In this module, you will:

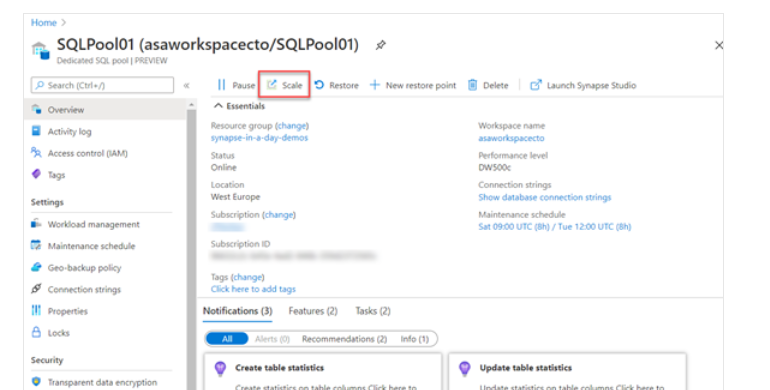
* Scale compute resources in Azure Synapse Analytics
* Pause compute in Azure Synapse Analytics
* Manage workloads in Azure Synapse Analytics
* Use Azure Advisor to review recommendations
* Use Dynamic Management Views to identify and troubleshoot query performance

Before taking this module, it is recommended that the student is able to:

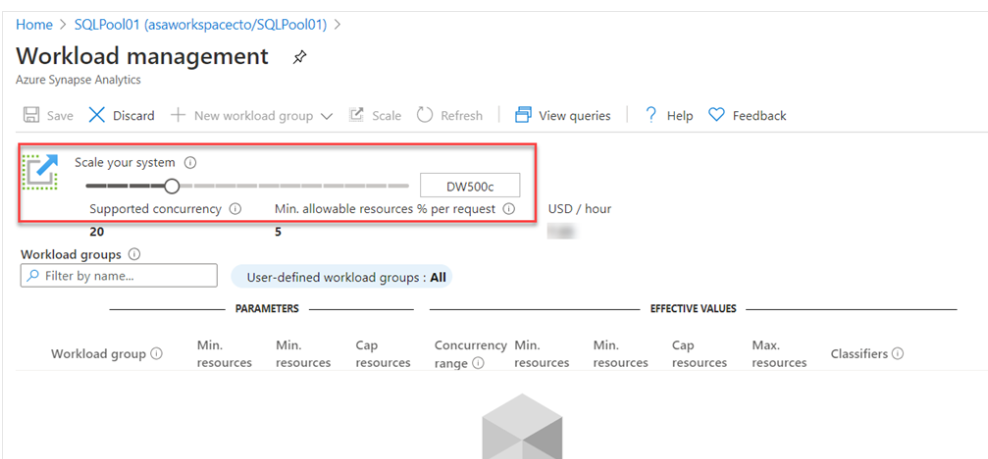
* Log into the Azure portal
* Create a Synapse Analytics Workspace
* Create an Azure Synapse Analytics SQL Pool

**Scale compute resources in Azure Synapse Analytics**

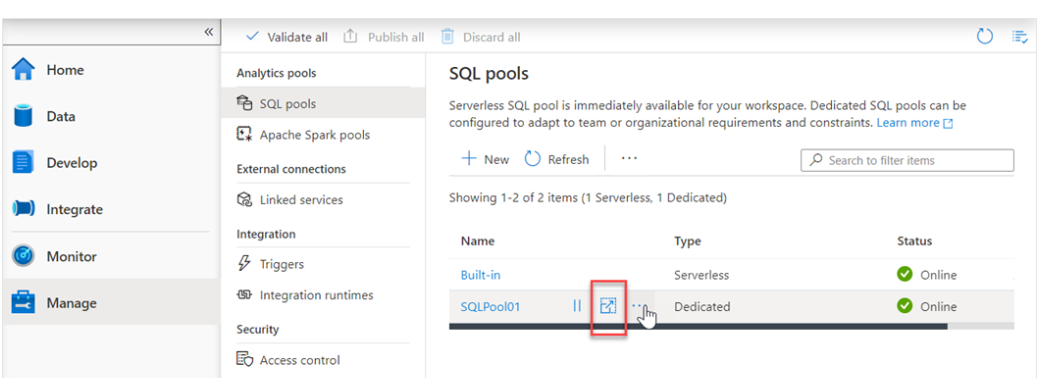
* One of the key management features that you have at your disposal within Azure Synapse Analytics, is the ability to scale the compute resources for SQL or Spark pools to meet the demands of processing your data. In SQL pools, the unit of scale is an abstraction of compute power that is known as a data warehouse unit. Compute is separate from storage, which enables you to scale compute independently of the data in your system. This means you can scale up and scale down the compute power to meet your needs.
* You can scale a Synapse SQL pool either through the Azure portal, Azure Synapse Studio or programmatically using TSQL or PowerShell.
* In the Azure portal, you can click on **scale** icon



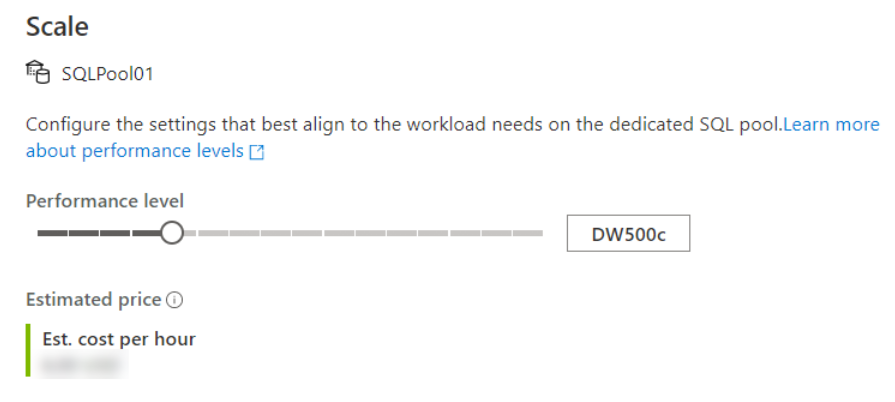
And then you can adjust the **slider** to scale the SQL Pool



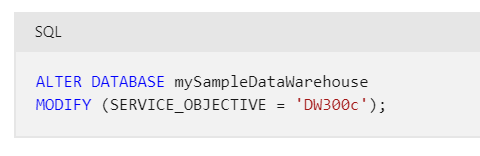
Another option to scale is within Azure Synapse Studio, click on the **scale** icon:



And then move the **slider** as follows:



You can also make the modification using Transact-SQL



Or by using PowerShell:

Set-AzSqlDatabase -ResourceGroupName "resourcegroupname" -DatabaseName "mySampleDataWarehouse" -ServerName "sqlpoolservername" -RequestedServiceObjectiveName "DW300c"

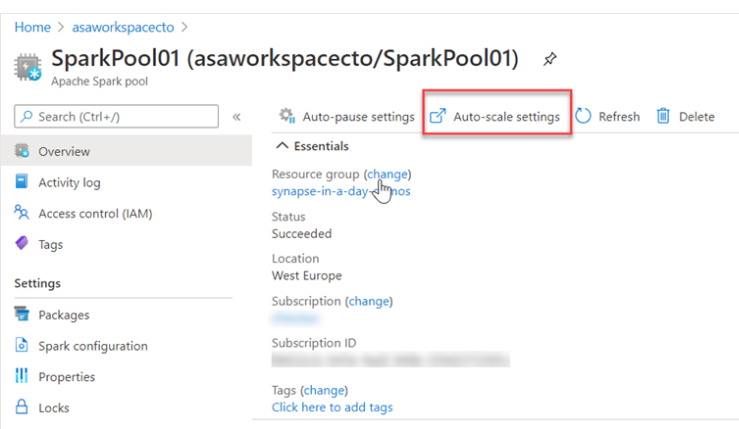
## **Scaling Apache Spark pools in Azure Synapse Analytics**

