Understanding Databricks/Spark Performance Tuning

Lesson 02: Easiest Fix – Optimizing the Hardware

by Bryan Cafferky from my YouTube channel

databricks

Where We're Going?

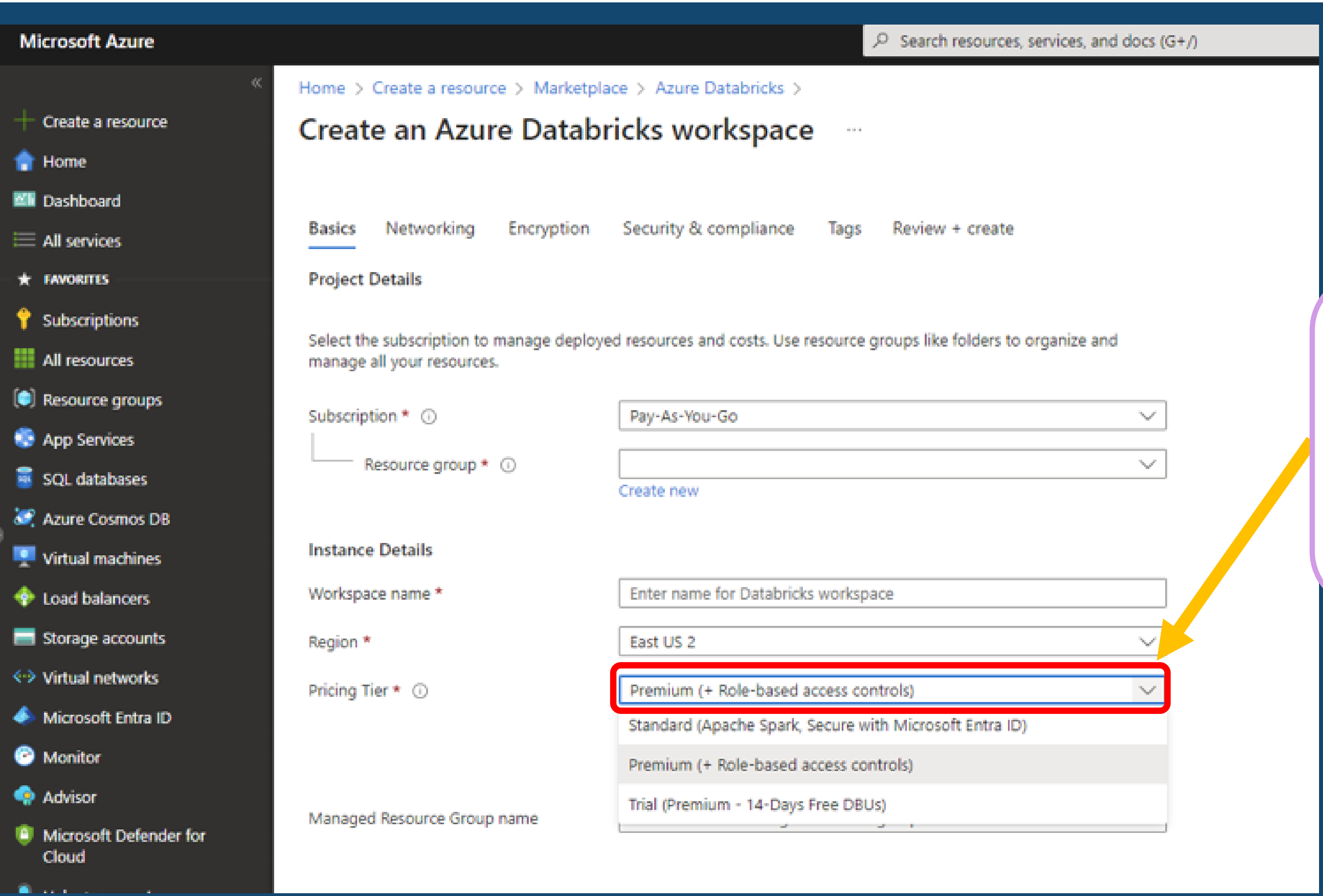
- Why Compute Resources Matter
- Databricks Workspace – Standard or Premium
- Databricks/Apache Spark Cluster Architecture
- Hardware Under the Cluster Architecture
- Optimizing the Cluster Configuration Step By Step
- Shuffles & Spills
- Wrap Up

Why Compute Resources Matter

- Never Talked About in Depth
- Cost vs. Performance Trade Off
- Single Easiest Thing You Can Change to Improve Performance
- There are Interdependencies Between Performance Optimization & the Compute Selection/Options

Databricks Workspace

Databricks Workspace



Create an Azure Databricks workspace ...

Project Details

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

Subscription * : Pay-As-You-Go

Resource group * :

Instance Details

Workspace name * : Enter name for Databricks workspace

Region * : East US 2

Pricing Tier * : Premium (+ Role-based access controls)

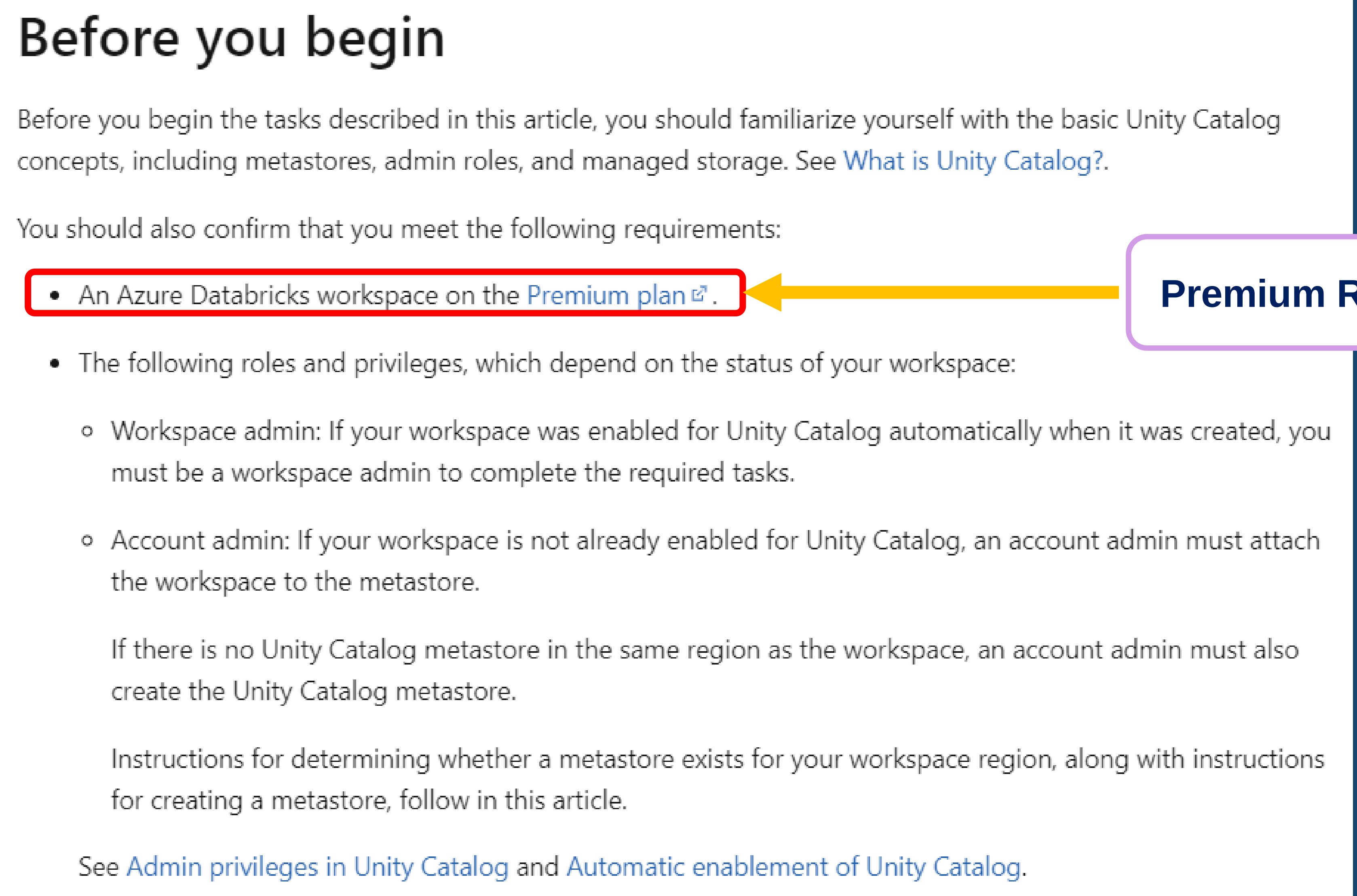
- Standard (Apache Spark, Secure with Microsoft Entra ID)
- Premium (+ Role-based access controls)
- Trial (Premium - 14-Days Free DBUs)

Managed Resource Group name

Premium vs. Standard

- ➢ Unity Catalog Requires Premium
- ➢ Delta Live Tables Requires Premium
- ➢ Photon Does NOT Require Premium

Unity Catalog



Before you begin

Before you begin the tasks described in this article, you should familiarize yourself with the basic Unity Catalog concepts, including metastores, admin roles, and managed storage. See What is Unity Catalog?.

You should also confirm that you meet the following requirements:

- An Azure Databricks workspace on the Premium plan.
- The following roles and privileges, which depend on the status of your workspace:
  - Workspace admin: If your workspace was enabled for Unity Catalog automatically when it was created, you must be a workspace admin to complete the required tasks.
  - Account admin: If your workspace is not already enabled for Unity Catalog, an account admin must attach the workspace to the metastore.

    If there is no Unity Catalog metastore in the same region as the workspace, an account admin must also create the Unity Catalog metastore.

