Introduction

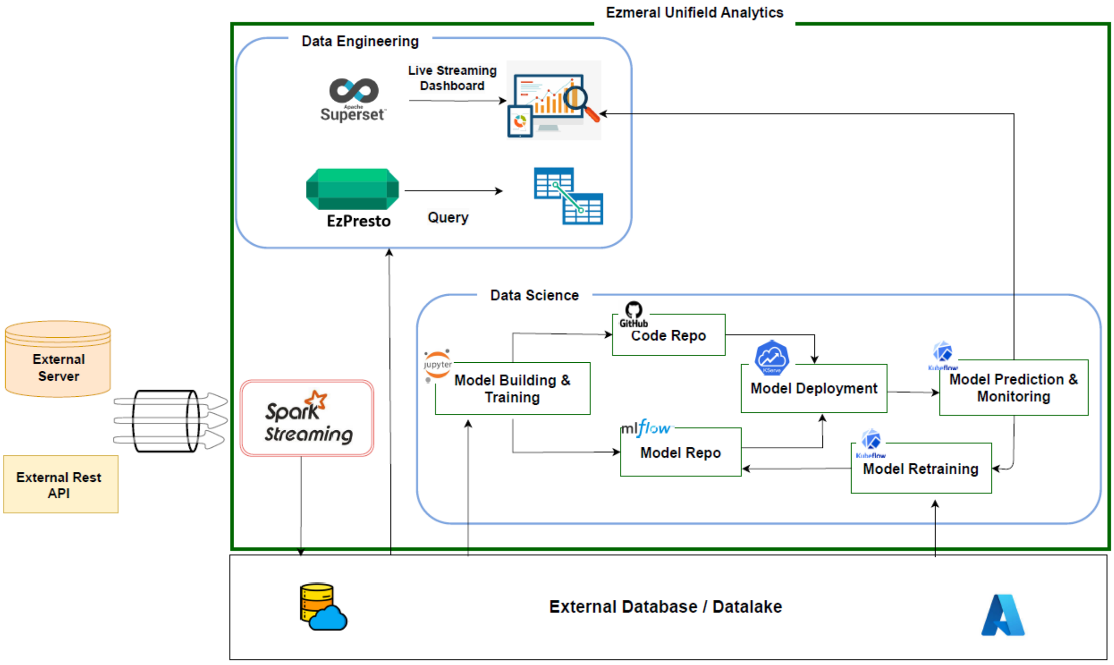
Welcome to the three-part blog series showcasing the remarkable capabilities of HPE Ezmeral Unified Analytics through a real-world use case: Stock Market Prediction. In Part 1 of this series, I will delve into the data engineering aspect of the platform, exploring how it facilitates seamless data management and analysis.

In Part 2 of the blog series, we will take you on a deep dive into the platform's ML/AI capabilities. Together, we will explore how the transformed data can be utilized for model building, leveraging Jupyter notebooks to perform interactive data exploration, pre-processing, and model training. Additionally, you will see how HPE Ezmeral Unified Analytics integrates seamlessly with MLflow for efficient model management and KServe for inference, allowing you to track and reproduce experiments easily.

Finally, in Part 3 of the series, I will focus on automation using MLOps. Now, let's embark on this exciting journey into the design and implementation of this cutting-edge solution.

What is HPE Ezmeral Unified Analytics?

HPE Ezmeral Unified Analytics software is a usage-based Software-as-a-Service (SaaS) platform that fully manages, supports, and maintains hybrid and multi-cloud modern analytics workloads through open-source tools. It goes beyond traditional analytics by seamlessly integrating machine learning and artificial intelligence capabilities, empowering users to develop and deploy data, analytics, and AI applications. By providing access to secure, enterprise-grade versions of popular open-source frameworks, the platform enables efficient and flexible scalability while securely accessing data stored in distributed data platforms. With its consistent SaaS experience, organizations can unlock data and insights faster, make data-driven predictions, and gain valuable business insights for faster decision-making, regardless of whether they operate on private, public, or on-premises infrastructure.

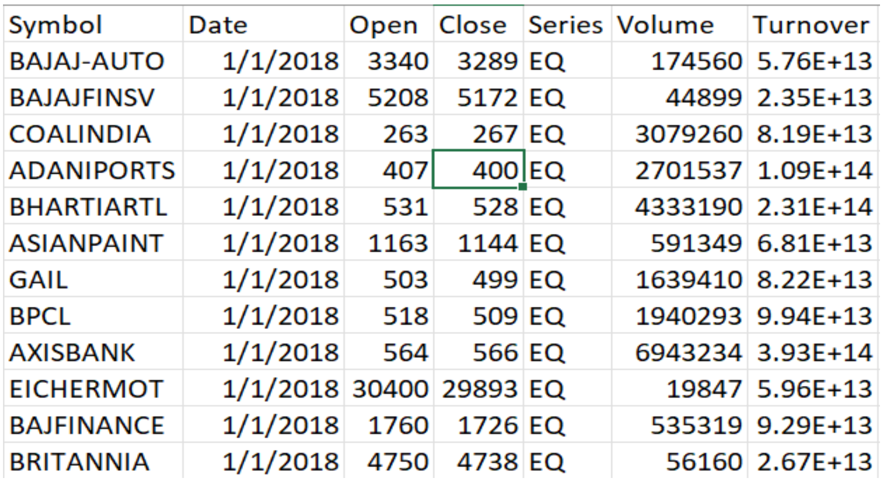


This use case involves leveraging external pricing server/rest API calls, which are streamed into the data lake/data warehouse of a cloud provider (Microsoft Azure) using Spark from HPE Ezmeral

Unified Analytics. Let me demonstrate how this platform enables data analysis using EzPresto (an enterprise-supported version of Presto) and empowers the creation of live dashboards using Superset. For a hands-on experience with the concepts discussed in this blog, explore the code examples on our GitHub repository (Link).

