9. Monitoring and optimizing data storage and processing

Good practice while building the data lake –

When a data lake is crated, there are multiple zones which can map to separate containers in the data lake. For example, Raw Zone would contain all the files in the original format that are coming in. Then we could have another container where basic filtering is carried out and lastly we would have curated zone would be the data where you want to perform analytics.

The hierarchy that is used is also important specially when the data is being streamed. For example, we could have a department for which the data is being streamed so the hierarchy could look like –
/department/raw/datasource/yyyy/mm/dd/file

Query Acceleration in Azure Data lake – When you have a .NET program that is accessing data from the data lake, from a csv or json file then we could use something called a query acceleratory to improve the performance of the query. To enable this feature, you have to register a provider using powershell. The name of the provider is Microsoft.Storage. More info on - https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-query-acceleration-how-to?tabs=azure-powershell%2Cpowershell

Monitoring service on Azure

You can open this section by searching Monitor at the top. From the left hand side, click on metrics. You can then select a resource and see the metrics for that particular resource.

- So, I click on my resource group, and select the dedicated sql pool and hit apply.
- Next we can select the metrics that we want to see. For example you can select the DWU used and see the usage of your dedicated sql pool over a period of time.
- Now we can also add alerts based on the DWU. So when we click on Crate alert rule, we see a new screen.
- On the screen we see a condition of Whenever the dwu used is greater then <logic undefined>
- Upon clicking upon the above condition, we would be able to set the type of DWU based on which we need to be alerted.
- After setting the threshold value, scroll down, to create an action group.
- Name the notification, select notification type like email, next we would have action that you want to take along with this alert, review and create.
- This action group can be used for other alert rules as well
- Finish creating the alert rule.

A thing to note about alerts is that once they are triggered, they are in the fired state. Meaning if the condition for the alert is generated again and alert is in the fired state, no action would be take.

In the **Activity log** from the left, you can see all the activities that happened for your subscription in the selected timespan.

Views in the Dedicated SQL pool

There are many different views that a user could see to monitor different things in Synapse. It could monitor connections, monitor query execution, monitor waiting queries, monitor memory, etc.

Link to see all the view names and their use cases – Monitor your dedicated SQL pool workload using DMVs - Azure Synapse Analytics | Microsoft Docs

Cache feature in your dedicated sql pool

When enabled, the sql pool automatically caches query results in the database. If the queries are executed often, the results can be taken from the cache instead. Doing this would reduce the compute power required to process queries.

Every 48 hours, if a resultset hasn't been used or the cache approaches maximum size, the data is removed.

- To see this feature go to you SSMS and execute the query –

```
SELECT name, is_result_set_caching_on
FROM sys.databases;
```

- You would observer that for the master database and for your dedicated sql pool it is False in the is_result_set_caching_on column
- Now turn on the query store –

```
ALTER DATABASE dedicatedsql
SET QUERY_STORE = ON;
```

- Next we need to turn caching on but it needs to be done with the context of the master database

```
ALTER DATABASE newpool
SET RESULT_SET_CACHING ON;
```

- Now execute a select statement in your dedicated sql pool. Execute the same statement again. We do this to get the data from the database the first time but use the cache store for getting the data the second time.
- Now go to the monitor section of Synapse studio and select SQL requests. Select your dedicated sql pool in pool.
- When you check the details of the two select statements you would see the first one has a partial move operation where as the second one has only the return operation meaning the data was fetched from the

You can also see if a sql request was a cache request or not by using the query with the query id –

```
SELECT request_id, command, result_cache_hit FROM sys.dm_pdw_exec_requests
WHERE request_id = 'QIDXXXX'
```

Workload management in dedicated sql pool – synapse has workgroups to which you can define different users. This is done to ensure the resources are always allocated properly. For example there could be a workgroup that loads that on the sql pool and there is a workgroup that would do analysis on it. To ensure that resources don't run out for either groups, we can distribute the resources between the two workgroups.

