Project Title: Fashion Supply Chain Optimization: Analyzing Order History for Holiday Inventory Management for an online store.

By :Geetika Gupta

FUTURE READY TALENT

https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/4914658340971203/2559883118436597/8373299318609809/latest.html

(DASHBOARD LINK)

Problem Statement :

Bquick Shopping, an online fashion store, faces the challenge of efficiently managing their supply chain to meet the increasing demand during the upcoming holiday season. They need a comprehensive solution to analyze their order history and ensure they have sufficient inventory.

This project aims to develop a supply chain dashboard that utilizes data from BSinventory.json and BSorders.json files stored in AWS S3. The data will be processed using Azure Data Factory pipelines, validated, and moved to the appropriate locations. Finally, the data will be transformed, validated using Azure Functions, Databricks and integrated into an Azure Data Lake Storage to support inventory monitoring and optimization. A comprehensive dashboard will be developed using Databricks for in-depth analysis, enabling better decision-making regarding inventory management.

Services/Tools Used: Azure Data Factory, Azure Functions, Azure Data Lake Storage Gen2, Azure Key Vault, App Registration , AWS S3, Databricks

Problem Description :

In the rapidly evolving world of e-commerce, meeting customer demands, especially during peak seasons like the holidays, is crucial for business success. The "Fashion Supply Chain Optimization" project is designed to streamline and enhance the supply chain for an online fashion store- Bquick Shopping Online Store, allowing the organization to efficiently manage inventory during the upcoming holiday season.

The process begins with gathering critical data from various sources, including BSinventory.json, BSorders.json hosted on AWS S3. Azure Data Factory (ADF) pipelines are employed to facilitate the seamless movement of these files to Azure Data Lake Storage (ADLS). Once the data is in ADLS, validation is performed using Azure Functions, allowing for the identification of any discrepancies or issues. Files that meet the validation criteria are moved to the Staging folder, while those with errors are routed to the Rejected folder.

Next, the data in the Staging area is processed and integrated into an ADLS Gen2, providing a centralized repository for further analysis. Utilizing Azure Databricks, a comprehensive and insightful dashboard is created, allowing the supply team to delve into the order history and inventory details.

Finally, to make this valuable dashboard accessible and user-friendly, Databricks tools are used. This project empowers the online fashion store to optimize its supply chain, enhance customer satisfaction, and maximize profitability during the high-demand holiday season.

Architecture Flow :-

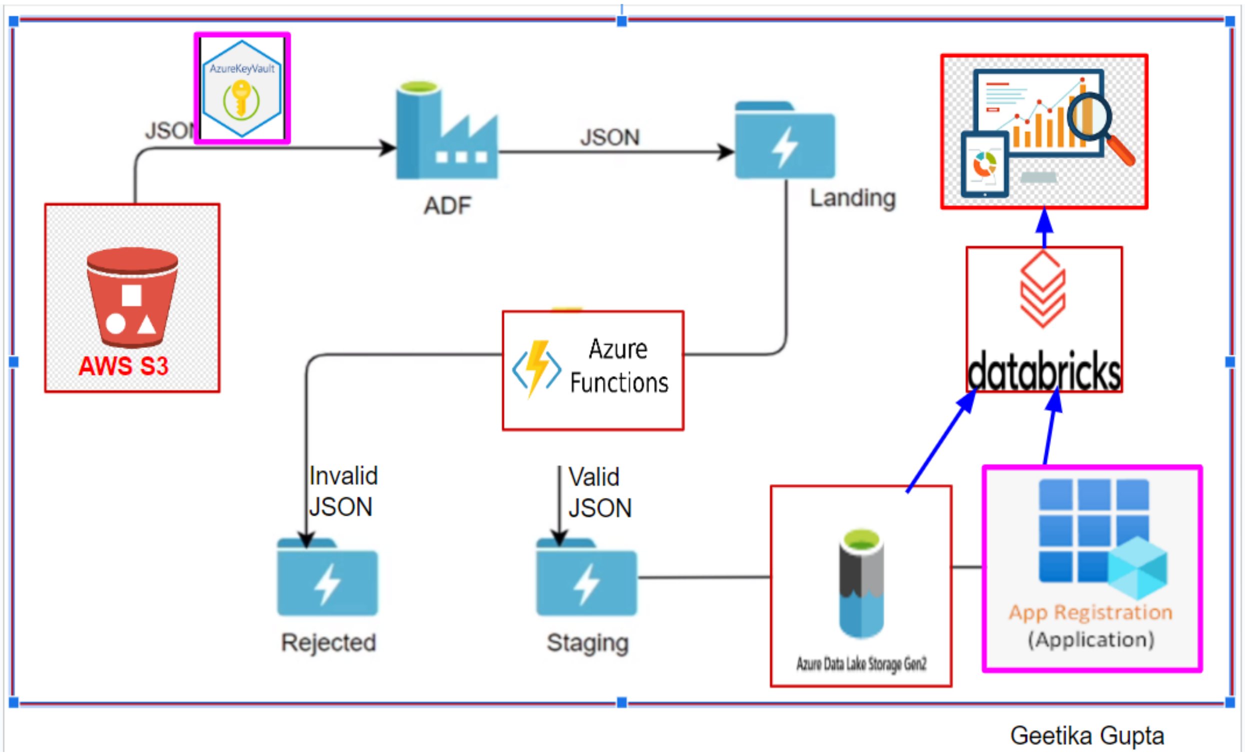
The architecture flow for the "Fashion Supply Chain Optimization: Analyzing Order History for Holiday Inventory Management" project involves several steps to seamlessly manage data from source to visualization. Here's a high-level architecture flow for this project:

1. **Data Ingestion and Extraction:**
    - Data files (BSinventory.json, BSorders.json, and BSupdateorders.json) are stored in AWS S3.
    - Azure Data Factory (ADF) is configured to periodically fetch the data from AWS S3 and move it to Azure Data Lake Storage (ADLS).
2. **Data Validation and Transformation:**
    - Azure Functions are triggered to validate and transform the ingested data.
    - Valid data is moved to the Staging folder in ADLS, and invalid data is sent to the Rejected folder.
3. **Data Integration and Storage:**
    - Data in the Staging area is processed and integrated into an Azure SQL Database for centralized storage.
4. **Data Analysis and Dashboard Creation:**
    - Azure Databricks is used to analyze the integrated data in Azure SQL Database.
    - Databricks performs advanced analytics and transforms the data into a suitable format for visualization.
5. **Dashboard Visualization:**
    - A dashboard is created using visualization tools in Databricks.
    - The dashboard provides insights into order history, inventory levels, product popularity, and other relevant metrics.
6. **Deployment and Visualization:**
    - The dashboard is published on Databricks to make it accessible over the web.

