

Monitor a Microsoft Fabric data warehouse

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/monitor-fabric-data-warehouse/01-introduction>

Introduction

Completed

A data warehouse is often central to enterprise analysis and reporting, and as such is a critical business asset. It's important to monitor a data warehouse to track and manage costs, identify and resolve query performance issues, and to gain insights into how your data is being used.

In this module, we'll explore some tools and techniques you can use to monitor your Microsoft Fabric data warehouse.

2. Monitor capacity metrics

<https://learn.microsoft.com/en-us/training/modules/monitor-fabric-data-warehouse/02-capacity-metrics>

Monitor capacity metrics

Completed

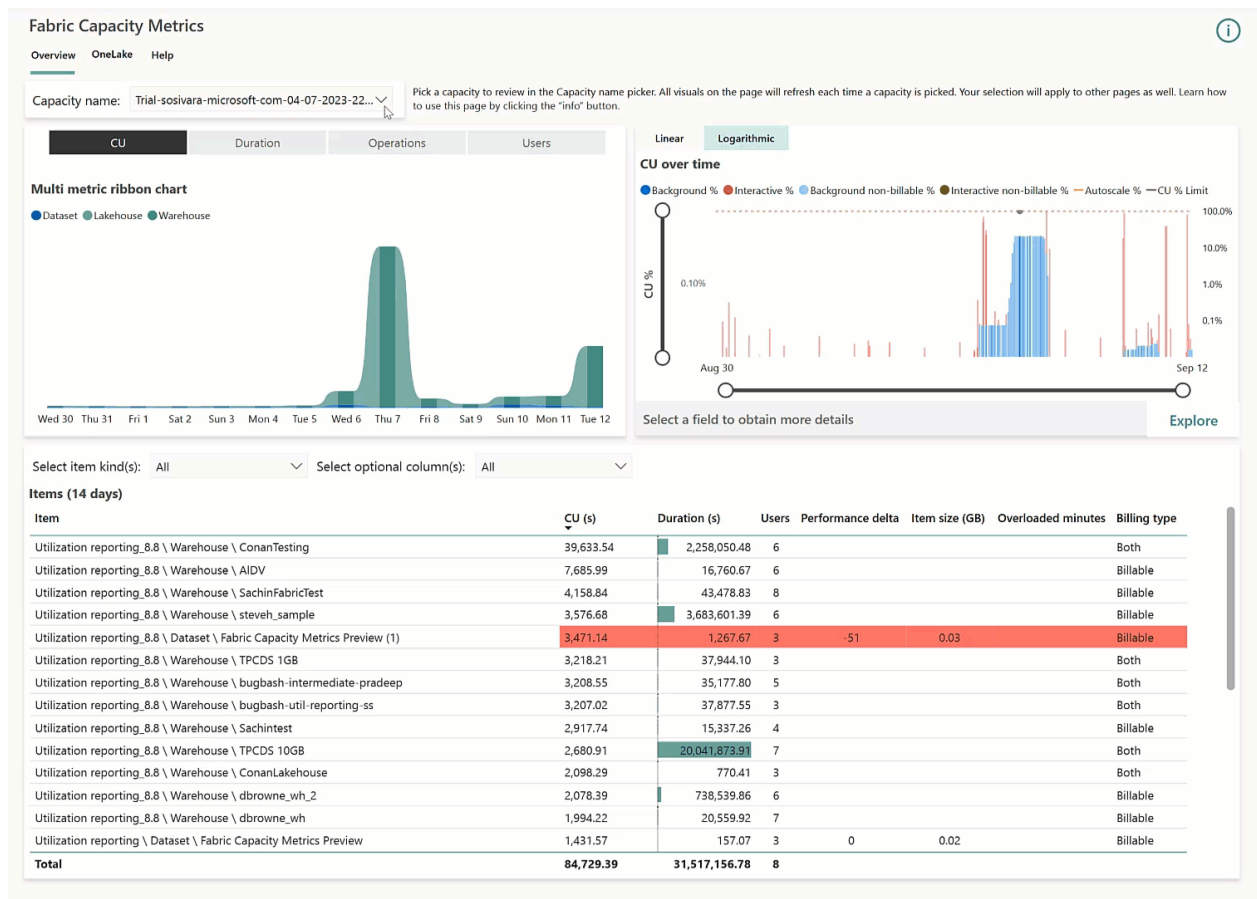
When your organization uses Microsoft Fabric, the license used to purchase the service determines the *capacity* available. A capacity is a pool of resources that you can use to implement Fabric capabilities.

The cost of using Fabric is based on *capacity units* (CUs). Each action you perform in a Fabric resource can consume CUs, for which your organization is billed. It's therefore important to be able to monitor capacity usage to plan and manage costs. In data warehouse workloads, CUs are consumed by data read and write activities, so queries in your data warehouse and the underlying file operations to OneLake storage are a significant factor in the cost of your Fabric analytics solution.

