SQL DW Management and Monitoring (ADW In-A-Day Lab 03)

Contents

SQL DW Management and Monitoring (ADW In-A-Day Lab 03) 1

Overview 1

Pre-requisites 1

Resource Monitoring in Azure Monitor 1

Azure Data Studio SQLDW Dashboard (Azure SQL Data Warehouse Insights) 2

Azure SQL Data Warehouse Table and Statistics Queries 3

Create User-defined Restore Points 7

Maintenance Window Scheduling, Service Health, Service Health Alerts 14

Querying ADW Diagnostic Logs using Azure Monitor 18

Overview

This module will walk you through a variety of tools and techniques for monitoring and managing your Azure Data Warehouse.

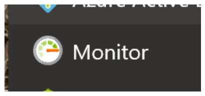
Pre-requisites

- Azure Portal Access
- Azure SQL Data Warehouse
- SQL Server Management Studio
- Azure Data Studio

Resource Monitoring in Azure Monitor

1. Logon to Azure Portal (portal.azure.com) using your credentials

2. Click on Monitor from the list of services.

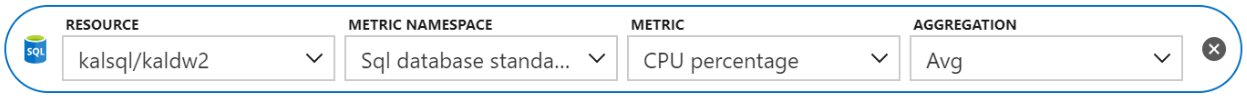


3. Click on 'Explore Metrics'

4. Click on 'Add metric' and select Subscription and Resource Group names from the drop-down.

5. Select your DW instance from the list of resources displayed.

6. Select 'CPU percentage' as the metric from the drop-down as shown below:


RESOURCE: kalsql/kaldw2
METRIC NAMESPACE: Sql database standa...
METRIC: CPU percentage
AGGREGATION: Avg

7. This will display the CPU percentage (Avg) chart for the last 24 hours.

8. Click on 'Add metric' again and 'Data IO Percentage' to the chart.

9. Repeat the process to add more metrics to the chart.
10. Click on the Time Range drop-down and change the time range from 'Last 24 hours' to 'Last hour'. Observe the chart data range change to display last hour data.

11. Click on 'Add chart' and select Subscription, Resource Group, Resource and Metric as before. You will observe a new chart created displaying the data for the metric you selected.

Azure Data Studio SQLDW Dashboard (Azure SQL Data Warehouse Insights)

1. Open Azure Data Studio application.
   Azure Data Studio (ADS) is a cross-platform database tool for monitoring and managing various data platforms, including Azure SQL DW.

2. Click on 'Extensions' icon from the list of icons on the left side ribbon of the window.

3. Select 'Azure SQL Data Warehouse Insights' from the list of extensions. (Note: Enter 'SQL' in the search window, if this extension is not showing up in the list)

4. Click 'Install' and wait for the extension installation to complete. (Note: Review the details of the extension to understand all the reports displayed by this extension. You can also click on the github link available to review the sql code for each of these reports)

5. Click on the 'Servers' icon on the left and click on 'Add Connection'.

6. Expand 'Databases' node in the explorer window and right-click on your database and click 'Manage. This will show Database Dashboard with details on your DW instance and tables.

7. Click on 'SQL DW Dashboard' tab in the details pane. Observe the reports on various metrics viz. Latest Backup date, User Activities, Data Distribution etc. (Note: Some reports maybe empty if there are no results to display.)

8. Observe the Data Distribution report. This shows data size in each of the 60 distributions. What can you infer from this report?

Azure SQL Data Warehouse Table and Statistics Queries

1. Open SQL Server Management Studio (SSMS). Connect to the your SQL DW instance.

2. Expand Databases node, right-click on your data warehouse database and click ‘New Query’. Execute the following query in your query window to create a useful View.

```
CREATE VIEW dbo.vTableSizes
AS
WITH base
AS
(
SELECT
 GETDATE()                                                              AS  [execution_time]
, DB_NAME()                                                             AS  [database_name]
, s.name                                                                AS  [schema_name]
, t.name                                                                AS  [table_name]
, QUOTENAME(s.name)+'.'+QUOTENAME(t.name)                               AS  [two_part_name]
, nt.[name]                                                             AS  [node_table_name]
, ROW_NUMBER() OVER(PARTITION BY nt.[name] ORDER BY (SELECT NULL))      AS
[node_table_name_seq]
, tp.[distribution_policy_desc]                                         AS
[distribution_policy_name]
, c.[name]                                                              AS  [distribution_column]
, nt.[distribution_id]                                                  AS  [distribution_id]
, i.[type]                                                              AS  [index_type]
, i.[type_desc]                                                         AS  [index_type_desc]
, nt.[pdw_node_id]                                                      AS  [pdw_node_id]
, pn.[type]                                                             AS  [pdw_node_type]
, pn.[name]                                                             AS  [pdw_node_name]
, di.name                                                               AS  [dist_name]
, di.position                                                           AS  [dist_position]
, nps.[partition_number]                                                AS  [partition_nmbr]
, nps.[reserved_page_count]                                             AS
[reserved_space_page_count]
, nps.[reserved_page_count] - nps.[used_page_count]                     AS
[unused_space_page_count]
, nps.[in_row_data_page_count]
    + nps.[row_overflow_used_page_count]
    + nps.[lob_used_page_count]                                         AS  [data_space_page_count]
, nps.[reserved_page_count]
 - (nps.[reserved_page_count] - nps.[used_page_count])
 - ([in_row_data_page_count]
         + [row_overflow_used_page_count]+[lob_used_page_count])        AS
[index_space_page_count]
, nps.[row_count]                                                       AS  [row_count]
from
    sys.schemas s
INNER JOIN sys.tables t
    ON s.[schema_id] = t.[schema_id]
INNER JOIN sys.indexes i
    ON  t.[object_id] = i.[object_id]
    AND i.[index_id] <= 1
INNER JOIN sys.pdw_table_distribution_properties tp
    ON t.[object_id] = tp.[object_id]
INNER JOIN sys.pdw_table_mappings tm
    ON t.[object_id] = tm.[object_id]
INNER JOIN sys.pdw_nodes_tables nt
    ON tm.[physical_name] = nt.[name]
INNER JOIN sys.dm_pdw_nodes pn
    ON  nt.[pdw_node_id] = pn.[pdw_node_id]
INNER JOIN sys.pdw_distributions di
    ON  nt.[distribution_id] = di.[distribution_id]
```

```
INNER JOIN sys.dm_pdw_nodes_db_partition_stats nps
    ON nt.[object_id] = nps.[object_id]
    AND nt.[pdw_node_id] = nps.[pdw_node_id]
    AND nt.[distribution_id] = nps.[distribution_id]
LEFT OUTER JOIN (select * from sys.pdw_column_distribution_properties where distribution_ordinal =
1) cdp
    ON t.[object_id] = cdp.[object_id]
LEFT OUTER JOIN sys.columns c
    ON cdp.[object_id] = c.[object_id]
    AND cdp.[column_id] = c.[column_id]
)
, size
AS
(
SELECT
   [execution_time]
,  [database_name]
,  [schema_name]
,  [table_name]
,  [two_part_name]
,  [node_table_name]
,  [node_table_name_seq]
,  [distribution_policy_name]
,  [distribution_column]
,  [distribution_id]
,  [index_type]
,  [index_type_desc]
,  [pdw_node_id]
,  [pdw_node_type]
,  [pdw_node_name]
,  [dist_name]
,  [dist_position]
,  [partition_nmbr]
,  [reserved_space_page_count]
,  [unused_space_page_count]
,  [data_space_page_count]
,  [index_space_page_count]
,  [row_count]
,  ([reserved_space_page_count] * 8.0)                        AS
[reserved_space_KB]
,  ([reserved_space_page_count] * 8.0)/1000                   AS
[reserved_space_MB]
,  ([reserved_space_page_count] * 8.0)/1000000                AS
[reserved_space_GB]
,  ([reserved_space_page_count] * 8.0)/1000000000             AS
[reserved_space_TB]
,  ([unused_space_page_count]   * 8.0)                        AS
[unused_space_KB]
,  ([unused_space_page_count]   * 8.0)/1000                   AS
[unused_space_MB]
,  ([unused_space_page_count]   * 8.0)/1000000                AS
[unused_space_GB]
,  ([unused_space_page_count]   * 8.0)/1000000000             AS
[unused_space_TB]
,  ([data_space_page_count]     * 8.0)                        AS
[data_space_KB]
,  ([data_space_page_count]     * 8.0)/1000                   AS
[data_space_MB]
,  ([data_space_page_count]     * 8.0)/1000000                AS
[data_space_GB]
,  ([data_space_page_count]     * 8.0)/1000000000             AS
[data_space_TB]
```

```
,  ([index_space_page_count]  * 8.0)                                AS
[index_space_KB]
,  ([index_space_page_count]  * 8.0)/1000                           AS
[index_space_MB]
,  ([index_space_page_count]  * 8.0)/1000000                        AS
[index_space_GB]
,  ([index_space_page_count]  * 8.0)/1000000000                     AS
[index_space_TB]
FROM base
)
SELECT *
FROM size
;
GO
```