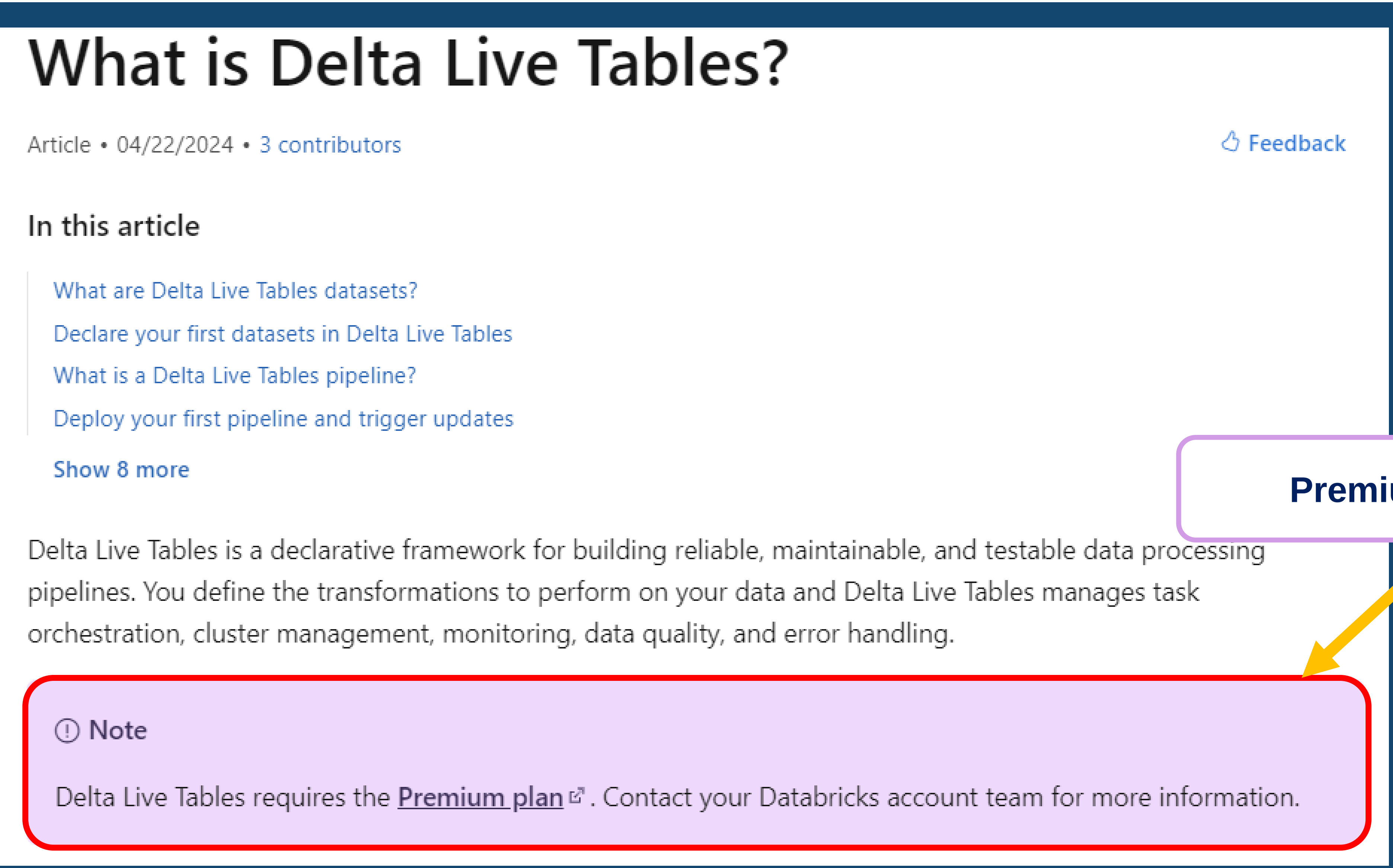
    Instructions for determining whether a metastore exists for your workspace region, along with instructions for creating a metastore, follow in this article.

  See Admin privileges in Unity Catalog and Automatic enablement of Unity Catalog.

Premium Required for Unity Catalog

https://learn.microsoft.com/en-us/azure/databricks/data-governance/unity-catalog/get-started#before-you-begin

Delta Live Tables (DLT)



What is Delta Live Tables?

Article • 04/22/2024 • 3 contributors

Feedback

In this article

Show 8 more

Delta Live Tables is a declarative framework for building reliable, maintainable, and testable data processing pipelines. You define the transformations to perform on your data and Delta Live Tables manages task orchestration, cluster management, monitoring, data quality, and error handling.

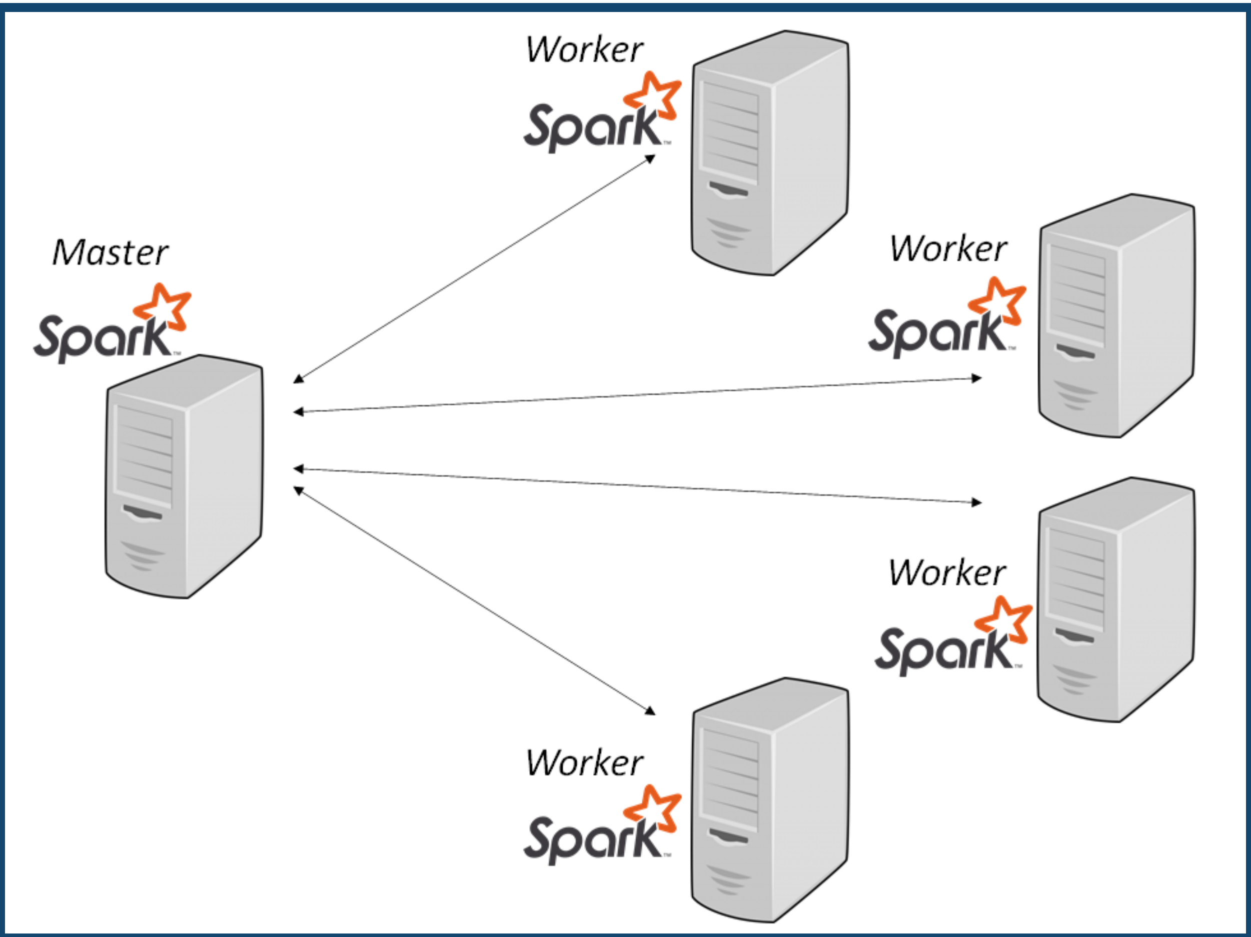
ⓘ Note

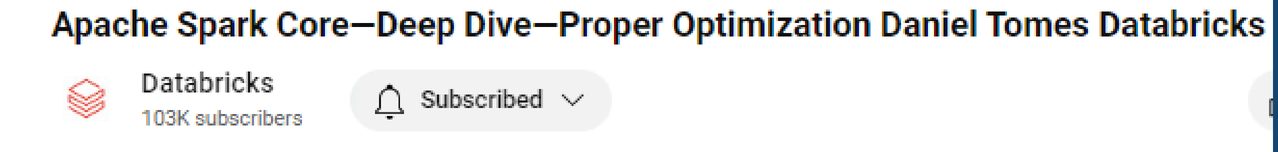
Delta Live Tables requires the Premium plan. Contact your Databricks account team for more information.

Premium Required for DLT

https://learn.microsoft.com/en-us/azure/databricks/delta-live-tables/

Cluster Architecture





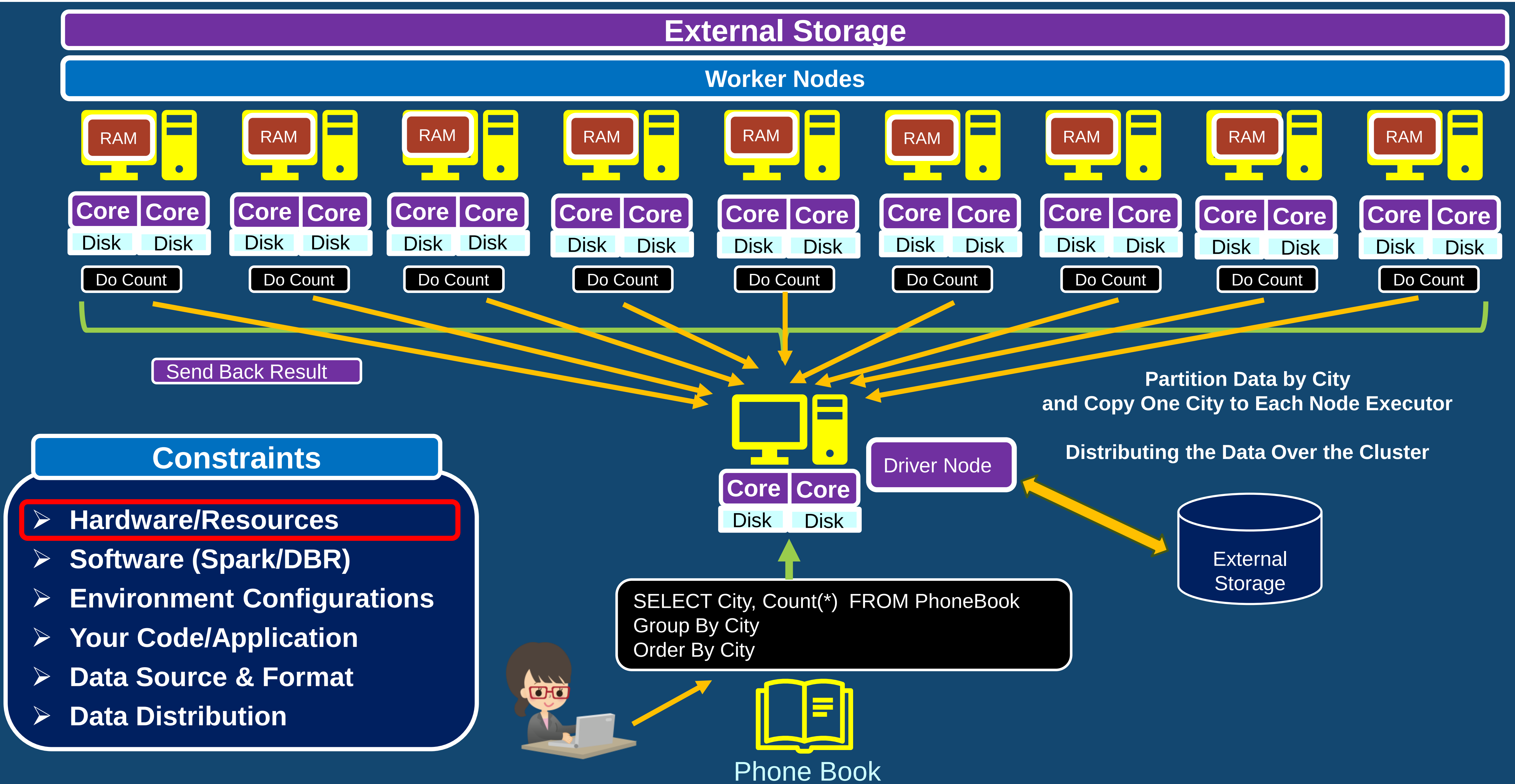
Apache Spark Core—Deep Dive—Proper Optimization Daniel Tomes Databricks