Step1: **Data Gathering**

The data consists of stock prices of different companies listed in National Stock Exchange (NSE) of India. The files consist of historical data from the year 2000 to 2021, which was transformed to a streaming data source. The data was pulled from external servers hosted publicly and then saved to HPE Ezmeral Data Fabric Volume.



Symbol	Date	Open	Close	Series	Volume	Turnover
BAJAJ-AUTO	1/1/2018	3340	3289	EQ	174560	5.76E+13
BAJAJFINSV	1/1/2018	5208	5172	EQ	44899	2.35E+13
COALINDIA	1/1/2018	263	267	EQ	3079260	8.19E+13
ADANIPORTS	1/1/2018	407	400	EQ	2701537	1.09E+14
BHARTIARTL	1/1/2018	531	528	EQ	4333190	2.31E+14
ASIANPAINT	1/1/2018	1163	1144	EQ	591349	6.81E+13
GAIL	1/1/2018	503	499	EQ	1639410	8.22E+13
BPCL	1/1/2018	518	509	EQ	1940293	9.94E+13
AXISBANK	1/1/2018	564	566	EQ	6943234	3.93E+14
EICHERMOT	1/1/2018	30400	29893	EQ	19847	5.96E+13
BAJFINANCE	1/1/2018	1760	1726	EQ	535319	9.29E+13
BRITANNIA	1/1/2018	4750	4738	EQ	56160	2.67E+13

Step 2: **Data Ingestion**

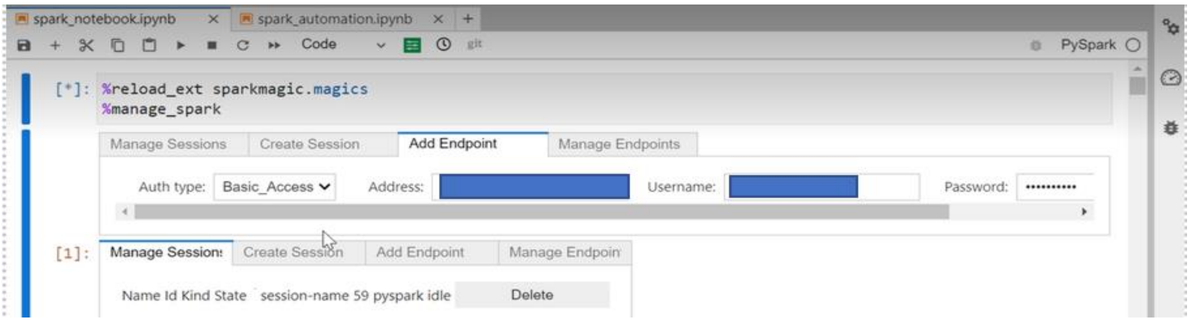
Apache Livy

HPE Ezmeral Unified Analytics gives access to Apache Livy, which enables easy interaction with the Spark cluster via REST interface. It simplifies the access between Spark cluster and application servers. It enables long running Spark contexts that can be used for multiple Spark jobs and multiple clients. Multiple Spark context can be managed that runs on the Spark Clusters. Spark applications can be either batch jobs or real-time streaming applications as per the business needs. Financial services have both long running batch applications as well as streaming applications, Apache Livy provides seamless management of Spark for the data engineers and application support team.

Apache Livy on the HPE Ezmeral platform enables programmatic, fault-tolerant, multi-tenant submission of Spark jobs from web/mobile apps (no Spark client needed). So, multiple users can interact with the Spark cluster concurrently and reliably. Livy speaks either Scala or Python, so clients can communicate with the Spark cluster via either language remotely. Also, batch job submissions can be done in Scala, Java, or Python.

It enables easy interaction with a Spark cluster over a REST interface. It enables easy submission of Spark jobs or snippets of Spark code, synchronous or asynchronous result retrieval, as well as Spark context management, all via a simple REST interface or an RPC client library.