9. Execute the following query in the query window. This will show table space summary information.

```
SELECT
     database_name
,    schema_name
,    table_name
,    distribution_policy_name
,      distribution_column
,    index_type_desc
,    COUNT(distinct partition_nmbr) as nbr_partitions
,    SUM(row_count)                 as table_row_count
,    SUM(reserved_space_GB)         as table_reserved_space_GB
,    SUM(data_space_GB)             as table_data_space_GB
,    SUM(index_space_GB)            as table_index_space_GB
,    SUM(unused_space_GB)           as table_unused_space_GB
FROM
    dbo.vTableSizes
GROUP BY
     database_name
,    schema_name
,    table_name
,    distribution_policy_name
,      distribution_column
,    index_type_desc
ORDER BY
    table_reserved_space_GB desc
```

10. Click ‘New Query’. Execute the following query in your query window. This will list tables with >10% skew.

```
    select two_part_name
       ,[distribution_column]
       ,(max(row_count * 1.000) - min(row_count * 1.000))/max(row_count * 1.000) as
'skew'
       ,max(row_count) as 'largest dist'
       ,min(row_count) as 'smallest dist'
       ,avg(row_count) as 'average dist'
       ,sum(row_count) as 'total row_count'
    from dbo.vTableSizes
    where row_count > 0
       AND distribution_policy_name = 'HASH'
    group by two_part_name, [distribution_column]
    having (max(row_count * 1.000) - min(row_count * 1.000))/max(row_count * 1.000)
>= .10
       order by 3 DESC -- two_part_name
```

3. Expand Databases node, right-click on your data warehouse database and click ‘New Query’. Execute the following query in your query window to create another useful View.

```
CREATE  VIEW dbo.vStats_Columns
AS
SELECT
        sm.[name]                           AS [schema_name]
,       tb.[name]                           AS [table_name]
,       st.[name]                           AS [stats_name]
,       st.[filter_definition]              AS [stats_filter_defiinition]
,       st.[has_filter]                     AS [stats_is_filtered]
,       STATS_DATE(st.[object_id],st.[stats_id])
                                            AS [stats_last_updated_date]
,             st.[user_created]                                  AS [user_created]
,       co.[name]                           AS [stats_column_name]
,       ty.[name]                           AS [column_type]
,       co.[max_length]                     AS [column_max_length]
,       co.[precision]                      AS [column_precision]
,       co.[scale]                          AS [column_scale]
,       co.[is_nullable]                    AS [column_is_nullable]
,       co.[collation_name]                 AS [column_collation_name]
,       QUOTENAME(sm.[name])+'.'+QUOTENAME(tb.[name])
                                            AS two_part_name
,       QUOTENAME(DB_NAME())+'.'+QUOTENAME(sm.[name])+'.'+QUOTENAME(tb.[name])
                                            AS three_part_name
,       QUOTENAME(sm.[name])+'.'+QUOTENAME(tb.[name])+'.'+QUOTENAME(st.[name])
                                            AS full_stats_name
--,           st.[stats_generation_method_desc]  AS [stats_generation_method_desc]
FROM    sys.objects                         AS ob
JOIN    sys.stats           AS st ON    ob.[object_id]      = st.[object_id]
JOIN    sys.stats_columns   AS sc ON    st.[stats_id]       = sc.[stats_id]
                            AND         st.[object_id]      = sc.[object_id]
JOIN    sys.columns         AS co ON    sc.[column_id]      = co.[column_id]
                            AND         sc.[object_id]      = co.[object_id]
JOIN    sys.types           AS ty ON    co.[user_type_id]   = ty.[user_type_id]
JOIN    sys.tables          AS tb ON  co.[object_id]        = tb.[object_id]
JOIN    sys.schemas         AS sm ON  tb.[schema_id]        = sm.[schema_id]
;
GO
```

11. Execute the following query in the query window. This will show table statistics information.

```
SELECT * FROM dbo.vStats_Columns ORDER BY 1,2,3
```

Create User-defined Restore Points

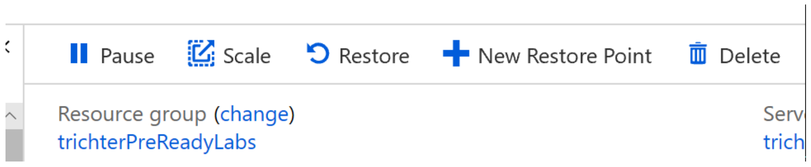
Azure SQL Data Warehouse includes automated database snapshots that can be leveraged to recover or copy a data warehouse to a previous state. These snapshots support an eight-hour recovery point objective (RPO) and are available to be used for 7 days. If you require a faster RPO or you require your snapshot to be available for longer than 7 days, you can manually trigger a snapshot to save the current database state. This is good practice to follow before and after large modifications to your data warehouse - it allows quicker recovery times in the event of any workload interruptions or user errors.

We will create a user-defined restore point, but not actually perform a restore due to the limitations of working in this lab environment and time constraints.

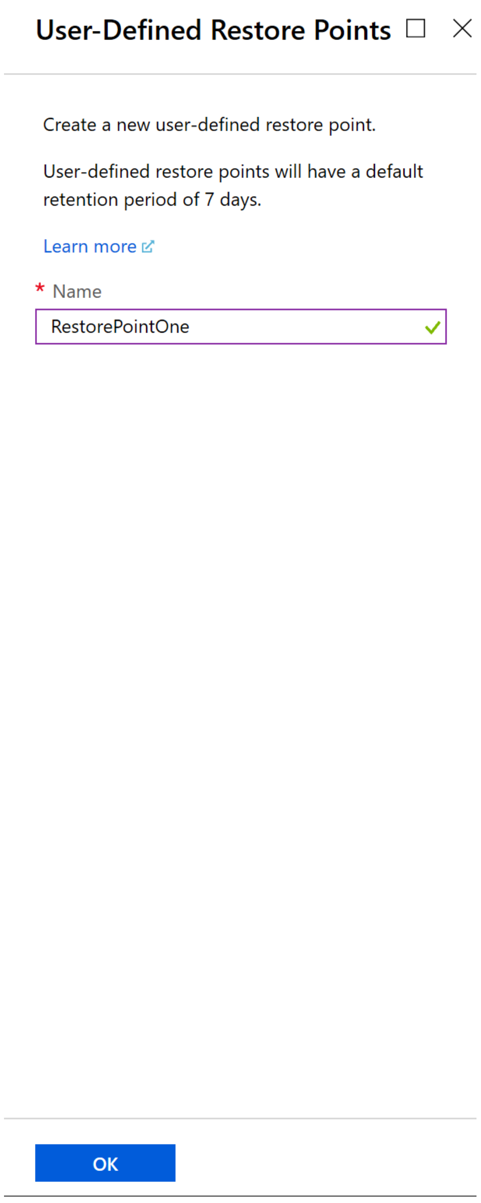
1. Logon to Azure Portal (portal.azure.com) using your credentials

2. Navigate to the Overview page of your Azure Data Warehouse.

3. Select New Restore Point



4. Specify a name for your restore point


User-Defined Restore Points

Create a new user-defined restore point.

User-defined restore points will have a default retention period of 7 days.

Learn more

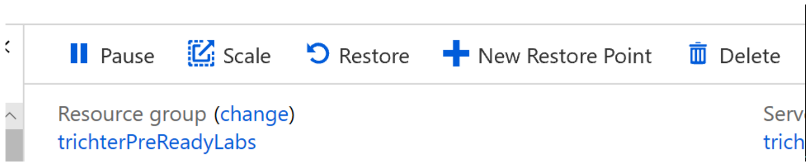
* Name

RestorePointOne

5. Click Ok

6. Navigate to the Overview page of your Azure Data Warehouse.

7. Select Restore

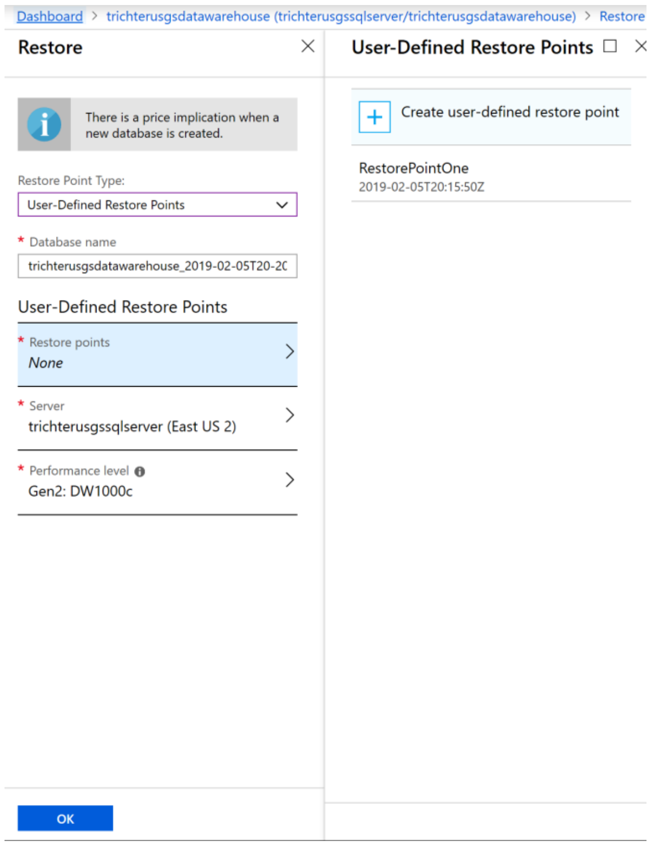


Resource group (change)
trichterPreReadyLabs

Serv
trich

8. Select Restore Point Type: User-Defined Restore Points
   Click Restore points to activate the User-Defined Restore Points blade

Notice your recently created user-defined restore point.


