

Microsoft Azure Databricks



Azure
Databricks

Cognitive Convergence is Subject Matter Expert in Office 365, Dynamics 365, SharePoint, Project Server, Power Platform: Power Apps-Power BI-Power Automate-Power Virtual Agents.

We offer Power BI consulting services covering solution architecture refinement, customization, integration, transformation, visualization and analytics to uncover insights hidden within data and enhance data exploration.

CONTENTS

Objectives.....	3
Background	3
DataBricks	3
What is Azure Databricks.....	3
Azure Databricks Workspace?	4
Azure DataBricks for Datascientists.....	5
<i>Optimized spark engine</i>	5
<i>Machine learning run time</i>	5
<i>MLflow</i>	5
<i>Choice of language</i>	5
<i>Seamless Collaboration</i>	5
<i>Native integrations with Azure services</i>	5
<i>Enterprise-grade security</i>	5
<i>Production-ready</i>	6
<i>Optimized environment</i>	6
<i>Easy to use</i>	6
Azure Databricks workspace Architecture.....	6
<i>Control Plane</i>	7
<i>Data Plane</i>	7
DataBricks utilities	9
Azure Databricks working demo	10
Azure Databricks workspace	10
<i>Notebook</i>	10
<i>Dashboard</i>	10
<i>Library</i>	10
<i>Experiment</i>	10



Create an Azure Databricks workspace	10
Create a Spark cluster in Databricks	11
Getting Data.....	12
<i>Visualizing Data in Databricks.....</i>	<i>13</i>
<i>Dashboard in Azure Databricks.....</i>	<i>15</i>
Integrating Azure Databrick Data with Power BI.....	16
<i>Power BI.....</i>	<i>19</i>
<i>Connection elements.....</i>	<i>19</i>
<i>Integrating with Power BI.....</i>	<i>20</i>
Reason to use Azure Databricks.....	22
<i>Familiar languages and environment</i>	<i>22</i>
<i>Higher productivity and collaboration</i>	<i>22</i>
<i>Integrates easily with the whole Microsoft stack.....</i>	<i>23</i>
<i>Extensive list of data sources.....</i>	<i>23</i>
<i>Suitable for small jobs too.....</i>	<i>23</i>
<i>Extensive support available</i>	<i>23</i>
Real Time Use Case.....	24
Azure Databricks Pricing.....	24
Conclusion	24
Contact us	24



OBJECTIVES

The objective of this paper is to explore Azure Databricks, what is its solution architecture and why to use azure Databricks. In this paper creating azure databrick workspace from azure portal, creating spark cluster in Azure Databricks from workspace, use python in new notebook to work with data and its integration with Power BI to load data from notebook to Power BI and create its reports will also discuss. The purpose of this white paper is to discuss how both Azure Databricks and Power BI can be used for Data Visualization tasks and how to connect clusters in Databricks to Power BI

BACKGROUND

In real life, the need to deliver data in an understandable format that provides actionable insights extends the needs of just Data Engineers and Scientists. With that in mind, how can Marketers, salesman and business executives to understand and utilize comprehensive analytics platforms such as Azure Databricks to perform day-to-day tasks? Thankfully, our clusters now can connect within Azure Databricks to BI tools such as Power BI.

DATABRICKS

Databricks was originally developed by the creators of Apache Spark and aims to deliver a unified platform where data scientists & engineers can work together to build end-to-end machine learning solutions from data discovery up to production. Databricks is a platform where users can log in & work. It's built on top of Apache Spark computing technology & can be mounted on premise or in a Cloud set-up giving the users any needed compute power to work in an abstracted and simplified way.

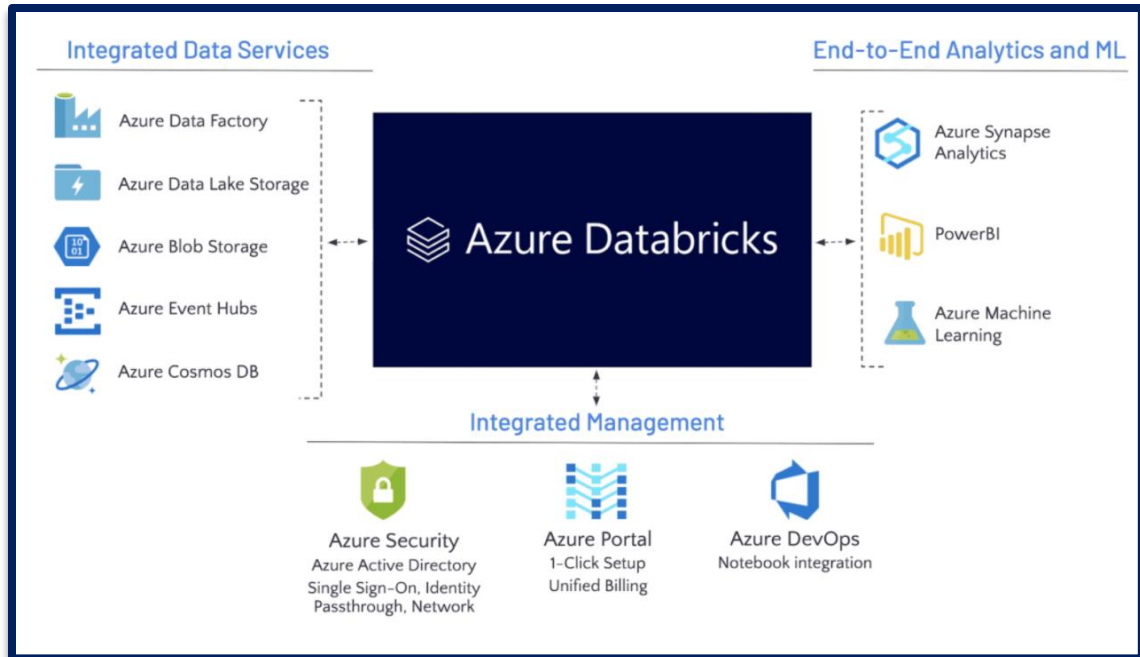
WHAT IS AZURE DATABRICKS

Azure Databricks is a data analytics platform optimized for the Microsoft Azure cloud services platform. Azure Data bricks provides the latest versions of Apache Spark and allows to seamlessly integrate with open source libraries. Spin up clusters and build quickly in a fully managed Apache Spark environment with the global scale and availability of Azure. Azure Databricks offers two environments for developing data intensive applications.

Azure Databricks is one of the most popular services in the Azure platform. It leverages Apache Spark to process data in a distributed environment, which can expedite the performance dramatically. Azure Databricks also support Delta Lake that is an open-sourced storage layer in a distributed environment. It can be used very similarly as most of the traditional database management systems because it supports ACID transactions.

