

Cloudera Runtime 7.1.8

CDS 3 Powered by Apache Spark

Date published: 2021-02-29

Date modified: 2024-06-06

CLOUdera

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2024. 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

CDS 3.3 Powered by Apache Spark Overview.....	5
CDS 3.3 Powered by Apache Spark Requirements.....	5
Installing CDS 3.3 Powered by Apache Spark.....	7
Enabling Spark rolling event log files in CDP.....	8
Enabling CDS 3.3 with GPU Support.....	9
Set up a Yarn role group to enable GPU usage.....	9
Configure NVIDIA RAPIDS Shuffle Manager.....	10
Updating Spark 2 applications for Spark 3.....	11
Running Applications with CDS 3.3 Powered by Apache Spark.....	11
The Spark 3 job commands.....	11
Canary test for pyspark3 command.....	11
Fetching Spark 3 Maven Dependencies.....	12
Accessing the Spark 3 History Server.....	12
Running applications using CDS 3.3 with GPU SupportCDS 3.3 with GPU Support.....	12
CDS 3.3 Powered by Apache Spark version and download information.....	15
Using the CDS 3.3 Powered by Apache Spark Maven Repository.....	16
CDS 3.3 Powered by Apache Spark Maven Artifacts.....	16
Cumulative hotfixes for CDS.....	17
Cumulative hotfix CDS 3.3.7180.11-1 (CDS 3.3 CHF1 for 7.1.8).....	17
Cumulative hotfix CDS 3.3.7180.12-5 (CDS 3.3 CHF2 for 7.1.8).....	18
Cumulative hotfix CDS 3.3.7180.14-1 (CDS 3.3 CHF3 for 7.1.8).....	19
Cumulative hotfix CDS 3.3.7180.15-1 (CDS 3.3 CHF4 for 7.1.8).....	20
Cumulative hotfix CDS 3.3.7180.16-1 (CDS 3.3 CHF5 for 7.1.8).....	20
Cumulative hotfix CDS 3.3.7180.17-1 (CDS 3.3 CHF6 for 7.1.8).....	21
Cumulative hotfix CDS 3.3.7180.18-1 (CDS 3.3 CHF7 for 7.1.8).....	22

Cumulative hotfix CDS 3.3.7180.19-1 (CDS 3.3 CHF8 for 7.1.8).....	22
Cumulative hotfix CDS 3.3.7180.20-1 (CDS 3.3 CHF9 for 7.1.8).....	23

CDS 3.3 Powered by Apache Spark Overview

Apache Spark is a general framework for distributed computing that offers high performance for both batch and interactive processing. It exposes APIs for Java, Python, and Scala. This document describes CDS 3.3 Powered by Apache Spark. CDS (Cloudera Distribution of Spark) enables you to install and evaluate the [features](#) of Apache Spark 3 without upgrading your CDP Private Cloud Base cluster.

For detailed API information, see the [Apache Spark project site](#).

CDS 3.3 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, distributed as a parcel and the Cloudera Service Descriptor file is available in Cloudera Manager for CDP 7.1.8.

On CDP Private Cloud Base, a Spark 3 service can coexist with the existing Spark 2 service. The configurations of the two services do not conflict and both services use the same YARN service. The port of the Spark History Server is 18088 for Spark 2 and 18089 for Spark 3.



Note: Spark 3.3 is the first Spark version with the log4j2 dependency. Previous versions contained the log4j1 dependency. If you are using any custom logging related changes, you must rewrite the original log4j properties' files using log4j2 syntax, that is, XML, JSON, YAML, or properties format.

CDS 3 for GPUs

CDS 3.3 with GPU Support is an add-on service that enables you to take advantage of the RAPIDS Accelerator for Apache Spark to accelerate Apache Spark 3 performance on existing CDP Private Cloud Base clusters.

Unsupported connectors

This release does not support the following connectors:

- SparkR
- Oozie
- Zeppelin

Unsupported Features

This release does not support the following feature:

- Hudi

Limitations of Spark in CDP

Limitations of Spark (in comparison to Apache Spark 3.3) in CDP are described below:

- spark.sql.orc.compression.codec config doesn't accept zstd value.
- spark.sql.avro.compression.codec config doesn't accept zstandard value.
- Specifying avroSchemaUrl is not supported in datasource options.
- Spark3 parcel is not available on the IBM PowerPC platform.

CDS 3.3 Powered by Apache Spark Requirements

The following sections describe software requirements for CDS 3.3 Powered by Apache Spark.

CDP Versions



Important: CDS 3.3 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Supported versions of CDP are described below.

CDS Powered by Apache Spark Version	Supported CDP Versions
3.3.0.3.3.7180.0-274	CDP Private Cloud Base with Cloudera Runtime 7.1.8

A Spark 2 service (included in CDP) can co-exist on the same cluster as Spark 3 (installed as a separate parcel). The two services are configured to not conflict, and both run on the same YARN service. Spark 3 installs and uses its own external shuffle service.

Although Spark 2 and Spark 3 can coexist in the same CDP Private Cloud Base cluster, you cannot use multiple Spark 3 versions simultaneously. All clusters managed by the same Cloudera Manager Server must use exactly the same version of CDS Powered by Apache Spark.

Software requirements

For CDS 3.3

Each cluster host must have the following software installed:

Java

JDK 8 or JDK 11. Cloudera recommends using JDK 8, as most testing has been done with JDK 8. Remove other JDK versions from all cluster and gateway hosts to ensure proper operation.

Python

Python 3.7 - 3.10

For CDS for GPUs

Each cluster host with a GPU must have the following software installed:

Java

JDK 8 or JDK 11. Cloudera recommends using JDK 8, as most testing has been done with JDK 8. Remove other JDK versions from all cluster and gateway hosts to ensure proper operation.