Dashboard > trichterusgsdatawarehouse (trichterusgssqlserver/trichterusgsdatawarehouse) > Restore

Restore

There is a price implication when a new database is created.

Restore Point Type: User-Defined Restore Points

* Database name: trichterusgsdatawarehouse_2019-02-05T20-2C

User-Defined Restore Points

* Restore points: *None*

* Server: trichterusgssqlserver (East US 2)

* Performance level: Gen2: DW1000c

OK

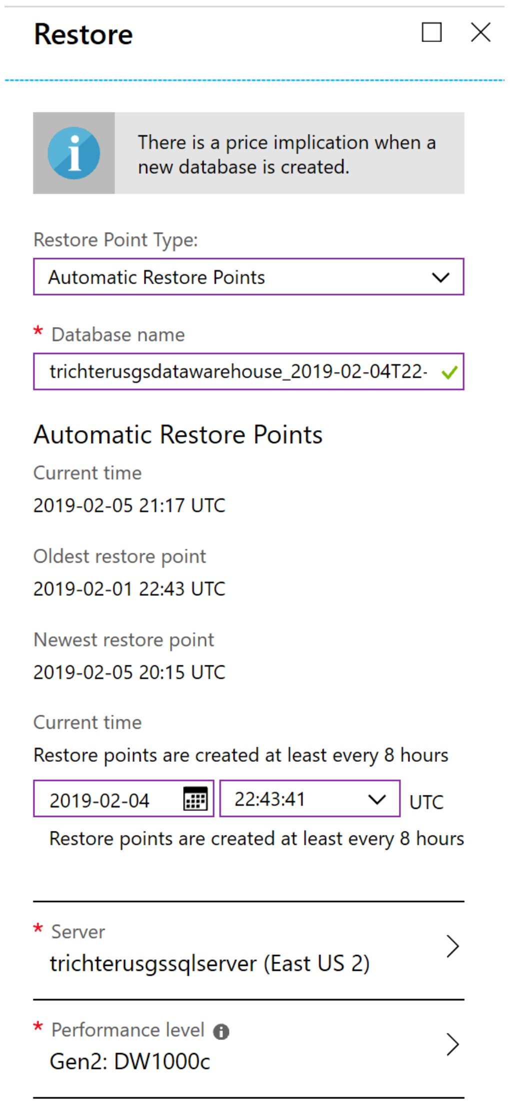
User-Defined Restore Points

Create user-defined restore point

RestorePointOne
2019-02-05T20:15:50Z

9. Click X to close the User-Defined Restore Points blade

10. Investigate the Automatic Restore Points
    Select Restore Point Type: Automatic Restore Points
    Notice the Oldest restore point and Newest restore point information
    Select the calendar control and click a **bold** date
    Notice that the time drop down is now populated with the restore point times available.



Restore

There is a price implication when a new database is created.

Restore Point Type: Automatic Restore Points

* Database name: trichterusgsdatawarehouse_2019-02-04T22- ✓

Automatic Restore Points

Current time
2019-02-05 21:17 UTC

Oldest restore point
2019-02-01 22:43 UTC

Newest restore point
2019-02-05 20:15 UTC

Current time
Restore points are created at least every 8 hours

2019-02-04 22:43:41 UTC
Restore points are created at least every 8 hours

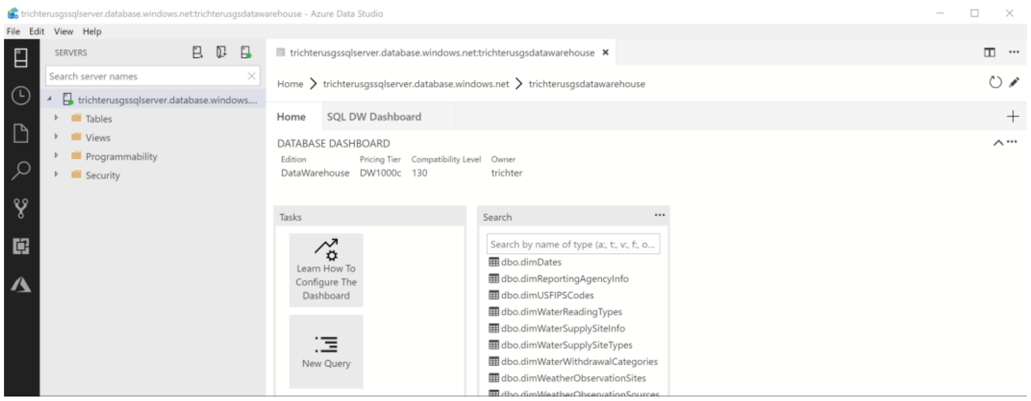
* Server
trichterusgssqlserver (East US 2)

* Performance level
Gen2: DW1000c

11. Click x to close the Restore blade.
    We will not perform an actual restore due to time and resource constraints of this lab environment.

12. Let's see how restore points can be queried using Azure Data Studio
    Open your Azure Data Studio application and connect to your Azure Data Warehouse
    (You must have completed the prior module Azure SQL Data Studio SQLDW Dashboard)

13. Right click your Azure Data Warehouse server and select Manage


trichterusgssqlserver.database.windows.net:trichterusgsdatawarehouse - Azure Data Studio

File Edit View Help

SERVERS

Search server names

trichterusgssqlserver.database.windows....
- Tables
- Views
- Programmability
- Security

trichterusgssqlserver.database.windows.net:trichterusgsdatawarehouse

Home > trichterusgssqlserver.database.windows.net > trichterusgsdatawarehouse

Home | SQL DW Dashboard

DATABASE DASHBOARD

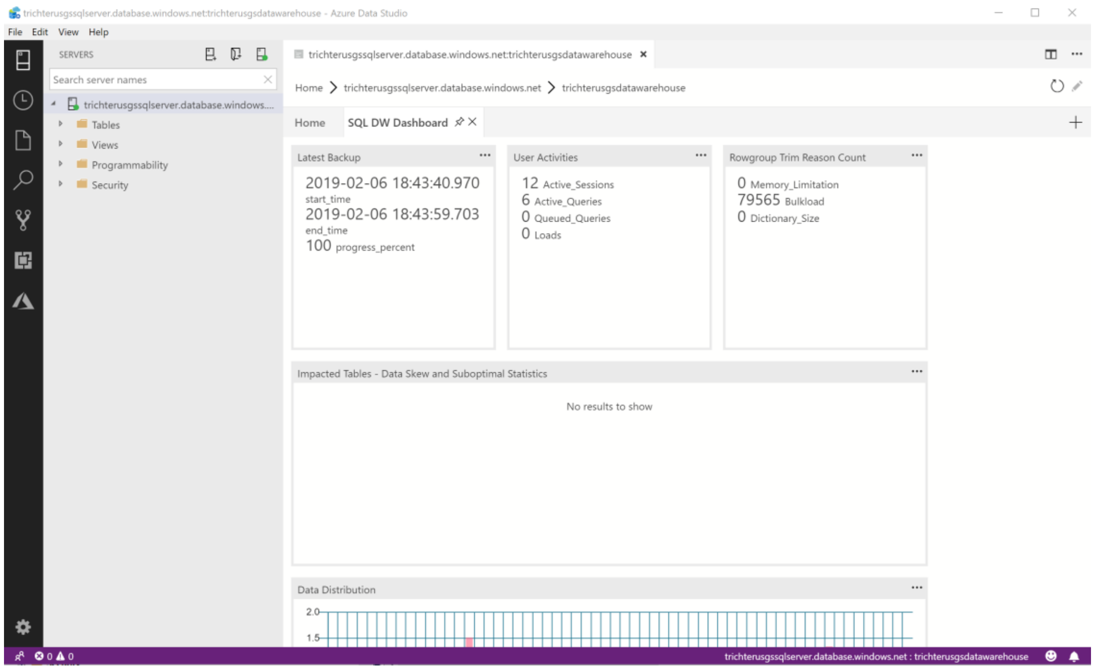
Edition	Pricing Tier	Compatibility Level	Owner
DataWarehouse	DW1000c	130	trichter

