DEFINITIVE GUIDE TO MANAGING SPEND IN SNOWFLAKE

snowflake

WHITE PAPER

TABLE OF CONTENTS

FOR EVERYONE

FOR SNOWFLAKE ADMINISTRATORS

FOR EVERYONE

HOW TO GET STARTED WITH MANAGING SPEND IN SNOWFLAKE

This guide provides a detailed overview of how to manage your spend in Snowflake so that you can use the platform efficiently, consume sustainably and deliver maximum business value.

Consumption models like Snowflake's offer several advantages including flexibility, scalability and speed. Snowflake also automatically rolls out regular performance enhancements in query times and query processing that customers can benefit from without taking any action. These improvements in Snowflake performance experienced by customers over time can be tracked regularly with the **Snowflake Performance Index (SPI)**. However, it is important to proactively manage your consumption costs to ensure that you are getting the most value from your Snowflake investment. Organizations, especially larger ones with more complex and at-scale Snowflake deployments, should consider implementing a Financial Operations (FinOps) framework to collaboratively manage Snowflake costs across multiple business functions including finance, technology, operations and procurement teams. Such a framework should factor in strategy, people, process and technology to optimize costs, improve predictability and better align costs with business value. With the right FinOps framework in place, your organization can fully leverage Snowflake to gain better visibility, control and optimization of your spend.

To give you a better understanding of your Snowflake bill, the different components of Snowflake costs will be dissected later in this chapter.

The rest of the chapters in this guide are aimed toward Snowflake administrators ("admins"). You'll read about practical features and technical best practices, organized by difficulty level of implementation, that you can use right away to gain better visibility, control and optimization of your Snowflake spend. Every organization should immediately implement the "**Quick actions to take in every Snowflake account to optimize costs**" and the "Beginner" level best practices outlined in each of the chapters on **Visibility**, **Control** and **Optimization**. If you are more familiar with Snowflake and would like to maximize the return on your Snowflake investment across different use cases, implement the "Intermediate" and "Expert" level best practices for managing spend.

Consumption models enable flexibility for changing business priorities

Cloud-based consumption models like Snowflake's provide customers with the flexibility to pull different levers based on what's most important to your business at a specific time. When supporting growth is your top priority, you can scale up instantly to meet increased demand. For instance, organizations can add new users, activate new use cases and onboard new data sources quickly and easily.

As workloads grow and priorities change, teams periodically look for ways to use resources more efficiently. Some organizations may not have the necessary frameworks implemented to effectively manage spend. Others might not be maximizing Snowflake's built-in capabilities to identify inefficiencies and determine how to scale them back.

Implementing a FinOps framework to understand, manage and optimize Snowflake costs

Organizations transitioning from on-premises infrastructure to cloud solutions like Snowflake may face challenges related to cloud waste and unsustainable consumption patterns, primarily due to being unfamiliar with effectively managing resources in a new consumption-based cloud world. Without the right strategies to identify and eliminate waste, organizations risk escalating costs over time. Adopting a FinOps framework can help overcome these challenges by fostering collaboration across finance, technology, operations and procurement teams to ensure financial accountability in cloud consumption. This is important because it promotes cost transparency and efficient resource use, enabling the enterprise to better understand, manage and optimize Snowflake spend. Organizations should consider setting up a holistic FinOps framework that takes into account strategy, people and process in addition to technology.

STRATEGY

Data and analytics strategies traditionally have prioritized time-to-value and performance, but this is changing. To balance these priorities with cost management, the following are best practices:

- Make cost management a key pillar of your data platform strategy while ensuring that costs are aligned with business value.
- Build a culture that prioritizes the management and optimization of cost through continuous education and enablement.
- Invest in people and processes to understand and continuously manage costs. Cost management is a continuous journey, not a one-time event.

PEOPLE

For organizations that have deployed Snowflake at scale, a Center of Excellence (CoE) can be an effective initiative for managing and optimizing your Snowflake usage. A CoE can help to centralize cost management efforts and provide expertise and guidance to data teams. A CoE will facilitate regular communication with stakeholders to keep them informed and engaged on spend. The CoE will fund itself through optimizations and will ensure cultural changes so that every Snowflake user is:

- Cost-aware and has visibility into their consumption
- Familiar with the platform and cost management best practices
- Trained to write efficient SQL (and now Python)

PROCESS

For transparent cost management, organizations need to implement detailed tagging and reporting systems and create an allocation model that promotes accountability through methods such as through chargebacks or showbacks to the business.

Organizations should also consider documenting the business value of cost management initiatives. Cost management on Snowflake will inevitably lead to the prioritization of use cases that drive the most business value. Customers should track the benefits realized from your Snowflake investment. This information can be used to justify the cost of Snowflake and to prioritize use cases for optimization.

TECHNOLOGY

The rest of this guide will provide in-depth coverage of best practices for the built-in product capabilities administrators should implement to optimize costs on Snowflake.

Snowflake cost management capabilities

Snowflake focuses on three dimensions of cost management capabilities and best practices:

VISIBILITY

This focuses on:

- Understanding usage, costs and value
- Taking a unit economics approach to identify relevant KPIs tied to business objectives
- Ensuring warehouses and storage can be allocated and tagged for showback. The typical target is to allocate more than 70% of total spend to individual teams and/or data products.

Customer examples:

- **Aligning unit economics with value**: One company in the delivery space is using Snowflake for embedded end-user analytics. They are tracking the cost of Snowflake per order over time to gain ongoing visibility into the relationship between cost and value delivered.
- **Showback**: A manufacturing customer's CoE is training all the distributed BI teams across various business units and departments on Snowflake credit consumption by warehouse size. They are leveraging this effort to build broad awareness on forecasted versus actual credit consumption and to emphasize the importance of efficient Snowflake usage.
- **Allocations**: One financial services company built a tagging system before and after Snowflake usage to capture the total cost of all the stack components of their data products, including but not limited to Snowflake. These names were replicated at the warehouse and schema level to cover compute and storage.

CONTROL

This focuses on:

- Converting the central IT budget into smaller team allocations that can be populated into Snowflake with different levels of guardrails
- Implementing some level of chargebacks
- Managing demand to minimize the forecast variance and documenting new use cases

Customer example:

- **Chargebacks**: One company effectively uses chargebacks to ensure that business value aligns with costs. They offer a default SLA and share warehouses across teams and workloads to optimize concurrency. However, they allocate larger, dedicated warehouses for high-priority business-critical workloads with stricter SLAs to strike a balance between performance needs and costs.

