# SQLMI Portal Metrics Query

Last updated by | Akio Hose | Jun 22, 2022 at 8:14 PM PDT

| Author | Date |
| --- | --- |
| Akio Hose | Jun 23, 2022 |

Contents



## T-SQL query to get similar data as displayed in Azure portal

If your customer wants to know the T-SQL query that represents the metrics shown at the Azure portal, here is a sample query you can share with the customer. The query is based on the Node Agent module which monitors the metrics (Slightly modified).

You can adust granularity by setting @timeGrain in seconds.

```
declare  @starttime datetime2 = '<YYYY-MM-DD HH:MM:ss>'
declare @endtime datetime2 = '<YYYY-MM-DD HH:MM:ss>'
declare @timeGrain int = 900  -- Set the granularity (seconds)


SELECT max(start_time) as 'start', max(end_time) as 'end',
((CONVERT(BIGINT, DATEDIFF(day, 0, [end_time])) * 24 * 3600 + DATEDIFF(second, DATEADD(day, DATEDIFF(day, 0, [
                    , AVG(cpu_percent) as 'Average CPU percentage, Avg'
                , AVG(storage_limit) as 'Storage space reserved, Avg'
               , AVG(storage_used) as 'Storage space used, Avg'
                , AVG(vcore_count) as 'Virtual core count, Avg'
                , AVG(io_requests_view) as 'IO requests count, Avg'
                , AVG(io_bytes_read_view) as 'IO bytes read, Avg'
                , AVG(io_bytes_written_view)  as 'IO bytes written, Avg'
                , MIN(cpu_percent) as 'Average CPU percentage, Min'
                , MIN(storage_limit) as 'Storage space reserved, Min'
                , MIN(storage_used) as 'Storage space used, Min'
                , MIN(vcore_count) as 'Virtual core count, Min'
                , MIN(io_requests_view) as 'IO requests count, Min'
                , MIN(io_bytes_read_view) as 'IO bytes read, Min'
                , MIN(io_bytes_written_view) as 'IO bytes written, Min'
                , MAX(cpu_percent) as 'Average CPU percentage, Max'
                , MAX(storage_limit) as 'Storage space reserved, Max'
                , MAX(storage_used) as 'Storage space used, Max'
                , MAX(vcore_count) as 'Virtual core count, Max'
                , MAX(io_requests_view) as 'IO requests count, Max'
                , MAX(io_bytes_read_view) as 'IO bytes read, Max'
                , MAX(io_bytes_written_view) as 'IO bytes written, Max'
                , SUM(io_requests_view) as io_requests_sum
                , SUM(io_bytes_read_view) as io_bytes_read_sum
                , SUM(io_bytes_written_view) as io_bytes_written_sum
                , COUNT(cpu_percent) as count
            FROM
                -- raw data points of 15 seconds
                (SELECT
                                start_time,
                    end_time
                    , ISNULL(avg_cpu_percent, 0) as cpu_percent
                    , ISNULL(reserved_storage_mb, 0) as storage_limit
                    , ISNULL(storage_space_used_mb , 0) as storage_used
                    , ISNULL(virtual_core_count , 0) as vcore_count
                    , ISNULL(io_requests , 0) as io_requests_view
                    , ISNULL(io_bytes_read , 0) as io_bytes_read_view
                    , ISNULL(io_bytes_written , 0) as io_bytes_written_view
                FROM sys.server_resource_stats
                WHERE [end_time] >= @startTime AND [end_time] <= @endTime
                ) t
            GROUP BY ((CONVERT(BIGINT, DATEDIFF(day, 0, [end_time])) * 24 * 3600 +
                        DATEDIFF(second, DATEADD(day, DATEDIFF(day, 0, [end_time]), 0), [end_time])) / @timeGr
```

## External reference

- Microsoft.Sql/managedInstances
- Monitoring and performance tuning in Azure SQL Database and Azure SQL Managed Instance

## Internal reference

- NodeAgent.Metrics.cs

## Kusto version

TBD

## How good have you found this content?