

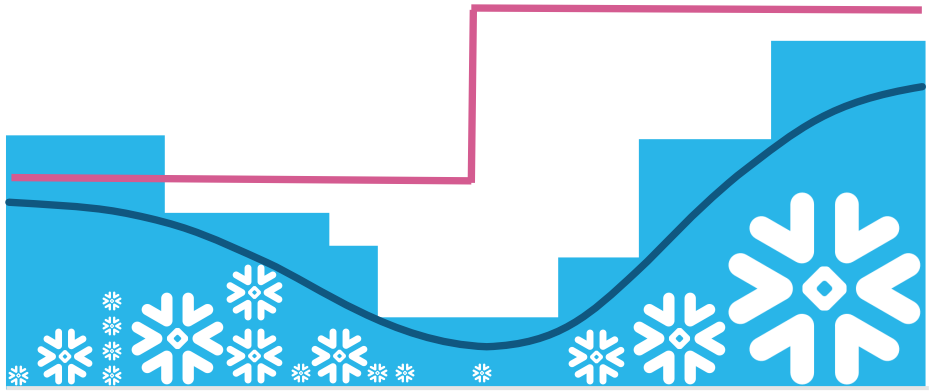
Optimising Admin

Manage Costs And Optimize Resources on Snowflake

© 2023 Snowflake Inc. All Rights Reserved

REAL WORLD USAGE IS VARIABLE

Only pay for what you use



Scale compute up and down, transparently and automatically

Reduced need for capacity planning, flexibility to make decisions on the fly

Pay for only what you use with no overprovisioning, eliminate overbuy

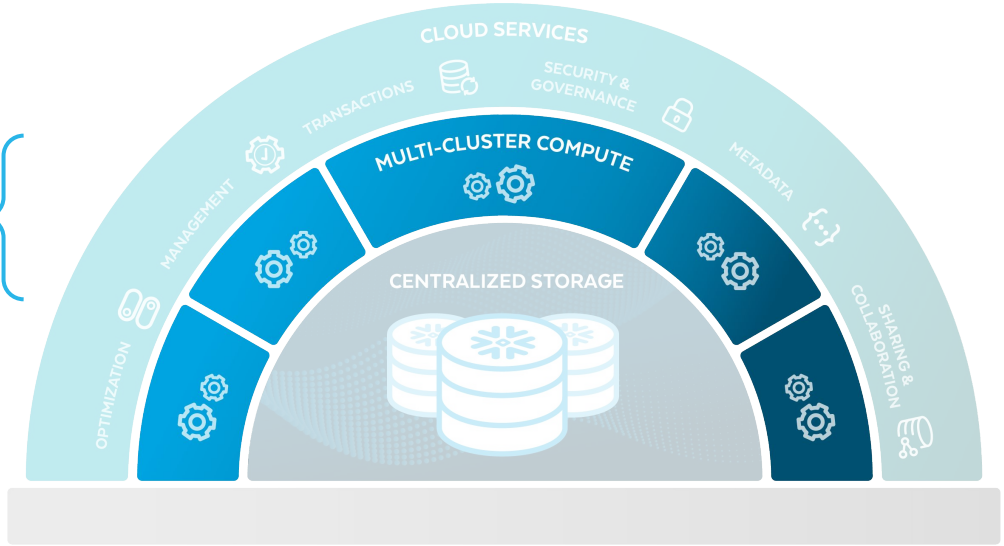
© 2022 Snowflake Inc. All Rights Reserved

Understanding Cost

© 2022 Snowflake Inc. All Rights Reserved

COMPONENTS OF COST ON SNOWFLAKE

COMPUTE



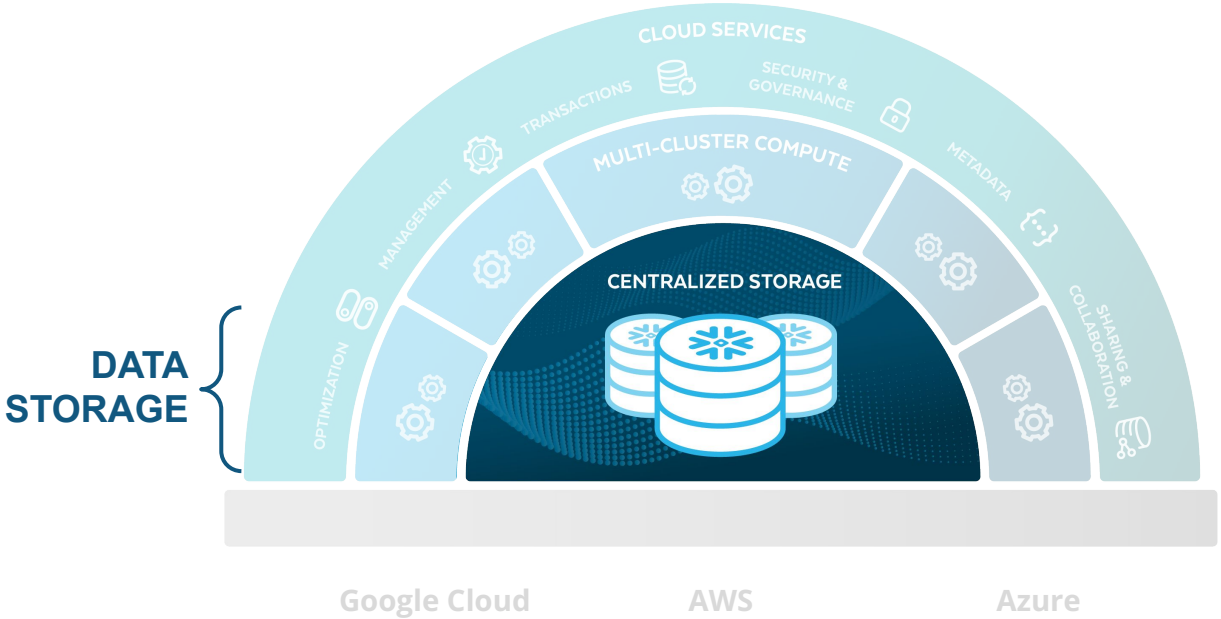
ON-DEMAND VIRTUAL WAREHOUSES

- Elastic (turn on instantly)
- Scale up (bigger)
- Scale out (more)
- Suspend
- Per second billing

Virtual Warehouses account for most of an accounts credit consumption.

© 2022 Snowflake Inc. All Rights Reserved

COMPONENTS OF COST ON SNOWFLAKE



ON-DEMAND VIRTUAL WAREHOUSES

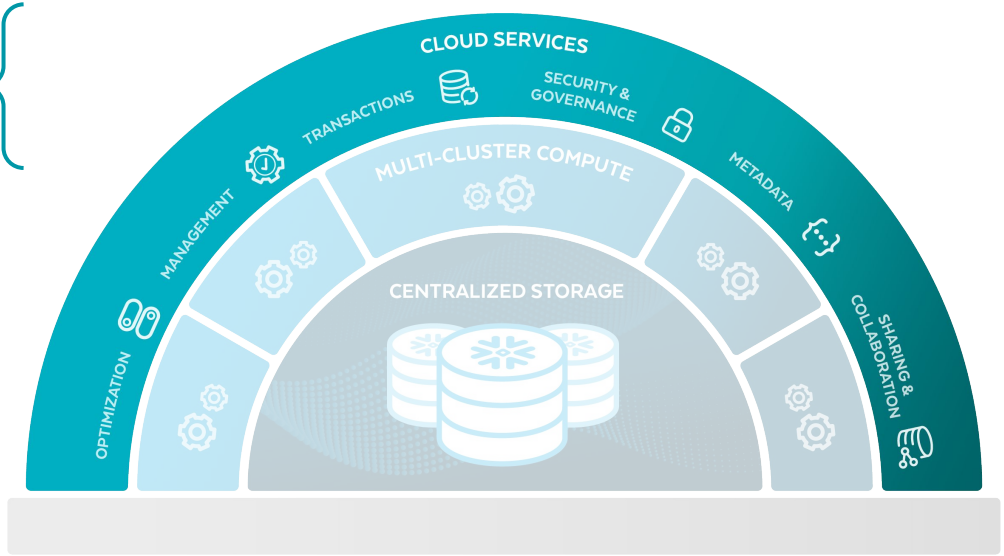
- Pass through pricing
- Up to 7x compression
- Native support for semi structured data
- Zero-copy cloning
- Time travel

