BIG DATA

PROJECT REPORT

REAL TIME TRACKING OF

INTERNATIONAL SPACE STATION

USING PYSPARK + KAFKA

By:

Ayush Ranjan (21CSU020)

Vanshak Vats(21CSU110)

Anurag Sharma (21CSU496)

Batch: DS-VI-A (DS-1)

To: Dr. Meghna Sharma

Department of Computer Science and Engineering

The NorthCap University

Gurugram- 122017, India

Acknowledgment

I would like to express my sincere gratitude to all those who have contributed to the completion of this project.

First and foremost, I extend my deepest appreciation to Dr. Meghna Sharma Ma'am for their invaluable guidance, support, and encouragement throughout the development of this real-time tracking system for the International Space Station. Their expertise and insights have been instrumental in shaping the direction of this project.

I would also like to thank my family and friends for their unwavering encouragement and understanding during this endeavor. Their belief in my abilities has been a constant source of motivation.

Furthermore, I am grateful to The Northcap University for providing the necessary resources and facilities that enabled me to carry out this research work.

Last but not least, I extend my heartfelt thanks to all the researchers and practitioners in the field of data streaming, PySpark, and Kafka whose work has laid the foundation for this project. Their contributions have been invaluable in helping me navigate the complexities of real-time data processing and analysis.

Thank you all for your support and encouragement throughout this journey.

Table of Contents

1. Problem Statement

Problem Statement: Real-Time Tracking of International Space Station

In an era of advancing space exploration and increasing reliance on satellites for communication and observation, there's a growing need for real-time monitoring and tracking of space assets such as the International Space Station (ISS). However, traditional tracking systems often face challenges in providing up-to-date information due to limitations in data processing and transmission.

The objective of this project is to develop a real-time tracking system for the International Space Station (ISS) using PySpark and Kafka. By harnessing the power of distributed stream processing and messaging systems, the system aims to continuously collect and process telemetry data from various sources to provide accurate and timely information about the ISS's current position and trajectory.

This project seeks to overcome the limitations of existing tracking systems by leveraging PySpark for parallel processing of large volumes of streaming data and Kafka for seamless communication and integration between different components of the system. Additionally, the system will utilize machine learning algorithms to predict the ISS's future trajectory based on historical data and external factors such as atmospheric conditions and orbital dynamics.

2. Objectives

The primary objective of this project is to develop a real-time tracking system for the International Space Station (ISS) using PySpark and Kafka. The specific objectives are outlined as follows:

1. Data Acquisition and Processing
2. Integration with Kafka
3. Real-Time Visualization
4. Scalability and Performance Optimization
5. Evaluation and Validation
6. Documentation and Reporting
7. By achieving these objectives, this project aims to develop a robust and efficient real-time tracking system for the International Space Station, enhancing our ability to monitor and understand its movements in Earth's orbit.

3. Project Description

In an era of advancing space exploration and satellite technology, real-time monitoring and tracking of the International Space Station (ISS) have become paramount. Traditional tracking systems often face challenges in providing up-to-date information due to limitations in data processing and transmission. To address these challenges, we propose the development of a real-time tracking system for the ISS, utilizing PySpark and Kafka technologies.

Key Features:

1. Data Acquisition and Processing: The system will collect real-time telemetry data from various sources and process it efficiently using PySpark. This will involve extracting relevant information about the ISS's current position, and other pertinent parameters.
2. Integration with Kafka: Kafka will be employed for seamless ingestion and distribution of real-time data streams. This integration ensures reliability and fault tolerance in data processing, enabling continuous monitoring of the ISS.
3. Real-Time Visualization: Real-time visualization tools will be implemented to display the current position and trajectory of the ISS on a map. This provides users with an intuitive interface to track the ISS's movement and location.
4. Scalability and Performance Optimization: The system architecture and algorithms will be optimized to ensure scalability and performance. This enables the system to handle large volumes of data efficiently and provide timely updates to users.
5. Evaluation and Validation: The accuracy and reliability of the tracking system will be evaluated by comparing its predictions with ground-truth data and conducting validation tests in real-world scenarios.

Outcome:

The successful implementation of our real-time tracking system for the ISS will provide accurate and timely information about its movement and location. This will enhance our ability to monitor and understand the ISS's activities in Earth's orbit, contributing to advancements in space exploration and satellite technology.

4. ANALYSIS

SOFTWARE REQUIREMENTS

Data Processing and Analysis Software:

• Python for preprocessing, coding, and analysis.

• PySpark for distributed stream processing and data analysis.

• Kafka for real-time data ingestion and distribution.

• Apache Spark MLlib for machine learning model development**.**

Development Environment:

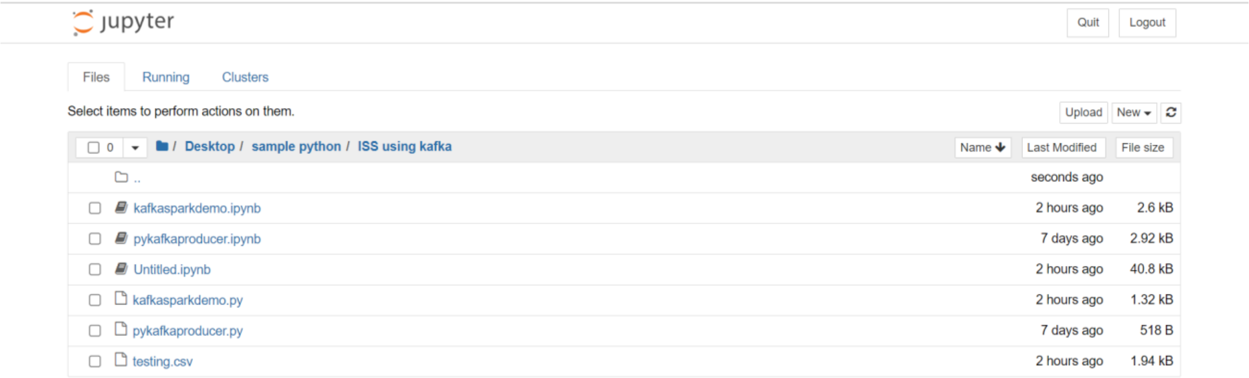
• Integrated Development Environments (IDEs) for coding, debugging, and testing, such as Jupyter Notebook and PyCharm.

• Apache Kafka for setting up the Kafka messaging system.

• Apache Spark for running PySpark applications.