This architecture ensures a streamlined flow of data from various sources, effective data processing, and meaningful visualization for data-driven decision-making in the fashion supply chain.

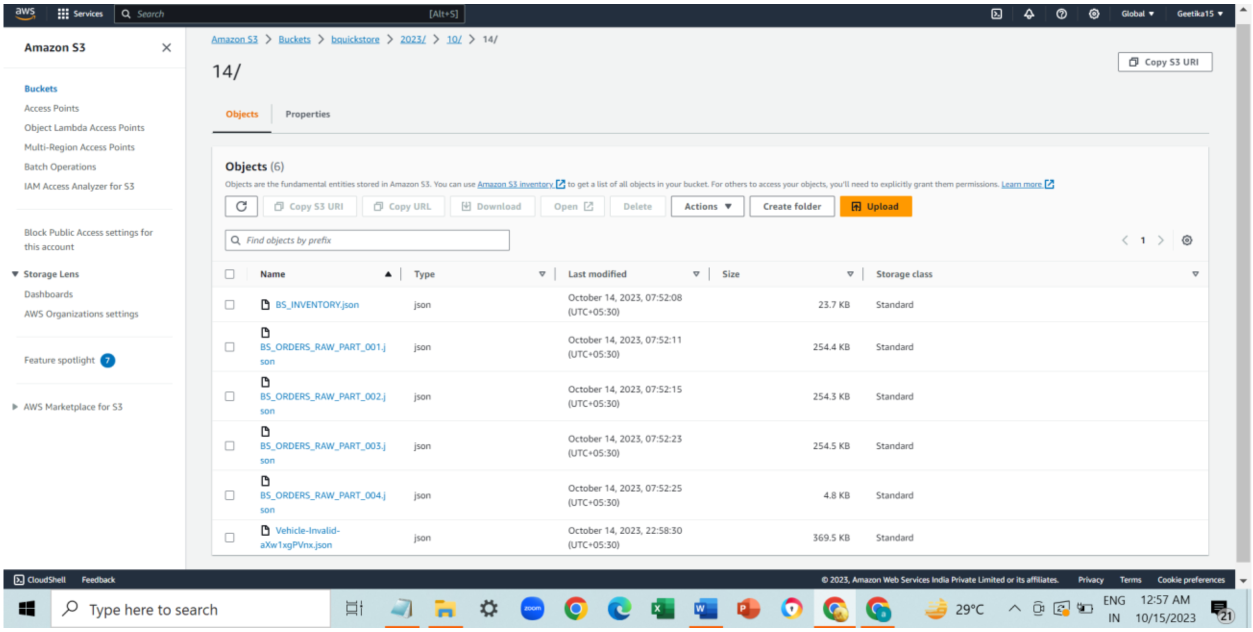
ARCHITECTURE DIAGRAM

Fashion Supply Chain Optimization: Analyzing Order History for Holiday Inventory Management for an online store.



Project Phases:

- **Data Ingestion and Transfer:**
- **AWS S3 BUCKET DATA FILES-JSON FORMAT**



Amazon S3 > Buckets > bquickstore > 2023/ > 10/ > 14/

14/

Objects (6)

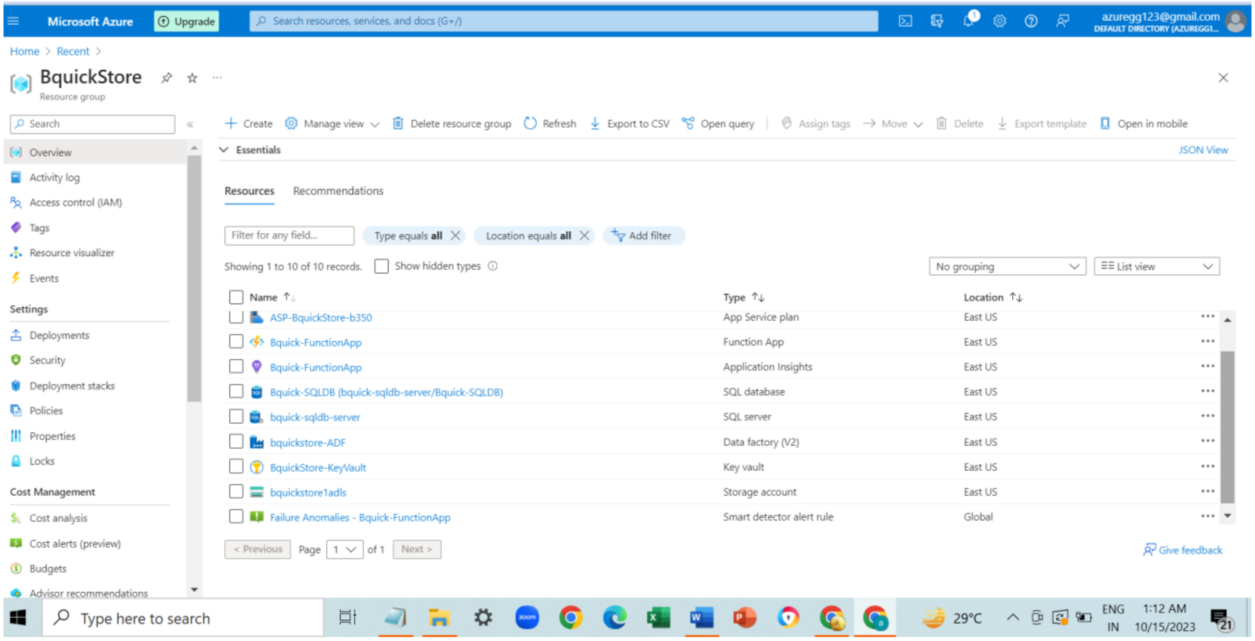
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	BS_INVENTORY.json	json	October 14, 2023, 07:52:08 (UTC+05:30)	23.7 KB	Standard
□	BS_ORDERS_RAW_PART_001.json	json	October 14, 2023, 07:52:11 (UTC+05:30)	254.4 KB	Standard
□	BS_ORDERS_RAW_PART_002.json	json	October 14, 2023, 07:52:15 (UTC+05:30)	254.3 KB	Standard
□	BS_ORDERS_RAW_PART_003.json	json	October 14, 2023, 07:52:23 (UTC+05:30)	254.5 KB	Standard
□	BS_ORDERS_RAW_PART_004.json	json	October 14, 2023, 07:52:25 (UTC+05:30)	4.8 KB	Standard
□	Vehicle-Invalid-aXw1xgPVnx.json	json	October 14, 2023, 22:58:30 (UTC+05:30)	369.5 KB	Standard

© 2023, Amazon Web Services India Private Limited or its affiliates.

