

Azure Databricks

August 2021 Camilla Gaardsted

SuperUsers A/S

Azure Data Bricks

- Azure Databricks is an **Apache Spark**-based analytics platform optimized for the Microsoft Azure cloud services platform
- Integration with **Azure Active Directory** enables you to run complete Azure-based solutions using Azure Databricks
- Azure Databricks **integrates deeply** with **Azure** databases and stores: SQL Data Warehouse, Cosmos DB, Data Lake Store, and Blob Storage

Azure Databricks

Standard

- All users are admins

Premium

Role based access

Azure Databricks

New Cluster

Cancel

Create Cluster

2-4 Workers: 28.0-56.0 GB Memory, 8-16 Cores, 1.5-3 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name

sparkcluster

Cluster Mode

Standard |

Pool

None |

Databricks Runtime Version

[Learn more](#)

Runtime: 7.5 ML (Scala 2.12, Spark 3.0.1) |

Autopilot Options

☒ Enable autoscaling

☒ Terminate after minutes of inactivity

Worker Type

Min Workers

Max Workers

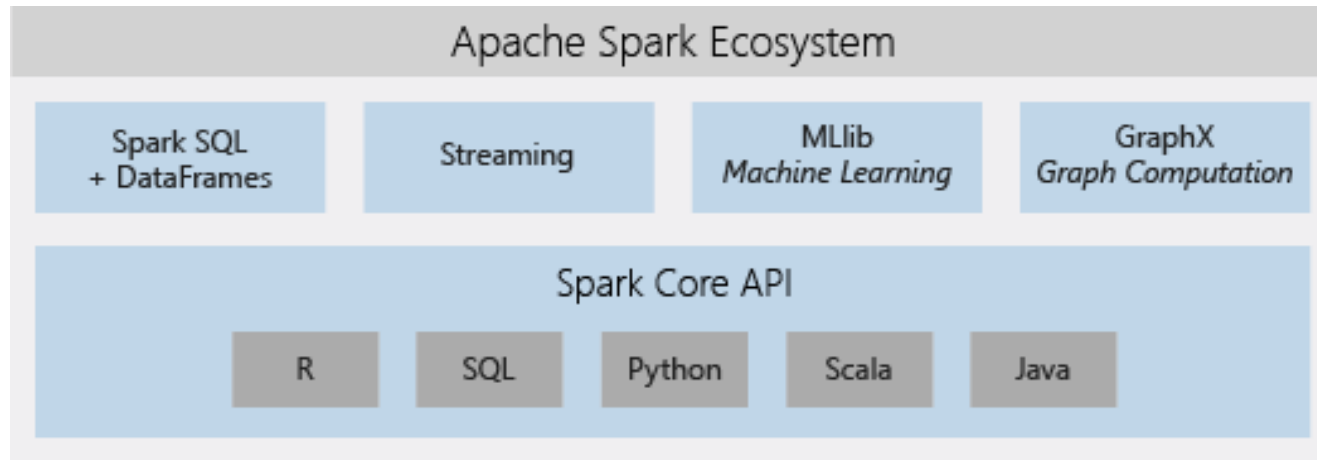
Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU |

Driver Type

Same as worker 14.0 GB Memory, 4 Cores, 0.75 DBU |

► Advanced Options

Azure Databricks components



Databricks - Cluster

- You run these workloads as a set of commands in a [notebook](#) or as an automated [job](#). Databricks makes a distinction between *all-purpose clusters* (interactive) and *job clusters (automated clusters)*.
- You use all-purpose clusters to analyze data collaboratively using interactive notebooks. You use job clusters to run fast and robust automated jobs.

Notebooks

A notebook is a **document** and contains

- Computer **code** (e.g. python)
- Rich **text** elements (paragraph, equations, links, etc...)
- data **resultsets**/visualizations

Jupyter Notebooks were originally used for ipython (interactive python)

- Run and edit notebooks in a web browser/text editor
- Document is a json file with ipynb file extension

Now Widely used for python, sql, powershell etc

Databricks - Notebooks

A notebook is a web-based interface to a document that contains runnable code, visualizations, and narrative text

Databricks supports code written in

- Python
- R
- Scala
- SQL

Notebook external formats

A source file with the extension `.scala`, `.py`, `.sql`, or `.r`.

