

To Open Azure Databricks Service

The screenshot shows the Azure portal interface. At the top, there's a navigation bar with the Microsoft Azure logo and a search bar. Below this, there's a row of service icons including 'Create a resource', 'Monitor', 'Virtual machines', 'Azure Databricks', 'SQL databases', 'Quickstart Center', 'App Services', 'Storage accounts', 'Azure Cosmos DB', and 'More services'. The 'Resources' section is active, displaying a table of recent resources. A tooltip for 'azddatabbiba_Hexaware' is visible, showing a 'View' button. The 'Navigate' section at the bottom has links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'. The 'Tools' section at the bottom shows the URL for the Azure Databricks workspace.

Resources

Name	Type	Last Viewed
azddatabbiba_Hexaware	Azure Databricks Service	18 minutes ago
msdocs-core-sql-plh-data	SQL database	5 hours ago

Navigate

- Subscriptions
- Resource groups
- All resources
- Dashboard

Tools

https://portal.azure.com/#@ihti.onmicrosoft.com/asset/Microsoft_Azure_Databricks/Workspace/subscriptions/984f097c-963c-4eb6-a20d-839457ae9f08/resourceGroups/rg-azuser934_mml.local-n/vb13/providers/Microsoft.Databricks/workspaces/azddatabbiba_Hexaware

azddatabbiba_Hexaware

Azure Databricks Service

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Virtual Network Peerings

Encryption

Networking

Properties

Locks

Monitoring

Diagnostic settings

Essentials

Status : Active

Resource group : rg-azuser934_mml.local-n/vb13

Location : East US

Subscription : Azure subscription 1

Subscription ID : 984f097c-963c-4eb6-a20d-839457ae9f08

Tags (edit) : Add tags

Managed Resource Group : databricks-rg-azddatabbiba_Hexaware-grhdmlov6e...

URL : https://adb-3271486371374355.15.azuredatabricks...

Pricing Tier : Premium (+ Role-based access controls)(Click to ch...

Launch Workspace

Microsoft Azure databricks

Get started with Databricks

We're warming up your SQL warehouse
You'll be connected to your warehouse shortly.

00:04 Starting warehouse

Explore sample projects
Don't have your data handy? Learn how to gain insights in just a few steps with these sample projects.

```
1 SELECT
2 COUNT(distinct(custkey))
3 FROM
4 'samples','tpch','customer'
5 WHERE
```