5. Outputs:

Project Main File:



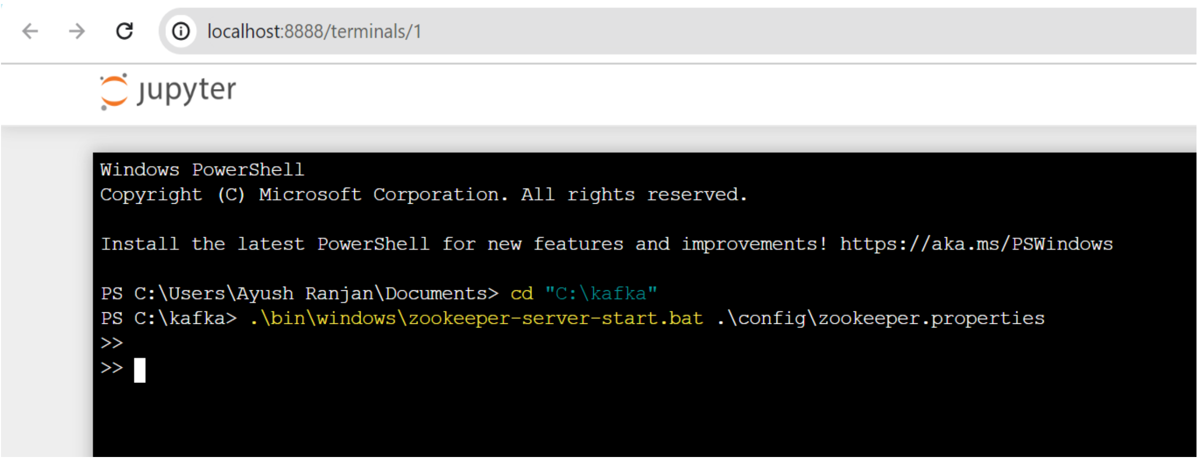
Select items to perform actions on them.

/ Desktop / sample python / ISS using kafka

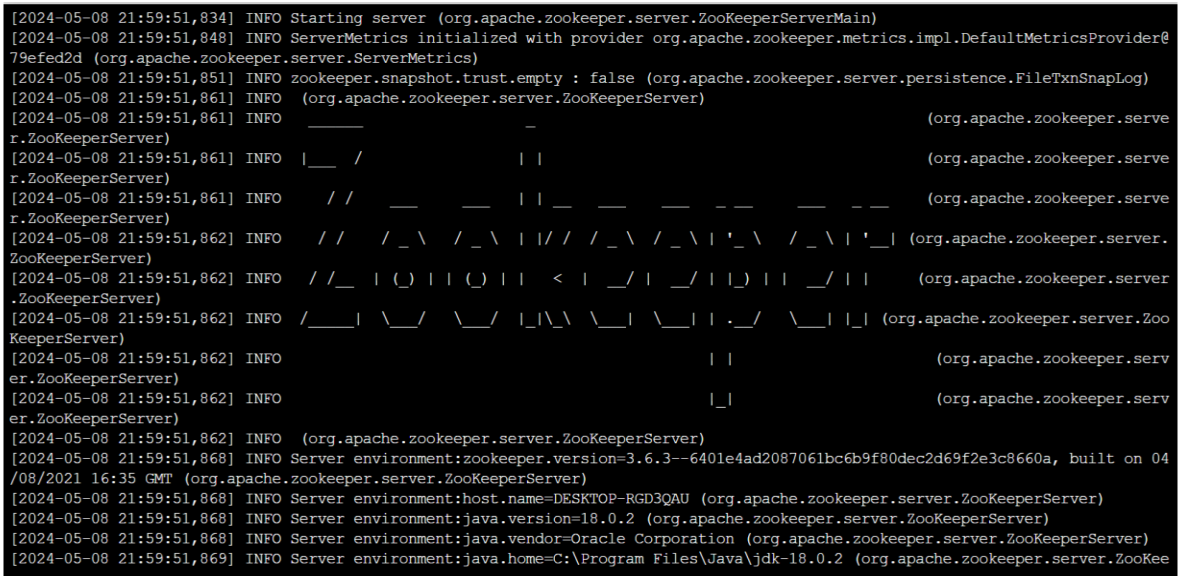
Name	Last Modified	File size
..	seconds ago	
kafkasparkdemo.ipynb	2 hours ago	2.6 kB
pykafkaproducer.ipynb	7 days ago	2.92 kB
Untitled.ipynb	2 hours ago	40.8 kB
kafkasparkdemo.py	2 hours ago	1.32 kB
pykafkaproducer.py	7 days ago	518 B
testing.csv	2 hours ago	1.94 kB

Terminals:

1. Kafka zookeeper:

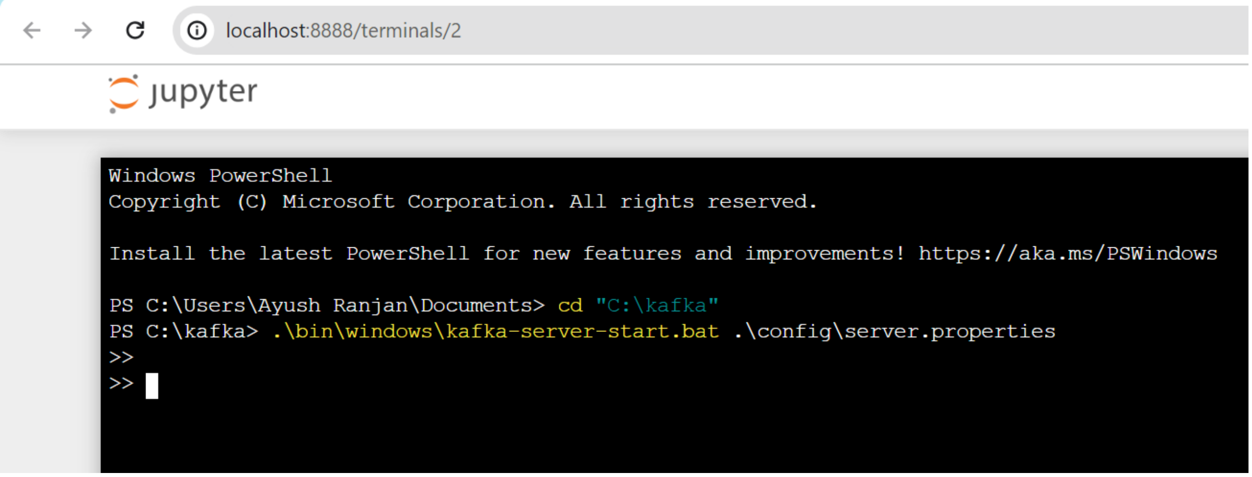


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Ayush Ranjan\Documents> cd "C:\kafka"
PS C:\kafka> .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
>>
>>
```

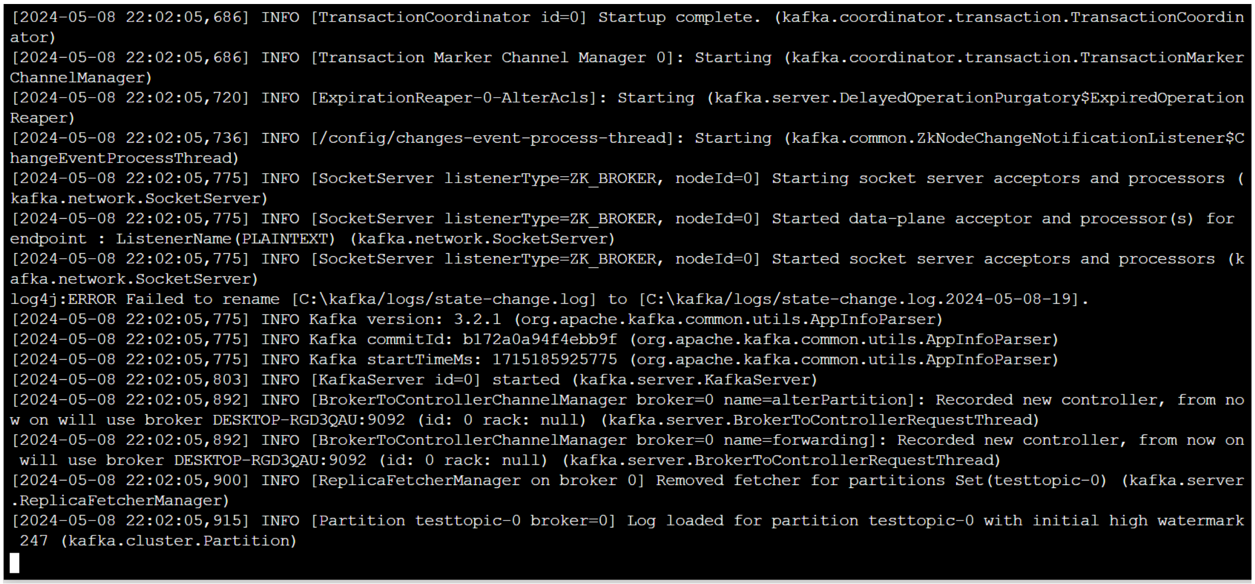


```
[2024-05-08 21:59:51,834] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2024-05-08 21:59:51,848] INFO ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsProvider@
79efed2d (org.apache.zookeeper.server.ServerMetrics)
[2024-05-08 21:59:51,851] INFO zookeeper.snapshot.trust.empty : false (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2024-05-08 21:59:51,861] INFO  (org.apache.zookeeper.server.ZooKeeperServer)
[2024-05-08 21:59:51,861] INFO   ______                  _                                          (org.apache.zookeeper.serve
r.ZooKeeperServer)
[2024-05-08 21:59:51,861] INFO  |___  /                 | |                                         (org.apache.zookeeper.serve
r.ZooKeeperServer)
[2024-05-08 21:59:51,861] INFO     / /    ___     ___   | | __   ___    ___   _ __     ___   _ __     (org.apache.zookeeper.serve
r.ZooKeeperServer)
[2024-05-08 21:59:51,862] INFO    / /    / _ \   / _ \  | |/ /  / _ \  / _ \ | '_ \   / _ \ | '__| (org.apache.zookeeper.server.
ZooKeeperServer)
[2024-05-08 21:59:51,862] INFO   / /__  | (_) | | (_) | |   <  |  __/ |  __/ | |_) | |  __/ | |     (org.apache.zookeeper.server
.ZooKeeperServer)
[2024-05-08 21:59:51,862] INFO  /_____|  \___/   \___/  |_|\_\  \___|  \___| | .__/   \___| |_| (org.apache.zookeeper.server.Zoo
KeeperServer)
[2024-05-08 21:59:51,862] INFO                                               | |                     (org.apache.zookeeper.serv
er.ZooKeeperServer)
[2024-05-08 21:59:51,862] INFO                                               |_|                     (org.apache.zookeeper.serv
er.ZooKeeperServer)
[2024-05-08 21:59:51,862] INFO  (org.apache.zookeeper.server.ZooKeeperServer)
[2024-05-08 21:59:51,868] INFO Server environment:zookeeper.version=3.6.3--6401e4ad2087061bc6b9f80dec2d69f2e3c8660a, built on 04
/08/2021 16:35 GMT (org.apache.zookeeper.server.ZooKeeperServer)
[2024-05-08 21:59:51,868] INFO Server environment:host.name=DESKTOP-RGD3QAU (org.apache.zookeeper.server.ZooKeeperServer)
[2024-05-08 21:59:51,868] INFO Server environment:java.version=18.0.2 (org.apache.zookeeper.server.ZooKeeperServer)
[2024-05-08 21:59:51,868] INFO Server environment:java.vendor=Oracle Corporation (org.apache.zookeeper.server.ZooKeeperServer)
[2024-05-08 21:59:51,869] INFO Server environment:java.home=C:\Program Files\Java\jdk-18.0.2 (org.apache.zookeeper.server.ZooKee
```

2. Kafka server:

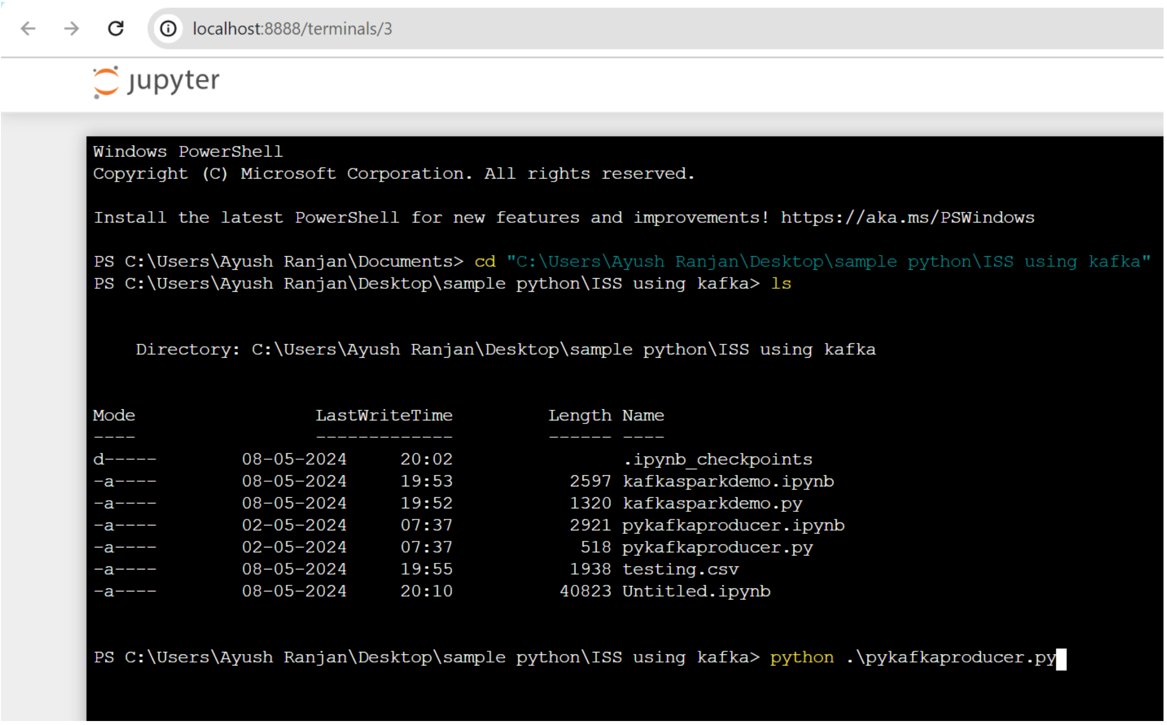


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Ayush Ranjan\Documents> cd "C:\kafka"
PS C:\kafka> .\bin\windows\kafka-server-start.bat .\config\server.properties
>>
>>
```