Databricks

103K subscribers

https://www.youtube.com/watch?v=daXEp4HmS-E&t=2111s

Cluster Architecture



Partition Data by City
and Copy One City to Each Node Executor

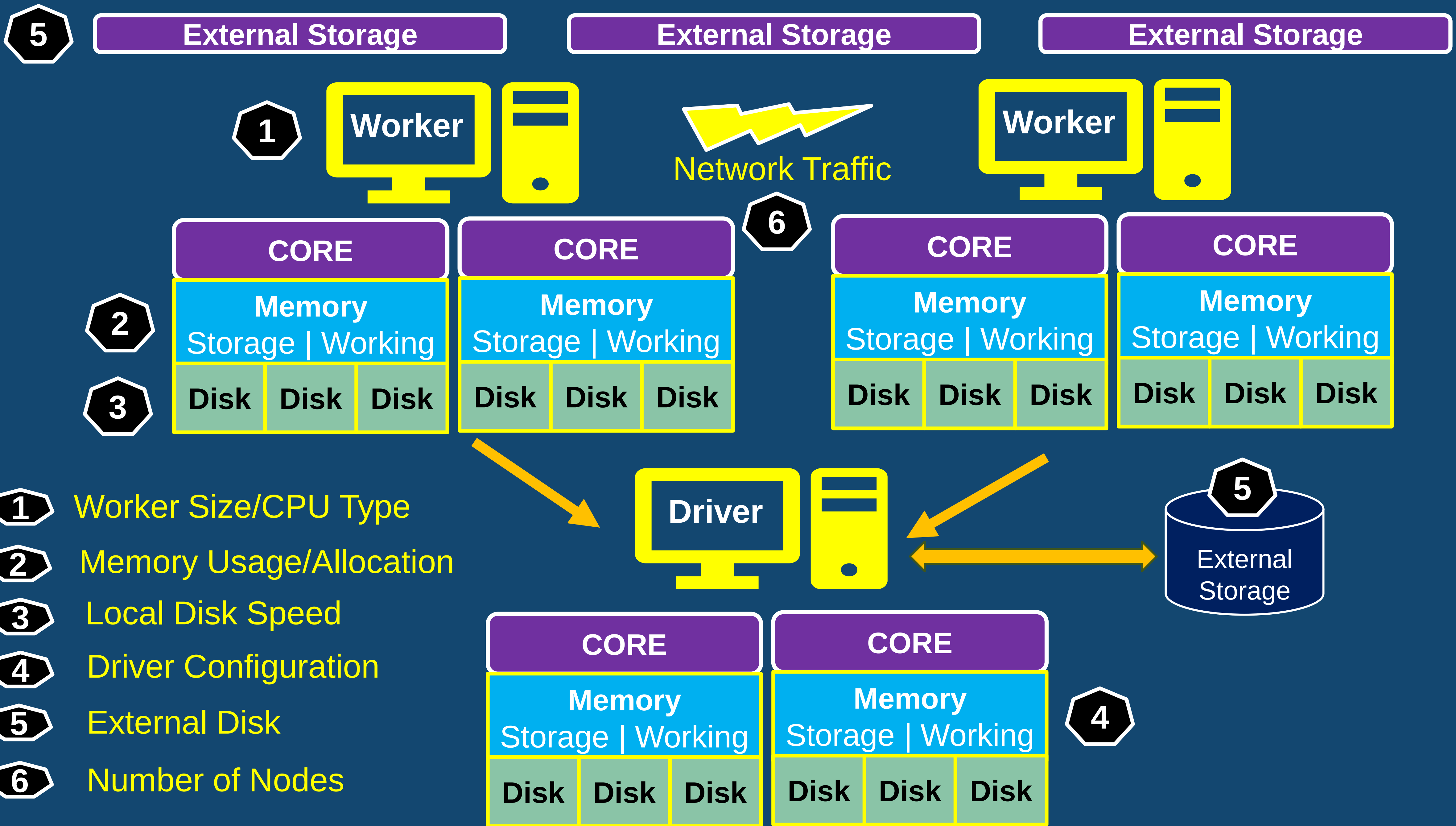
Distributing the Data Over the Cluster

```
SELECT City, Count(*)  FROM PhoneBook
Group By City
Order By City
```

Constraints

- **Hardware/Resources**
- **Software (Spark/DBR)**
- **Environment Configurations**
- **Your Code/Application**
- **Data Source & Format**
- **Data Distribution**

Hardware Focus



1 Worker Size/CPU Type

2 Memory Usage/Allocation

3 Local Disk Speed

4 Driver Configuration

5 External Disk

6 Number of Nodes

Hardware – CPU & Memory

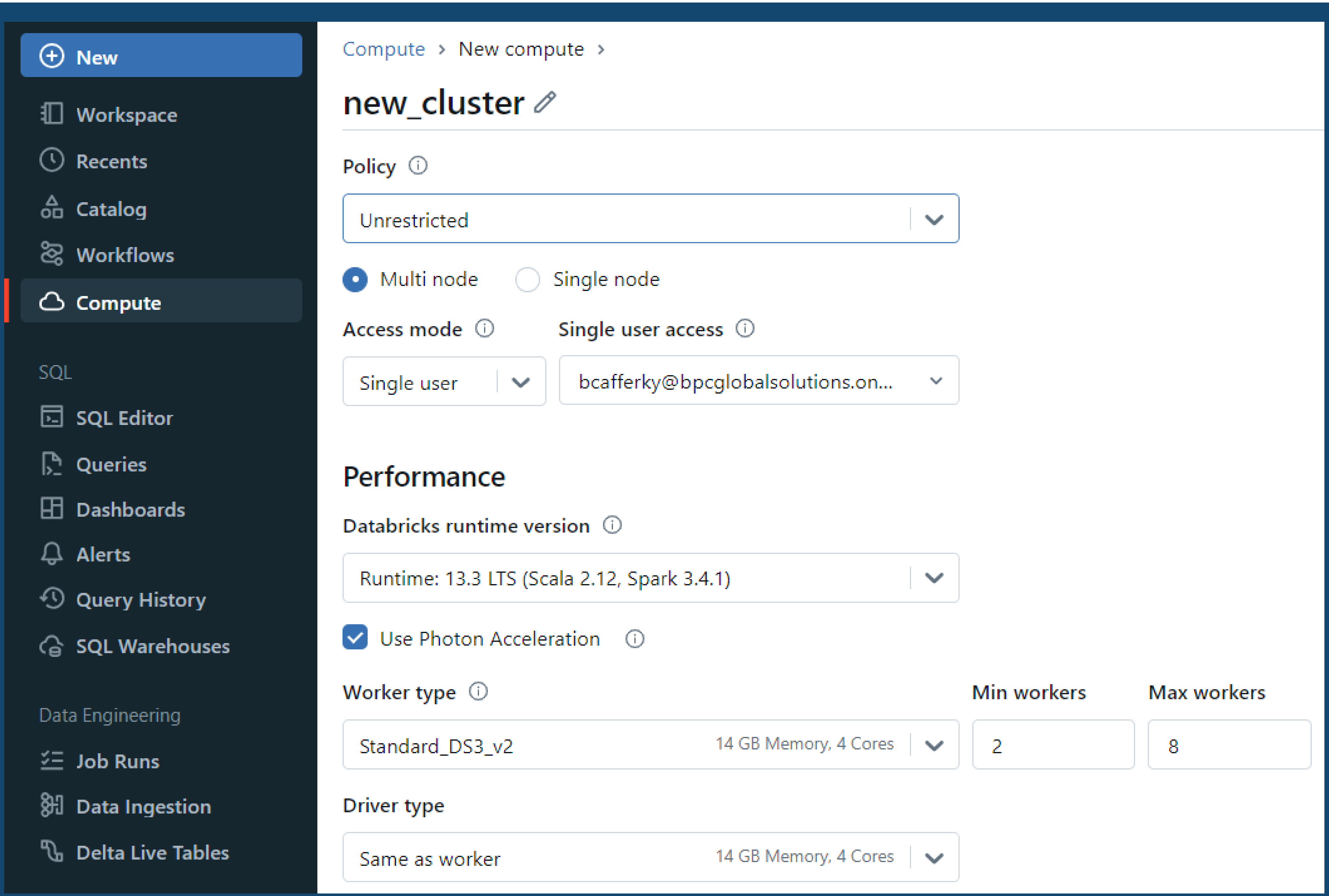
- **CPU/GPU Type**
  - Workload Type
- **Number of Cores**
  - Core = Executor = Task = Partition
  - # of Cores = Degree of Parallelism
- **Memory**
  - Enough to Support Your Workload
  - Avoid Spills
  - Split Between Working & Storage Memory

Hardware – Input/Output

- **Network**
  - Latency (Local vs. Cross Region vs. On Prem)
  - Bandwidth
  - External Resource Latency (example: Cosmos DB vs Azure SQL)
- **Storage (Local, Remote (External))**
  - Type/Speed (HDD, SSD)
  - Locality
- **I/O**
  - Spills
  - Waiting for Disk Writes
- **Databricks Runtime Version (Effects Features)**

Optimizing the Cluster Configuration

Cluster – Configuration Settings



Compute › New compute ›

new_cluster

Policy: Unrestricted

● Multi node ○ Single node

Access mode: Single user

Single user access: bcafferky@bpcglobalsolutions.on...

Performance

Databricks runtime version: Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

☑ Use Photon Acceleration

Worker type	Min workers	Max workers
Standard_DS3_v2 14 GB Memory, 4 Cores	2	8

Driver type: Same as worker 14 GB Memory, 4 Cores

We'll Walk Through Filling Out This Screen.

https://docs.databricks.com/en/compute/configure.html

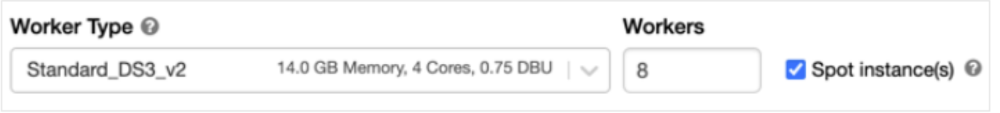
https://learn.microsoft.com/en-us/azure/databricks/compute/configure

Cluster – Configuration Best Practices

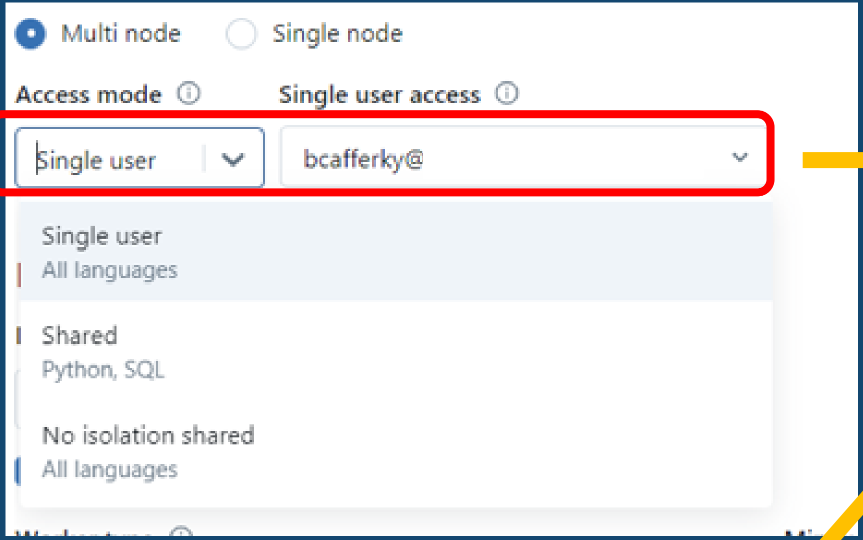
Use Case	Notes	Recommendation
Analysis ★	Interactive Development	➢ Single Node with High Memory and Cores ➢ Likely require reading the same data repeatedly, so recommended node types are storage optimized with disk cache enabled.
Basic Batch ETL ★	No Wide Transformations	Compute Optimized
Complex ETL ★	Has Wide Transformations	Compute Optimized with Less Nodes
ML Training Experimentation/Dev ★		Single Node Type with High Memory and Cores
ML Training Production ★	Minimal Worker Nodes	Storage Optimized with Disk Caching Enabled or GPU (lacks disk caching)

https://learn.microsoft.com/en-us/azure/databricks/compute/cluster-config-best-practices

Cluster Options

Use Case	Notes	Recommendation
Spot Instances ★	Saves money by uses available capacity. https://learn.microsoft.com/en-us/azure/virtual-machines/spot-vms	 Worker Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU Workers: 8 ☑ Spot instance(s)
Serverless Compute ★	Near instant Cluster. Unity Catalog must be enabled. https://learn.microsoft.com/en-us/azure/databricks/compute/serverless https://learn.microsoft.com/en-us/azure/databricks/release-notes/serverless#limitations https://www.databricks.com/trust/security-features/serverless-security	Like having a set of VMs on standby. When you need them, they are allocated to your work almost instantly.
Photon ★	Vastly faster processing.	

