**Snowflake Architecture**

What’s unique with snowflake architecture?

How snowflake architecture is different from traditional data bases(RDBMS like SQL Server, Oracle, My SQL) ?

What’s your understanding of Snowflake architecture?

Tell me about snowflake architecture?

It's a hybrid of shared-disk and shared-nothing architecture

Basically its Multi cluster shared data architecture. Storage and compute are separate and can be scaled independently to any extent and also it charges separately for storage and compute

(i.e. storage and compute are decoupled)

It’s 3 layer architecture

Storage layer => that stores data in hybrid columnar format and saved in cloud

Query Processing => Virtual Warehouse to run queries/for compute resources (muscle)

Cloud Services => brain of snowflake architecture coordinates between storage and query processing layers

/\*Data is stored in the cloud storage and works as a shared-disk model

(i.e. centralized data store like RDBMS)

Compute it will take advantage of performance and scale-out benefits of shared nothing architecture

(i.e. compute is distributed like hadoop) \*/

Why to go with Snowflake instead of other Cloud Data warehouses like AWS Redshift, Azure Synapse, Google Big query?

Snowflake is available on all cloud platforms like AWS, Azure and GCP (Multi cloud). Other cloud Data warehouses were restricted to that one cloud example: Redshift will be available in only AWS, Big Query will be available in GCP.

Unique features like Data Market place, Zero Copy Clone, Time travel, Data sharing, Streams for CDC

Separate Storage and compute costs, Zero Administration/Maintenance (partitioning, data backups, index maintenance, performance tuning will be taken care by snowflake itself).No software or hardware required to be installed or upgraded as snowflake is SaaS (Software as a service). Synapse and BigQuery are PaaS (Platform as a service)

How Snowflake charges for Storage and compute?

Snowflake Charges separately for Storage and separately for compute

**Virtual Warehouse(Query Processing)**

How is compute done in Snowflake?

Using virtual warehouses

What is Virtual Warehouse?

Cluster of compute nodes/machines

Cluster => group of nodes/machines connected together

Compute Resources => CPU + Memory+ Temporary storage

Different sizes of warehouses?

XS(1) will have one compute nodes/machines, S(2) will have 2 compute nodes, M(4), L(8), XL(16), 2XL(32), 3XL(64), 4XL(128) -- generally available

5XL(256) and 6 XL(512) are in preview state

What is the default Auto suspend time in a Warehouse?

10 minutes (By default, auto-suspend is enabled. Snowflake automatically suspends the warehouse if it is IDLE for the specified period of time)

What is Auto resume in Virtual Warehouse?

If the warehouse is in suspended state and if we set Auto Resume as true then as soon as a query is ran using this warehouse it will automatically get resumed

(i.e. from suspended state it becomes active)

How many Virtual Warehouses have you used in your project?

ETL team ETL\_S\_WH

ETL\_L\_WH

Reporting\_WH

Datascience\_WH

Testing\_WH

We were using Snowflake\_WH warehouse

Maximum Warehouse size that we used in our project is XL or 2XL

ETL tools => Informatica, Microsoft Onpremises ETL tool SSIS, ADF (azure data factory), Datastage

Reporting tools => Power BI (Microsoft’s reporting tool), Tableau

Reporting\_WH

Testing\_WH

Datascience\_WH

ML\_WH

Can we increase or decrease the warehouse size any time?

Yes on fly (instantly) it will get resized

Can we increase or decrease the warehouse size while warehouse is in suspended?

Yes we can increase/decrease warehouse size irrespective whether your warehouse is in running state or suspended state

Can we start or stop the warehouse any time?

Yes, suspend will stop it and resume will start it again

We were using Snowflake\_WH warehouse for our development

What’s the Maximum Warehouse size that you have used in your project XL?

Can we increase or decrease the warehouse size any time?

Yes on fly (instantly) it will get resized

What will happen to already running queries if we resize the warehouse?

Already running queries will still use the old warehouse size but subsequent queries in queue will run with the resized warehouse

Can we start or stop the warehouse any time?

Yes, suspend will stop it and resume will start it again

What is difference between Scale up and scale out?

Increasing the warehouse size is Scale up and it’s to increase performance of complex queries and for large volumes of data (when remote disk spillage happens) (from S to X)

Scale down => decreasing the warehouse size (for example from L to S)

Adding clusters to a warehouse is scale out and it is to handle concurrency (parallel queries) issues

When to go with scale up, when to with scale out?

to handle long running/complex queries, we will increase the warehouse size i.e scale up

to handle concurrency issues(parallel users), we add clusters to a warehouse i.e scale out

how to set a warehouse to auto scale mode?

While creating a warehouse If you set max\_cluster\_count (10)> min\_cluster\_count(1) then the Virtual warehouse will be in auto scale mode

it will start with minimum number of clusters and scale out to maximum clusters based on workloads on demand

**Maximized Mode**=> max\_cluster\_count(5)= min\_cluster\_count (5)

Snowflake starts all the clusters so that maximum resources are available while the warehouse is running.

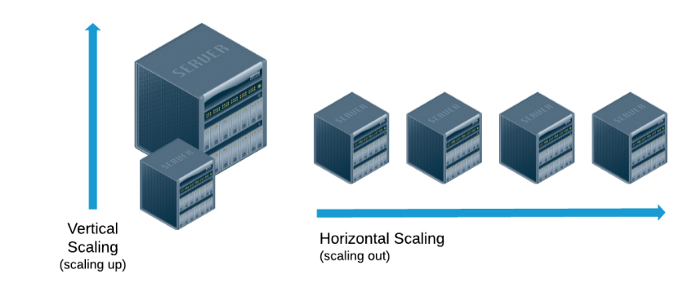
what is horizontal scaling and vertical scaling?