Apache Spark pools for Azure Synapse Analytics uses an **Autoscale** feature that automatically scales the number of nodes in a cluster instance up and down. During the creation of a new Spark pool, a minimum and maximum number of nodes can be set when **Autoscale** is selected. Autoscale then monitors the resource requirements of the load and scales the number of nodes up or down. To enable the Autoscale feature, complete the following steps as part of the normal pool creation process:

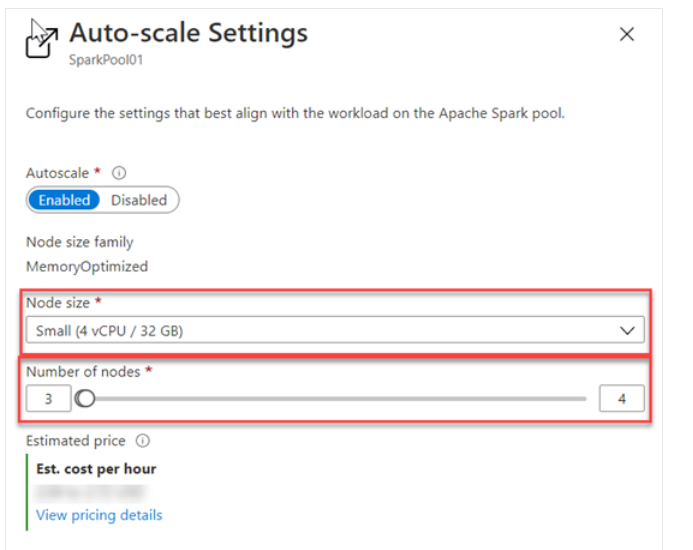
1. On the **Basics** tab, select the **Enable autoscale** checkbox.
2. Enter the desired values for the following properties:
   * **Min** number of nodes.
   * **Max** number of nodes.

The initial number of nodes will be the minimum. This value defines the initial size of the instance when it's created. The minimum number of nodes can't be fewer than three.

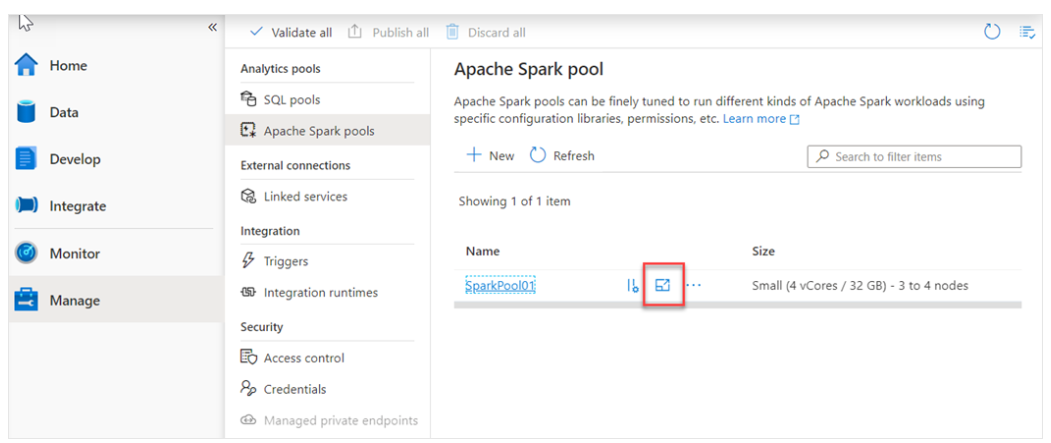
You can also modify this in the Azure portal, you can click on **auto-scale settings** icon



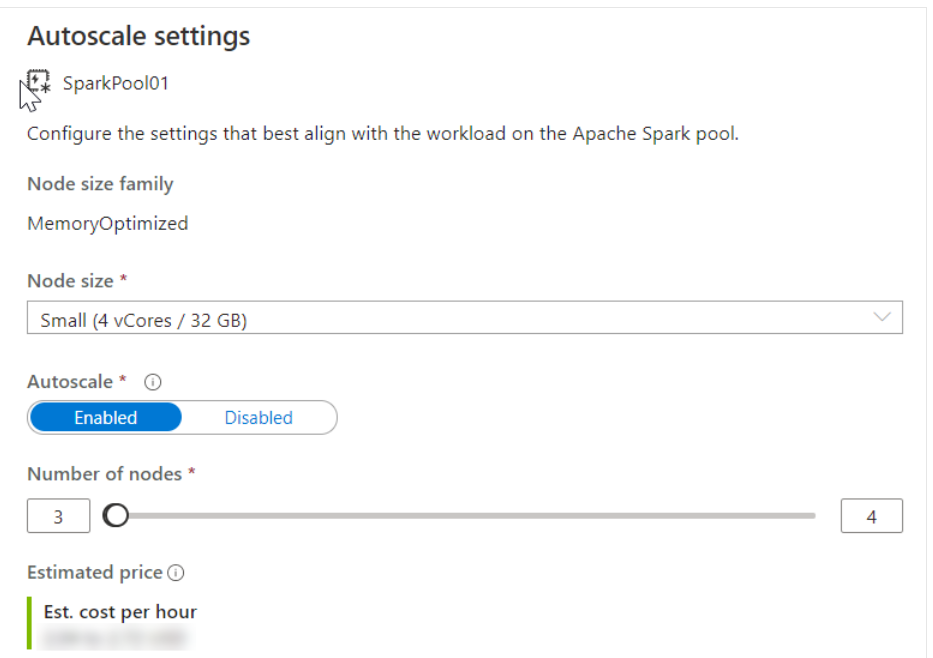
Choose the node size and the number of nodes:



and for Azure Synapse Studio as follows



And Choose the node size and the number of nodes



Autoscale continuously monitors the Spark instance and collects the following metrics:

| **Metric** | **Description** |
| --- | --- |
| Total Pending CPU | The total number of cores required to start execution of all pending nodes. |
| Total Pending Memory | The total memory (in MB) required to start execution of all pending nodes. |
| Total Free CPU | The sum of all unused cores on the active nodes. |
| Total Free Memory | The sum of unused memory (in MB) on the active nodes. |
| Used Memory per Node | The load on a node. A node on which 10 GB of memory is used, is considered under more load than a worker with 2 GB of used memory. |

The following conditions will then autoscale the memory or CPU

| **Scale-up** | **Scale-down** |
| --- | --- |
| Total pending CPU is greater than total free CPU for more than 1 minute. | Total pending CPU is less than total free CPU for more than 2 minutes. |
| Total pending memory is greater than total free memory for more than 1 minute. | Total pending memory is less than total free memory for more than 2 minutes. |