Storage is usually a very small fraction of the overall bill

© 2022 Snowflake Inc. All Rights Reserved

COMPONENTS OF COST ON SNOWFLAKE

CLOUD SERVICES



COMPUTE THAT DOESN'T RUN ON A VIRTUAL WAREHOUSE

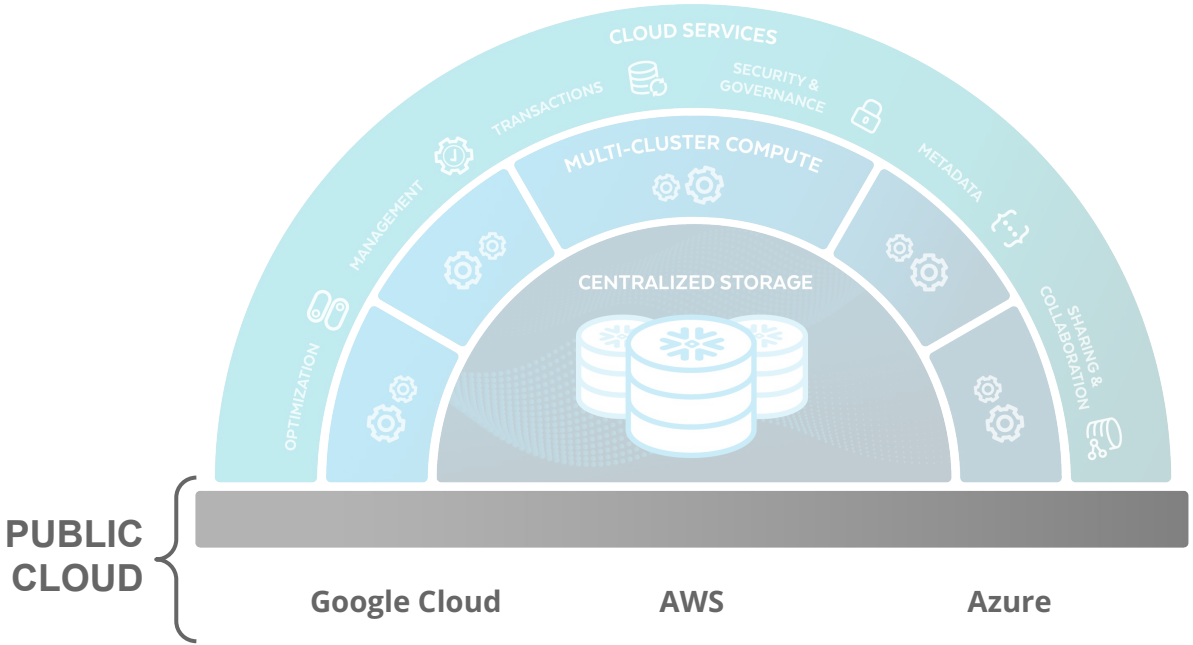
- SHOW commands
- Query result cache

SERVERLESS

- Search Optimization Service
- Snowpipe
- Auto-clustering
- Materialized view maintenance
- Cross-cloud replication

© 2022 Snowflake Inc. All Rights Reserved

COMPONENTS OF COST ON SNOWFLAKE



CLOUD PROVIDERS

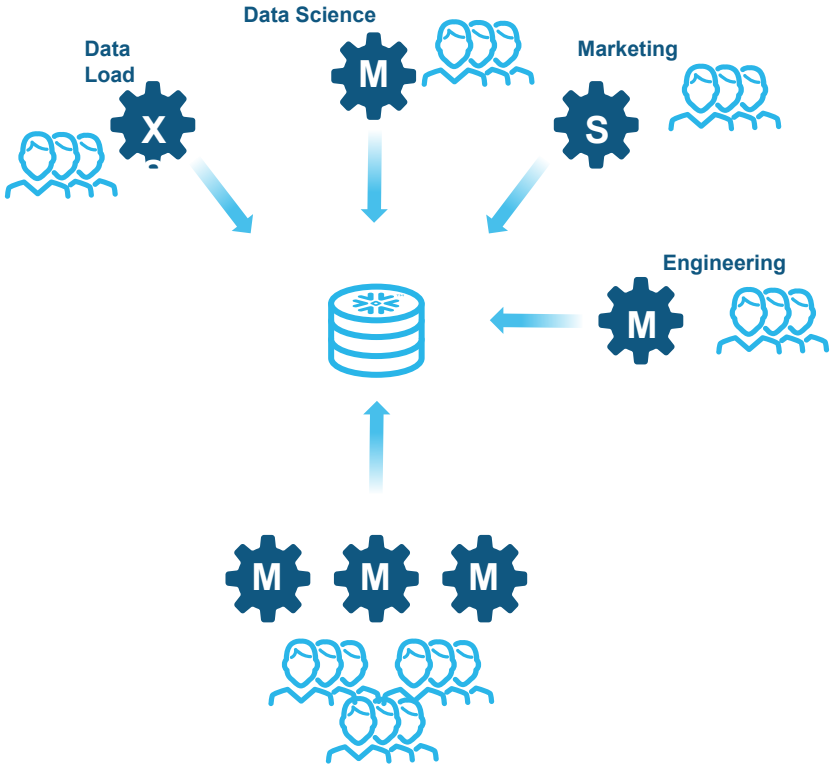
- Per credit cost implications
- Egress and data transfer

© 2022 Snowflake Inc. All Rights Reserved

Cost Governance

© 2022 Snowflake Inc. All Rights Reserved

Pattern 1: Flexibility Focused



Flexible allocation of resources

Each team has the freedom to create and manage their own dedicated warehouses.

Default warehouse cost controls

Warehouse size offers predictable per second pricing. Auto suspend/resume turned on by default.

Scale up/out

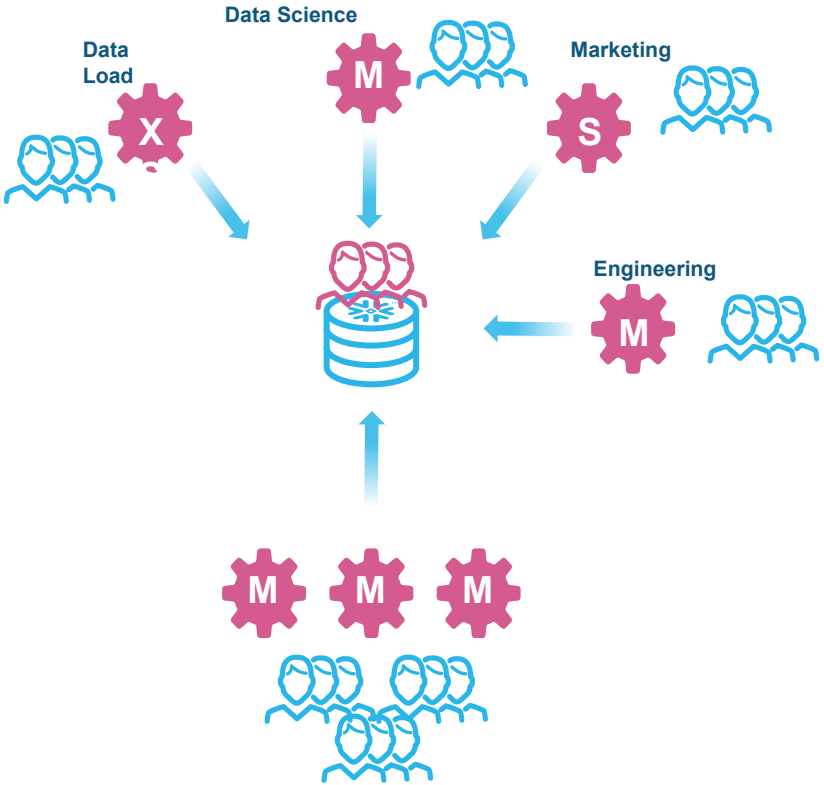
Scale up to handle larger, more complex queries or out to serve more users concurrently.

Usage Reporting

Out of the box usage dashboards on the most expensive warehouses and queries.

© 2022 Snowflake Inc. All Rights Reserved

Pattern 2: Efficiency Focused



Controlled allocation of resources

A central team reviews and approves when warehouses can be created.

Restricted scaling permissions

The central team manages scaling up/out operations to ensure predictability of spend.

Chargeback reporting

Attribute cost back to individual teams (via tagging) with periodic reviews.

Monitoring and alerting

Set up warehouse alerts and spend limits (via resource monitors) to guard against overspending.

