Azure Databrics

Agenda

What is Azure Databrics?

Create Workspace and Cluster

Working with Notebooks and Jobs

Libraries Overview

Administration, Manage Users & Groups

What is Azure Databricks?

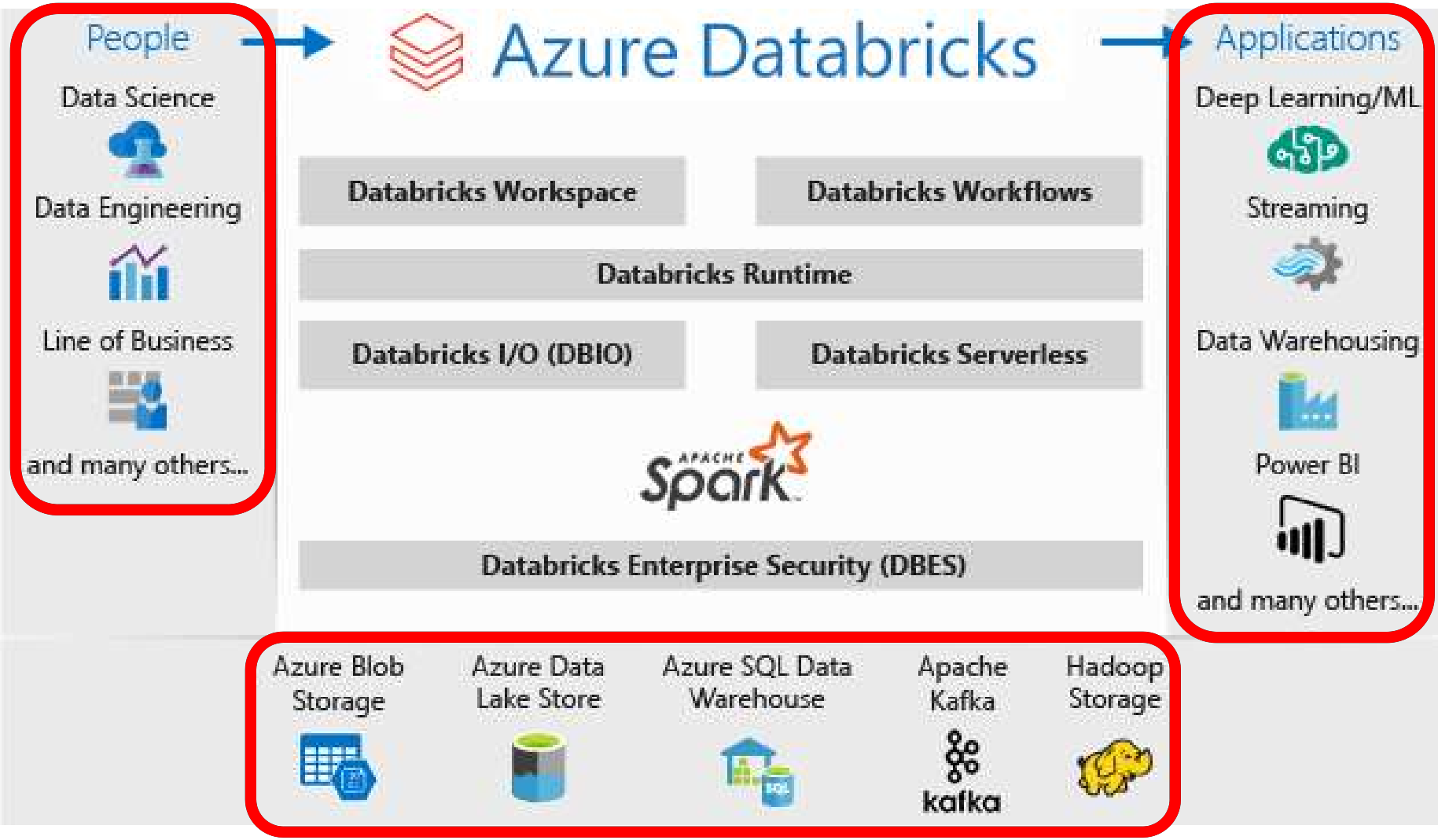
Apache Spark-based

Analytics platform

Provides

- One-click setup
- Streamlined workflows and
- An interactive workspace
- Enables collaboration between data scientists, data engineers, and business analysts.

Azure Databricks



Azure Databricks

For a big data pipeline, the data is ingested into Azure

This data lands in

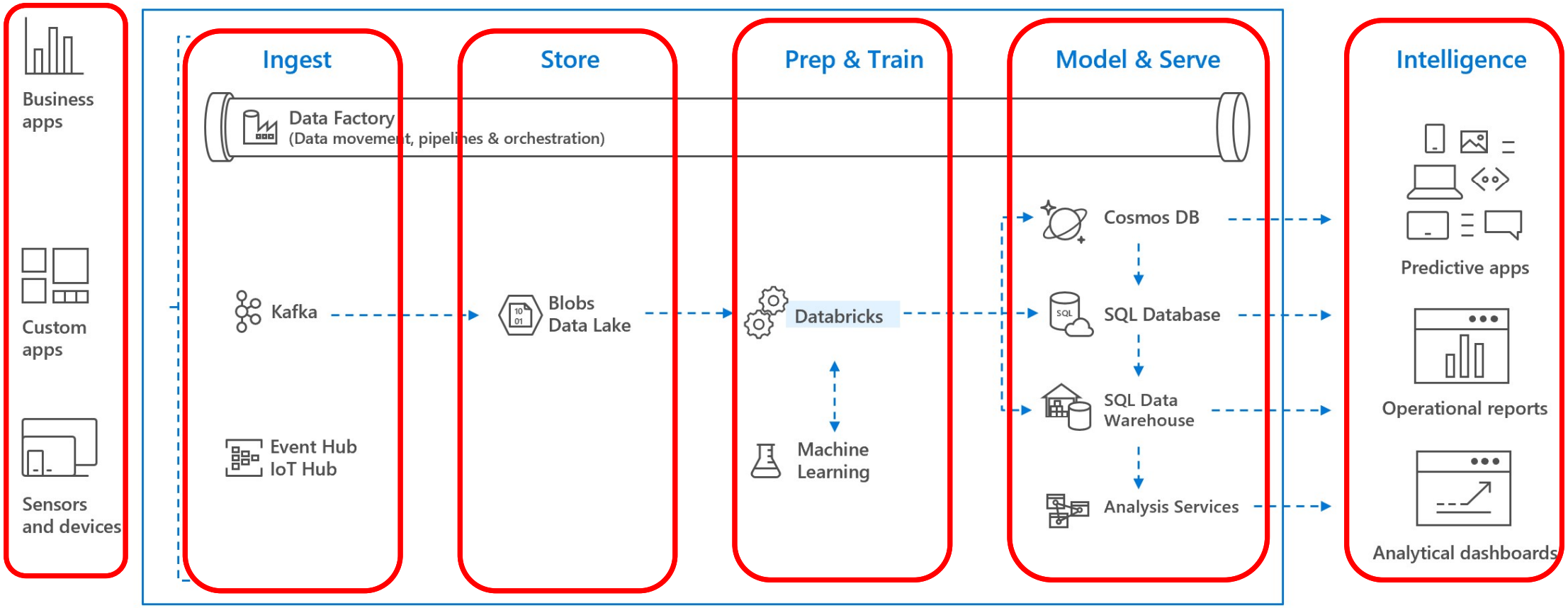
- Azure Blob Storage or
- Azure Data Lake Storage

