

Overview of warehouses

Warehouses are required for queries, as well as all DML operations, including loading data into tables. In addition to being defined by its type as either Standard or Snowpark-optimized, a warehouse is defined by its size, as well as the other properties that can be set to help control and automate warehouse activity.

Warehouses can be started and stopped at any time. They can also be resized at any time, even while running, to accommodate the need for more or less compute resources, based on the type of operations being performed by the warehouse.

Warehouse size

Size specifies the amount of compute resources available per cluster in a warehouse. Snowflake supports the following warehouse sizes:

Warehouse Size	Credits / Hour	Credits / Second	Notes
X-Small	1	0.0003	Default size for warehouses created in Snowsight and using CREATE WAREHOUSE .
Small	2	0.0006	
Medium	4	0.0011	
Large	8	0.0022	
X-Large	16	0.0044	Default size for warehouses created using the Classic Console.
2X-Large	32	0.0089	

3X-Large	64	0.0178	
4X-Large	128	0.0356	
5X-Large	256	0.0711	Generally available in Amazon Web Services (AWS) and Microsoft Azure regions, and in preview in US Government regions.
6X-Large	512	0.1422	Generally available in Amazon Web Services (AWS) and Microsoft Azure regions, and in preview in US Government regions.

Larger warehouse sizes

Larger warehouse sizes 5X-Large and 6X-Large are generally available in all Amazon Web Services (AWS) and Microsoft Azure regions.

Larger warehouse sizes are in preview in US Government regions (requires FIPS support on ARM).

Impact on credit usage and billing

As shown in the above table, there is a doubling of credit usage as you increase in size to the next larger warehouse size for each full hour that the warehouse runs; however, note that Snowflake utilizes per-second billing (with a 60-second minimum each time the warehouse starts) so warehouses are billed only for the credits they actually consume.

The total number of credits billed depends on how long the warehouse runs continuously. For comparison purposes, the following table shows the billing totals for three different size warehouses based on their running time (totals rounded to the nearest 1000th of a credit):

Running Time	Credits (X-Small)	Credits (X-Large)	Credits (5X-Large)
--------------	-------------------	-------------------	--------------------

0-60 seconds	0.017	0.267	4.268
61 seconds	0.017	0.271	4.336
2 minutes	0.033	0.533	8.532
10 minutes	0.167	2.667	42.668
1 hour	1.000	16.000	256.000

Note

- For a [multi-cluster warehouse](#), the number of credits billed is calculated based on the warehouse size and the number of clusters that run within the time period.
For example, if a 3X-Large multi-cluster warehouse runs 1 cluster for one full hour and then runs 2 clusters for the next **full** hour, the total number of credits billed would be 192 (i.e. 64 + 128).
Multi-cluster warehouses are an [Enterprise Edition](#) feature.

Impact on data loading

Increasing the size of a warehouse does **not** always improve data loading performance. Data loading performance is influenced more by the number of files being loaded (and the size of each file) than the size of the warehouse.

Tip

Unless you are bulk loading a large number of files concurrently (i.e. hundreds or thousands of files), a smaller warehouse (Small, Medium, Large) is generally sufficient. Using a larger warehouse (X-Large, 2X-Large, etc.) will consume more credits and may not result in any performance increase.

For more data loading tips and guidelines, see [Data loading considerations](#).

Impact on query processing

The size of a warehouse can impact the amount of time required to execute queries submitted to the warehouse, particularly for larger, more complex queries. In general, query performance scales with warehouse size because larger warehouses have more compute resources available to process queries.

If queries processed by a warehouse are running slowly, you can always resize the warehouse to provision more compute resources. The additional resources do not impact any queries that are already running, but once they are fully provisioned they become available for use by any queries that are queued or newly submitted.

Tip

Larger is not necessarily faster for small, basic queries.

For more warehouse tips and guidelines, see [Warehouse considerations](#).

Auto-suspension and auto-resumption

A warehouse can be set to automatically resume or suspend, based on activity:

- By default, auto-suspend is enabled. Snowflake automatically suspends the warehouse if it is inactive for the specified period of time.
- By default, auto-resume is enabled. Snowflake automatically resumes the warehouse when any statement that requires a warehouse is submitted **and** the warehouse is the current warehouse for the session.

These properties can be used to simplify and automate your monitoring and usage of warehouses to match your workload. Auto-suspend ensures that you do not leave a warehouse running (and consuming credits) when there are no incoming queries. Similarly, auto-resume ensures that the warehouse starts up again as soon as it is needed.

Note

Auto-suspend and auto-resume apply only to the entire warehouse and not to the individual clusters in the warehouse. For a [multi-cluster warehouse](#):

- Auto-suspend only occurs when the minimum number of clusters is running and there is no activity for the specified period of time. The minimum is typically 1 (cluster), but could be more than 1.
- Auto-resume only applies when the entire warehouse is suspended (i.e. no clusters are running).

Query processing and concurrency

The number of queries that a warehouse can concurrently process is determined by the size and complexity of each query. As queries are submitted, the warehouse calculates and reserves the compute resources needed to process each query. If the warehouse does not have enough remaining resources to process a query, the query is queued, pending resources that become available as other running queries complete.

Snowflake provides some object-level parameters that can be set to help control query processing and concurrency:

- [STATEMENT_QUEUED_TIMEOUT_IN_SECONDS](#)
- [STATEMENT_TIMEOUT_IN_SECONDS](#)

Note

If queries are queuing more than desired, another warehouse can be created and queries can be manually redirected to the new warehouse. In addition, resizing a warehouse can enable limited scaling for query concurrency and queuing; however, warehouse resizing is primarily intended for improving query performance.

To enable fully automated scaling for concurrency, Snowflake recommends [multi-cluster warehouses](#), which provide essentially the same benefits as creating additional warehouses and redirecting queries, but without requiring manual intervention.

Multi-cluster warehouses are an [Enterprise Edition](#) feature.

Warehouse usage in sessions

When a session is initiated in Snowflake, the session does not, by default, have a warehouse associated with it. Until a session has a warehouse associated with it, queries cannot be submitted within the session.

Default warehouse for users

To facilitate querying immediately after a session is initiated, Snowflake supports specifying a default warehouse for each individual user. The default warehouse for a user is used as the warehouse for all sessions initiated by the user.

A default warehouse can be specified when creating or modifying the user, either through the web interface or using [CREATE USER/ALTER USER](#).

Default warehouse for client utilities/drivers/connectors

In addition to default warehouses for users, any of the Snowflake clients (SnowSQL, JDBC driver, ODBC driver, Python connector, etc.) can have a default warehouse:

- SnowSQL supports both a configuration file and command line option for specifying a default warehouse.
- The drivers and connectors support specifying a default warehouse as a connection parameter when initiating a session.

For more information, see [Connect to Snowflake](#).

Precedence for warehouse defaults

When a user connects to Snowflake and start a session, Snowflake determines the default warehouse for the session in the following order:

1. Default warehouse for the user,
» **overridden by...**
2. Default warehouse in the configuration file for the client utility (SnowSQL, JDBC driver, etc.) used to connect to Snowflake (if the client supports configuration files),
» **overridden by...**
3. Default warehouse specified on the client command line or through the driver/connector parameters passed to Snowflake.

Note

In addition, the default warehouse for a session can be changed at any time by executing the [USE WAREHOUSE](#) command within the session.