Tasks: Learn How To Configure The Dashboard; New Query

Search

Search by name of type (a:, t:, v:, f:, o...
- dbo.dimDates
- dbo.dimReportingAgencyInfo
- dbo.dimUSFIPSCodes
- dbo.dimWaterReadingTypes
- dbo.dimWaterSupplySiteInfo
- dbo.dimWaterSupplySiteTypes
- dbo.dimWaterWithdrawalCategories
- dbo.dimWeatherObservationSites

14. Select the SQL DW Dashboard tab
    Notice the default summary information provided for user activities, CCI Health (Rowgroup Trim Reason Count), Tables that are being impacted by skew or by statistics issues, Etc.
    The queries that run behind these visualizations are available to you. Let's look into the Latest Backup information.


trichterusgssqlserver.database.windows.net:trichterusgsdatawarehouse - Azure Data Studio

File Edit View Help

SERVERS

Search server names

trichterusgssqlserver.database.windows....
- Tables
- Views
- Programmability
- Security

trichterusgssqlserver.database.windows.net:trichterusgsdatawarehouse

Home > trichterusgssqlserver.database.windows.net > trichterusgsdatawarehouse

Home | SQL DW Dashboard

Latest Backup

2019-02-06 18:43:40.970
start_time
2019-02-06 18:43:59.703
end_time
100 progress_percent

User Activities

12 Active_Sessions
6 Active_Queries
0 Queued_Queries
0 Loads

Rowgroup Trim Reason Count

0 Memory_Limitation
79565 Bulkload
0 Dictionary_Size

Impacted Tables - Data Skew and Suboptimal Statistics

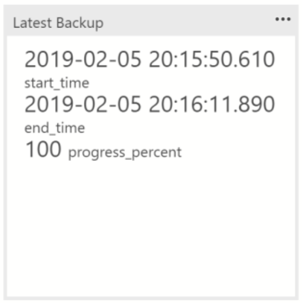
No results to show

Data Distribution

2.0
1.5

trichterusgssqlserver.database.windows.net : trichterusgsdatawarehouse

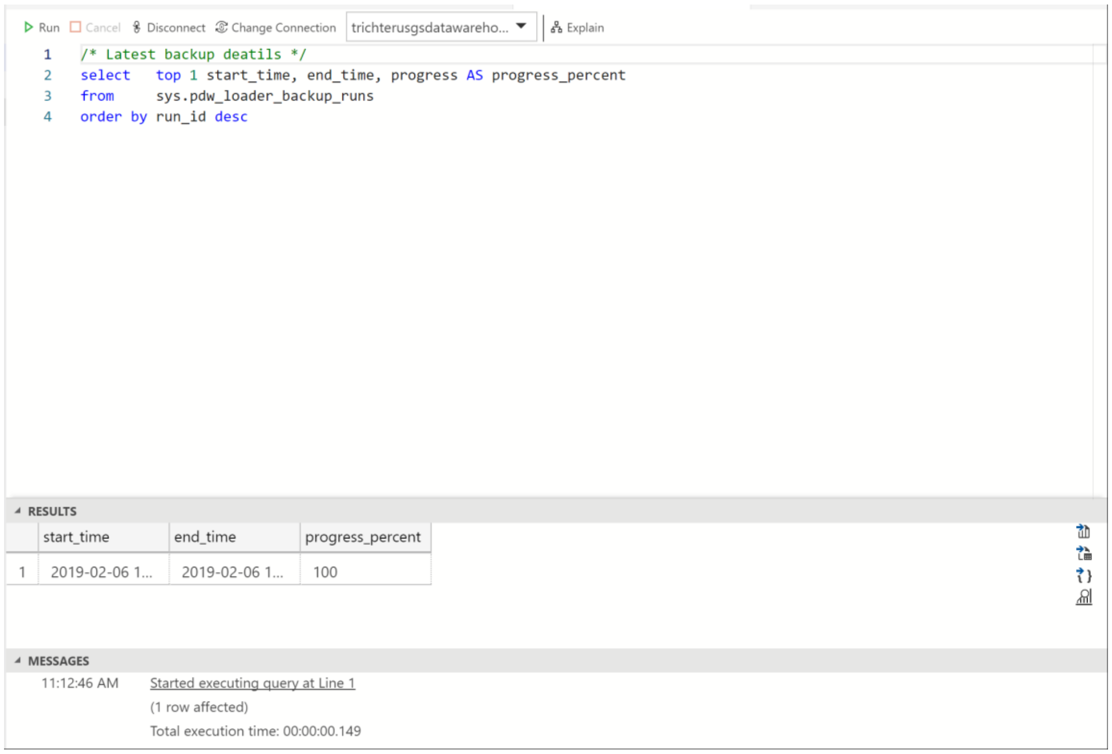
15. Select the ellipses (…) in the right-hand corner of the Latest Backup visualization


Latest Backup

2019-02-05 20:15:50.610
start_time
2019-02-05 20:16:11.890
end_time
100 progress_percent

16. Select Run Query

A new tab will open and execute the query that populates the Latest Backup visualization and the result set will be displayed.



```
/* Latest backup deatils */
select   top 1 start_time, end_time, progress AS progress_percent
from     sys.pdw_loader_backup_runs
order by run_id desc
```

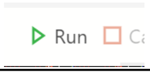
RESULTS

	start_time	end_time	progress_percent
1	2019-02-06 1...	2019-02-06 1...	100

MESSAGES

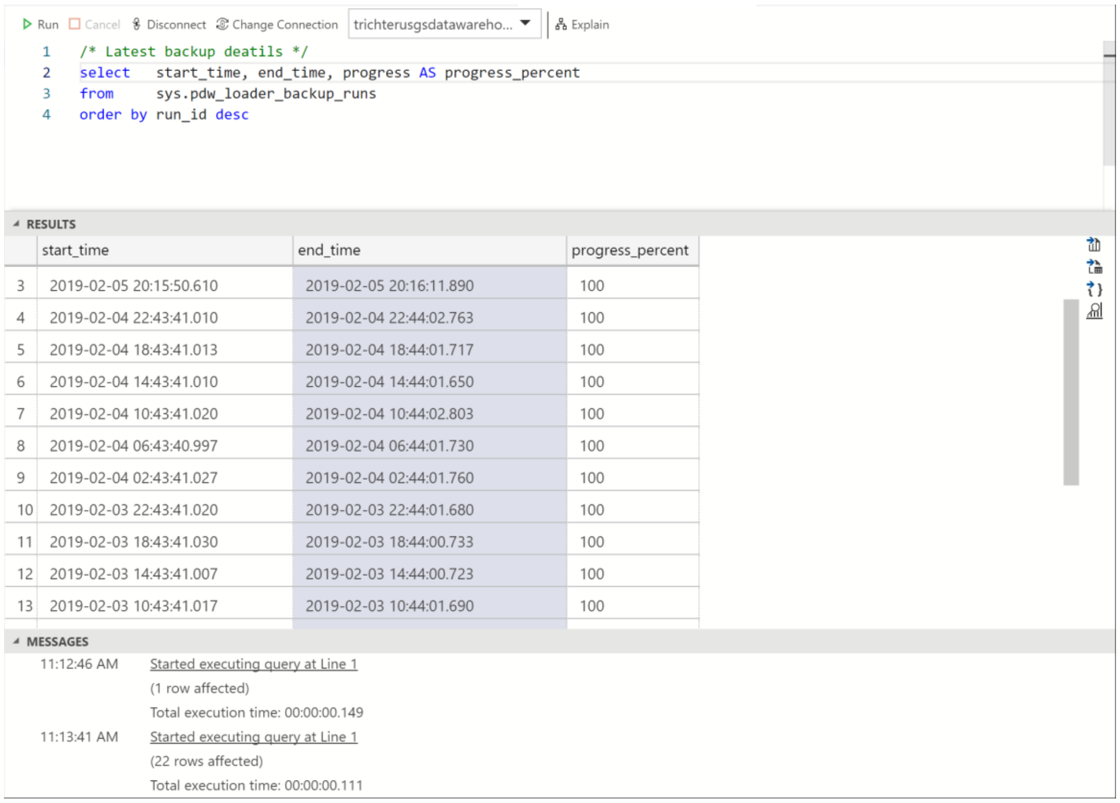
```
11:12:46 AM    Started executing query at Line 1
               (1 row affected)
               Total execution time: 00:00:00.149
```

17. To review all of the backups that are available, modify the query by removing the top 1 clause from the select statement and click run



```
select   start_time, end_time, progress AS
progress_percent
from     sys.pdw_loader_backup_runs
order by run_id desc
```