Use Azure Databricks to read data from multiple data sources

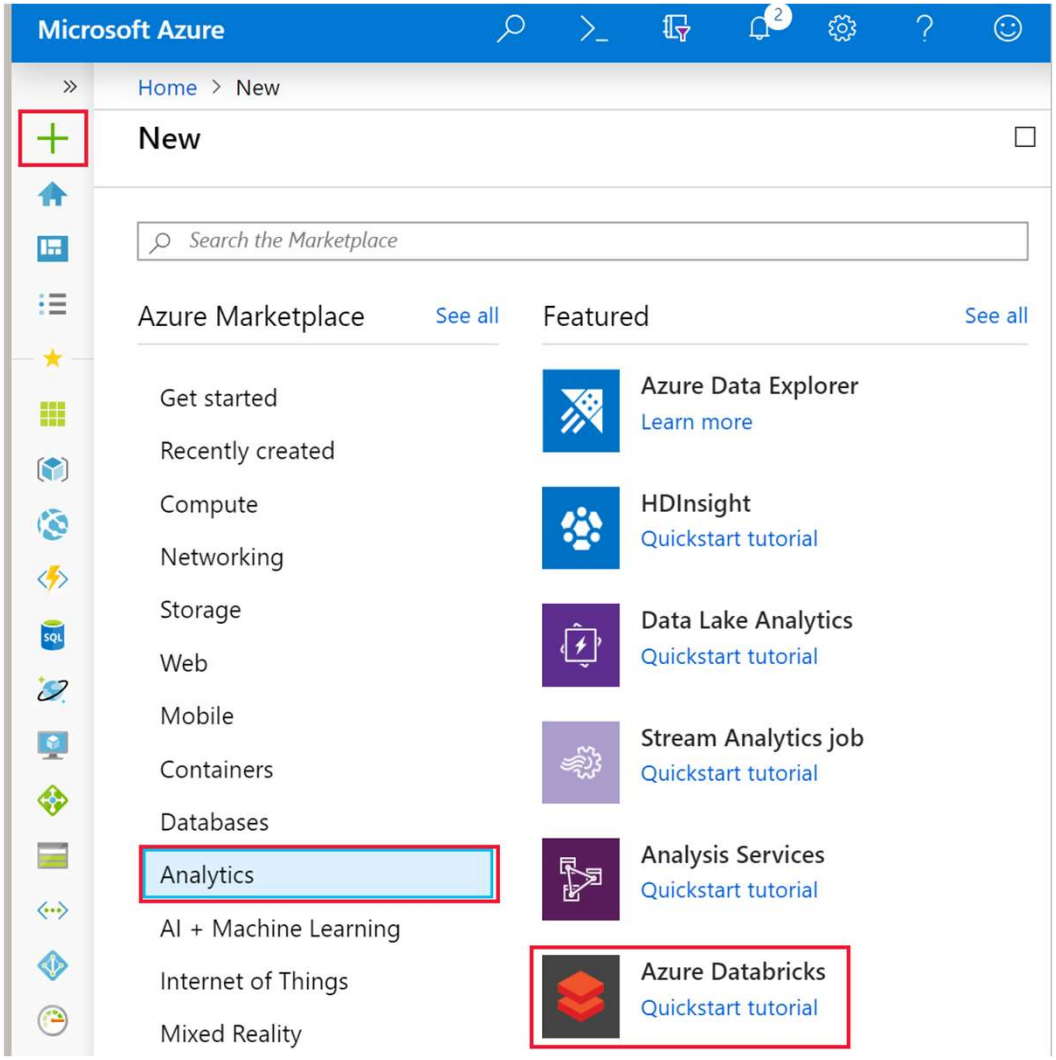
- Azure Blob Storage
- Azure Data Lake Storage
- Azure Cosmos DB, or
- Azure SQL Data Warehouse

Using Databricks, turn it into breakthrough insights

Azure Databricks



Hands-On: Create Databricks Workspace



Hands-On: Create Databricks Workspace

Basics * | Networking | 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 * ⓘ: <your subscription>