- Set Up Data Ingestion: Create data ingestion pipelines using Azure Data Factory to transfer data from AWS to Azure securely.I have stored the access key and secret access key from AWS S3, in Azure Key Vault, as best practices.Set up an Azure Data Factory pipeline using copy activity, to ingest data from the third-party IoT devices hosted on AWS. copy activity is used to move data from AWS S3 buckets to Azure Storage ADLS Gen2 input/landing folder.

- RESOURCE GROUP AND RESOURCES IN AZURE PORTAL:-

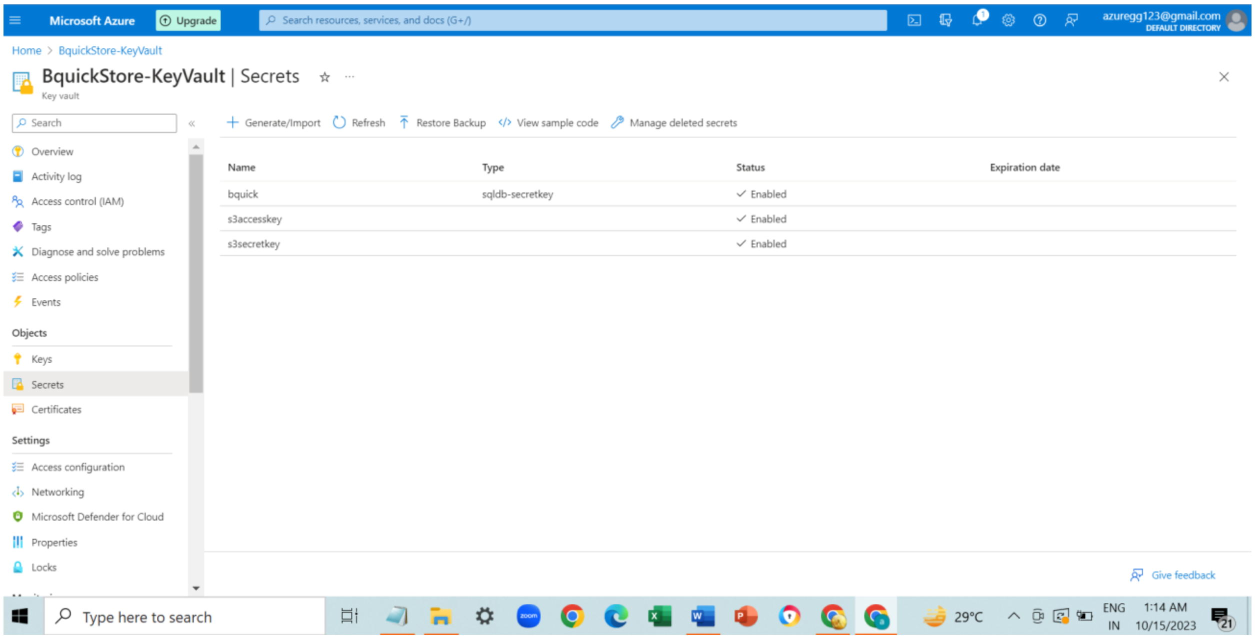


BquickStore
Resource group

Showing 1 to 10 of 10 records. □ Show hidden types

□	Name	Type	Location
□	ASP-BquickStore-b350	App Service plan	East US
□	Bquick-FunctionApp	Function App	East US
□	Bquick-FunctionApp	Application Insights	East US
□	Bquick-SQLDB (bquick-sqldb-server/Bquick-SQLDB)	SQL database	East US
□	bquick-sqldb-server	SQL server	East US
□	bquickstore-ADF	Data factory (V2)	East US
□	BquickStore-KeyVault	Key vault	East US
□	bquickstore1adls	Storage account	East US
□	Failure Anomalies - Bquick-FunctionApp	Smart detector alert rule	Global

AZURE KEY VAULT

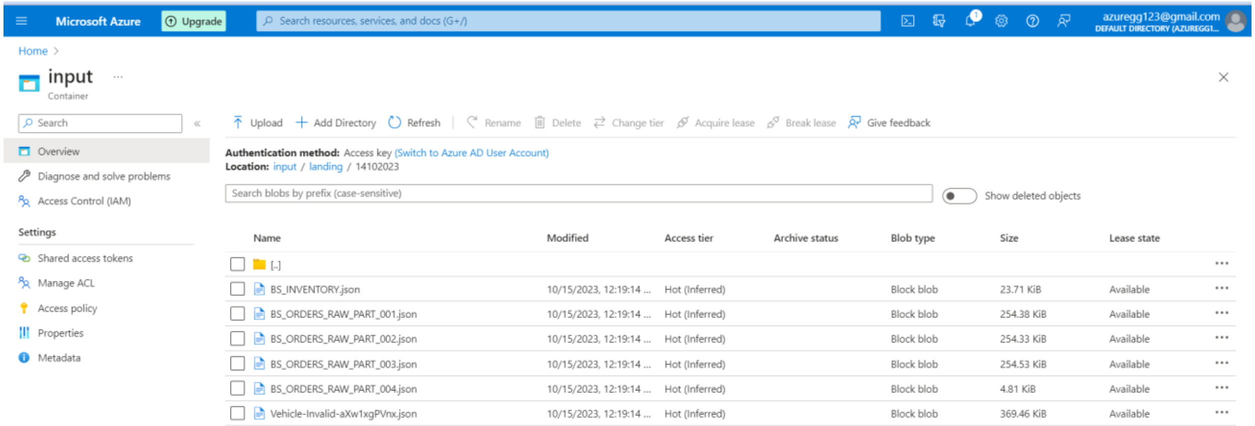

Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
azuregg123@gmail.com
DEFAULT DIRECTORY
Home > BquickStore-KeyVault

BquickStore-KeyVault | Secrets

Key vault

Name	Type	Status	Expiration date
bquick	sqldb-secretkey	✓ Enabled	
s3accesskey		✓ Enabled	
s3secretkey		✓ Enabled	

- **Data Landing Zone:**
  - Create a landing folder in Azure Storage to temporarily store the incoming JSON data files. You can organize this storage account with a container specifically for landing data.
  - input/landing