© 2022 Snowflake Inc. All Rights Reserved

COST GOVERNANCE IN SNOWFLAKE

Visibility



Understand, attribute and monitor spend

Control



Limit and control spend

Optimization



Identify and optimize inefficient spend

© 2022 Snowflake Inc. All Rights Reserved

COST GOVERNANCE IN SNOWFLAKE

Visibility

Understand
- Usage Dashboards
- Usage Data Shares

Attribute
- Object/Query Tagging

Monitor
- Resource Monitors

Control

- Warehouse Size & Limits
- Auto suspend/resume
- Resource Monitors

Optimization

- Platform improvements
- Query profile and history
- Warehouse load and events history
- Access history

COST GOVERNANCE IN SNOWFLAKE



Visibility

Understand

- Usage Dashboards
- Usage Data Shares

Attribute

- Object/Query Tagging

Monitor

- Resource Monitors



Control

- Warehouse Size & Limits
- Auto suspend/resume
- Resource Monitors



Optimization

- Platform improvements
- Query profile and history
- Warehouse load and events history
- Access history

Top Tips

© 2022 Snowflake Inc. All Rights Reserved

#1

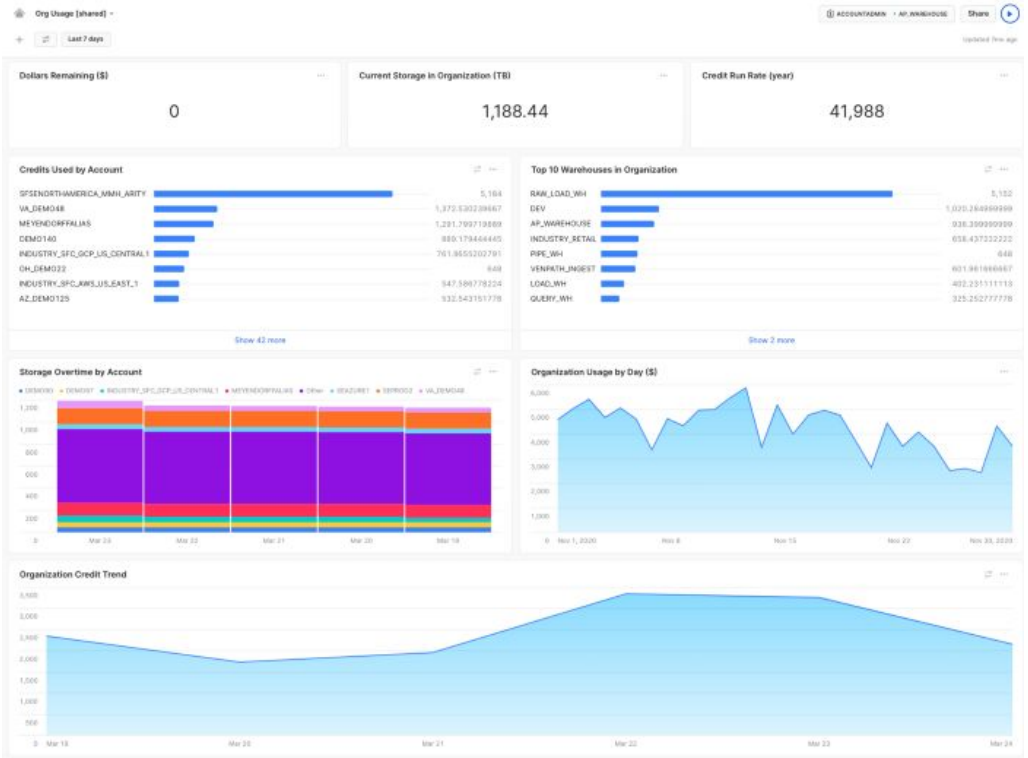
VIEW USAGE ACROSS BUSINESS

ORGANIZATION

Architect an overall account structure for cost governance

Cost vs Security

- Get overall reporting on spend and drill down to usage and cost data for specific resources.
- Get granular, analytics-ready usage data to build custom reports or dashboards.



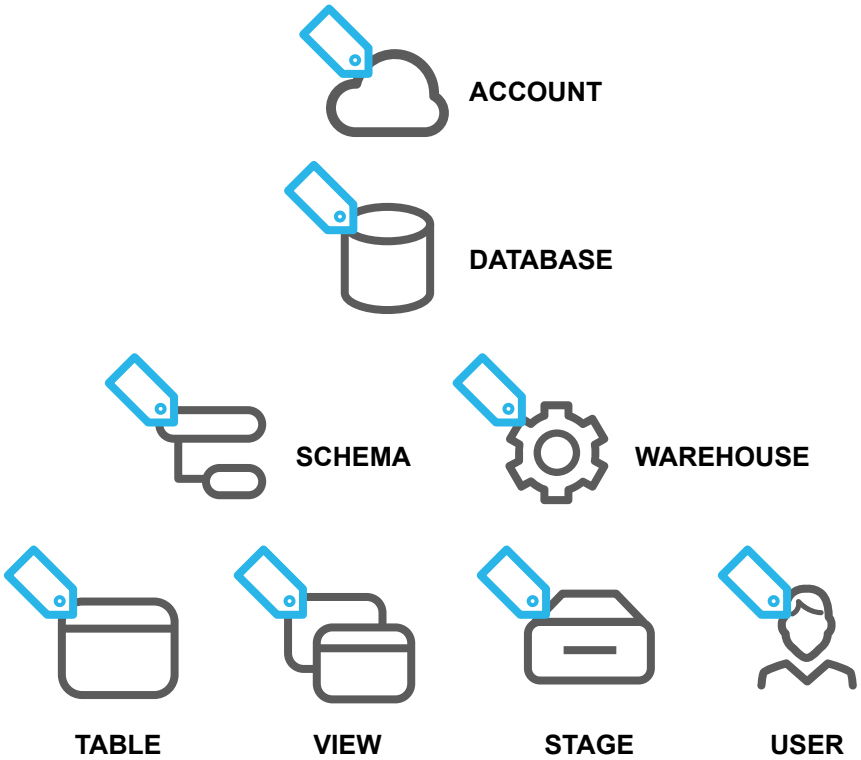
© 2022 Snowflake Inc. All Rights Reserved

Dashboard from "Monitoring Your Snowflake Organization With Snowsight" by Ashish Patel

#2

Object Tagging

Track sensitive data and compute objects



Track sensitive and PII data

Track resource usage for cost visibility and attribution

Flexible privilege management options

© 2023 Snowflake Inc. All Rights Reserved

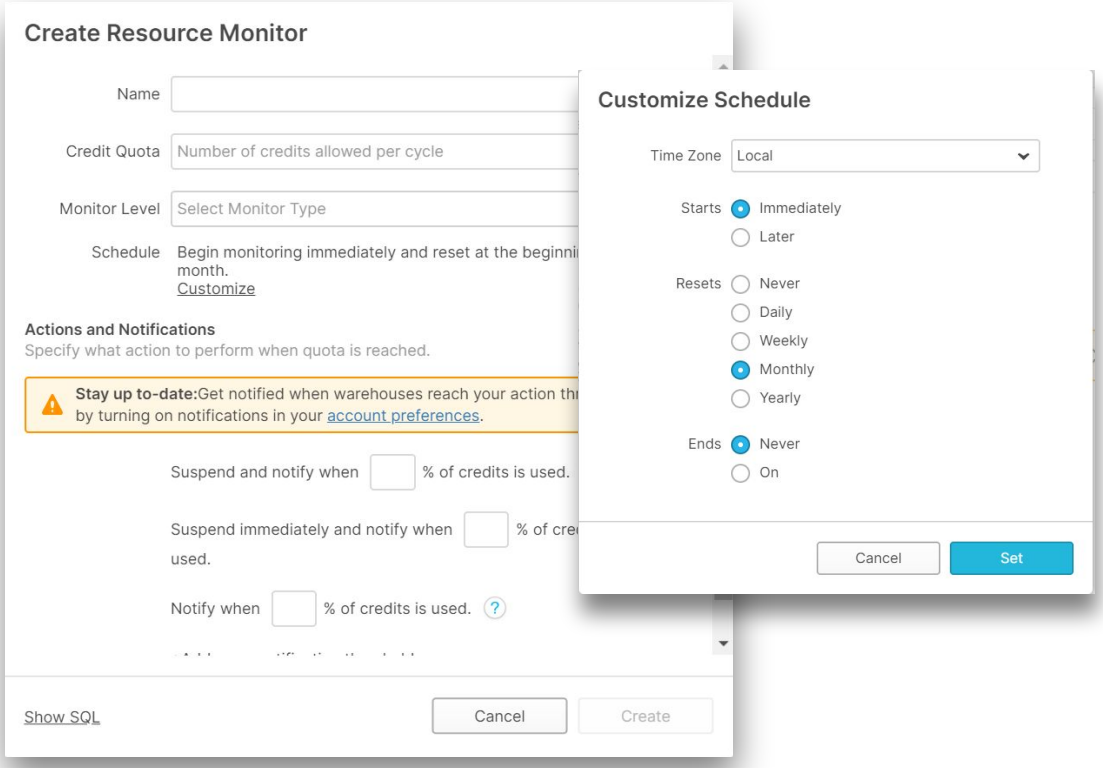
#3

APPLY RESOURCE MONITORS

RESOURCE MONITORS

Create alerts and hard limits so that you don't overspend or accidentally use up credits

- Create quotas for individual warehouses to limit spend
- Get alerted when warehouses reach a limit or are suspended
- Set up custom scheduled to track usage