SQL and visualizations KPI dashboard Data analysis in Python

Bring in your own data

Skip onboarding

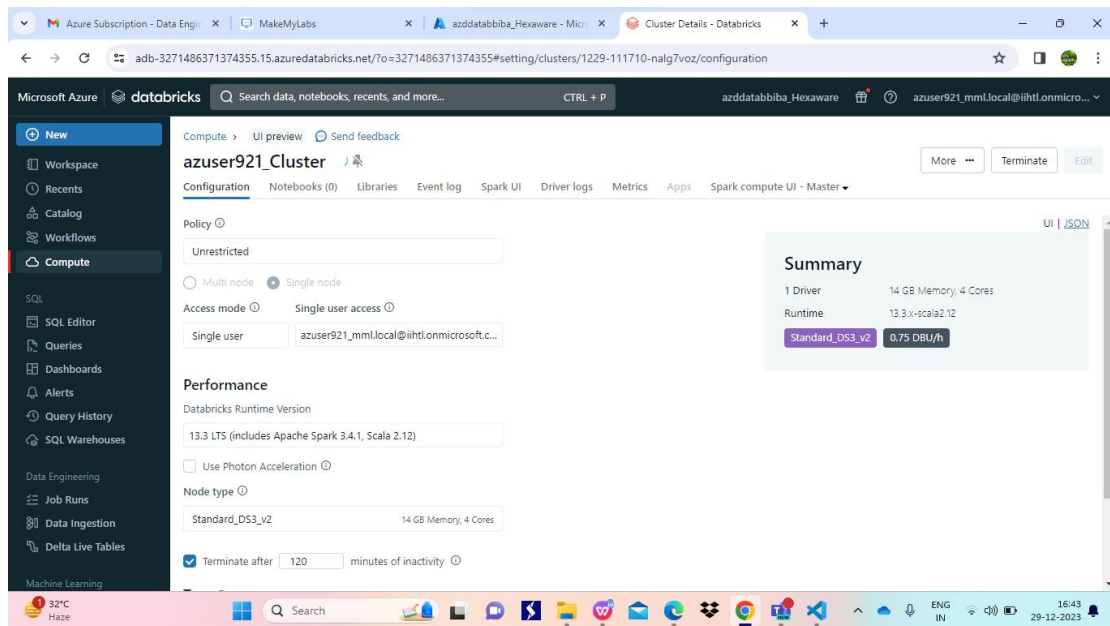
To create cluster in azure databricks

Microsoft Azure databricks

Cluster

Policy Runtime Active m... Active co... Active D... Source Creator Notebooks

Personal Comp	13.3	14 GB	4 cores	0.75	UI	azuser906_mml...	-	■	⋮
Personal Comp	14.2 ML	14 GB	4 cores	0.75	UI	azuser910_mml...	2	■	⋮
Personal Comp	13.3	14 GB	4 cores	0.75	UI	azuser911_mml...	1	■	⋮
Personal Comp	13.3	14 GB	4 cores	0.75	UI	azuser914_mml...	2	■	⋮
Personal Comp	13.3	14 GB	4 cores	0.75	UI	azuser917_mml...	1	■	⋮
Personal Comp	13.3	14 GB	4 cores	0.75	UI	azuser929_mml...	5	■	⋮
Personal Comp	14.2 ML	14 GB	4 cores	0.75	UI	azuser931_mml...	1	■	⋮
Personal Comp	13.3	-	-	1.5	UI	azuser908_mml...	-	■	⋮



To create notebook in cAzure data bricks

Untitled Notebook 2023-12-29 17:17:24 Python

File Edit View Run Help Last edit was in 3 minutes Provide feedback

Run all azuser921_Cluster Schedule Share

Connected Go to last run cell

azuser921_Cluster Runtime Driver DBR 13.3 LTS • Spark 3.4.1 • Scala 2.12 Standard_DS3_v2 • 14 GB • 4 Cores

Recent resources

Resource	DBR
azuser921_Cluster	DBR 13.3 LTS
azuser919_mmllocal@iitl.onmicrosoft.com's Clu...	DBR 13.3 LTS
azuser915_cluster	DBR 13.3 LTS

More... Create new resource...

Detach Detach & re-attach Restart Terminate Configuration Driver logs Spark UI Web Terminal

```
1 import pyspark
2 from pyspark.sql import SparkSession
3 from pyspark.sql.functions import approx_count_distinct
4 from pyspark.sql.functions import collect_set, sum,
5 from pyspark.sql.functions import first, last, kurtosis
6 from pyspark.sql.functions import stddev, stddev_samp
7 from pyspark.sql.functions import variance, var_samp
8
9 spark = SparkSession.builder.appName('SparkByExample').getOrCreate()
10
11 simpleData = [{"James", "Sales", 3000},
12               {"Michael", "Sales", 4600},
13               {"Robert", "Sales", 4100},
14               {"Maria", "Finance", 3000},
15               {"James", "Sales", 3000},
16               {"Scott", "Finance", 3300},
17               {"Jen", "Finance", 3900},
18               {"Jeff", "Marketing", 3000},
19               {"Kumar", "Marketing", 2000},
20               {"Saif", "Sales", 4100}]
21
22 schema = ["employee_name", "department", "salary"]
```

Untitled Notebook 2023-12-29 17:17:24 Python

File Edit View Run Help Last edit was in 1 minute Provide feedback

Run all azuser921_Cluster Schedule Share

Run all cells in this notebook.

```
26 df.printSchema()
27 df.show(truncate=False)
28
29 print("approx_count_distinct: " + \
30       str(df.select(approx_count_distinct("salary")).collect()[0][0]))
31
32 print("avg: " + str(df.select(avg("salary")).collect()[0][0]))
33
34 df.select(collect_list("salary")).show(truncate=False)
35
36 df.select(collect_set("salary")).show(truncate=False)
37
38 df2 = df.select(countDistinct("department", "salary"))
39 df2.show(truncate=False)
40 print("Distinct Count of Department & Salary: "+str(df2.collect()[0][0]))
41
42 print("count: "+str(df.select(count("salary")).collect()[0][0]))
43 df.select(first("salary")).show(truncate=False)
44 df.select(last("salary")).show(truncate=False)
45 df.select(kurtosis("salary")).show(truncate=False)
46 df.select(max("salary")).show(truncate=False)
47 df.select(min("salary")).show(truncate=False)
48 df.select(mean("salary")).show(truncate=False)
49 df.select(skewness("salary")).show(truncate=False)
50 df.select(stddev("salary"), stddev_samp("salary"), \
51         stddev_pop("salary")).show(truncate=False)
```


A spark program run on notepad it shows the following output

The screenshot shows a Databricks notebook titled "Untitled Notebook 2023-12-29 17:17:24". The code in the notebook is as follows:

```
52 df.select(sum("salary")).show(truncate=False)
53 df.select(sumDistinct("salary")).show(truncate=False)
54 df.select(variance("salary"),var_samp("salary"),var_pop("salary")) \
55   .show(truncate=False)
```