18. All of the available restore points will be listed. Notice that they do not have friendly names. Your user-defined restore point is likely at the top, check the date and time.



```sql
/* Latest backup deatils */
select   start_time, end_time, progress AS progress_percent
from     sys.pdw_loader_backup_runs
order by run_id desc
```

RESULTS

	start_time	end_time	progress_percent
3	2019-02-05 20:15:50.610	2019-02-05 20:16:11.890	100
4	2019-02-04 22:43:41.010	2019-02-04 22:44:02.763	100
5	2019-02-04 18:43:41.013	2019-02-04 18:44:01.717	100
6	2019-02-04 14:43:41.010	2019-02-04 14:44:01.650	100
7	2019-02-04 10:43:41.020	2019-02-04 10:44:02.803	100
8	2019-02-04 06:43:40.997	2019-02-04 06:44:01.730	100
9	2019-02-04 02:43:41.027	2019-02-04 02:44:01.760	100
10	2019-02-03 22:43:41.020	2019-02-03 22:44:01.680	100
11	2019-02-03 18:43:41.030	2019-02-03 18:44:00.733	100
12	2019-02-03 14:43:41.007	2019-02-03 14:44:00.723	100
13	2019-02-03 10:43:41.017	2019-02-03 10:44:01.690	100

MESSAGES

```
11:12:46 AM    Started executing query at Line 1
               (1 row affected)
               Total execution time: 00:00:00.149
11:13:41 AM    Started executing query at Line 1
               (22 rows affected)
               Total execution time: 00:00:00.111
```

19. Close the tab. Don't Save your changes

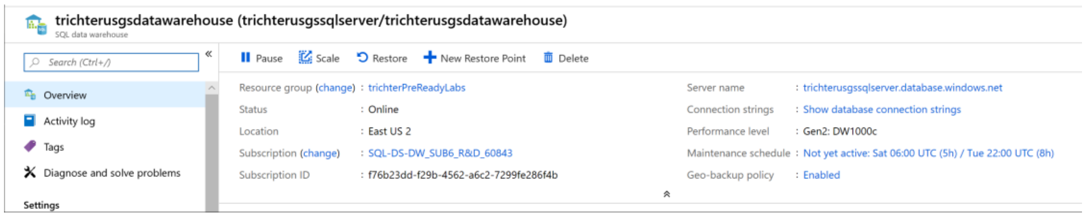
Maintenance Window Scheduling, Service Health, Service Health Alerts

By default, all newly created Azure SQL Data Warehouse instances have an eight-hour primary and secondary maintenance window applied during deployment. You can change the windows as soon deployment is complete. No maintenance will take place outside the specified maintenance windows without prior notification.

1. Logon to Azure Portal (portal.azure.com) using your credentials

2. Navigate to your data warehouse overview blade

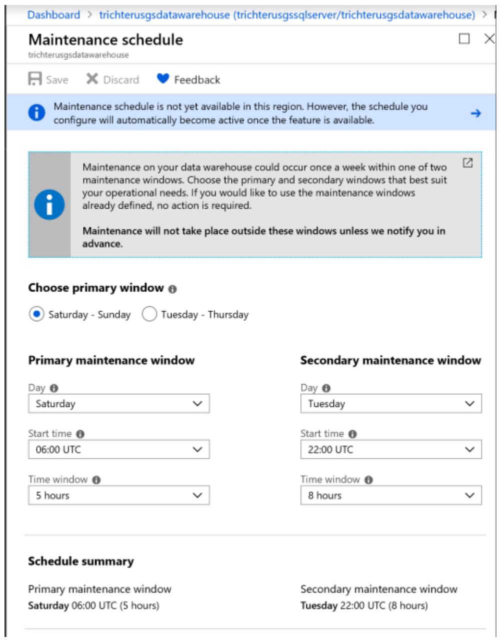
   Notice the Maintenance schedule text link
   The state of any maintenance will be displayed in this area. "Not yet active" indicates that the current day and time are not within an active maintenance window.



trichterusgsdatawarehouse (trichterusgssqlserver/trichterusgsdatawarehouse)
SQL data warehouse

Resource group (change)	trichterPreReadyLabs
Status	Online
Location	East US 2
Subscription (change)	SQL-DS-DW_SUB6_R&D_60843
Subscription ID	f76b23dd-f29b-4562-a6c2-7299fe286f4b
Server name	trichterusgssqlserver.database.windows.net
Connection strings	Show database connection strings
Performance level	Gen2: DW1000c
Maintenance schedule	Not yet active: Sat 06:00 UTC (5h) / Tue 22:00 UTC (8h)
Geo-backup policy	Enabled

3. Click on Not Yet Active text link to activate the Maintenance schedule blade


Dashboard > trichterusgsdatawarehouse (trichterusgssqlserver/trichterusgsdatawarehouse) >

Maintenance schedule
trichterusgsdatawarehouse

Maintenance schedule is not yet available in this region. However, the schedule you configure will automatically become active once the feature is available.

Maintenance on your data warehouse could occur once a week within one of two maintenance windows. Choose the primary and secondary windows that best suit your operational needs. If you would like to use the maintenance windows already defined, no action is required.

Maintenance will not take place outside these windows unless we notify you in advance.

Choose primary window

[✓ Saturday - Sunday] Tuesday - Thursday

Primary maintenance window

Day: Saturday

Start time: 06:00 UTC

Time window: 5 hours

Secondary maintenance window

Day: Tuesday

Start time: 22:00 UTC

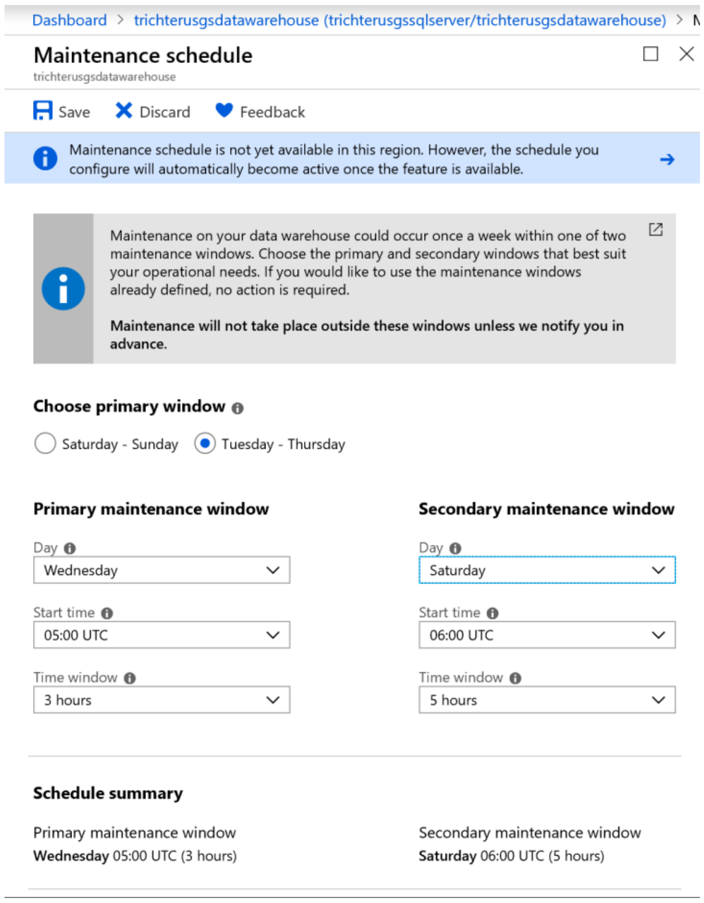
Time window: 8 hours

Schedule summary

Primary maintenance window
Saturday 06:00 UTC (5 hours)

Secondary maintenance window
Tuesday 22:00 UTC (8 hours)

4. Notice the various configuration options and configure a Primary and Secondary maintenance window


Dashboard > trichterusgsdatawarehouse (trichterusgssqlserver/trichterusgsdatawarehouse) > M

Maintenance schedule
trichterusgsdatawarehouse

Maintenance schedule is not yet available in this region. However, the schedule you configure will automatically become active once the feature is available.

Maintenance on your data warehouse could occur once a week within one of two maintenance windows. Choose the primary and secondary windows that best suit your operational needs. If you would like to use the maintenance windows already defined, no action is required.

Maintenance will not take place outside these windows unless we notify you in advance.

