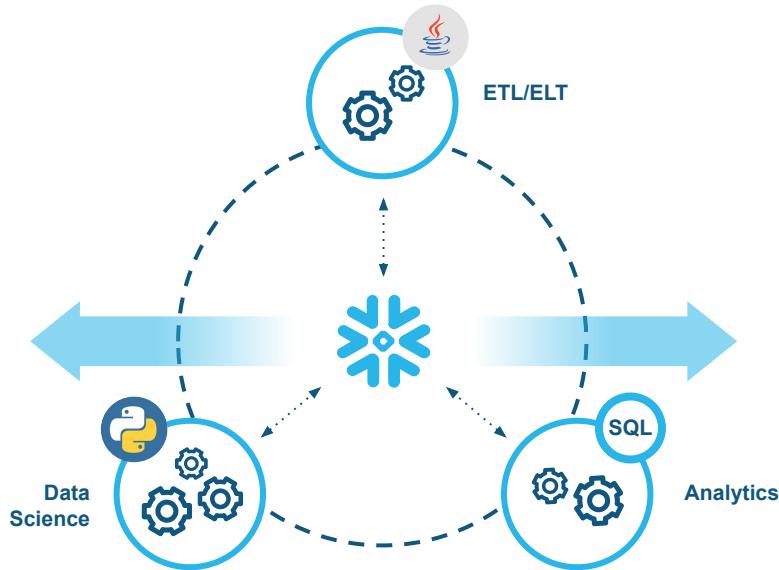


Seamlessly Bring Data Together in One Place



- > **Unsiloed Access to Your Data**
Bring together unstructured, semi-structured, and structured data and scale on demand to near-infinite data volumes.
- > **Easily Manage Data at Scale**
Automate micro-partitioning, clustering, compression, time travel, and encryption to ensure fast and efficient at any scale.
- > **Flexibility & interoperability**
Adapt to changing architectural patterns, including data mesh, and seamlessly connect with data on-premises or in open Iceberg format.

Fast Processing Engine with No Operational Overhead



> Scalable Compute

Support a virtually unlimited number of concurrent users and queries with near-unlimited, multi-cluster compute.

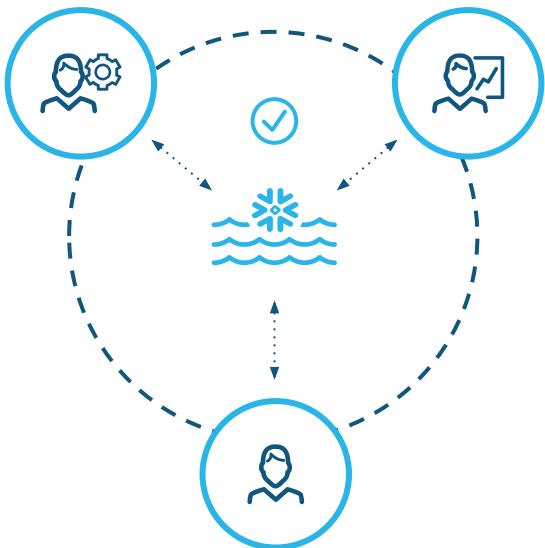
> Reliable Pipelines

Run pipelines with Snowflake's elastic processing engine for reliable performance, cost savings, and near-zero maintenance

> Language of Choice

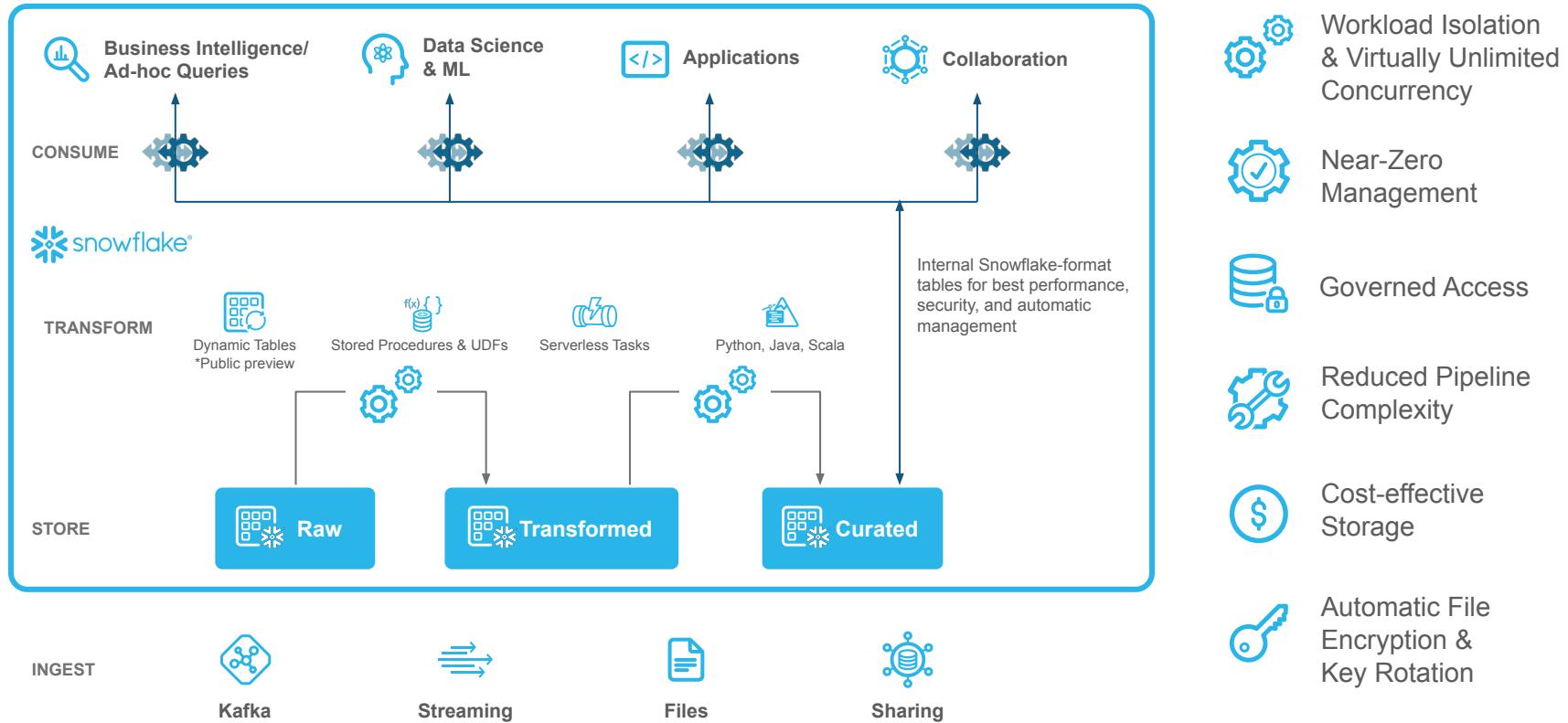
Streamline pipeline development using SQL or your language of choice with Snowpark—no additional clusters, services, or copies of your data to manage.