© 2022 Snowflake Inc. All Rights Reserved

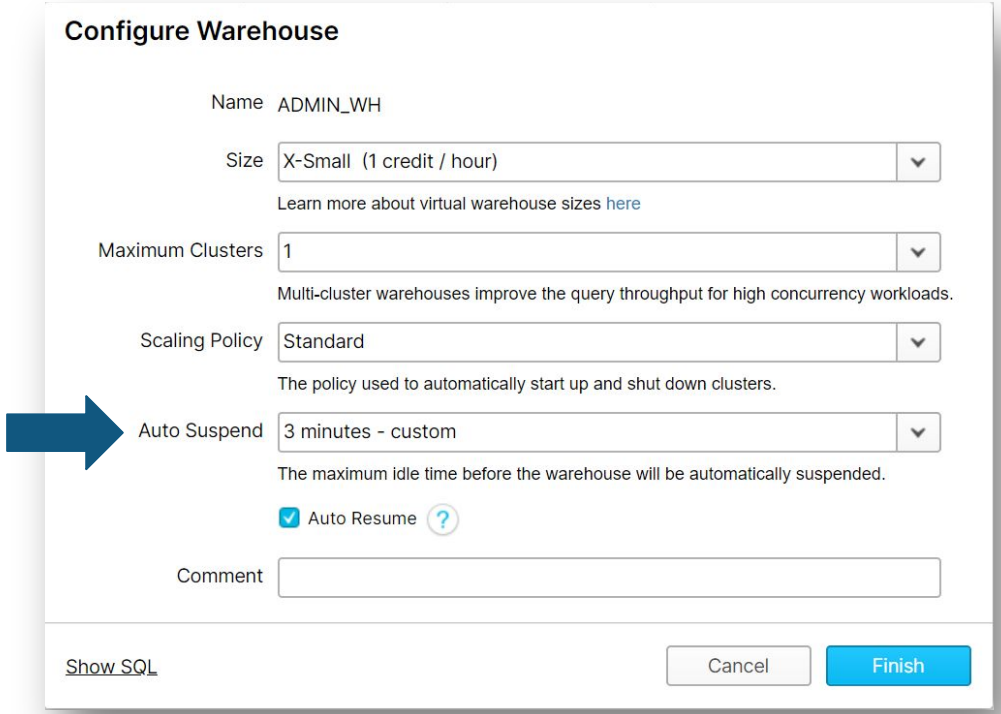
#4

ENABLE AUTO-SUSPEND

AUTO SUSPEND POLICIES

Turning virtual warehouses off quickly after they go idle saves customers money.

- Use cases that take advantage of query caching should be left on longer
- ELT and data loading that don't use query caching should be turned off immediately



© 2022 Snowflake Inc. All Rights Reserved

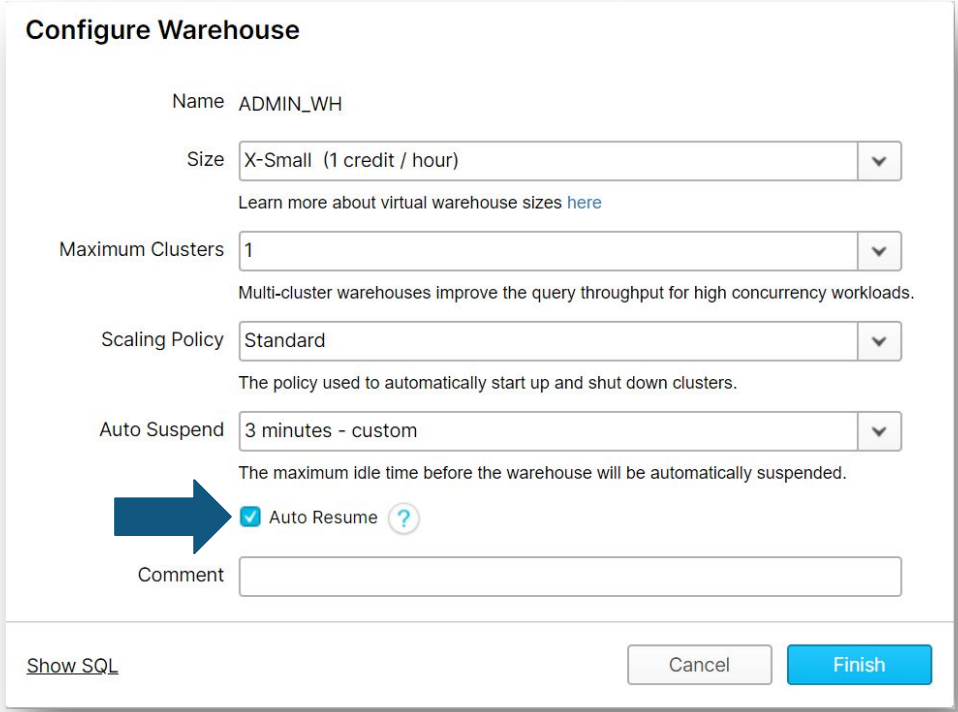
#5

ENABLE AUTO-RESUME

AUTO RESUME POLICIES

With an auto-suspended warehouse auto-resume turns it back on when a user or system runs a query against the warehouse

- Contracted up and down or ETL processes having incomplete data exposure time could be a reason not to use auto_resume
- Auto_resume brings the warehouse up on the first query to the warehouse



© 2022 Snowflake Inc. All Rights Reserved

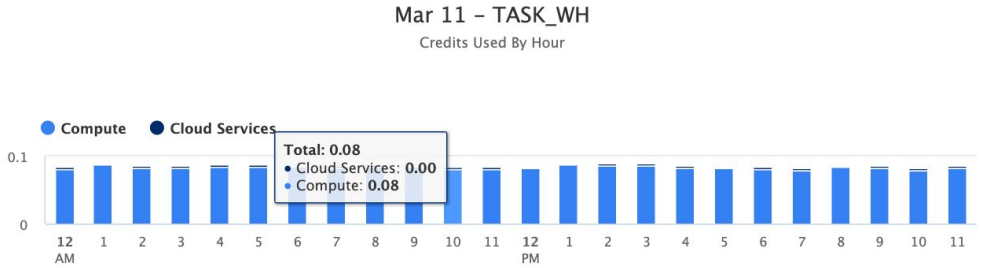
#6

WAREHOUSES APPROACHING CLOUD SERVICES BILLING

CLOUD SERVICES BILLING

Warehouses with heavy meta-data queries my drive to the account billing threshold

- Check which warehouses drive potential spend here




```
WITH
cloudServices AS (SELECT
     WAREHOUSE_NAME, MONTH(START_TIME) AS MONTH,
     SUM(CREDITS_USED_CLOUD_SERVICES) AS CLOUD_SERVICES_CREDITS,
     COUNT(*) AS NO_QUERYS
     FROM "SNOWFLAKE"."ACCOUNT_USAGE"."QUERY_HISTORY"
     GROUP BY WAREHOUSE_NAME,MONTH
     ORDER BY WAREHOUSE_NAME,NO_QUERYS DESC),
warehouseMetering AS (SELECT
     WAREHOUSE_NAME, MONTH(START_TIME) AS MONTH,
     SUM(CREDITS_USED) AS CREDITS_FOR_MONTH
     FROM "SNOWFLAKE"."ACCOUNT_USAGE"."WAREHOUSE_METERING_HISTORY"
     GROUP BY WAREHOUSE_NAME,MONTH
     ORDER BY WAREHOUSE_NAME,CREDITS_FOR_MONTH DESC)
```

© 2022 Snowflake Inc. All Rights Reserved

#7

SET ACCOUNT & WAREHOUSE STATEMENT TIMEOUTS

ACCOUNT TIMEOUTS

Timeouts on the account will stop all long running queries that reach the threshold or scope to the warehouse for select vs etl work

- Consider user experience along with optimizations
- It's a good idea to report on queries that were timed out and send notices to users

```
STATEMENT_QUEUED_TIMEOUT_IN_SECONDS
      Scoped to Account, Warehouse, User or Session

STATEMENT_TIMEOUT_IN_SECONDS
      Scoped to Account, Warehouse, User or Session



ALTER WAREHOUSE LOAD_WH SET STATEMENT_TIMEOUT_IN_SECONDS =
3600;
SHOW PARAMETERS IN WAREHOUSE LOAD_WH;
```

© 2022 Snowflake Inc. All Rights Reserved

#8

PURGE DORMANT USERS

DORMANT USERS

Users that do not login to the system pose risk

- Stale users
- Users that never logged in

```
--Never Logged In Users
SHOW USERS;
SELECT * FROM TABLE(RESULT_SCAN(LAST_QUERY_ID()))
WHERE "last_success_login" IS NULL
AND DATEDIFF('Day',"created_on",CURRENT_DATE) > 30;

--Stale Users
SHOW USERS;
SELECT * FROM TABLE(RESULT_SCAN(LAST_QUERY_ID()))
WHERE DATEDIFF('Day',"last_success_login",CURRENT_DATE) >
30;
```