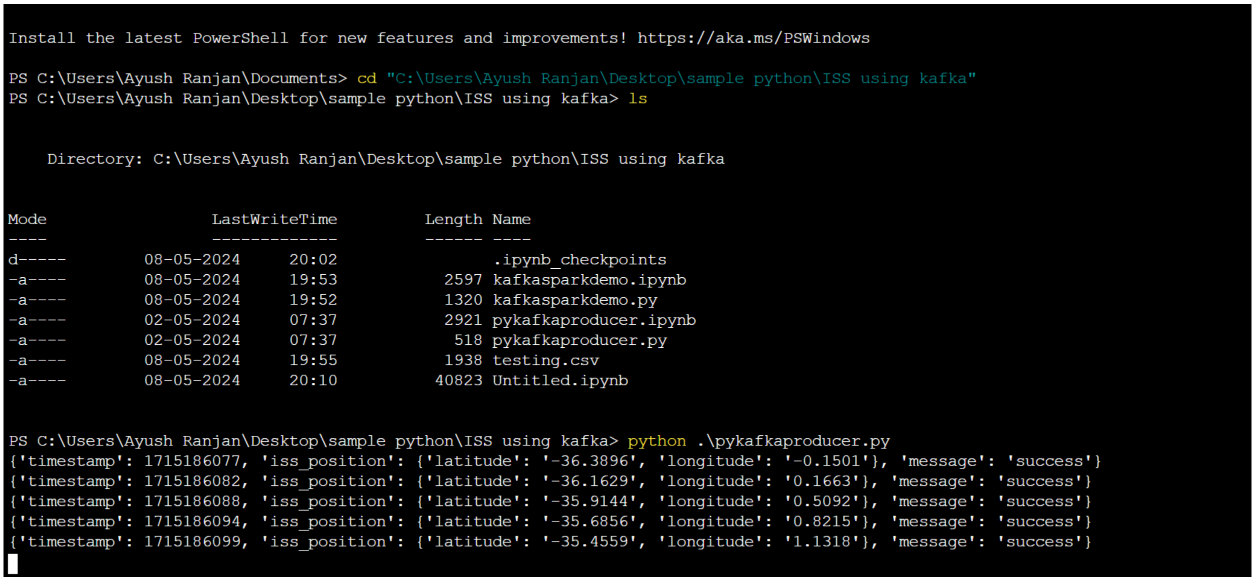


```
[2024-05-08 22:02:05,686] INFO [TransactionCoordinator id=0] Startup complete. (kafka.coordinator.transaction.TransactionCoordin
ator)
[2024-05-08 22:02:05,686] INFO [Transaction Marker Channel Manager 0]: Starting (kafka.coordinator.transaction.TransactionMarker
ChannelManager)
[2024-05-08 22:02:05,720] INFO [ExpirationReaper-0-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperation
Reaper)
[2024-05-08 22:02:05,736] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificationListener$C
hangeEventProcessThread)
[2024-05-08 22:02:05,775] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Starting socket server acceptors and processors (
kafka.network.SocketServer)
[2024-05-08 22:02:05,775] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started data-plane acceptor and processor(s) for
endpoint : ListenerName(PLAINTEXT) (kafka.network.SocketServer)
[2024-05-08 22:02:05,775] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started socket server acceptors and processors (k
afka.network.SocketServer)
log4j:ERROR Failed to rename [C:\kafka/logs/state-change.log] to [C:\kafka/logs/state-change.log.2024-05-08-19].
[2024-05-08 22:02:05,775] INFO Kafka version: 3.2.1 (org.apache.kafka.common.utils.AppInfoParser)
[2024-05-08 22:02:05,775] INFO Kafka commitId: b172a0a94f4ebb9f (org.apache.kafka.common.utils.AppInfoParser)
[2024-05-08 22:02:05,775] INFO Kafka startTimeMs: 1715185925775 (org.apache.kafka.common.utils.AppInfoParser)
[2024-05-08 22:02:05,803] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
[2024-05-08 22:02:05,892] INFO [BrokerToControllerChannelManager broker=0 name=alterPartition]: Recorded new controller, from no
w on will use broker DESKTOP-RGD3QAU:9092 (id: 0 rack: null) (kafka.server.BrokerToControllerRequestThread)
[2024-05-08 22:02:05,892] INFO [BrokerToControllerChannelManager broker=0 name=forwarding]: Recorded new controller, from now on
 will use broker DESKTOP-RGD3QAU:9092 (id: 0 rack: null) (kafka.server.BrokerToControllerRequestThread)
[2024-05-08 22:02:05,900] INFO [ReplicaFetcherManager on broker 0] Removed fetcher for partitions Set(testtopic-0) (kafka.server
.ReplicaFetcherManager)
[2024-05-08 22:02:05,915] INFO [Partition testtopic-0 broker=0] Log loaded for partition testtopic-0 with initial high watermark
 247 (kafka.cluster.Partition)
```