Easily Govern All Data and Enable Secure Collaboration

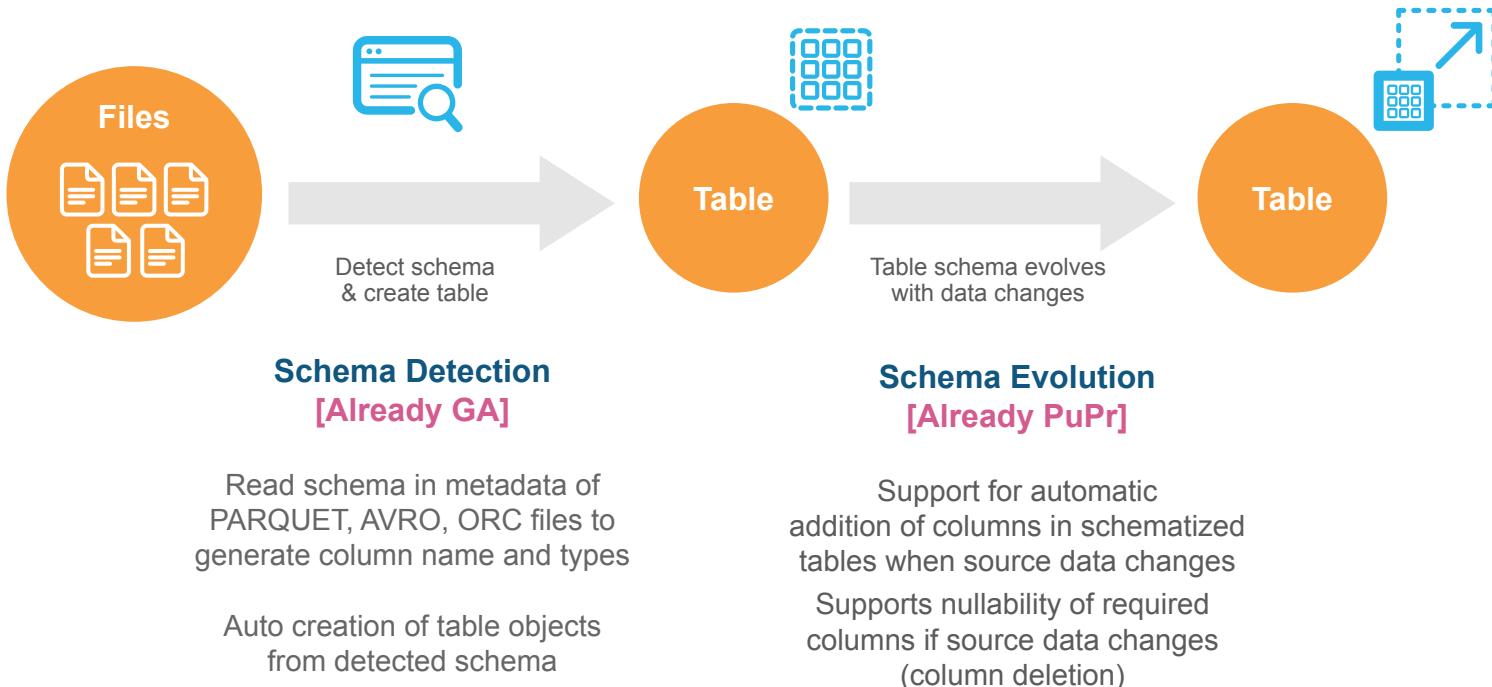


- > Scale Governance Efforts**
With flexible role-based policies, enforce row and column-level security dynamically, eliminating the need to manage multiple versions of the same data.
- > Confidently Protect Sensitive Data**
Track sensitive data for compliance with Object Tagging and Access History. Protect sensitive data with Dynamic Data Masking.
- > Enrich Your Data Lake**
Enable collaboration among internal and external stakeholders, and even enrich your data lake, with live, secure data sharing.

End-to-End Platform for Data Lake



SCHEMATIZATION



Iceberg Tables

New table type that brings the choice of Apache Iceberg to the Snowflake platform

Description

A new table type that stores data externally in an open format.

Value

Support data architectures requiring interoperability while retaining the easy management and great performance of Snowflake

Functionality

Table data is stored in customer-provided object storage in Apache Iceberg and Parquet format with options for catalog implementations.

ICEBERG TABLES

One object, multiple catalog options



Snowflake-managed

Iceberg Table with **Snowflake-managed** catalog.

SDK for Apache Spark™ access



Externally managed

Iceberg Table with **externally managed** catalog.

AWS Glue, Object Store

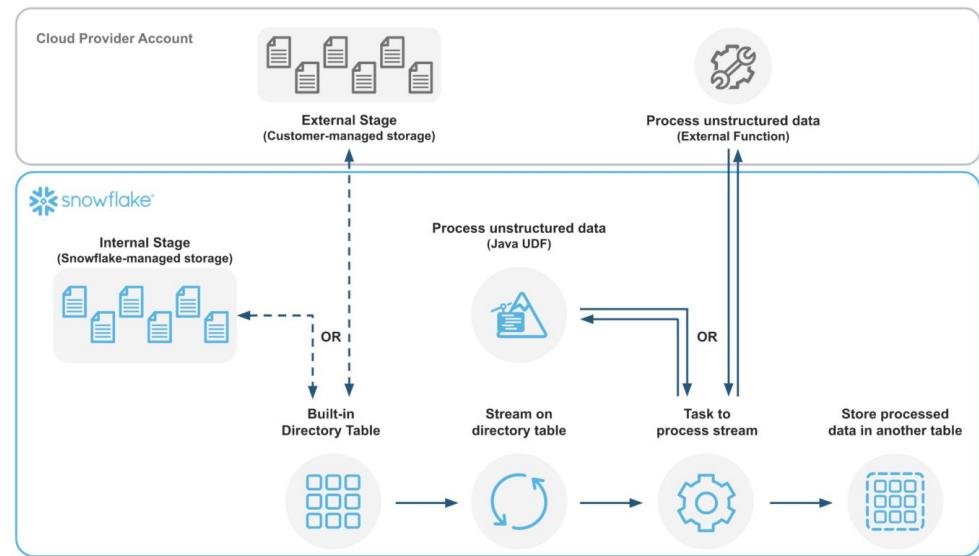
Your Turn

Snowflake for Data Lakes



- https://quickstarts.snowflake.com/guide/vhol_data_lake/index.html
- External Table over S3 Bucket
- Unstructured Processing

The screenshot shows the landing page for the "Snowflake for Data Lake" quickstart. It includes the title, duration (30 min), last updated date (Jan 19, 2024), and a prominent "START" button.



Advanced Data Engineering & Machine Learning



Your Turn

Intro to Machine Learning with Snowpark ML



- https://quickstarts.snowflake.com/guide/intro_to_machine_learning_with_snowpark_ml_for_python/index.html

What you will learn

- How to perform feature engineering and train a model in Snowflake with [Snowpark ML Modeling API](#)
- How to manage models and execute batch inference in Snowflake via [Snowpark Model Registry](#) for model inference



Snowpark Python



Snowpark Overview

What is Snowpark?

Set of libraries and runtimes that securely deploy and process Python and other programming languages in Snowflake to develop data pipelines, machine learning models, apps, and more.

Code development & deployment
client-side libraries

Code execution
elastic compute runtimes

Any notebook / IDE



Snowpark API

For data pipelines, apps, and more

Snowpark ML API

For ML features & models

Virtual Warehouse

Python | Java | Scala
CPU

Snowpark Container Services

Any language
CPU & GPU



Language of Choice on a Single Platform



Scalability Without Operational Complexity

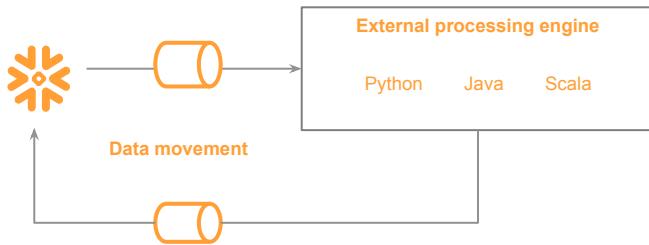


No Governance and Security Trade-offs

Programmability with Snowpark

Without Snowpark

Data is moved in and out of Snowflake for processing



- 🚫 Complexity and costs from managing multiple environments
- 🚫 Manual management and tuning of clusters by experts
- 🚫 Loose and inconsistent security and governance controls from data movement