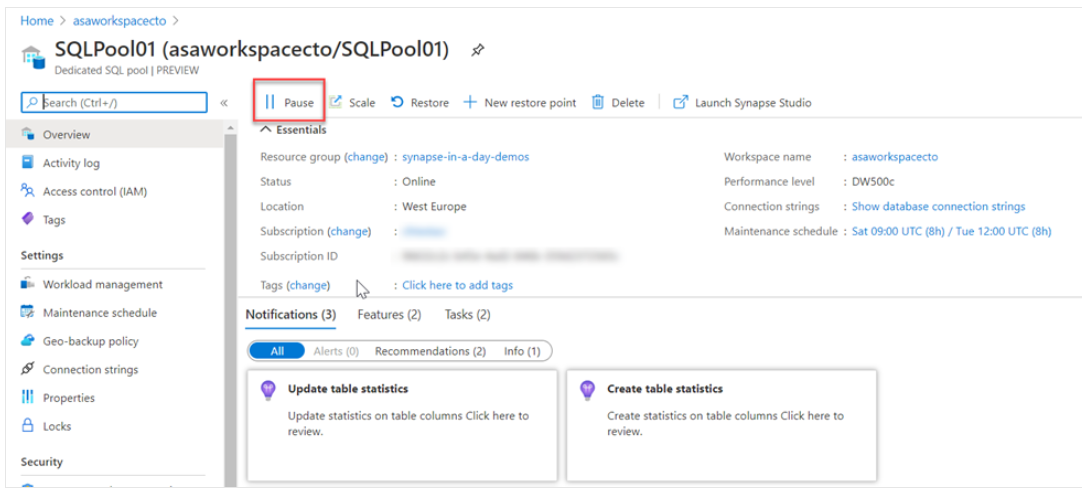
The scaling operation can take between 1 -5 minutes. During an instance where there is a scale down process, Autoscale will put the nodes in decommissioning state so that no new executors can launch on that node.

The running jobs will continue to run and finish. The pending jobs will wait to be scheduled as normal with fewer available nodes.

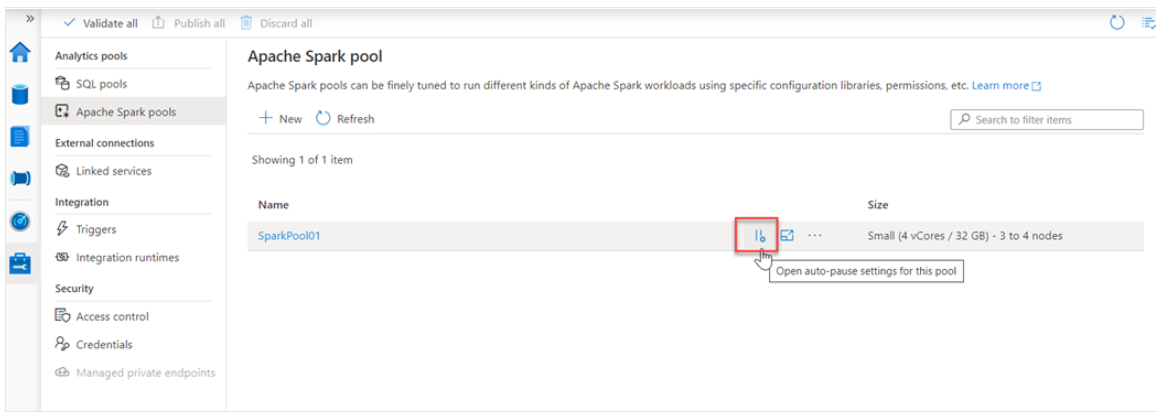
# Pause compute in Azure Synapse Analytics

When performing the batch movement of data to populate a data warehouse, it is typical for the data engineer to understand the schedule on which the data loads take place. In these circumstances, you may be able to predict the periods of downtime in the data loading and querying process and take advantage of the pause operations to minimize your costs.

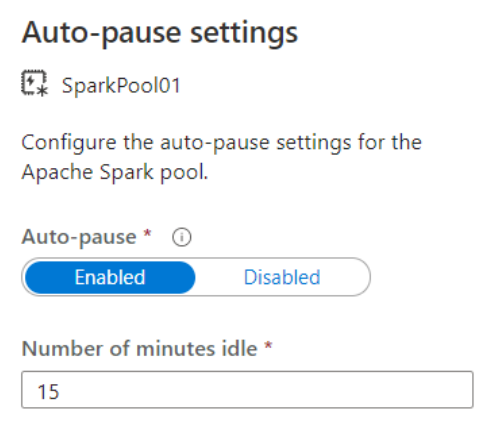
In the Azure portal you can use the Pause command within the dedicated SQL pool



And this can also be used within Azure Synapse Studio for Apache Spark pools too, in the Manage hub.



Which allows you to enable it, and set the number of minutes idle



# Manage workloads in Azure Synapse Analytics

Azure Synapse Analytics allows you to create, control and manage resource availability when workloads are competing. This allows you to manage the relative importance of each workload when waiting for available resources.

To facilitate faster load times, you can create a workload classifier for the load user with the “importance” set to above\_normal or High. Workload importance ensures that the load takes precedence over other waiting tasks of a lower importance rating. Use this in conjunction with your own workload group definitions for workload isolation to manage minimum and maximum resource allocations during peak and quiet periods.

Dedicated SQL pool workload management in Azure Synapse consists of three high-level concepts:

* Workload Classification
* Workload Importance
* Workload Isolation

These capabilities give you more control over how your workload utilizes system resources.

## **Workload classification**

Workload management classification allows workload policies to be applied to requests through assigning resource classes and importance.

While there are many ways to classify data warehousing workloads, the simplest and most common classification is **load** and **query**. You load data with insert, update, and delete statements. You query the data using selects. A data warehousing solution will often have a workload policy for load activity, such as assigning a higher resource class with more resources. A different workload policy could apply to queries, such as lower importance compared to load activities.

You can also subclassify your load and query workloads. Subclassification gives you more control of your workloads.

For example, (a) **query workloads** can consist of cube refreshes, dashboard queries or ad-hoc queries. You can classify each of these query workloads with different resource classes or importance settings. (b) **Load workloads** can also benefit from subclassification. Large transformations can be assigned to larger resource classes. Higher importance can be used to ensure key sales data is loaded before weather data or a social data feed.

Not all statements are classified as they do not require resources or need importance to influence execution. DBCC commands, BEGIN, COMMIT, and ROLLBACK TRANSACTION statements are not classified.

NOTE: DBCC - Database Console Commands for SQL Server

Database Console Command statements are grouped into the following categories.

| **Command category** | **Perform** |
| --- | --- |
| Maintenance | Maintenance tasks on a database, index, or filegroup. |
| Miscellaneous | Miscellaneous tasks such as enabling trace flags or removing a DLL from memory. |
| Informational | Tasks that gather and display various types of information. |
| Validation | Validation operations on a database, table, index, catalog, filegroup, or allocation of database pages. |

DBCC commands take input parameters and return values. All DBCC command parameters can accept both Unicode and DBCS literals.

* DBCC CHECKALLOC
* DBCC CHECKCATALOG
* DBCC CHECKDB
* DBCC CHECKFILEGROUP
* DBCC CHECKTABLE
* [DBCC INPUTBUFFER](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-inputbuffer-transact-sql?view=sql-server-ver16)
* [DBCC SHOWCONTIG](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-showcontig-transact-sql?view=sql-server-ver16)
* [DBCC OPENTRAN](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-opentran-transact-sql?view=sql-server-ver16)
* [DBCC OUTPUTBUFFER](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-outputbuffer-transact-sql?view=sql-server-ver16)
* [DBCC PROCCACHE](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-proccache-transact-sql?view=sql-server-ver16)
* [DBCC SHOW\_STATISTICS](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-show-statistics-transact-sql?view=sql-server-ver16)
* [DBCC SQLPERF](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-sqlperf-transact-sql?view=sql-server-ver16)
* [DBCC TRACESTATUS](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-tracestatus-transact-sql?view=sql-server-ver16)
* [DBCC USEROPTIONS](https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-useroptions-transact-sql?view=sql-server-ver16)

More about DBCC:

<https://learn.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-transact-sql?view=sql-server-ver16>

## **Workload importance**

Workload importance influences the order in which a request gets access to resources. On a busy system, a request with higher importance has first access to resources. Importance can also ensure ordered access to locks. There are five levels of importance: low, below\_normal, normal, above\_normal, and high. Requests that don't set importance are assigned the default level of normal. Requests that have the same importance level have the same scheduling behavior that exists today.