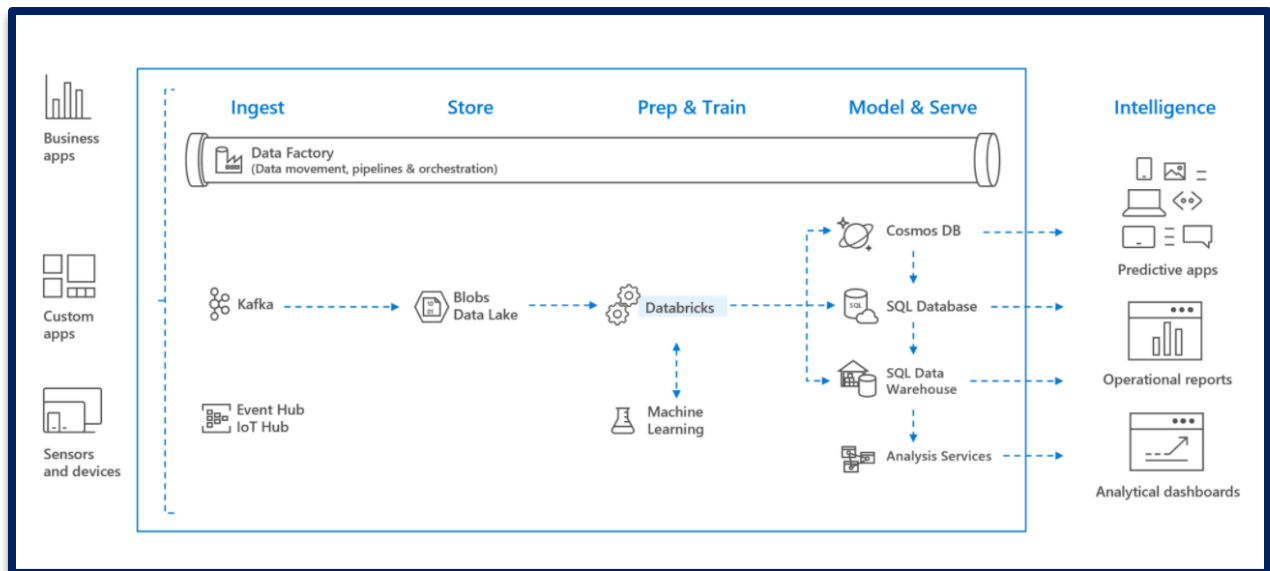


Apache Spark + Databricks + Enterprise Cloud = Azure Databricks



AZURE DATABRICKS WORKSPACE?

Databricks Azure Workspace is an analytics platform based on Apache Spark. For the big data pipeline, the data is ingested into Azure using Azure Data Factory. This data lands in a data lake and for analytics, we use Databricks to read data from multiple data sources and turn it into breakthrough insights.



AZURE DATABRICKS FOR DATASCIENTISTS

Optimized spark engine

Simple data processing on auto scaling infrastructure, powered by highly optimized Apache Spark™ for up to 50x performance gains.

Machine learning run time

One-click access to preconfigured machine learning environments for augmented machine learning with state-of-the-art and popular frameworks such as PyTorch, TensorFlow, and scikit-learn.

MLflow

Track and share experiments, reproduce runs, and manage models collaboratively from a central repository.

Choice of language

Use your preferred language, including Python, Scala, R, Spark SQL and .Net—whether you use server less or provisioned compute resources.

Seamless Collaboration

Quickly access and explore data, find and share new insights, and build models collaboratively with the languages and tools of your choice. Notebooks on Databricks are live and shared, with real-time collaboration, so that everyone in your organization can work with your data. Dashboards enable business users to call an existing job with new parameters. And Databricks integrates closely with Power BI for interactive visualization. All this is possible because Azure Databricks is backed by Azure Database and other technologies that enable highly concurrent access, fast performance and geo-replication.

Native integrations with Azure services

Complete your end-to-end analytics and machine learning solution with deep integration with Azure services such as Azure Data Factory, Azure Data Lake Storage, Azure Machine Learning, and Power BI.

Enterprise-grade security

Effortless native security protects your data where it lives and creates compliant, private, and isolated analytics workspaces across thousands of users and datasets.



Production-ready

Run and scale your most mission-critical data workloads with confidence on a trusted data platform, with ecosystem integrations for CI/CD and monitoring.

Optimized environment

Azure Databricks is optimized from the ground up for performance and cost-efficiency in the cloud. The Databricks Runtime adds several key capabilities to Apache Spark workloads that can increase performance and reduce costs by as much as 10-100x when running on Azure:

1. High-speed connectors to Azure storage services such as Azure Blob Store and Azure Data Lake, developed together with the Microsoft teams behind these services.
2. Auto-scaling and auto-termination for Spark clusters to automatically minimize costs.
3. Performance optimizations including caching, indexing, and advanced query optimization, which can improve performance by as much as 10-100x over traditional Apache Spark deployments in cloud or on premise environments.

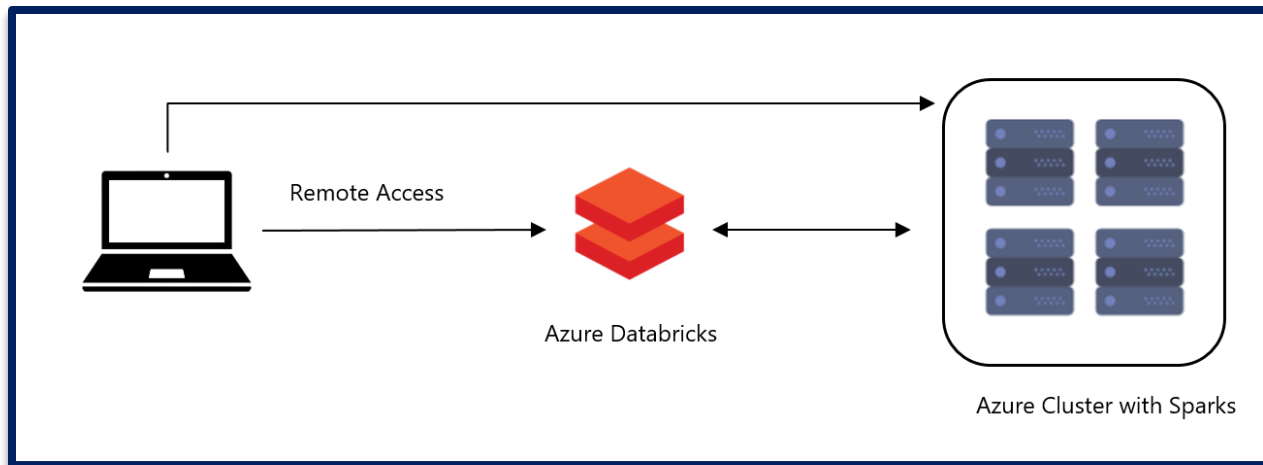
Easy to use

Azure Databricks comes packaged with interactive notebooks that let you connect to common data sources, run machine learning algorithms, and learn the basics of Apache Spark to get started quickly. It also features an integrated debugging environment to let you analyze the progress of your Spark jobs from within interactive notebooks, and powerful tools to analyze past jobs. Finally, other common analytics libraries, such as the Python and R data science stacks, are preinstalled so that you can use them with Spark to derive insights. We really believe that big data can become 10x easier to use, and we are continuing the philosophy started in Apache Spark to provide a unified, end-to-end platform.

AZURE DATABRICKS WORKSPACE ARCHITECTURE

Azure Databricks is architecturally a cloud service that allow to set up and use a cluster of azure instances with apache spark installed whit a master worker nodal dynamic- similar to a local Hadoop / Spark cluster.





Azure Databricks operates out of a **control plane** and a **data plane**.