Choose primary window

Saturday - Sunday [✓ Tuesday - Thursday]

Primary maintenance window

Day: Wednesday
Start time: 05:00 UTC
Time window: 3 hours

Secondary maintenance window

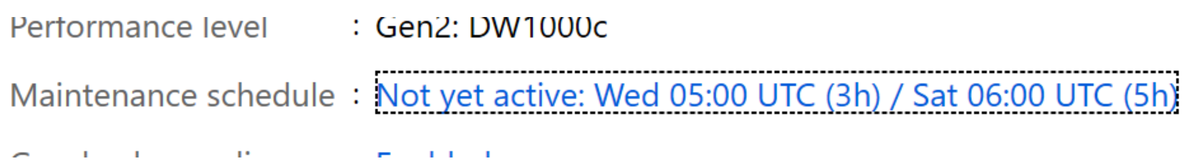
Day: Saturday
Start time: 06:00 UTC
Time window: 5 hours

Schedule summary

Primary maintenance window
Wednesday 05:00 UTC (3 hours)

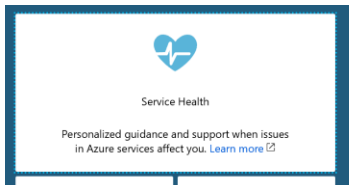
Secondary maintenance window
Saturday 06:00 UTC (5 hours)

5. Click Save
6. Return to your Overview blade and notice the new days and times listed for Maintenance schedule

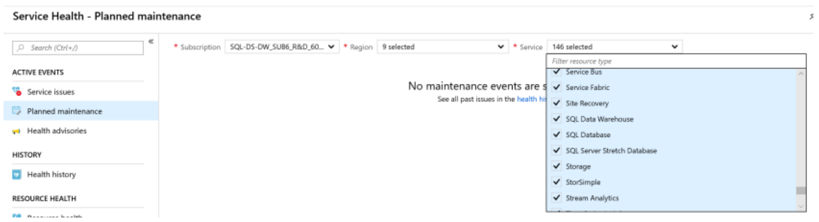

Performance level : Gen2: DW1000c

Maintenance schedule : Not yet active: Wed 05:00 UTC (3h) / Sat 06:00 UTC (5h)

7. Planned Maintenance Activities will appear in Service Health
   From the Dashboard select Service Health


Service Health
Personalized guidance and support when issues in Azure services affect you. Learn more

8. Click Planned Maintenance from the side menu to view Planned Maintenance events and notifications
   Filter for SQL Data Warehouse



Service Health - Planned maintenance

Search (Ctrl+/)

ACTIVE EVENTS
- Service issues
- Planned maintenance
- Health advisories

HISTORY
- Health history

RESOURCE HEALTH

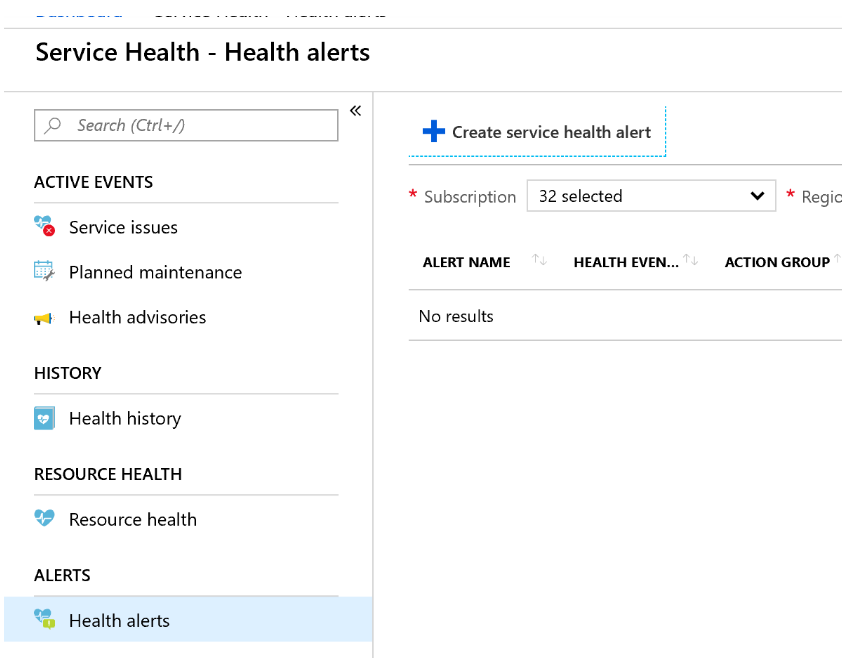
* Subscription: SQL-DS-DW_SUB6_R&D_60... * Region: 9 selected * Service: 146 selected

Filter resource type
- ✓ Service Bus
- ✓ Service Fabric
- ✓ Site Recovery
- ✓ SQL Data Warehouse
- ✓ SQL Database
- ✓ SQL Server Stretch Database
- ✓ Storage
- ✓ StorSimple
- ✓ Stream Analytics

No maintenance events are s

See all past issues in the health hi

9. To be alerted for Planned SQL Data Warehouse planned maintenance Select Health alerts and click Create service health alert



Service Health - Health alerts

Search (Ctrl+/)

ACTIVE EVENTS
- Service issues
- Planned maintenance
- Health advisories

HISTORY
- Health history

RESOURCE HEALTH
- Resource health

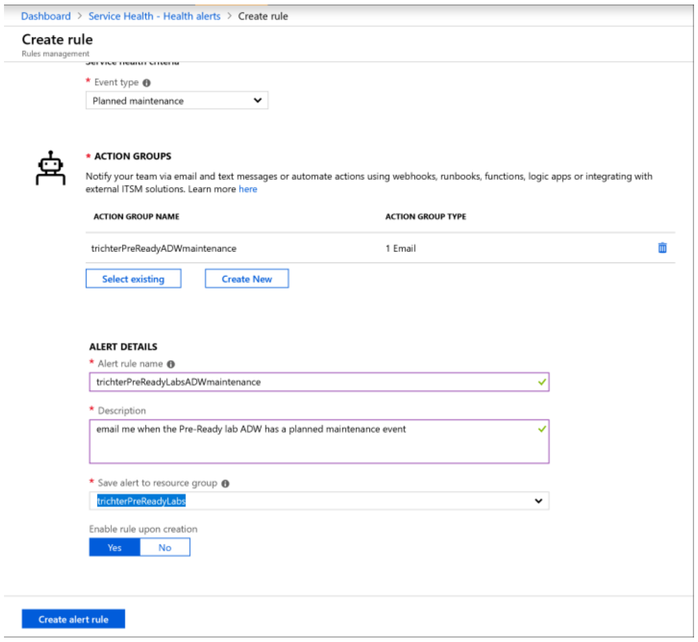
ALERTS
- Health alerts

➕ Create service health alert

* Subscription: 32 selected * Regic

ALERT NAME	HEALTH EVEN...	ACTION GROUP
No results		

10. Configure a new rule


Dashboard > Service Health - Health alerts > Create rule

Create rule

Rules management

* Event type: Planned maintenance

* ACTION GROUPS

Notify your team via email and text messages or automate actions using webhooks, runbooks, functions, logic apps or integrating with external ITSM solutions. Learn more here

ACTION GROUP NAME	ACTION GROUP TYPE
trichterPreReadyADWmaintenance	1 Email

Select existing | Create New

ALERT DETAILS

* Alert rule name: trichterPreReadyLabsADWmaintenance

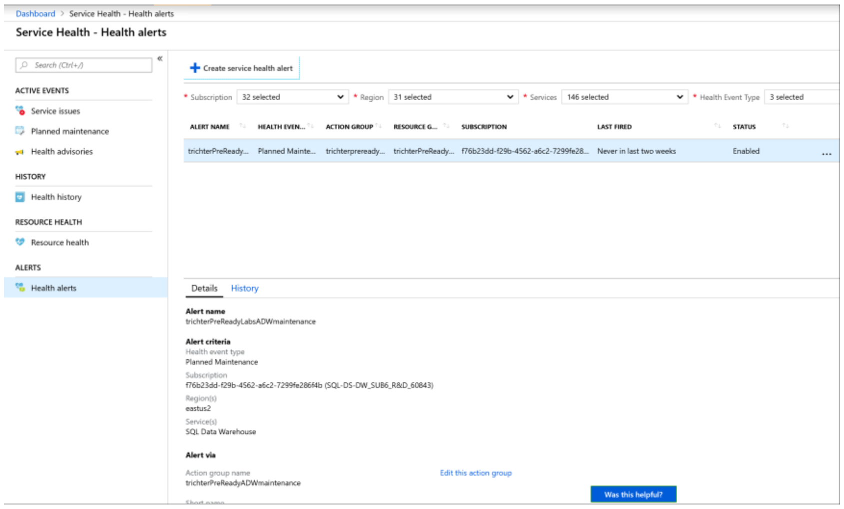
* Description: email me when the Pre-Ready lab ADW has a planned maintenance event

* Save alert to resource group: trichterPreReadyLabs

Enable rule upon creation: [✓ Yes] No

Create alert rule

11. Click Create alert rule and refresh the Health alerts blade


Dashboard > Service Health - Health alerts

Service Health - Health alerts

Search (Ctrl+/)

ACTIVE EVENTS: Service issues, Planned maintenance, Health advisories

HISTORY: Health history

RESOURCE HEALTH: Resource health

ALERTS: Health alerts

Create service health alert

* Subscription: 32 selected | * Region: 31 selected | * Services: 146 selected | * Health Event Type: 3 selected

ALERT NAME	HEALTH EVEN...	ACTION GROUP	RESOURCE G...	SUBSCRIPTION	LAST FIRED	STATUS
trichterPreReady...	Planned Mainte...	trichterpreready...	trichterPreReady...	f76b23dd-f29b-4562-a6c2-7299fe28...	Never in last two weeks	Enabled

Details | History

Alert name
trichterPreReadyLabsADWmaintenance

Alert criteria
Health event type: Planned Maintenance
Subscription: f76b23dd-f29b-4562-a6c2-7299fe286f4b (SQL-DS-DW_SUB6_R&D_60843)
Region(s): eastus2
Service(s): SQL Data Warehouse

Alert via
Action group name: trichterPreReadyADWmaintenance
Edit this action group

Was this helpful?

