

# Accelerating CDE Jobs and Sessions using GPU

Date published: 2020-07-30

Date modified: 2023-11-02



# Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

|                                                                               |          |
|-------------------------------------------------------------------------------|----------|
| <b>Accelerating CDE Jobs and Sessions using GPUs (Technical Preview).....</b> | <b>4</b> |
| Accelerating CDE Jobs and Sessions.....                                       | 4        |
| Validating your Spark job or session run.....                                 | 4        |

## Accelerating CDE Jobs and Sessions using GPUs (Technical Preview)

CDE supports accelerating the Spark jobs and sessions using GPUs. You can optionally choose GPU acceleration using CDE UI or CLI for a Spark job and session. The supported Spark version is Spark 3.



**Note:** This feature is in Technical Preview and not recommended for production deployments. Cloudera recommends that you try this feature in test or development environments.

You can leverage the power of GPUs to benefit from the faster execution times and reduced infrastructure costs without changing the existing CDE application code. By enabling GPU support, data engineers can make use of GPU resources available to the CDE service. You can configure GPU resource quota per virtual cluster which can be requested for running the Spark job or session.

Before you use GPUs to accelerate CDE jobs and sessions, you must ensure that the following are performed:

1. Have nodes with GPU and met the [Software and Hardware requirements](#) before installing CDP Private Cloud Data Services.
2. [Set up GPU nodes](#).
3. [Test GPU node setup](#).
4. Set the [GPU resource quota](#) to allocate GPU resources effectively for CDE. GPU resources are limited in the cluster and usually shared among all data services.

## Accelerating CDE Jobs and Sessions

You can accelerate your Spark jobs and Sessions using GPUs after you add GPU resources to the CDE service and virtual cluster.

Perform the following to use GPU resources:

1. [Create a CDE service with GPU quota](#).
2. [Create a virtual cluster in that CDE service with GPU quota](#).
3. Enable GPU acceleration when creating Spark jobs using [CDE UI](#) or when running jobs using [CDE CLI](#).
4. Enable GPU acceleration when creating a session using [CDE UI](#) or [CDE CLI](#).
5. [Check whether the Spark job or session ran using the GPU resources](#).

## Validating your Spark job or session run

After you run a job that utilizes the GPUs, you can verify if the GPUs were used. If you run any SQL query, you will observe GPU Filters in the SQL query plan in the "History Server UI" or in the Driver logs.

### For Spark Job

You can check if the GPUs were utilized in the Job run using the SQL query.