© 2022 Snowflake Inc. All Rights Reserved

#9

DROP UNUSED TABLES

UNUSED TABLES

If you are not using it why are you paying to store it

- 2.0 use the table access history

```
--DML from the Information Schema to identify Table sizes and Last
Updated Timestamps
SELECT TABLE_CATALOG || '.' || TABLE_SCHEMA || '.' || TABLE_NAME AS
TABLE_PATH,
    TABLE_NAME, TABLE_SCHEMA AS SCHEMA,
    TABLE_CATALOG AS DATABASE, BYTES,
    TO_NUMBER(BYTES / POWER(1024,3),10,2) AS GB,
    LAST_ALTERED AS LAST_USE,
    DATEDIFF('Day',LAST_USE,CURRENT_DATE) AS DAYS_SINCE_LAST_USE
FROM INFORMATION_SCHEMA.TABLES
WHERE DAYS_SINCE_LAST_USE > 90 --Use your Days Threshold
ORDER BY BYTES DESC;

-- Last DML on Object
SELECT (SYSTEM$LAST_CHANGE_COMMIT_TIME(
'DATABASE.SCHEMA.TABLE_NAME')/1000)::TIMESTAMP_NTZ;

-- Queries on Object in Last NN Days
SELECT COUNT(*) FROM "SNOWFLAKE"."ACCOUNT_USAGE"."QUERY_HISTORY"
WHERE CONTAINS(UPPER(QUERY_TEXT),'TABLE_NAME')
AND DATEDIFF('Day',START_TIME,CURRENT_DATE) < 90;

-- Last Query on Object in Last NN Days
SET TABLE_NAME = 'MY_TABLE';

SELECT START_TIME, QUERY_ID, QUERY_TEXT
FROM "SNOWFLAKE"."ACCOUNT_USAGE"."QUERY_HISTORY"
WHERE CONTAINS(UPPER(QUERY_TEXT),$TABLE_NAME)
```

© 2022 Snowflake Inc. All Rights Reserved

#10

IDENTIFY DEVIANT PERFORMANCE

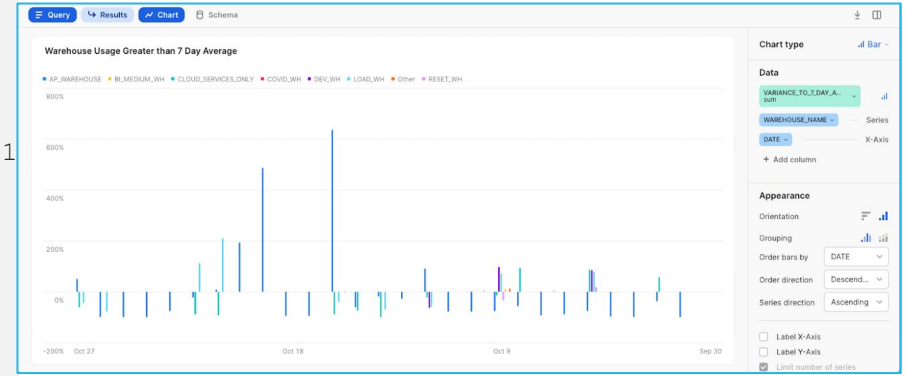
Setup Warehouse Usage Greater than 7 Day Average

```
SELECT
   WAREHOUSE_NAME,
   DATE(START_TIME) AS DATE,
   SUM(CREDITS_USED) AS CREDITS_USED,
   AVG(SUM(CREDITS_USED)) OVER (
       PARTITION BY WAREHOUSE_NAME
       ORDER BY
           DATE ROWS 7 PRECEDING
   ) AS CREDITS_USED_7_DAY_AVG,

(TO_NUMERIC(SUM(CREDITS_USED)/CREDITS_USED_7_DAY_AVG*100,1
0,2)-100)::STRING || '%' AS VARIANCE_TO_7_DAY_AVERAGE
FROM

"SNOWFLAKE"."ACCOUNT_USAGE"."WAREHOUSE_METERING_HISTORY"
WHERE
   START_TIME = :daterange
GROUP BY
   DATE,
   WAREHOUSE_NAME
ORDER BY
   DATE DESC;
```

Visualize Credit Usage Greater than a 7 Day Average in Snowsight



© 2022 Snowflake Inc. All Rights Reserved

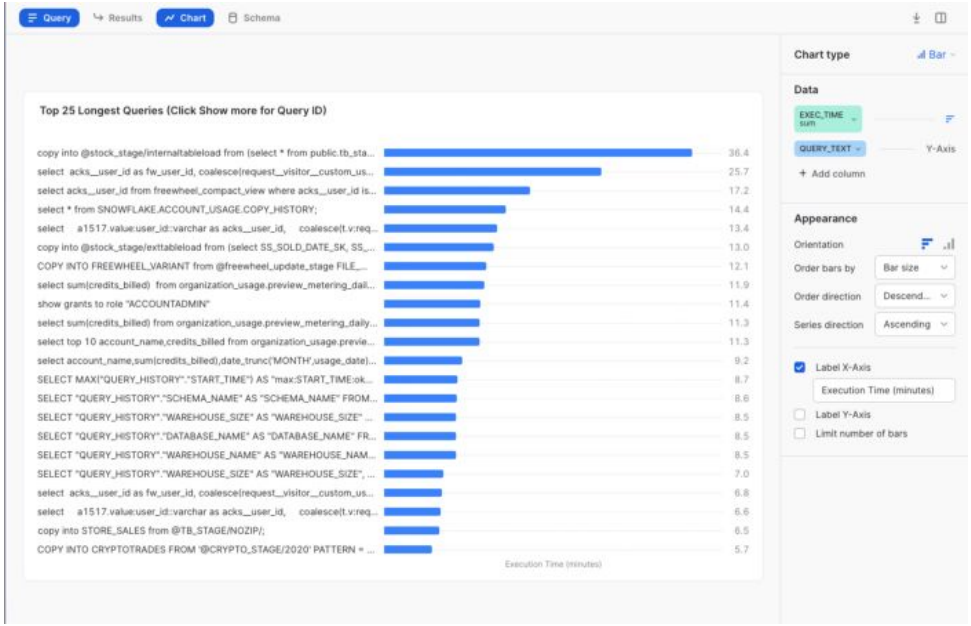
Dashboard from Monitoring Snowflake with Snowsight by Ashish Patel

#11 IDENTIFY LONGEST RUNNING QUERIES

Top 25 Longest Queries

```
select
    query_id,
    query_text,
    (execution_time / 60000) as exec_time
from
    account_usage.query_history
where
    execution_status = 'SUCCESS'
order by
    execution_time desc
limit
    25
```

Dive into specific queries with Query Profile



© 2022 Snowflake Inc. All Rights Reserved

Dashboard from Monitoring Snowflake with Snowsight by Ashish Patel

#12

BONUS: QUERIES THAT SPILL TO DISK OR BLOB

Top Disk Spillers

```
select top 25
    hash(query_text) as query_hash
    ,query_text
    ,count(query_id) as query_count
    ,avg(total_elapsed_time) as
avg_total_elapsed_time
    ,sum(bytes_spilled_to_local_storage) /
1024 / 1024 / 1024 as sum_gb_spilled_to_disk
    ,sum(bytes_spilled_to_remote_storage) /
1024 / 1024 / 1024 as sum_gb_spilled_to_blob
from
    snowflake.account_usage.query_history
where
    start_time::date >=
dateadd('day',$date_delta,current_date)
group by
    query_hash,
    query_text
having
    query_count > 5
order by
    sum_gb_spilled_to_disk;
```

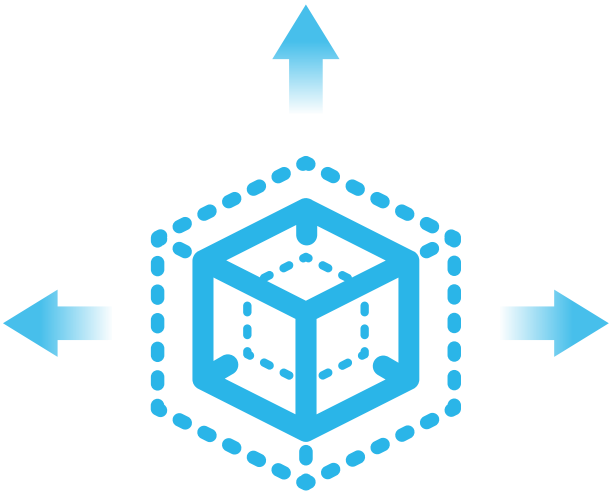
Top BLOB Spillers

```
select top 25
    hash(query_text) as query_hash
    ,query_text
    ,count(query_id) as query_count
    ,avg(total_elapsed_time) as
avg_total_elapsed_time
    ,sum(bytes_spilled_to_local_storage) /
1024 / 1024 / 1024 as sum_gb_spilled_to_disk
    ,sum(bytes_spilled_to_remote_storage) /
1024 / 1024 / 1024 as sum_gb_spilled_to_blob
from
    snowflake.account_usage.query_history
where
    start_time::date >=
dateadd('day',$date_delta,current_date)
group by
    query_hash,
    query_text
having
    query_count > 5
order by
    sum_gb_spilled_to_blob;
```