The output of the first query is shown below:

```
root
|-- employee_name: string (nullable = true)
|-- department: string (nullable = true)
|-- salary: long (nullable = true)
```

employee_name	department	salary
James	Sales	3000
Michael	Sales	4600
Robert	Sales	4100
Maria	Finance	3000
James	Sales	3000
Scott	Finance	3300
Jen	Finance	3900
Jeff	Marketing	3000
Kumar	Marketing	2000
Saif	Sales	4100

Current date program

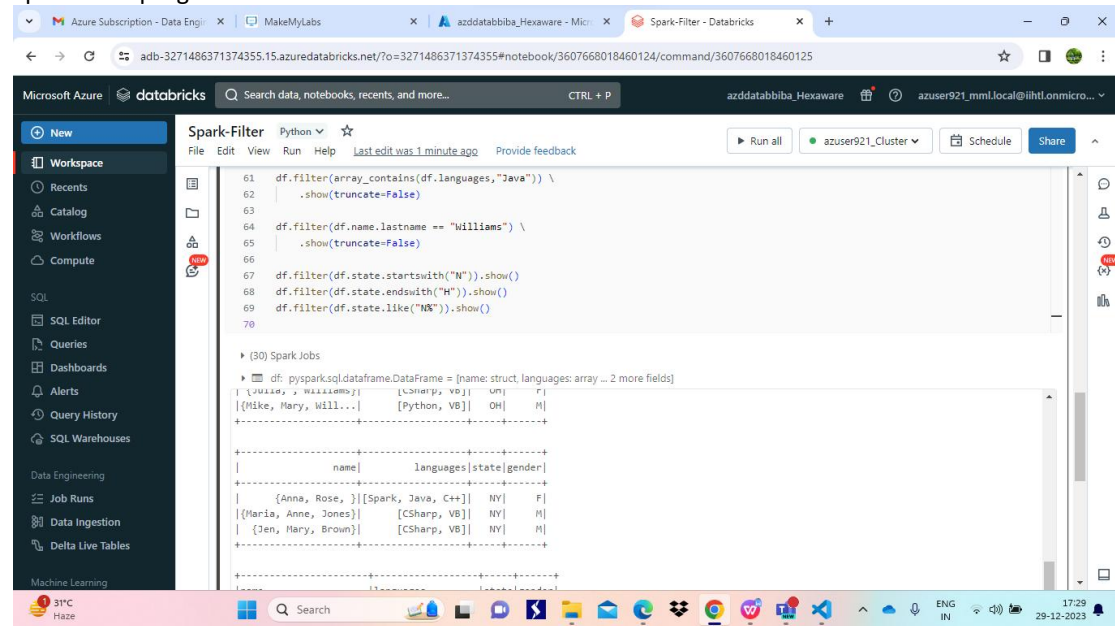
The screenshot shows a Databricks notebook titled "currentdate". The code in the notebook is as follows:

```
1 import pyspark
2 from pyspark.sql import SparkSession
3 from pyspark.sql.functions import col
4 from pyspark.sql.functions import to_timestamp, current_timestamp
5 from pyspark.sql.types import StructType, StructField, StringType, IntegerType, LongType
6
7 spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()
8
9 schema = StructType([
10     StructField("seq", StringType(), True)])
11
12 dates = ['1']
13
14 df = spark.createDataFrame([list('1')], schema=schema)
15
16 df.show()
```

The output of the program is shown below:

```
df: pyspark.sql.dataframe.DataFrame = [seq: string]
+----+
|seq|
+----+
| 1|
+----+
```