HPE Ezmeral Unified Analytics provides functions like %reload_ext sparkmagics and %manage_spark for seamless connection to the Spark cluster. %reload_ext sparkmagics loads the Spark session and authenticates the user for secured access to the Spark session. %manage_spark will create the Spark session with predefined Spark cluster configuration in the background.



```
[*]: %reload_ext sparkmagic.magics
     %manage_spark
```

Manage Sessions | Create Session | Add Endpoint | Manage Endpoints

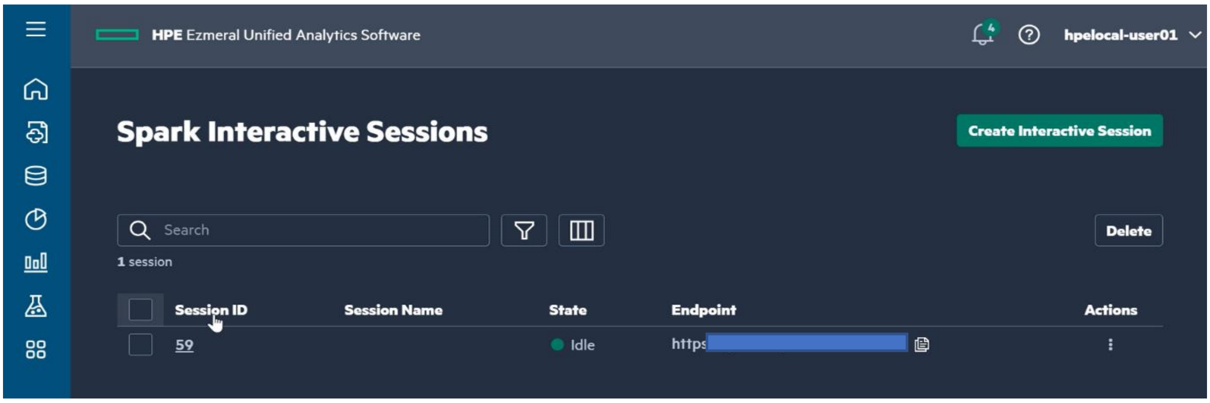
Auth type: Basic_Access   Address: [redacted]   Username: [redacted]   Password: ••••••••••

```
[1]:
```

Manage Sessions | Create Session | Add Endpoint | Manage Endpoints

Name Id Kind State   session-name 59 pyspark idle   Delete

Once the Livy session is enabled, the code can be run on the notebook servers.


HPE Ezmeral Unified Analytics Software
hpelocal-user01
Create Interactive Session
Search
Delete
1 session

Spark Interactive Sessions

	Session ID	Session Name	State	Endpoint	Actions
□	59		● Idle	https[redacted]	⋮

Spark Streaming

Financial applications like real-time transaction processing, fraud detection, trade matching and settlement systems are widely distributed and deal with large volume and variety of data. These systems require parallel processing of transactions in a distributed computing architecture. Hence, Spark streaming best suits the needs of such financial applications like the stock market prediction analysis.

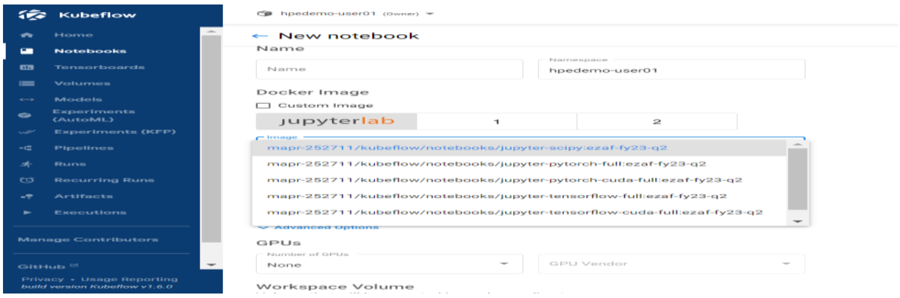
Spark Streaming is a real-time data processing module in Apache Spark, a popular distributed computing framework for big data processing. It enables processing and analysis of live data streams in a scalable and fault-tolerant manner. Spark Streaming brings the power and flexibility of Spark's batch processing capabilities to real-time data streams, making it a versatile choice for various real-time data processing use cases.

Micro-Batch Processing: Spark Streaming follows the micro-batch processing model, where it divides the continuous stream of data into small, discrete batches. Each batch of data is processed as a RDD (Resilient Distributed Dataset), which is Spark's fundamental data abstraction for distributed computing. This approach allows Spark Streaming to process data in mini-batches, providing low-latency processing and better resource utilization.

Data Sources and Sinks: Spark Streaming can ingest data from various sources, including Kafka, Flume, Kinesis, HDFS, TCP sockets, and more. It supports a wide range of input formats, making it compatible with different streaming data pipelines. Similarly, Spark Streaming can write the processed data to various sinks, such as HDFS, databases (e.g., MySQL, Cassandra), and external systems.

Notebook servers

HPE Ezmeral Unified Analytics is equipped with notebook servers that can execute Python commands seamlessly along with scalable resources like CPUs, GPUs, and memory. Notebook servers can be spun up on Kubeflow using pre-defined Jupyter notebook images or custom-built notebook images based on your requirement. It will take few minutes to bring the notebook server up and running.