## **Workload isolation**

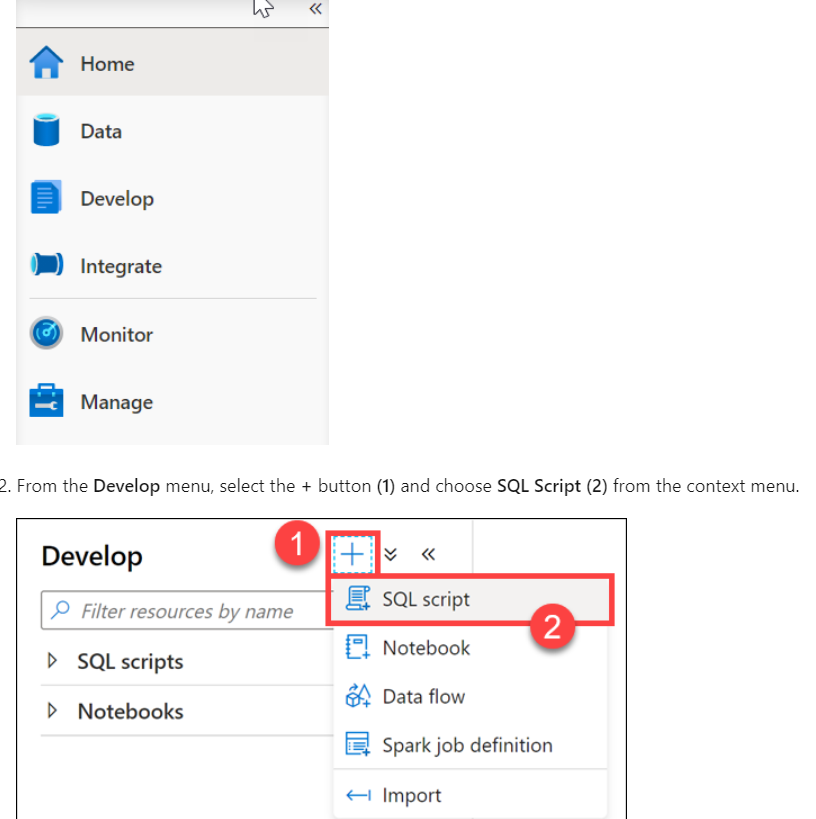
Workload isolation reserves resources for a workload group. Resources reserved in a workload group are held exclusively for that workload group to ensure execution. Workload groups also allow you to define the amount of resources that are assigned per request, much like resource classes do. Workload groups give you the ability to reserve or cap the amount of resources a set of requests can consume. Finally, workload groups are a mechanism to apply rules, such as query timeout, to requests.

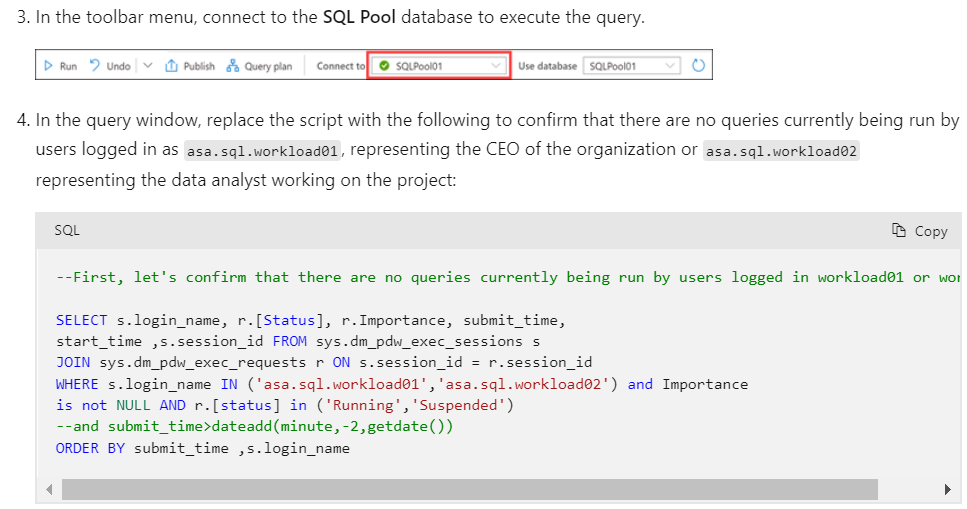
You can perform the following steps to implement workload management

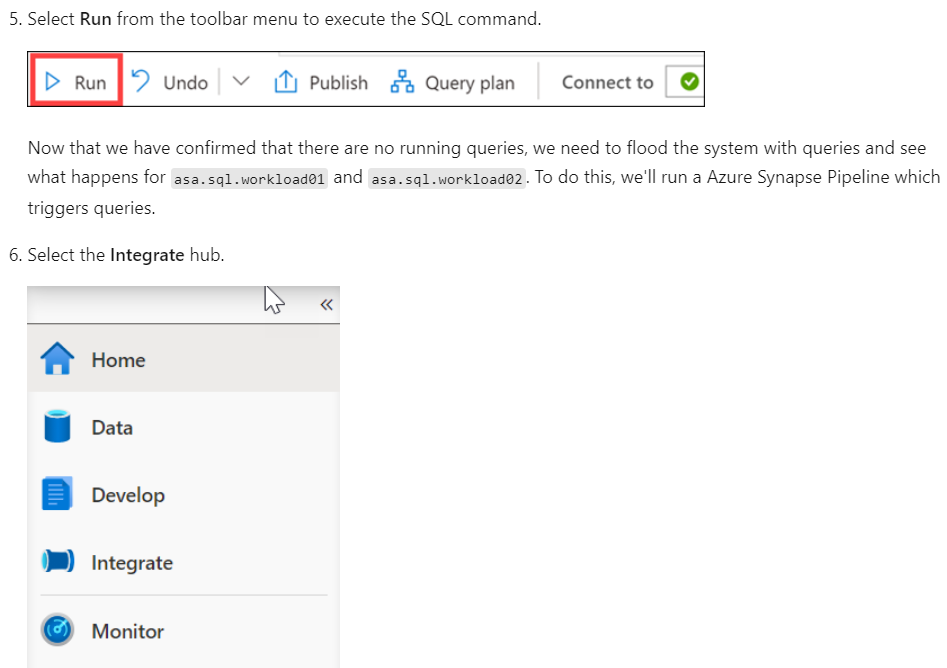
### Create a workload classifier to add importance to certain queries