Resource group * ⓘ: (New) databricks-quickstart

Create new

Intance Details

Workspace name *: mydatabricksws

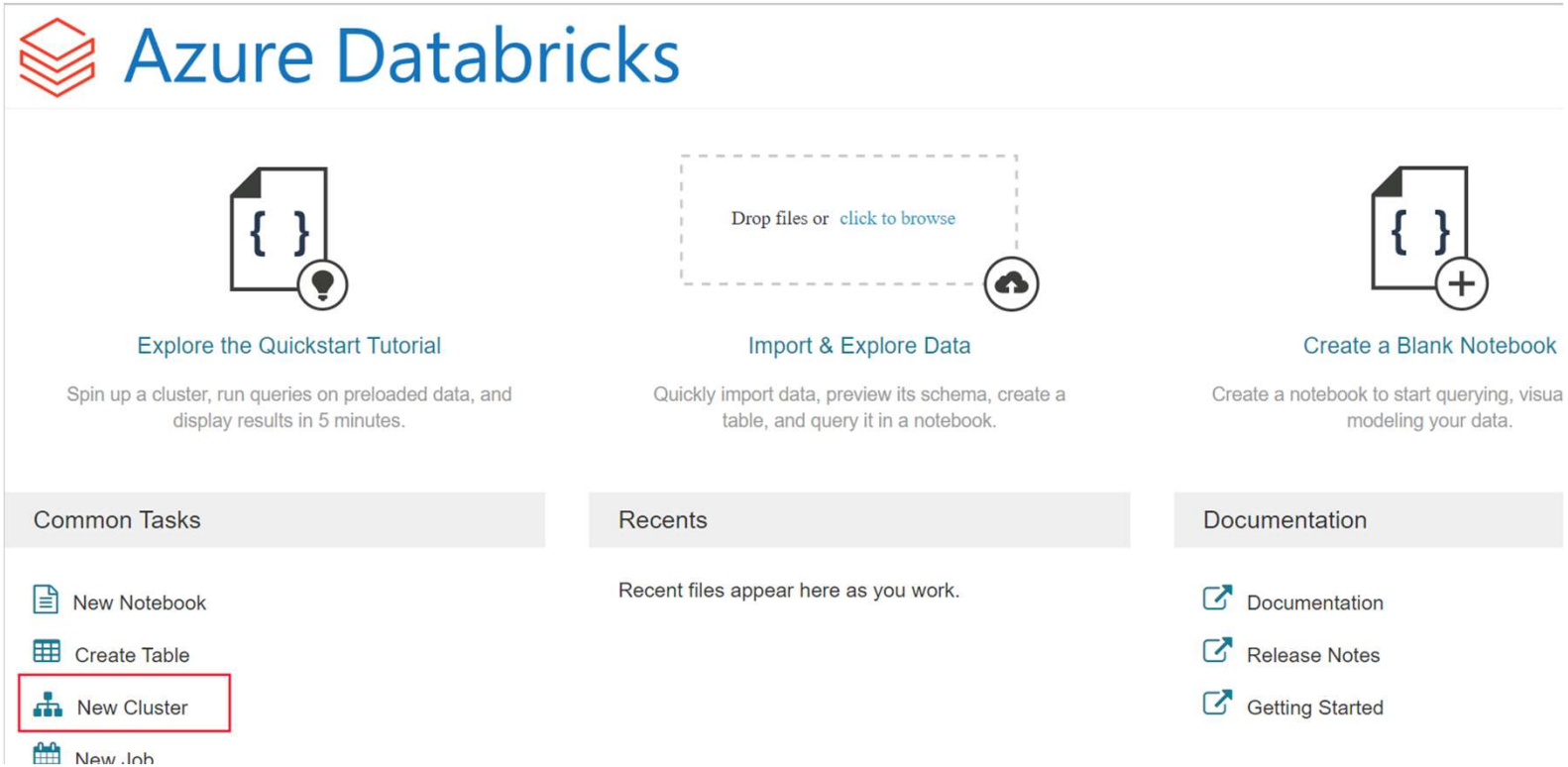
Location *: West US 2

Pricing Tier * ⓘ: Standard (Apache Spark, Secure with Azure AD)

- Standard (Apache Spark, Secure with Azure AD)
- Premium (+ Role-based access controls)
- Trial (Premium - 14-Days Free DBUs)

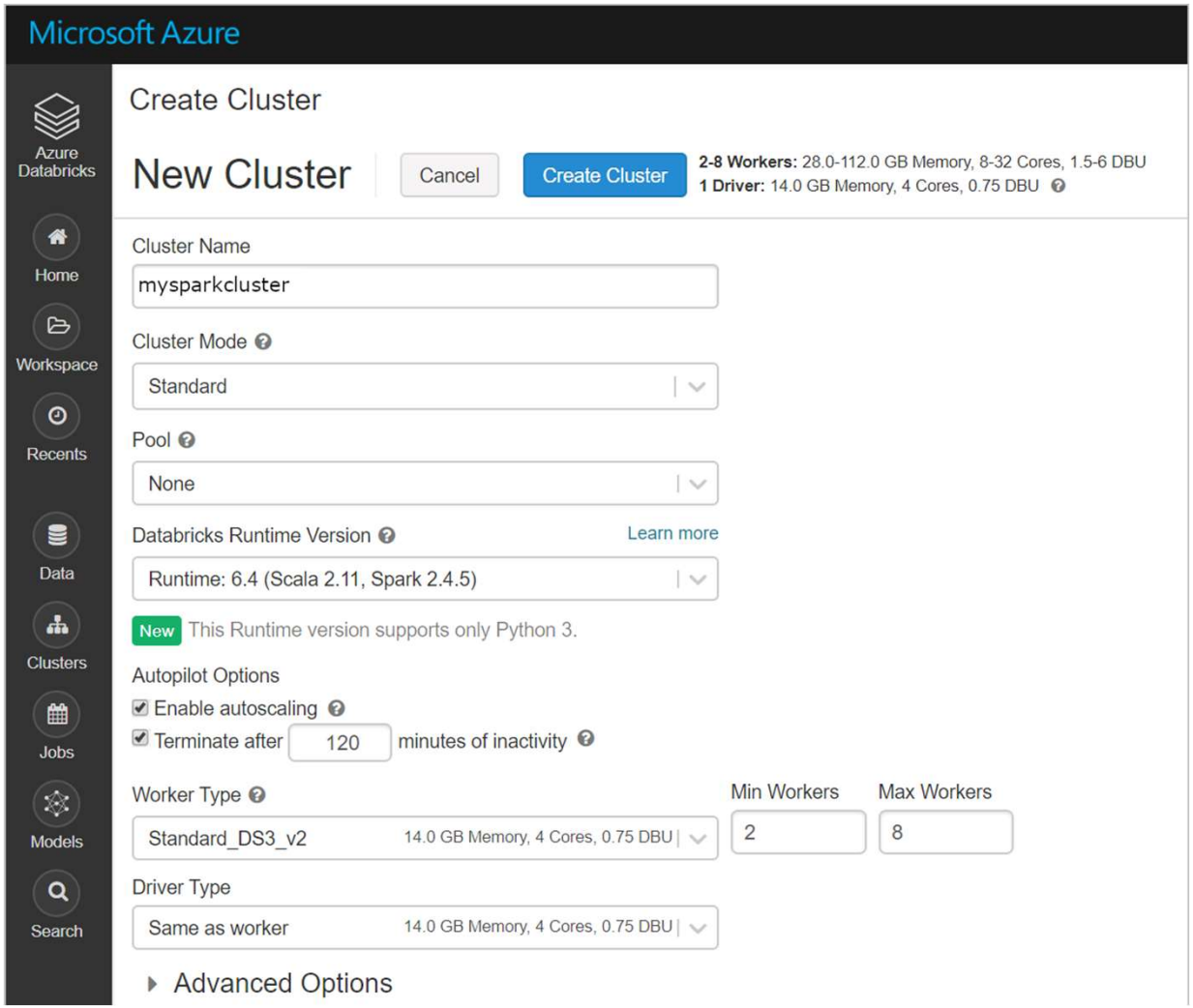
Hands-On: Create a Spark cluster in Databricks

- Go to the Databricks workspace that you created, and then click Launch Workspace.
- You are redirected to the Azure Databricks portal.
- Click New Cluster