12. You will be able to view the details and history of this health alert from this blade.

13. Disable your alert by clicking the ellipses(…) and selecting Disable

Querying ADW Diagnostic Logs using Azure Monitor

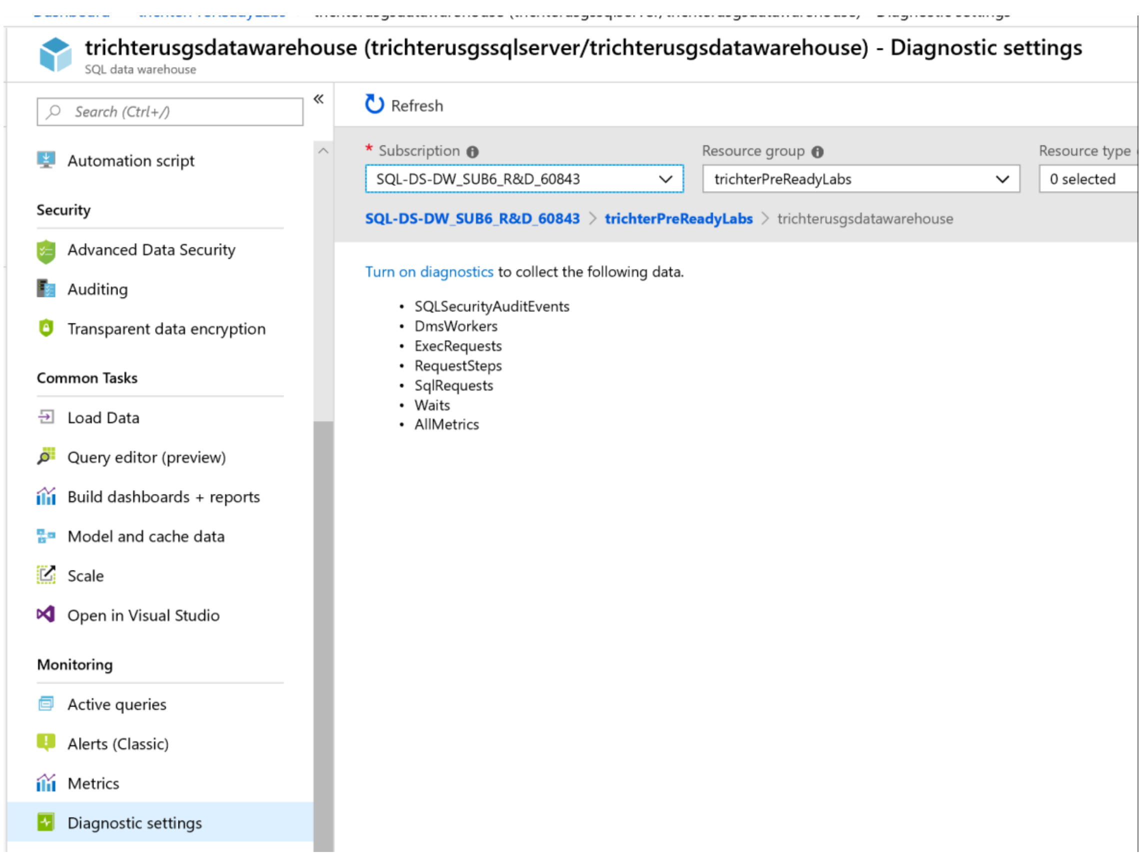
SQL Data Warehouse (SQL DW) now enables enhanced insights into analytical workloads by integrating directly with Microsoft Azure Monitor diagnostic logs. This new capability enables developers to analyze workload behavior over an extended time period and make informed decisions on query optimization or capacity management.

If you did not configure Diagnostics earlier:

1. Logon to Azure Portal (portal.azure.com) using your credentials

2. Navigate to your Azure Data Warehouse

3. Click on Diagnostic Settings from the side menu



trichterusgsdatawarehouse (trichterusgssqlserver/trichterusgsdatawarehouse) - Diagnostic settings

SQL data warehouse

Subscription: SQL-DS-DW_SUB6_R&D_60843

Resource group: trichterPreReadyLabs

Resource type: 0 selected

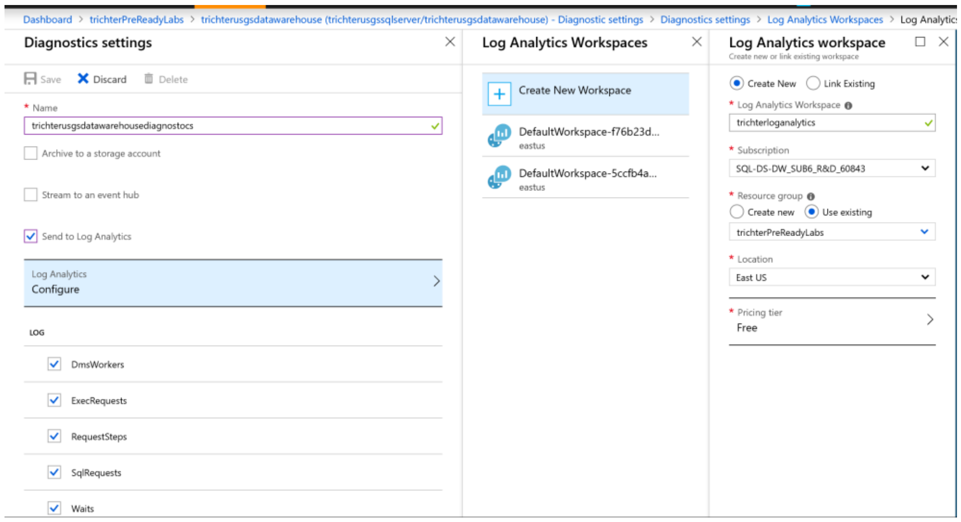
SQL-DS-DW_SUB6_R&D_60843 > trichterPreReadyLabs > trichterusgsdatawarehouse

Turn on diagnostics to collect the following data.

- SQLSecurityAuditEvents
- DmsWorkers
- ExecRequests
- RequestSteps
- SqlRequests
- Waits
- AllMetrics

4. Click on the Turn on diagnostics text link in the blade

5. Provide a name for your diagnostics
   - Check Send to Log Analytics
   - Select all options in LOG
   - Click Log Analytics Configure to activate the Log Analytics Workspaces blade
     - Create a new Workspace in your resource group using one of the charged pricing tiers



Dashboard > trichterPreReadyLabs > trichterusgsdatawarehouse (trichterusgssqlserver/trichterusgsdatawarehouse) - Diagnostic settings > Diagnostics settings > Log Analytics Workspaces > Log Analytic

Diagnostics settings

Save | Discard | Delete

* Name: trichterusgsdatawarehousediagnostocs

- [ ] Archive to a storage account
- [ ] Stream to an event hub
- [x] Send to Log Analytics

Log Analytics: Configure

LOG

- [x] DmsWorkers
- [x] ExecRequests
- [x] RequestSteps
- [x] SqlRequests
- [x] Waits

Log Analytics Workspaces

Create New Workspace

DefaultWorkspace-f76b23d... eastus

DefaultWorkspace-5ccfb4a... eastus

Log Analytics workspace

Create new or link existing workspace

[● Create New] ○ Link Existing

* Log Analytics Workspace: trichterloganalytics

* Subscription: SQL-DS-DW_SUB6_R&D_60843

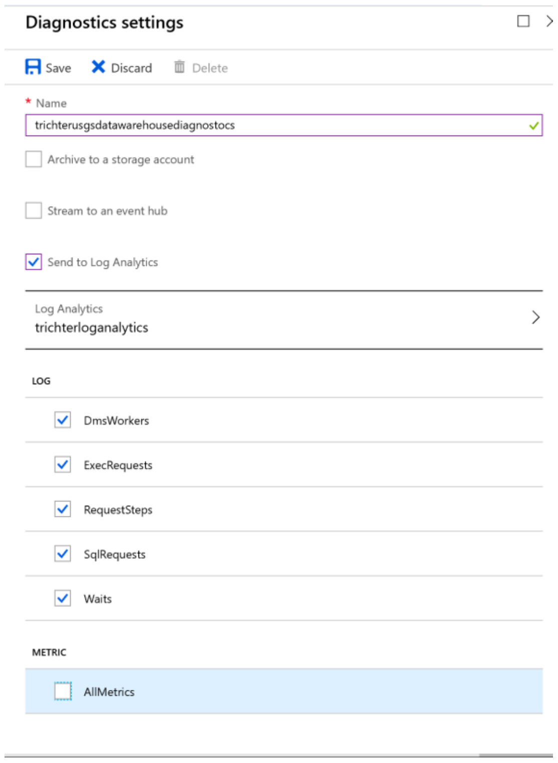
* Resource group: ○ Create new [● Use existing]

trichterPreReadyLabs

* Location: East US

* Pricing tier: Free

6. Click Save on Diagnostics settings blade



Diagnostics settings

Save | Discard | Delete

* Name: trichterusgsdatawarehousediagnostocs

- [ ] Archive to a storage account
- [ ] Stream to an event hub
- [x] Send to Log Analytics