Spark-filter program



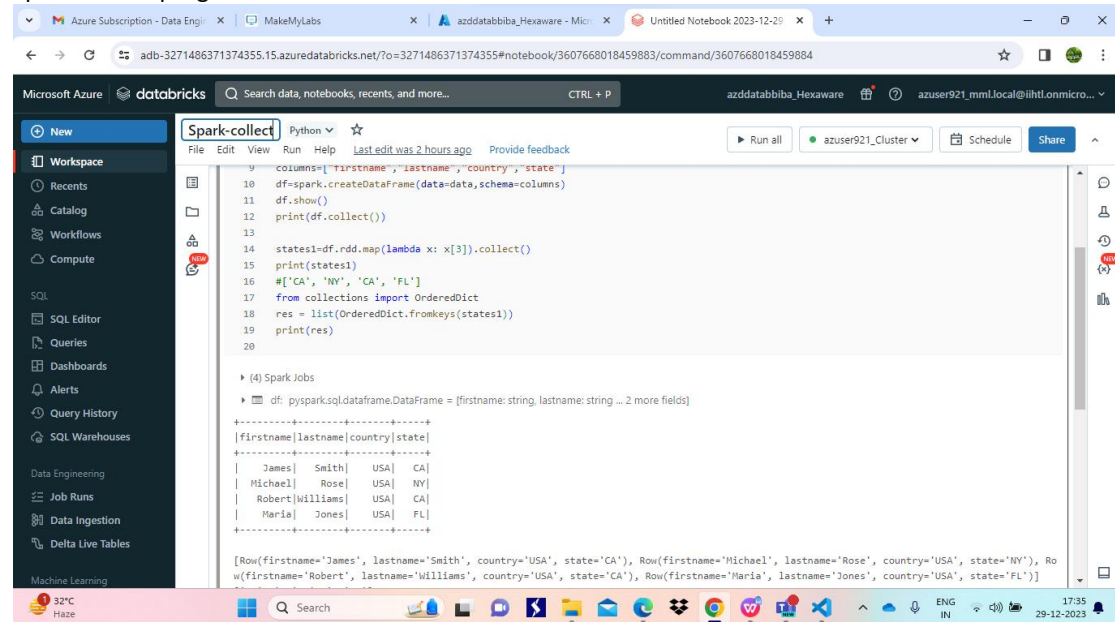
The screenshot shows a Databricks notebook interface with the title "Spark-Filter". The notebook is written in Python and contains the following code:

```
61 df.filter(array_contains(df.languages,"Java")) \
62     .show(truncate=False)
63
64 df.filter(df.name.lastname == "Williams") \
65     .show(truncate=False)
66
67 df.filter(df.state.startswith("N")).show()
68 df.filter(df.state.endswith("H")).show()
69 df.filter(df.state.like("N%")).show()
70
```

The output of the notebook shows the Spark Jobs section with a table of results:

name	languages	state	gender
{Anna, Rose, ...}	{Spark, Java, C++}	NY	F
{Maria, Anne, Jones}	{CSharp, VB}	NY	M
{Jen, Mary, Brown}	{CSharp, VB}	NY	M

Spark Collect program



The screenshot shows a Databricks notebook interface with the title "Spark-collect". The notebook is written in Python and contains the following code:

```
9 columns=["firstname", "lastname", "country", "state"]
10 df=spark.createDataFrame(data=data,schema=columns)
11 df.show()
12 print(df.collect())
13
14 states1=df.rdd.map(lambda x: x[3]).collect()
15 print(states1)
16 #['CA', 'NY', 'CA', 'FL']
17 from collections import OrderedDict
18 res = list(OrderedDict.fromkeys(states1))
19 print(res)
20
```

The output of the notebook shows the Spark Jobs section with a table of results:

firstname	lastname	country	state
James	Smith	USA	CA
Michael	Rose	USA	NY
Robert	Williams	USA	CA
Maria	Jones	USA	FL

Spark-Joins program

The screenshot displays the Databricks workspace interface for a notebook titled "Spark-Joins". The notebook is written in Python and contains the following code:

```
65 empDF.createOrReplaceTempView("EMP")
66 deptDF.createOrReplaceTempView("DEPT")
67
68 joinDF = spark.sql("select * from EMP e, DEPT d where e.emp_dept_id == d.dept_id") \
69     .show(truncate=False)
70
71 joinDF2 = spark.sql("select * from EMP e INNER JOIN DEPT d ON e.emp_dept_id == d.dept_id") \
72     .show(truncate=False)
73
74
75
```

The output of the notebook shows the results of the SQL queries. The first query (joinDF) returns a table with 7 columns: emp_id, name, superior_emp_id, year_joined, emp_dept_id, gender, salary, dept_name, and dept_id. The second query (joinDF2) returns a table with 7 columns: emp_id, name, superior_emp_id, year_joined, emp_dept_id, gender, and salary.

emp_id	name	superior_emp_id	year_joined	emp_dept_id	gender	salary	dept_name	dept_id
1	Smith	-1	2018	10	M	3000	Finance	10
3	Williams	1	2010	10	M	1000	Finance	10
4	Jones	2	2005	10	F	2000	Finance	10
2	Rose	1	2010	20	M	4000	Marketing	20
5	Brown	2	2010	40		-1	IT	40

emp_id	name	superior_emp_id	year_joined	emp_dept_id	gender	salary
1	Smith	-1	2018	10	M	3000
3	Williams	1	2010	10	M	1000
4	Jones	2	2005	10	F	2000
2	Rose	1	2010	20	M	4000
5	Brown	2	2010	40		-1

The interface also shows a sidebar with navigation options like Workspace, Recents, Catalog, Workflows, Compute, SQL, SQL Editor, Queries, Dashboards, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, Data Ingestion, and Delta Live Tables. The bottom status bar indicates the temperature is 32°C and the time is 17:37 on 29-12-2023.