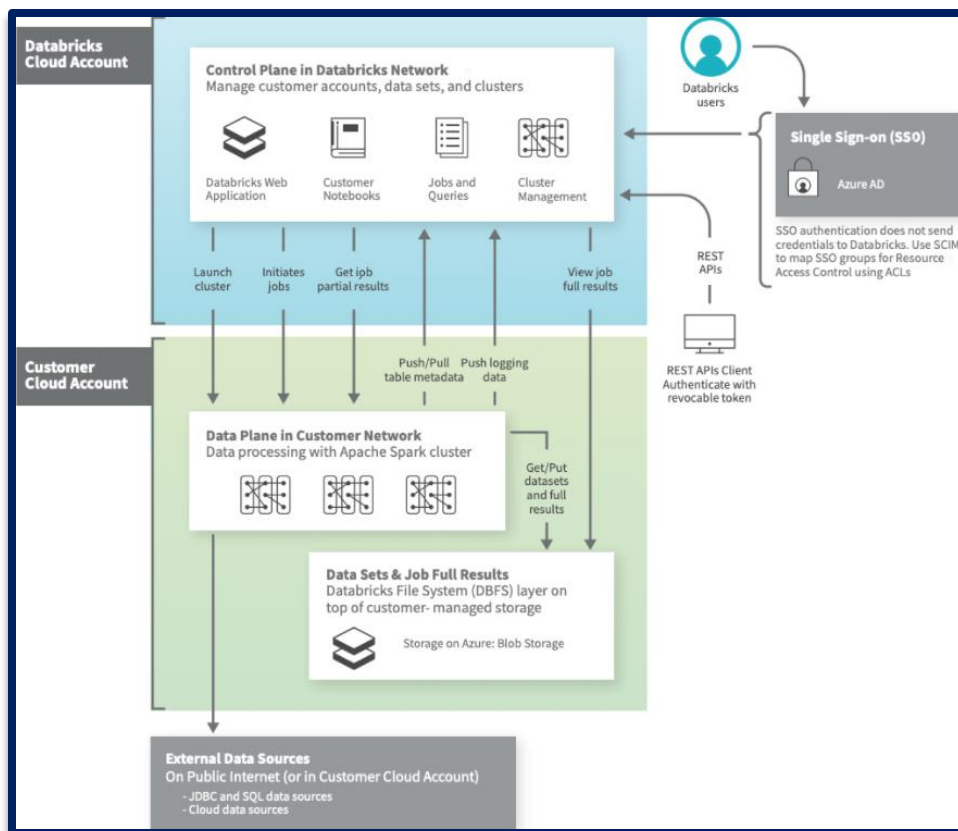
Control Plane

The control plane includes the backend services that Azure Databricks manages in its own Azure account. Notebook commands and many other workspace configurations are stored in the control plane and encrypted at rest.

Data Plane

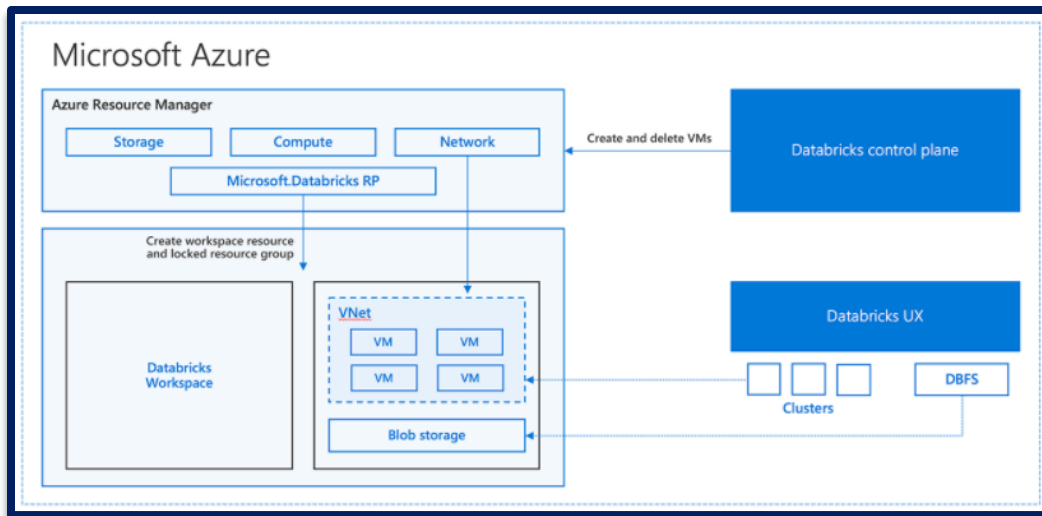
The data plane is managed by your Azure account and is where your data resides. This is also where data is processed. It allows Azure Databricks connectors so that clusters can connect to external data sources outside of Azure account to ingest data or for storage. Also allow to ingest data from external streaming data sources, such as events data, streaming data, IoT data, and more.





So how is Azure Databricks put together? At a high level, the service launches and manages worker nodes in each Azure customer's subscription, letting customers leverage existing management tools within their account.

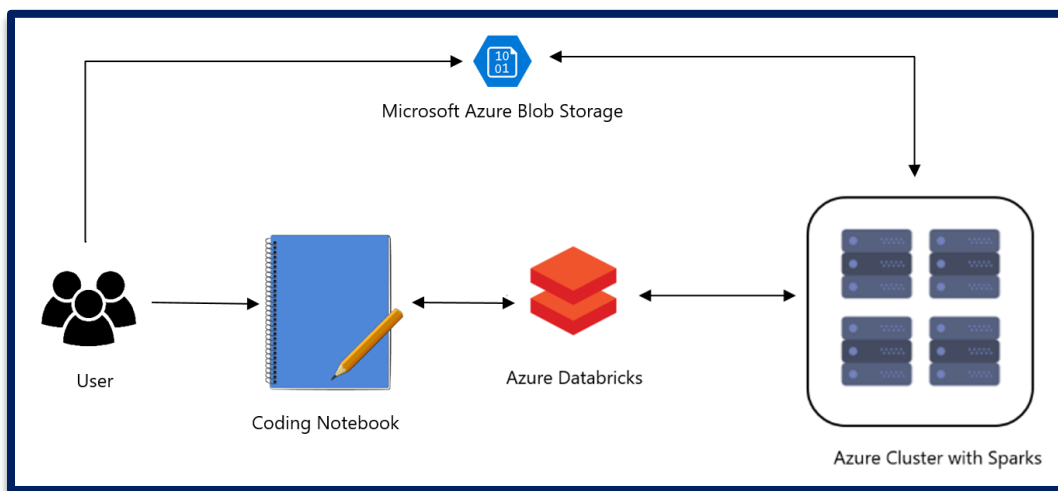
Specifically, when a customer launches a cluster via Databricks, a "Databricks appliance" is deployed as an Azure resource in the customer's subscription. The customer specifies the types of VMs to use and how many, but Databricks manages all other aspects. In addition to this appliance, a managed resource group is deployed into the customer's subscription that we populate with a VNet, a security group, and a storage account. These are concepts Azure users are familiar with. Once these services are ready, users can manage the Databricks cluster through the Azure Databricks UI or through features such as auto scaling. All metadata (such as scheduled jobs) is stored in an Azure Database with geo-replication for fault tolerance.



For users, this design means two things. First, they can easily connect Azure Databricks to any storage resource in their account, e.g., an existing Blob Store subscription or Data Lake. Second, Databricks is managed centrally from the Azure control center, requiring no additional setup.

DATABRICKS UTILITIES

Databricks utilities or DBUtils help us perform a variety of powerful tasks which include efficient object storage, chaining notebooks together and working with secrets. All DBUtils are available for notebooks for Python, Scala and R. DBUtils are not supported outside notebooks.



AZURE DATABRICKS WORKING DEMO

Microsoft Azure provides a multitude of services. It is often beneficial to combine multiple services together to approach use-case.

AZURE DATABRICKS WORKSPACE

A workspace is an environment for accessing all of Azure Databricks assets. A workspace organizes objects (notebooks, libraries, dashboards, and experiments) into folders and provides access to data objects and computational resources.

Notebook

A web-based interface to documents that contain runnable commands, visualizations, and narrative text.

Dashboard

An interface that provides organized access to visualizations.

Library

A package of code available to the notebook or job running on your cluster. Databricks runtimes include many libraries and you can add your own.

Experiment

A collection of MLflow runs for training a machine learning model.

CREATE AN AZURE DATABRICKS WORKSPACE

Create an Azure Databricks workspace using the Azure portal or the Azure CLI.

1. In the Azure portal, Create a resource > Analytics > Azure Databricks.
2. Under Azure Databricks Service, provide the values to create a Databricks workspace.