New notebook

Name: Name

Namespace: hpedemo-user01

Docker Image

□ Custom Image

Image:
mapr-252711/kubeflow/notebooks/jupyter-scipy:ezaf-fy23-q2
mapr-252711/kubeflow/notebooks/jupyter-pytorch-full:ezaf-fy23-q2
mapr-252711/kubeflow/notebooks/jupyter-pytorch-cuda-full:ezaf-fy23-q2
mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezaf-fy23-q2
mapr-252711/kubeflow/notebooks/jupyter-tensorflow-cuda-full:ezaf-fy23-q2

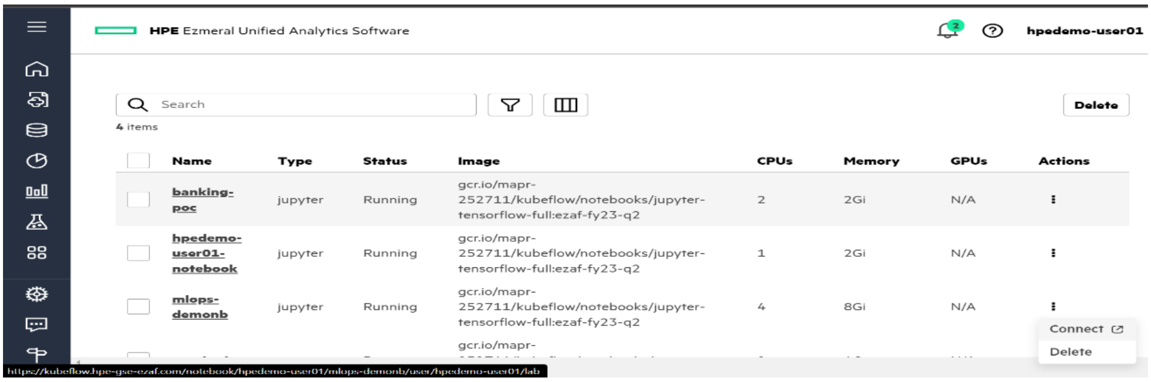
GPUs

Number of GPUs: None

GPU Vendor

Workspace Volume

Once it is available, you can connect to the notebook server either on HPE Ezmeral Unified Analytics Notebooks Tab or directly from the Kubeflow Notebooks.



4 items

	Name	Type	Status	Image	CPUs	Memory	GPUs	Actions
□	banking-poc	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezaf-fy23-q2	2	2Gi	N/A	⋮
□	hpedemo-user01-notebook	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezaf-fy23-q2	1	2Gi	N/A	⋮
□	mlops-demonb	jupyter	Running	gcr.io/mapr-252711/kubeflow/notebooks/jupyter-tensorflow-full:ezaf-fy23-q2	4	8Gi	N/A	⋮
□				gcr.io/mapr-				

https://kubeflow.hpe-gse-ezaf.com/notebook/hpedemo-user01/mlops-demonb/user/hpedemo-user01/lab

MySQL Database

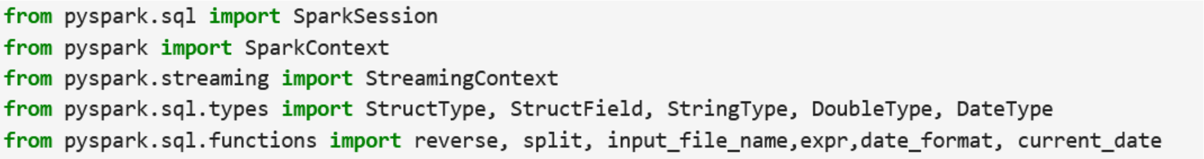
A MySQL database was created and hosted in Microsoft Azure to capture the structured streaming data to a single table. The database server is configured to permit access to the select IP addresses.

Step 3: **Streaming data to database**

The data is read from HPE Ezmeral Data Fabric volume by the Spark Streaming engine in constant time intervals. The Spark engine converts the files into batches and does some data engineering like transformations and aggregations on the data. Finally, it is saved to MySQL database using jdbc connections. It is mandatory for all the incoming files to share the same schema.

3.1 Load the required Spark libraries.

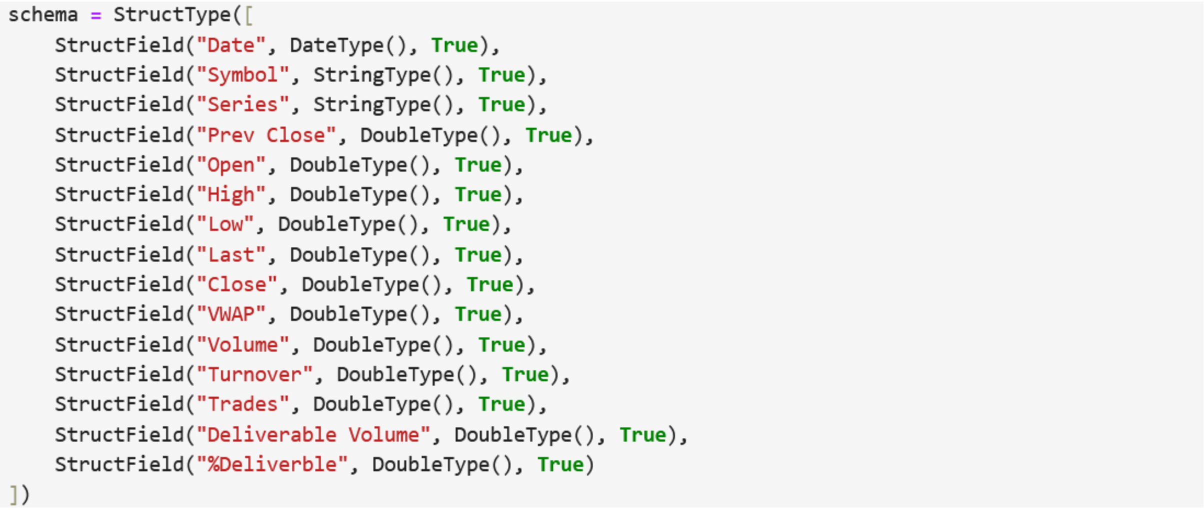
Once connected to the Livy server, the Spark connection is configured and managed internally by HPE Ezmeral Unified Analytics platform. Now you can directly import the required libraries and you'll be ready to use Spark.



```
from pyspark.sql import SparkSession
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark.sql.types import StructType, StructField, StringType, DoubleType, DateType
from pyspark.sql.functions import reverse, split, input_file_name,expr,date_format, current_date
```