Python

Python 3.7 - 3.10

GPU drivers and CUDA toolkit

GPU driver v450.80.02 or higher

CUDA version 11.0 or higher

Download and install the [CUDA Toolkit](#) for your operating system. The toolkit installer also provides the option to install the GPU driver.

NVIDIA Library

NVIDIA RAPIDS version 22.06. For more information, see [NVIDIA Release Notes](#)

UCX (Optional)

Clusters with Infiniband or RoCE networking can leverage [Unified Communication X \(UCX\)](#) to enable the [RAPIDS Shuffle Manager](#). For information on UCX native libraries support, see [\(Optional\) Installing UCX native libraries](#).

Hardware requirements

For CDS 3.3

CDS 3.3 Powered by Apache Spark has no specific hardware requirements on top of what is required for Cloudera Runtime deployments.

For CDS for GPUs

CDS 3.3 with GPU Support requires cluster hosts with NVIDIA Pascal™ or better GPUs, with a [compute capability](#) rating of 6.0 or higher.

For more information, see [Getting Started](#) at the RAPIDS website.

Cloudera and NVIDIA recommend using NVIDIA-certified systems. For more information, see [NVIDIA-Certified Systems](#) in the NVIDIA GPU Cloud documentation.

Installing CDS 3.3 Powered by Apache Spark

CDS 3.3 Powered by Apache Spark (CDS 3.3) is distributed as a parcel. There are no external Custom Service Descriptors (CSD) for Livy for Spark3 or Spark3 using CDS 3.3 for CDP 7.1.8 parcel because they are already part of Cloudera Manager 7.7.1.



Note: Due to the potential for confusion between CDS Powered by Apache Spark and the initialism CSD, references to the custom service descriptor (CSD) file in this documentation use the term service descriptor.

Install CDS 3.3 Powered by Apache Spark



Note:

Although Spark 2 and Spark 3 can coexist in the same CDP Private Cloud Base cluster, you cannot use multiple Spark 3 versions simultaneously. All clusters managed by the same Cloudera Manager Server must use exactly the same version of CDS 3.3 Powered by Apache Spark.

Follow these steps to install CDS 3.3:

1. Check that all the software [prerequisites](#) are satisfied. If not, you might need to upgrade or install other software components first.
2. In the Cloudera Manager Admin Console, add the CDS parcel repository to the Remote Parcel Repository URLs in Parcel Settings as described in [Parcel Configuration Settings](#).



Note: If your Cloudera Manager Server does not have Internet access, you can use the CDS Powered by Apache Spark parcel files: put them into a [new parcel repository](#), and then configure the Cloudera Manager Server to target this newly created repository.

3. Download the CDS 3.3 parcel, distribute the parcel to the hosts in your cluster, and activate the parcel. For instructions, see [Managing Parcels](#).
4. Add the Spark 3 service to your cluster.
 - a. In step 1, select any optional dependencies, such as HBase and Hive, or select No Optional Dependencies.
 - b. In step 2, when customizing the role assignments, add a [gateway role](#) to every host.
 - c. On the Review Changes page, you can enable TLS for the Spark History Server.
 - d. Note that the History Server port is 18089 instead of the usual 18088.
 - e. Complete the remaining steps in the wizard.
5. Return to the Home page by clicking the Cloudera Manager logo in the upper left corner.
6. Click the stale configuration icon to launch the Stale Configuration wizard and restart the necessary services.

Install the Livy for Spark 3 Service

CDS 3 supports Apache Livy, but it cannot use the included Livy service, which is compatible with only Spark 2. To add and manage a Livy service compatible with Spark 3, you must install the Livy for Spark 3 service.

1. In the Cloudera Manager Admin Console, add the CDS 3 parcel repository to the Remote Parcel Repository URLs in Parcel Settings as described in [Parcel Configuration Settings](#).



Note: If your Cloudera Manager Server does not have Internet access, you can use the Livy parcel files: put them into a new parcel repository, and then configure the Cloudera Manager Server to target this newly created repository.

2. Download the CDS 3.3 parcel, distribute the parcel to the hosts in your cluster, and activate the parcel. For instructions, see [Managing Parcels](#).
3. Add the Livy for Spark 3 service to your cluster.
 - a. Note that the Livy port is 28998 instead of the usual 8998.
 - b. Complete the remaining steps in the wizard.
4. Return to the Home page by clicking the Cloudera Manager logo in the upper left corner.
5. Click the stale configuration icon to launch the Stale Configuration wizard and restart the necessary services.

If you want to activate the CDS 3.3 with GPU Support feature, [Set up a Yarn role group to enable GPU usage](#) on page 9 and optionally [Configure NVIDIA RAPIDS Shuffle Manager](#) on page 10

Enabling Spark rolling event log files in CDP

While running a long-running Spark application in CDP, (for example, a streaming application), the Spark Job generates a large single event log file until the Spark application is killed or stopped. A single event log file is not cost effective, requires high maintenance, and is resource-intensive. Therefore, to avoid creating a large event log file, you can use a rolling event log file.

About this task

If the Spark application is still running, only the `spark.history.fs.eventLog.rolling.maxFilesToRetain / spark.history.fs.eventLog.rolling.compaction.score.threshold` parameters are considered (the `spark.history.fs.cleaner.maxAge` parameter is not used in this case).

Procedure

1. Navigate to Cloudera Manager > Spark 3 > Configuration > Spark 3 Client Advanced Configuration Snippet (Safety Valve) for `spark3-conf/spark-defaults.conf` and set the following properties:

```
spark.eventLog.rolling.enabled=true
spark.eventLog.rolling.maxFileSize=128m
```

The default `spark.eventLog.rolling.maxFileSize` value is 128 MB. The minimum value allowed is 10 MB.