HTML

- A Databricks notebook with an `.html` extension.

DBC Archive

- A Databricks archive.

IPython Notebook

- A Jupyter notebook with the extension `.ipynb`.

R

- An R Markdown document with the extension `.Rmd`.

Databricks - Notebook

Default sprog her er SQL

The screenshot displays the Databricks Notebook interface. At the top, the header shows 'Microsoft Azure | Databricks'. Below this, the notebook title is 'Quickstart Notebook (SQL)'. The interface includes a sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main workspace area shows two commands. Command 5 is a SQL statement to create a table from a CSV file. Command 6 is a SQL statement to select all data from the 'diamonds' table. The output of Command 6 is displayed as a table with 13 columns and 4 rows of data. The interface also shows execution times and user information for each command.

Quickstart Notebook (SQL)

mlclustercg

The next command creates a table from a Databricks dataset

Cmd 5

```
1 DROP TABLE IF EXISTS diamonds;
2
3 CREATE TABLE diamonds
4 USING csv
5 OPTIONS (path "/databricks-datasets/Rdatasets/data-001/csv/ggplot2/diamonds.csv", header "true")
6
```

OK

Command took 1.14 seconds -- by a user at 19.10.2020 21.34.10 on unknown cluster

Cmd 6

```
1 SELECT * from diamonds
```

	_c0	carat	cut	color	clarity	depth	table	price	x	y	z
1	1	0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
2	2	0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
3	3	0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
4	4	0.29	Premium	I	VS2	62.4	58	334	4.2	4.23	2.63

Showing the first 1000 rows.

Command took 0.45 seconds -- by a user at 19.10.2020 21.34.10 on unknown cluster

Textblok

Kodeblok

Kodeoutput

Databricks – Datasources

Directly supported in Databricks runtime or accesible via simple shell commands:

Avro file	MLflow experiment
Binary file	Parquet file
CSV file	XML file
Image	ZIP file
JSON file	Hive table
LZO compressed file	

Databricks – Datasource connections

Datasources which require a connection to the storage:

Azure Blob storage	MongoDB
Azure Data Lake Storage Gen1	Neo4j
Azure Data Lake Storage Gen2	Redis
Azure Cosmos DB	Riak Time Series
Azure Synapse Analytics	Snowflake
Cassandra	SQL Databases using JDBC
Couchbase	SQL Databases using the Apache Spark connector
ElasticSearch	

Databricks File System (DBFS)

Databricks File System (DBFS) is a distributed file system mounted into an Azure Databricks workspace and available on Azure Databricks clusters.

- Allows you to [mount](#) storage objects so that you can seamlessly access data without requiring credentials.
- Allows you to interact with object storage using directory and file semantics instead of storage URLs.
- Persists files to object storage, so you won't lose data after you terminate a cluster.

Databricks – Azure Blob storage

Directly access in a notebook **session** via storage account access key or SAS:

```
%python
spark.conf.set(
    "fs.azure.account.key.<storage-account-name>.blob.core.windows.net",
    "<storage-account-access-key>")

dbutils.fs.ls("wasbs://<container-name>@<storage-account-name>.blob.core.windows.net/<directory-name>")
```

Or via Spark Configuration property for **all cluster users**
For testing purposes only! Credentials are visible for all!

DBFS –Mount Azure Blob storage

- Mount a Blob storage container or a folder inside a container (block blobs only)

Databricks – Secret scopes

Two types of secret scopes:

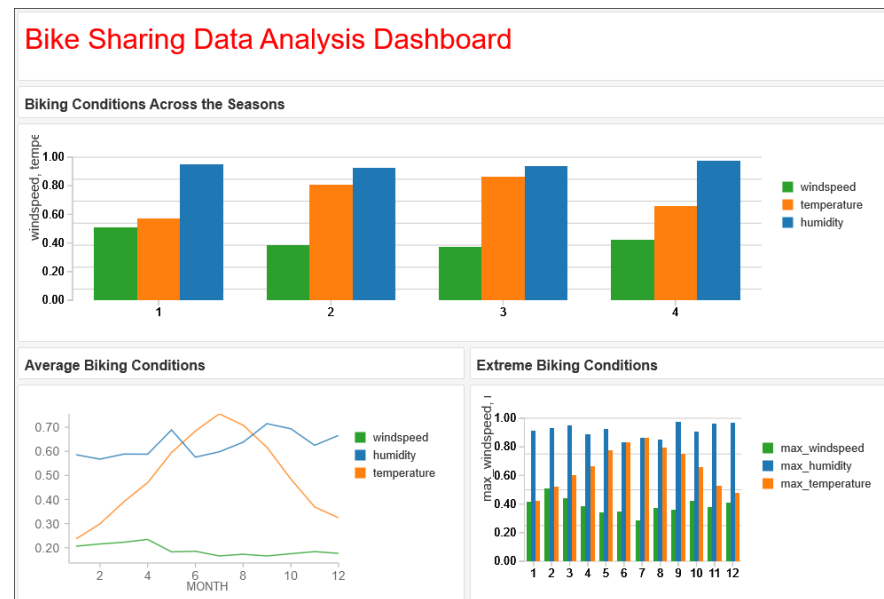
- Azure Key Vault-backed scopes
- Databricks-backed scopes

```
camilla@Azure:/usr/bin$ databricks secrets list-scopes
```

Scope	Backend	KeyVault URL
azuresecrets	AZURE_KEYVAULT	https://westuskeyvault2021.vault.azure.net/
bricksscope	DATABRICKS	N/A

Databricks - Dashboard

Dashboard allow you to publish graphs and visualizations derived from notebook output and share them in a presentation format with your organization



Databricks - Azure Key Vault backed scopes

Anvender credentials i en Azure Key Vault

Opret secret scope via url/databricks CLI

- Angiv Azure Key vault URI and ResourceID

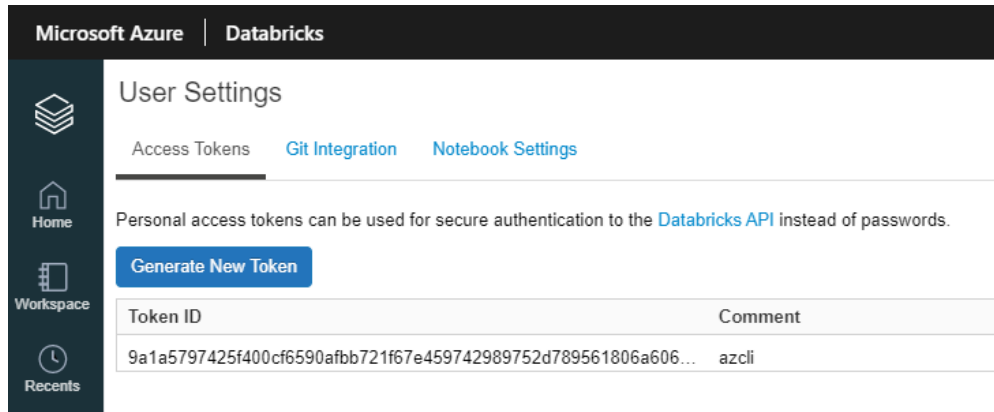
Databricks - Mount

- <mount-name> becomes the DBFS path
- <conf-key> either
 - fs.azure.**account.key**.<storage-account-name>.blob.core.windows.net
 - fs.azure.**sas**.<container-name>.<storage-account-name>.blob.core.windows.net

```
dbutils.fs.mount(  
  source = "wasbs://<container-name>@<storage-account-name>.blob.core.windows.net",  
  mount_point = "/mnt/<mount-name>",  
  extra_configs = {"<conf-key>":dbutils.secrets.get(scope = "<scope-name>", key = "<key-name>")})
```