OPTIMIZATION

This focuses on:

- Optimizing pricing through initiatives such as changing service level agreements and/or product editions
- Optimizing usage through the rationalization of lower usage and duplicate data products and extending refresh cycles, such as changing from hourly runs to daily runs
- Optimizing cost through both passive performance improvements that are automatically passed on to customers by Snowflake (see the Snowflake Performance Index) and active performance improvements that are done to optimize warehouses, storage and queries. These could be done through serverless capabilities such as query acceleration service, Automatic Clustering, materialized views and search optimization.

Customer examples:

- **Price optimization**: One online retail company had a special project to do a very big audit on a massive number of logs. They created a new Snowflake account using the Standard Edition to do the audit without impacting any production workloads that are using the Business Critical Edition, resulting in cost savings of 50%.
- **Usage optimization**: One financial services company reviewed all their dashboards and realized that 30% were either unused or duplicates. They went back and deleted all the back-end workflows from ingestion to curated data, reducing their Snowflake overall consumption (excluding growth) by almost 20%.
- **Cost optimization**: While storing data in Snowflake is cost-efficient due to compression savings, Snowflake is not an archiving system; the data is expected to be used primarily for compute. One media and entertainment company decreased its storage retention policy from five years to three, dramatically reducing the amount of bytes scanned in queries and resulting in storage cost savings of 36%.

The components of Snowflake cost

Snowflake's service has three primary pricing meters: compute, storage and data transfer.

COMPUTE

Compute covers spend for a) virtual warehouses, b) cloud services and c) serverless services. All compute consumption on Snowflake is measured in "Snowflake Credits."

Virtual warehouses are typically responsible for most of an account's spend. Virtual warehouses are the primary compute primitive in Snowflake with elastic and instant-on, scale up and scale out, and auto-suspend and auto-resume properties that are metered on a per-second basis.

Cloud services costs include the cost footprint of items including Snowflake's distributed back-end systems, SHOW commands, query compilation and result caches, and are only charged when these exceed 10% of the daily compute footprint from warehouses (which is rare).

Serverless services are computational services that run outside of the customer's virtual warehouses and include Snowflake capabilities such as query acceleration service, Automatic Clustering, materialized views, search optimization, Snowpipe and replication.

STORAGE

Storage covers all customer data stored in a Snowflake account and is typically a small fraction of the overall Snowflake bill. Snowflake storage is priced and measured in terabytes per month. Usage is measured daily and the monthly charge is calculated based on the average usage across the number of calendar days in the month.

Snowflake charges customers based on "compressed" storage volume—that means Snowflake storage use is measured using the compressed file size instead of the original uncompressed volume. Snowflake optimizes storage for customers with compression.

DATA TRANSFER

Data transfer costs are incurred when customers transfer data out of Snowflake or from one Snowflake deployment to another and are usually a very small fraction of the overall Snowflake bill. The Snowflake data transfer price mirrors the cloud provider data transfer pricing and is metered based on the number of terabytes transferred. Data transfer charges are accrued when functionalities such as copy, replication and external functions are used.

FOR SNOWFLAKE ADMINISTRATORS

QUICK ACTIONS TO TAKE IN EVERY SNOWFLAKE ACCOUNT TO OPTIMIZE COST

Below you'll find detailed best practices for better visibility, control and optimization of your Snowflake spend, organized by difficulty level of implementation. Links to relevant queries housed in Snowflake's resources are provided so that you can easily copy and paste scripts right into your worksheets. To start, every admin should set up the below features right away as they are low-hanging fruit to help you better manage costs immediately.

Enable auto-suspend and auto-resume (on by default)

A warehouse can be set to automatically resume or suspend so that organizations are billed for resources consumed based on actual usage. By default, Snowflake automatically suspends the warehouse if it is inactive for a specified period of time. Also by default, Snowflake automatically resumes when any job arrives at the warehouse. Auto-suspend and auto-resume behaviors apply to the entire warehouse, not to the individual clusters in the warehouse. An important callout is that setting auto-suspend duration to zero does not mean that the warehouse will shut down immediately. Instead, this zero setting will disable auto-suspend.

The "Warehouses without Auto-Suspend" query identifies all warehouses that do not have auto-suspend enabled. Enable auto-suspend on these warehouses to prevent runaway costs during inactivity.

The "Warehouses without Auto-Resume" query identifies all warehouses that do not have auto-resume enabled. Enable auto-resume on these warehouses to ensure that users can query the system when a job arrives.

Set up resource monitors

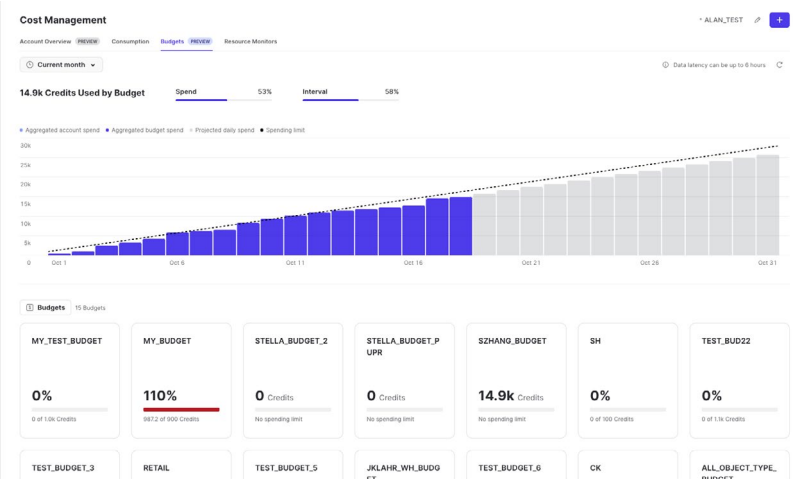
Resource monitors provide alerts and hard limits to prevent overspend via credit quotas for individual virtual warehouses during a specific time interval or date range. When warehouses reach a limit, a notification and/or suspension can be initiated. Resource monitors can also be scheduled to track and control credit usage by virtual warehouses.

The "Warehouses without Resource Monitors" query identifies all warehouses without resource monitors, which have a greater risk of unintentionally consuming more credits than typical. You should immediately set up resource monitors on all of these warehouses.

Set up an account-level budget

The **Budgets** feature (public preview on AWS) in Snowflake can be used to define a spending limit over a time interval for an account or a custom grouping of credit-consuming objects. It provides both proactive and reactive alerts on customer-stipulated spending limits.

Budgets monitor not just warehouse usage but also serverless usage (for example, Automatic Clustering, materialized views, search optimization, Snowpipe and replication). A notification is sent if the spending limit is projected to be exceeded.



Start with a small warehouse

Start with a smaller warehouse and upsize once the need for a larger warehouse is proven. See the section below, **Change to Snowpark-optimized warehouses for memory-intensive workloads**, for guidance on when to increase your warehouse size or change to a Snowpark-optimized warehouse. Choosing the right warehouse size for your workloads is one of the most powerful ways to efficiently manage Snowflake spend.