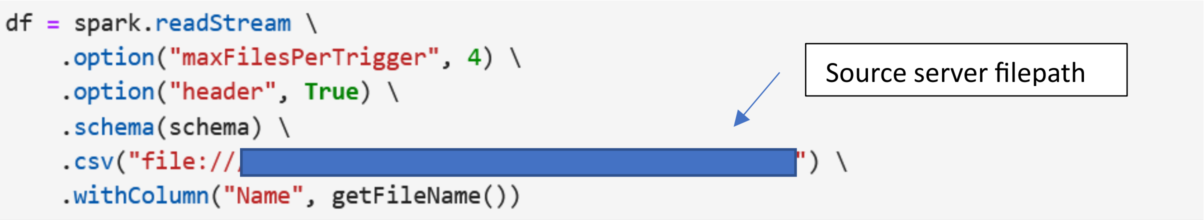
3.2 Define the Data Schema

Define the data schema for the data to stream in the application.



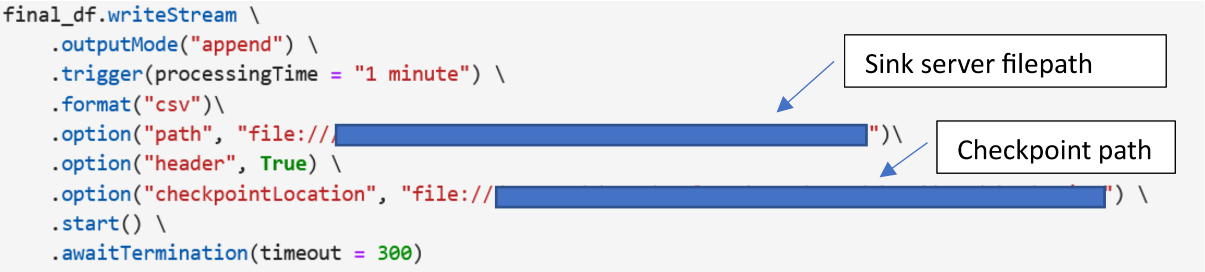
```
schema = StructType([
    StructField("Date", DateType(), True),
    StructField("Symbol", StringType(), True),
    StructField("Series", StringType(), True),
    StructField("Prev Close", DoubleType(), True),
    StructField("Open", DoubleType(), True),
    StructField("High", DoubleType(), True),
    StructField("Low", DoubleType(), True),
    StructField("Last", DoubleType(), True),
    StructField("Close", DoubleType(), True),
    StructField("VWAP", DoubleType(), True),
    StructField("Volume", DoubleType(), True),
    StructField("Turnover", DoubleType(), True),
    StructField("Trades", DoubleType(), True),
    StructField("Deliverable Volume", DoubleType(), True),
    StructField("%Deliverble", DoubleType(), True)
])
```

3.3 Read the input stream of files from the external server



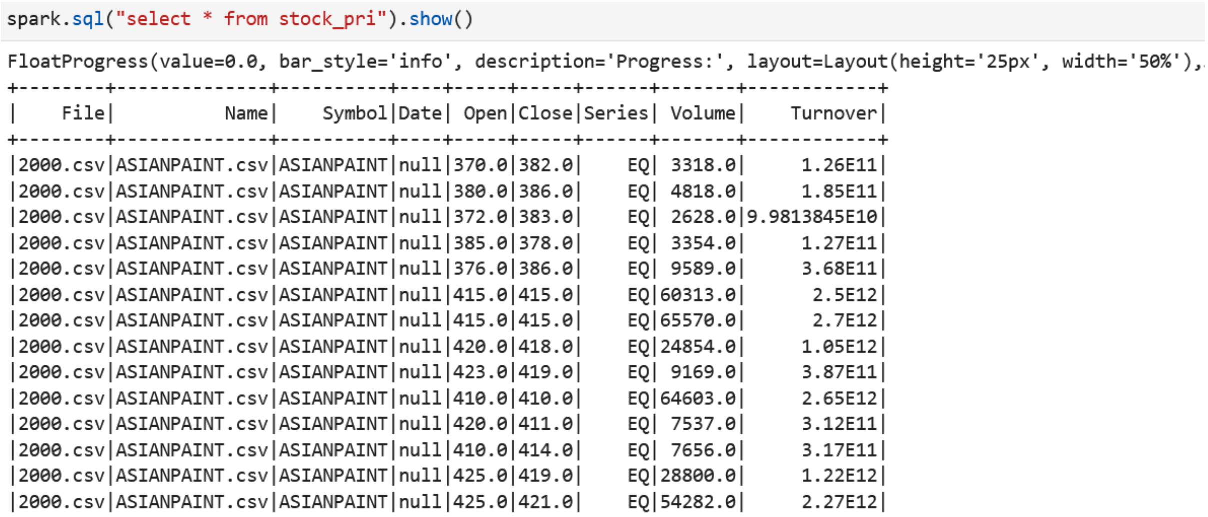
```
df = spark.readStream \
    .option("maxFilesPerTrigger", 4) \
    .option("header", True) \
    .schema(schema) \
    .csv("file://[redacted]") \
    .withColumn("Name", getFileName())
```