Cluster – Access Mode



● Multi node ○ Single node

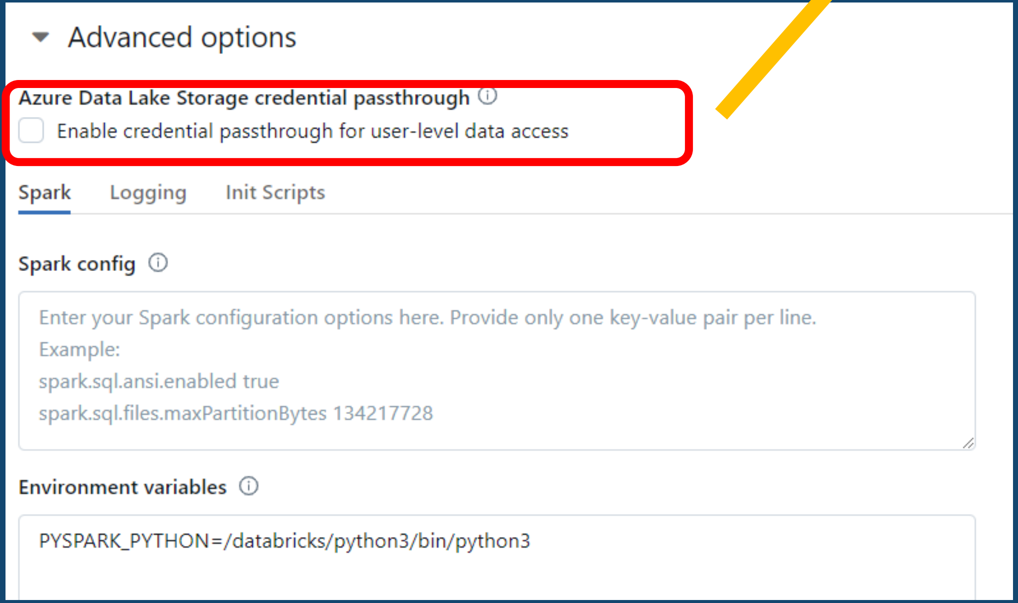
Access mode: Single user

Single user access: bcafferky@

Single user
All languages

Shared
Python, SQL

No isolation shared
All languages



Advanced options

Azure Data Lake Storage credential passthrough

□ Enable credential passthrough for user-level data access

Spark config

```
Enter your Spark configuration options here. Provide only one key-value pair per line.
Example:
spark.sql.ansi.enabled true
spark.sql.files.maxPartitionBytes 134217728
```

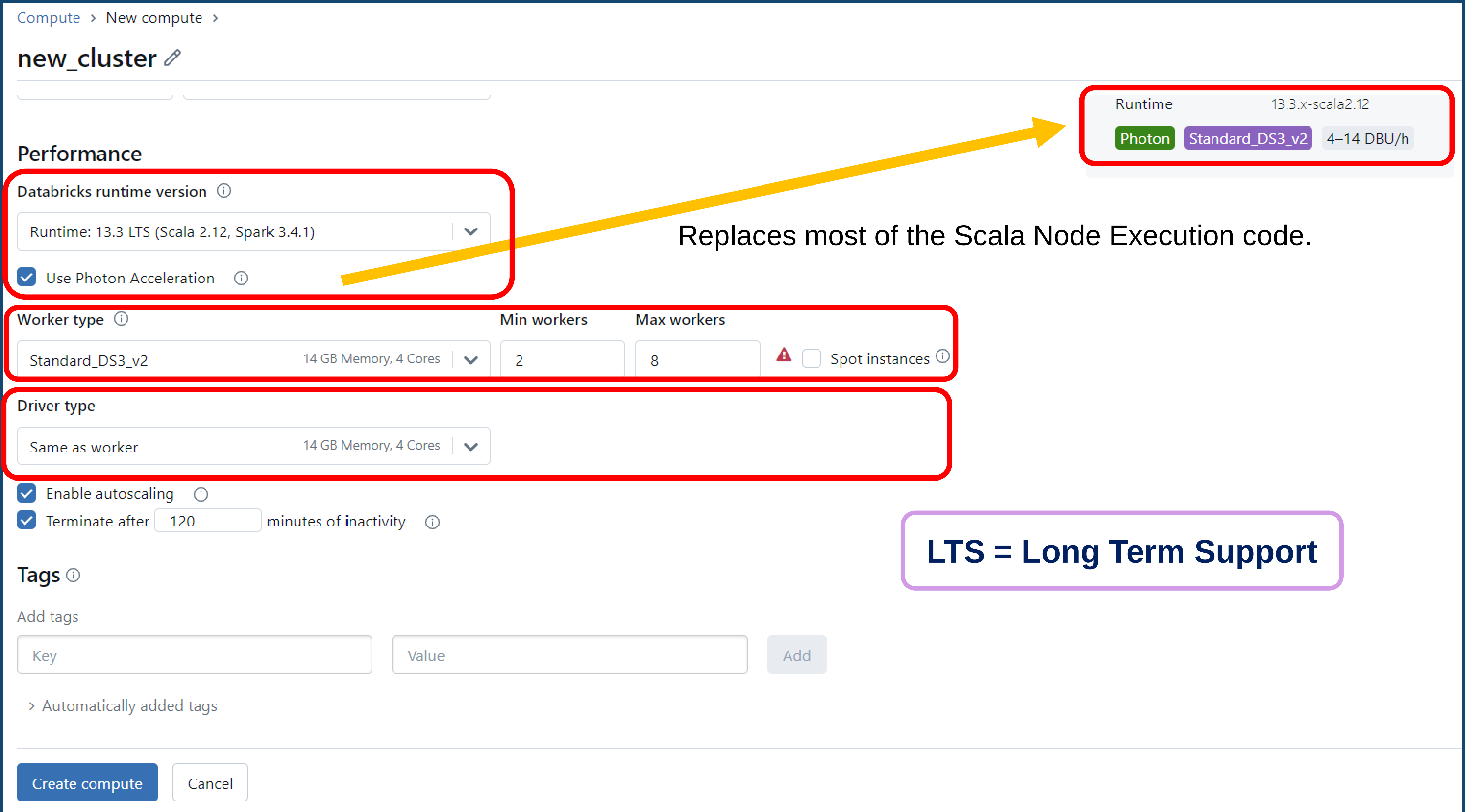
Environment variables

```
PYSPARK_PYTHON=/databricks/python3/bin/python3
```

Single User for Credential Pass Through.

Credential Passthrough Automatically Passes Your Credentials Through to Backend Resources like ADLS.

Cluster – Runtime and Node Types



Compute › New compute ›

new_cluster

Runtime 13.3.x-scala2.12

Photon | Standard_DS3_v2 | 4–14 DBU/h

Performance

Databricks runtime version ⓘ

Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

☑ Use Photon Acceleration ⓘ

Replaces most of the Scala Node Execution code.

Worker type ⓘ	Min workers	Max workers	
Standard_DS3_v2 14 GB Memory, 4 Cores	2	8	□ Spot instances ⓘ

Driver type

Same as worker 14 GB Memory, 4 Cores

☑ Enable autoscaling ⓘ

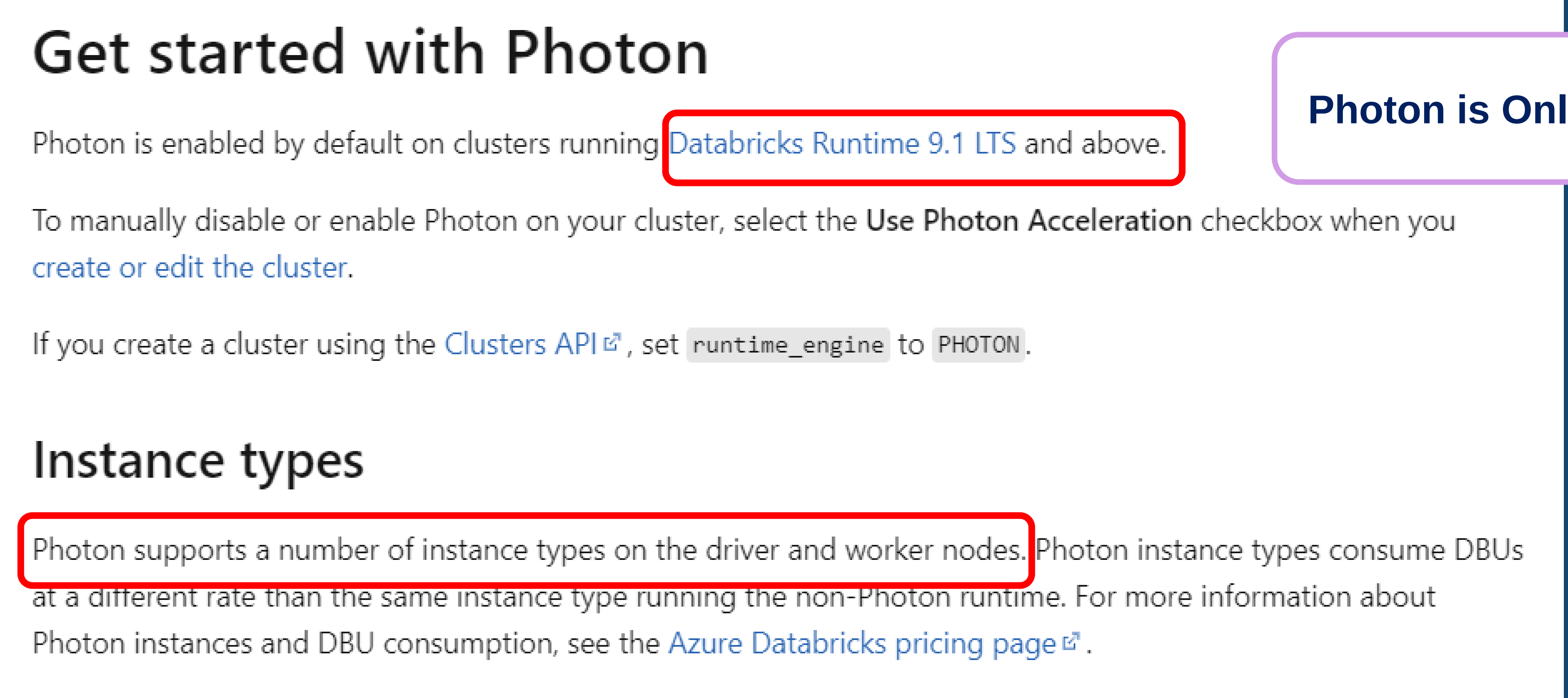
☑ Terminate after 120 minutes of inactivity ⓘ

LTS = Long Term Support

Tags ⓘ

Add tags

Photon – a New Execution Engine on Databricks



Get started with Photon

Photon is enabled by default on clusters running Databricks Runtime 9.1 LTS and above.