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
azuregg123@gmail.com
DEFAULT DIRECTORY (AZUREGG1...
Home >

input

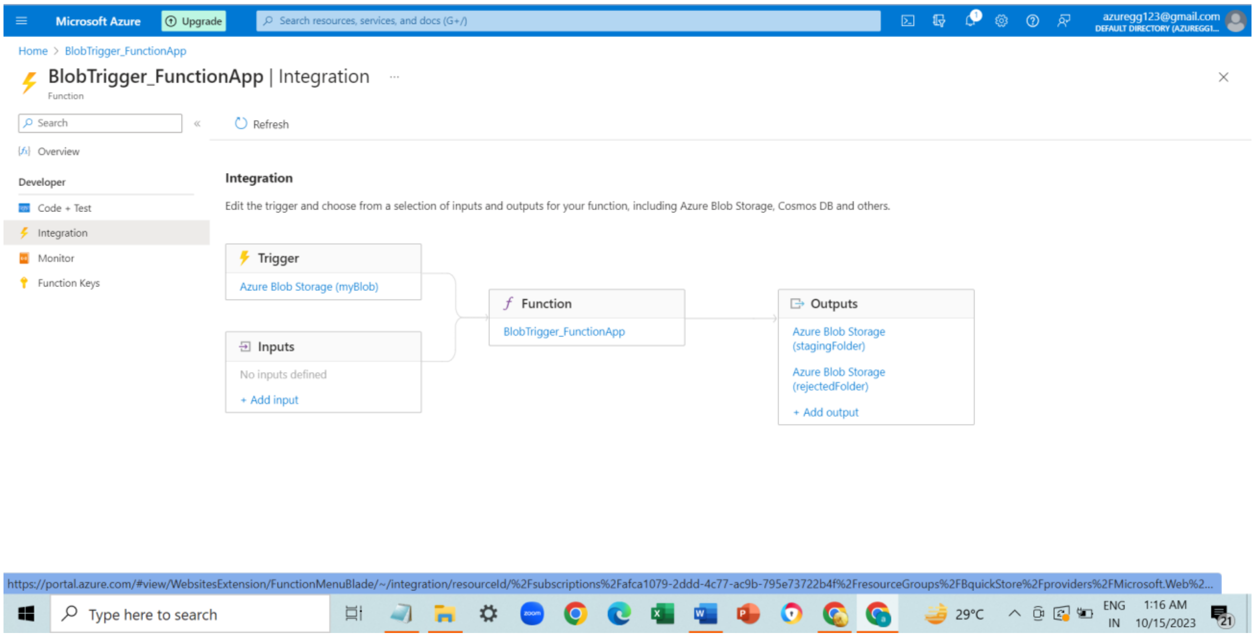
Container

Authentication method: Access key (Switch to Azure AD User Account)
Location: input / landing / 14102023

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
BS_INVENTORY.json	10/15/2023, 12:19:14 ...	Hot (Inferred)		Block blob	23.71 KiB	Available
BS_ORDERS_RAW_PART_001.json	10/15/2023, 12:19:14 ...	Hot (Inferred)		Block blob	254.38 KiB	Available
BS_ORDERS_RAW_PART_002.json	10/15/2023, 12:19:14 ...	Hot (Inferred)		Block blob	254.33 KiB	Available
BS_ORDERS_RAW_PART_003.json	10/15/2023, 12:19:14 ...	Hot (Inferred)		Block blob	254.53 KiB	Available
BS_ORDERS_RAW_PART_004.json	10/15/2023, 12:19:14 ...	Hot (Inferred)		Block blob	4.81 KiB	Available
Vehicle-Invalid-aXw1xgPVnx.json	10/15/2023, 12:19:14 ...	Hot (Inferred)		Block blob	369.46 KiB	Available

- **Azure Function for Validation:**
  - Develop an Azure Function that uses a storage-based trigger (Blob Trigger-BlobTrigger1) to monitor the landing folder for incoming JSON files.

- When a new file arrives, the Azure Function is triggered to validate the JSON format and content. If validation fails, move the file to the "rejectedFolder" folder; otherwise, move it to the "stagingFolder" folder.



Edit the trigger and choose from a selection of inputs and outputs for your function, including Azure Blob Storage, Cosmos DB and others.

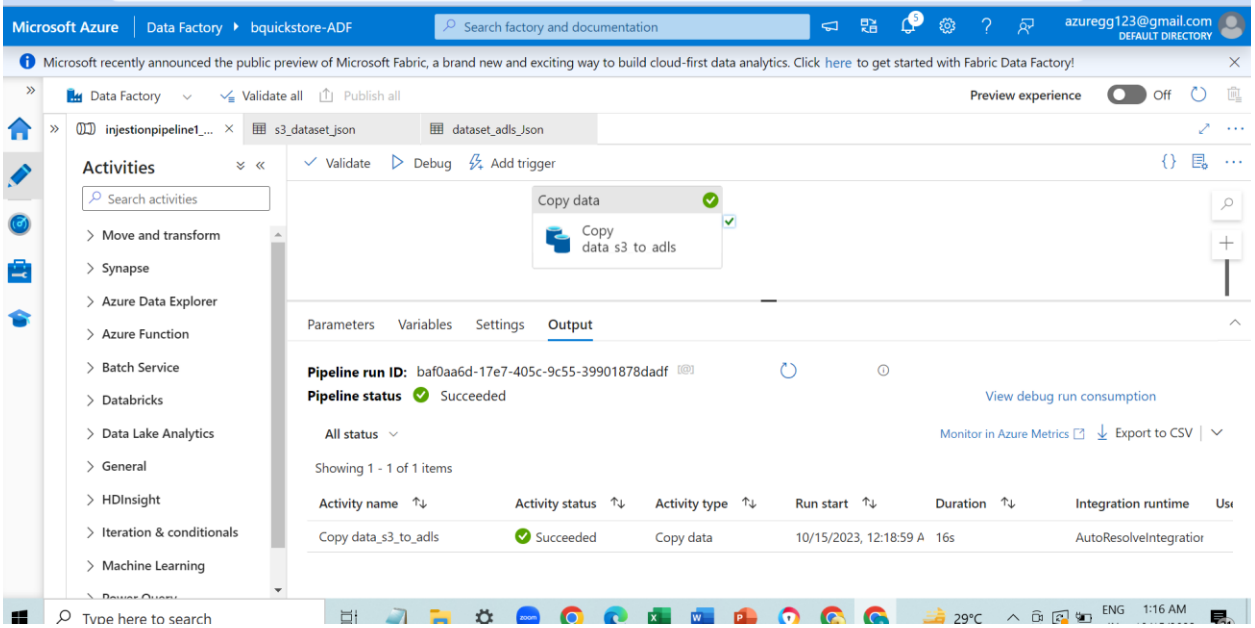
https://portal.azure.com/#view/WebsitesExtension/FunctionMenuBlade/~/integration/resourceId/%2Fsubscriptions%2Fafca1079-2ddd-4c77-ac9b-795e73722b4f%2FresourceGroups%2FBquickStore%2Fproviders%2FMicrosoft.Web%2...