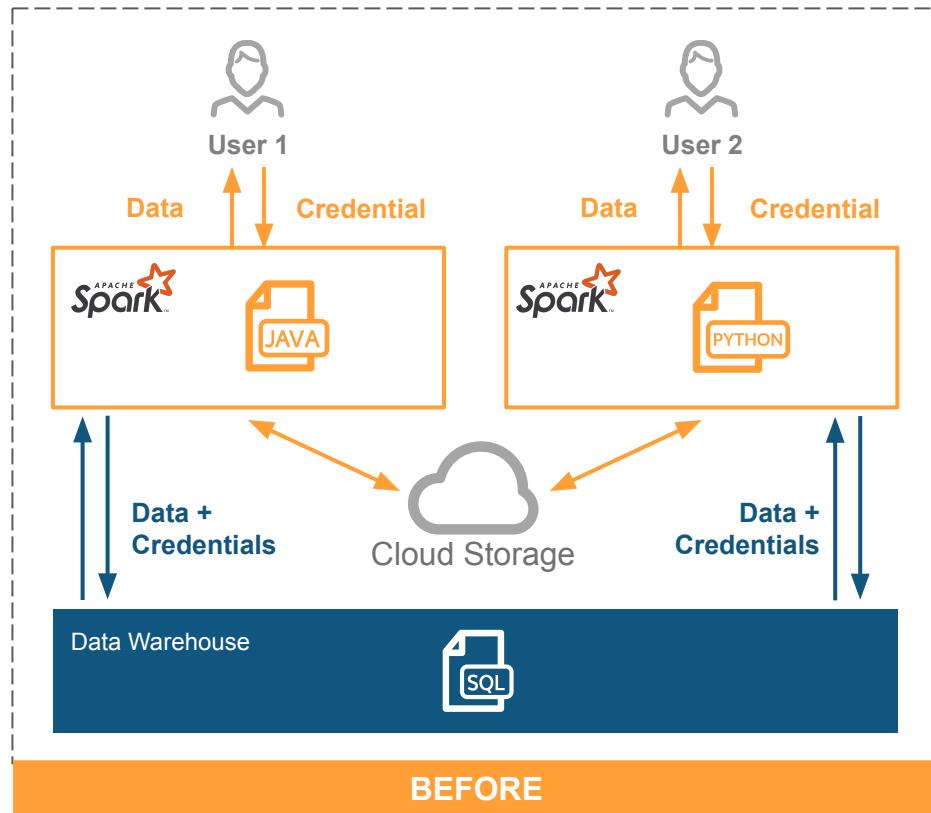
With Snowpark

Python and other programming code are processed in Snowflake



- ✅ Single platform with native support for any programming language
- ✅ Elastic scalability without maintenance or overhead
- ✅ Consistent, enterprise-grade governance controls & security

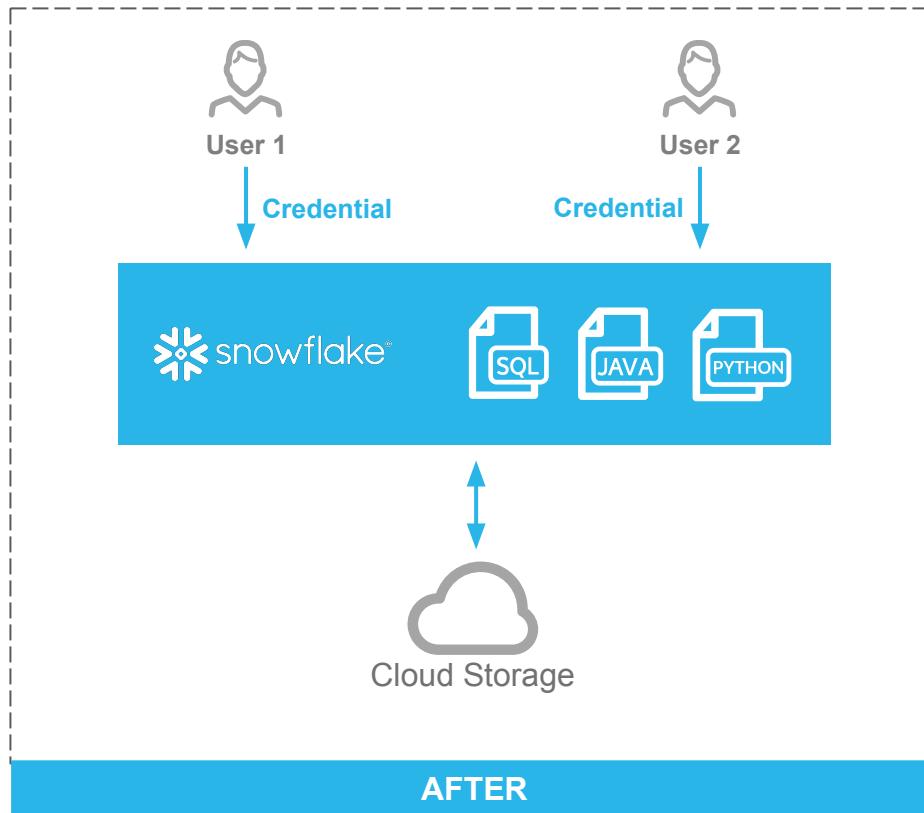
COMPLEXITY WITH TRADITIONAL APPROACH



- Customers often run separate processing clusters for different languages
- Complex capacity management & resource sizing
- Lots of data movement and data silos
- Loose governance control and security loopholes



STREAMLINED ARCHITECTURE WITH SNOWFLAKE



- One single platform with native support for different languages
- Simpler capacity management & resource sizing
- Streamline architecture and collaborate on the same data
- Consistent governance and security policies



Snowpark

CLIENT SIDE
LIBRARIES



PYTHON • JAVA • SCALA

Snowpark
DataFrame API

Snowpark ML API

ML Modeling
API (PuPr)

ML Operations
API (PrPr)

SERVER SIDE
RUNTIMES

UDFs

Stored Procedures

Built-in Anaconda
Packages



& more

Warehouses
(Standard & Snowpark-Optimized)

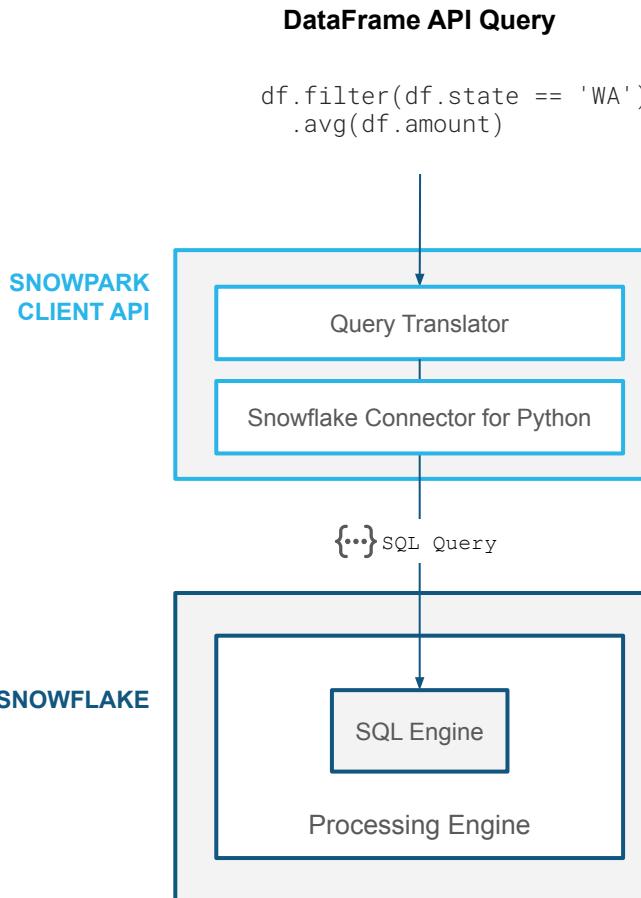
Services

Jobs

Snowpark Container Services
(PrPr)