To manually disable or enable Photon on your cluster, select the **Use Photon Acceleration** checkbox when you create or edit the cluster.

If you create a cluster using the Clusters API, set `runtime_engine` to `PHOTON`.

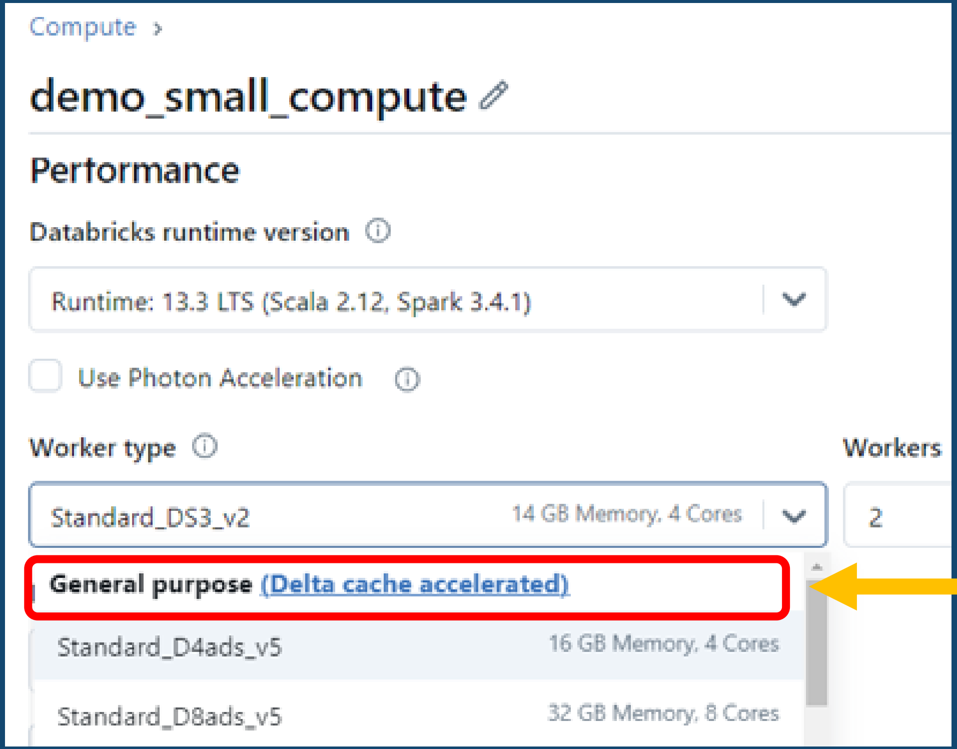
Instance types

Photon supports a number of instance types on the driver and worker nodes. Photon instance types consume DBUs at a different rate than the same instance type running the non-Photon runtime. For more information about Photon instances and DBU consumption, see the Azure Databricks pricing page.

Photon is Only Available on Databricks

https://learn.microsoft.com/en-us/azure/databricks/compute/photon

Cluster – Delta cache accelerated



Compute ›

demo_small_compute

Performance

Databricks runtime version

Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

□ Use Photon Acceleration

Worker type: Standard_DS3_v2 — 14 GB Memory, 4 Cores

Workers: 2

General purpose (Delta cache accelerated)

Standard_D4ads_v5	16 GB Memory, 4 Cores
Standard_D8ads_v5	32 GB Memory, 8 Cores

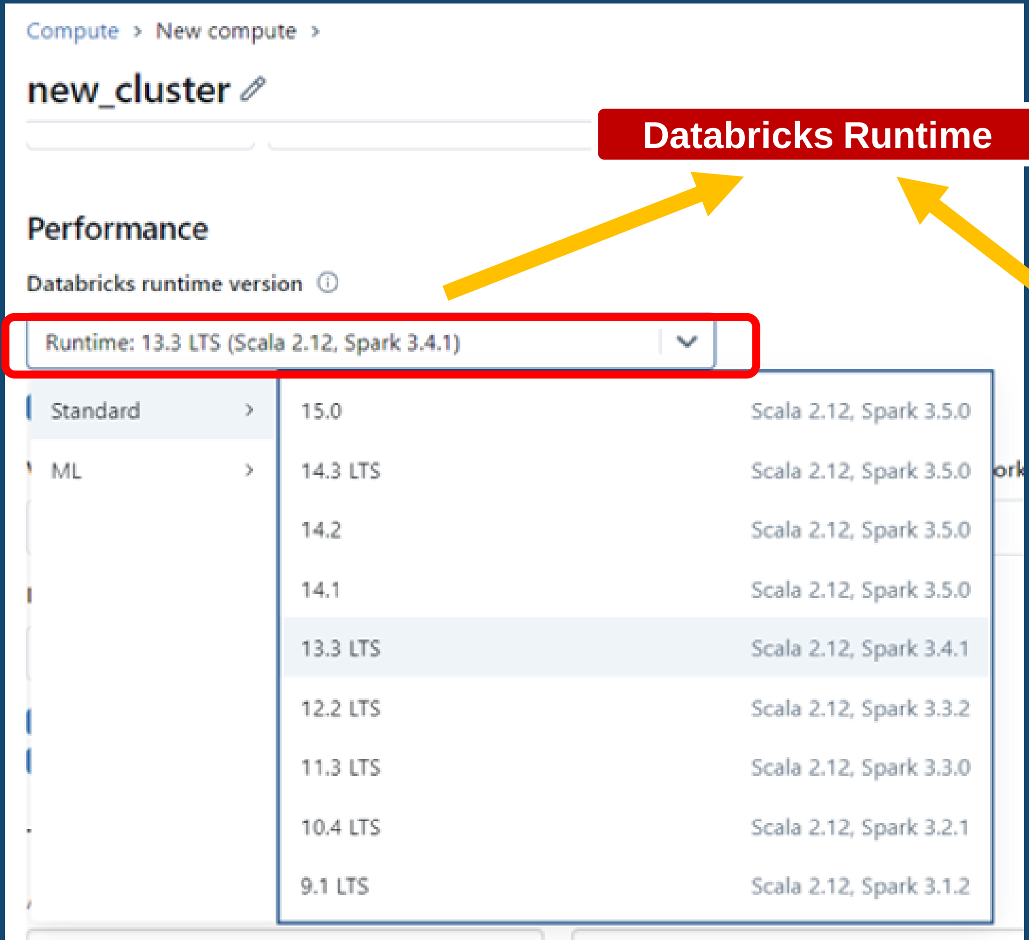
Improves Reading of Parquet and Delta Files.

Renamed to disk cache.

https://learn.microsoft.com/en-us/azure/databricks/optimizations/disk-cache

Cluster – Databricks Runtime

Standard



Compute › New compute ›

new_cluster

Performance

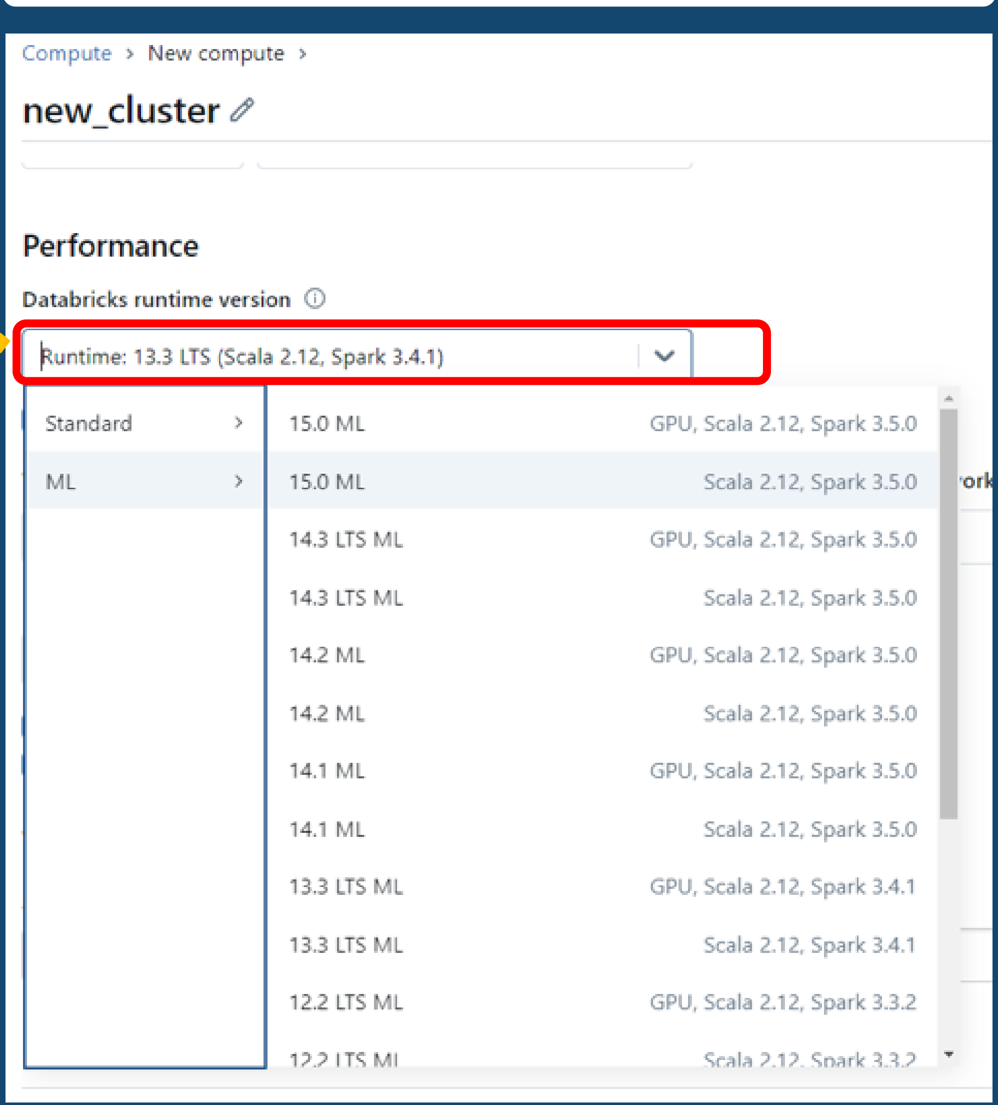
Databricks runtime version

Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

15.0	Scala 2.12, Spark 3.5.0
14.3 LTS	Scala 2.12, Spark 3.5.0
14.2	Scala 2.12, Spark 3.5.0
14.1	Scala 2.12, Spark 3.5.0
13.3 LTS	Scala 2.12, Spark 3.4.1
12.2 LTS	Scala 2.12, Spark 3.3.2
11.3 LTS	Scala 2.12, Spark 3.3.0
10.4 LTS	Scala 2.12, Spark 3.2.1
9.1 LTS	Scala 2.12, Spark 3.1.2