2. Set the maximum number of rolling event log files to retain:
 - a) Navigate to Cloudera Manager > Spark 3 > Configuration > History Server Advanced Configuration Snippet (Safety Valve) for `spark3-conf/spark-history-server.conf` and set the following property:


```
spark.history.fs.eventLog.rolling.maxFilesToRetain=2
```

By default, `spark.history.fs.eventLog.rolling.maxFilesToRetain` value is infinity, meaning all event log files are retained. Therefore, this parameter needs to be specified to avoid retaining more files. The minimum value allowed is 1.

3. Verify the output from the Spark history server event log directory. An example output is displayed below:

Results

```
[root@c3543-node4 ~]# sudo -u spark hdfs dfs -ls -R /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002
-rw-rw---- 3 spark spark 0 2023-01-04 07:03 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
appstatus_application_1672813574470_0002.inprogress -rw-
rw---- 3 spark spark 10485458 2023-01-04 07:05 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
events_1_application_1672813574470_0002 -rw-rw---- 3 spark
spark 0 2023-01-04 07:05 /user/spark/spark3ApplicationHistory/
eventlog_v2_application_1672813574470_0002/
events_2_application_1672813574470_0002 [root@c3543-
node4 ~]# sudo -u spark hdfs dfs -ls -R /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002
-rw-rw---- 3 spark spark 0 2023-01-04 07:03 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
appstatus_application_1672813574470_0002.inprogress -rw-
rw---- 3 spark spark 492014 2023-01-04 07:06 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
events_1_application_1672813574470_0002.compact -rw-
rw---- 3 spark spark 10489509 2023-01-04 07:06 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
events_2_application_1672813574470_0002 -rw-rw----
3 spark spark 227068 2023-01-04 07:06 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
events_3_application_1672813574470_0002 [root@c3543-
node4 ~]# sudo -u spark hdfs dfs -ls -R /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002
-rw-rw---- 3 spark spark 0 2023-01-04 07:03 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
appstatus_application_1672813574470_0002.inprogress -rw-
rw---- 3 spark spark 873356 2023-01-04 07:06 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
events_2_application_1672813574470_0002.compact -rw-
rw---- 3 spark spark 10484816 2023-01-04 07:06 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
events_3_application_1672813574470_0002 -rw-rw----
3 spark spark 339165 2023-01-04 07:06 /user/spark/
spark3ApplicationHistory/eventlog_v2_application_1672813574470_0002/
events_4_application_1672813574470_0002
```

Enabling CDS 3.3 with GPU Support

To activate the CDS 3.3 with GPU Support feature on suitable hardware, you need to create a Yarn role group and optionally make configuration changes to enable the NVIDIA RAPIDS Shuffle Manager.

Set up a Yarn role group to enable GPU usage

Create a Yarn role group so that you can selectively enable GPU usage for nodes with GPUs within your cluster.

Before you begin

[GPU scheduling and isolation](#) must be configured.

About this task

Enabling GPU on YARN through the Enable GPU Usage tickbox operates on cluster-level. Role groups in Yarn enable you to apply settings selectively, to a subset of nodes within your cluster.

Role groups are configured on the service level.

Procedure

1. In Cloudera Manager navigate to **Yarn Instances**.
2. Create a role group where you can add nodes with GPUs.
For more information, see [Creating a Role Group](#).
3. Move role instances with GPUs to the group you created.
On the Configuration tab select the source role group with the hosts you want to move, then click **Move Selected Instances To Group** and select the role group you created.
You may need to restart the cluster.
4. Enable GPU usage for the role group.
 - a) On the Configuration tab select **Categories GPU Management**.
 - b) Under GPU Usage click **Edit Individual Values** and select the role group you created.
 - c) Click **Save Changes**.

Configure NVIDIA RAPIDS Shuffle Manager

The NVIDIA RAPIDS Shuffle Manager is a custom ShuffleManager for Apache Spark that allows fast shuffle block transfers between GPUs in the same host (over PCIe or NVLink) and over the network to remote hosts (over RoCE or Infiniband).

About this task

NVIDIA RAPIDS Shuffle Manager has been shown to accelerate workloads where shuffle is the bottleneck when using the RAPIDS accelerator for Apache Spark. It accomplishes this by using a GPU shuffle cache for fast shuffle writes when shuffle blocks fit in GPU memory, avoiding the cost of writes to host using the built-in Spark Shuffle, a spill framework that will spill to host memory and disk on demand, and [Unified Communication X \(UCX\)](#) as its transport for fast network and peer-to-peer (GPU-to-GPU) transfers.

CDS 3.3 with GPU Support has built in support for UCX, no separate installation is required.

Cloudera and NVIDIA recommend using the RAPIDS shuffle manager for clusters with Infiniband or RoCE networking.

Procedure

1. Validate your UCX environment following the instructions provided in the NVIDIA [spark-rapids](#) documentation.
2. Before running applications with the RAPIDS Shuffle Manager, make the following configuration changes:

```
--conf "spark.shuffle.service.enabled=false" \  
--conf "spark.dynamicAllocation.enabled=false"
```

3. You are recommended to make the following UCX settings:

```
spark.executorEnv.UCX_TLS=cuda_copy,cuda_ipc,rc,tcp  
spark.executorEnv.UCX_RNDV_SCHEME=put_zcopy  
spark.executorEnv.UCX_MAX_RNDV_RAILS=1
```

```
spark.executorEnv.UCX_IB_RX_QUEUE_LEN=1024
```

For more information on environment variables, see the NVIDIA [spark-rapids](#) documentation.



Note: Running a job with the `--rapids-shuffle=true` flag does not affect these optional settings. You need to set them manually.

Updating Spark 2 applications for Spark 3

You must update your Apache Spark 2 applications to run on Spark 3. The Apache Spark documentation provides a migration guide.

For instructions on updating your Spark 2 applications for Spark 3, see the [migration guide](#) in the Apache Spark documentation.