Scale up/scale down=> Horizontal scaling

Scale out/scale in=> Vertical scaling

What is the maximum clusters that can be added in case of scale out?

Maximum 10 clusters can be added



Can we automate scale up? No, we have to manual resize(increase/decrease) the warehouse

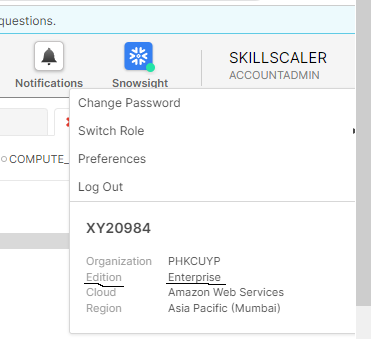
Scaling out => automated process taken care by snowflake based on your work load

**Scaling policy is used to determine when to start or shut down a warehouse.**

**Standard** (default policy) – performance (starts immediately when a query is queued)

**Economy** -- conserves credits (cluster starts only if it estimates there is enough query load to keep the cluster busy for at least 6 minutes)

Where can we see the edition/cloud/region details of your snowflake account?

right top when we expand on User and Role tab

our edition details will be available in service agreement also or we can see it when we create reader accounts if we have account admin access

Table Types

What’s are different types of tables

Permanent, Temporary, transient and external tables

What’s different between permanent, transient and temporary tables?

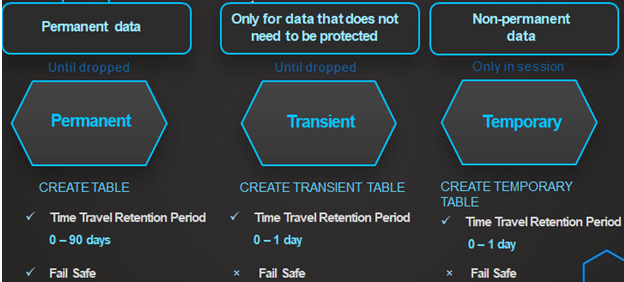
Retention period=> 0-90 days for permanent tables, 0-1 days for temporary and transient tables

Fail safe=> not available for temporary and transient tables

Scope=> until explicitly dropped for permanent and transient tables, Available only for that session for temporary tables

What is time travel in snowflake?

Time travel enables accessing historical data (i.e. data that has been changed or deleted) at any point within a defined period based on retention period. If we drop a table, with time travel we can restore it if it is with in retention period



What’s the maximum retention period for permanent tables?

Standard edition => 1 day

Enterprise edition and above => 90 days

What’s the maximum retention period for temporary tables and transient tables? 1 day

What’s the default retention period in snowflake? 1 day

What is failsafe?

Fail-safe offers protection of data in case of disaster/hardware failure, manages 7 days of historical data by snowflake

Can we configure failsafe period?

Its Non-configurable 7 days period and can be defined only for permanent

Can we disable Fail-safe in Snowflake?

Fail safe cannot be disabled on a permanent table

Transient

Can we interact/read/access data from in failsafe?

No user interaction and recoverable only by snowflake, We need to approach snowflake team to get data from failsafe

If you change the retention period at the account level, all databases, schemas, and tables that do not have an explicit retention period automatically inherit the new retention period

Account level = 1 day

Retention period at database level => 10 days

Schema => 15 days

For one table under this database I have set retention period as 30 days=> it will have 30 days of time travel

Another table I’m not setting any retention period at table level => it takes next level hierarchy I.e schema level retention period I.e 15 days

What are the differences between Time Travel and Fail-safe?

Time travel: Helps the user to query the data, how does it looks some time before and we can restore the previous state of the table.

Fail safe: internally used by snowflake to restore the data during the times of hardware failure/disaster, user can't access/read data directly from Fail safe we will have to approach snowflake team

After Time travel period is over, next 7 days data will be stored in Snowflake

Will time travel contribute to additional storage costs?

Yes, since it has to maintain snapshots/historical data

Will failsafe contribute to additional storage costs?

Yes, since it has to maintain historical data for 7 days

Which scenario did you use time travel? Restore data that got accidentally deleted using UNDROP, Backing up of data from key points in the past i.e data a week back, month back. Query data in the past that has since been updated or deleted.

Why Fail-safe?

Although data is backed up at regular intervals, there can be situations between the two back-ups during which the data has been changed accidentally or deleted due to some mistake. So the fail-safe provides a quick and efficient way to recover such data with minimal cost and minimal effort.

What are the different ways of getting/accessing/querying data from Time travel? It can be done in 3 ways

1. **By timestamp :** check data at a specific point in time

SELECT \* FROM my\_table **AT(timestamp => 'Mon, 01 May 2021 08:00:00 -0700'::timestamp\_tz);**

SELECT \* FROM my\_table **BEFORE(timestamp => 'Mon, 01 May 2021 08:00:00 -0700'::timestamp\_tz);**

1. **By offset →** In this example, we select the historical data from a table as of 15 minutes ago:

SELECT \* FROM my\_table **AT(offset => -60\*15);**

SELECT \* FROM my\_table **BEFORE(offset => -60\*15);**

1. **By query statement ID** → We can see the Query Statement ID in the history tab

SELECT \* FROM my\_table **BEFORE(STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');**

SELECT \* FROM my\_table **AT(STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');**

Accessing data with **AT => Data at that point**

Accessing data with **BEFORE => Data before that point**

Failsafe period starts immediately after Time travel ends