Hands-On: Create a Spark cluster in Databricks

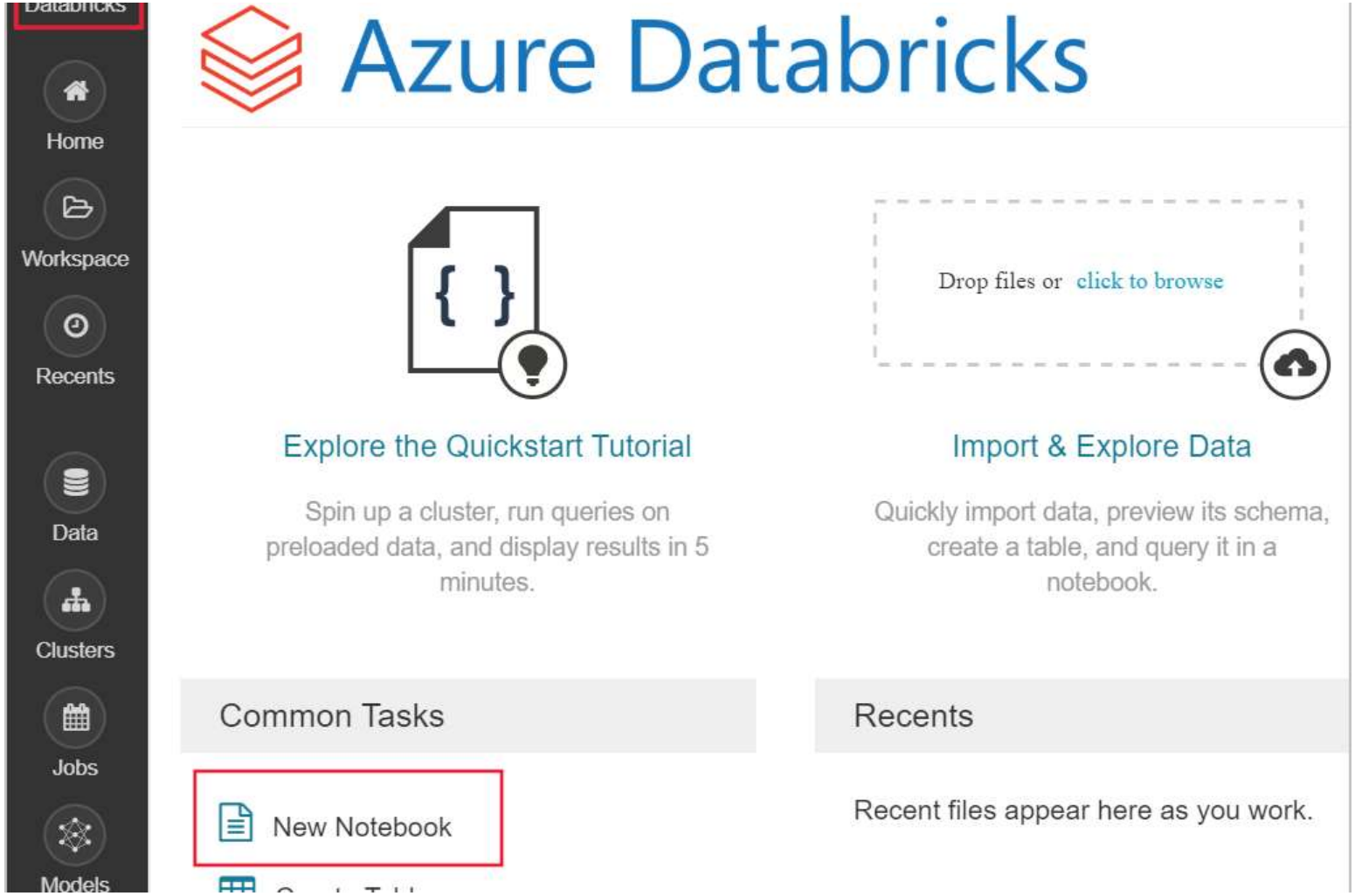
- Make sure you select the Terminate after __ minutes of inactivity checkbox
- Provide a duration (in minutes) to terminate the cluster, if the cluster is not being used.



Run a Spark SQL job

Hands-On: Run a Spark SQL job

Source Code:
atinNotebook1.ipynb



Hands-On: Run a Spark SQL job

- The following command sets the Azure storage access information.
  - blob_account_name = "azureopendatastorage"
  - blob_container_name = "citydatacontainer"
  - blob_relative_path = "Safety/Release/city=Boston"
  - blob_sas_token = r"?st=2019-02-26T02%3A34%3A32Z&se=2119-02-27T02%3A34%3A00Z&sp=rl&sv=2018-03-28&sr=c&sig=XlJVWA7fMXCSxCKqJm8psMOh0W4h7cSYO28coRqF2fs%3D"

Hands-On: Run a Spark SQL job

- The following command allows Spark to read from Blob storage remotely
  - wasbs_path = 'wasbs://%s@%s.blob.core.windows.net/%s' % (blob_container_name, blob_account_name, blob_relative_path)
  - spark.conf.set('fs.azure.sas.%s.%s.blob.core.windows.net' % (blob_container_name, blob_account_name), blob_sas_token)
  - print('Remote blob path: ' + wasbs_path)

Hands-On: Run a Spark SQL job

- The following command creates a DataFrame
  - df = spark.read.parquet(wasbs_path)
  - print('Register the DataFrame as a SQL temporary view: source')
  - df.createOrReplaceTempView('source')

Hands-On: Run a Spark SQL job

- Run a SQL statement return the top 10 rows of data
  - print('Displaying top 10 rows: ')
  - display(spark.sql('SELECT * FROM source LIMIT 10'))

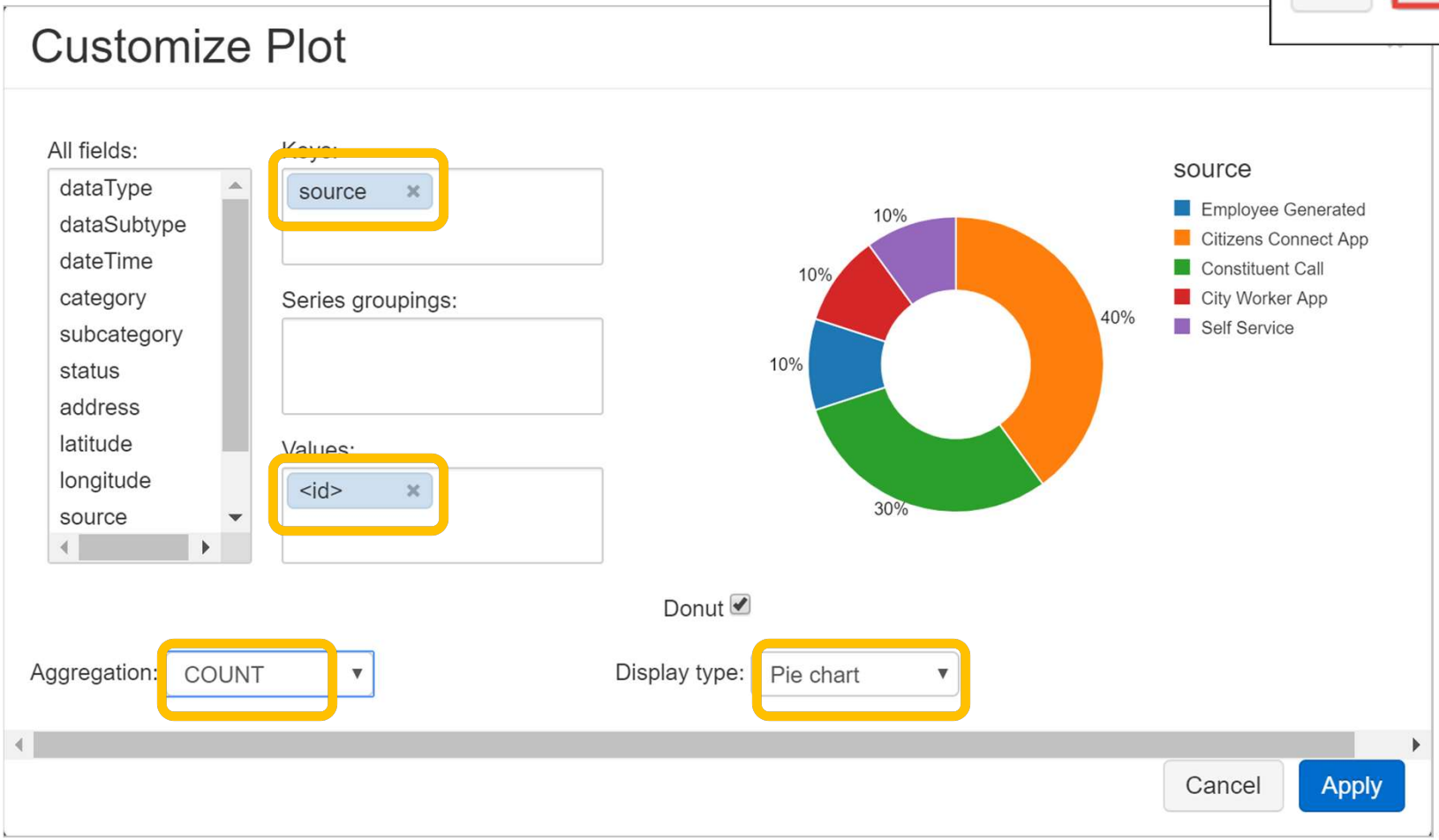
```
print('Displaying top 10 rows: ')
display(spark.sql('SELECT * FROM source LIMIT 10'))
```