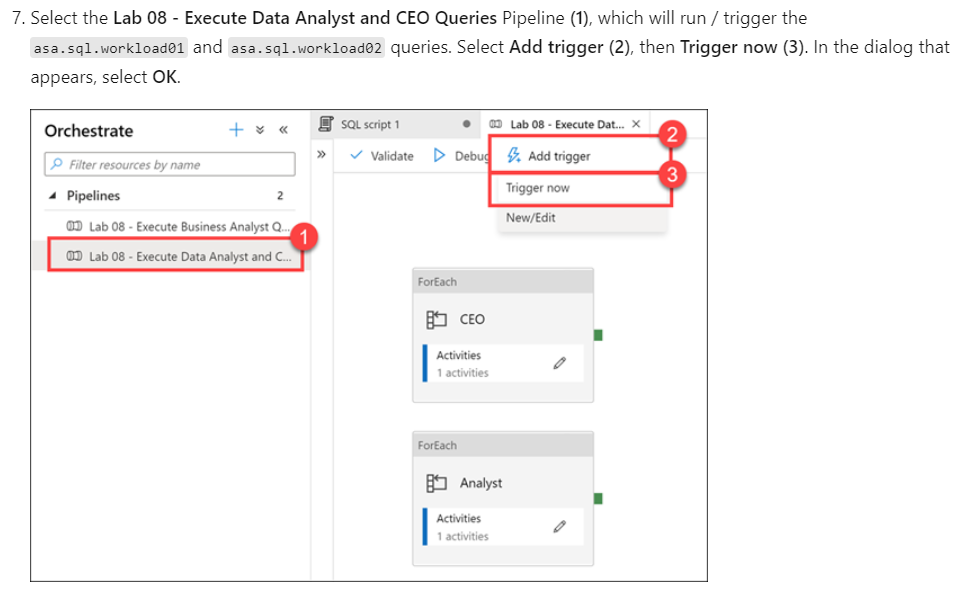
Your organization has asked you if there is a way to mark queries executed by the CEO as more important than others, so they don't appear slow due to heavy data loading or other workloads in the queue. You decide to create a workload classifier and add importance to prioritize the CEO's queries.