HOW TO CONFIGURE AND USE SNOWFLAKE VISIBILITY FEATURES TO UNDERSTAND, ATTRIBUTE AND MONITOR SPEND

Beginner

TRACK USAGE

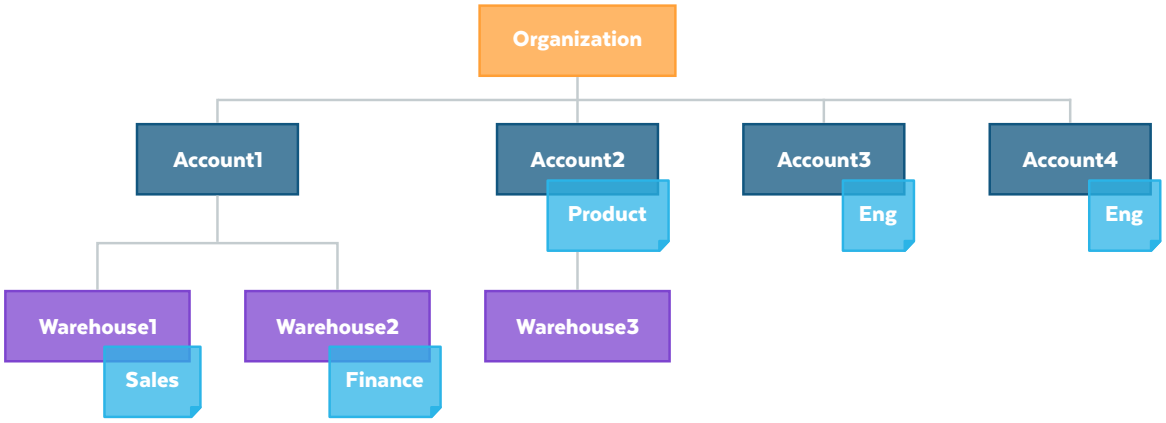
Snowflake's recommendation is that customers track usage at multiple granularities (such as at the level of the organization, account, workloads, warehouses, users and tasks). To help, Snowflake provides granular usage data in **Account and Organization usage views** that you can query. This includes data meant for both finance teams and admins to realize a robust cost management practice.

The "Billing Metrics" queries show key metrics that pertain to total compute costs from warehouses and serverless features, and total storage costs. It can serve as a starting point for additional analysis of where most of the costs come from across compute, serverless and storage and whether they are aligned with expectations.

USE UI-BASED COST EXPLORATION

Go to the **Usage** interface in Snowflake for quick and easy visual examinations of dashboards detailing usage, usage trends and timeline-based drill-downs of this usage data.

The **Snowflake Usage Insights Streamlit App** can be used to visualize usage data in Snowflake for richer insights into spend via visualizations such as heatmaps, flame graphs and scatter plots.



Intermediate

USE TAGS TO IMPLEMENT A CHARGEBACK OR SHOWBACK MODEL

Customers should not only track usage but also attribute spending to specific teams, business units, customers or cost centers via tagging to actualize "chargeback" or "showback" scenarios. Such mechanisms also help ensure that business units are held accountable for value realization reporting with a set of clearly defined criteria.

Virtual warehouses consume billable compute as Snowflake credits; by using separate virtual warehouses for every workload, the cost per workload is easily obtained. This can be aggregated at the level of the business unit for chargeback scenarios. If virtual warehouses are shared across business units or cost centers, per-job cost attribution (private preview soon) under QUERY_HISTORY can be used to attribute the cost of individual jobs to help with chargeback use cases.

Furthermore, system tags for cost optimization in the SNOWFLAKE.CORE schema: COST_CENTER, ENVIRONMENT, PROJECT (in private preview) can be used to organize and attribute spend across organizational hierarchies.

Tagging helps attribute spend to specific projects, cost centers and business units.

Expert

RUN ADVANCED QUERIES FOR DEEPER SPEND VISIBILITY

Gain deeper visibility into Snowflake spend and usage with the collection of queries below that provide more granular detail into consumption patterns over time. The results from these queries can help you identify spikes or other irregularities in the credit consumption pattern, which are launching points to drive further investigation. Consistently high consumption for certain features can provide insight into patterns of resource allocation that should be flagged for additional action.

Most expensive queries from the last 30 days

You can get the most expensive queries from the last 30 days based on warehouse size used and query execution time using the "Most Expensive Queries" query. With the results, the admin can look at the query profile, contact the user who executed the query and take action to optimize how these queries are run.

Partner tools consuming most credits

The "Partner Tools Consuming Credits" query identifies which of Snowflake's partner tools or solutions (such as those for BI, ETL, etc.) are consuming the most credits so you can evaluate investments across your technology stack.

Advanced queries for visibility on replication usage

- List of replicated databases details: The "Replication Cost History (by Day by Object)" query provides a full list of replicated databases and the volume of credits consumed via the replication service over the last 30 days, broken out by day.
- Average daily credits consumed by replication: The "Replication History & 7-Day Average" query shows the average daily credits consumed by replication grouped by week over the last year.

Advanced queries for visibility on data ingestion activity

- Daily summary of all loads for each table: The "Data Ingest with Snowpipe and 'Copy'" query returns an aggregated daily summary of all loads for each table in Snowflake and shows average file size, total rows, total volume and the ingest method (copy or Snowpipe). If file sizes are too small or too big for optimal ingestion, additional investigation and optimization may be required. By mapping the volume to credit consumption, you can determine which tables are consuming more credits per TiB loaded.
- Credits consumed by pipes: The "Snowpipe Cost History (by Day by Object)" query provides a full list of pipes and the volume of credits consumed via the service over the last 30 days, broken out by day. A pipe is a named, first-class Snowflake object that contains a COPY statement used by Snowpipe. The COPY statement identifies the source location of the data files and a target table.
- Average daily credits consumed by Snowpipe: The "Snowpipe History & 7-Day Average" query shows the average daily credits consumed by Snowpipe grouped by week over the last year.

Visit the Resource Optimization Quickstart on Billing Metrics for more queries that provide visibility into the usage of serverless features such as Automatic Clustering, materialized views, and search optimization.

HOW TO CONFIGURE AND USE SNOWFLAKE CONTROL FEATURES TO LIMIT AND CONTROL SPEND

Beginner

SET UP RESOURCE MONITORS

Resource monitors provide alerting and hard limits to prevent overspend via credit quotas for individual virtual warehouses during a specific time interval or date range. When warehouses reach a limit, a notification and/or suspension can be initiated. Resource monitors can also be scheduled to track and control credit usage by virtual warehouses.