Create an Azure Databricks workspace ...

Basics Networking Advanced Tags Review + create

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Microsoft Partner Network

Resource group * ⓘ (New) dataBrick-quickstart
[Create new](#)

Instance Details

Workspace name * demo_databricks ✓

Region * East US

Pricing Tier * ⓘ Trial (Premium - 14-Days Free DBUs)

- **Workspace name** Provide a name for your Databricks workspace
- **Subscription** From the drop-down, select your Azure subscription.
- **Resource group** Specify whether you want to create a new resource group or use an existing one. A resource group is a container that holds related resources for an Azure solution. For more information, see Azure Resource Group overview.
- **Location** Select West US 2. For other available regions, see Azure services available by region.
- **Pricing Tier** Choose between Standard, Premium, or Trial. For more information on these tiers, see Databricks pricing page.

CREATE A SPARK CLUSTER IN DATABRICKS

An all-purpose cluster that can be shared by multiple users. These are typically used to run notebooks. All-Purpose clusters remain active until terminated by users.

In the Azure portal, Launch from newly created Databrick workspace and create new cluster.



The screenshot shows the 'Create Cluster' page in the Microsoft Azure Databricks portal. The left sidebar contains navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main content area is titled 'Create Cluster' and 'New Cluster'. It includes a 'Cluster Name' field with the value 'mysparkcluster'. The 'Cluster Mode' is set to 'Standard'. The 'Pool' is set to 'None'. The 'Databricks Runtime Version' is 'Runtime: 6.4 (Scala 2.11, Spark 2.4.5)'. A green banner indicates 'Now This Runtime version supports only Python 3.' The 'Autopilot Options' section has 'Enable autoscaling' checked and 'Terminate after' set to '120 minutes of inactivity'. The 'Worker Type' is 'Standard_DS3_v2' with '14.0 GB Memory, 4 Cores, 0.75 DBU'. The 'Min Workers' is '2' and 'Max Workers' is '8'. The 'Driver Type' is 'Same as worker' with '14.0 GB Memory, 4 Cores, 0.75 DBU'. A 'Create Cluster' button is visible, along with a summary of resources: '2-8 Workers: 28.0-112.0 GB Memory, 8-32 Cores, 1.5-6 DBU' and '1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU'. An 'Advanced Options' link is at the bottom.

GETTING DATA

For scripting work outside Databricks python is used. There are some really cool data visualization libraries that are available in Python. A pre-loaded datasets will be used here that come with Azure Databricks.

Notebook in Databricks need to be created, configure to read data from an Azure Open Datasets, and run a Spark SQL job on the data. Select **Python** as the language

The screenshot shows the 'Create Notebook' dialog box. It has three fields: 'Name' with the value 'mynotebook', 'Language' set to 'Python', and 'Cluster' set to 'mysparkcluster (90 GB, Running)'. There are 'Cancel' and 'Create' buttons at the bottom right.

Firstly, need to load dataset, which displays the following table in our Databricks notebook. whenever a display() function run in Databricks, it givest a limit of 1000 rows in dataset.

1 display(diamonds)

» (1) Spark Jobs

_c0	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58	334	4.2	4.23	2.63
5	0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48
7	0.24	Very Good	I	VVS1	62.3	57	336	3.95	3.98	2.47
8	0.26	Very Good	H	SI1	61.9	55	337	4.07	4.11	2.53
9	0.22	Fair	E	VS2	65.1	61	337	3.87	3.78	2.49

Showing the first 1000 rows.

Table icons: Grid, Bar chart, Pie chart, Download

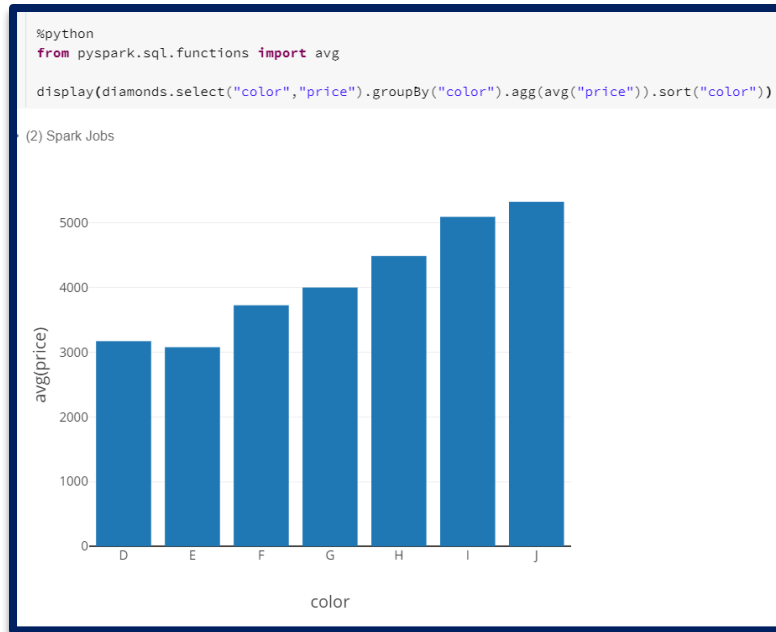
Visualizing Data in Databricks

Let's start off by grouping diamonds by color and showing their average price. Create a new data frame, now importing average function from `pyspark.sql.functions` and Selects color and price columns, averages the price, and groups and orders by color.

```
# Group by color
diamonds_color = diamonds.groupBy('color').avg("price")
display(diamonds_color)
```

It give the tabular output from the bottom of the tabular output, select the Bar chart icon, and then click Plot Options.

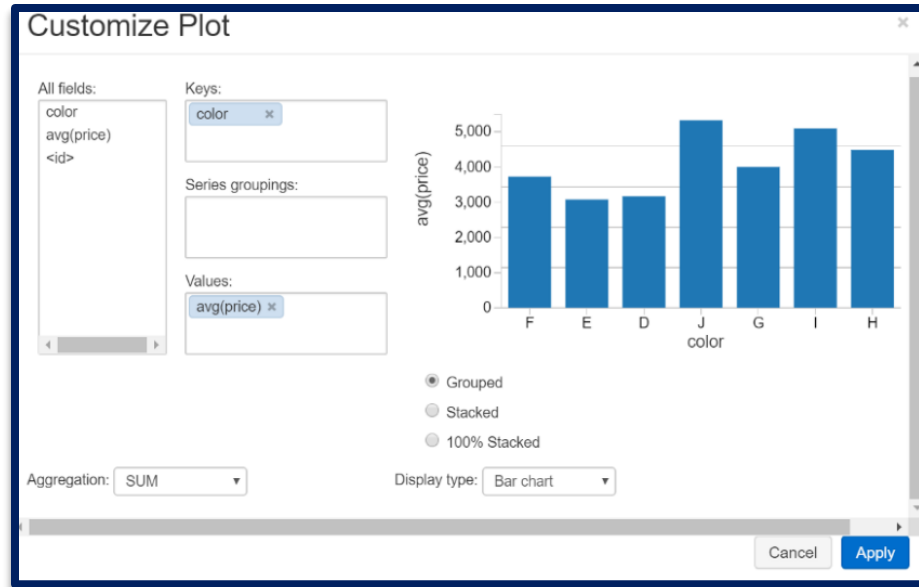




In Azure Databricks, different types of visualizations can be created



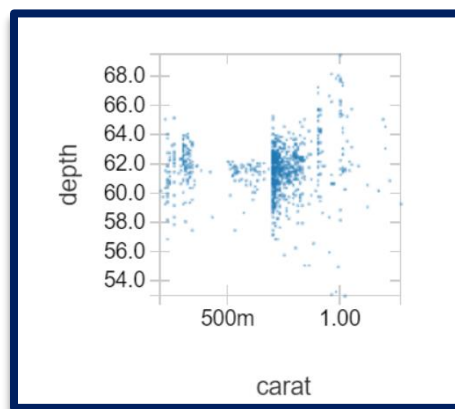
Customization of plots can also be done using Plot Options



This is a pretty basic example, but using this features can be customize what fields need to use in chart, the keys, values, groups, type of aggregation and how chart is displayed.