- **Data Validation and Movement:**
    - Configure the Azure Function to perform JSON validation using built-in JSON parsing libraries and validation logic to check the correctness of JSON files.
    - Use Azure Function bindings to move files between folders based on the validation result. If validation passes, move the file to the "stagingFolder" folder; if it fails, move it to the "rejectedFolder" folder.

```
module.exports = async function (context, myBlob) {
  context.log("JavaScript blob trigger function processed blob \n Blob:");
  context.log("********Azure Function Started********");
  var result =true;
  try{
    context.log(myBlob.toString());
    JSON.parse(myBlob.toString().trim().replace('\n', ' '));
  }catch(exception){
    context.log(exception);
    result =false;
  }
  if(result){
    context.bindings.stagingFolder = myBlob.toString();
    context.log("********File Copied to Staging Folder Successfully********");
  } else{
    context.bindings.rejectedFolder = myBlob.toString();
    context.log("********Inavlid JSON File Copied to Rejected Folder Successfully********");
  }
context.log("*******Azure Function Ended Successfully*******");
};
```

- **Staging Data in Azure SQL DB:**
  - Create an Azure SQL Database to serve as the staging area for your data. Design the database schema to accommodate the JSON data structure.
  - Develop another ADF Pipeline that triggers when files are moved to the "stagingFolder" folder. This function reads the JSON data, transforms it if needed

PIPELINE RUNS:-



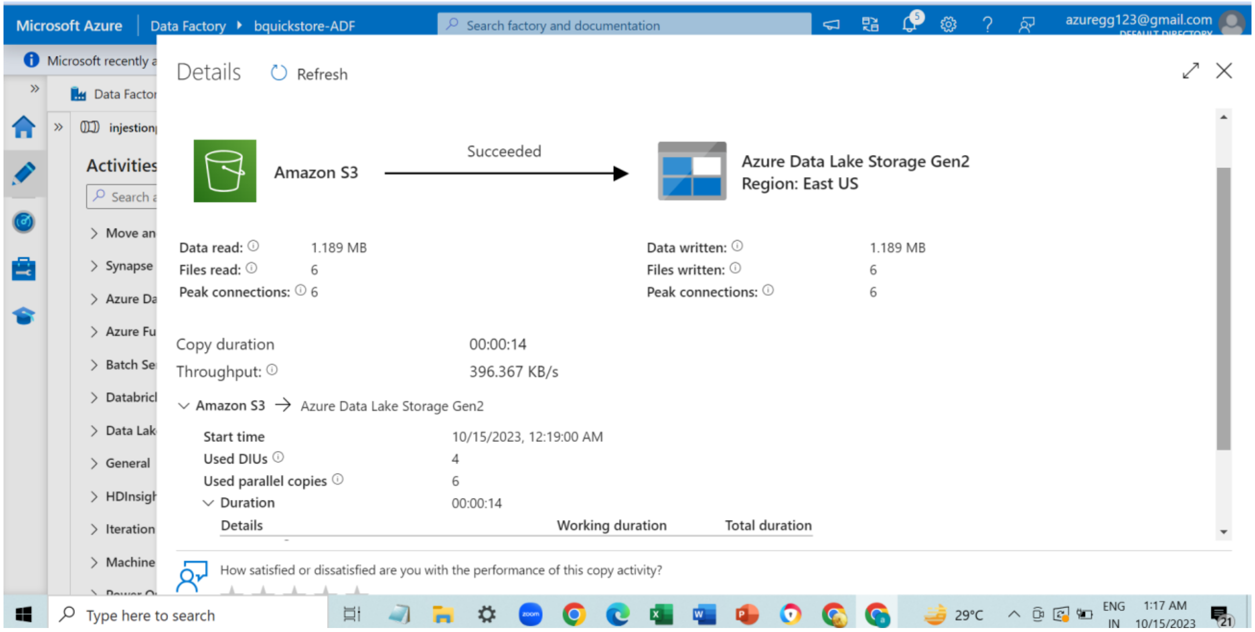
Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Pipeline run ID: baf0aa6d-17e7-405c-9c55-39901878dadf

Pipeline status ✓ Succeeded

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	Us
Copy data_s3_to_adls	✓ Succeeded	Copy data	10/15/2023, 12:18:59 A	16s	AutoResolveIntegratior	



Data read:	1.189 MB	Data written:	1.189 MB
Files read:	6	Files written:	6
Peak connections:	6	Peak connections:	6

Copy duration	00:00:14
Throughput:	396.367 KB/s

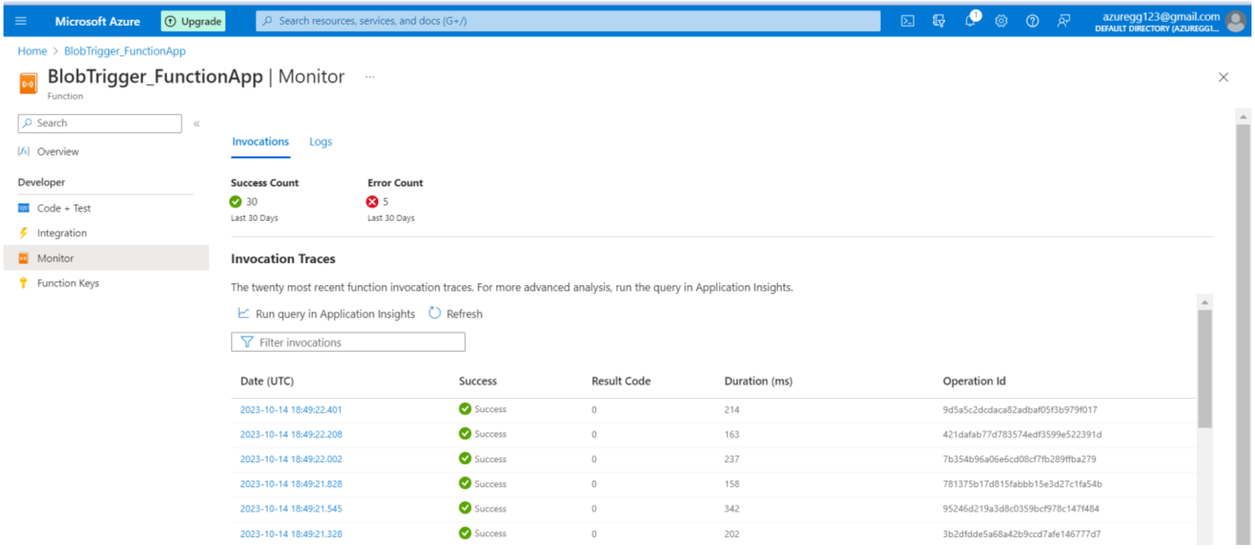
Amazon S3 → Azure Data Lake Storage Gen2

Start time	10/15/2023, 12:19:00 AM
Used DIUs	4
Used parallel copies	6
Duration	00:00:14

Details	Working duration	Total duration

How satisfied or dissatisfied are you with the performance of this copy activity?