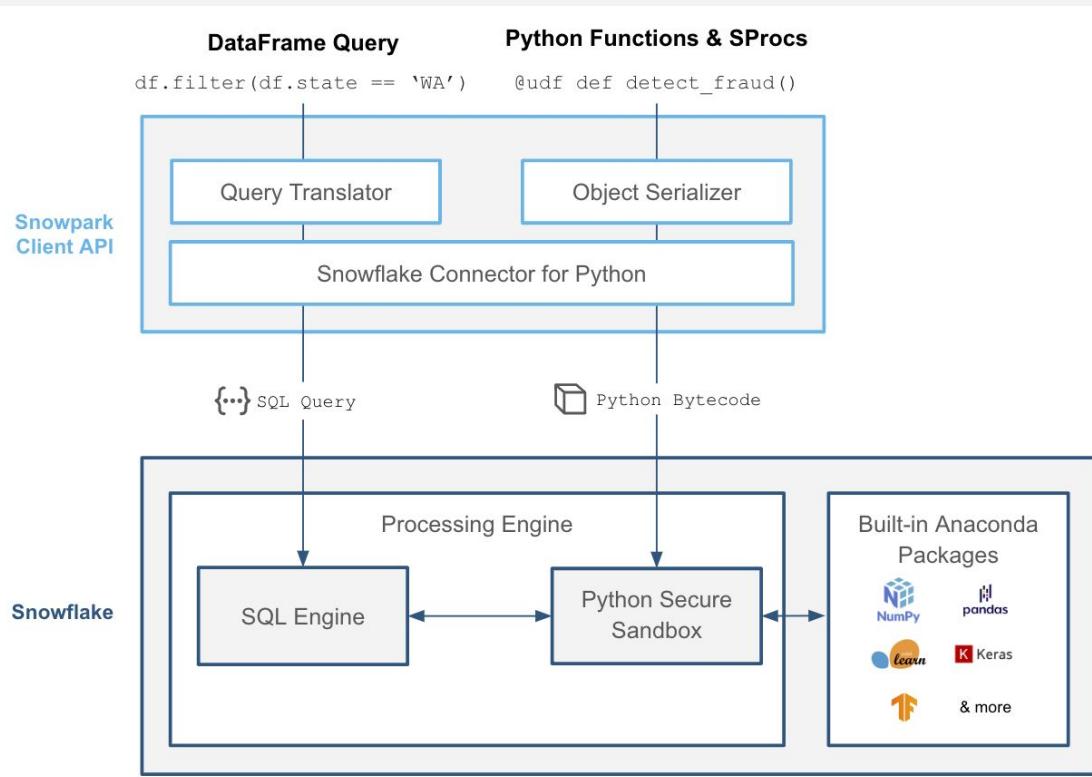
DataFrame API



- Query Snowflake data with Python
- Familiar DataFrame API
- 100% push-down to Snowflake
- Native Snowflake performance and scale



Snowflake Processing Engine



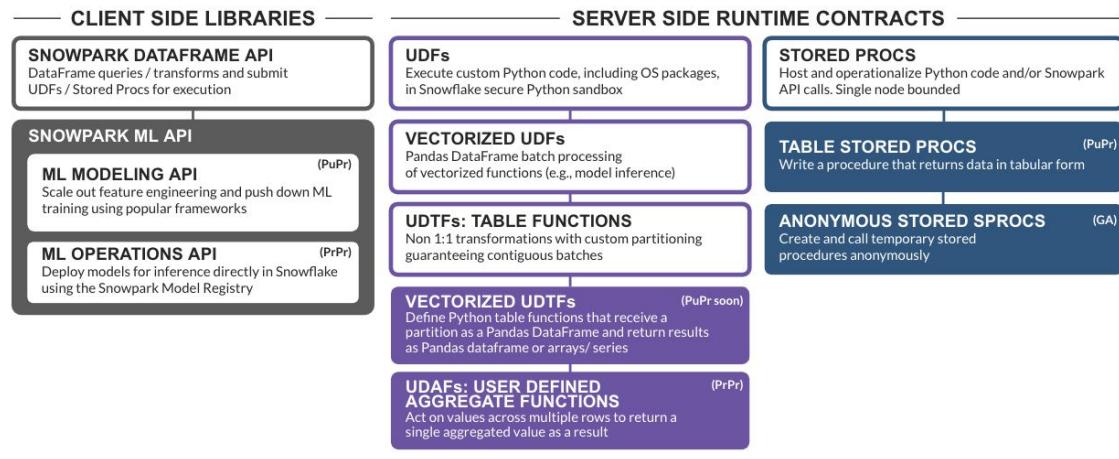
Snowpark Dataframes & Python Code

An advantage Snowpark provides is dataframes that push down transformations to the Snowflake processing engine. This is very efficient for ML pipelines because transformations can be defined using dataframes, and data does not leave the platform.

Python code can leverage all Anaconda packages and can be serialized into Snowflake in the form of Stored Procedures or UDF/UDTs. This provides a secure environment to build and run inference in your data pipelines.

This removes the complexity of having to use Spark clusters for transformation and inference.

Snowpark for Python



Client Libraries & Server Side Runtime

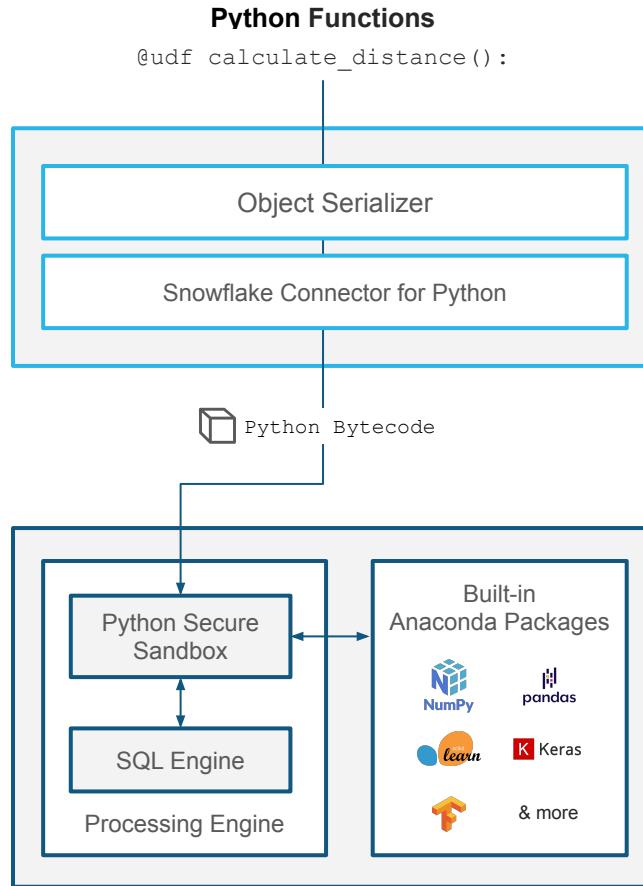
These are the different libraries and contracts available within Snowpark for Python. This allows to push down processing into Snowflake. Either using client side libraries run from any notebook or defining all possibilities of User Defined Functions (UDFs) and Stored Procedures.

[Snowpark ML Modeling](#) provides the Python APIs for feature engineering, model training and deployment.

These contracts allow Data Scientist to embed first party objects within Snowflake data pipelines. Data does not have to leave the platform while doing training or inference.



SNOWPARK CLIENT API



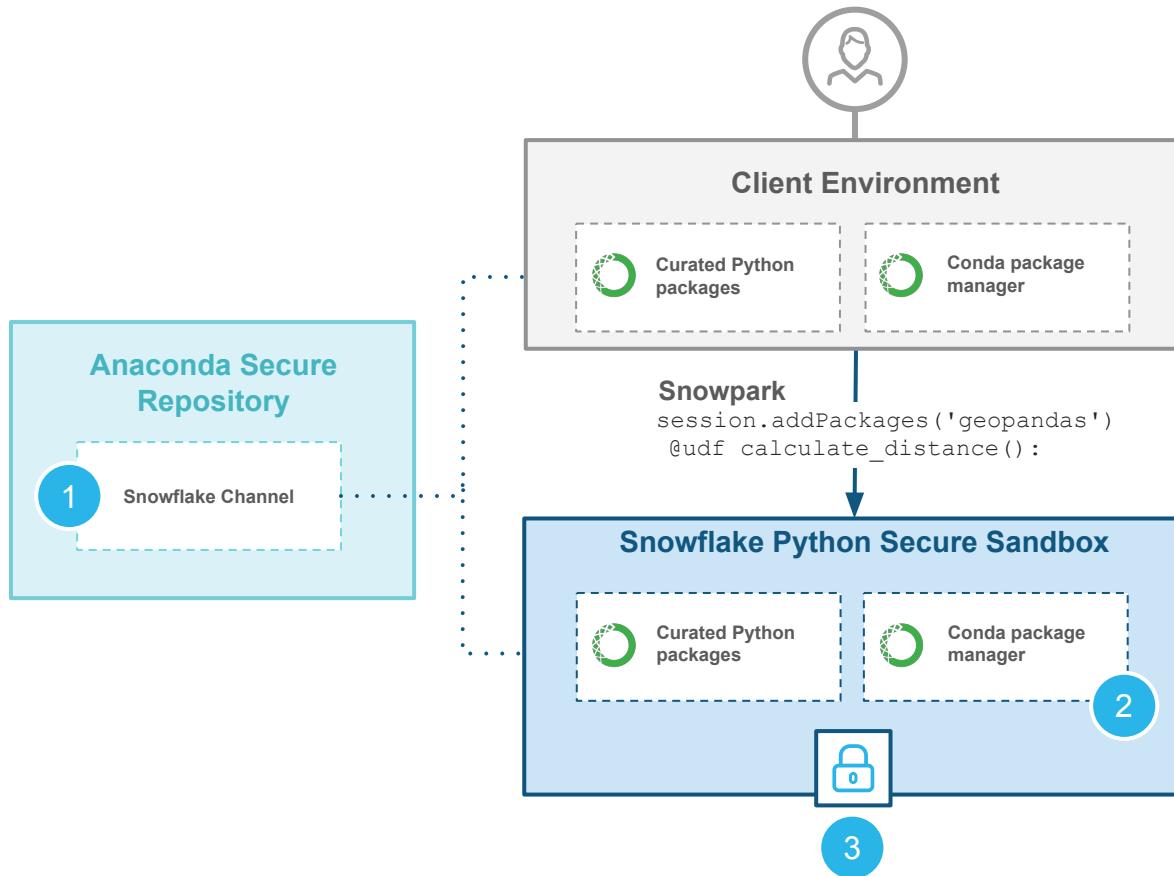
SNOWFLAKE

Python Functions

- > Bring custom Python code to Snowflake as User Defined Functions (UDFs)
- > Code is serialized and pushed down to run in a secure sandboxed environment
- > Seamlessly access third-party packages with Anaconda integration