Log Analytics: trichterloganalytics

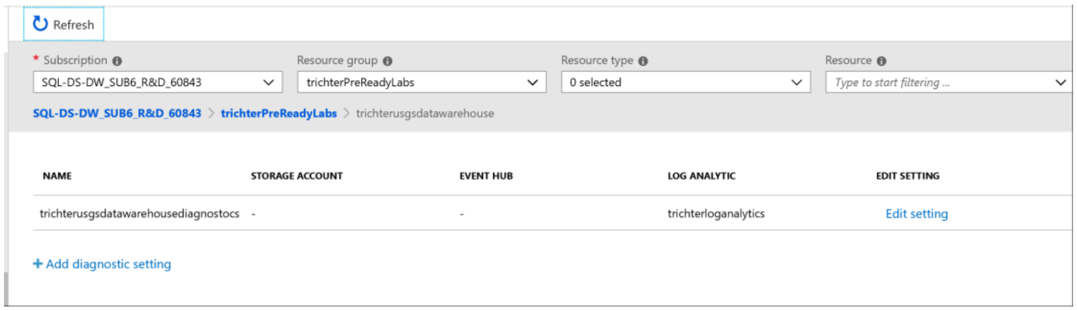
LOG

- [x] DmsWorkers
- [x] ExecRequests
- [x] RequestSteps
- [x] SqlRequests
- [x] Waits

METRIC

- [ ] AllMetrics

7. Click refresh on Diagnostic settings blade to see your new diagnostics



Subscription	Resource group	Resource type	Resource
SQL-DS-DW_SUB6_R&D_60843	trichterPreReadyLabs	0 selected	Type to start filtering ...

SQL-DS-DW_SUB6_R&D_60843 > trichterPreReadyLabs > trichterusgsdatawarehouse

NAME	STORAGE ACCOUNT	EVENT HUB	LOG ANALYTIC	EDIT SETTING
trichterusgsdatawarehousediagnostocs	-	-	trichterloganalytics	Edit setting

Review Diagnostics Logs:

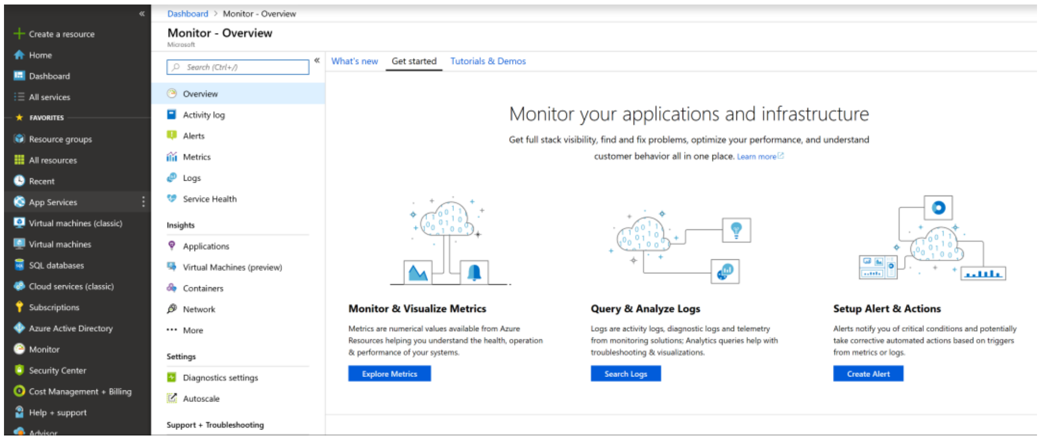
1. To collect some data, using SSMS connect to your Azure Data Warehouse and execute some queries

```
SELECT FIS.SalesAmount, DG.PostalCode, DC.YearlyIncome AS CustomerIncome,
DD.FullDateAlternateKey AS OrderDate
              FROM FactInternetSales AS FIS
                        JOIN DimCustomer AS DC
                        ON (FIS.CustomerKey = DC.CustomerKey)
                        JOIN DimDate AS DD
                        ON (FIS.OrderDateKey = DD.DateKey)
```

```
SELECT fis.SalesAmount, dst.Gender, dst.NumberCarsOwned, dst.YearlyIncome
AS CustomerYearlyIncome, dst.TotalChildren
      FROM FactInternetSales AS fis
      LEFT OUTER JOIN DimCustomer AS dst
      ON (fis.CustomerKey=dst.CustomerKey);
```

```
SELECT fis.SalesAmount, dst.ProductLine
      FROM FactInternetSales AS fis
      LEFT OUTER JOIN DimProduct AS dst
      ON (fis.ProductKey=dst.ProductKey);
```

2. In the portal select Monitor from the sidebar to activate the Monitor – Overview blade



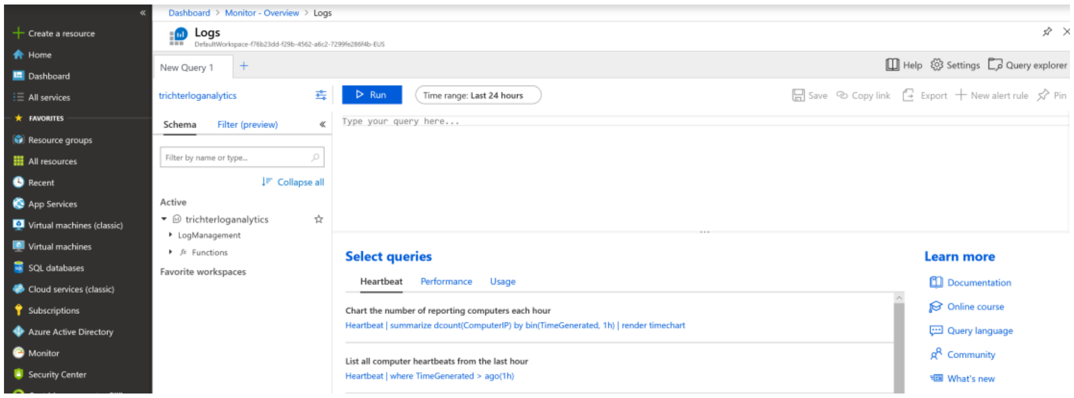
Get full stack visibility, find and fix problems, optimize your performance, and understand customer behavior all in one place. Learn more

Metrics are numerical values available from Azure Resources helping you understand the health, operation & performance of your systems.

Logs are activity logs, diagnostic logs and telemetry from monitoring solutions; Analytics queries help with troubleshooting & visualizations.

Alerts notify you of critical conditions and potentially take corrective automated actions based on triggers from metrics or logs.

3. Click Search Logs



DefaultWorkspace-f76b23dd-f29b-4562-a6c2-7299fe286f4b-EUS

Type your query here...

Chart the number of reporting computers each hour

```
Heartbeat | summarize dcount(ComputerIP) by bin(TimeGenerated, 1h) | render timechart
```

List all computer heartbeats from the last hour

```
Heartbeat | where TimeGenerated > ago(1h)
```

4. Execute these sample queries to get a feel for the information that can be retrieved from Azure Diagnostics.

Chart to determine the most active resource classes by request

```
    AzureDiagnostics
    | where Category contains "ExecRequests"
    | where Status_s == "Completed"
    | summarize totalQueries = dcount(RequestId_s) by
    ResourceClass_s
| render barchart
```

Count of all queued queries

```
    AzureDiagnostics
    | where Category contains "waits"
    | where Type_s == "UserConcurrencyResourceType"
    | summarize totalQueuedQueries =
    dcount(RequestId_s)
```

Chart for top requests most impacted by data movement operations

```
    AzureDiagnostics
    | where Category == "RequestSteps"
    | where OperationType_s in ("ShuffleMoveOperation",
    "BroadcastMoveOperation", "PartitionMoveOperation",
    "RoundRobinMoveOperation",
    "SingleSourceRoundRobinMoveOperation",
    "MoveOperation", "TrimMoveOperation")
    | where Status_s == "Complete"
    | project RequestId_s,
    duration=datetime_diff('millisecond',EndTime_t,
    StartTime_t)
    | order by duration desc
    | take 10
 | render barchart
```