1. Select the **Develop** hub.



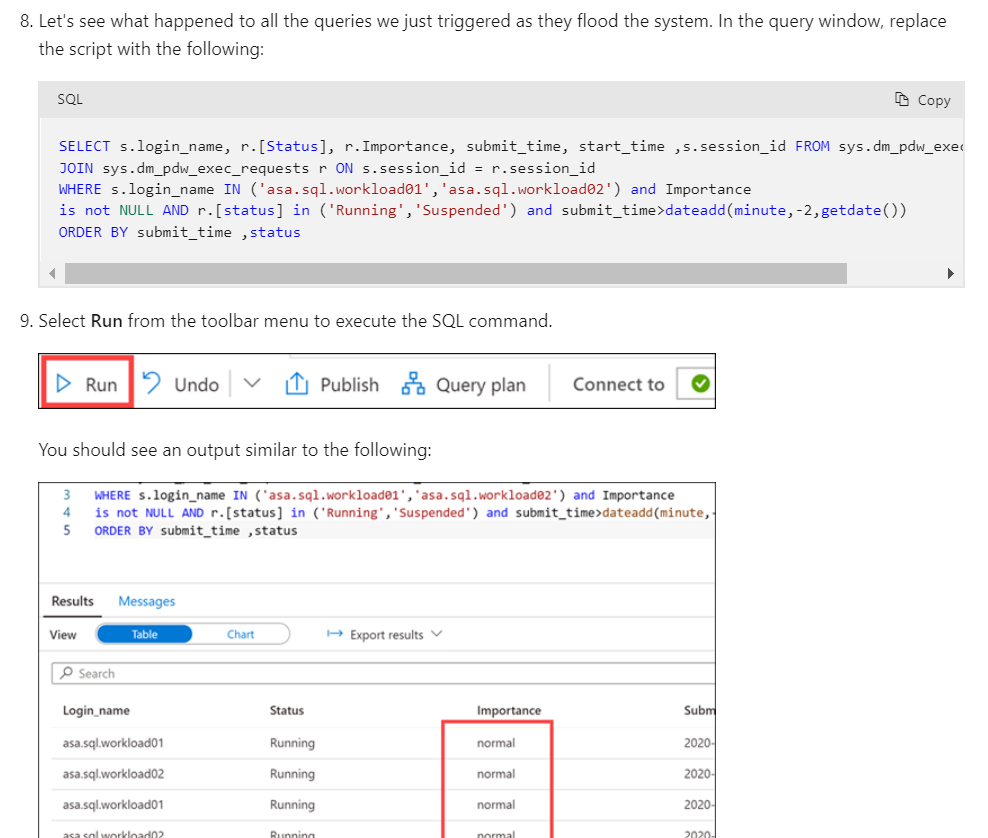


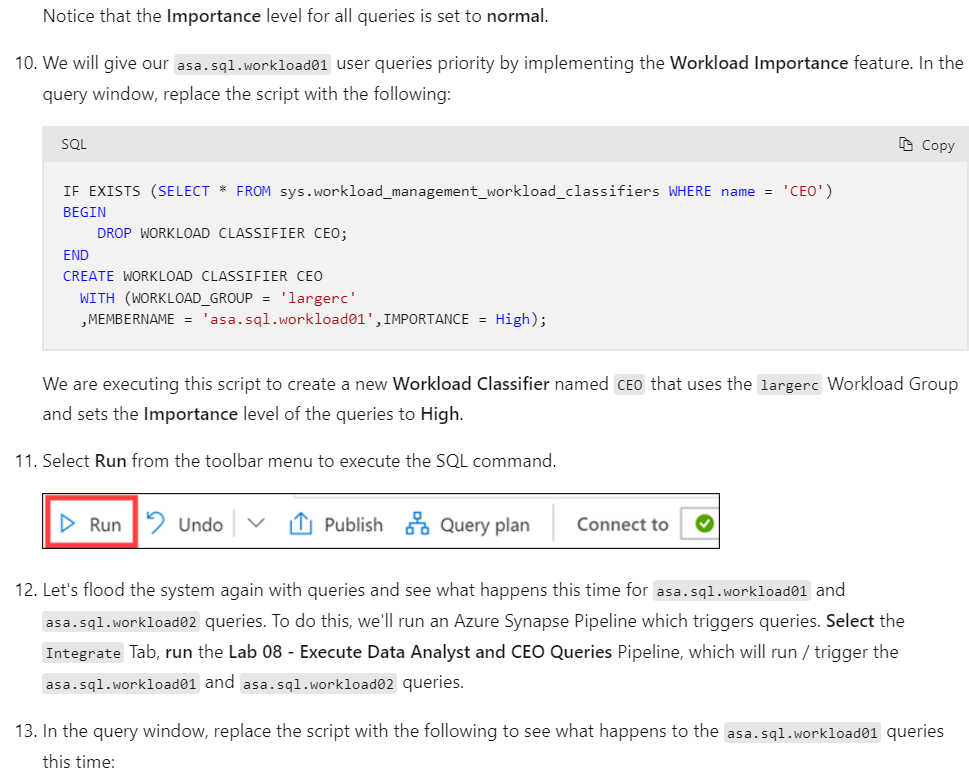


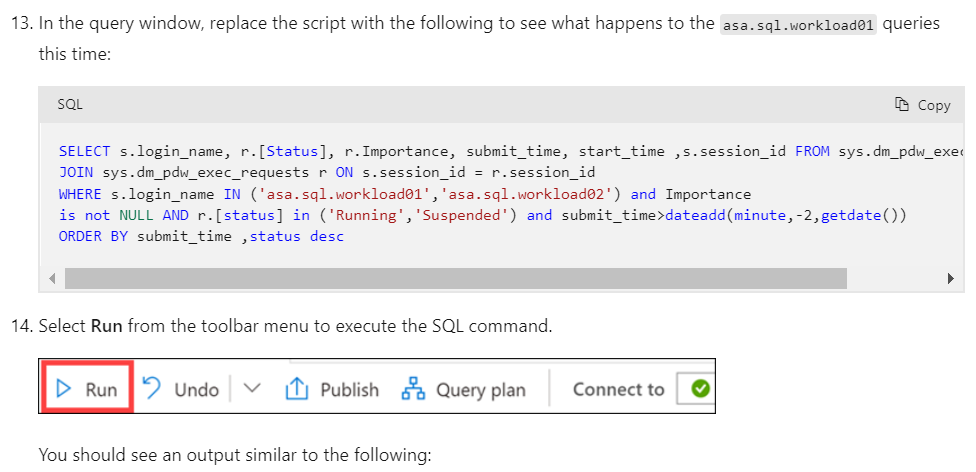


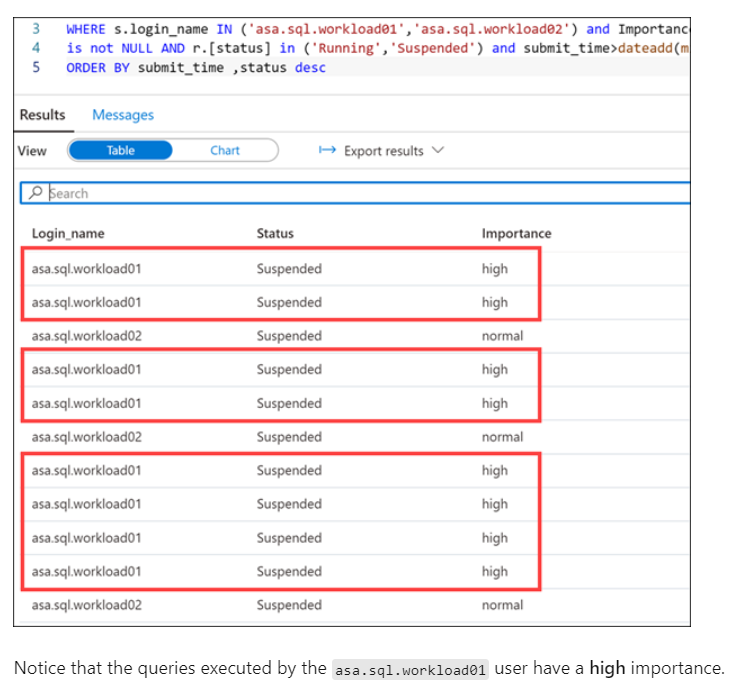
Go to this link in case needed:

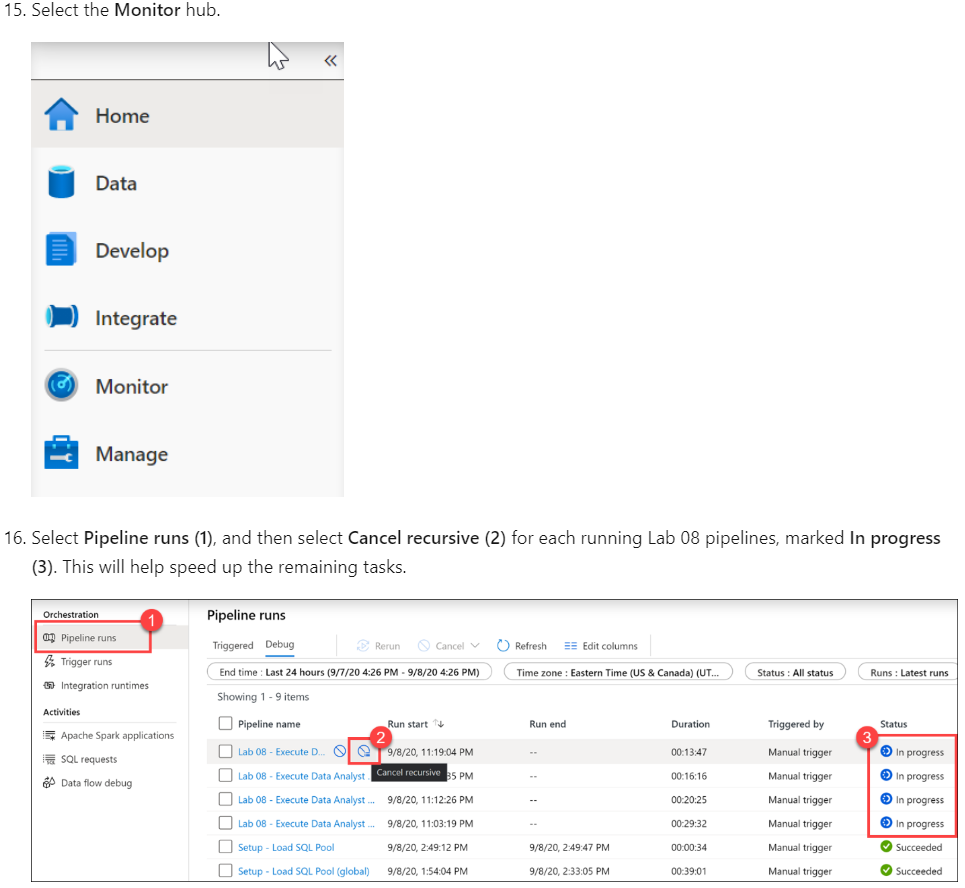
<https://learn.microsoft.com/en-us/training/modules/manage-monitor-data-warehouse-activities-azure-synapse-analytics/4-manage-workloads>











### **Reserve resources for specific workloads through workload isolation**

Workload isolation means resources are reserved, exclusively, for a workload group. Workload groups are containers for a set of requests and are the basis for how workload management, including workload isolation, is configured on a system. A simple workload management configuration can manage data loads and user queries.

In the absence of workload isolation, requests operate in the shared pool of resources. Access to resources in the shared pool is not guaranteed and is assigned on an importance basis.

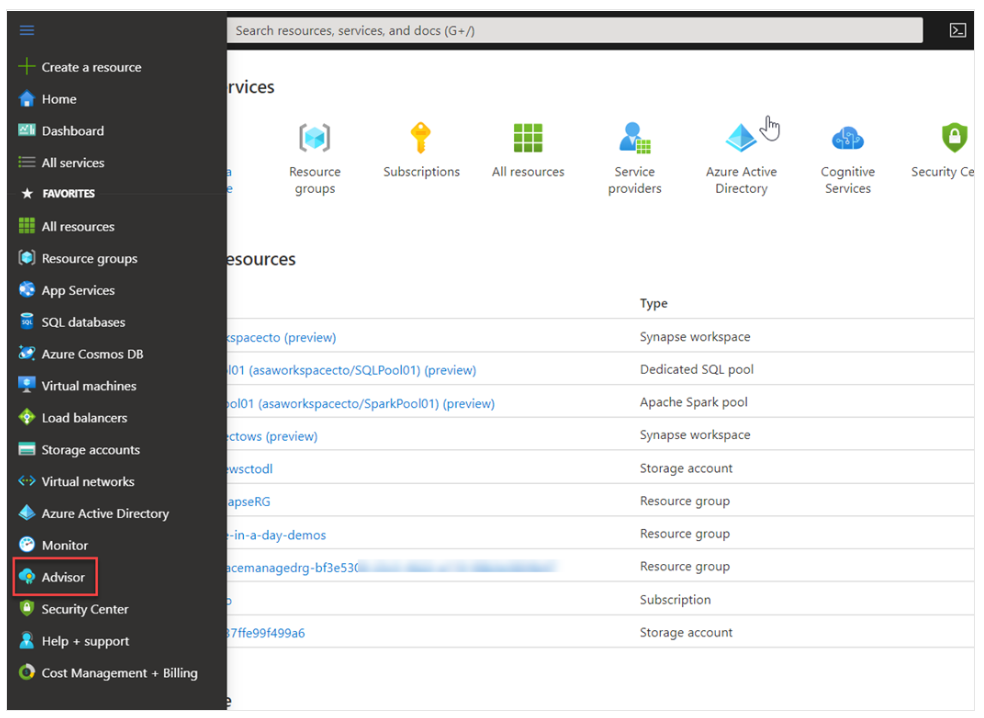
Given the workload requirements provided by Tailwind Traders, you decide to create a new workload group called CEODemo to reserve resources for queries executed by the CEO.