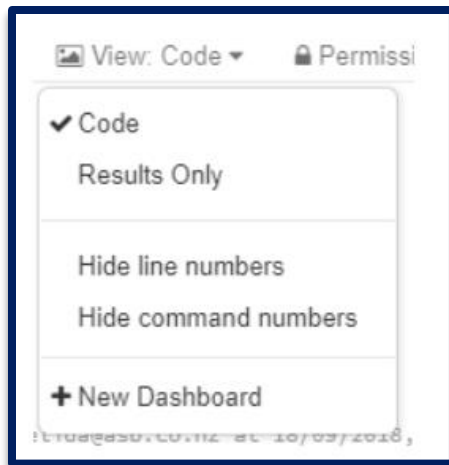
In data frame, to see if there is a relationship between the depth of a diamond and its carat value. Let's create a scatter plot to see if there is:

```
# depth to carat
depthVcarat = diamonds.select("depth", "carat")
display(depthVcarat)
```

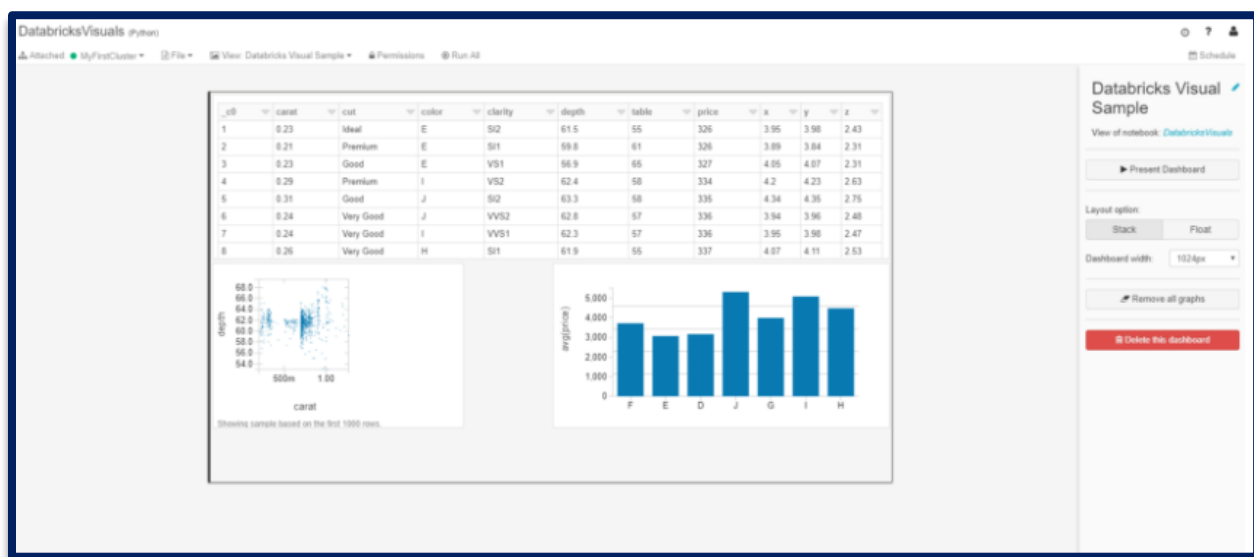


Dashboard in Azure Databricks

Visuals can consolidate to create a neat dashboard. To do this, use the drop down menu in notebook where it says view: Code and click New Dashboard:



Here visuals can be moved to create dashboards. Here visuals can move around to fit dashboard. The controls are pretty simple, choose a layout option (either stacked or floated) and a dashboard width. Dashboards can either be really simple in Databricks or they can be more sophisticated.



While the visualization tools in Databricks are good, they aren't as comprehensive as Power BI.

DATA STREAMING IN DATABRICKS

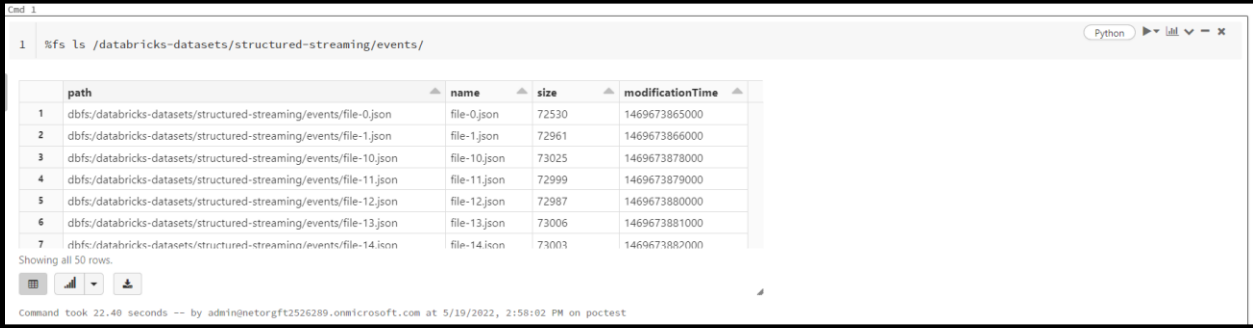
Sensors, IoT devices, social networks, and online transactions all generate data that must be constantly monitored and acted on. As a result, there is a greater need than ever for large-scale, real-time stream processing. A data stream is treated as a table that is continuously appended in Structured Streaming. As a result, a stream processing model that is very similar to a batch processing model is created. Streaming computation is expressed as a standard batch-like query on a static table, but Spark executes it as an incremental query on the unbounded input table.

This is done in four steps:

- Load sample data
- Initialize a stream
- Start a stream job
- Query a stream

Load Sample Data

Azure Databricks has sample event data as files in `/databricks-datasets/structured-streaming/events/` to use to build a Structured Streaming application. Let's take a look at the contents of this directory.



```
1 %fs ls /databricks-datasets/structured-streaming/events/
```

	path	name	size	modificationTime
1	dbfs:/databricks-datasets/structured-streaming/events/file-0.json	file-0.json	72530	1469673865000
2	dbfs:/databricks-datasets/structured-streaming/events/file-1.json	file-1.json	72961	1469673866000
3	dbfs:/databricks-datasets/structured-streaming/events/file-10.json	file-10.json	73025	1469673878000
4	dbfs:/databricks-datasets/structured-streaming/events/file-11.json	file-11.json	72999	1469673879000
5	dbfs:/databricks-datasets/structured-streaming/events/file-12.json	file-12.json	72987	1469673880000
6	dbfs:/databricks-datasets/structured-streaming/events/file-13.json	file-13.json	73006	1469673881000
7	dbfs:/databricks-datasets/structured-streaming/events/file-14.json	file-14.json	73003	1469673882000

Showing all 50 rows.