Scale vs Cost

© 2023 Snowflake Inc. All Rights Reserved

3 Dimensions of Scaling



ACROSS

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

© 2023 Snowflake Inc. All Rights Reserved

3 Dimensions of Scaling

UP

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



ACROSS

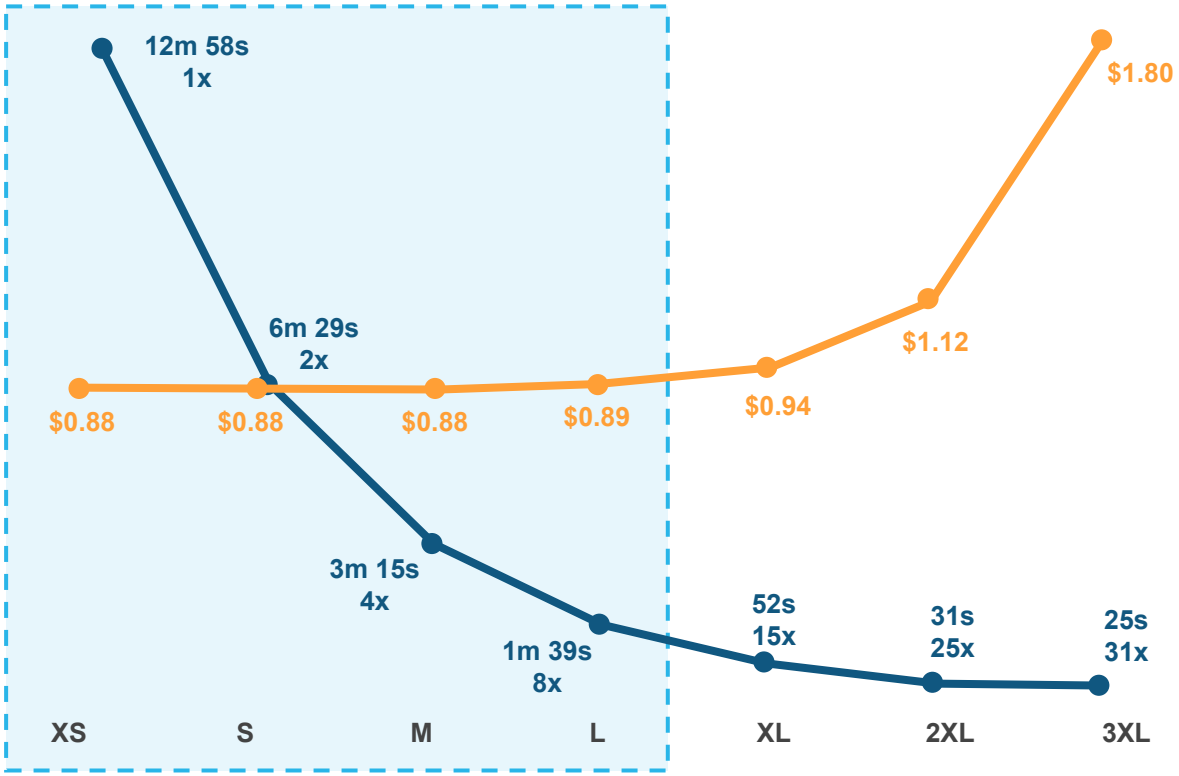
- Many competing workloads
- Resource contention
- Isolate on separate warehouses

© 2023 Snowflake Inc. All Rights Reserved

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

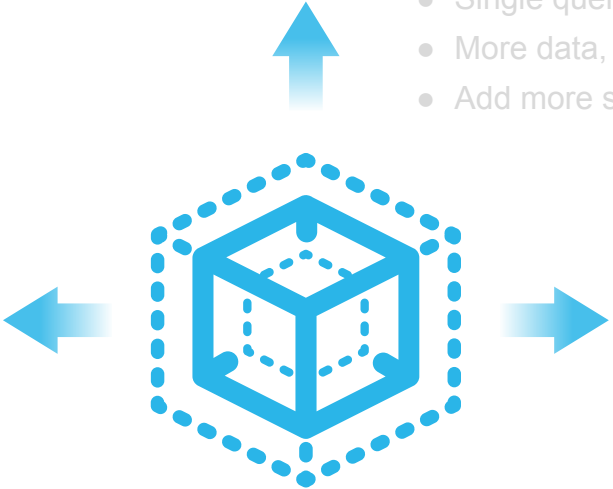


© 2023 Snowflake Inc. All Rights Reserved

3 Dimensions of Scaling

UP

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



ACROSS

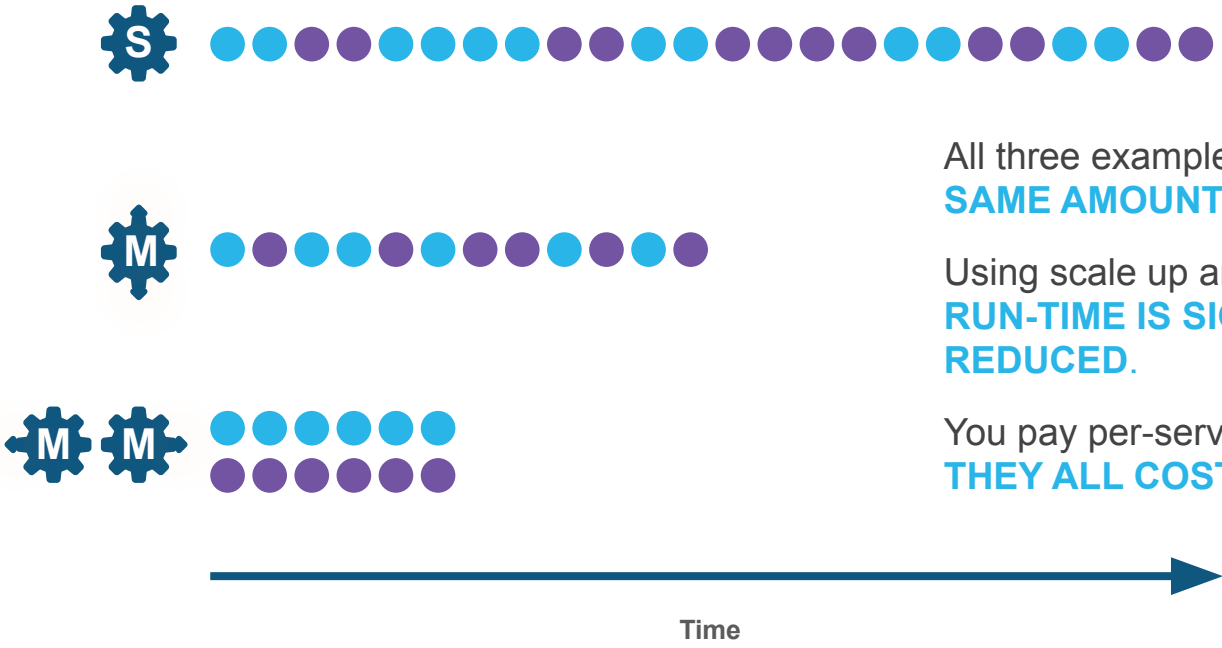
- Many competing workloads
- Resource contention
- Isolate on separate warehouses

Out

- More users
- More queries simultaneously
- Spin up more clusters

© 2023 Snowflake Inc. All Rights Reserved

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**.

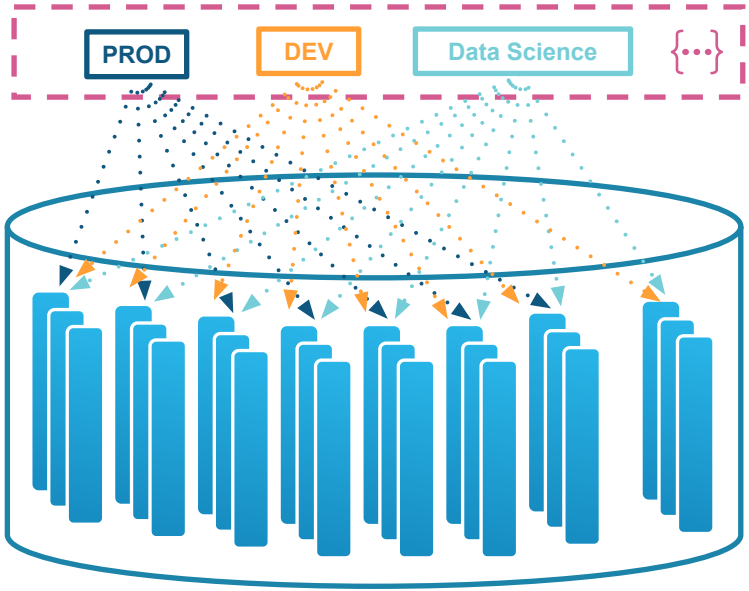
You pay per-server, per-second so **THEY ALL COST THE SAME**.