3.4 Write the output stream to the destination path.



```
final_df.writeStream \
    .outputMode("append") \
    .trigger(processingTime = "1 minute") \
    .format("csv")\
    .option("path", "file://[redacted]")\
    .option("header", True) \
    .option("checkpointLocation", "file://[redacted]") \
    .start() \
    .awaitTermination(timeout = 300)
```

3.5 Read the data using Spark SQL and perform Exploratory Data Analysis .

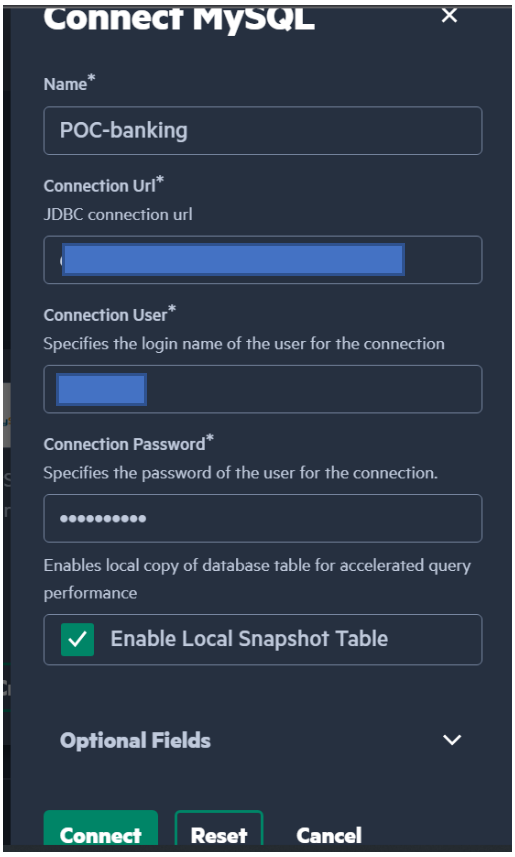


```
spark.sql("select * from stock_pri").show()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),
+--------+-------------+---------+----+-----+-----+------+-------+------------+
|    File|         Name|   Symbol|Date| Open|Close|Series| Volume|    Turnover|
+--------+-------------+---------+----+-----+-----+------+-------+------------+
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|370.0|382.0|    EQ| 3318.0|     1.26E11|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|380.0|386.0|    EQ| 4818.0|     1.85E11|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|372.0|383.0|    EQ| 2628.0|9.9813845E10|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|385.0|378.0|    EQ| 3354.0|     1.27E11|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|376.0|386.0|    EQ| 9589.0|     3.68E11|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|415.0|415.0|    EQ|60313.0|      2.5E12|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|415.0|415.0|    EQ|65570.0|      2.7E12|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|420.0|418.0|    EQ|24854.0|     1.05E12|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|423.0|419.0|    EQ| 9169.0|     3.87E11|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|410.0|410.0|    EQ|64603.0|     2.65E12|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|420.0|411.0|    EQ| 7537.0|     3.12E11|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|410.0|414.0|    EQ| 7656.0|     3.17E11|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|425.0|419.0|    EQ|28800.0|     1.22E12|
|2000.csv|ASIANPAINT.csv|ASIANPAINT|null|425.0|421.0|    EQ|54282.0|     2.27E12|
```

Step 4: **Connecting the database to HPE Ezmeral Unified Analytics**

HPE Ezmeral Unified Analytics provides users a quick and simple process to connect to external data sources like different databases, Hive, Snowflake, Teradata, etc. Here a new data source connection is added, and the source is selected as MySQL. The connection is established once the jdbc connection url, username and password are validated.



Connect MySQL

Name*

POC-banking

Connection Url*

JDBC connection url

[redacted]

Connection User*

Specifies the login name of the user for the connection

[redacted]