SNOWFLAKE + ANACONDA



1 Easy Access

Curated packages pre-installed in Snowflake also available for local development

2 No Dependency Hell

Conda package manager integrated in Snowflake secure sandbox

3 Scalable and Secure

Process with secure sandbox integrated into Snowflake processing engine

All of this with no additional charges beyond warehouse usage



SNOWPARK

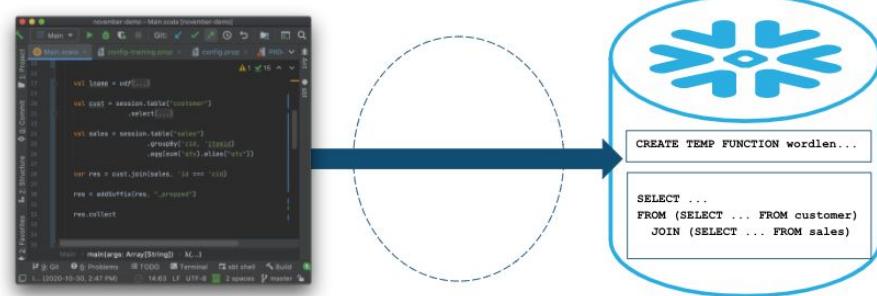


Example Use Cases:

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

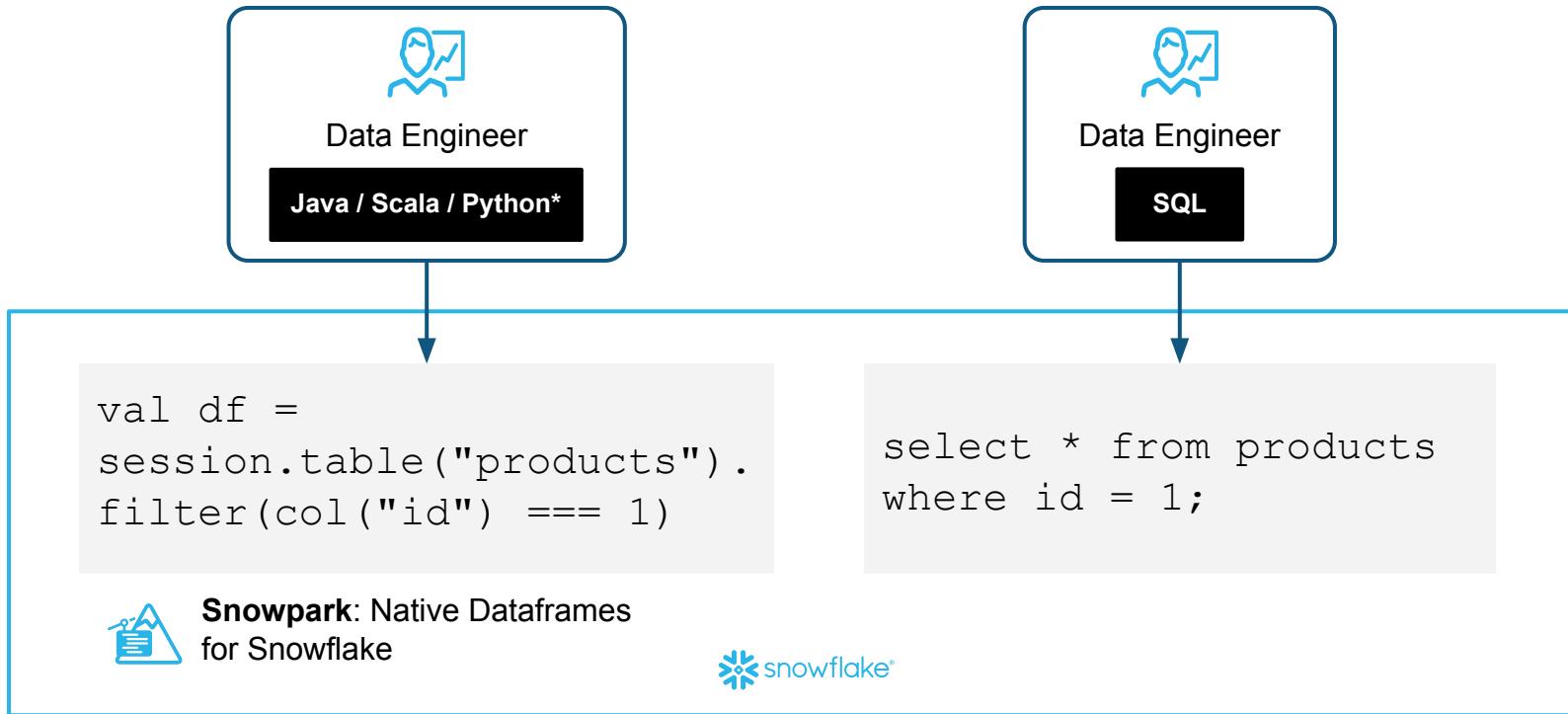
Allows coders to:

- **Code the same way, execute faster:** Use familiar language, constructs, libraries and IDE to code the same way, but get better price/performance and efficiency by optimizing execution with Snowflake's elastic performance for automated scalability.
- **Focus on what matters with streamlined architecture:** Fully managed platform with automation, no more maintenance and tuning. Fewer systems to interact with, and no need to build and manage unnecessary data pipelines to move data in and out.
- **Eliminate redundant data processing:** Benefit from the Data Cloud to process data once and make it available for all of your use cases.



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

HOW IT WORKS



Your Turn

Basics of Using Snowpark Python



- Open Labs → snowpark → README.md
- Lab generally follows this flow
 - Set up vs.code
 - Set up your Python environment
 - There are 4 labs of increasing complexity
 - Part 1 - Basics - connection, retrieval, etc.
 - Part 2 - Joins & Views
 - Part 3 - Writing Data
 - Part 4 - Stored Procs & Functions (WE WILL SKIP)