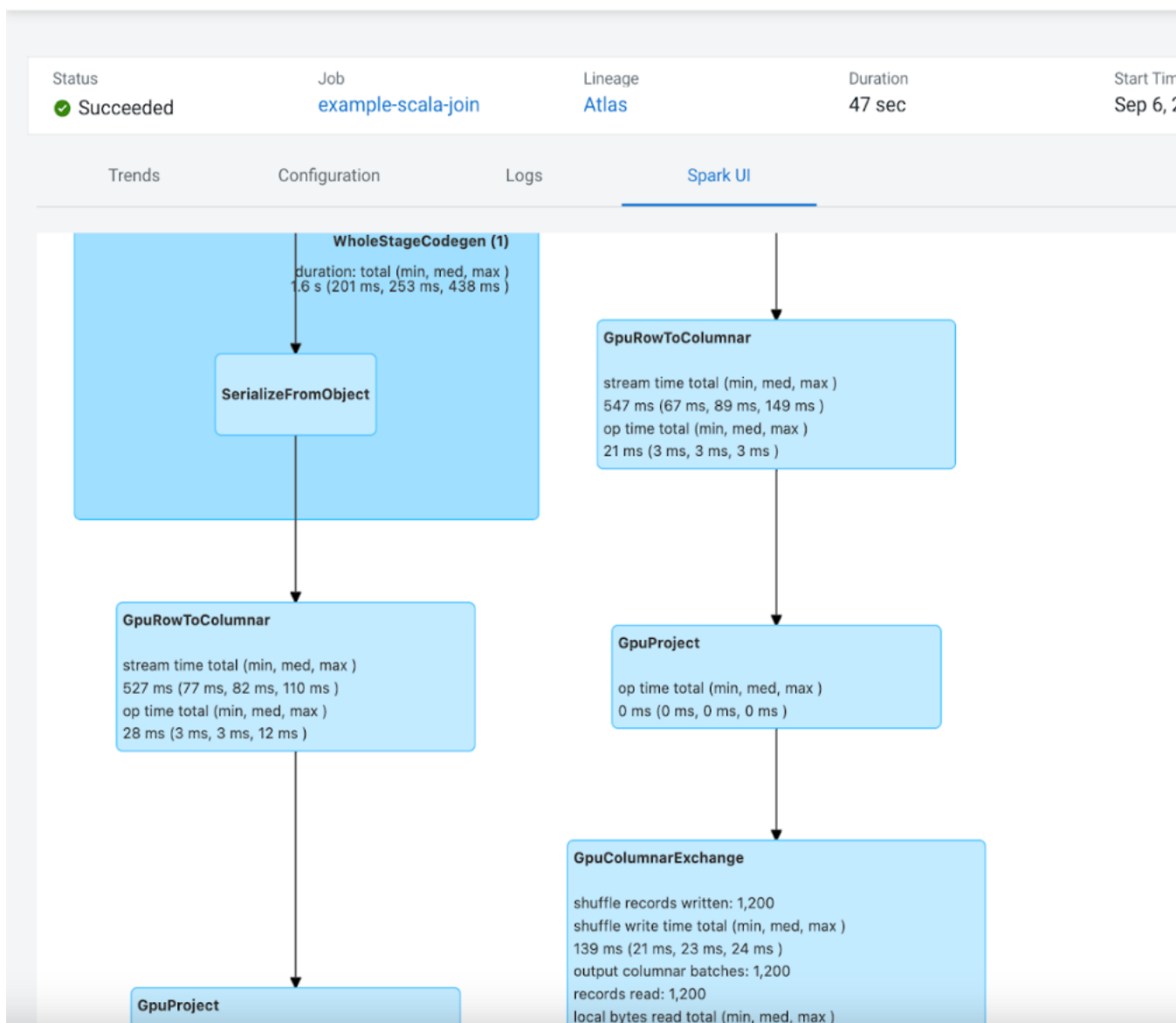
1. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE **Home** page displays.
2. Click **Jobs Runs** on the left navigation menu. The **Job Runs** page displays.
3. Click on the job **Run Id** of the job which utilized GPU.

4. Click the **Spark UI** tab and click SQL. The SQL plan is displayed which shows that the job has utilized the GPUs.



**Note:** This GPU details appears only if the job utilizes the GPU. Otherwise, job requests for GPU but never utilizes it.

GpuVC / Job Runs / 23



### For Sessions

You can check if the GPUs were utilized by creating a session with Scala type. Open a session terminal and execute the below sample Scala code.

```
val df = spark.sparkContext.parallelize(Seq(1)).toDF()
df.createOrReplaceTempView("df")
spark.sql("SELECT value FROM df WHERE value <=1").show
spark.sql("SELECT value FROM df WHERE value <=1").explain
```

This should output a SQL plan with GPU Filter as below.

Default-Virtual-Cluster-1722 / Sessions / 1

te: gpu

Kill

Delete

| STATUS               | TYPE  | CREATED BY   | START TIME               | TIMEOUT   |
|----------------------|-------|--------------|--------------------------|-----------|
| <div>Available</div> | Scala | demo@cdh1722 | Sep 26, 2023, 1:59:59 PM | 8 hour(s) |

Configuration

Interact

Logs

Connect

Help and Guides

```
val df = spark.sparkContext.parallelize(Seq(1)).toDF()
df.createOrReplaceTempView("df")
spark.sql("SELECT value FROM df WHERE value <>1").show
spark.sql("SELECT value FROM df WHERE value <>1").explain
```

Hide Output

```
df: org.apache.spark.sql.DataFrame = [value: int]
+-----+
|value|
+-----+
+-----+

== Physical Plan ==
GpuColumnarToRow false
+- GpuFilter NOT (value#2 = 1)
   +- GpuRowToColumnar targetsize(1073741824)
      +- *(1) SerializeFromObject [input[0, int, false] AS value#2]
         +- Scan[obj#1]
```