TRIGGER RUNS —



BlobTrigger_FunctionApp | Monitor
Function

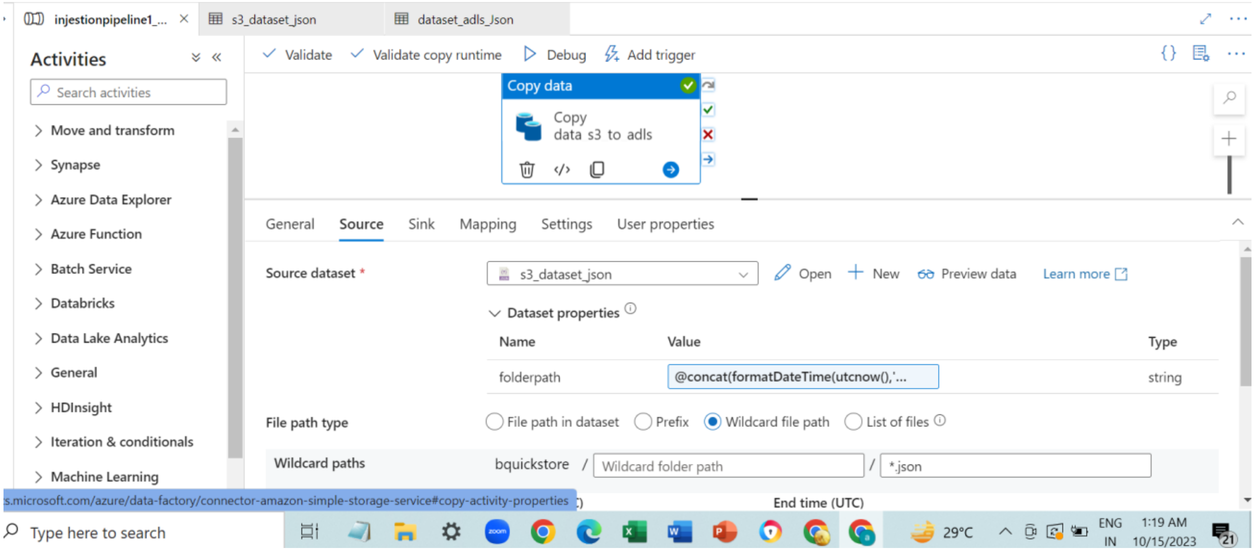
Success Count: 30 (Last 30 Days)
Error Count: 5 (Last 30 Days)

Invocation Traces

The twenty most recent function invocation traces. For more advanced analysis, run the query in Application Insights.

Date (UTC)	Success	Result Code	Duration (ms)	Operation Id
2023-10-14 18:49:22.401	✓ Success	0	214	9d5a5c2dcdaca82adbaf05f3b979f017
2023-10-14 18:49:22.208	✓ Success	0	163	421dafab77d783574edf3599e522391d
2023-10-14 18:49:22.002	✓ Success	0	237	7b354b96a06e6cd08cf7fb289ffba279
2023-10-14 18:49:21.828	✓ Success	0	158	781375b17d815fabbb15e3d27c1fa54b
2023-10-14 18:49:21.545	✓ Success	0	342	95246d219a3d8c0359bcf978c147f484
2023-10-14 18:49:21.328	✓ Success	0	202	3b2dfdde5a68a42b9ccd7afe146777d7

SOURCE:-



Source dataset *: s3_dataset_json

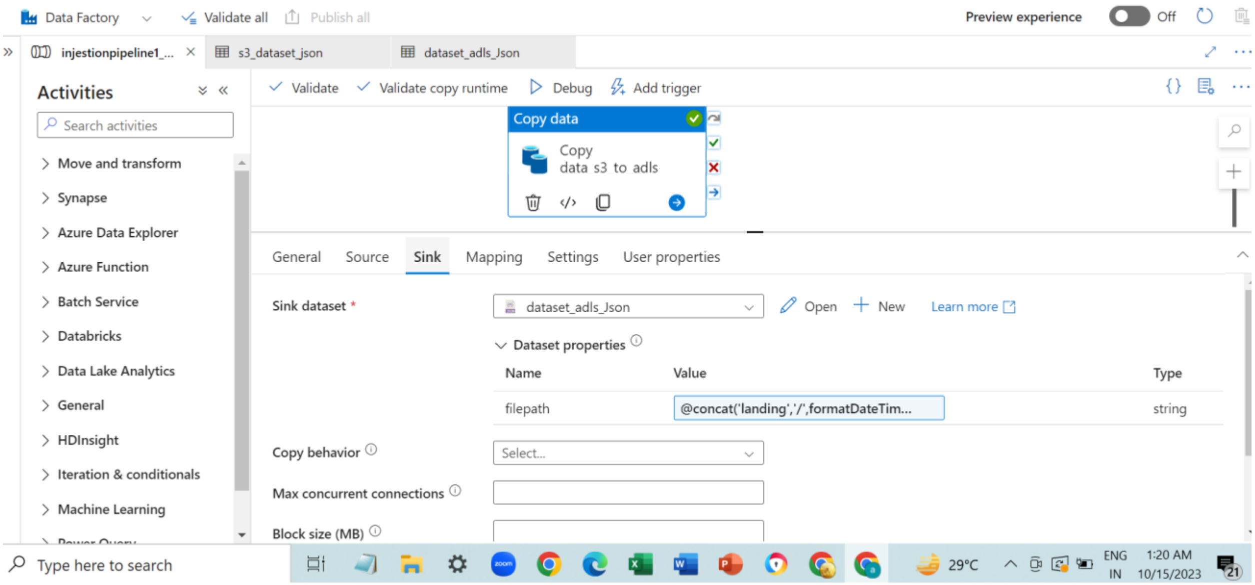
Dataset properties

Name	Value	Type
folderpath	@concat(formatDateTime(utcnow(),'...	string

File path type: ○ File path in dataset ○ Prefix ● Wildcard file path ○ List of files