Your Turn

Basics of Using Snowpark Python



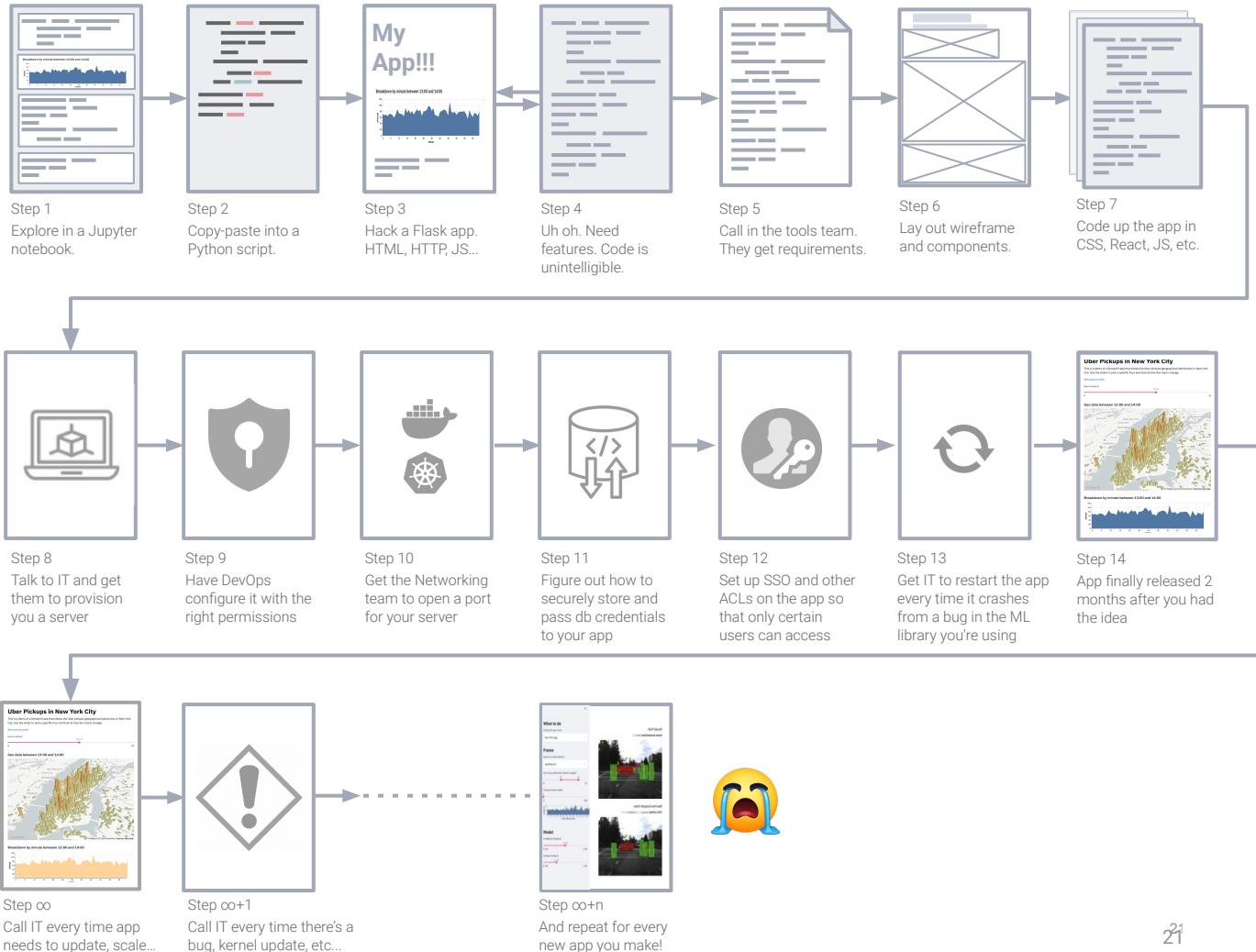
Streamlit



© 2024 Snowflake Inc. All Rights Reserved

Getting from data to action is hard

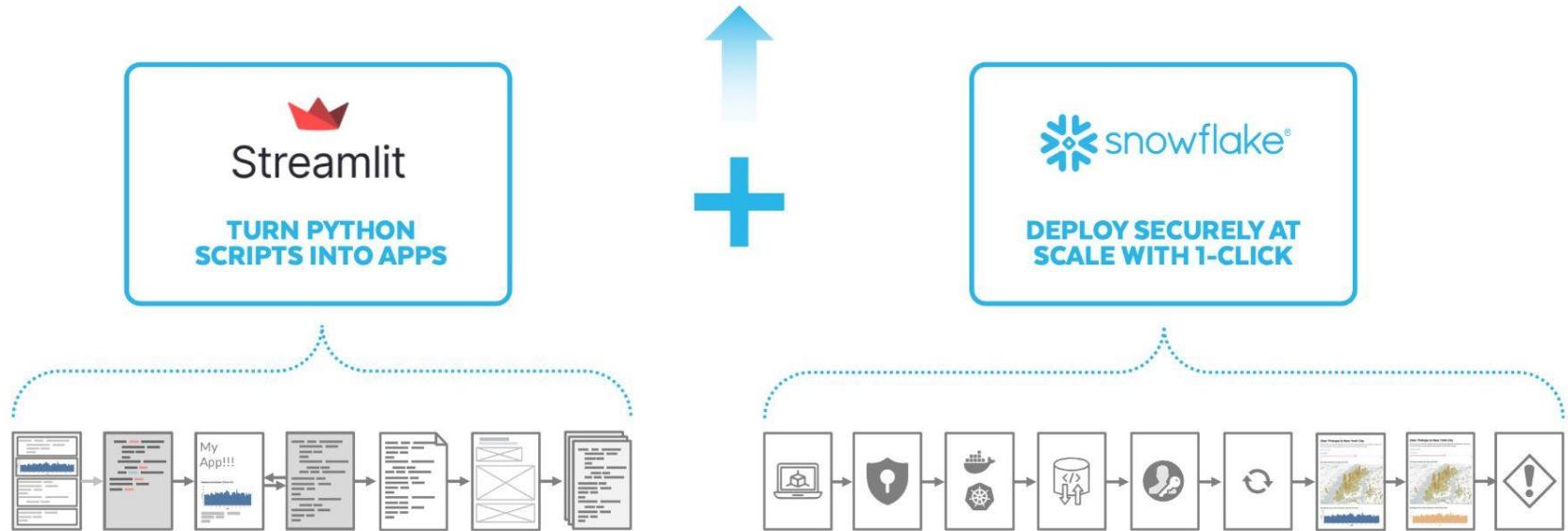
Only a fraction of data questions are ever answered since companies haven't built tools to capture or analyze the data. 73% of data is rendered effectively unused* because it's just too hard to build the tools to use it.



*Forrester report



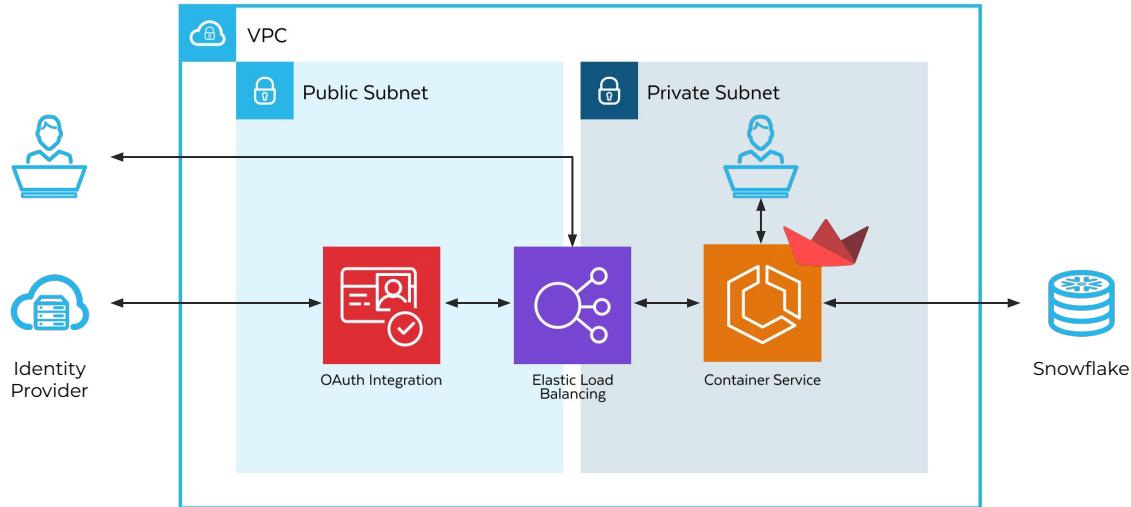
STREAMLIT IN SNOWFLAKE (SiS)



Note Streamlit Management & Administration Types

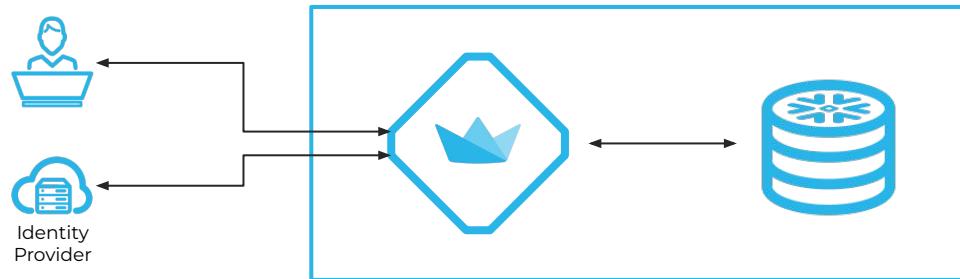
Streamlit OSS

- Security and authentication has to be self built and maintained
- Deployment pipelines have to be built and maintained



