



Getting the Most out of Synapse Notebooks

As data production continues to skyrocket, finding the right solution to store and analyze big data while keeping cost in mind is a top consideration for companies looking to leverage big data for actionable insights. Azure Synapse Analytics workspace provides a one stop shop for your organization's analytical needs. With the same functionality as Azure Databricks, Azure Synapse Analytics workspace allows you to leverage the power of creating enterprise analytics solutions within a seamless UI experience. Additionally, with native connections to Azure Data Lake, your organization can also store structured and unstructured data at extremely low costs.

The workspace comes with Synapse Notebooks, powered by Apache Spark. Apache Spark allows your workloads to scale and parallelly distribute jobs to analyze big data. Notebooks create fully integrated sessions where your developers can seamlessly collaborate and combine their code. But how can we get the most out of Synapse Notebooks? Here are four tips to maximize the benefits of this big data analysis tool:

Cohesive Collaboration



Synapse Notebooks allows for cohesive collaboration among team members with its user-friendly formatting. Notebooks consist of coding cells and markdown cells to better format your code for end user readability. By utilizing headers in markdown cells, Synapse creates a table of contents or outline for easier navigability. For additional review and collaboration, developers can easily leave comments on code cells, providing instant feedback and a way to track the progress of conversation threads.

Reduce Hassle of Language Switching



Reduce the time and efficiency lost from developers jumping from one coding platform to another by utilizing Synapse Notebook's multi-language and temp table capabilities. When creating a new coding cell, simply specify the magic command (`%%`) and the coding language you wish to use. At time of this writing, Synapse Notebooks currently supports Python (PySpark), Scala (Spark), Spark SQL, C# (.NET Spark), and R (Spark R). To reference the same data or variables across these languages, create SQL temp tables or data frames. Temp tables and data frames are persisted within a local session and do not have additional storage costs after the session ends.

Connect Notebooks to ETL Jobs



Utilize Synapse Notebook's connection capabilities by configuring notebook parameters and their direct storage connectors. To integrate a notebook parameter into a pipeline, define a parameter variable within a notebook cell, select the cell's ellipses, and choose to "Toggle Parameter Cell." Within your Synapse pipelines, use the notebook activity to specify the notebook name and input parameters. Notebooks also directly connect to your Azure Blob Storage or Azure Data Lake Gen2 using the integrated connection to quickly see query results, removing the boundary between ETL tool and storage account.

Scalable Workflows



Ensure your Notebooks can scale with ease with dynamic pool scaling. Set your pool to dynamically scale while maintaining cost by setting the max node, and executor size. Dynamic scaling ensures your jobs scale to meet workload demands. To view resource usage, within the Monitor pane, you can easily monitor the executor utilization, job status, and pool constraints. Additionally, the monitor pane connects directly to Spark's Web UI which provides a detailed view on job performance. Monitoring allows developers to make informed decisions on their ETL job performance and resource utilization. For example, if queuing exists, consider scaling up the total nodes of your pool.