Wildcard paths: bquickstore / Wildcard folder path / *.json

SINK



Sink dataset * dataset_adls_Json

Dataset properties

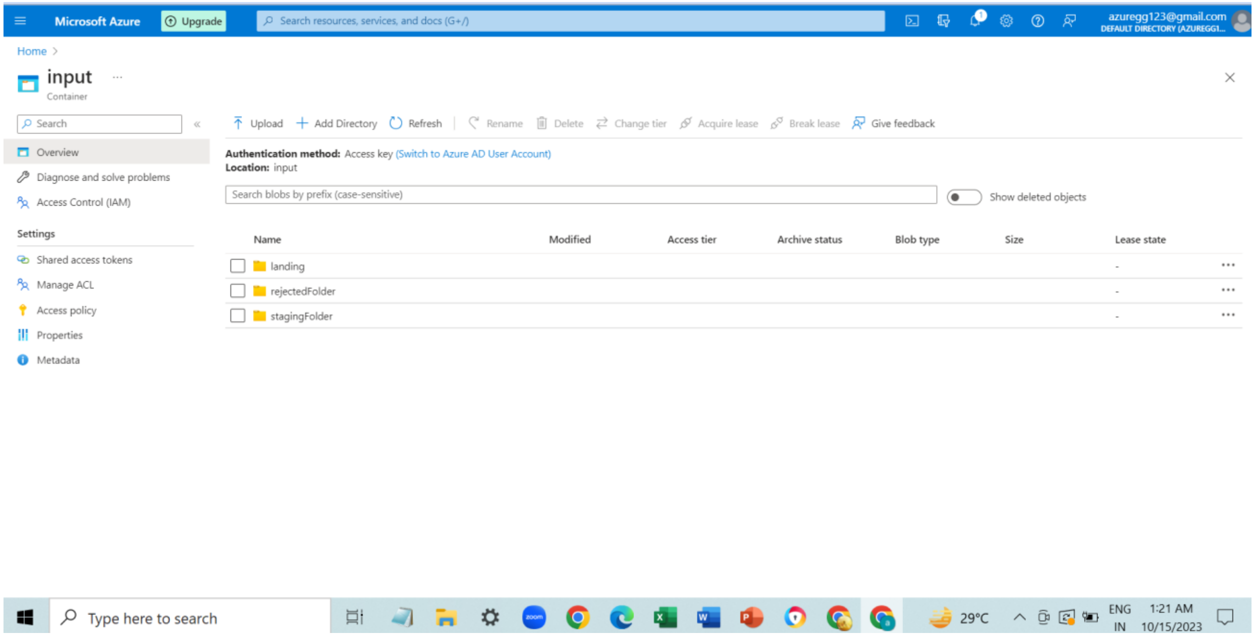
Name	Value	Type
filepath	@concat('landing','/',formatDateTim...	string

Copy behavior: Select...

Max concurrent connections:

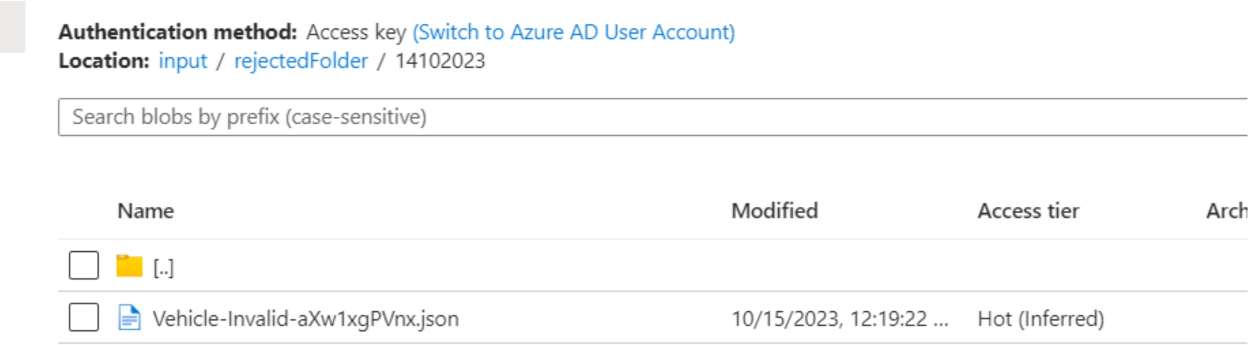
Block size (MB):

ADLS CONTAINER AFTER PIPELINE RUN:-



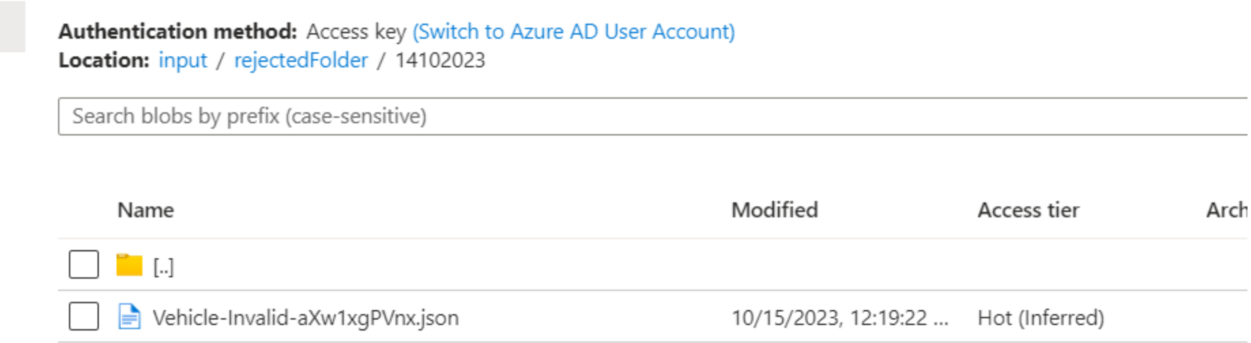
Authentication method: Access key (Switch to Azure AD User Account)
Location: input

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
landing						-
rejectedFolder						-
stagingFolder						-



Authentication method: Access key (Switch to Azure AD User Account)
Location: input / rejectedFolder / 14102023