Running Applications with CDS 3.3 Powered by Apache Spark

With CDS 3.3 Powered by Apache Spark (CDS 3.3), you can run Apache Spark 3 applications locally or distributed across a cluster, either by using an interactive shell or by submitting an application. Running Spark applications interactively is commonly performed during the data-exploration phase and for ad hoc analysis.

The Spark 3 job commands

With Spark 3, you use slightly different command names than with Spark 2, so that you can run both versions of Spark side-by-side without conflicts:

- `spark3-submit` instead of `spark-submit`.
- `spark3-shell` instead of `spark-shell`.
- `pyspark3` instead of `pyspark`.

For development and test purposes, you can also configure each host so that invoking the Spark 2 command name runs the corresponding Spark 3 executable.

Canary test for pyspark3 command

The following example shows a simple `pyspark3` session that refers to the `SparkContext`, calls the `collect()` function which runs a Spark 3 job, and writes data to HDFS. This sequence of operations helps to check if there are obvious configuration issues that prevent Spark 3 jobs from working at all. For the HDFS path for the output directory, substitute a path that exists on your own system.

```
$ hdfs dfs -mkdir /user/jdoe/spark
$ pyspark3
...
SparkSession available as 'spark'.
>>> strings = ["one","two","three"]
>>> s2 = sc.parallelize(strings)
>>> s3 = s2.map(lambda word: word.upper())
>>> s3.collect()
['ONE', 'TWO', 'THREE']
>>> s3.saveAsTextFile('hdfs:///user/jdoe/spark/canary_test')
>>> quit()
$ hdfs dfs -ls /user/jdoe/spark
```

```

Found 1 items
drwxr-xr-x  - jdoe spark-users  0 2016-08-26 14:41 /user/jdoe/spark/canary_
test
$ hdfs dfs -ls /user/jdoe/spark/canary_test
Found 3 items
-rw-r--r--  3 jdoe spark-users  0 2016-08-26 14:41 /user/jdoe/spark/cana
ry_test/_SUCCESS
-rw-r--r--  3 jdoe spark-users  4 2016-08-26 14:41 /user/jdoe/spark/canary
_test/part-00000
-rw-r--r--  3 jdoe spark-users 10 2016-08-26 14:41 /user/jdoe/spark/canary
_test/part-00001
$ hdfs dfs -cat /user/jdoe/spark/canary_test/part-00000
ONE
$ hdfs dfs -cat /user/jdoe/spark/canary_test/part-00001
TWO
THREE

```

Fetching Spark 3 Maven Dependencies

The Maven coordinates are a combination of groupId, artifactId and version. The groupId and artifactId are the same as for the upstream Apache Spark project. For example, for spark-core, groupId is org.apache.spark, and artifactId is spark-core_2.12, both the same as the upstream project. The version is different for the Cloudera packaging: see [Using the CDS 3 Powered by Apache Spark Maven Repository](#) for the exact name depending on which release you are using.

Accessing the Spark 3 History Server

The Spark 3 history server is available on port 18089, rather than port 18088 as with the Spark 2 history server.

Running applications using CDS 3.3 with GPU Support

1. Log on to the node where you want to run the job.
2. Run the following command to launch spark3-shell:

```

spark3-shell --conf "spark.rapids.sql.enabled=true" \
             --conf "spark.executor.memoryOverhead=5g"

```

where

--conf spark.rapids.sql.enabled=true

enables the following environment variables for GPUs:

```

"spark.task.resource.gpu.amount" - sets GPU resource amount per
task

"spark.rapids.sql.concurrentGpuTasks" - sets the number of con
current tasks per GPU
"spark.sql.files.maxPartitionBytes" - sets the input partition
size for DataSource API, The recommended value is "256m".
"spark.locality.wait" - controls how long Spark waits to obtain b
etter locality for tasks.
"spark.sql.adaptive.enabled" - enables Adaptive Query Execution.
"spark.rapids.memory.pinnedPool.size" - sets the amount of pinned
memory allocated per host.

```

```
"spark.sql.adaptive.advisoryPartitionSizeInBytes" - sets the advisory size in bytes of the shuffle partition during adaptive optimization.
```

For example,

```
$SPARK_HOME/bin/spark3-shell \
  --conf spark.task.resource.gpu.amount=2 \
  --conf spark.rapids.sql.concurrentGpuTasks=2 \
  --conf spark.sql.files.maxPartitionBytes=256m \
  --conf spark.locality.wait=0s \
  --conf spark.sql.adaptive.enabled=true \
  --conf spark.rapids.memory.pinnedPool.size=2G \
  --conf spark.sql.adaptive.advisoryPartitionSizeInBytes=1g
```

--conf "spark.executor.memoryOverhead=5g"

sets the amount of additional memory to be allocated per executor process



Note: cuDF uses a Just-In-Time (JIT) compilation approach for some kernels, and the JIT process can add a few seconds to query wall-clock time. You are recommended to set a JIT cache path to speed up subsequent invocations with: `--conf spark.executorEnv.LIBCUDF_KERNEL_CACHE_PATH=[local path]`. The path should be local to the executor (not HDFS) and not shared between different cluster users in a multi-tenant environment. For example, the path may be in `/tmp`: `(/tmp/cudf-[***USER**])`. If the `/tmp` directory is not writable, consult your system administrator to find a path that is.

You can override these configuration settings both from the command line and from code. For more information on environment variables, see the NVIDIA [spark-rapids](#) documentation and the [Spark SQL Performance Tuning Guide](#).