Command took 22.49 seconds -- by admin@netorgft2526289.onmicrosoft.com at 5/19/2022, 2:58:02 PM on poc-test

Initialize the Stream

Since the sample data is just a static set of files, you can emulate a stream from them by reading one file at a time, in the chronological order in which they were created.



```
streamingInputDF: pyspark.sql.dataframe.DataFrame
  time: timestamp
  action: string
streamingCountsDF: pyspark.sql.dataframe.DataFrame
  action: string
  window: struct
    start: timestamp
    end: timestamp
  count: long
```

Command took 3.68 seconds -- by admin@netorgft2526289.onmicrosoft.com at 5/19/2022, 3:17:09 PM on poc-test

Start the Streaming Job

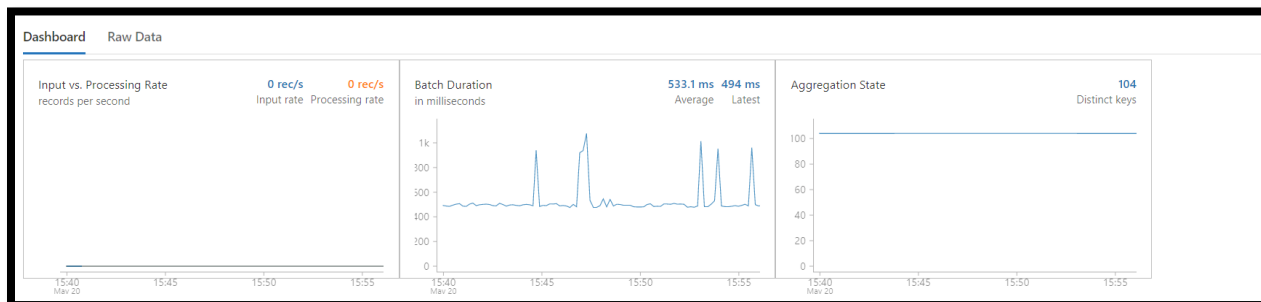
Start a streaming computation by defining a sink and starting it. In our case, to query the counts interactively, set the `complete` set of 1 hour counts to be in an in-memory table.

query is a handle to the streaming query named counts that is running in the background. This query continuously picks up files and updates the windowed counts.

The command window reports the status of the stream:



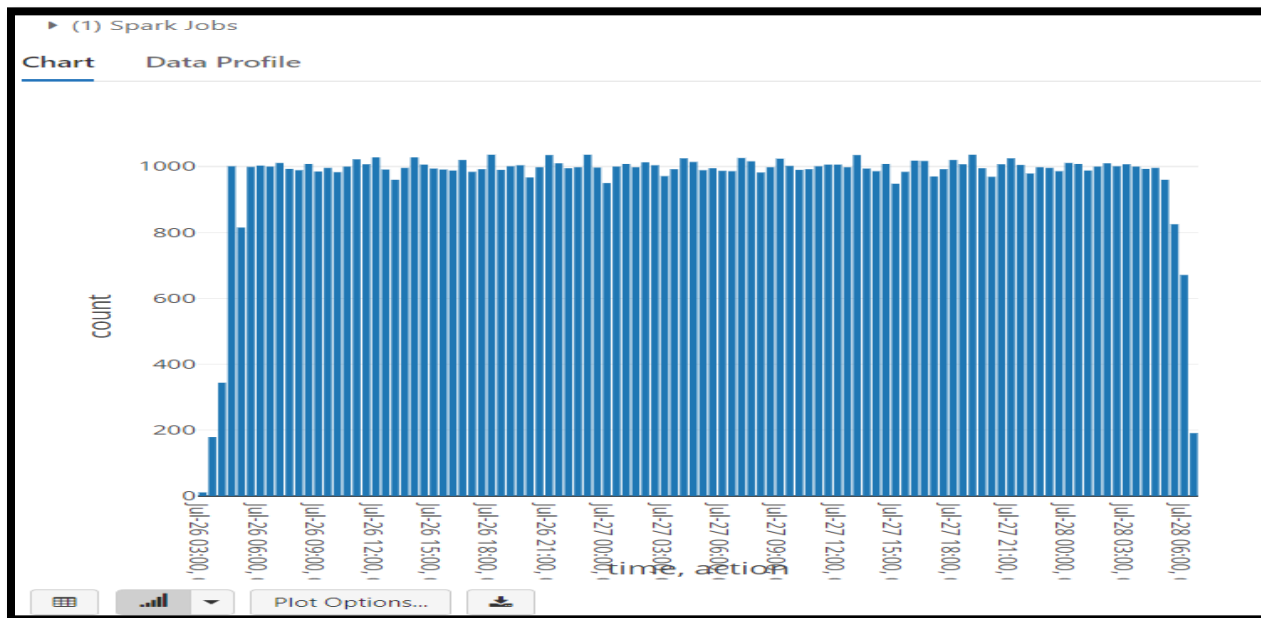
When expanding counts, get a dashboard of the number of records processed, batch statistics, and the state of the aggregation:

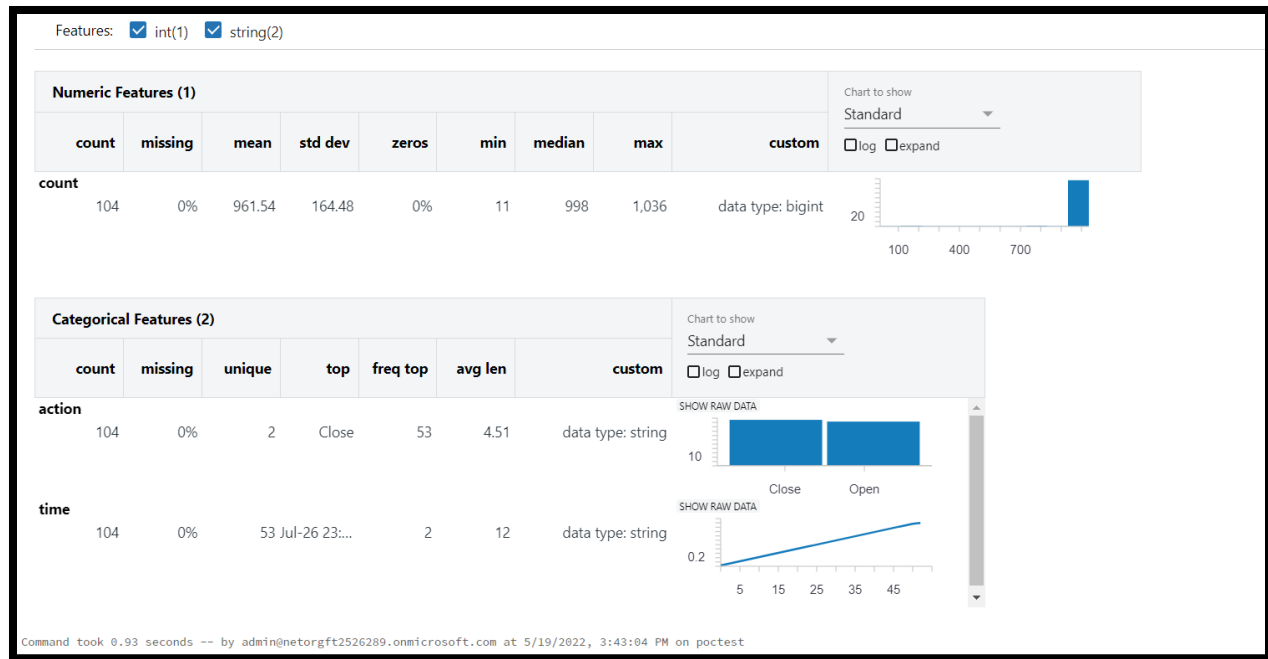


Interactively Query the Stream

We can periodically query the counts aggregation:

As it can see from this series of screenshots, the query changes every time you execute it to reflect the action count based on the input stream of data.