The "Warehouses without Resource Monitors" query identifies all warehouses without resource monitors, which have a greater risk of unintentionally consuming more credits than typically expected. You should immediately set up resource monitors on all of these warehouses.

CHOOSE THE APPROPRIATE WAREHOUSE TYPE AND SIZE

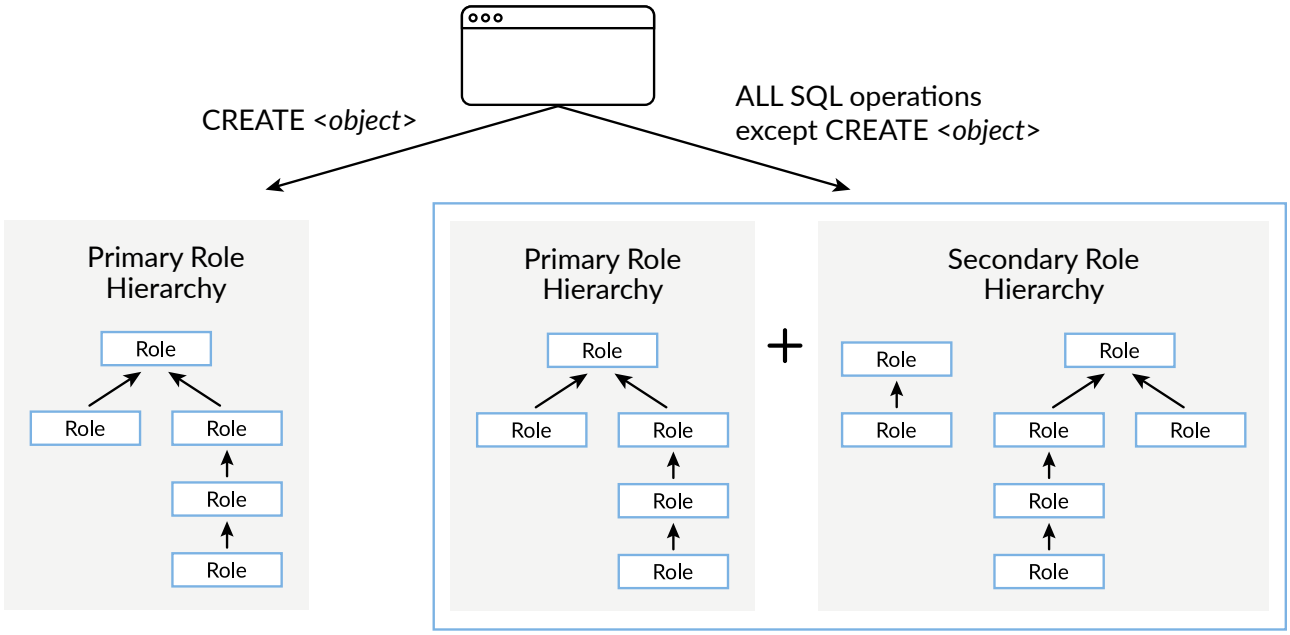
There are two warehouse types (standard and Snowpark-optimized warehouses, with the latter providing more memory-to-CPU ratio) and multiple T-shirt sizes for warehouses (XS, S, M, L, XL, … 6XL). The warehouse type determines the memory-to-CPU ratio of the virtual warehouse, while the size determines the total amount of CPU and memory available. You can modify virtual warehouse size and type, even while the cluster is currently executing queries.

You'll need to translate the criteria you reason about such a workload as a collection of queries, desired completion time for the workload (that is, service level objective), and cost budget into appropriate knob settings for warehouses. Start with a smaller warehouse size and incrementally increase the size as needed.

Warehouse sizes provide implicit upper bounds and controls for the credit burn rate and cost.

Warehouse Size (Credits / Hour)

Size	XS	S	M	L	XL	2XL	3XL	4XL	5XL	6XL
Standard	1	2	4	8	16	32	64	128	256	512
Snowpark-Optimized	N/A	N/A	**6**	**12**	**24**	**48**	**96**	**192**	**384**	**786**



Intermediate

USE MULTI-CLUSTER WAREHOUSES

While scaling a warehouse "up" increases single-query performance, using multiple clusters in the same warehouse helps with query concurrency. Each cluster is scheduled independently and can be spun up or down dynamically based on load.

With multi-cluster warehouses, Snowflake supports allocating—either statically or dynamically—additional clusters to make a larger pool of compute resources available to the same virtual warehouse, increasing job concurrency without managing multiple distinct virtual warehouses. You can specify a minimum and maximum number of clusters for a given virtual warehouse.

When different values for the minimum and maximum number of clusters are specified, Snowflake will automatically start or stop additional clusters to adjust dynamically to incoming load. As the number of concurrent user sessions and/or jobs increases and starts to queue due to insufficient resources, additional clusters, up to the specified maximum, automatically start. Similarly, if load decreases, Snowflake automatically shuts down clusters.

When multi-cluster warehouses are used, the **standard scaling policy** optimizes for minimizing queuing over conserving credits. Alternatively, the **economy scaling** policy optimizes for conserving credits by keeping running clusters fully loaded.

IMPLEMENT ACCESS CONTROL PRIVILEGES FOR CREDIT-CONSUMING OPERATIONS

For sufficiently large and mature Snowflake use cases delivered via a central platform team, implement policies that restrict which users or teams can perform credit-consuming operations such as warehouse creation, sizing, and scaling limits to control costs and reduce unintended spend by additional users.

To control costs and reduce indeterminacy, Snowflake provides the ability to **set permissions to use, create and modify credit-consuming resources** (such as warehouse creation, sizing, scaling limits and query acceleration service scale factor).

MANAGE TABLE LIFECYCLE TO CONTROL STORAGE COST IMPLICATIONS OF TIME TRAVEL AND FAIL-SAFE CAPABILITIES

Snowflake's **Time Travel** capability enables accessing data that has been changed or deleted within a predefined period. It restores data-related objects that were accidentally or intentionally deleted, backs up data from points in the past, and/or analyzes usage over specified periods. The Snowflake Fail-Safe capability provides a (non-configurable) seven-day period during which historical data is recoverable; this period starts immediately after the Time Travel retention period ends.

To help manage the storage costs implications of Time Travel and Fail-Safe capabilities, Snowflake provides "temporary" and "transient" table types in addition to the default "permanent" table type.

Transient tables exist until explicitly dropped, after which the historical data beyond the Time Travel retention period cannot be recovered. Temporary tables only exist for the lifetime of their associated session, after which data is purged and unrecoverable. Neither transient nor temporary tables come with a Fail-Safe period and can have a Time Travel retention period of either 0 or 1 day.

For long-lived tables, such as fact tables or reference data, use the "permanent" table type to ensure full protection by Fail-Safe. For short-lived tables (for example, ETL work tables that need to stay for less than a day), use "transient" or "temporary" table types to eliminate Fail-Safe costs.