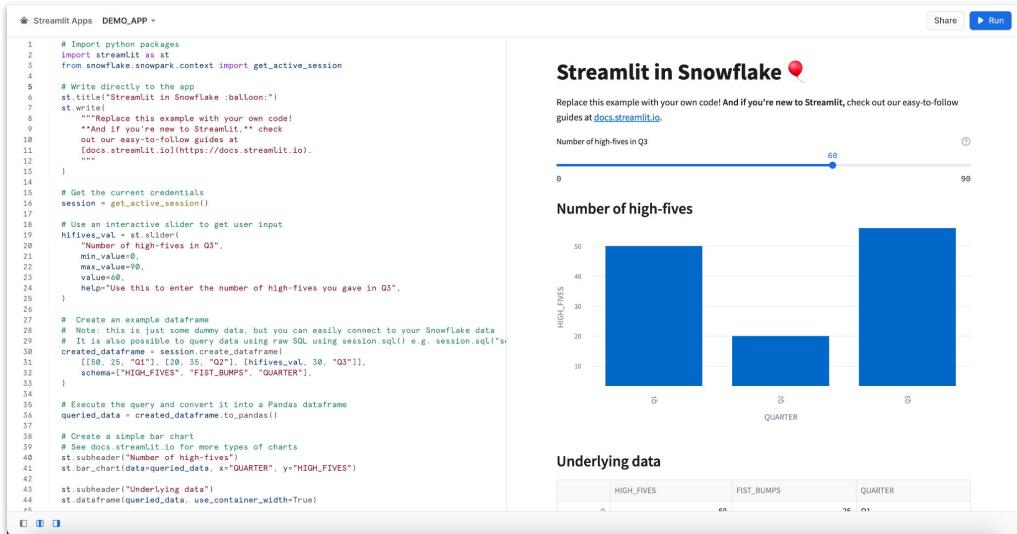
Streamlit in Snowflake

- Securely Build
- Deploy in one-click
- Share apps on Snowflake's Data cloud



Streamlit in Snowflake

Goal: Make it easy to **build**, **deploy**, **iterate**, and **share** Streamlit apps



The screenshot shows the Streamlit in Snowflake interface. On the left, there is a code editor window titled "Streamlit Apps DEMO_APP" containing Python code for building a Streamlit application. The code imports Streamlit and connects to a Snowflake session. It includes a slider input for "Number of high-fives in Q3", a bar chart titled "Number of high-fives" showing data for three quarters, and a table titled "Underlying data" showing the same data in a tabular format. The right side of the interface displays the resulting Streamlit app with the title "Streamlit in Snowflake". The app features a header with a "Share" button and a "Run" button. Below the header, a note says: "Replace this example with your own code! And if you're new to Streamlit, check out our easy-to-follow guides at [docs.streamlit.io](#)! (<https://docs.streamlit.io>)". The main content area shows a slider set to 68, a bar chart with values [50, 20, 60] for the three quarters, and a table with the same data.

```
1 # Import python packages
2 import streamlit as st
3 from snowflake.snowpark.context import get_active_session
4
5 # Write directly to the app
6 st.title("Streamlit in Snowflake :balloon:")
7 st.write(
8     """Replace this example with your own code!
9     **And if you're new to Streamlit:** check
10    out our easy-to-follow guides at
11    [docs.streamlit.io](https://docs.streamlit.io).
12    """
13 )
14
15 # Get the current credentials
16 session = get_active_session()
17
18 # Use an interactive slider to get user input
19 highfives_val = st.slider(
20     "Number of high-fives in Q3",
21     min_value=0,
22     max_value=90,
23     value=68,
24     help="Use this to enter the number of high-fives you gave in Q3",
25 )
26
27
28 # Create an example dataframe
29 # Note: this is just some dummy data, but you can easily connect to your Snowflake data
30 # It is also possible to query data using raw SQL using session.sql() e.g. session.sql("sql")
31 created_dataframe = session.create_dataframe([
32     [50, 25, "Q1"], [20, 35, "Q2"], [highfives_val, 30, "Q3"],
33 ], schema=["HIGH_FIVES", "FIST_BUMPS", "QUARTER"],
34 )
35
36 # Execute the query and convert it into a Pandas dataframe
37 queried_data = created_dataframe.to_pandas()
38
39 # Create a simple bar chart
40 # See docs.streamlit.io for more types of charts
41 st.subheader("Number of high-fives")
42 st.bar_chart(queried_data, x="QUARTER", y="HIGH_FIVES")
43
44 st.subheader("Underlying data")
45 st.dataframe(queried_data, use_container_width=True)
46
```

- Designed for internal apps and internal users
 - Accessed via Snowsight
- Part of the overall Snowflake ecosystem
 - Close to the data
 - Simple, secure, scalable, etc
 - Leverage Python to turn data and ML models into interactive applications
 - Is simply another Snowflake object



What can you do with a Streamlit Object?

- STREAMLIT is a managed database object inside Snowflake
- Sits in a Database and a Schema
- Needs a Warehouse to run
 - All users use this same warehouse when they visit the Streamlit
- CREATE, ALTER, DROP, etc SQL DDL
 - Can create a Streamlit from code stored in a Stage
- Access controlled through RBAC
- Can be included in a Native Application



SNOWPARK DAY: Learn AI/ML Workflows in Snowflake. Register now for free [A-Z](#) [RECENT](#) [DURATION](#)

Choose a language ▾

Filter by category ▾

A Image Recognition App in Snowflake using Snowpark Python, PyTorch, Streamlit and OpenAI

28 min

Updated Jan 19, 2024

[START](#)

Analyzing real estate properties using Streamlit

36 min

Updated Jan 19, 2024

[START](#)

Building a data application with Snowflake Marketplace, Snowpark and Streamlit

43 min

Updated Jan 19, 2024

[START](#)

Data Mapping in Snowflake Native Apps using Streamlit

42 min

Updated Jan 19, 2024

[START](#)

Frosty: Build an LLM Chatbot in Streamlit on your Snowflake Data

46 min

Updated Jan 19, 2024

[START](#)

Getting Started With Snowpark for Python and Streamlit

38 min

Updated Feb 27, 2024

[START](#)

Getting Started with Amazon Managed Service for Grafana and Streamlit On Real-time Dashboarding

46 min

Updated Jan 19, 2024

[START](#)

Getting Started with Bedrock, Streamlit and Snowflake

55 min

Updated Feb 21, 2024

[START](#)

Getting Started with Generative AI in Snowflake and Streamlit

60 min

Updated Jan 19, 2024

[START](#)

NLP and ML with Snowpark Python and Streamlit for Sentiment Analysis

35 min

Updated Jan 19, 2024

[START](#)

Quickstart Guide: Cybersyn Streamlit in Snowflake - Financial Demo

9 min

Updated Jan 19, 2024

[START](#)

Snowflakeマーケットプレイス、Snowpark、Streamlitによるデータアプリケーションの構築

42 min

Updated Feb 21, 2024

[START](#)

Wrap-Up



Resources



SNOWFLAKE UNIVERSITY



<https://community.snowflake.com/s/snowflake-university>

A black tablet device is shown from a slightly elevated angle, displaying the Snowflake University website on its screen. The website has a dark blue header with the text 'SNOWFLAKE UNIVERSITY' and 'Snowflake's on-demand training tool.' Below this is a large orange 'GET STARTED' button. The main content area is titled 'COURSES AVAILABLE' and lists four categories: 'EARN YOUR DIGITAL BADGE', 'LEARN IN PHASES', 'LEARN THE BASICS', and 'MOVE ONTO THE ADVANCED'. Each category has a small icon and a brief description. At the bottom of the screen, there is a footer bar with links for 'Hands On Essentials Web User Interface Course', 'Level Up Series', 'Snowflake Customer Onboarding', and 'Snowflake Advanced Capabilities and Transformation'. The overall theme of the website is professional and educational.

1. Snowflake Customer Onboarding

6 Foundation courses that introduce the Snowflake platform and concepts needed to successfully get started with Snowflake (Overview, UI, Security, Administration, Technical Support, Loading Data).