3. Run a job in spark3-shell.

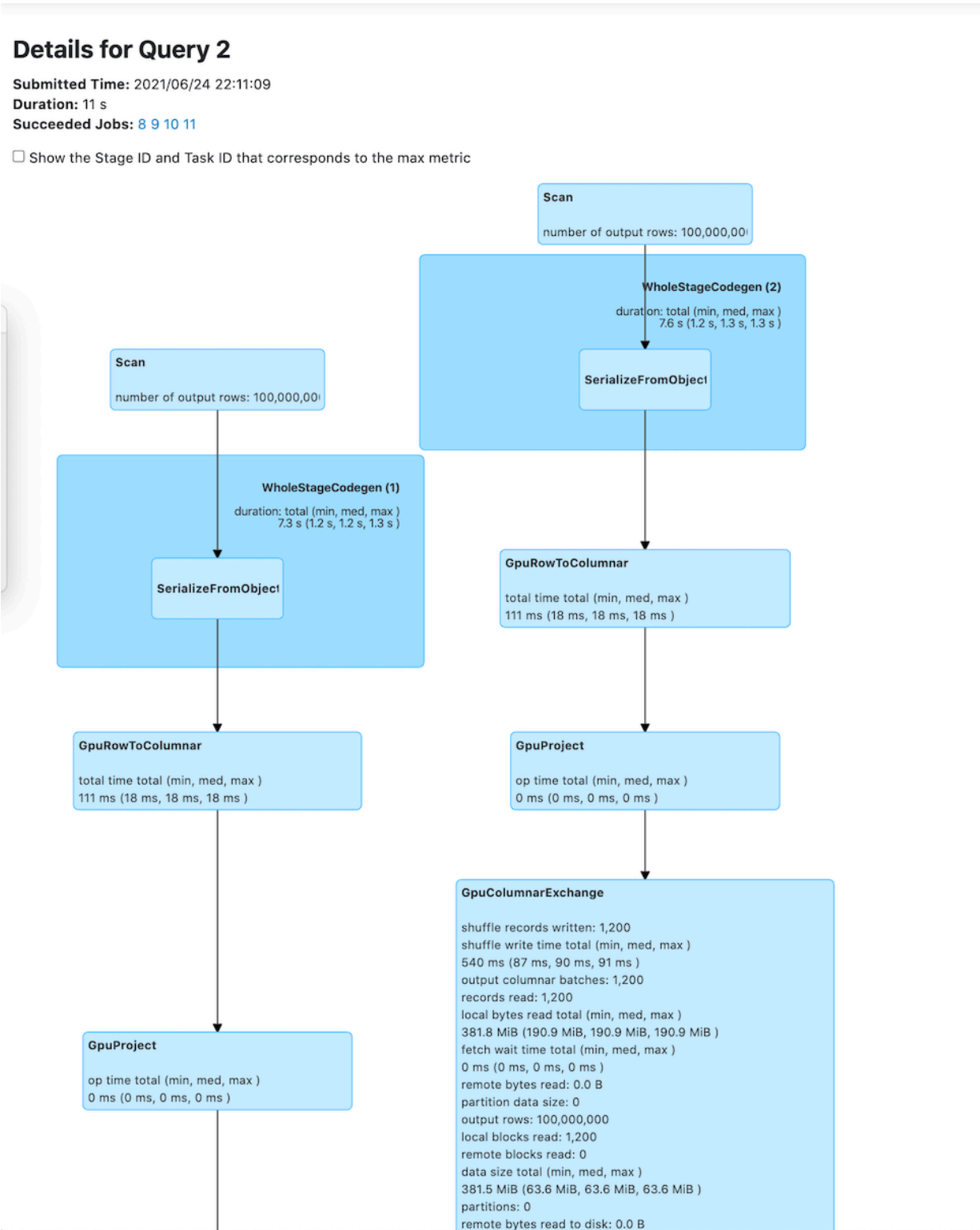
For example:

```
scala> val df = sc.makeRDD(1 to 100000000, 6).toDF
df: org.apache.spark.sql.DataFrame = [value: int]

scala> val df2 = sc.makeRDD(1 to 100000000, 6).toDF
df2: org.apache.spark.sql.DataFrame = [value: int]
scala> df.select($"value" as "a").join(df2select($"value" as "b"), $"a"
  === $"b").count
res0: Long = 100000000
```

4. You can verify that the job run used GPUs, by logging on to the Yarn UI v2 to review the execution plan and the performance of your spark3-shell application:

Select the Applications tab then select your [spark3-shell application]. Select ApplicationMaster SQL count at <console>:28 to see the execution plan.



Running a Spark job using CDS 3.3 with GPU Support with UCX enabled

1. Log on to the node where you want to run the job.
2. Run the following command to launch spark3-shell:

```
spark3-shell --conf "spark.rapids.sql.enabled=true" \
--conf "spark.executor.memoryOverhead=5g"
--rapids-shuffle=true
```

where

--rapids-shuffle=true

makes the following configuration changes for UCX:

```
spark.shuffle.manager=com.nvidia.spark.rapids.spark330cdh.Rapid
sShuffleManager
spark.executor.extraClassPath=/opt/cloudera/parcels/SPARK3/lib/s
park3/rapids-plugin/*
spark.executorEnv.UCX_ERROR_SIGNALS=
spark.executorEnv.UCX_MEMTYPE_CACHE=n
```

For more information on environment variables, see the NVIDIA [spark-rapids](#) documentation.

3. Run a job in spark3-shell.

CDS 3.3 Powered by Apache Spark version and download information

Cloudera Service Descriptors (CSD) file for CDS 3.3 is available in Cloudera Manager for CDP 7.1.8. The following section provides links to the parcel for both CDS 3.3 and CDS 3.3 with GPU Support.