INTEGRATING AZURE DATABRICK DATA WITH POWER BI

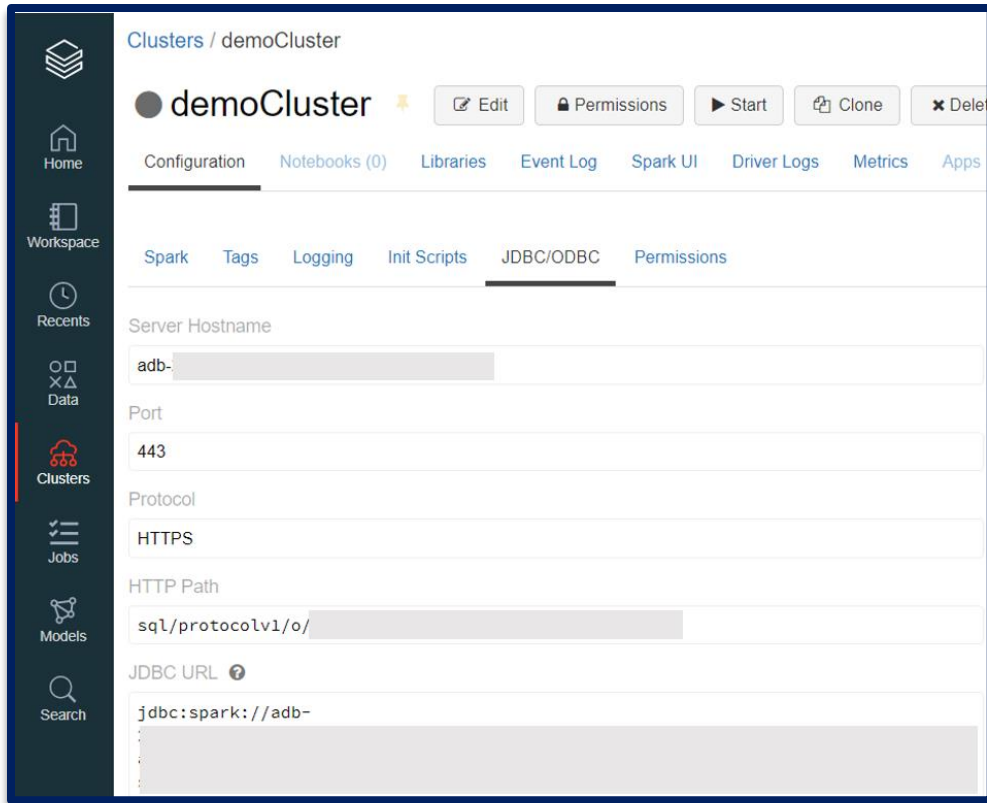
Power BI

Microsoft Power BI is a business analytics service that provides interactive visualizations with self-service business intelligence capabilities, enabling end users to create reports and dashboards by themselves without having to depend on information technology staff or database administrators. When Azure Databricks are used as a data source with Power BI, advantages of Azure Databricks can add to the performance and technology beyond data scientists and data engineers to all business users. Power BI Desktop can be connected to Azure Databricks clusters using the built-in Azure Databricks connector. Also publish Power BI reports to the Power BI service and enable users to access the underlying Azure Databricks data using SSO, passing along the same Azure AD credentials they use to access the report.

Microsoft Power BI is becoming more and more popular recently as a Data Analytics tool. Also, it is ubiquitous for a company to have a whole bucket of Microsoft products that include Azure.

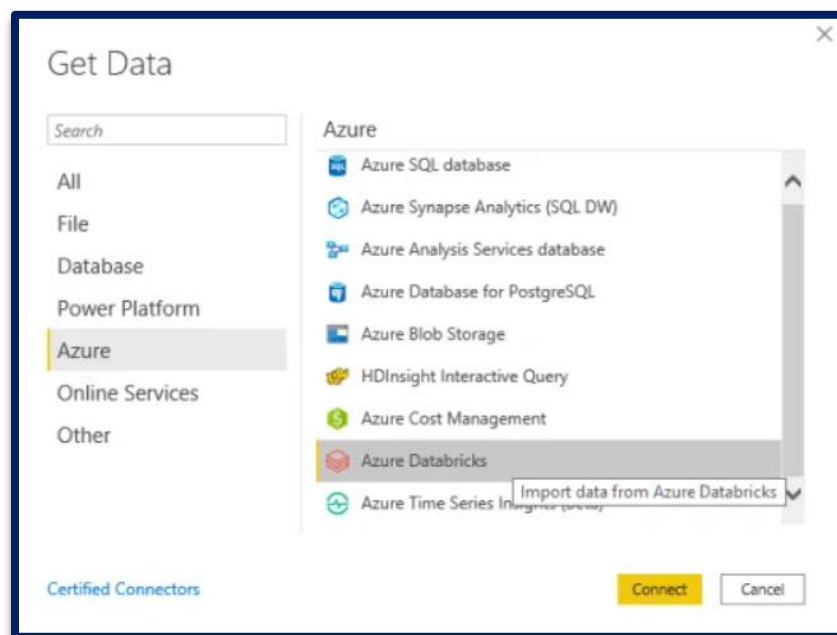
Connection elements

From Azure cluster advance option open JDBC/ODBC tab and use server hostname, HTTP path, JDBC url for connection with Power BI



Integrating with Power BI

There is a built-in data connector in Power BI that connects Azure Databricks data to Power BI.



Get Azure Databricks connection information the server hostname, port, and HTTP path.

- In Power BI Desktop, go to Get Data > Azure and select the Azure Databricks connector.
- Paste the Server Hostname and HTTP Path
- Azure Active Directory: Use your Azure account credentials. Click the Sign in button. In the sign-in dialog, enter your Azure account username
- Select the Azure Databricks data to query from the Power BI Navigator.

Azure Databricks

Server Hostname ①

HTTP Path ②

Example: `sql/protocolv1/o/1814582234607533/7508-187377-agent704`

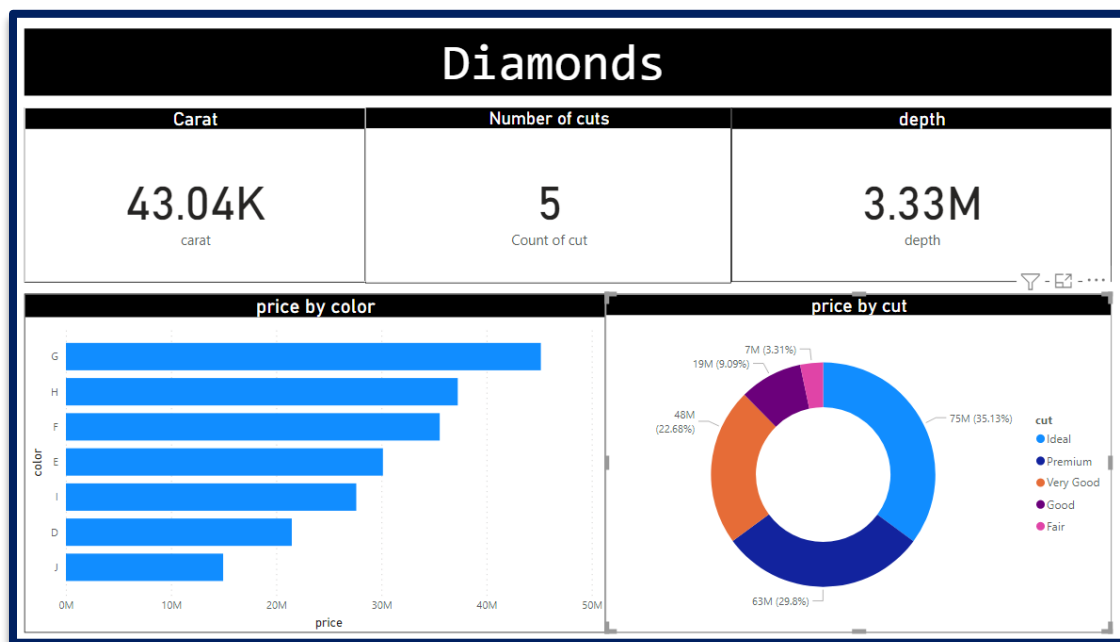
Data Connectivity mode ③

☒ Import

☐ DirectQuery

OK Cancel

Data created in cluster notebook can be used to create dashboard in Power BI



Before built-in databrick data connector in Power BI Spark was used to connect Azure Databrick data to Power BI here is a brief comparison between both