Expert

DO METRICS-GUIDED WAREHOUSE RIGHTSIZING

Metrics exposed by Snowflake such as warehouse load metrics, warehouse utilization metrics (in private preview), metrics on data spilling to disk/object storage and warehouse events history (in public preview) can inform and guide optimizations to right-size compute.

Utilization data for each cluster of the virtual warehouse as a percentage (in private preview) can help identify idle capacity and inform rightsizing decisions when used with warehouse load metrics.

FOLLOW GENERAL RULES OF THUMB FOR SETTING DURATION ON AUTO-SUSPEND/AUTO-RESUME

While Snowflake defaults to automatically suspending inactive warehouses after a certain amount of time, set the auto-suspend duration based on the ability of the workload to take advantage of warehouse caches. This exercise balances the benefits of cost savings from suspending compute quickly and the price-performance benefits of Snowflake's sophisticated caching. In general, the directional guidance is as follows:

- For tasks, loading, and ETL/ELT use cases, **1 second for suspension of** virtual warehouses is likely to be the optimal choice.
- For BI and SELECT use cases, query warehouses are likely to be cost-optimal with **~10 minutes for suspension** to keep data caches warm for end users.
- For DevOps, DataOps and Data Science use cases, warehouses are usually cost-optimal at **~5 minutes for suspension**, as a warm cache is not as important for ad-hoc and highly unique queries.

COLUMN NAME	DATA TYPE	DESCRIPTION
START_TIME	TIMESTAMP_LTZ	Start of the utlization time window (in UTC)
END_TIME	TIMESTAMP_LTZ	End of the utilization time window (in UTC)
WAREHOUSE_ID	INTEGER	Interal/system-generated identifier for the warehouse
WAREHOUSE_NAME	TEXT	Name of the warehouse
CLUSTER_NUMBER	INTEGER	Internal/system-generated number for the cluster. When NULL, this is a warehouse-level metric row
UTLIZATION	NUMBER(4,1)	Utilization value between 0.0 and 100.0

HOW TO CONFIGURE AND USE SNOWFLAKE OPTIMIZATION FEATURES TO IDENTIFY AND OPTIMIZE INEFFICIENT SPEND

Beginner

CHANGE TO SNOWPARK-OPTIMIZED WAREHOUSES FOR MEMORY-INTENSIVE WORKLOADS

The Snowpark-optimized warehouse type, which can help unlock machine learning training and memory-intensive analytics use cases, provides 16x more memory and 10x more local SSD cache per VM compared to standard warehouses. The increased memory speeds up computations while larger local storage provides additional speed when cached intermediate results and artifacts, such as Python packages and JARs, are reused on subsequent runs.

When execution artifacts in Snowflake perform write operations of intermediate data, first, main memory on the virtual warehouse VMs are used; if this memory is full, data is "spilled" to local disk/SSD on virtual warehouse VMs. When this local disk is also full, data spills to remote persistent storage (object storage such as Amazon S3). This scheme removes the need to handle out-of-memory or out-of-disk errors. For performance-critical workloads, you can choose larger warehouse sizes to ensure that the intermediate data fits in memory, or at least in disk, and does not spill to object storage (for example, S3 on AWS).

Queries to help identify when Snowpark-optimized warehouses should be used

- **Scenarios that can benefit from increased warehouse size or Snowpark-optimized warehouses:** The "Scale Up vs. Out" query lists queries and warehouses that can benefit from increased size or by changing the warehouse type to Snowpark-optimized warehouses.
- Metrics exposed under the QUERY_HISTORY view such as **BYTES_SPILLED_TO_LOCAL_STORAGE** and **BYTES_SPILLED_TO_REMOTE_STORAGE** indicate the extent of memory pressure that, in many cases, can be addressed in a cost-efficient manner by moving to Snowpark-optimized warehouses of the same size.
- **Top 10 queries with bytes spilled to local and remote storage**: The "Top 10 Spillers Remote" query identifies the top 10 worst offending queries in terms of bytes spilled to local and remote storage. If these are performance-critical workloads, the queries can potentially help identify which warehouses to prioritize for increasing size or changing to a Snowpark-optimized warehouse.

Intermediate

FOLLOW GENERAL RULES OF THUMB FOR RIGHTSIZING WAREHOUSES

1. If your workload has adequate throughput and/or latency performance AND queued query load is low OR total query load <1 for prolonged periods:
   - Consider downsizing the warehouse or reducing the number of clusters.
   - Additionally, consider starting a separate warehouse and moving queued jobs there.
2. If the warehouse runs during recurring periods but the total job load is <1 for substantial periods:
   - Consider downsizing the warehouse and/or reducing the number of clusters.
3. If your workload is running slower than desired (based on throughput and/or latency measurements) AND running query load is low:
   - Consider upsizing the warehouse or adding clusters.
4. If there are recurring usage spikes (based on warehouse utilization history, in private preview):
   - Consider moving queries that represent the spikes to a new warehouse or adding clusters.
   - Also, consider running the remaining workload on a smaller warehouse.
5. If a workload has considerably higher than normal load:
   - Investigate which jobs are contributing to the higher load.

Expert

The serverless features showcased below may improve query performance for specific use cases. When configured properly, these features can help drive performance efficiencies and potential cost savings. These features may not improve price-for-performance across all use cases; talk to your account team to further explore the right optimization options for your organization's unique use cases and needs.

ENABLE FEATURES FOR OPTIMIZING WAREHOUSES

Use query acceleration service to accelerate performance for ad hoc queries

Query Acceleration Service (QAS) enables another form of vertical scaling to accelerate query performance that could potentially improve price-for-performance. Effectively, QAS acts like a powerful additional cluster that is temporarily available, when needed, to deploy alongside existing virtual warehouses. Eligible queries can offload a portion of their execution to QAS, thereby speeding up their execution.

Guidance for when QAS is likely helpful

QAS is particularly useful for large table scans with selective filters and/or aggregations. Additionally, when Snowflake detects a massive query that will scan a lot of data, QAS can free up resources on the virtual warehouse cluster to execute other queries. In some cases, this can be less expensive than scaling up to a larger warehouse and lead to more efficient use of resources. QAS compute resources are billed per second, with no minimum.

You can use the **account usage view** to identify warehouses with eligible queries. The more accelerated the queries and the higher the impact, the higher the QAS costs. If you enable QAS on a warehouse and there aren't any accelerated queries, there is no cost.