Azure Synapse SQL Pool – there are regular backups happening of your dedicated sql pool that are taken throughout the day. These restore points are then available for a duration of 7 days. You can restore your data warehouse in the primary region form any one of the snapshots taken in the past seven days.

When you go to the screen for your dedicated sql pool, you can see a restore option. You would be able to see the automatic restore points that have been made by the service. You can also define your own restore point (in case you are going to make a big change).

Monitoring in Data Factory – we've used this module before. We can monitor pipelines in ADF. The information about the pipeline and other information is only there for about 45 days. To save data for more than this duration we would have to store data in the log analytics workspace.

Go to all resources and create Log Analytics workspace – using your subscription, name it, select same region as your other resources and create.

- Now go to data factory and go to diagnostic setting from the left. Using this you can send information such as activity runs, pipeline runs, trigger runs, etc on to the Log Analytics Workspace.
- So we add a diagnostic setting, select the activityruns, pipelineruns, triggerruns and the Send to Log Analytics workspace option. Select your log analytics that you just crated. Name the setting and save.

- It would take about half an hour for the information to show up
- In the logs section of the log analytics workspace we can see the tables have showed up. We would need kusto query language to query the data.

Note – In the alerts and metrics section of the monitor section, you would be redirected to the Monitor service that we saw earlier. We can then create alerts and actions based on pipeline run conditions and so on.

Integration runtimes available in Data Factory

In the manage section of your Data factory Studio, you can see the integration runtimes that are there. By default we have the AzureIntegrationRuntime . We can also create our own integration runtimes – if we have any SQL Server Management packages and want to make use of those packages in Data Factory, then we can use Azure-SSIS integration runtime that is available when you create a new integration run time.

Apart from that, if you want to create a new Azure integration runtime, we can do that too. One of the reasons why we would need to do it is the region. By default, it is set to auto resolve, meaning if our data is in East US, and getting the data into a dedicated sql pool in the same location, so ADF would create the underlying compute infrastructure in the same location. Another reason is by manually setting the region, you can specify that the data is to never leave the specified region.

Failed Pipeline runs – let's say you have multiple pipelines that are running consecutively (one of more pipeline runs are part of a pipeline) and one of the internal pipelines fail, this would mean the outer pipeline would also fail as the outer pipeline depends on the running of the internal pipelines.

High Availability for Key Vault – when you create a key vault, the contents are replicated within the region and a secondary region that is defined by Azure paired regions. So if the primary region does go down, the requests to the key vault would be accepted at the secondary region.

Metrics that are available for Stream Analytics –

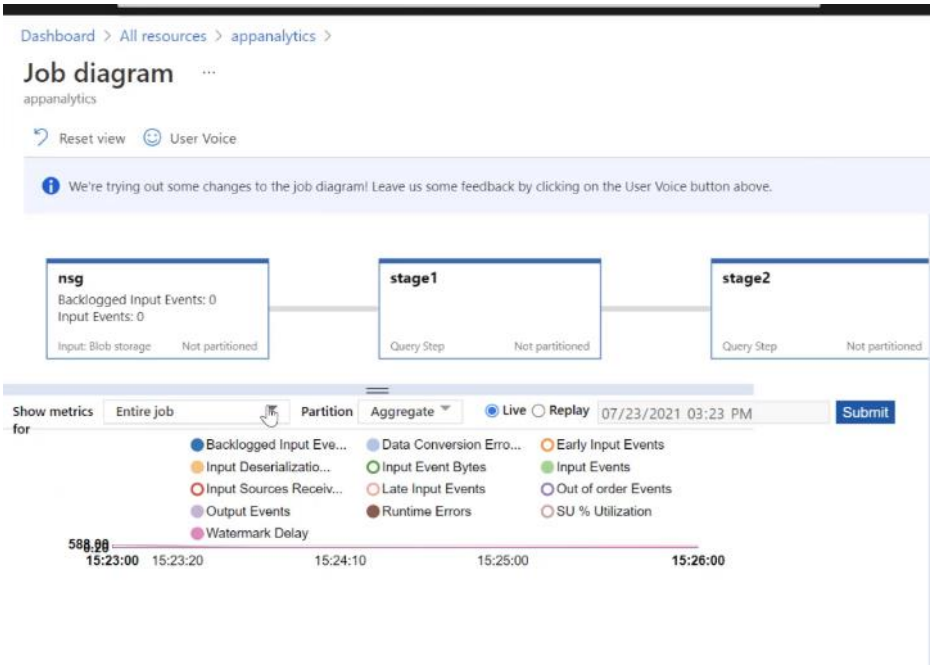
1. Backglogged input events – this is the number of input events that are backlogged. A non zero value for this metric implies that your job isn't able to keep up with the number of incoming events. For this you can decide to scale up the number of streaming units.
2. Data conversion errors – Number of output events that could not be converted to the expected output schema.
3. Early input events – events whose application timestamp is earlier than their arrival time by more than five minutes
4. Late input events – events that arrived later than the configured late arrival tolerance window
5. Out of order events – number of events received out of order that were either dropped or given an adjusted timestamp, based on the event ordering policy.
6. Watermark Delay