Feature Comparison	Spark Connector	Databricks Connector
--------------------	-----------------	----------------------

Power BI Desktop	✓	✓
Power BI Service	✓	✓
Direct Query (Desktop)	✓	✓
Direct Query (Service)	✓	✓
Import Mode	✓	✓
Manual Refresh (Service)	✓	✓
Scheduled Refresh (Service)	✓	✓
Azure Active Directory (AAD) Authentication	✗	✓
Personal Access Token Authentication	✓	✓
Username/Password Authentication	✓	✓
General Available	✓	✓
Performance Improvements with Spark 3.x	✗	✓
Supports On-Premises data gateway	✓	✗

REASON TO USE AZURE DATABRICKS

Familiar languages and environment

While Azure Databricks is Spark based, it allows commonly used programming languages like Python, R, and SQL to be used. These languages are converted in the backend through APIs, to interact with Spark. This saves users having to learn another programming language, such as Scala, for the sole purpose of distributed analytics.

Higher productivity and collaboration

- *Production Deployments*

Deploying work from Notebooks into production can be done almost instantly by just tweaking the data sources and output directories.

- *Workspaces*

Databricks creates an environment that provides workspaces for collaboration (between data scientists, engineers, and business analysts), deploys production jobs (including the use of a scheduler), and has an optimized Databricks engine for running. These interactive workspaces allow multiple members to collaborate for data model creation, machine learning, and data extraction.

- *Version Control*

Version control is automatically built in, with very frequent changes by all users saved. Troubleshooting and monitoring is a painless task on Azure Databricks.



Integrates easily with the whole Microsoft stack

Azure Databricks uses the Azure Active Directory (AAD) security framework. Existing credentials authorization can be utilized, with the corresponding security settings. Access and identity control are all done through the same environment. Using AAD allows easy integration with the entire Azure stack including Data Lake Storage (as a data source or an output), Data Warehouse, Blob storage, and Azure Event Hub.

Extensive list of data sources

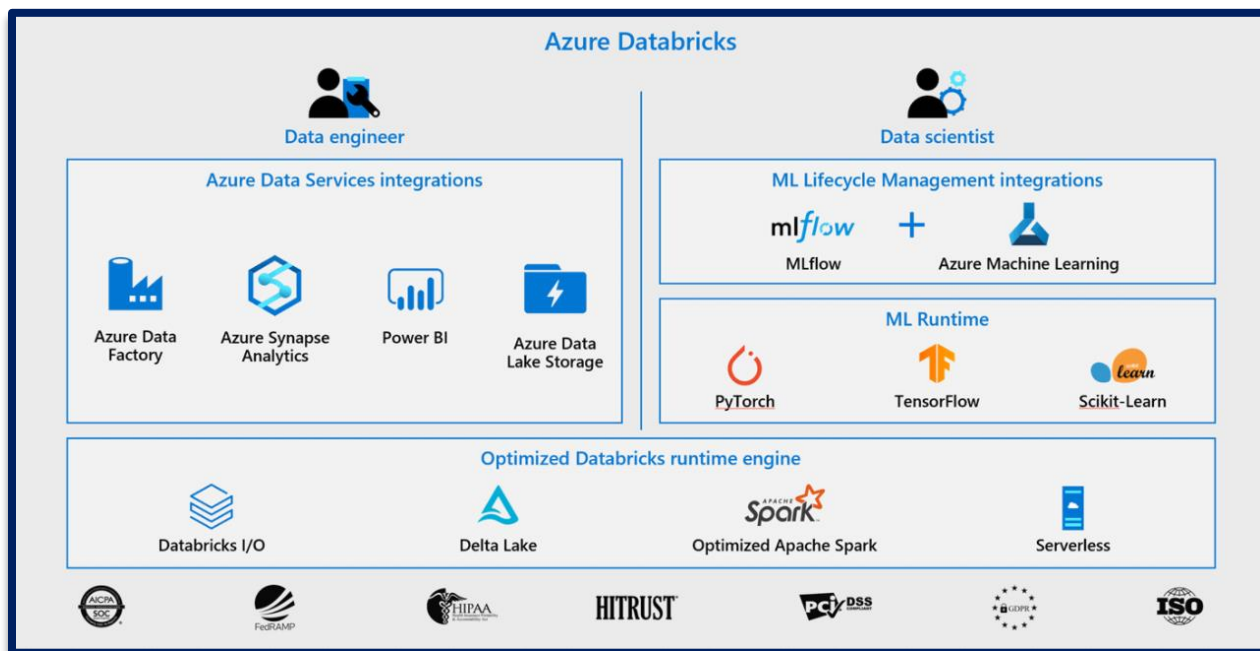
Aside from those Azure-based sources mentioned, Databricks easily connects to sources including on premise SQL servers, CSVs, and JSONs. Other data sources include MongoDB, Avro files, and Couchbase.

Suitable for small jobs too

While Azure Databricks is ideal for massive jobs, it can also be used for smaller scale jobs and development/ testing work. This allows Databricks to be used as a one-stop shop for all analytics work. We no longer need to create separate environments or VMs for development work.

Extensive support available

Extensive support are available for all aspects of Databricks, including the programming languages needed. Azure Databricks is powerful and cheap. As the current digital revolution continues, using big data technologies will become a necessity for many organizations. Azure Databricks is extremely flexible and easy to get started on, making distributed analytics much easier to use.



REAL TIME USE CASE

- As mobile apps and other advances in technology continue to upgrade the way users choose and utilize information, recommendation engines are becoming an essential part of applications and software products.
- Churn analysis also known as customer defection, customer attrition, or customer turnover, is the loss of clients or customers. Forecasting and restricting customer churn are vital to a range of businesses.
- Intrusion detection is required to track network or system activities for malicious activities or policy violations and generate electronic reports to a management station.

AZURE DATABRICKS PRICING

- Pay as you go: Azure Databricks cost you for virtual machines (VMs) manage in clusters and Databricks Units (DBUs) depend on the VM instance selected.
- A DBU is a unit of the processing facility, billed on per-second usage, and DBU consumption depends on the type and size of the instance running Databricks.

Workload	DBU prices – standard tier	DBU prices – premium tier
All- Purpose Compute	\$0.40/DBU-hour	\$0.55/DBU-hour
Job Compute	\$0.15/DBU-hour	\$0.30/DBU-hour
Job Light Compute	\$0.07/DBU-hour	\$0.22/DBU-hour

CONCLUSION

Azure Databricks is a cloud analytics platform that can meet the needs to both data engineers and data scientists to build a full end-to-end big data solution and deploy it in production. It can be used by data engineers to set up the whole architecture by setting up clusters, scheduling and running jobs, connections to data sources etc. and by data scientists to perform machine learning and real-time analytics. Business users can also use the data transformed in Azure Databricks directly in Power BI or other analytics tool for reporting needs just by connecting the cluster to the analytics tool. In this paper Azure Databricks, its solution architecture and why to use azure Databricks are discussed. In this paper creating azure workspace, creating spark cluster in Databricks and its integration with Power BI are also discussed.

CONTACT US

Shahzad Sarwar

Entrepreneur/Architect/Consultant

Cognitive Convergence

<http://www.cognitiveconvergence.com>



Voice: +1 4242530744

Skype: Shahzad.Sarwar.Online

shahzad@cognitiveconvergence.com