Queries to monitor usage of QAS

The **Query Acceleration scale factor** sets an upper boundary on the amount of compute resources a warehouse can lease for acceleration, thereby serving as a cost control mechanism for QAS. The scale factor is a multiplier based on warehouse size and cost. For example, if the scale factor is 5 for a medium-sized warehouse (4 credits/hour), the warehouse can lease compute up to five times its size (that is, 4 x 5 = 20 credits/hour). By default, the scale factor is set to 8. The upper limit scale factor is the maximum scale factor where the query will see any additional benefit from QAS. Start with a low scale factor (8 or less) and gradually increase it.

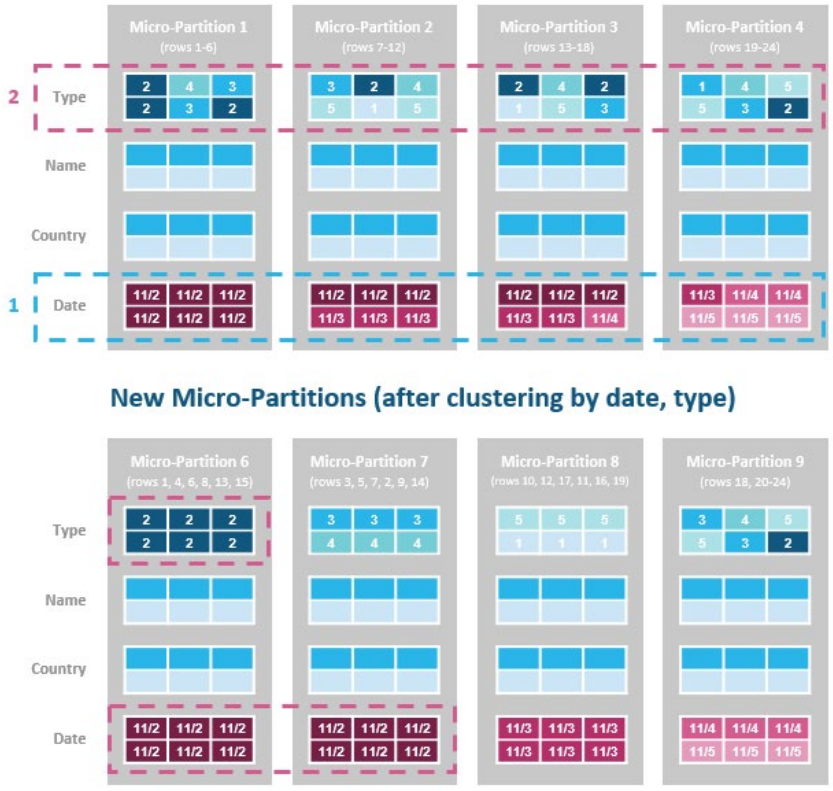
View the total number of queries with each upper limit scale factor

```
SELECT upper_limit_scale_factor,
COUNT(upper_limit_scale_factor)

FROM SNOWFLAKE.ACCOUNT_USAGE.
QUERY_ACCELERATION_ELIGIBLE

WHERE warehouse_name = '<warehouse_name>'

GROUP BY 1 ORDER BY 1;
```

ENABLE FEATURES FOR OPTIMIZING STORAGE

Use Automatic Clustering to process only relevant data from large tables

Snowflake's **Automatic Clustering** helps reorganize table data to align with query patterns that process only relevant data from large tables, thereby speeding up queries and consuming fewer compute credits. You can set a clustering key to change the organization of micro-partitions so that data is grouped by specific columns. This can improve the performance of queries that filter, join or aggregate by the columns defined in the cluster key. Once enabled, Automatic Clustering automatically reclusters micro-partitions as new data is added to the table. The pricing for Automatic Clustering is based on credits for background clustering maintenance.



Guidance for when Automatic Clustering is likely helpful

Automatic Clustering is performed in the background in an incremental and non-blocking manner without hindering running workloads. This feature helps in cases where many queries perform filter/join/aggregate operations on the same few columns. Pick a clustering key that optimizes the performance of queries with repeated latency-sensitive filters/join/aggregate operations. Usually, the largest price-for-performance boost comes from a WHERE clause that filters on a column from the clustering key.

Example query that could benefit from Automatic Clustering

This query uses the shipdate column in the WHERE clause. In this example, if you clustered the "lineitem" table by "shipdate," you could reduce the number of micro-partitions that the query engine needs to scan. As a result, each execution of the query would finish faster and with a smaller amount of warehouse resources.

```
SELECT
SUM(quantity) AS sum_qty,
SUM(extendedprice) AS sum_base_price,
AVG(quantity) AS avg_qty,
AVG(extendedprice) AS avg_price,
COUNT(*) AS count_order
FROM lineitem
    WHERE shipdate >= DATEADD(day, -90,
    to_date('2023-01-01));
```

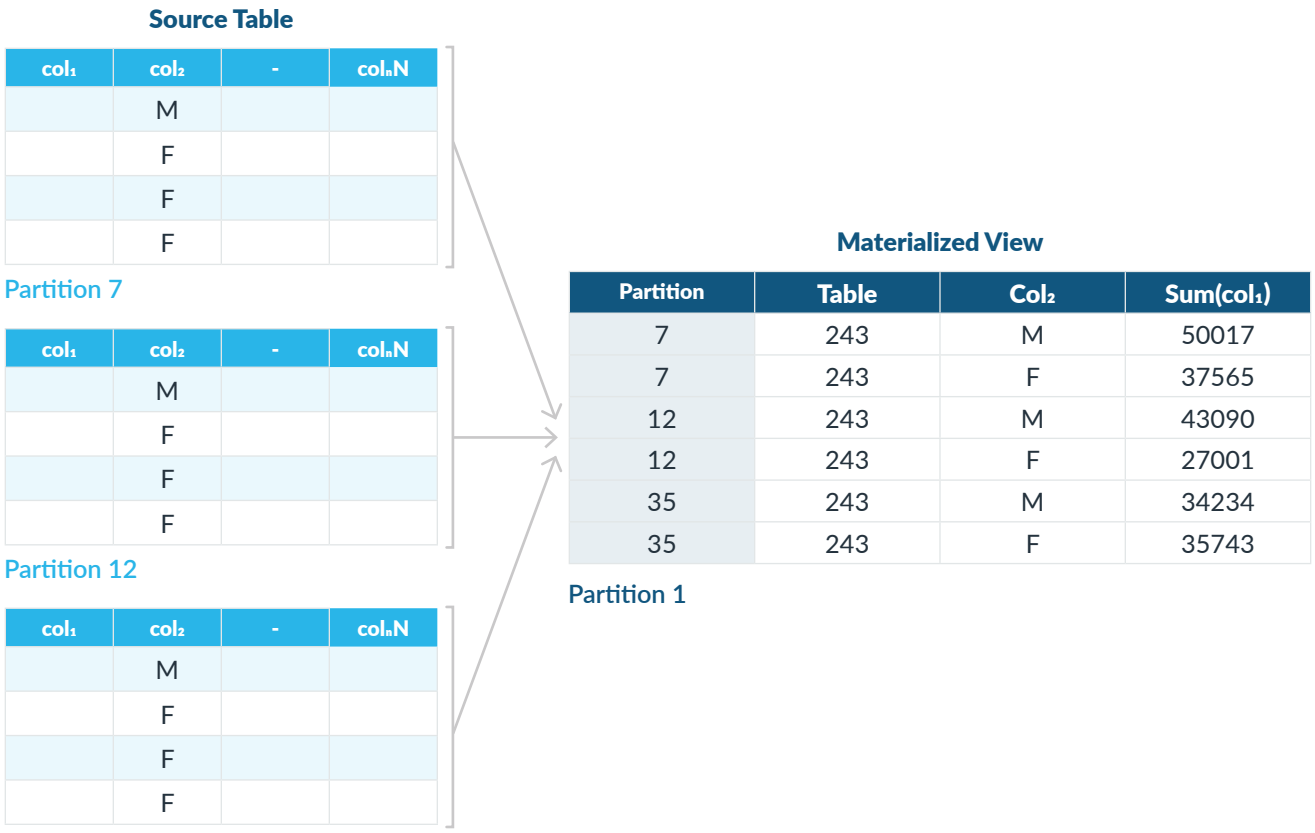
Queries to monitor usage of Automatic Clustering