Note:

The CDS version label is constructed in v.v.v.w.w.xxxx.y-z...z format and carries the following information:

v.v.v

Apache Spark upstream version, for example: 3.2.0

w.w

Cloudera internal version number, for example: 3.2

xxxx

CDP version number, for example: 7180 (referring to CDP Private Cloud Base 7.1.8)

y

maintenance version, for example, 1

z...z

build number, for example: 279

Table 1: Available CDS Versions

Version	Parcel Repository
3.3.0.3.3.7180.0-274	https://archive.cloudera.com/p/spark3/3.3.7180.0/parcels/

Using the CDS 3.3 Powered by Apache Spark Maven Repository



Important: CDS 3.3 Powered by Apache Spark (CDS 3.3) does not include an assembly JAR. When you build an application JAR, do not include Cloudera Runtime or CDS JARs, because they are already provided. If you do, upgrading Cloudera Runtime or CDS can break your application. To avoid this situation, set the Maven dependency scope to provided. If you have already built applications which include the Cloudera Runtime or CDS JARs, update the dependency to set scope to provided and recompile.

If you want to build applications or tools for use with CDS 3.3, and you are using Maven or Ivy for dependency management, you can pull the CDS artifacts from the Cloudera Maven repository. The repository is available at <https://repository.cloudera.com/artifactory/cloudera-repos/>.

The following is a sample POM (pom.xml) file:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <repositories>
    <repository>
      <id>cloudera</id>
      <url>https://repository.cloudera.com/artifactory/cloudera-repos/</url>
    </repository>
  </repositories>
</project>
```

CDS 3.3 Powered by Apache Spark Maven Artifacts

The following table lists the groupId, artifactId, and version required to access the artifacts for CDS 3.3 Powered by Apache Spark:

POM fragment

The following pom fragment shows how to access a CDS 3.3 artifact from a Maven POM.