Databricks Runtime

Machine Learning



Compute › New compute ›

new_cluster

Performance

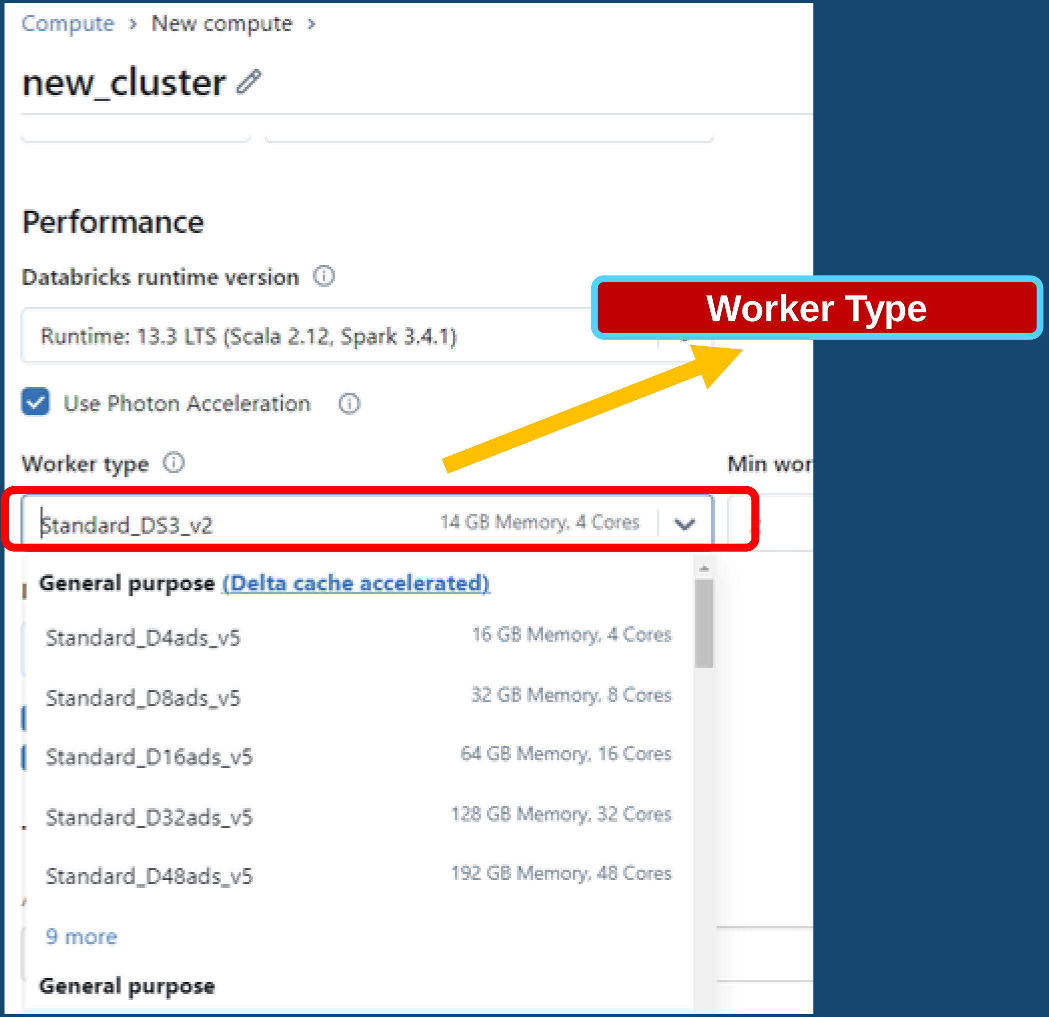
Databricks runtime version

Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

15.0 ML	GPU, Scala 2.12, Spark 3.5.0
15.0 ML	Scala 2.12, Spark 3.5.0
14.3 LTS ML	GPU, Scala 2.12, Spark 3.5.0
14.3 LTS ML	Scala 2.12, Spark 3.5.0
14.2 ML	GPU, Scala 2.12, Spark 3.5.0
14.2 ML	Scala 2.12, Spark 3.5.0
14.1 ML	GPU, Scala 2.12, Spark 3.5.0
14.1 ML	Scala 2.12, Spark 3.5.0
13.3 LTS ML	GPU, Scala 2.12, Spark 3.4.1
13.3 LTS ML	Scala 2.12, Spark 3.4.1
12.2 LTS ML	GPU, Scala 2.12, Spark 3.3.2
12.2 LTS ML	Scala 2.12, Spark 3.3.2

Cluster – Worker Type

Worker Type



Compute › New compute ›

new_cluster

Performance

Databricks runtime version

Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

☑ Use Photon Acceleration

Worker type

Min wor

Standard_DS3_v2 14 GB Memory, 4 Cores

General purpose (Delta cache accelerated)

Standard_D4ads_v5	16 GB Memory, 4 Cores
Standard_D8ads_v5	32 GB Memory, 8 Cores
Standard_D16ads_v5	64 GB Memory, 16 Cores
Standard_D32ads_v5	128 GB Memory, 32 Cores
Standard_D48ads_v5	192 GB Memory, 48 Cores

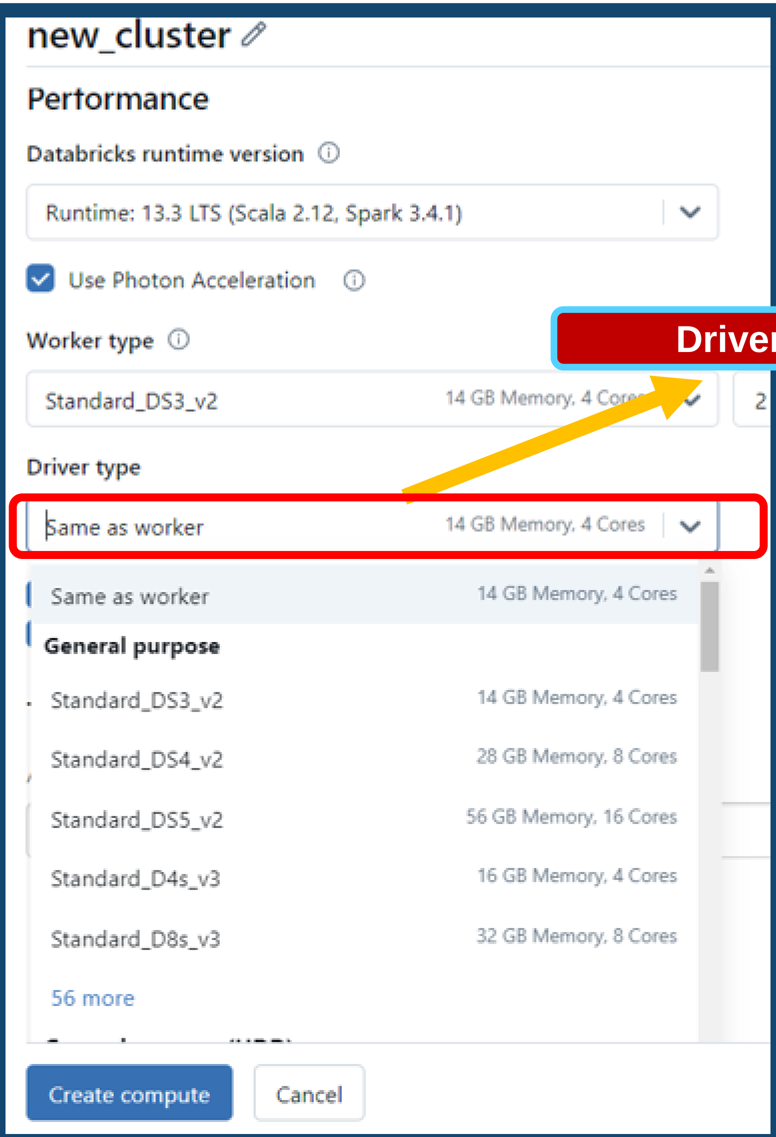
9 more

General purpose

Larger Workers means less nodes required.

Cluster – Driver Type

Driver Type



new_cluster

Performance

Databricks runtime version: Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

☑ Use Photon Acceleration

Worker type: Standard_DS3_v2 — 14 GB Memory, 4 Cores — 2

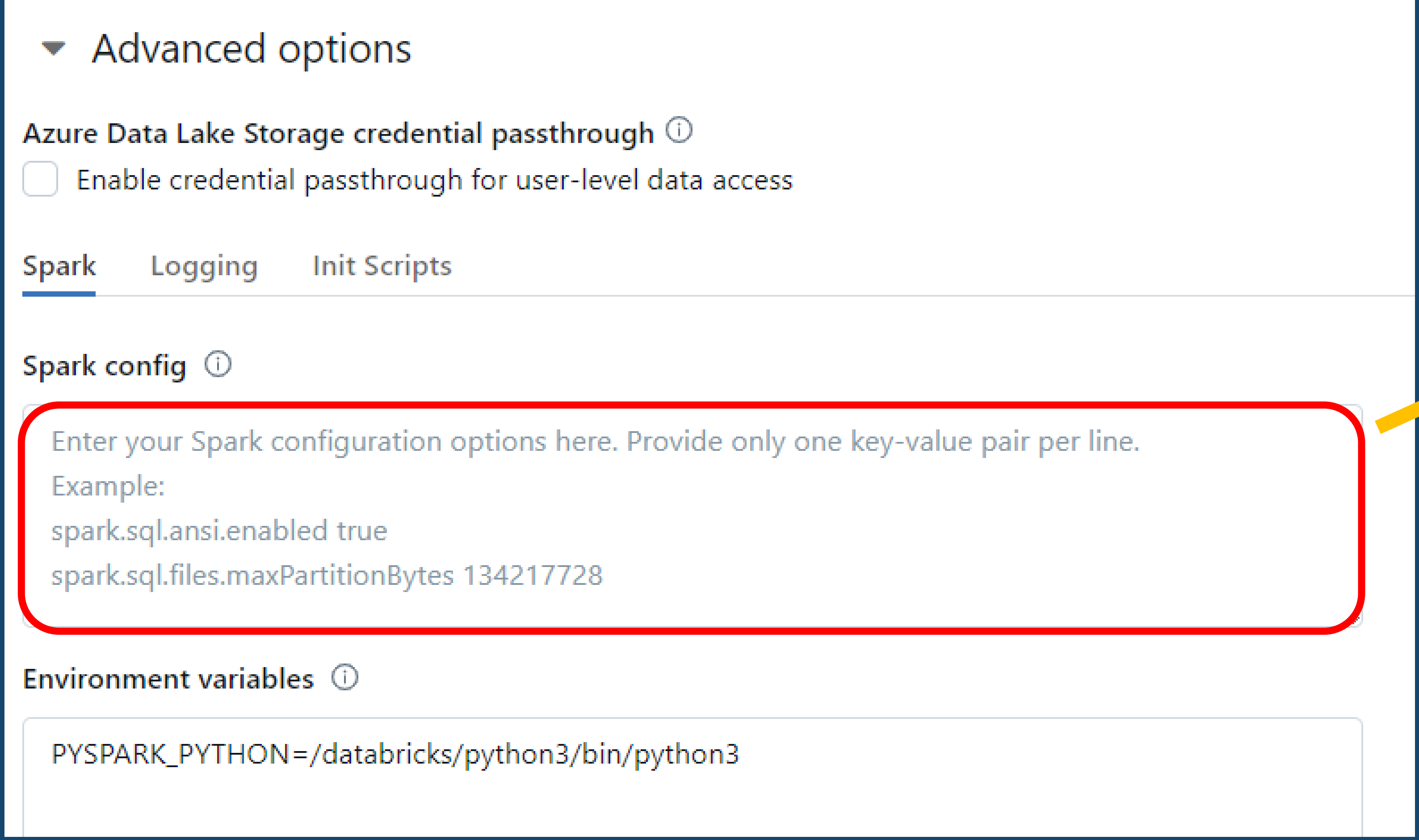
Driver type: Same as worker — 14 GB Memory, 4 Cores

Same as worker	14 GB Memory, 4 Cores
General purpose	
Standard_DS3_v2	14 GB Memory, 4 Cores
Standard_DS4_v2	28 GB Memory, 8 Cores
Standard_DS5_v2	56 GB Memory, 16 Cores
Standard_D4s_v3	16 GB Memory, 4 Cores
Standard_D8s_v3	32 GB Memory, 8 Cores
56 more	

Driver Type

Make this larger if you want to do a lot of work on the driver, i.e., collecting data, local processing.