Connection Password*

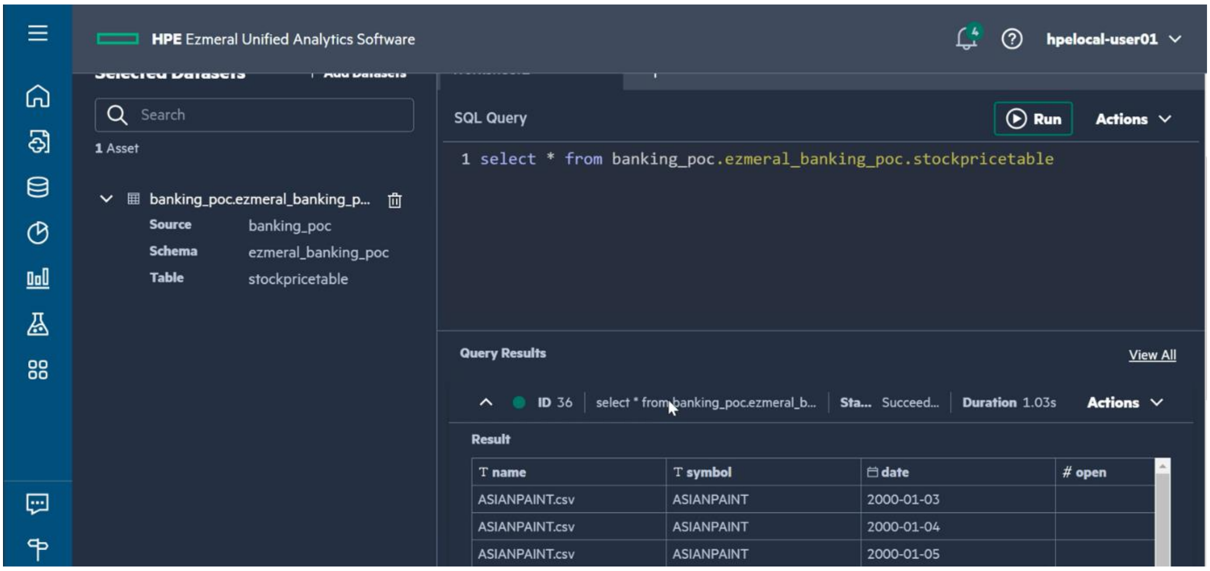
Specifies the password of the user for the connection.

••••••••••

Enables local copy of database table for accelerated query performance

☑ Enable Local Snapshot Table

This will connect the database to EzPresto, which is a distributed analytic query engine for big data, integrated into HPE Ezmeral Unified Analytics. This enables users to query the tables in the database using SQL commands. This service helps users to use the database seamlessly by enabling them to insert, delete, update and query records from the tables. The data can be accessed from a remote server or on HPE Ezmeral Data Fabric Software.



Source	banking_poc
Schema	ezmeral_banking_poc
Table	stockpricetable

```
1 select * from banking_poc.ezmeral_banking_poc.stockpricetable
```

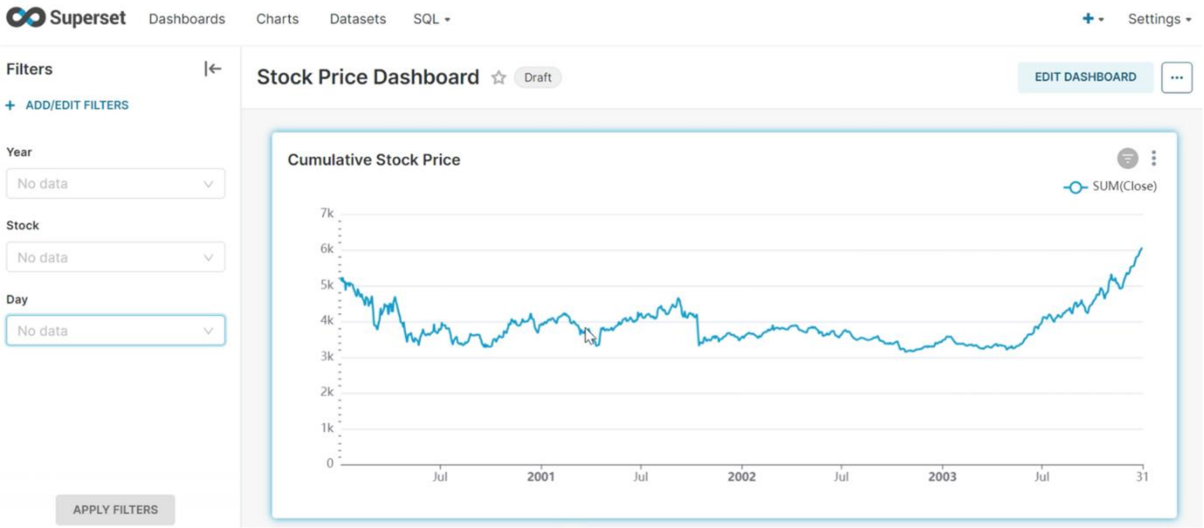
ID 36 | select * from banking_poc.ezmeral_b... | Sta... Succeed... | Duration 1.03s