(1) Spark Jobs

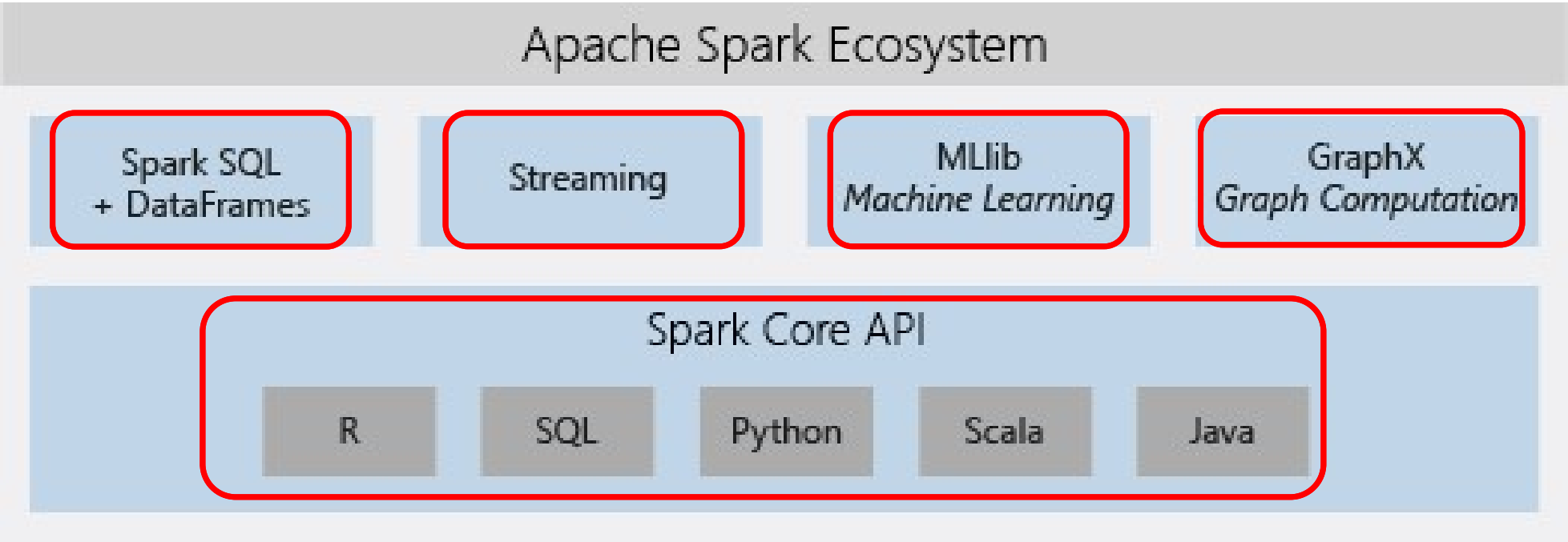
dataType	dataSubtype	dateTime	category	subcategory	status	address	latitude	longitude	source	extendedProperties
Safety	311_All	2011-08-11T11:02:16.000+0000	Recycling	Request for Recycling Cart	Closed	43 Howell St Dorchester MA 02125	42.3255	-71.0587	Employee Generated	null
Safety	311_All	2016-12-15T09:08:21.000+0000	Street Cleaning	Pick up Dead Animal	Closed	74 Aldie St Allston MA 02134	42.3588	-71.1335	Citizens Connect App	null
Safety	311_All	2017-01-26T18:45:00.000+0000	Enforcement & Abandoned Vehicles	Parking Enforcement	Closed	98 Waltham St Roxbury MA 02118	42.3436	-71.0713	Constituent Call	null

Hands-On: Run a Spark SQL job

- You now create a visual representation of this data



Apache Spark-based analytics platform



Azure Databricks concepts

Azure Databricks concepts

Workspace

- Environment for accessing all of your Azure Databricks assets.
- Organizes objects into folders