Cluster – Advanced Settings



Advanced options

Azure Data Lake Storage credential passthrough

□ Enable credential passthrough for user-level data access

Spark config

```
Enter your Spark configuration options here. Provide only one key-value pair per line.
Example:
spark.sql.ansi.enabled true
spark.sql.files.maxPartitionBytes 134217728
```

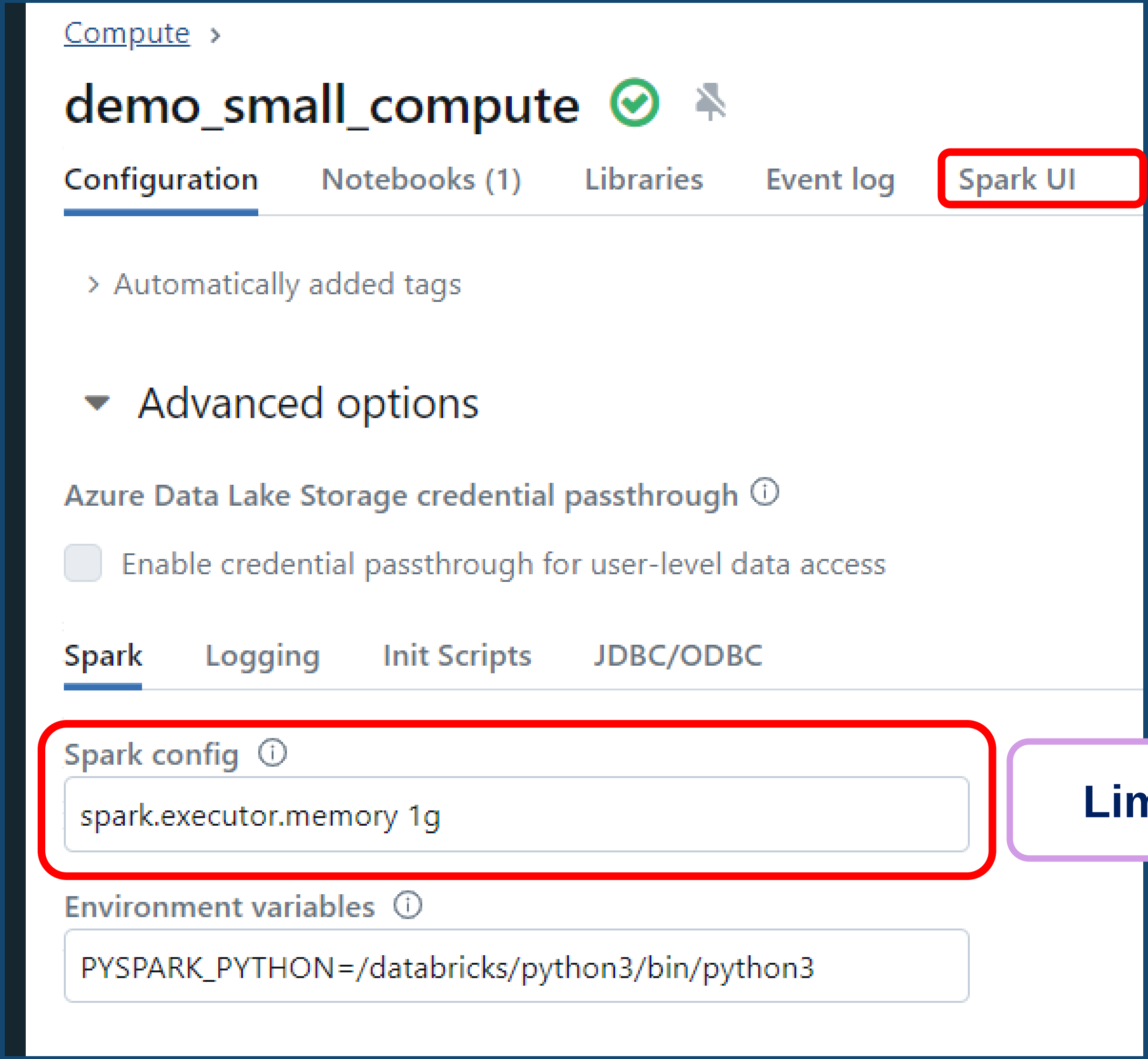
Environment variables

```
PYSPARK_PYTHON=/databricks/python3/bin/python3
```

Modify Spark Configuration Settings to Improve performance

Cluster – Shuffles & Spills

Getting to the Spark UI from your Notebook



Compute ›

demo_small_compute

› Automatically added tags

▾ Advanced options

Azure Data Lake Storage credential passthrough ⓘ

□ Enable credential passthrough for user-level data access

Spark config ⓘ

```
spark.executor.memory 1g
```

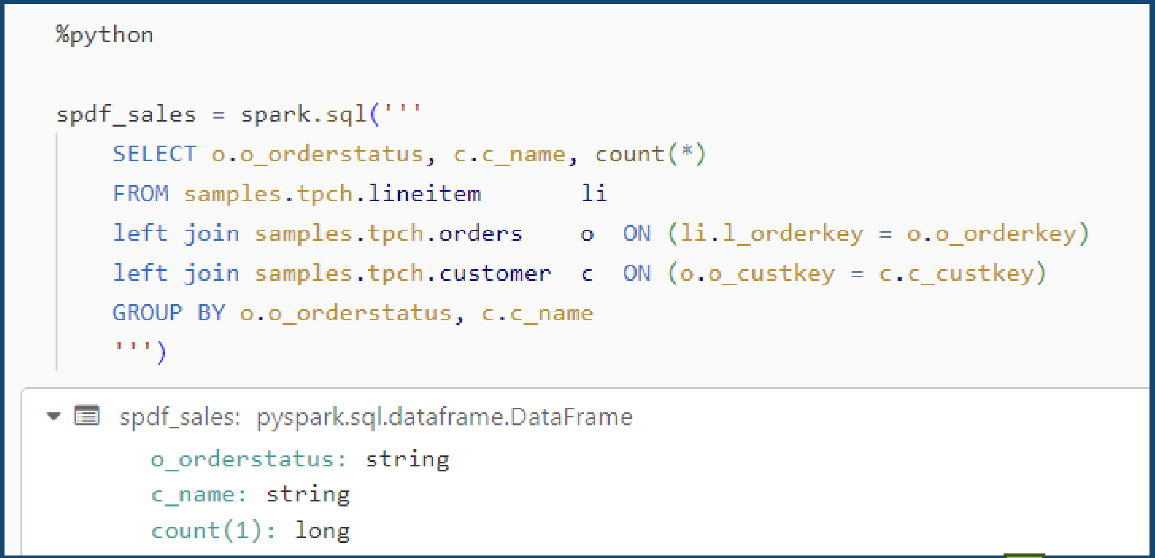
Environment variables ⓘ

```
PYSPARK_PYTHON=/databricks/python3/bin/python3
```

Click for the Spark UI

Limiting Memory to Force a Spill

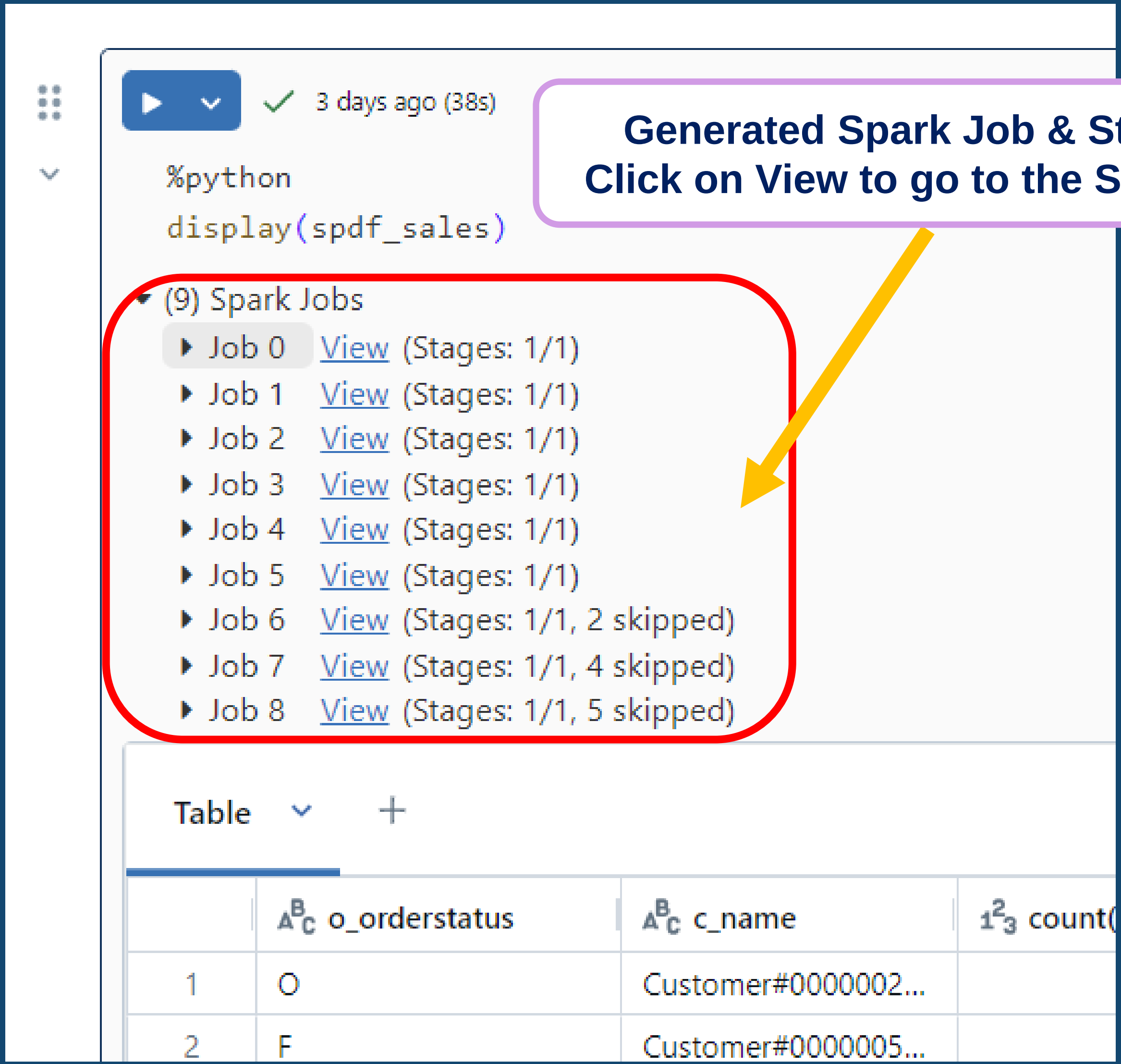
Getting to the Spark UI from your Notebook



```
%python

spdf_sales = spark.sql('''
    SELECT o.o_orderstatus, c.c_name, count(*)
    FROM samples.tpch.lineitem       li
    left join samples.tpch.orders     o  ON (li.l_orderkey = o.o_orderkey)
    left join samples.tpch.customer   c  ON (o.o_custkey = c.c_custkey)
    GROUP BY o.o_orderstatus, c.c_name
    ''')
```

```
▾ spdf_sales:  pyspark.sql.dataframe.DataFrame
      o_orderstatus:  string
      c_name:  string
      count(1):  long
```