Using the Microsoft Fabric Capacity Metrics app

The Microsoft Fabric Capacity Metrics app is an app that an administrator can install in a Fabric environment and use to monitor capacity utilization. To monitor capacity utilization related to data warehousing, you can

filter the interface to show only **warehouse** activity, like this:



By using the Fabric Capacity Metrics app, you can observe capacity utilization trends to determine what processes are consuming CUs in your Fabric environment and whether any *throttling* is occurring (which indicates that your processes require more capacity than is available within the constraints of your purchased capacity license). With this information, you can optimize your capacity license for your needs.

Tip

For more information about Microsoft Fabric Capacity Metrics app, refer to [Billing and utilization reporting in Data Warehouse](#) in the Microsoft Fabric documentation.

3. Monitor current activity

<https://learn.microsoft.com/en-us/training/modules/monitor-fabric-data-warehouse/03-dynamic-management-views>

Monitor current activity

Completed

You can use dynamic management views (DMVs) to retrieve information about the current state of the data warehouse. Specifically, Microsoft Fabric data warehouses include the following DMVs:

- **sys.dm_exec_connections:** Returns information about data warehouse connections.
- **sys.dm_exec_sessions:** Returns information about authenticated sessions.
- **sys.dm_exec_requests:** Returns information about active requests.

The schema of these tables is shown here:

sys.dm_exec_sessions	
session_id	
login_time	
host_name	
program_name	
host_process_id	
client_version	
client_interface_name	
security_id	
login_name	
nt_domain	
nt_user_name	
status	
context_info	
cpu_time	
memory_usage	
total_scheduled_time	
total_elapsed_time	
endpoint_id	
last_request_start_time	
last_request_end_time	
reads	
writes	
logical_reads	
is_user_process	
text_size	
language	
date_format	
date_first	
quoted_identifier	
arithabort	
ansi_null_dflt_on	
ansi_defaults	
ansi_warnings	
ansi_padding	
ansi_nulls	
concat_null_yields_null	
transaction_isolation_level	
lock_timeout	
deadlock_priority	
row_count	
prev_error	
original_security_id	
original_login_name	
last_successful_logon	
last_unsuccessful_logon	
unsuccessful_logons	
group_id	
database_id	
authenticating_database_id	
open_transaction_count	
is_filtered	
page_server_reads	
contained_availability_group_id	

sys.dm_exec_connections	
session_id	
most_recent_session_id	
connect_time	
net_transport	
protocol_type	
protocol_version	
endpoint_id	
encrypt_option	
auth_scheme	
node_affinity	
num_reads	
num_writes	
last_read	
last_write	
net_packet_size	
client_net_address	
client_tcp_port	
local_net_address	
local_tcp_port	
connection_id	
parent_connection_id	
most_recent_sql_handle	

sys.dm_exec_requests	
session_id	
request_id	
start_time	
status	
command	
sql_handle	
statement_start_offset	
statement_end_offset	
plan_handle	
database_id	
user_id	
connection_id	
blocking_session_id	
wait_type	
wait_time	
last_wait_type	
wait_resource	
open_transaction_count	
open_resultset_count	
transaction_id	
context_info	
percent_complete	
estimated_completion_time	
cpu_time	
total_elapsed_time	
scheduler_id	
task_address	
reads	
writes	
logical_reads	
text_size	
language	
date_format	
date_first	
quoted_identifier	
arithabort	
ansi_null_dflt_on	
ansi_defaults	
ansi_warnings	
ansi_padding	
ansi_nulls	
concat_null_yields_null	
transaction_isolation_level	
lock_timeout	
deadlock_priority	
row_count	
prev_error	
nest_level	
granted_query_memory	
executing_managed_code	
group_id	
query_hash	
query_plan_hash	
statement_sql_handle	
statement_context_id	
dop	
parallel_worker_count	
external_script_request_id	
is_resumable	
page_resource	
page_server_reads	
dist_statement_id	

Querying DMVs

You can retrieve detailed information about current activities in the data warehouse by querying the `dm_exec-*` DMVs. For example, consider the following query:

```
SELECT sessions.session_id, sessions.login_name,
       connections.client_net_address,
       requests.command, requests.start_time, requests.total_elapsed_time
FROM sys.dm_exec_connections AS connections
INNER JOIN sys.dm_exec_sessions AS sessions
    ON connections.session_id=sessions.session_id
INNER JOIN sys.dm_exec_requests AS requests
    ON requests.session_id = sessions.session_id
WHERE requests.status = 'running'
    AND requests.database_id = DB_ID()
ORDER BY requests.total_elapsed_time DESC
```

This query returns details about the active requests in the current database, ordered by the duration for which they have been executing; which may be useful to identify long-running queries that could benefit from optimization. An example result set from the query is shown here:

session_id	login_name	client_net_address	command	start_time	total_elapsed_time
60	fred@contoso.com	10.23.139.162	SELECT	2023-12-07T14:56:41.3530000	57266
126	nandita@contoso.com	10.23.137.98	SELECT	2023-12-07T14:57:22.7800000	15840
137	zoe@contoso.com	10.23.119.171	SELECT	2023-12-07T14:57:38.6070000	4

Tip

For more information about using DMVs, refer to [Monitor connections, sessions, and requests using DMVs](#) in the Microsoft Fabric documentation.