2. Hands-On Essentials Course

11 lessons to be completed in less than 10 hours for Snowflake fundamentals. Earn a badge.

3. Snowflake Advanced Capabilities

Topics: Cloud Services, Backup & Recovery, Resource Monitors, Ecosystem, Account Considerations

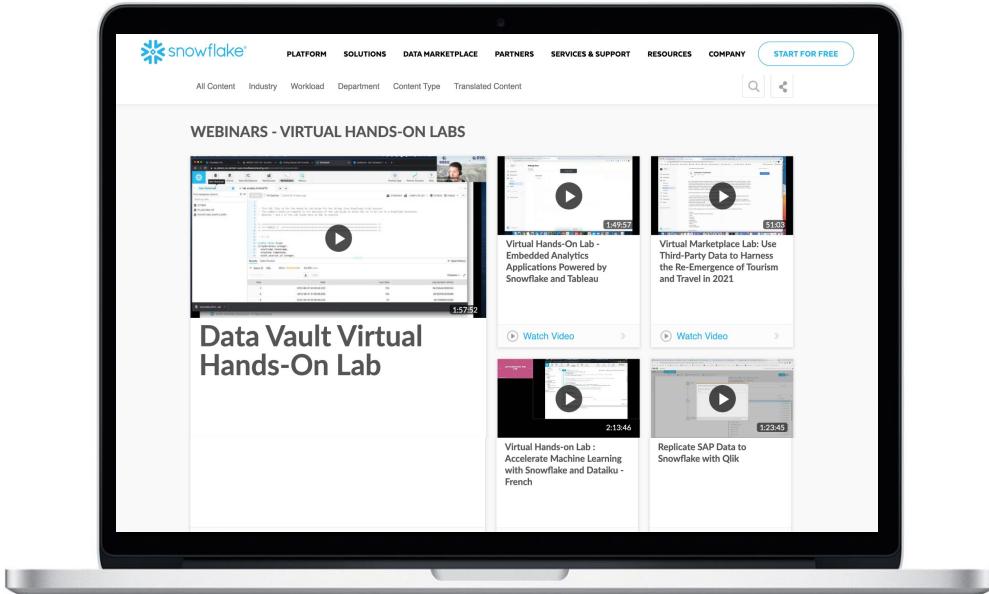
4. Level-Up Series

For more advanced topics and preparation for Snowpro Certification.



SNOWFLAKE HANDS-ON LABS

<https://resources.snowflake.com/webinars-virtual-hands-on-labs>



Join an instructor-led, virtual hands-on lab to learn how to get started with Snowflake

Webinar topics include:

Zero to Snowflake

Getting Started with Snowflake Data Marketplace

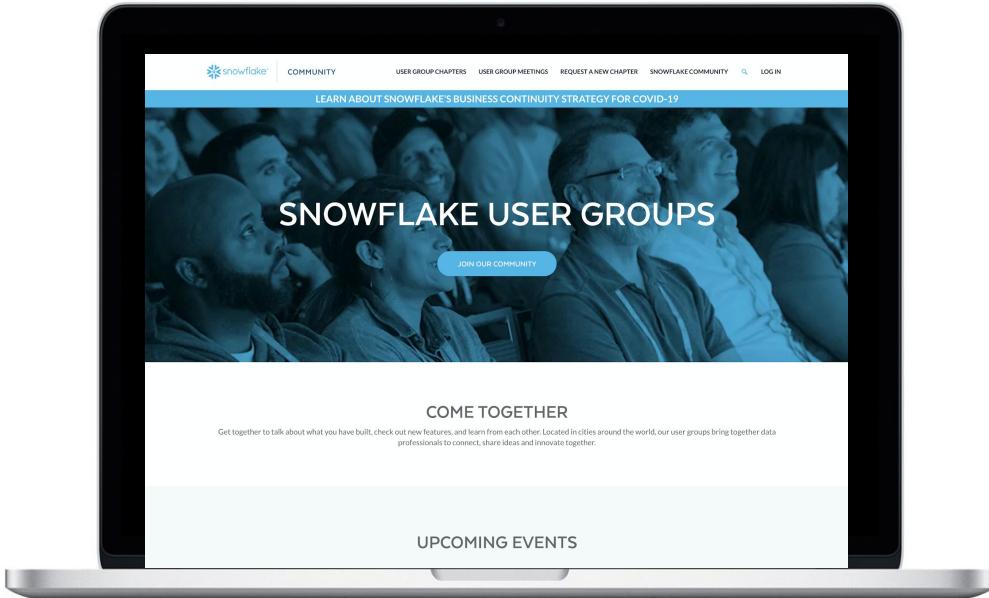
Drive Advanced Analytics with Snowflake's Data Marketplace and Snowsight

and more!



SNOWFLAKE USER GROUPS

<https://usergroups.snowflake.com/>



COME TOGETHER

Get together to talk about what you have built, check out new features, and learn from each other. Located in cities around the world, our user groups bring together data professionals to connect, share ideas and innovate together.



Free Snowflake Resources and Certifications

Starting Your Snowflake Journey. Learn the concepts.

- [Create a Snowflake Community Account](#)
- [Level Up 100: First Concepts](#)
- [Level Up 100: Key Concepts](#)
- [Level Up 100: Snowflake Ecosystem](#)
- [Level Up 100: Accounts and Assurances](#)
- [Level Up 100: Account Hierarchy](#)
- [Level Up 100: Data Loading](#)
- [Level Up 100: Resource Monitoring](#)
- [Level Up 100: Backup and Recovery](#)
- [Level Up 100: Final Exam](#)
- [Level Up 200: Query History and Caching](#)
- [Level Up 200: Context](#)

Now let's apply those concepts to workshops (and earn badges!!!)

- [Badge 1: Data Warehouse Workshop](#)
- [Badge 2: Data Application Builders Workshop](#)
- [Badge 3: Sharing, Marketplace & Exchanges Workshop](#)
- [Badge 4: Data Lake Workshop](#)
- [Badge 5: Data Engineering Workshop](#)

Excellent! We've learned the concepts, we've completed the workshops, now let's complete some labs using our Snowflake account!

- [Getting Started with Snowflake - Zero to Snowflake](#)
- [Snowflake in 20 Minutes \(Learning SnowSQL - CLI Client\)](#)



Additional Content

Snowflake Quickstarts: <https://quickstarts.snowflake.com/>

Snowflake Architecture Patterns: [Click Here](#)

Stack Overflow: [Stack](#)

Snowflake Free Trial: [Click Here](#)

Snowflake University: <https://community.snowflake.com/s/snowflake-university>

Snowflake Community: <https://community.snowflake.com/s/>

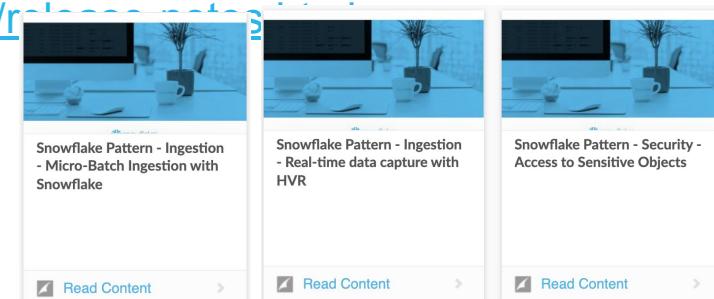
Snowflake Documentation: <https://docs.snowflake.com/en/>

Snowflake Demos & Best Practices: [Click Here](#)

Snowflake Release Notes: <https://docs.snowflake.com/en/rn/>

Snowflake Public Preview: [Click Here](#)

Snowflake Blog: <https://www.snowflake.com/blog/>



THANK YOU!!!!

Jeremy Lemmon

jeremy.lemmon@snowflake.com

[linkedin.com/jlemonn/](https://www.linkedin.com/in/jlemonn/)

615.335.0320



Data for Breakfast

START YOUR DAY WITH
AI—THE EASY WAY

NASHVILLE TUESDAY, MARCH 26 AT 8:00 A.M. THE WESTIN NASHVILLE

[https://www.snowflake.com/events/
data-for-breakfast/nashville/](https://www.snowflake.com/events/data-for-breakfast/nashville/)



© 2024 Snowflake Inc. All Rights Reserved