✓ 3 days ago (38s)

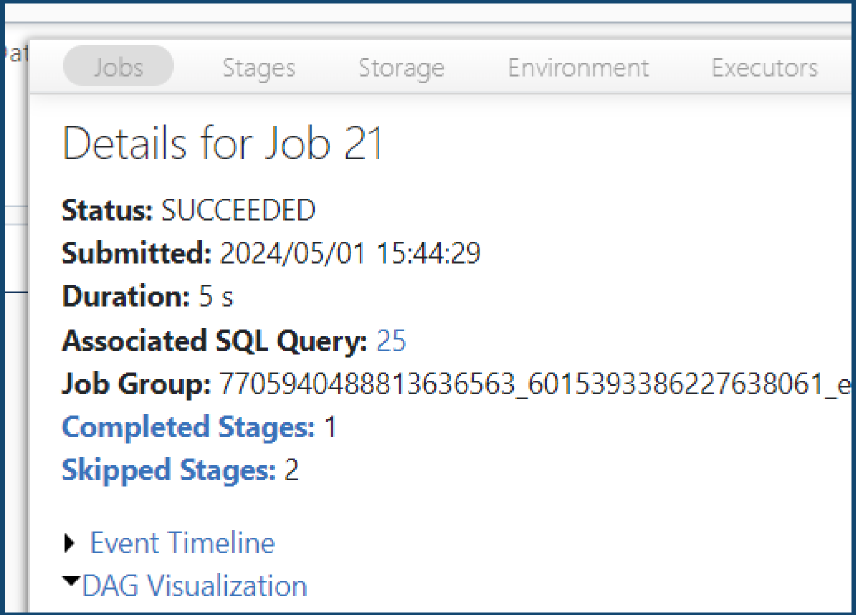
```
%python
display(spdf_sales)
```

```
▾ (9) Spark Jobs
   ▸ Job 0   View (Stages: 1/1)
   ▸ Job 1   View (Stages: 1/1)
   ▸ Job 2   View (Stages: 1/1)
   ▸ Job 3   View (Stages: 1/1)
   ▸ Job 4   View (Stages: 1/1)
   ▸ Job 5   View (Stages: 1/1)
   ▸ Job 6   View (Stages: 1/1, 2 skipped)
   ▸ Job 7   View (Stages: 1/1, 4 skipped)
   ▸ Job 8   View (Stages: 1/1, 5 skipped)
```

	o_orderstatus	c_name	count(
1	O	Customer#0000002...	
2	F	Customer#0000005...	

Generated Spark Job & Stages.
Click on View to go to the Spark UI.

Cluster – Shuffle & Spills

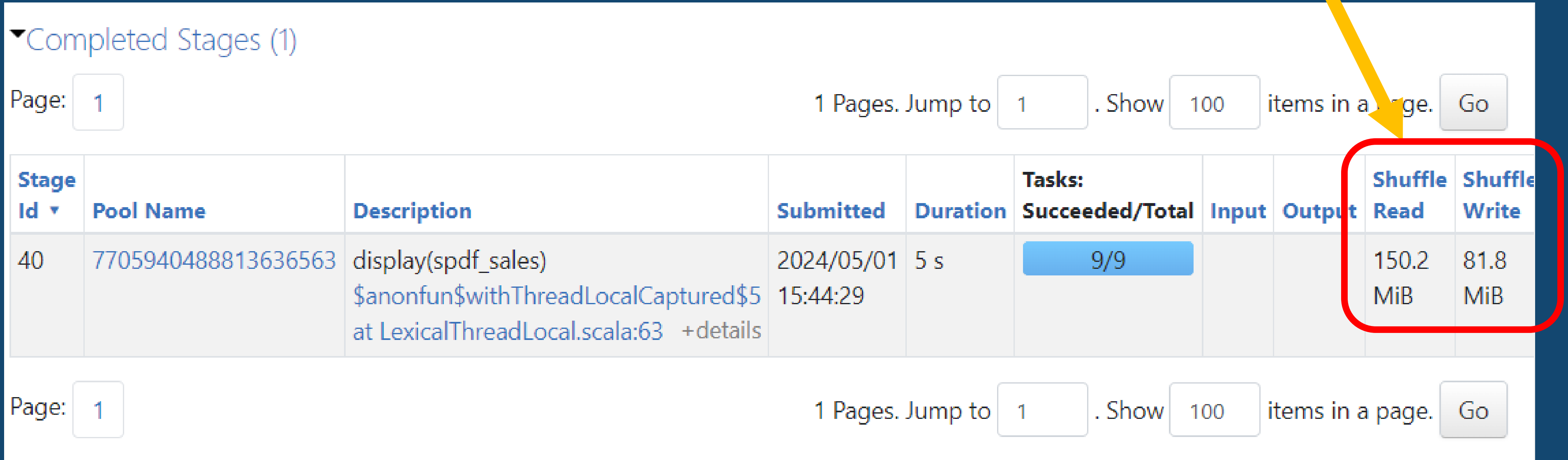


Details for Job 21

Status: SUCCEEDED
Submitted: 2024/05/01 15:44:29
Duration: 5 s
Associated SQL Query: 25
Job Group: 7705940488813636563_6015393386227638061_e
Completed Stages: 1
Skipped Stages: 2

▸ Event Timeline
▾DAG Visualization

Shuffle Read/Writes



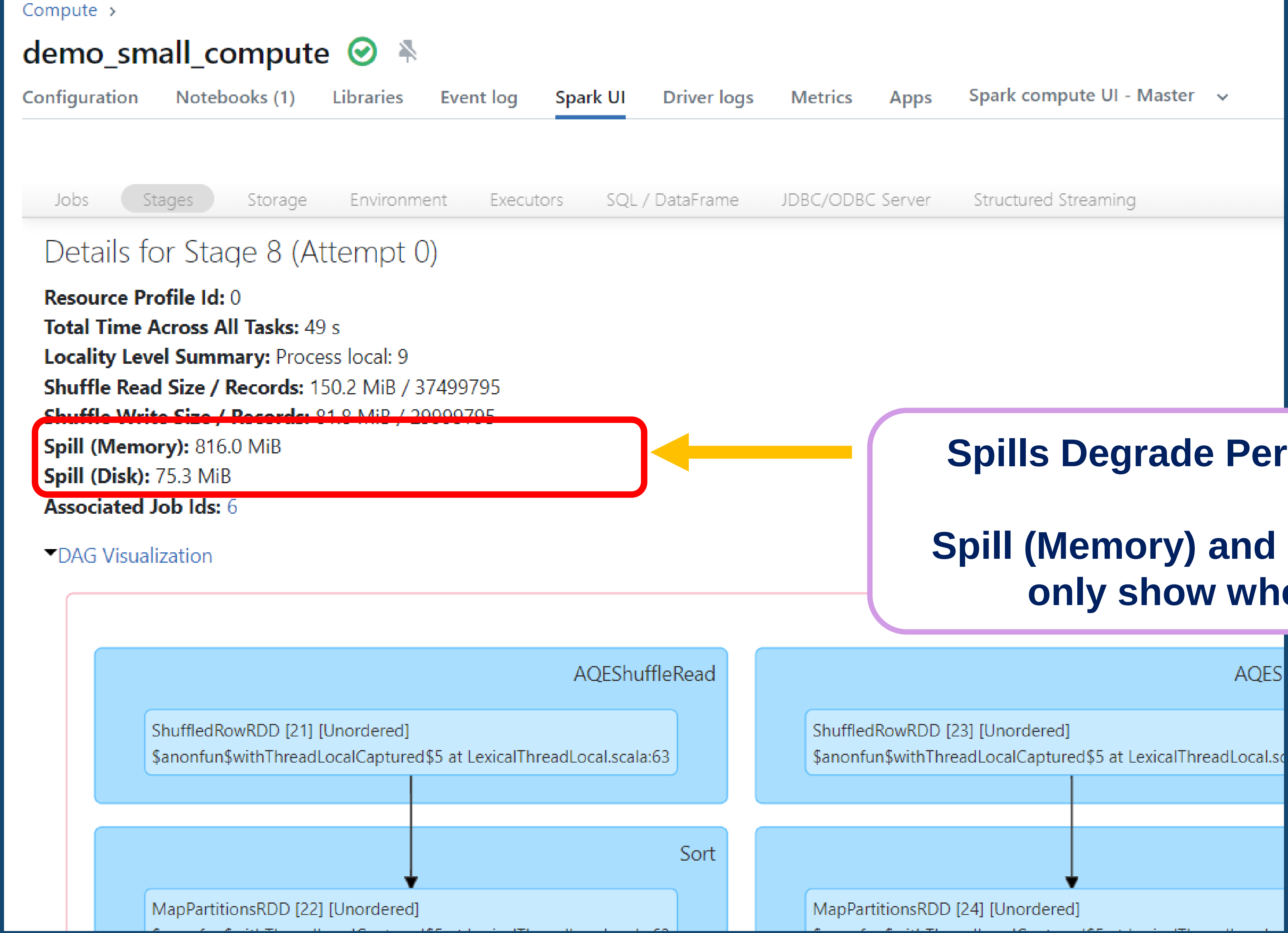
▾Completed Stages (1)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id ▾	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
40	7705940488813636563	display(spdf_sales) $anonfun$withThreadLocalCaptured$5 at LexicalThreadLocal.scala:63 +details	2024/05/01 15:44:29	5 s	9/9			150.2 MiB	81.8 MiB

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Cluster – Shuffle & Spills



Compute ›

demo_small_compute

Configuration
Notebooks (1)
Libraries
Event log
Spark UI
Driver logs
Metrics
Apps
Spark compute UI - Master
Jobs
Stages
Storage
Environment
Executors
SQL / DataFrame
JDBC/ODBC Server
Structured Streaming

Details for Stage 8 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 49 s
Locality Level Summary: Process local: 9
Shuffle Read Size / Records: 150.2 MiB / 37499795
Shuffle Write Size / Records: 81.8 MiB / 29999795
Spill (Memory): 816.0 MiB
Spill (Disk): 75.3 MiB
Associated Job Ids: 6

▾DAG Visualization

AQEShuffleRead
ShuffledRowRDD [21] [Unordered]
$anonfun$withThreadLocalCaptured$5 at LexicalThreadLocal.scala:63
Sort
MapPartitionsRDD [22] [Unordered]
AQES
ShuffledRowRDD [23] [Unordered]
$anonfun$withThreadLocalCaptured$5 at LexicalThreadLocal.sc
MapPartitionsRDD [24] [Unordered]

Spills Degrade Performance.

Spill (Memory) and Spill (Disk) only show when > 0

Wrapping Up

- **Why Compute Resources Matter**
- **Data Standard or Premium**
- **Databricks/Apache Spark Cluster Architecture**
- **Hardware Under the Cluster Architecture**
- **Optimizing the Cluster Configuration Step By Step**
- **Shuffles & Spills**

Thank You!