Average credits consumed by Automatic Clustering

The "AutoClustering History & 7-Day Average" query shows the average daily credits consumed by Automatic Clustering grouped by week over the last year.It can help identify anomalies in daily averages over the year and investigate any unexpected changes in consumption.

Automatic Clustering history for specific table

This query shows the Automatic Clustering history for the past week for a specified table in the current account to evaluate credit spend.



Use materialized views to capitalize on pre-computed data sets

Materialized views (MVs) store frequently uses projections and aggregations to avoid expensive re-computations, which could potentially lead to improved price-for-performance. An MV is a pre-computed data set derived from a query specification stored for later use. The results from MVs are guaranteed to be current and because data is pre-computed, querying an MV is faster than executing a query against the base table on which the view is defined. An MV cannot be based on more than one table.

Guidelines for when MVs are likely helpful

MVs are recommended for workloads composed of common, repeated query patterns that return a small number of rows and/or columns relative to the base table. Consider using it in a focused manner on specific query/subquery calculations that are both intensive and frequent.

In particular, MVs are useful in the following cases:

- **Query results contain a small number of rows and/or columns relative to the base table**
- **Queries include analysis of semi-structured data or complex aggregations**
- **Queries are on an external table**

The more frequently an MV is used, the greater the benefit. MVs are billed based on the amount of required MV maintenance. The more changes that occur on the base table, the higher the MV maintenance bill. In general, MVs that are frequently used and created on base tables that don't change frequently represent the biggest opportunity for savings.

Queries to monitor usage of MVs

Average daily credits consumed by MVs

The "Materialized Views History & 7-Day Average" query shows the average daily credits consumed by MVs grouped by week over the last year. Look for spikes or changes in consumption in the daily average over the course of the year and investigate further.

MV maintenance history for current account

This query shows the MV maintenance history for the past week for the current account. This query will show you how often the MVs are being refreshed, which helps indicate credit consumption for this feature.

Use search optimization service to accelerate point lookup queries

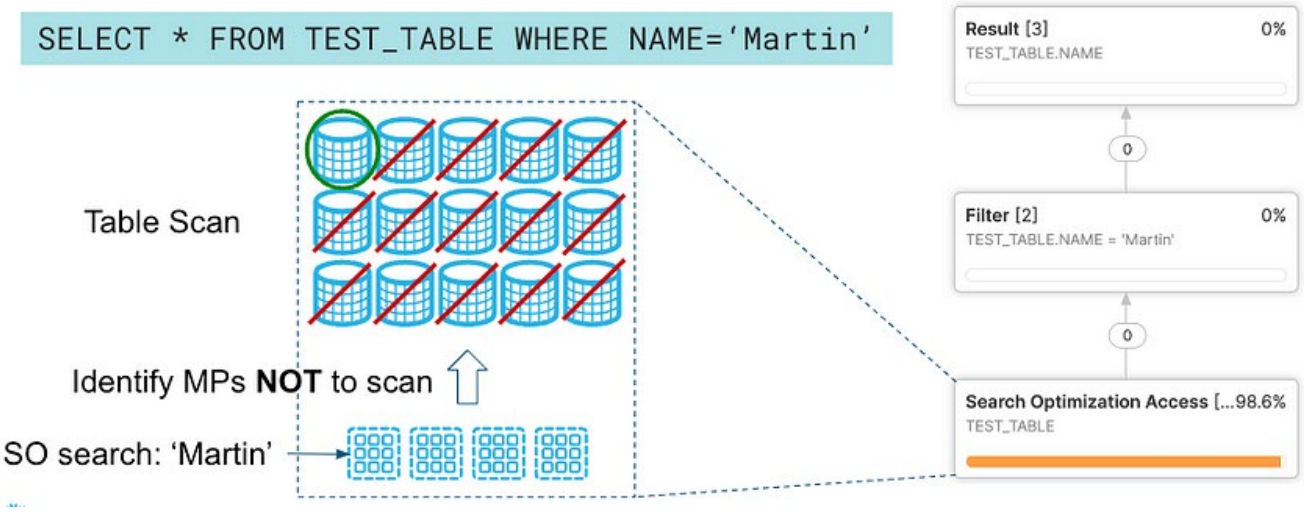
The search optimization (SO) service makes it possible to quickly find "a needle in a haystack" using highly selective filters to return a small number of rows from large tables. This includes data exploration and filtering workloads such as investigative log searches and threat/anomaly detection. It accelerates point lookup queries on columns of supported types by building a data structure that further prunes partitions. In the example below, SO reduces the number of partitions to be scanned from 15 to one, speeding up a query to find all rows with a certain name.

Guidance for when SO is likely helpful

You can enable SO for an entire table or for specific columns. If the filters are sufficiently selective, equality searches, substring searches and geo searches against those columns can be sped up significantly. When specific queries access a well-defined subset of a table's data, use the characteristics of the query to decide whether to use SO or MVs. For example, the presence of important point lookup queries can inform the use of SO for a table or column.

There is general guidance in the documentation to help you identify queries that benefit from SO.

Always run the SYSTEM$ESTIMATE_SEARCH_OPTIMIZATION_COSTS function first to estimate the cost of adding SO to a column or entire table. The estimated costs are based on the number of columns that will be enabled and how much the table has recently changed.