4. Monitor queries

<https://learn.microsoft.com/en-us/training/modules/monitor-fabric-data-warehouse/04-query-insights>

Monitor queries

Completed

Microsoft Fabric data warehouses include *query insights* feature that provides historical, aggregated information about the queries that have been run; enabling you to identify frequently used or long-running queries, and helping you analyze and tune query performance.

Query insights are provided through the following views:

- **queryinsights.exec_requests_history**: Details of each completed SQL query.
- **queryinsights.long_running_queries**: Details of query execution time.
- **queryinsights.frequently_run_queries**: Details of frequently run queries.

The schema for these tables is shown here:

queryinsights.exec_requests_history	queryinsights.long_running_queries	queryinsights.frequently_run_queries
batch_id	last_dist_statement_id	avg_total_elapsed_time_ms
command	last_run_command	last_dist_statement_id
connection_id	last_run_session_id	last_run_command
distributed_statement_id	last_run_start_time	last_run_start_time
end_time	last_run_total_elapsed_time_ms	last_run_total_elapsed_time_ms
login_name	median_total_elapsed_time_ms	max_run_total_elapsed_time_ms
program_name	number_of_runs	min_run_total_elapsed_time_ms
query_hash		number_of_canceled_runs
root_batch_id		number_of_failed_runs
row_count		number_of_runs
session_id		number_of_successful_runs
start_time		
status		
total_elapsed_time_ms		

Retrieving query insights

The query insights views are a useful source of information about the queries that are being run in your data warehouse.

For example, consider the following query:

```
SELECT start_time, login_name, command
FROM queryinsights.exec_requests_history
WHERE start_time >= DATEADD(MINUTE, -60, GETUTCDATE())
```

This query uses the **queryinsights.exec_requests_history** view to identify queries that were run in the previous hour.

Note

Depending on the concurrent workloads, queries may take up to 15 minutes to be reflected in the query insights views.

You can get details of long-running queries from the **queryinsights.long_running_queries** view like this:

```
SELECT last_run_command, number_of_runs, median_total_elapsed_time_ms, last_run_start_time
FROM queryinsights.long_running_queries
WHERE number_of_runs > 1
ORDER BY median_total_elapsed_time_ms DESC;
```

This query identifies long-running SQL commands that have been used more than once and returns them in descending order of their median time to complete.

Note

To enable the views to provide aggregated metrics, queries with predicates are parameterized and considered the same query if the parameterized statements are an exact match. For example, the following queries would be considered to be the same command:

```
SELECT * FROM sales WHERE orderdate > '01/01/2023'
```

```
SELECT * FROM sales WHERE orderdate > '12/31/2021'
```

To find commonly used queries, you can use the **queryinsights.frequently_run_queries** view, like this:

```
SELECT last_run_command, number_of_runs, number_of_successful_runs, number_of_failed_runs  
FROM queryinsights.frequently_run_queries  
ORDER BY number_of_runs DESC;
```

This query returns details of successful and failed runs for frequently run commands.

Tip

For more information about using query insights refer to [Query insights in Fabric data warehousing](#) in the Microsoft Fabric documentation.

5. Exercise - Monitor a data warehouse in Microsoft Fabric

<https://learn.microsoft.com/en-us/training/modules/monitor-fabric-data-warehouse/05-exercise-monitor>

Exercise - Monitor a data warehouse in Microsoft Fabric

Completed

Now it's time to try monitoring a Microsoft Fabric data warehouse for yourself.

Note

You need a Microsoft Fabric trial license with the Fabric preview enabled in your tenant. See [Getting started with Fabric](#) to enable your Fabric trial license. Launch the exercise and follow the instructions.

[Launch Exercise](#)

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/monitor-fabric-data-warehouse/06-knowledge-check>

Module assessment

Completed

7. Summary

<https://learn.microsoft.com/en-us/training/modules/monitor-fabric-data-warehouse/07-summary>

Summary

Completed

Microsoft Fabric data warehouses are an important asset in any organization, and it's important to monitor them to track and manage costs, identify and resolve query performance issues, and to gain insights into how your data is being used.

This module explored the following ways in which you can monitor a data warehouse:

- The Microsoft Fabric Capacity Metrics app
- Dynamic Management Views
- Query Insights views