<https://learn.microsoft.com/en-us/training/modules/manage-monitor-data-warehouse-activities-azure-synapse-analytics/4-manage-workloads>

# Use Azure Advisor to review recommendations

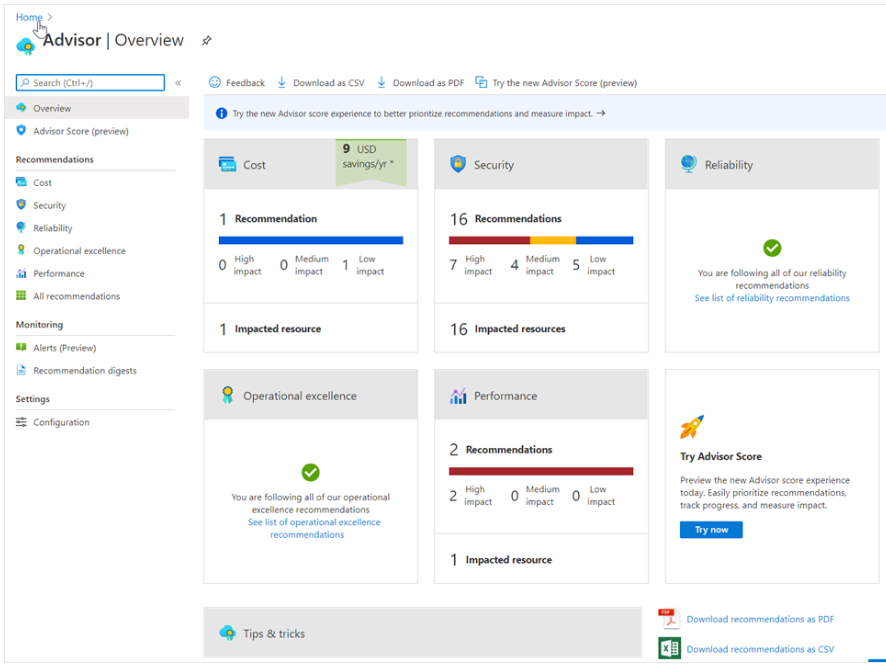
Azure Advisor provides you with personalized messages that provide information on best practices to optimize the setup of your Azure services. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, Reliability (formerly called High availability), and security of your Azure resources.

The Advisor may appear when you log into the Azure portal, but you can also access the Advisor by selecting Advisor in the navigation menu.

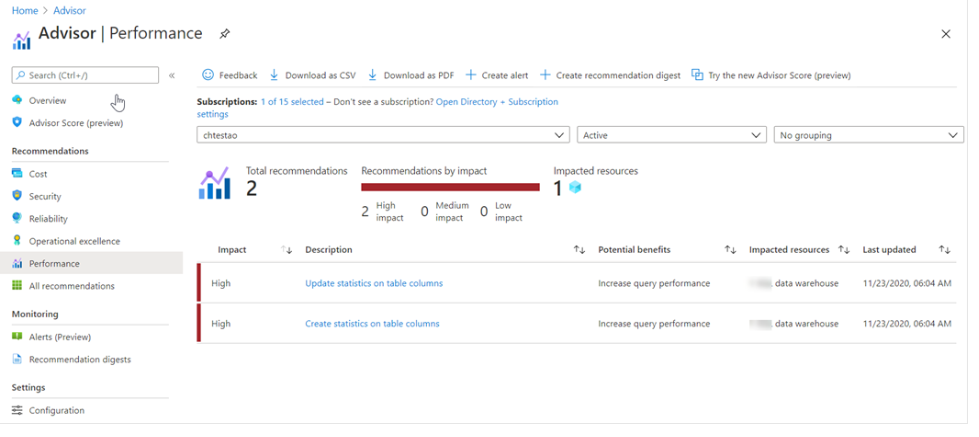


On accessing Advisor, a dashboard is presented that provides recommendations in the following areas:

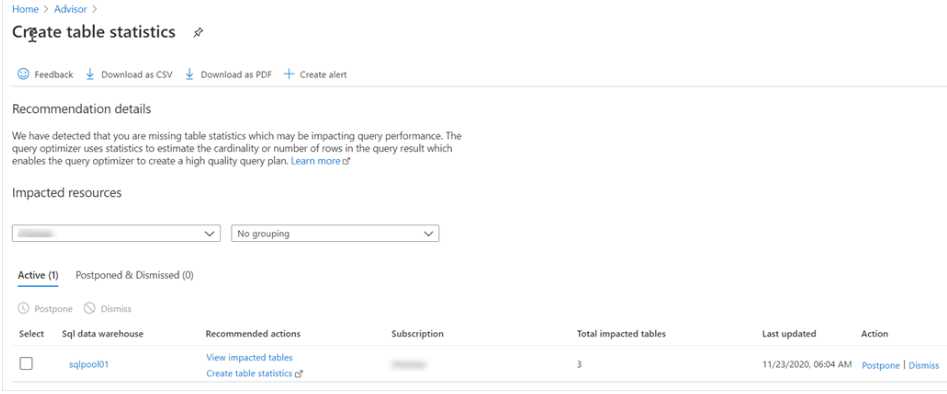
* Cost
* Security
* Reliability
* Operational excellence
* Performance



You can click on any of the dashboard items for more information. In the following example, the performance dashboard item is showing more information on two high impact items in Azure Synapse Analytics.



You can also click on each item to get even more information that can help you resolve the issue. In the following example, this is the information that is shown when clicking on the **Create statistics on table columns** recommendation.

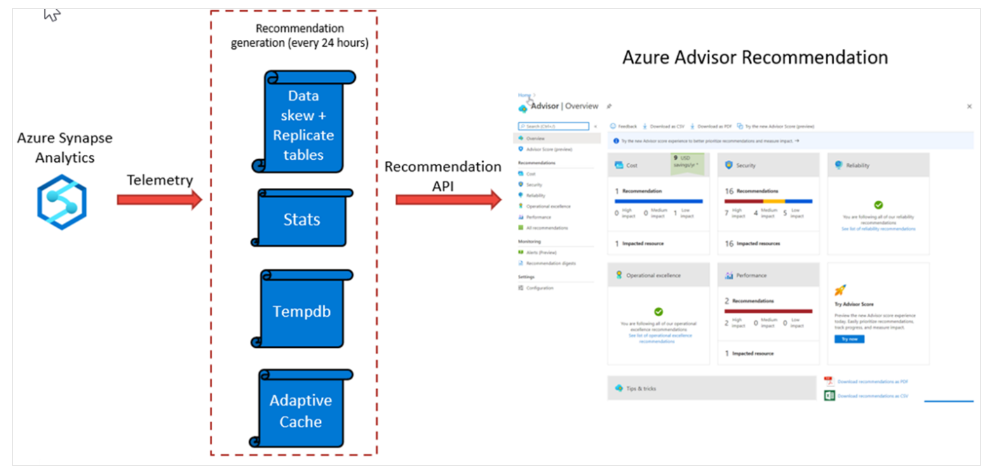


In this screen, you can click on the **view impacted tables** to see which tables are being impacted specifically, and there are also links to the help in the Azure documentation that you can use to get more understanding of the issue.

## **How Azure Synapse Analytics works with Azure Advisor**

Azure Advisor recommendations are free, and the recommendations are based on telemetry data that is generated by Azure Synapse Analytics. The telemetry data that is captured by Azure Synapse Analytics include

* Data Skew and replicated table information.
* Column statistics data.
* TempDB utilization data.
* Adaptive Cache.