© 2023 Snowflake Inc. All Rights Reserved

OPTIMIZED STORAGE

© 2023 Snowflake Inc. All Rights Reserved

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

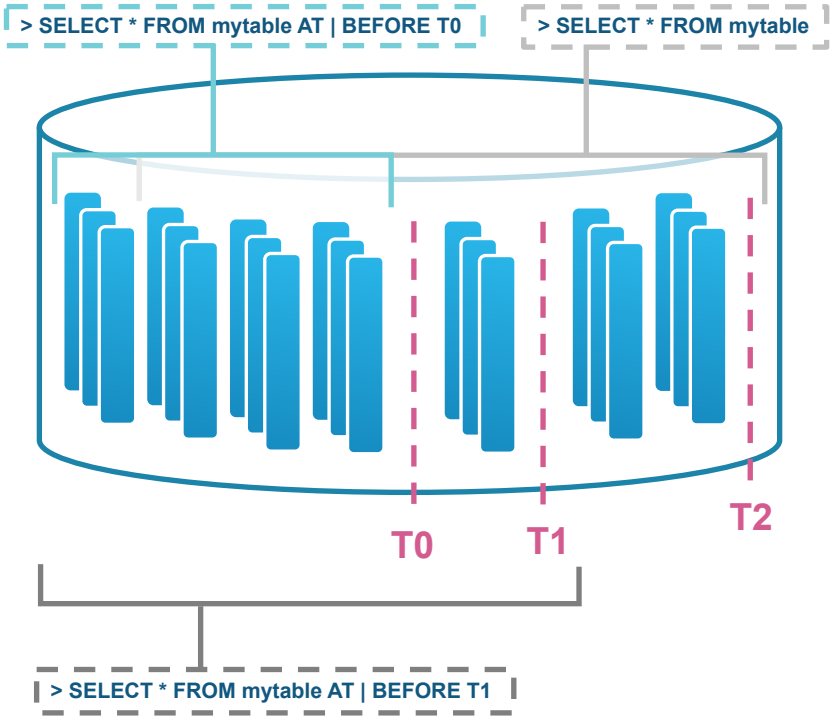
Simply “point” to the same files

Consumes zero additional storage

Changes to either DB are isolated

© 2023 Snowflake Inc. All Rights Reserved

Time Travel



Cost Saving:

TEMPORARY

TRANSIENT

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

© 2023 Snowflake Inc. All Rights Reserved

Parquet Performance Improvements

External Tables are now more flexible and faster with Parquet



Vectorized Parquet Scanner

We have developed a mechanism to work with Parquet files in external stages and tables.

Performance becomes even better when using **external Iceberg tables**.

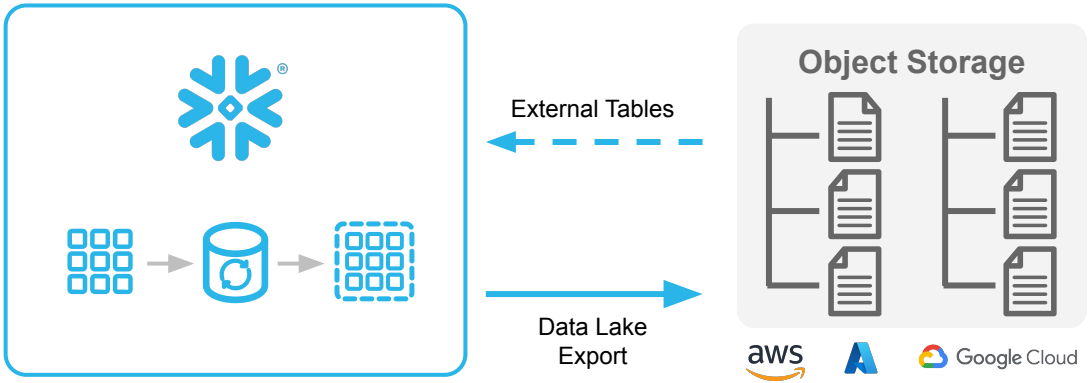
© 2023 Snowflake Inc. All Rights Reserved

ELASTIC PERFORMANCE ENGINE

© 2023 Snowflake Inc. All Rights Reserved

Data Lake Export

Export data from Snowflake tables to files in partitioned folders in object storage



Export large parquet files with 128MB row groups

Export with Hive-style partitioning

```
COPY INTO @processed_data
FROM processed_table
PARTITION BY('dt='|| date_col)
   file_format = (type = parquet)
   max_file_size = 512000000;
```

© 2023 Snowflake Inc. All Rights Reserved

Search Optimization Service

Adoption across a diverse range of workloads

Case Studies

143x

Improvement (5:16 minutes to 2.2 seconds) in a lookup for financial transactions and account lookups for a given user

Fortune 500 Financial Services Company

120x

Improvement (37:14 minutes to 18.54 seconds) in a lookup of all the logs and data associated with one of many attributes

HYAS

5x

Improvement (1:22 minutes minutes to 16 seconds) in lookups of specific user accounts and activity

Fortune 500 Media Company

4x

Improvement (8.91 seconds minutes to 2.3 seconds) looking for a specific coupon usage on an item

Fortune 500 Retail Company

Other Common Use Cases

DATA SHARING
Property included on table in data sharing

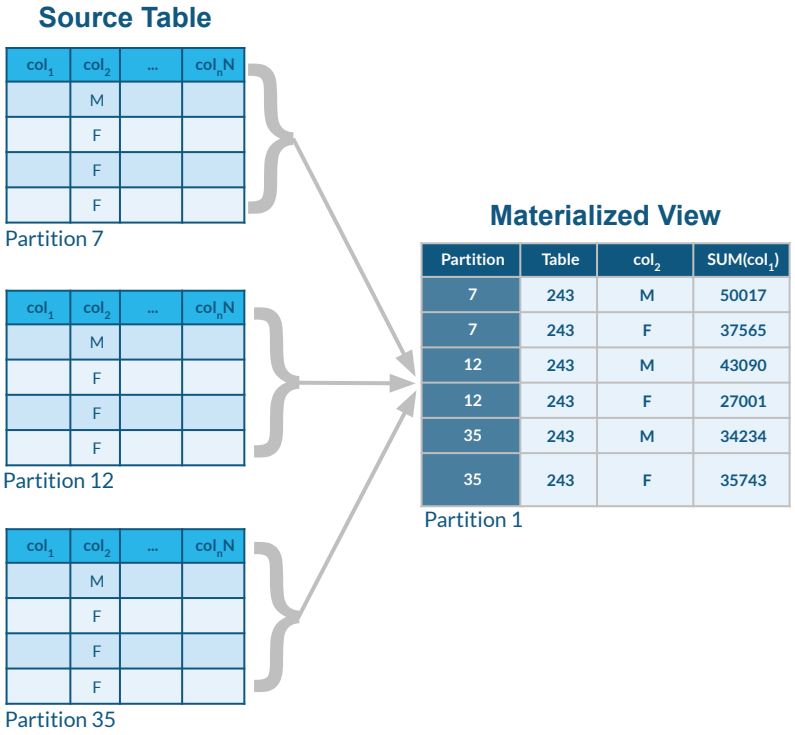
SECURITY
Property effective in log analysis

GDPR
Quickly delete users

CANARY QUERIES
Use in your ad-hoc analysis

© 2023 Snowflake Inc. All Rights Reserved

Materialized Views



What is it?

Store frequently used projections and aggregations to avoid wasting time and money re-computing

Results from MV queries guaranteed to be up-to-date

Value

Run faster queries, spend fewer compute credits

Asynchronous, incremental maintenance

No impact on DML performance

MV on External Tables

- Fast, governed access to external storage
- Benefit from Snowflake performance while maintaining source of truth outside