Databricks - CLI

- Configure with url and token



Microsoft Azure | Databricks

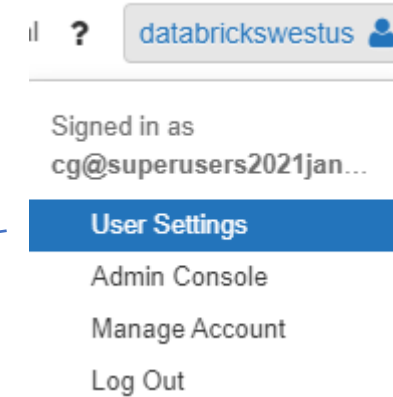
User Settings

Access Tokens [Git Integration](#) [Notebook Settings](#)

Personal access tokens can be used for secure authentication to the [Databricks API](#) instead of passwords.

[Generate New Token](#)

Token ID	Comment
9a1a5797425f400cf6590afbb721f67e459742989752d789561806a606...	azcli



il ? databrickswestus

Signed in as
cg@superusers2021jan...

[User Settings](#)

[Admin Console](#)

[Manage Account](#)

[Log Out](#)

Databricks - Jobs

A job is a way of running a notebook or JAR either immediately or on a scheduled basis. The other way to run a notebook is interactively in the [notebook UI](#)

Create and run jobs using the UI, the CLI, and by invoking the Jobs API. You can monitor job run results in the UI, using the CLI, by querying the API, and through email alerts

Databricks – Delta lake

Delta Lake is an open format storage layer that delivers reliability, security and performance on your data lake — for both streaming and batch operations

ACID

Time travel

Databricks - dokumentation

<https://docs.microsoft.com/en-us/azure/databricks>

<https://docs.databricks.com>

<https://spark.apache.org/docs/latest>

Databricks - DBFS

Browse enable under advanced settings in admin console

Spark - Dataset/DataFrame

A Dataset is a distributed collection of data (Spark 1.6+ replaced RDD)

The Dataset API is available in [Scala](#) and [Java](#)

Python does not have the support for the Dataset API

A DataFrame is a *Dataset* organized into named columns. It is conceptually equivalent to a table in a relational database or a data frame in R/Python, but with richer optimizations under the hood

Spark - Databases and tables

An Azure Databricks database is a collection of tables you query with

- SPARK API
- SPARK SQL

Global tables are persist and available across all clusters (Hive compatibility)

A local table is just a temporary view in a notebook

SPARK SQL

CREATE TABLE gives a table stored in files

REFRESH <tablename> will update the table data from the files

INSERTs are allowed

UPDATE and DELETE are only supported on tables that support ACID (transaction log) e.g. **DELTA** tables

DELTA format is default for tables from Databricks Runtime 8.0

Spark – Structured Streaming

Structured Streaming is a scalable and fault-tolerant stream processing engine built on the Spark SQL engine

When you load a Delta table as a **stream source** and use it in a streaming query, the query processes all of the data present in the table as well as any new data that arrives after the stream is started.

Read and write

MapReduce

MapReduce is a framework for processing parallelizable problems across large datasets using a large number of computers (nodes)

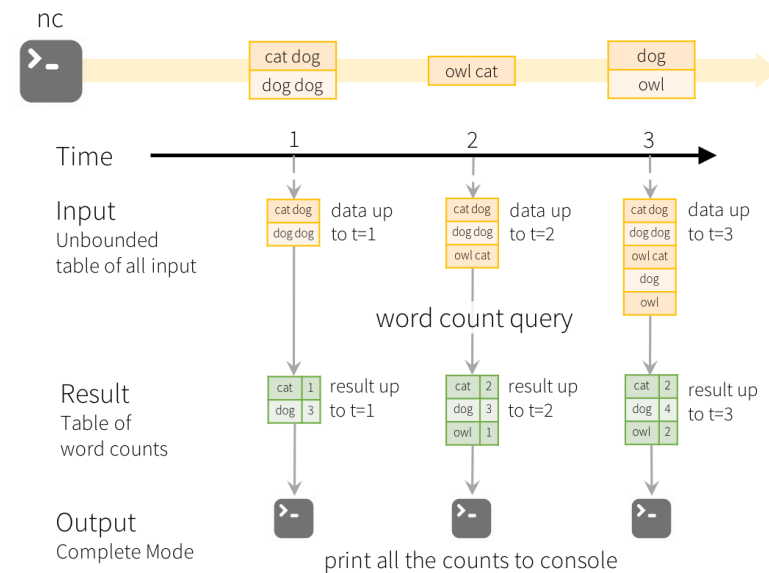
Hyperspace

An open source indexing subsystem that brings index-based query acceleration to Apache Spark™ and big data workloads.