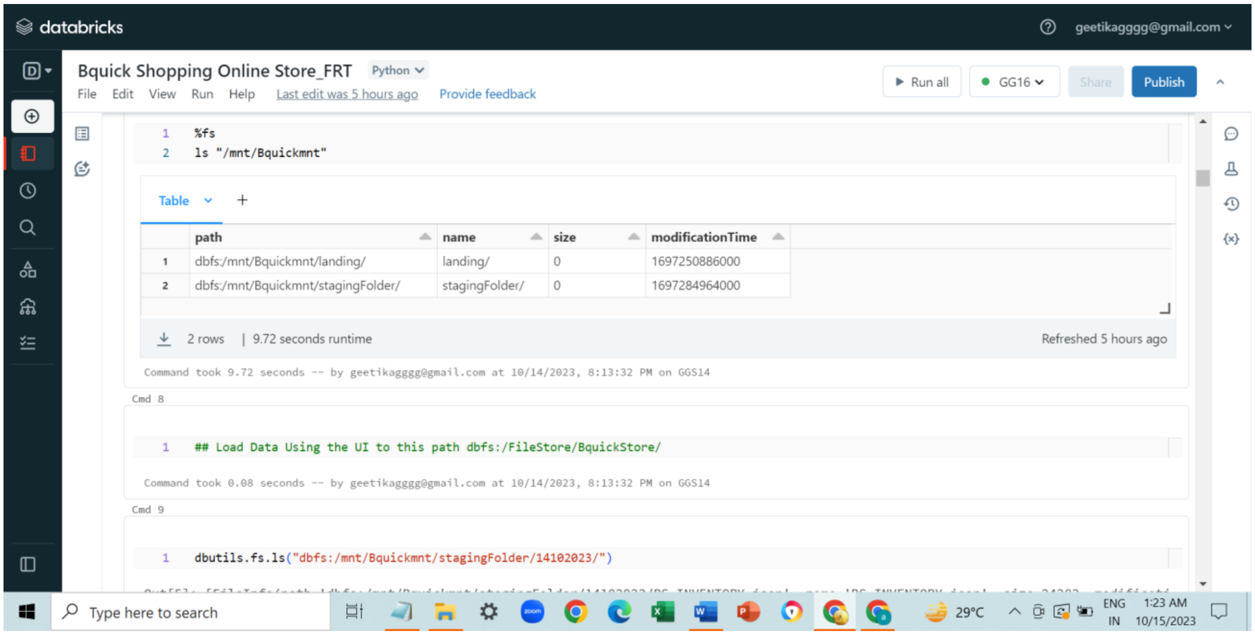
	Name	Modified	Access tier	Arch
□	[..]			
□	Vehicle-Invalid-aXw1xgPVnx.json	10/15/2023, 12:19:22 ...	Hot (Inferred)	



Authentication method: Access key (Switch to Azure AD User Account)
Location: input / rejectedFolder / 14102023

	Name	Modified	Access tier	Arch
□	[..]			
□	Vehicle-Invalid-aXw1xgPVnx.json	10/15/2023, 12:19:22 ...	Hot (Inferred)	

PIPELINE 1 SUCCESSFUL:-DATABRICKS MOUNTING



```
%fs
ls "/mnt/Bquickmnt"
```

	path	name	size	modificationTime
1	dbfs:/mnt/Bquickmnt/landing/	landing/	0	1697250886000
2	dbfs:/mnt/Bquickmnt/stagingFolder/	stagingFolder/	0	1697284964000

2 rows | 9.72 seconds runtime   Refreshed 5 hours ago

```
Command took 9.72 seconds -- by geetikagggg@gmail.com at 10/14/2023, 8:13:32 PM on GGS14
```

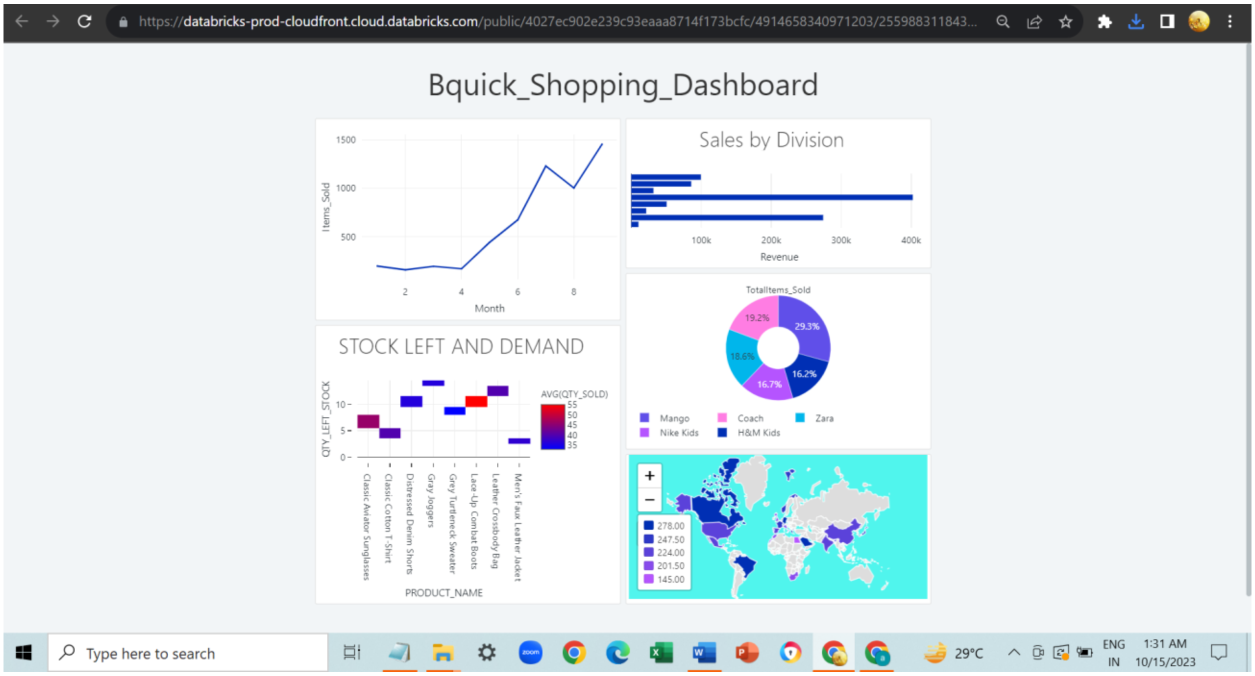
Cmd 8

```
## Load Data Using the UI to this path dbfs:/FileStore/BquickStore/
```

```
Command took 0.08 seconds -- by geetikagggg@gmail.com at 10/14/2023, 8:13:32 PM on GGS14
```

Cmd 9

```
dbutils.fs.ls("dbfs:/mnt/Bquickmnt/stagingFolder/14102023/")
```

https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/4914658340971203/255988311843...
Bquick_Shopping_Dashboard
1500
1000
500
Items_Sold
2
4
6
8
Month
Sales by Division
100k
200k
300k
400k
Revenue
TotalItems_Sold
19.2%
29.3%
18.6%
16.2%
16.7%
Mango
Coach
Zara
Nike Kids
H&M Kids
STOCK LEFT AND DEMAND
QTY_LEFT_STOCK
10
5
0
AVG(QTY_SOLD)
55
50
45
40
35
Classic Aviator Sunglasses
Classic Cotton T-Shirt
Distressed Denim Shorts
Gray Joggers
Grey Turtleneck Sweater
Lace-Up Combat Boots
Leather Crossbody Bag
Men's Faux Leather Jacket
PRODUCT_NAME
+
−
278.00
247.50
224.00
201.50
145.00
Type here to search
29°C
ENG
IN
1:31 AM
10/15/2023