3. Kafka Producer:

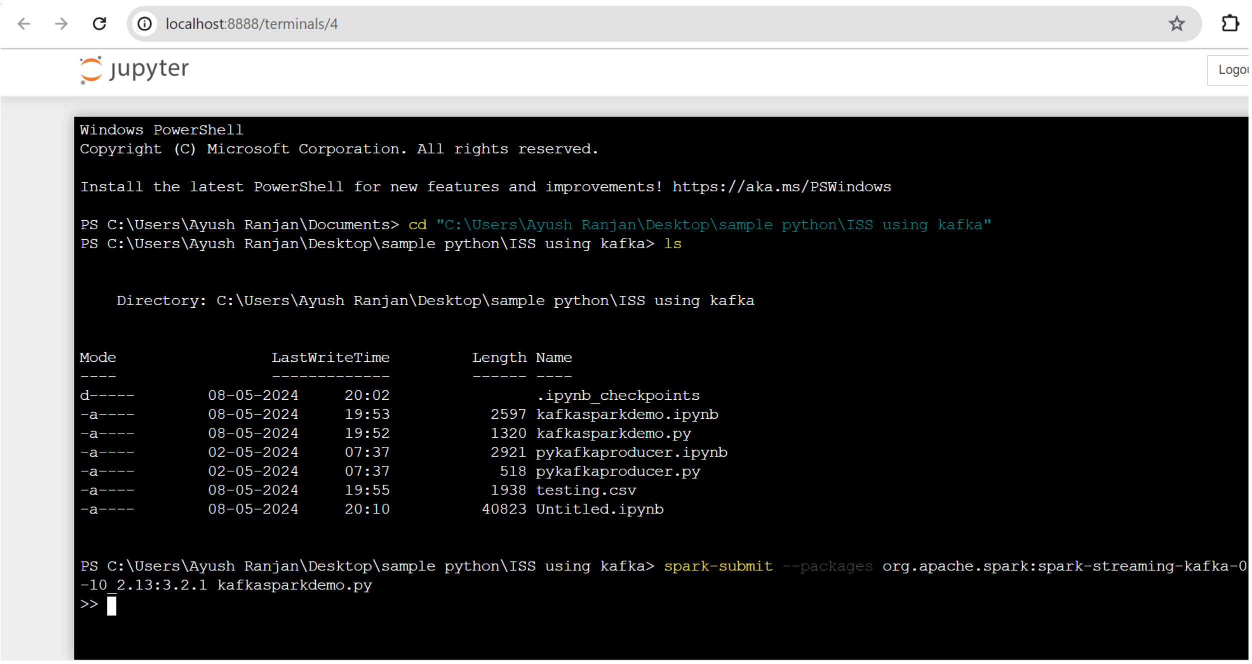


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Ayush Ranjan\Documents> cd "C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka"
PS C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka> ls


    Directory: C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08-05-2024     20:02                .ipynb_checkpoints
-a----        08-05-2024     19:53           2597 kafkasparkdemo.ipynb
-a----        08-05-2024     19:52           1320 kafkasparkdemo.py
-a----        02-05-2024     07:37           2921 pykafkaproducer.ipynb
-a----        02-05-2024     07:37            518 pykafkaproducer.py
-a----        08-05-2024     19:55           1938 testing.csv
-a----        08-05-2024     20:10          40823 Untitled.ipynb


PS C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka> python .\pykafkaproducer.py
```

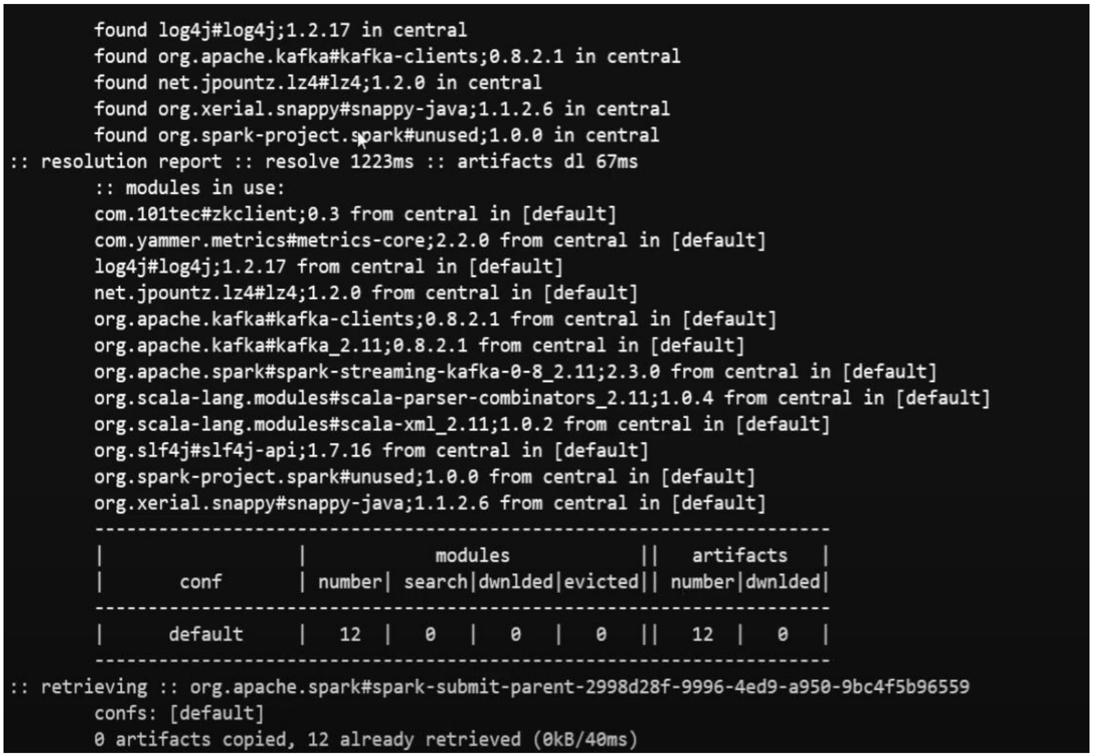


```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Ayush Ranjan\Documents> cd "C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka"
PS C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka> ls


    Directory: C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08-05-2024     20:02                .ipynb_checkpoints
-a----        08-05-2024     19:53           2597 kafkasparkdemo.ipynb
-a----        08-05-2024     19:52           1320 kafkasparkdemo.py
-a----        02-05-2024     07:37           2921 pykafkaproducer.ipynb
-a----        02-05-2024     07:37            518 pykafkaproducer.py
-a----        08-05-2024     19:55           1938 testing.csv
-a----        08-05-2024     20:10          40823 Untitled.ipynb