Streaming Units in Stream Analytics – when you create a stream analytics job, you assign a certain amount of streaming units. The streaming units determine the computing resources that are allocated to execute the job. They ensure low latency when it comes to stream processing, all of the jobs are performed in memory. If the streaming units reach 100% then the jobs start failing. Hence, it is important to monitor the streaming units that are being consumes for a job.

Monitoring a stream analytics job

When a streaming job is being run, from the left hand pane, you can scroll down in the left-hand menu to job diagram. It gives you a view of your entire flow in the stream analytics job.

After a job has been running for some time, we can see some metrics about the job.



When you hover over a particular stage, you can see the metrics at that point in time. You can see whether it is processing any events or not and the level of resources used and so on.

Importance of time in Azure Stream Analytics –

There are various factors to consider. If there are multiple different kinds of timestamps that an event would have, the time at which the event was generated, the time at which the event arrived at the event hub, etc. So based on this the stream analytics tries to understand the frequency of messages that are coming in. This is where the watermark delay comes into play.

Let's say the stream analytics receives an event at 10:00, the watermark number gets generated to understand the time aspect of the events that are coming in. Based on this, stream analytics can see whether the events are arriving late or are out of order event. And depending on if the events are arriving late, custom timestamps could be assigned to them or rejected if out of order.

Partitions in Event Hubs –

In event hubs you can define the number of partitions which would help with better throughput, ingesting more data at a time. So if there are multiple events that are coming into event hubs then the can be sent out to different partitions and different consumers can then receive these events from these partitions.

When sending events to event hubs, we can define which attribute can be used as the partition key. This would help separate the event into different partitions.

Then stream analytics can get the data from multiple partitions, work in parallel and send the data out to multiple partitions. The number of partitions that are being processed at each stage can also be seen in the job diagram in the Azure stream analytics.

Azure stream analytics can already make use of partitions due to something called compatibility level. But if you have a lower compatibility level, then you need to explicitly mention the partition key in the query.

Using stream analytics, you can also decide the number of partitions if you don't have control over the partition key. You can also specify the number of output partitions especially fi your destination is also azure event hubs. Keep in mind that the number of input partitions and the number of output partitions should be the same.

Identifying errors that might come up in Streaming –

When we see our stream analytics job has stopped due to some error. We can go to the Monitor from our dashboard and then to the activity log. We would see the error at the top and upon expanding it, we would be able to see what is the error that we are having.

Event hubs high availability – Azure event hubs can be paired with a secondary geo location so that if the primary geolocation becomes unavailable, the secondary one would take up its place. When a secondary region is created, the shared access policies would have all the events that were created on event hubs with their secondary connection strings.

It would be better to use the secondary connection strings because they are made to work in such a way that if the primary region goes down, the requests would automatically go to the secondary region without you having to make any changes.