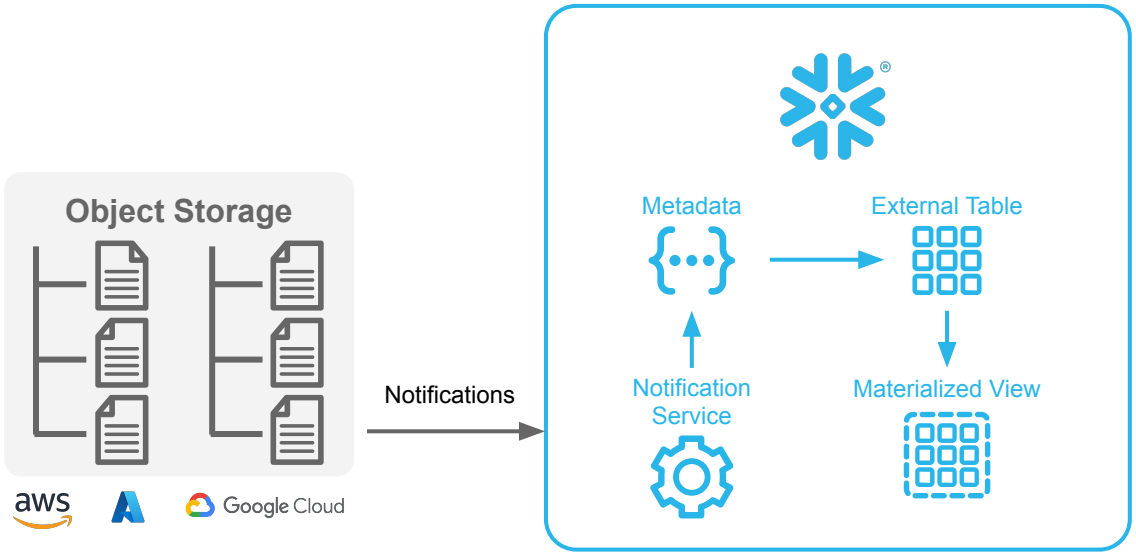
Cost

Serverless refresh, additional storage

© 2023 Snowflake Inc. All Rights Reserved

Materialized Views on External Tables

Faster queries, fewer compute credits



© 2023 Snowflake Inc. All Rights Reserved

Query Acceleration Service

Elastic scale out of compute resources without changing size of virtual warehouse

What It Does

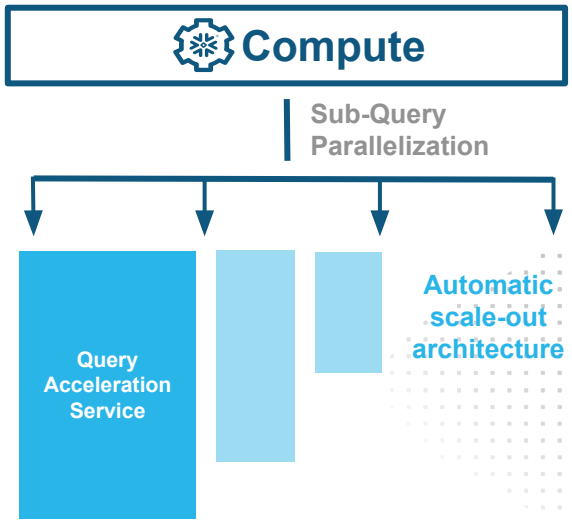
Improved performance with existing compute resources for spikes in workload compute needs.

How It Works

Enables vertical scaling on a query by creating sub-query parallelization.

Performance Improvements

Reduce execution times



© 2023 Snowflake Inc. All Rights Reserved

Build Efficient Pipelines With Serverless Tasks

Before

```
create task t1
  schedule = '1 minute'
  warehouse = 'transform_wh’
  AS …
```

After

```
create task t1
  schedule = '1 minute'
  warehouse = 'transform_wh’
  AS … z
```

- No need to specify a warehouse: size selection, packing, resume/suspend
- Simpler and more cost efficient
- Create new task as serverless or alter existing task

© 2023 Snowflake Inc. All Rights Reserved

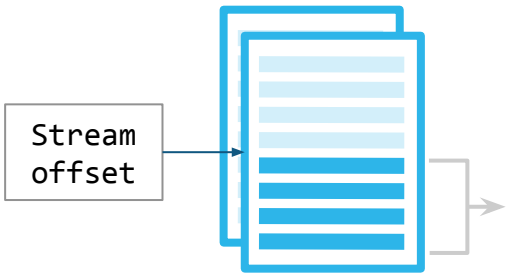
Incremental Processing & Orchestration

Streams on Views

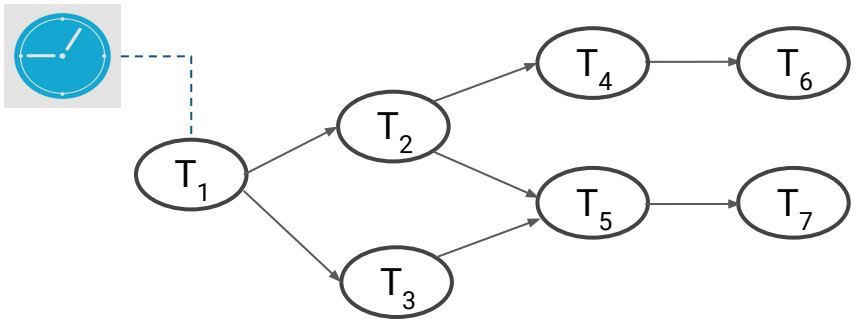
Perf improvements (ROW_VERSION, bitsets)

Replication

Streams



Tasks



DAG
Pipeline level history
Direct execution
Error notifications (starting with AWS)
Replication
UI: operations & monitoring

© 2023 Snowflake Inc. All Rights Reserved

Cost Governance in Snowflake



Visibility

Understand, attribute, and monitor spend



Control

Limit and control spend

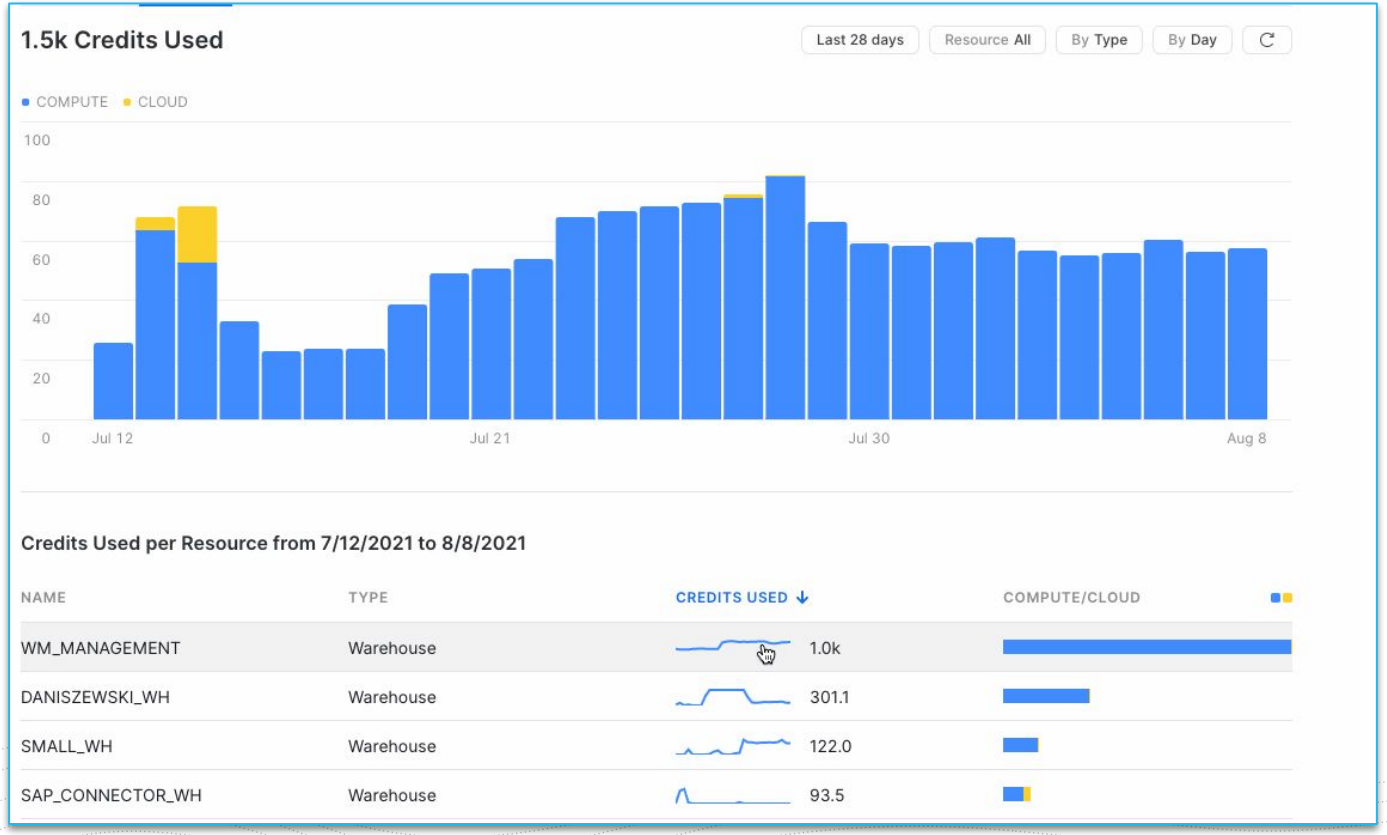


Optimization

Identify and optimize inefficient spend

© 2023 Snowflake Inc. All Rights Reserved

Pre-Built Usage Visualizations



© 2023 Snowflake Inc. All Rights Reserved

Q&A

Please enter any questions in the chat window.

© 2022 Snowflake Inc. All Rights Reserved



THANK YOU

© 2022 Snowflake Inc. All Rights Reserved