Objects

Notebooks

Libraries

Dashboards

Experiments

Notebook

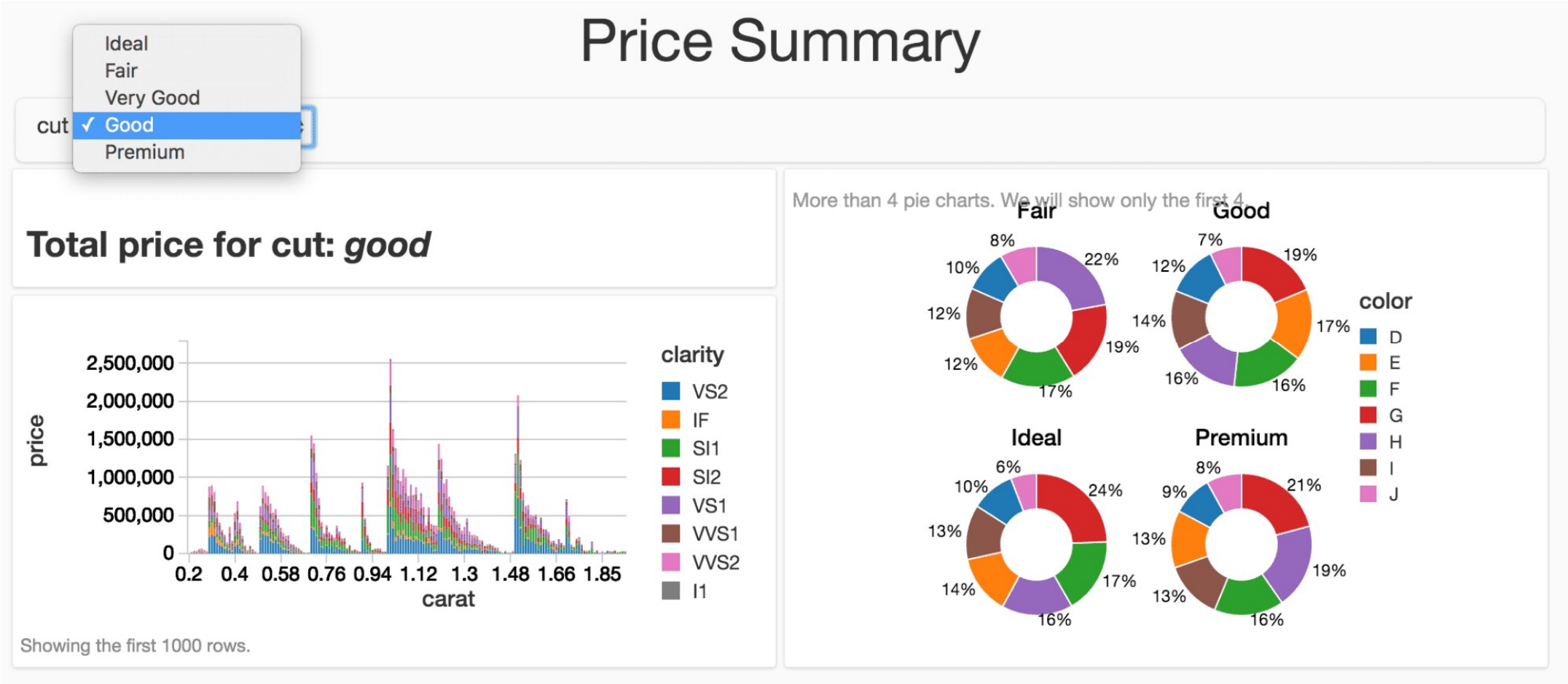
A web-based interface for documents

Document contain

- Runnable commands
- Visualizations, and
- Narrative text.

Dashboard

- Provides access to visualizations

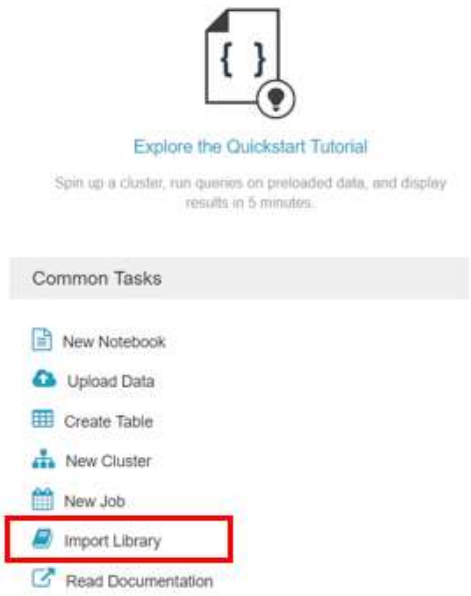


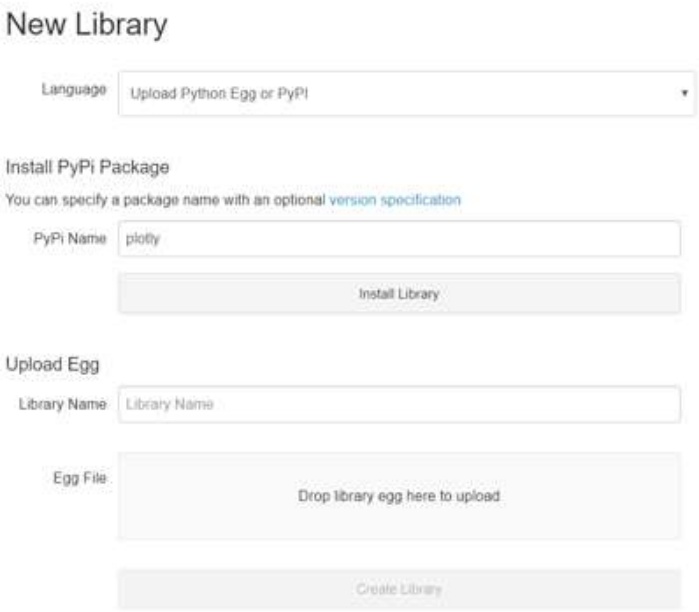
Library

A package of code available to the notebook

Databricks runtimes include many libraries

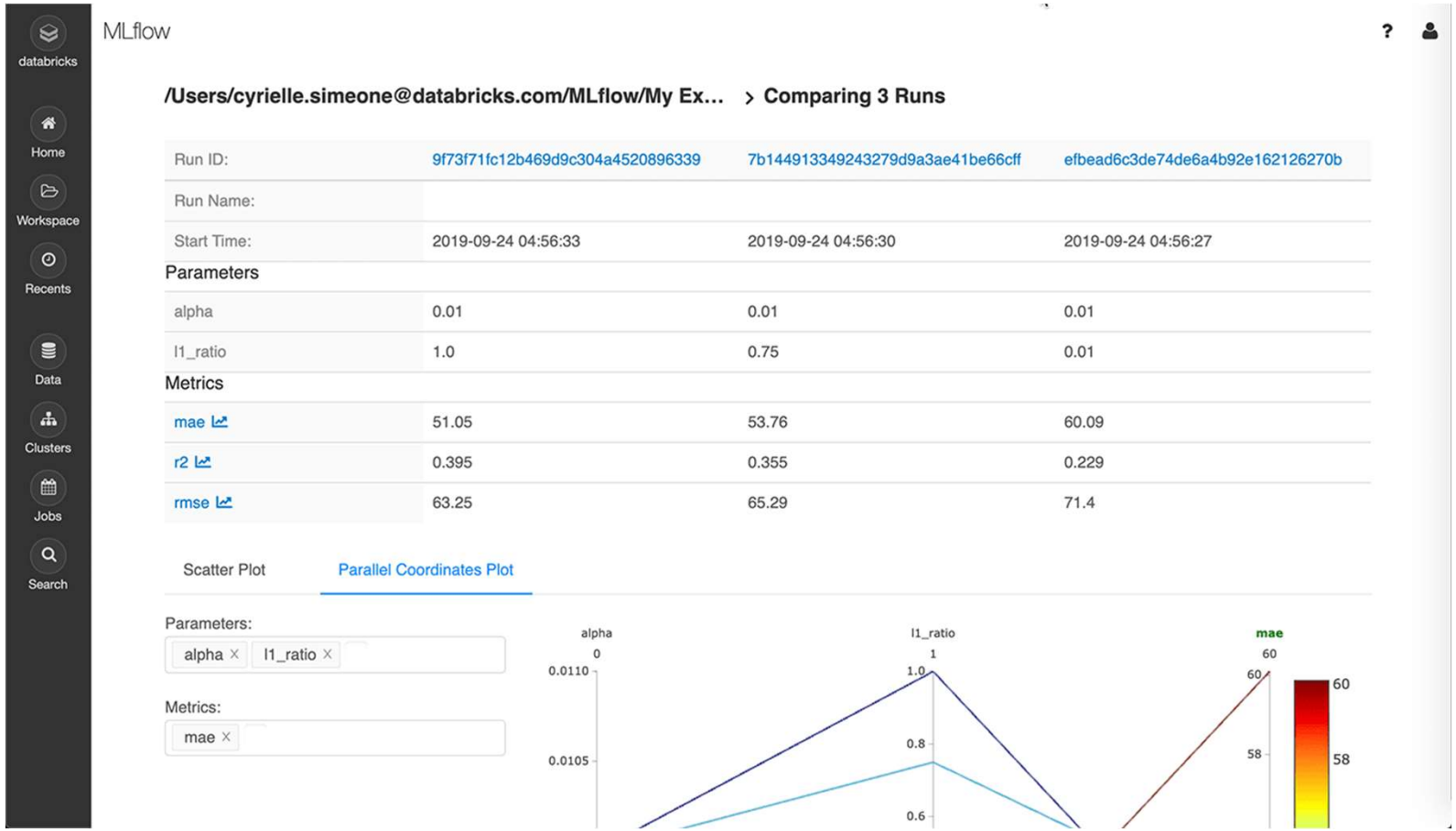
You can add your own.