PS C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka> python .\pykafkaproducer.py
{'timestamp': 1715186077, 'iss_position': {'latitude': '-36.3896', 'longitude': '-0.1501'}, 'message': 'success'}
{'timestamp': 1715186082, 'iss_position': {'latitude': '-36.1629', 'longitude': '0.1663'}, 'message': 'success'}
{'timestamp': 1715186088, 'iss_position': {'latitude': '-35.9144', 'longitude': '0.5092'}, 'message': 'success'}
{'timestamp': 1715186094, 'iss_position': {'latitude': '-35.6856', 'longitude': '0.8215'}, 'message': 'success'}
{'timestamp': 1715186099, 'iss_position': {'latitude': '-35.4559', 'longitude': '1.1318'}, 'message': 'success'}
```

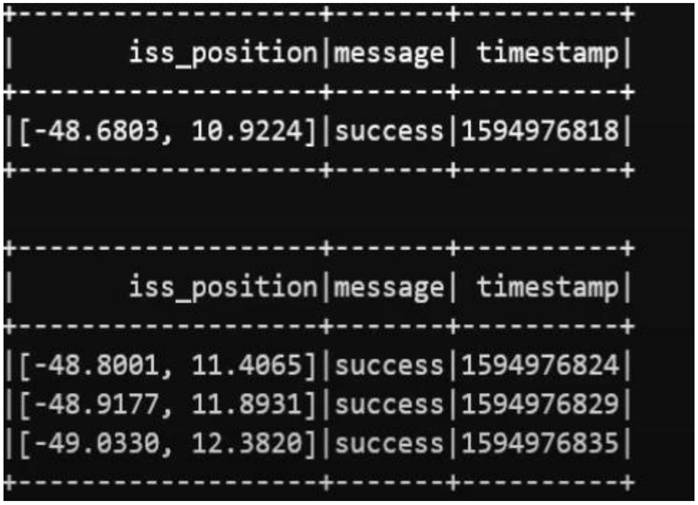
4. Kafka consumer:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Ayush Ranjan\Documents> cd "C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka"
PS C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka> ls


    Directory: C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        08-05-2024     20:02                .ipynb_checkpoints
-a----        08-05-2024     19:53           2597 kafkasparkdemo.ipynb
-a----        08-05-2024     19:52           1320 kafkasparkdemo.py
-a----        02-05-2024     07:37           2921 pykafkaproducer.ipynb
-a----        02-05-2024     07:37            518 pykafkaproducer.py
-a----        08-05-2024     19:55           1938 testing.csv
-a----        08-05-2024     20:10          40823 Untitled.ipynb


PS C:\Users\Ayush Ranjan\Desktop\sample python\ISS using kafka> spark-submit --packages org.apache.spark:spark-streaming-kafka-0
-10_2.13:3.2.1 kafkasparkdemo.py
>>
```



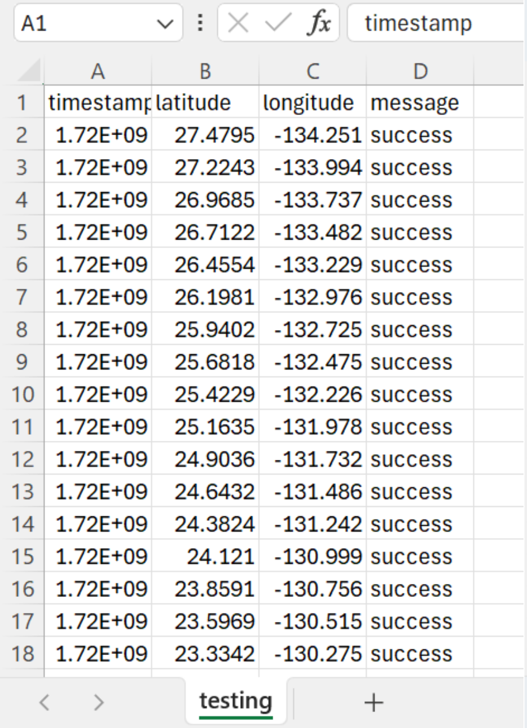
```
        found log4j#log4j;1.2.17 in central
        found org.apache.kafka#kafka-clients;0.8.2.1 in central
        found net.jpountz.lz4#lz4;1.2.0 in central
        found org.xerial.snappy#snappy-java;1.1.2.6 in central
        found org.spark-project.spark#unused;1.0.0 in central
:: resolution report :: resolve 1223ms :: artifacts dl 67ms
        :: modules in use:
        com.101tec#zkclient;0.3 from central in [default]
        com.yammer.metrics#metrics-core;2.2.0 from central in [default]
        log4j#log4j;1.2.17 from central in [default]
        net.jpountz.lz4#lz4;1.2.0 from central in [default]
        org.apache.kafka#kafka-clients;0.8.2.1 from central in [default]
        org.apache.kafka#kafka_2.11;0.8.2.1 from central in [default]
        org.apache.spark#spark-streaming-kafka-0-8_2.11;2.3.0 from central in [default]
        org.scala-lang.modules#scala-parser-combinators_2.11;1.0.4 from central in [default]
        org.scala-lang.modules#scala-xml_2.11;1.0.2 from central in [default]
        org.slf4j#slf4j-api;1.7.16 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.2.6 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   12  |   0   |   0   |   0   ||   12  |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-2998d28f-9996-4ed9-a950-9bc4f5b96559
        confs: [default]
        0 artifacts copied, 12 already retrieved (0kB/40ms)
```



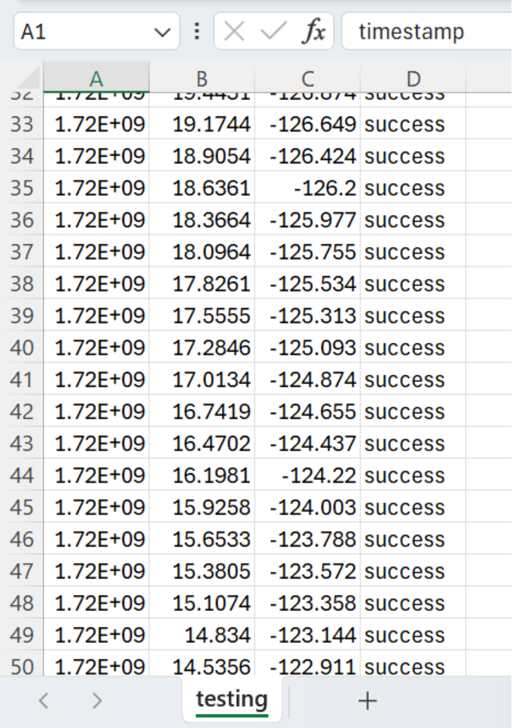
```
+------------------+-------+----------+
|      iss_position|message| timestamp|
+------------------+-------+----------+
|[-48.6803, 10.9224]|success|1594976818|
+------------------+-------+----------+

+------------------+-------+----------+
|      iss_position|message| timestamp|
+------------------+-------+----------+
|[-48.8001, 11.4065]|success|1594976824|
|[-48.9177, 11.8931]|success|1594976829|
|[-49.0330, 12.3820]|success|1594976835|
+------------------+-------+----------+
```