Queries to monitor usage of SO

List of tables with SO and credits consumed

The "Search Optimization Cost History (by Day by Object)" query provides a full list of tables using SO and the volume of credits consumed via the service over the last 30 days, broken out by day. Any irregularities or consistently high credit consumption are flags for additional investigation.

Average daily credits consumed by SO

The "Search Optimization History & 7-Day Average" query shows the average daily credits consumed by SO grouped by week over the last year. It can identify anomalies in daily averages over the year, prompting investigation into unexpected changes in consumption.

Concurrently use multiple storage optimizations

Implementing storage strategies can require greater time and financial investments than other performance optimizations (such as rewriting individual queries or optimizing warehouse sizing), but they can unlock meaningful price-for-performance improvements. Snowflake uses compute resources to maintain storage optimizations as new data is added to a table. The more changes to a table, the higher the maintenance costs. If a table is constantly being updated, the cost of maintaining a storage optimization increases.

Unlike SO and MVs, Automatic Clustering reorganizes existing data rather than creating additional storage. Similar to DML operations, reclustering consumes credits depending on the size of the table and amount of data. Each time data is reclustered, rows are physically grouped based on the clustering key for the table, resulting in new micro-partitions. Modifying a small number of rows in a table can cause all micro-partitions that contain those values to be recreated, which can result in significant data turnover and increased storage cost because the original micro-partitions are retained to support Snowflake's Time Travel and Fail-Safe features.

You can implement more than one of these strategies for a table, and an individual query with multiple filters may benefit from both Automatic Clustering and SO. If more than one strategy can potentially improve the price-for-performance of a query, start with Automatic Clustering or SO because other queries with similar access patterns can also be improved. For queries that return more than a few records, use Automatic Clustering over SO (which is best suited for point lookup queries that return a small number of rows).

If you use Automatic Clustering for optimization on a table but the table has a variety of common access patterns, you have a few different options:

- **Use a clustering key with multiple columns**
- **Create an MV based on the table (or a subset of its columns) and cluster that**
- **Use SO indexes**

It is also noteworthy that the cost of maintaining MVs or SO can be significant when Automatic Clustering is enabled for the underlying table because reclustering activities can trigger compute usage toward ongoing maintenance of MVs and SO. Additionally, SO and MVs incur the cost of additional storage and require Snowflake's Enterprise Edition or higher, increasing the price of a credit. SO and QAS can also be used together; after SO prunes micro-partitions that are not needed for a query, the rest of the work can be accelerated.

Learn more about using multiple storage optimizations together.

ADDITIONAL SNOWFLAKE RESOURCES FOR COST MANAGEMENT AND OPTIMIZATION

Snowflake is committed to continuously improving economics for customers and offers an abundance of resources on cost management and optimization. Below is a comprehensive collection of different types of content, training and additional services to help organizations of all sizes better manage and optimize usage and spend on Snowflake.

Content

Snowflake cost management documentation

Snowflake provides detailed documentation on the features described in this guide to help users better understand and easily set up cost management capabilities for their organization.

Performance and cost optimization blog

Visit the "Performance and Cost Optimization" hub of the Snowflake Medium blog to learn more about the latest features and best practices for optimizing costs on Snowflake.

10 ways to save: How to manage costs and optimize resources on Snowflake

Watch this webinar from the Snowflake product and sales engineering team to learn more about how to better understand costs on Snowflake and how to optimize resources with usage-based pricing and fine-grained controls.

Quickstart tutorials for resource optimization

The resource optimization quickstart tutorials on billing metrics, performance, setup and configuration, and usage monitoring provide step-by-step tutorials with queries to run to better monitor and manage credit consumption on Snowflake.

Training

Level up: Resource monitoring

Snowflake offers a 30-minute on-demand course on resource monitoring to oversee virtual warehouse consumption, manage costs and avoid unexpected credit usage.

Snowflake cost governance on-demand course

This five-hour Snowflake cost governance on-demand course provides the core knowledge and skills to successfully examine, control and optimize costs to maximize use of Snowflake credits.

Support

Priority support

For organizations looking for ongoing support with performance and cost-related issues, Snowflake's priority support team can proactively monitor your Snowflake environment. Using a customized performance profile, priority support can provide you with query and warehouse performance data while consulting with you on potential issues to manage your Snowflake usage.

Professional services engagements

For organizations looking for more custom and dedicated technical expertise around optimizing Snowflake, the professional services team offers several options below.

Snowflake QuickStart Service package

The Snowflake QuickStart Service package assists your team in getting started with Snowflake through a series of interactive workshops, including content on cost management, delivered by a solutions architect. The QuickStart Service package includes Snowflake best practice designs, implementation patterns, and script setup for the Snowflake platform.

Best Practices consultation

The Snowflake Best Practices package helps you quickly understand Snowflake best practices to implement Snowflake correctly the first time, shorten your time-to-value, and gain valuable insight that you can leverage after the engagement. Through knowledge transfer and interactive discussions, the Snowflake solutions architect will help the customer team make key decisions regarding security, cost management and user management.

Snowflake 360

The Snowflake 360 offering brings a Snowflake solution architect alongside your team to provide recommendations regarding resource management, performance settings, data loading patterns and resource utilization. The Snowflake 360 offering includes knowledge transfer, analysis and change recommendations designed to help your organization maximize your Snowflake implementation.

Custom optimization services

Snowflake's professional services team offers optimization services to assist in the identification, prioritization and resolution of cost- or performance-related issues. Whether slow-performing critical queries or excessive credit consumption, these services provide targeted analyses, mentoring and specific recommendations for optimization.

Resident Solutions Architect

For organizations looking for a dedicated technical advisor on a partial or full-time basis, Snowflake can provide Resident Solutions Architects with expertise in areas that include cost optimization.

CHECKLIST OF QUICK ACTIONS AND BEGINNER-LEVEL BEST PRACTICES TO IMPLEMENT TODAY

- Enable auto-suspend and auto-resume
- Set up resource monitors
- Set up an account-level budget
- Start with a small warehouse
- Track usage
- Use UI-based cost exploration
- Choose the appropriate warehouse type and size
- Consider using Snowpark-optimized warehouses for memory-intensive workloads

ABOUT SNOWFLAKE

Snowflake enables every organization to mobilize their data with Snowflake's Data Cloud. Customers use the Data Cloud to unite siloed data, discover and securely share data, and execute diverse artificial intelligence (AI) / machine learning (ML) and analytic workloads. Wherever data or users live, Snowflake delivers a single data experience that spans multiple clouds and geographies. Thousands of customers across many industries, including 639 of the 2023 Forbes Global 2000 (G2K) as of July 31, 2023, use the Snowflake Data Cloud to power their businesses.

Learn more at **snowflake.com**