Experiment

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



Authentication and authorization

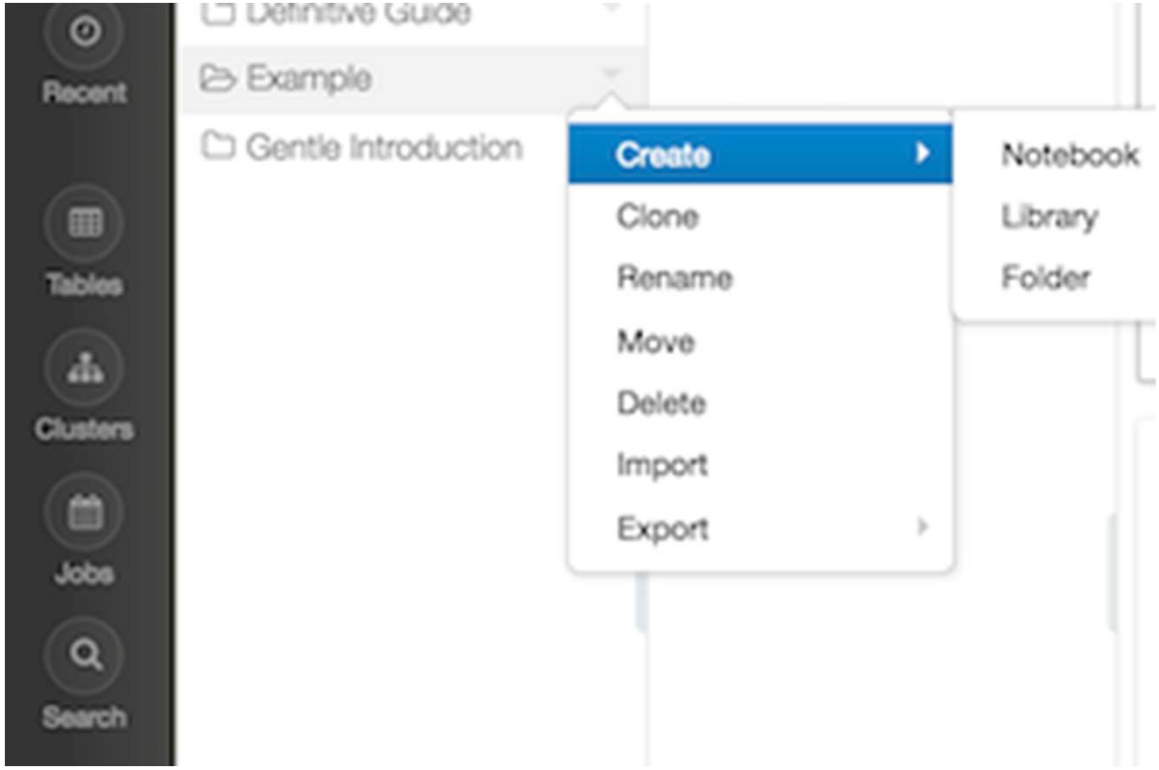
User	• A unique individual who has access to the system.
Group	• A collection of users.
Access control list (ACL)	• A list of permissions attached to the objects. • Specifies which users or system processes are granted access to the objects

Work with Notebooks

What is Notebook?

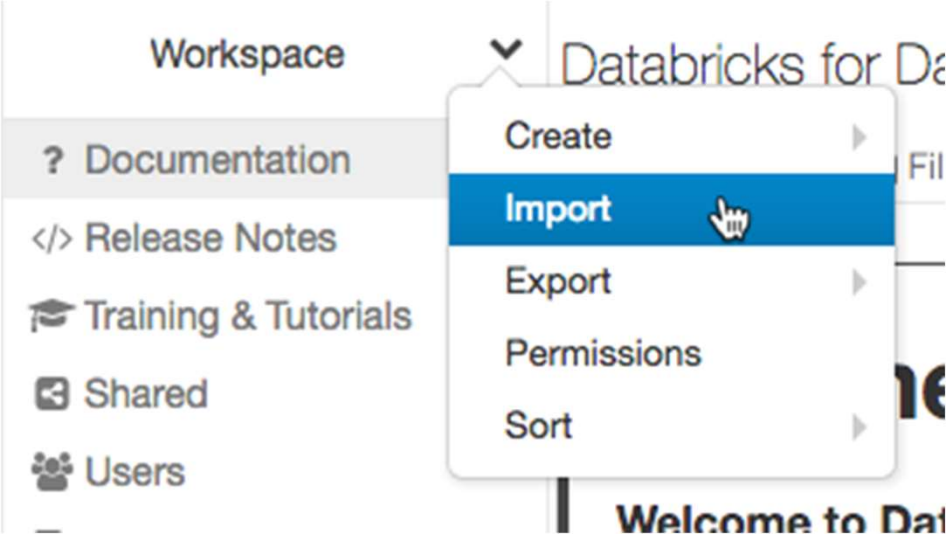
- A web-based interface to a document that contains
  - Runnable code
  - Visualizations, and
  - Narrative text

Hands-On: Create a notebook



Hands-On

- Open a Notebook
- Delete a Notebook
- Rename a notebook
- Import a notebook
- Export a notebook



Hands-On: Notebooks and clusters

- Before you can do any work in a notebook, you must first attach the notebook to a cluster
- Attach a notebook to a cluster
- Detach a notebook from a cluster
- View all notebooks attached to a cluster
- Schedule a notebook

Work with Jobs

What is a Job?