5. CSV WRITTEN THROUGH KAFKASPARKDEMO(KAFKA CONSUMER):

Real time location of International Space Station

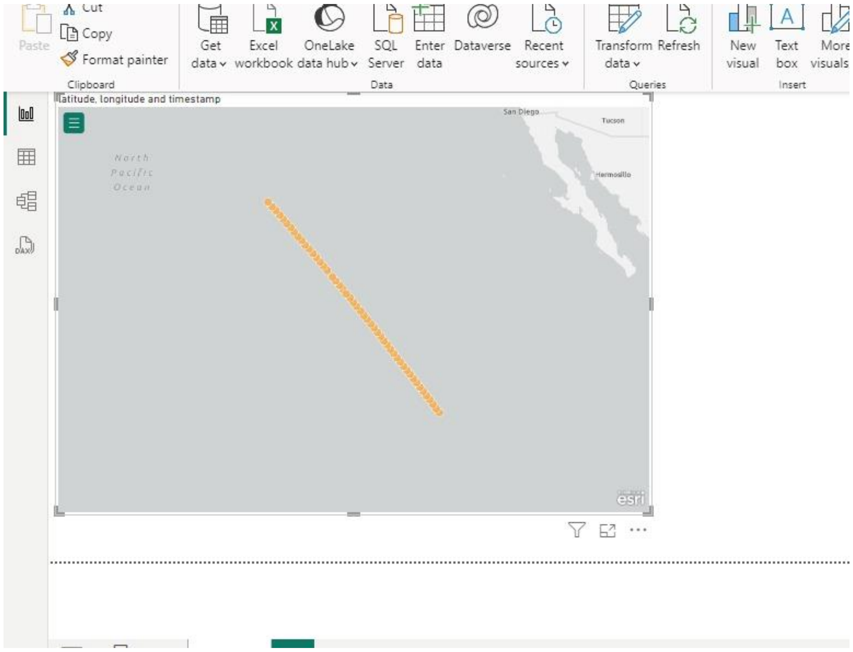


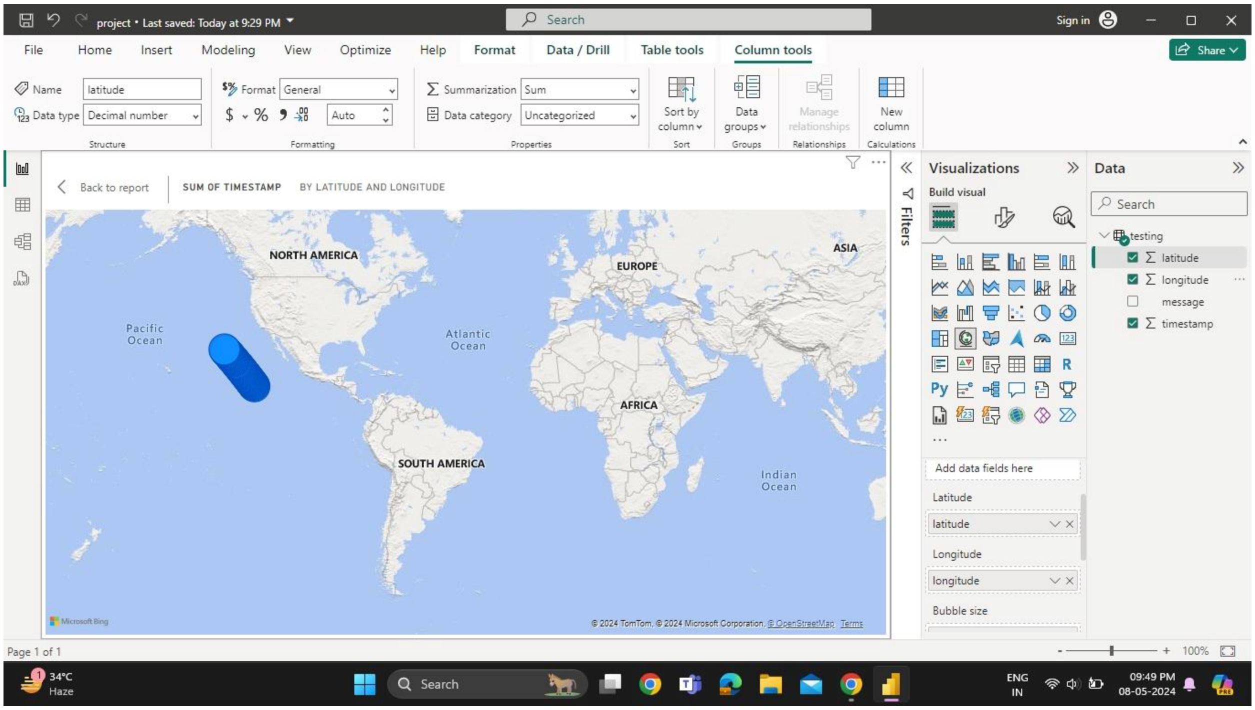
	A	B	C	D
1	timestamp	latitude	longitude	message
2	1.72E+09	27.4795	-134.251	success
3	1.72E+09	27.2243	-133.994	success
4	1.72E+09	26.9685	-133.737	success
5	1.72E+09	26.7122	-133.482	success
6	1.72E+09	26.4554	-133.229	success
7	1.72E+09	26.1981	-132.976	success
8	1.72E+09	25.9402	-132.725	success
9	1.72E+09	25.6818	-132.475	success
10	1.72E+09	25.4229	-132.226	success
11	1.72E+09	25.1635	-131.978	success
12	1.72E+09	24.9036	-131.732	success
13	1.72E+09	24.6432	-131.486	success
14	1.72E+09	24.3824	-131.242	success
15	1.72E+09	24.121	-130.999	success
16	1.72E+09	23.8591	-130.756	success
17	1.72E+09	23.5969	-130.515	success
18	1.72E+09	23.3342	-130.275	success



	A	B	C	D
32	1.72E+09	19.4431	-126.874	success
33	1.72E+09	19.1744	-126.649	success
34	1.72E+09	18.9054	-126.424	success
35	1.72E+09	18.6361	-126.2	success
36	1.72E+09	18.3664	-125.977	success
37	1.72E+09	18.0964	-125.755	success
38	1.72E+09	17.8261	-125.534	success
39	1.72E+09	17.5555	-125.313	success
40	1.72E+09	17.2846	-125.093	success
41	1.72E+09	17.0134	-124.874	success
42	1.72E+09	16.7419	-124.655	success
43	1.72E+09	16.4702	-124.437	success
44	1.72E+09	16.1981	-124.22	success
45	1.72E+09	15.9258	-124.003	success
46	1.72E+09	15.6533	-123.788	success
47	1.72E+09	15.3805	-123.572	success
48	1.72E+09	15.1074	-123.358	success
49	1.72E+09	14.834	-123.144	success
50	1.72E+09	14.5356	-122.911	success