```
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-core_2.12</artifactId>
  <version>3.3.0.3.3.7180.0-n</version>
  <scope>provided</scope>
</dependency>
```

The complete artifact list for this release follows.

List of CDS 3.3 Maven artifacts

Project	groupId	artifactId	version
Apache Spark	org.apache.spark	spark-assembly_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-avro_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-catalyst_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-core_2.12	3.3.0.3.3.7180.0-274

Project	groupId	artifactId	version
	org.apache.spark	spark-cypher_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-graph-api_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-graph_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-graphx_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-hadoop-cloud_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-hive-thriftserver_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-hive_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-kubernetes_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-kvstore_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-launcher_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-mllib-local_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-mllib_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-network-common_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-network-shuffle_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-network-yarn_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-parent_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-repl_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-sketch_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-sql-kafka-0-10_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-sql_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-streaming-kafka-0-10-assembly_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-streaming-kafka-0-10_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-streaming_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-tags_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-token-provider-kafka-0-10_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-unsafe_2.12	3.3.0.3.3.7180.0-274
	org.apache.spark	spark-yarn_2.12	3.3.0.3.3.7180.0-274

Cumulative hotfixes for CDS

You can review the list of cumulative hotfixes that were shipped for CDS.

Cumulative hotfix CDS 3.3.7180.11-1 (CDS 3.3 CHF1 for 7.1.8)

Know more about CDS 3.3 CHF1 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on March 28, 2023.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.0.3.3.7180.11-1-1.p0.39237531:

- CDPD-50603: Spark3 parcels required on SLES15
- CDPD-49113: Backport SPARK-41162: Anti-join must not be pushed below aggregation with ambiguous predicates
- CDPD-47685: Backport SPARK-40588: Fixed an issue that affected Spark 3
- CDPD-47467: Backport SPARK-41468: Fixed PlanExpression handling in EquivalentExpressions
- CDPD-47010: Supporting Parquet nanos timestamps in Spark SQL as long values
- CDPD-46731: Revert SPARK-38531 and backport SPARK-39612
- CDPD-45767: Spark3 - Upgraded Apache Commons Text to 1.10.0 due to CVE-2022-42889
- CDPD-45134: HWC - Cache disable configuration in secure access mode is not enforced at query-level
- CDPD-44184: HWC DIRECT_READER_V2 task fails with NPE on reading delete delta files
- CDPD-33992: HWC .table() needs to switch according to the specified read mode
- CDPD-28576: df.write using HWC throws an error if the order of DF columns is different than order of Hive table columns
- CDPD-21614: Spark SQL TRUNCATE table not permitted on external purge tables

Table 2: CDS cumulative hotfix 3.3.7180.11-1 download URL

Parcel Repository Location
<code>https://[username]:[password]@archive.cloudera.com/p/spark3/3.3.7180.11/parcels/</code>

Table 3: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.0.3.3.7180.11-1	Cloudera Runtime 7.1.8.15 (7.1.8 CHF4)	CDP Private Cloud Base with Cloudera Runtime 7.1.8

Cumulative hotfix CDS 3.3.7180.12-5 (CDS 3.3 CHF2 for 7.1.8)

Know more about CDS 3.3 CHF2 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on May 24, 2023.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8cd. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.2.3.3.7180.12-5-1.p0.41156209:

- CDPD-56193: [livy3] TypeError: required field type_ignores missing from Module
- CDPD-56187: Backport LIVY-783 to CDS 3.x
- CDPD-55155: Backport SPARK-43124 - Dataset.show must not trigger job execution on CommandResults
- CDPD-55150: Backport LIVY-763: FD leak in InteractiveSession and PythonInterpreter
- CDPD-55116: Fixed Spark and Livy vulnerability CVE-2023-22946
- CDPD-53718: Upgraded Spark 3 to Apache Spark 3.3.2
- CDPD-49526: Fixed Spark 3 UTs after Hadoop upgrade from log4j1 to reload4j
- CDPD-49294: Backport latest subquery merge related bugfixes
- CDPD-47452: Adjust tests similarly to SPARK-40851 as TimestampFormatter behavior changed when using the latest Java 8/11/17
- CDPD-46774: Enable batch mode for the Spark Atlas connector unit tests, do not use the Apache maven repo
- CDPD-46665: Livy - Upgraded commons-codec to 1.13 or higher

- CDPD-46542: Fixed the livy-for-spark3 gerrit unit test job
- CDPD-46541: Fixed the spark-atlas-connector-for-spark3 gerrit unit test job
- CDPD-46322: Update logredactor to 2.0.14 due to CVEs in jackson-databind
- CDPD-46306: CVE-2022-31777: Apache Spark XSS vulnerability in log viewer UI Javascript
- CDPD-45599: Livy for Spark 3 pre-commit unit test failures after recent Spark, log4j fixes
- CDPD-45570: Backport SPARK-35084
- CDPD-45302: spark_atlas_connector for Spark3 unit tests failures
- CDPD-45138: Install Failure - DeployClusterCC command failed during first run
- CDPD-45052: Add generic link hive-warehouse-connector-assembly.jar
- CDPD-44794: Livy3 - Upgrade Commons IO to 2.11.0/20030203.000550 due to medium CVEs
- CDPD-42384: Spark Atlas Connector - Upgrade Data Mapper for Jackson to 1.9.16-TALEND due to high CVEs
- CDPD-40882: Livy - Upgrade jersey to 2.35 / 3.0.2 due to CVE-2021-28168
- CDPD-8152: SAC : Tables created using spark-submit are created as both hive_table and spark_table entities

Table 4: CDS cumulative hotfix 3.3.7180.12-5 download URL

Parcel Repository Location
<code>https://[username]:[password]@archive.cloudera.com/p/spark3/3.3.7180.12/parcels/</code>

Table 5: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.2.3.3.7180.12-5	Cloudera Runtime 7.1.8.32 (7.1.8 CHF7)	CDP Private Cloud Base with Cloudera Runtime 7.1.8

Cumulative hotfix CDS 3.3.7180.14-1 (CDS 3.3 CHF3 for 7.1.8)

Know more about CDS 3.3 CHF3 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on August 17, 2023.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.2.3.3.7180.14-1-1.p0.44234236:

- CDPD-60192: [CDS 3.x] Unable to load data with array type through Hive warehouse connector
- CDPD-60190: Backport SPARK-39441: Speed up the Analyzer rule DeduplicateRelations
- CDPD-59381: [CDS 3.x] The spark.kerberos.keytab parameter handling is missing from AtlasDelegationTokenProvider.scala
- CDPD-59293: [7.1.x, CDS 3.x] Spark Atlas Connector - Upgrade jettison to 1.5.4 due to CVE-2023-1436, CVE-2022-40149, CVE-2022-40150, CVE-2022-45685 and CVE-2022-45693
- CDPD-59247: [7.1.x, CDS 3.x] Spark - Upgrade snappy-java to 1.1.10.1 due to CVE-2023-34453, CVE-2023-34454 and CVE-2023-34455
- CDPD-58273: [7.1.x, CDS 3.x] Backport SPARK-43470: Include the operating system and Java version, python version information in the Application log

Table 6: CDS cumulative hotfix 3.3.7180.14-1 download URL

Parcel Repository Location