- A way of running a notebook on a scheduled basis
- Can create and run jobs using the
  - UI
  - CLI
  - By invoking the Jobs API

View jobs

- Click the Jobs icon Jobs Menu Icon in the sidebar

Jobs

+ Create Job

All | Owned by me | Accessible By Me

Filter

	Name	Job ID	Created By	Task	Cluster	Schedule	Last Run	Action
	Job A	5	test		8 Workers: Standard_DS3_v2 (beta) 3.2 (includes Apache Spark 2.2.0, Scala 2.	None		
	Job B	6	test		8 Workers: Standard_DS3_v2 (beta) 3.2 (includes Apache Spark 2.2.0, Scala 2.	None		
	Job C	7	test		8 Workers: Standard_DS3_v2 (beta) 3.2 (includes Apache Spark 2.2.0, Scala 2.	None		

Hands-On: Create a job

Hands-On: Run a job

- Schedule a job
- Pause and resume a job schedule
- Run a job immediately

Hands-On: View job run details

Library dependencies

- To get the full list of the driver library dependencies, run the following command inside a notebook
  - %sh
  - ls /databricks/jars

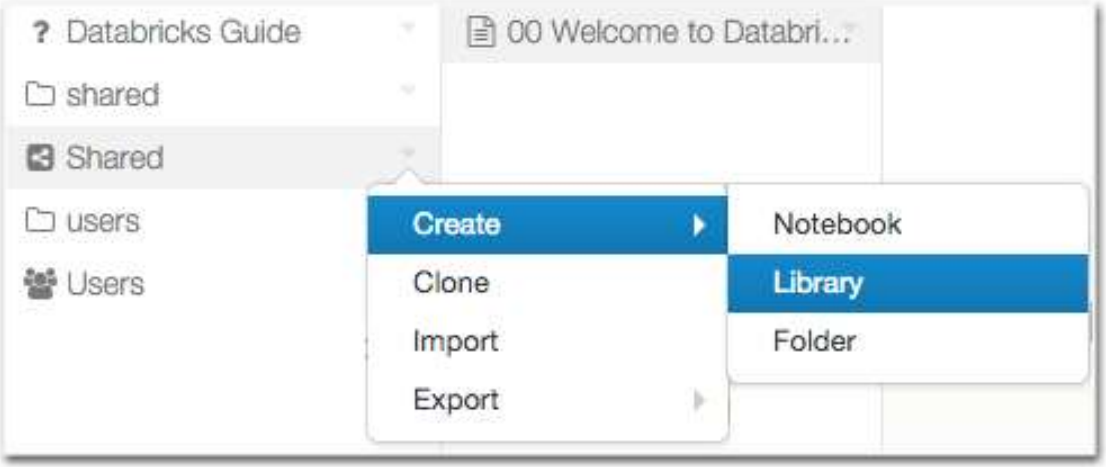
Libraries Overview

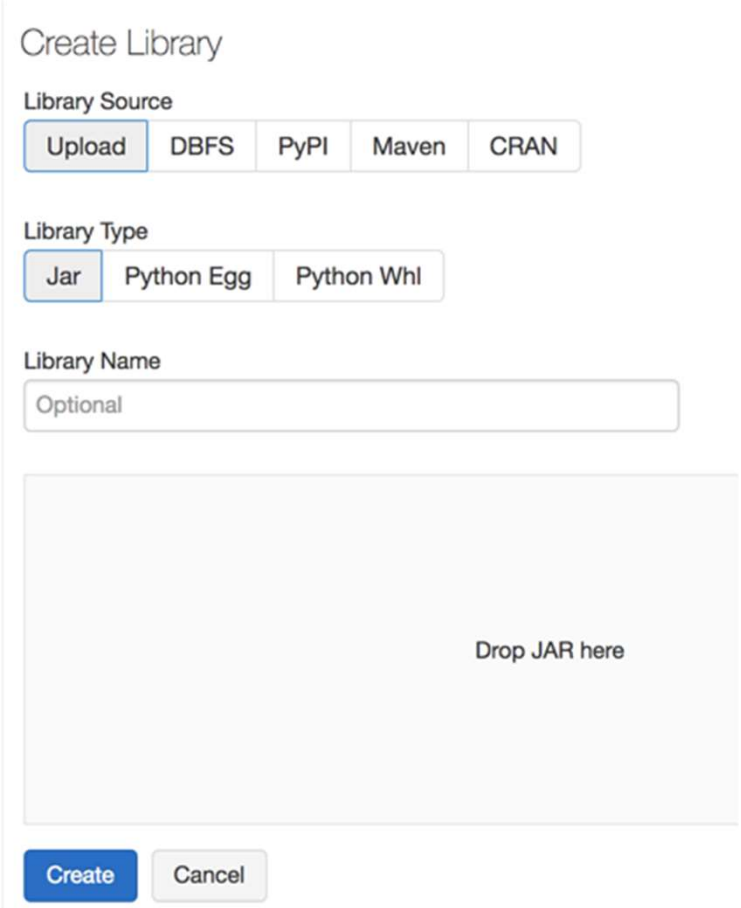
Libraries Overview

- To make third-party or custom code available to notebooks and jobs running on your clusters, you can install a library.
- Libraries can be installed using:
  - Workspace libraries
    - Serve as a local repository from which you create cluster-installed libraries
  - Cluster libraries
    - Can be used by all notebooks running on a cluster
    - Can install a cluster library directly from a public repository such as PyPI
  - Notebook-scoped Python libraries
    - Allow to install Python libraries and create an environment scoped to a notebook session
    - These libraries do not persist and must be re-installed for each session.

Hands-On: Create a workspace library

- Right-click the workspace folder where you want to store the library.
- Select Create > Library.





Hands-On: Install a library on a cluster

- Two ways to install a library on a cluster:
  - Install a workspace library that has been already been uploaded to the workspace.
  - Install a library for use with a specific cluster only

Manage Users & Groups

HDInsight vs Databricks

HDInsight is full fledged Hadoop with a decoupled storage and compute.

- Databricks is managed spark.
- Databricks is focused on collaboration, streaming and batch with a notebook experience.

In Databricks, all your notebooks are available and ready for use

- You don't need to think about anything else

HDInsight has Kafka, Storm and Hive that Databricks doesn’t have.

- It is better for processing very large data sets in a “let it run” kind of way.

Databricks is a lot more user friendly and easier to manage

Azure HDInsight brings both Hadoop and Spark under the same umbrella

When to use HDInsight?

Very reliable when we know the workloads and the cluster sizes

Using Hive is a perk, as its being open source and very similar to SQL

By using Hive, we take full advantage of MapReduce power

- Which shines in situations where there are huge amounts of data.

When to use Databricks?

Databricks is an ideal choice when

- The notebook interactive experience is a must
- When data engineers and data scientists must work together to get insights from data
- Another perk of using Databricks is its speed, thanks to Spark

Feature wise Comparison

	HDInsight	Databricks
Engine	Apache Hive or Apache Spark	Apache Spark, optimized for Databricks
Default Environment	Ambari (HortonWorks), Zeppelin if using Spark	Databricks Notebooks, RStudio for Databricks
De Facto Language	HiveQL, open source	R, Python, Scala, Java, SQL
Scalability	Not scalable, requires cluster shutdown to resize	Easy to change machines and allows autoscaling
Setup and managing	Complex, we must decide cluster types and sizes	Easy, Databricks offers two main types of services and clusters can be modified with ease

References:
https://www.clearpeaks.com/cloud-analytics-on-azure-databricks-vs-hdinsight-vs-data-lake-analytics/

Thanks