Azure Advisor recommendations are checked every 24 hours, as the recommendation API is queried against the telemetry generated from with Azure Synapse Analytics, and the recommendation dashboards are then updated to reflect the information that the telemetry has generated. This can then be viewed in the Azure Advisor dashboard.

# Use dynamic management views to identify and troubleshoot query performance

Dynamic Management Views provide a programmatic experience for monitoring the Azure Synapse Analytics SQL pool activity by using the Transact-SQL language. The views that are provided, not only enable you to troubleshoot and identify performance bottlenecks with the workloads working on your system, but they are also used by other services such as Azure Advisor to provide recommendations about Azure Synapse Analytics.

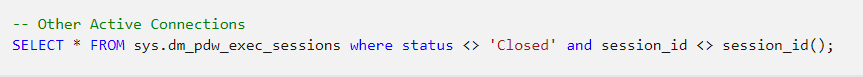
There are over 90 Dynamic Management Views that can queried against dedicated SQL pools to retrieve information about the following areas of the service:

* Connection information and activity
* SQL execution requests and queries
* Index and statistics information
* Resource blocking and locking activity
* Data movement service activity
* Errors

The following is an example of monitoring query execution of the Azure Synapse Analytics SQL pools. The first step involves checking the connections against the server first, before checking the query execution activity.

## **Monitoring connections**

All logins to your data warehouse are logged to sys.dm\_pdw\_exec\_sessions. The session\_id is the primary key and is assigned sequentially for each new logon.

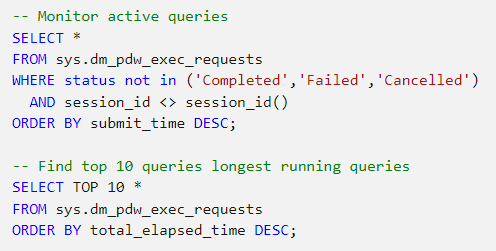


## **Monitor query execution**

All queries executed on SQL pool are logged to sys.dm\_pdw\_exec\_requests. The request\_id uniquely identifies each query and is the primary key for this DMV. The request\_id is assigned sequentially for each new query and is prefixed with QID, which stands for query ID. Querying this DMV for a given session\_id shows all queries for a given logon.

## Step 1

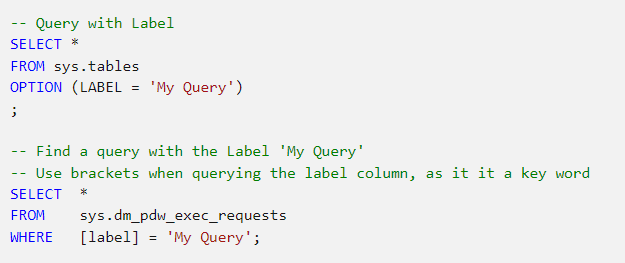
The first step is to identify the query you want to investigate



From the preceding query results, **note the Request ID** of the query that you would like to investigate.

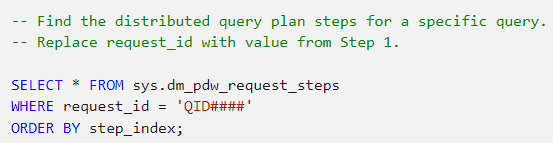
Queries in the **Suspended** state can be queued due to a large number of active running queries. These queries also appear in the sys.dm\_pdw\_waits waits query with a type of UserConcurrencyResourceType. For information on concurrency limits, see Memory and concurrency limits or Resource classes for workload management. Queries can also wait for other reasons such as for object locks. If your query is waiting for a resource, see Investigating queries waiting for resources further down in this article.

To simplify the lookup of a query in the sys.dm\_pdw\_exec\_requests table, use LABEL to assign a comment to your query, which can be looked up in the sys.dm\_pdw\_exec\_requests view.



## Step 2

Use the Request ID to retrieve the queries distributed SQL (DSQL) plan from sys.dm\_pdw\_request\_steps



When a DSQL plan is taking longer than expected, the cause can be a complex plan with many DSQL steps or just one step taking a long time. If the plan is many steps with several move operations, consider optimizing your table distributions to reduce data movement.

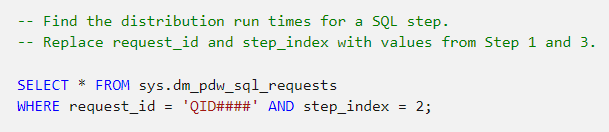
The [Table distribution](https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-tables-distribute) article explains why data must be moved to solve a query. The article also explains some distribution strategies to minimize data movement.

To investigate further details about a single step, the operation\_type column of the long-running query step and note the **Step Index**:

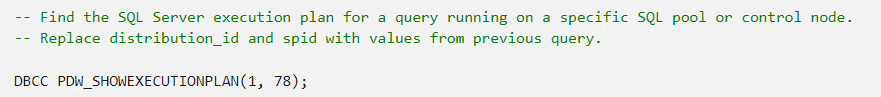
* Proceed with Step 3 for **SQL operations**: OnOperation, RemoteOperation, ReturnOperation.
* Proceed with Step 4 for **Data Movement operations**: ShuffleMoveOperation, BroadcastMoveOperation, TrimMoveOperation, PartitionMoveOperation, MoveOperation, CopyOperation.

## Step 3

Use the Request ID and the Step Index to retrieve details from sys.dm\_pdw\_sql\_requests, which contains execution information of the query step on all of the distributed databases.

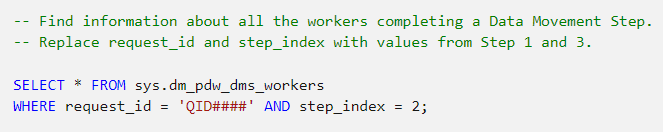


When the query step is running, DBCC PDW\_SHOWEXECUTIONPLAN can be used to retrieve the SQL Server estimated plan from the SQL Server plan cache for the step running on a particular distribution.



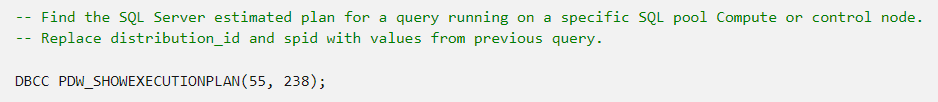
## Step 4

Use the Request ID and the Step Index to retrieve information about a data movement step running on each distribution from sys.dm\_pdw\_dms\_workers.



* Check the total\_elapsed\_time column to see if a particular distribution is taking longer than others for data movement.
* For the long-running distribution, check the rows\_processed column to see if the number of rows being moved from that distribution is larger than others. If so, this finding might indicate skew of your underlying data. One cause for data skew is distributing on a column with many NULL values (whose rows will all land in the same distribution). Prevent slow queries by avoiding distribution on these types of columns or filtering your query to eliminate NULLs when possible.

If the query is running, you can use DBCC PDW\_SHOWEXECUTIONPLAN to retrieve the SQL Server estimated plan from the SQL Server plan cache for the currently running SQL Step within a particular distribution.



Dynamic Management Views (DMV) only contains 10,000 rows of data. On heavily utilized systems this means that data held in this table may be lost with hours, or even minutes as data is managed in a first in, first out system. As a result you can potentially lose meaningful information that can help you diagnose query performance issues on your system. In this situation, you should use the Query Store.

You can also monitor additional aspects of Azure Synapse SQL pools including:

* Monitoring waits
* Monitoring tempdb
* Monitoring memory
* Monitoring transaction log
* Monitoring PolyBase

You can view information about [monitoring these areas here](https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-manage-monitor#monitor-waiting-queries)