<code>https://[username]:[password]@archive.cloudera.com/p/spark3/3.3.7180.14/parcels/</code>

Table 7: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.2.3.3.7180.14-1	Cloudera Runtime 7.1.8.44 (7.1.8 CHF12)	CDP Private Cloud Base with Cloudera Runtime 7.1.9

Cumulative hotfix CDS 3.3.7180.15-1 (CDS 3.3 CHF4 for 7.1.8)

Know more about CDS 3.3 CHF4 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on September 27, 2023.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for 3.3.2.3.3.7180.15-1-1.p0.45461358:

- CDPD-61357: The Atlas lineage is missing in case of the following Spark CLI parameters: --principal test@ROOT.HWX.SITE --keytab test.keytab
- CDPD-61021: Spark3: When saving Hive table, add configuration for disabling fallback and saving in Spark specific format, results in error
- CDPD-60501: Spark3 - Update pre_commit_hook to use JDK8 version to u371
- CDPD-57831: Update the application tag generation logic in Livy

Table 8: CDS cumulative hotfix 3.3.7180.15-1 download URL

Parcel Repository Location
<code>https://[username]:[password]@archive.cloudera.com/p/spark3/3.3.7180.15/parcels/</code>

Table 9: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.2.3.3.7180.15-1	Cloudera Runtime 7.1.8.46 (7.1.8 CHF14)	CDP Private Cloud Base with Cloudera Runtime 7.1.9

Cumulative hotfix CDS 3.3.7180.16-1 (CDS 3.3 CHF5 for 7.1.8)

Know more about CDS 3.3 CHF5 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on October 30, 2023.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.2.3.3.7180.16-1-1.p0.46608914:

- CDPD-61888: Fixed an issue with certain Kerberos implementations where the renew time in renewable tickets showed null
- CDPD-61564: Upgraded datanucleus-core dependency to 5.2.10

Table 10: CDS cumulative hotfix 3.3.7180.16-1 download URL

Parcel Repository Location
<code>https://[username]:[password]@archive.cloudera.com/p/spark3/3.3.7180.16/parcels/</code>

Table 11: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.2.3.3.7180.16-1	Cloudera Runtime 7.1.8.48 (7.1.8 CHF15)	CDP Private Cloud Base with Cloudera Runtime 7.1.8

Cumulative hotfix CDS 3.3.7180.17-1 (CDS 3.3 CHF6 for 7.1.8)

Know more about CDS 3.3 CHF6 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on February 29, 2024.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.2.3.3.7180.17-1-1.p0.50772350:

- CDPD-66981: [CDS 3.x] Broken lineage on Atlas when calling .cache() or .persist() method on a Spark DataFrame
- CDPD-66915: Livy3 server logs are missing on 7.1.9 FIPS clusters
- CDPD-65588: [CDS 3.x] Exclude log4j dependencies from spark-atlas-connector assembly
- CDPD-65584: [CDS 3.x] Spark - Upgrade Apache Derby to 10.17.1.0 due to CVE-2022-46337
- CDPD-64135: [7.1.7 SP2, CDS 3.x] Backport HBASE-27624: Cannot specify Namespace through the hbase.table option in Spark Connector
- CDPD-63799: [7.1.9, CDS 3.x] Livy - Upgrade snakeyaml to 1.33 due to high CVEs
- CDPD-63725: HWC - Unhandled VarcharType(100) in Pyspark3 shell
- CDPD-61742: Test failure: org.apache.spark.sql.hive.execution.HiveTableScanSuite.Spark-4077: timestamp query for null value
- CDPD-55423: Added a new Livy configuration property livy.server.send-server-version. By default this property is set as false. Set to true to send the server version in Cloudera Manager
- CDPD-46689: HWC - DIRECT_READER_V2 must handle delete delta files from delete & update queries
- CDPD-45380: Spark - Upgrade snakeyaml to 1.32 due to high CVEs
- CDPD-44220: Fixed issues with Livy session recovery/HA failover on FIPS clusters
- CDPD-43553: Spark - Upgrade jersey's jersey to 2.36/3.0.5 due to medium CVEs
- CDPD-40362: Spark - Support changing session catalog's default database

Table 12: CDS cumulative hotfix 3.3.7180.17-1 download URL

Parcel Repository Location
<code>https://[username]:[password]@archive.cloudera.com/p/spark3/3.3.7180.17/parcels/</code>

Table 13: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.2.3.3.7180.17-1	Cloudera Runtime 7.1.8.56 (7.1.8 CHF19)	CDP Private Cloud Base with Cloudera Runtime 7.1.8

Cumulative hotfix CDS 3.3.7180.18-1 (CDS 3.3 CHF7 for 7.1.8)

Know more about CDS 3.3 CHF7 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on May 30, 2024.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.2.3.3.7180.18-1-1.p0.53825331:

- CDPD-69324: Backport SPARK-46779 and SPARK-47955
- CDPD-67561: Backport SPARK-47319 to improve QueryPlan.missingInput calculation
- CDPD-67338: Handle the ClassCastException of CDPD-40874 in the HWC layer
- CDPD-67336: Revert the Spark change done as part of CDPD-40874, to add Identifier field
- CDPD-66940: Timezone value not getting updated in Livy 3
- CDPD-64801: Livy - Upgrade datatables to 1.10.23+ due to CVE-2020-28458
- CDPD-60979: Spark3 - Upgrade Apache Ivy to 2.5.2 due to CVE-2022-46751
- CDPD-58846: Spark3 - Upgrade Janino to 3.1.10 due to CVE-2023-33546
- CDPD-58844: Spark - Upgrade Janino to 3.1.10 due to CVE-2023-33546

Table 14: CDS cumulative hotfix 3.3.7180.18-1 download URL

Parcel Repository Location
<code>https://[username]:[password]@archive.cloudera.com/p/spark3/3.3.7180.18/parcels/</code>

Table 15: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.2.3.3.7180.18-1	Cloudera Runtime 7.1.8.60 (7.1.8 CHF23)	CDP Private Cloud Base with Cloudera Runtime 7.1.8

Cumulative hotfix CDS 3.3.7180.19-1 (CDS 3.3 CHF8 for 7.1.8)

Know more about CDS 3.3 CHF8 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on June 17, 2024.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.2.3.3.7180.19-1-1.p0.54387814:

- CDPD-69805: Spark - OpenCSVSerde treats blank value as null
- CDPD-61470: Spark - Do not call HMS to get list of pruned partitions when translated filter is empty
- CDPD-60845: Unable to write data to the non-default database using HWC.
- CDPD-47129: Spark - Handle empty CSV fields via OpenCSVSerde

Table 16: CDS cumulative hotfix 3.3.7180.19-1 download URL

Parcel Repository Location
<code>https://[***USERNAME***]:[***PASSWORD***]@archive.cloudera.com/p/spark3/3.3.7180.19/parcels/</code>

Table 17: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.7180.19-1	Cloudera Runtime 7.1.8.61 (7.1.8 CHF24)	CDP Private Cloud Base with Cloudera Runtime 7.1.8

Cumulative hotfix CDS 3.3.7180.20-1 (CDS 3.3 CHF9 for 7.1.8)

Know more about CDS 3.3 CHF9 for Cloudera Runtime 7.1.8. This cumulative hotfix was released on October 11, 2024.



Important: CDS 3.3 for 7.1.8 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.8. Spark 2 is included in CDP, and does not require a separate parcel.

Contact Cloudera Support for questions related to any specific hotfixes.

Following is the list of fixes that were shipped for CDS 3.3.2.3.3.7180.20-1-1.p0.58114089:

- CDPD-73191: HWC - MatchError when querying a char/varchar partitioned table
- CDPD-73013: Backport SPARK-41585
- CDPD-72621: HWC - Support default constraints while writing into a table
- CDPD-71987: Spark3 UCX build fails due to Centos7 EOL
- CDPD-71817 Spark - Upgrade Aircompressor to 0.27 due to CVE-2024-36114 (remove version pin from Spark and use Hive/ORC's version)
- CDPD-70400: Corruption of pre-epoch dates and timestamps over HWC operations
- CDPD-68917: HWC - DIRECT_READER_V2 mode gives wrong values for string columns after merge query

Table 18: CDS cumulative hotfix 3.3.7180.20-1 download URL

Parcel Repository Location
<code>https://[***USERNAME***]:[***PASSWORD***]@archive.cloudera.com/p/spark3/3.3.7180.20/parcels/</code>

Table 19: Supported Versions

CDS Powered by Apache Spark Version	Dependent Stack Version	Supported CDP Versions
3.3.7180.20	Cloudera Runtime 7.1.8.68-1 (7.1.8 CHF27)	CDP Private Cloud Base with Cloudera Runtime 7.1.8