6. Visualization of Real time location through POWER-BI:

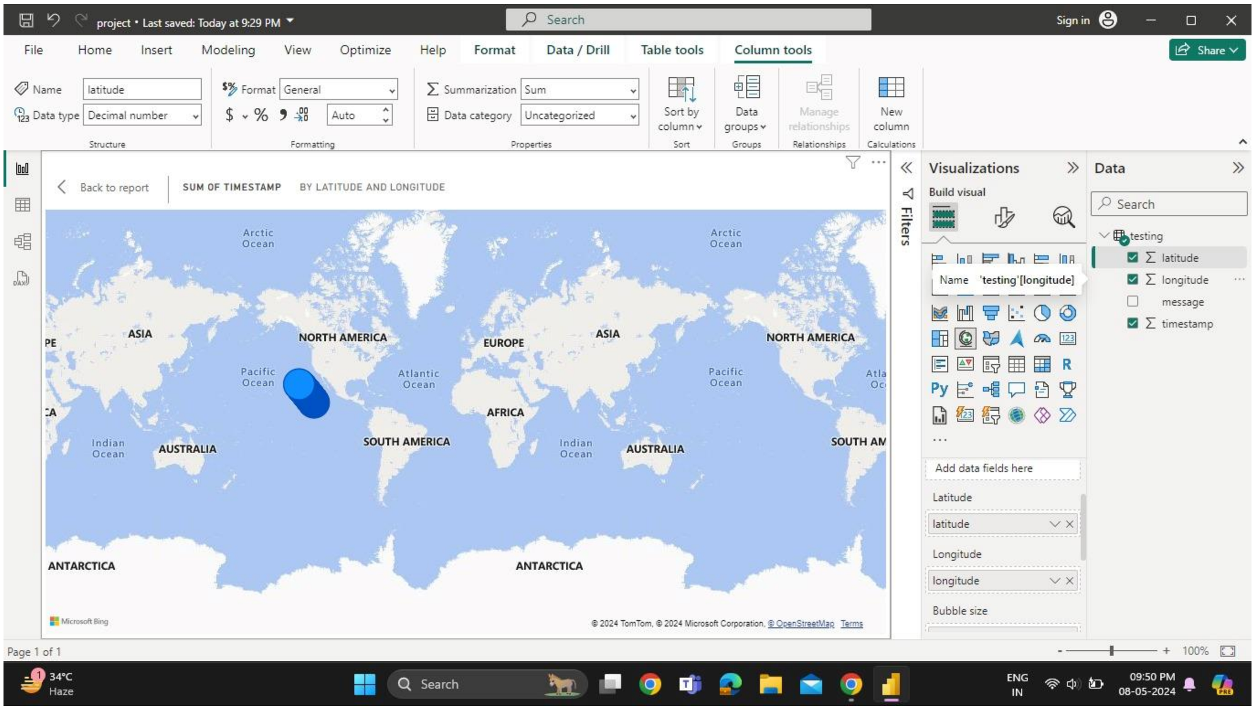




Name: latitude
Data type: Decimal number
Format: General
$ % Auto
Summarization: Sum
Data category: Uncategorized

testing
- ☑ Σ latitude
- ☑ Σ longitude
- □ message
- ☑ Σ timestamp

Latitude: latitude
Longitude: longitude



project • Last saved: Today at 9:29 PM

Name: latitude

Data type: Decimal number

Format: General

Summarization: Sum

Data category: Uncategorized

SUM OF TIMESTAMP BY LATITUDE AND LONGITUDE

Name 'testing'[longitude]

testing
- ☑ Σ latitude
- ☑ Σ longitude
- □ message
- ☑ Σ timestamp

Latitude: latitude

Longitude: longitude

Bubble size

6. Challenges

- Real-Time Data Processing: Processing real-time telemetry data from various sources and ensuring timely updates can be challenging. Delays or bottlenecks in data processing may lead to outdated information about the International Space Station's (ISS) current position and trajectory.

- Integration Complexity: Integrating PySpark with Kafka and ensuring seamless communication between different components of the system can be complex. Managing configurations, dependencies, and compatibility issues across these technologies requires careful planning and expertise.

- Scalability: Ensuring that the tracking system can handle increasing volumes of data and user requests while maintaining optimal performance is crucial. Scaling the system horizontally to accommodate growing demand and data volumes without sacrificing performance requires robust architecture and efficient resource management.

Validation and Verification: Validating the accuracy and reliability of the tracking system's predictions against ground-truth data and real-world observations is essential. Conducting thorough validation tests and verification processes to ensure the system's outputs align with actual ISS movements and events is challenging but necessary for building user trust and confidence in the system.

7. Future Enhancements

- Predictive Trajectory Analysis: Implementing predictive analytics to forecast the International Space Station's (ISS) trajectory more accurately. By incorporating advanced machine learning algorithms and historical data analysis, the system can anticipate potential deviations in the ISS's orbit and provide early warnings for trajectory adjustments or collision avoidance maneuvers.
- Enhanced Sensor Integration: Integrating additional sensors and telemetry data sources to gather more comprehensive information about the ISS's environmental conditions and operational status. By capturing data on factors such as radiation levels, temperature variations, and structural integrity, the system can provide more detailed insights into the ISS's health and safety.
- Automated Anomaly Detection: Developing automated anomaly detection algorithms to identify irregularities or anomalies in the ISS's telemetry data. By continuously monitoring sensor readings and system diagnostics, the system can detect potential malfunctions or anomalies in real-time, enabling prompt troubleshooting and maintenance interventions.
- Integration with Ground Control Systems: Strengthening integration with ground control systems and mission control centers to facilitate seamless coordination and communication between space agencies and the ISS. By enabling bi-directional data exchange and command execution, the system can support collaborative decision-making and operational planning for ISS missions.

8. Conclusion

In conclusion, the development of our real-time tracking system for the International Space Station (ISS) marks a significant advancement in space exploration and monitoring capabilities. By harnessing the power of PySpark and Kafka technologies, along with machine learning algorithms and real-time visualization tools, we have created a robust platform for accurately tracking the ISS's movement and trajectory.

Through seamless integration with telemetry data sources and ground-based tracking stations, our system delivers timely updates and insights into the ISS's orbit and operational status. By providing users with real-time information about the ISS's location, velocity, and environmental conditions, we contribute to enhancing safety protocols, mission planning, and scientific research endeavors in space.

Looking ahead, we remain committed to continuous refinement and optimization of our tracking system. We will explore opportunities to enhance predictive analytics, improve data accuracy, and expand coverage by integrating with additional satellite networks and sensor technologies. With our system, we aim to empower space agencies, researchers, and enthusiasts worldwide to gain a deeper understanding of the ISS's activities and contribute to advancements in space exploration and technology.