Databricks – Structured Streaming

Output is defined and written in

- Complete mode
- Append mode
- Update mode

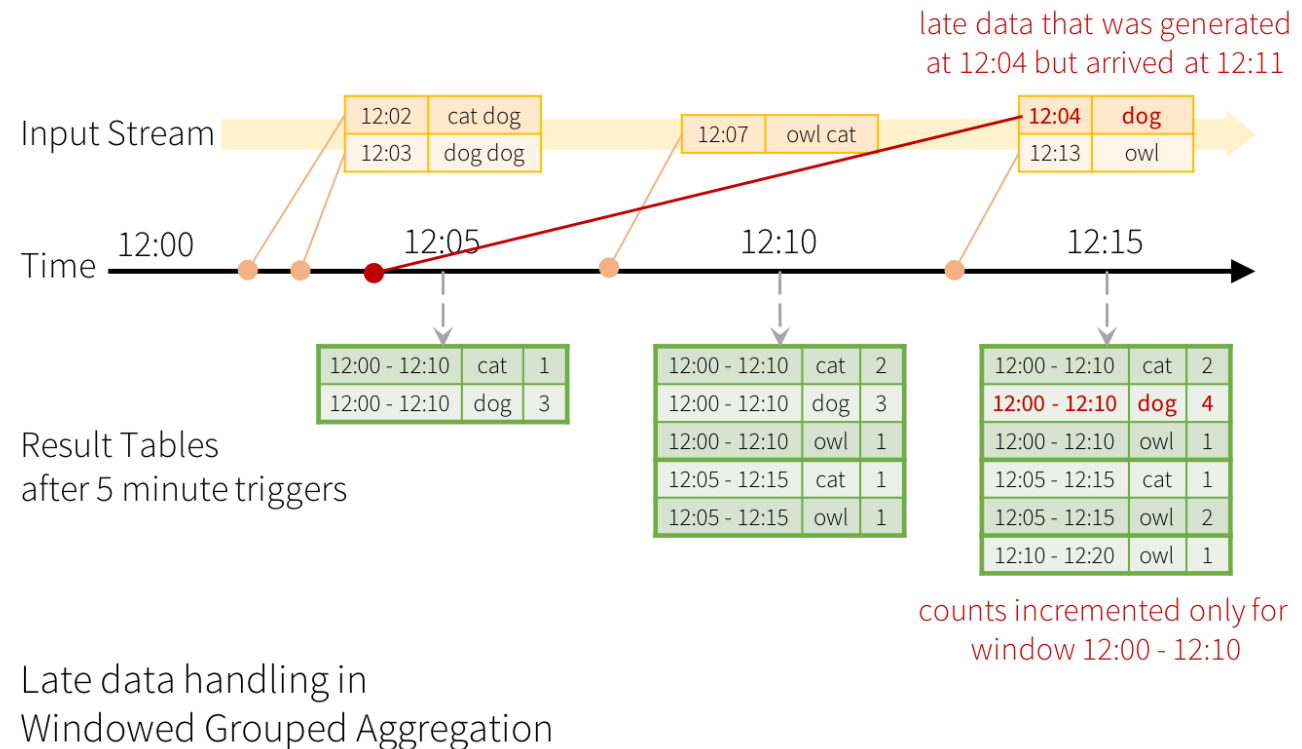


Model of the Quick Example

Delta Lake

- Databricks har nu delta som default format for oprettelse af tabeller via SQL kommandoer
- Dvs når man opretter en ny table via `df.write.saveasTable`, så er det default delta

Databricks – Structured Streaming late events



Databricks – Structured Streaming watermark

Late events are ignored