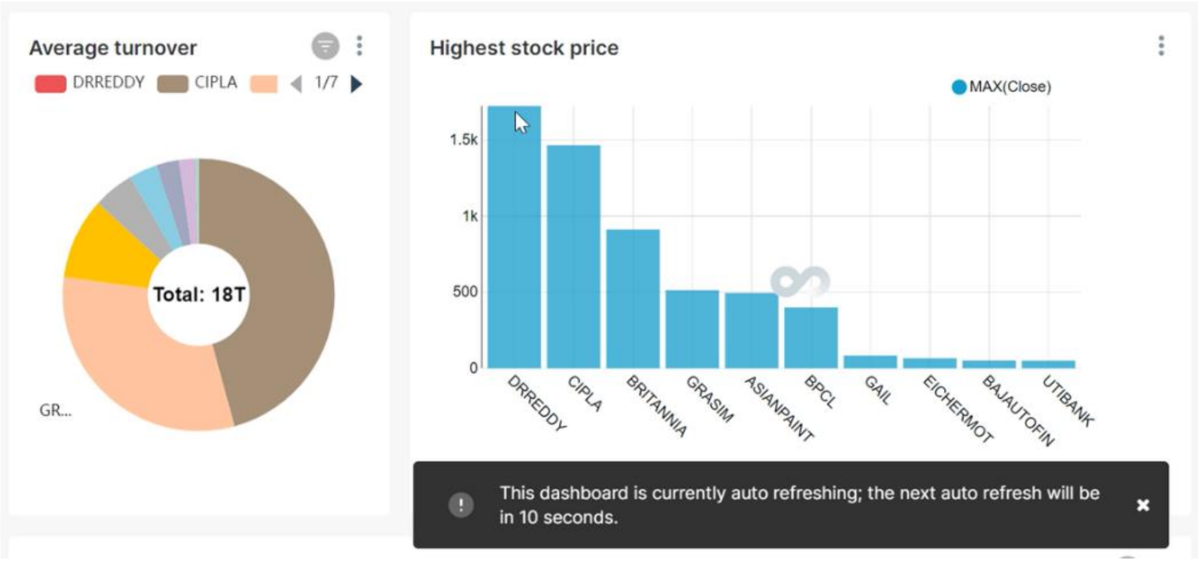
T name	T symbol	date	# open
ASIANPAINT.csv	ASIANPAINT	2000-01-03	
ASIANPAINT.csv	ASIANPAINT	2000-01-04	
ASIANPAINT.csv	ASIANPAINT	2000-01-05	

Step 5: **Visualization using Superset.**

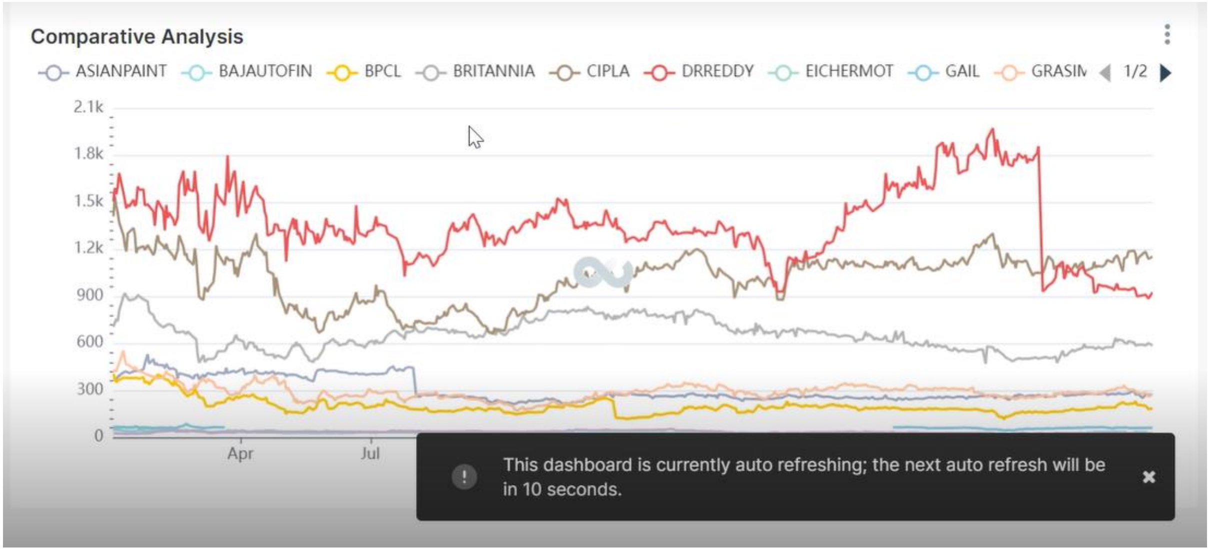
Apache Superset, a data visualization tool has been integrated into HPE Ezmeral Unified Analytics, which helps with the graphical representation of data, from simple line charts to highly detailed geospatial charts. The dashboards help users to get a clear picture of the KPIs and other relevant metrices of the business.

Here, a new dashboard is created in HPE Ezmeral Unified Analytics, and the connection to the database is established. Different visuals on the stock data are integrated into the dashboard and it is customized to auto refresh to a customer-defined time interval. Once the data starts streaming, the dashboard updates the visuals periodically and the latest data is available on the dashboard for analysis.





This dashboard is currently auto refreshing; the next auto refresh will be in 10 seconds.



This dashboard is currently auto refreshing; the next auto refresh will be in 10 seconds.

In concluding Part 1 of this blog series, you've journeyed through the data engineering and analytics aspects of using Spark, EzPresto, and Superset, powered by HPE Ezmeral Unified Analytics. With a spotlight on assimilating external pricing data to craft a dynamic dashboard, I hope I have illuminated how this platform brings together best of breed open-source tools to transform complex data into valuable insights.

Don't miss Part 2, where you'll get to explore the machine learning capabilities of our platform. To get familiar with HPE Unified Analytics Software, try it for free or visit our website for details. Let's unlock the future of analytics together!