

CDS 3 Powered by Apache Spark

Date published: 2020-01-30

Date modified:

CLOUdera

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

CDS 3 Powered by Apache Spark Overview.....	4
CDS 3 Powered by Apache Spark Requirements.....	4
Installing CDS 3.0 Powered by Apache Spark.....	5
Install CDS Powered by Apache Spark.....	5
Install the Livy for Spark 3 Service Descriptor.....	6
Updating Spark 2 applications for Spark 3.....	6
Running Applications with CDS 3 Powered by Apache Spark.....	6
The Spark 3 job commands.....	6
Canary test for pyspark3 command.....	7
Fetching Spark 3 Maven Dependencies.....	7
Accessing the Spark 3 History Server.....	7
CDS 3 Powered by Apache Spark Version and Download Information.....	8
CDS 3 versions available for download.....	8
Using the CDS 3 Powered by Apache Spark Maven Repository.....	8
CDS 3.0 Powered by Apache Spark Maven Artifacts.....	8
CDS 3.0 Maven Artifacts.....	9

CDS 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.

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

CDS Powered by Apache Spark is an add-on service for CDP Private Cloud Base, distributed as a parcel and custom service descriptor.

This document describes CDS 3.0 Powered by Apache Spark. It enables you to install and evaluate the [features](#) of Apache Spark 3 without upgrading your CDP Private Cloud Base cluster.

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.

Unsupported Features:

This release does not support the following features:

- Phoenix Connector
- SparkR
- Hive Warehouse Connector
- HBase Connector
- Oozie
- Zeppelin

CDS 3 Powered by Apache Spark Requirements

The following sections describe software requirements for CDS 3.1 Powered by Apache Spark.

CDP Versions



Important: CDS 3 Powered by Apache Spark is an add-on service for CDP Private Cloud Base, and is only supported with Cloudera Runtime 7.1.3, 7.1.4 and 7.1.5. 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
3.0.1.3.0.7110.0-81	CDP Private Cloud Base with Cloudera Runtime 7.1.3, 7.1.4 and 7.1.5

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.

Python Requirement

CDS 3 requires Python 3.4 or higher.

JDK Requirement

CDS 3 requires JDK 8 or JDK 11. Remove other JDK versions from all cluster and gateway hosts to ensure proper operation.

Installing CDS 3.0 Powered by Apache Spark

CDS 3.0 Powered by Apache Spark is distributed as two files: a custom service descriptor file and a parcel, both of which must be installed on the cluster.



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 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 Powered by Apache Spark.

Follow these steps to install CDS 3 Powered by Apache Spark:

1. Check that all the software [prerequisites](#) are satisfied. If not, you might need to upgrade or install other software components first.
2. Install the CDS Powered by Apache Spark service descriptor into Cloudera Manager.
 - a. To download the CDS Powered by Apache Spark service descriptor, click the [service descriptor](#) link for the version you want to install.
 - b. Log on to the Cloudera Manager Server host, and copy the CDS Powered by Apache Spark service descriptor in the location configured for service descriptor files.
 - c. Set the file ownership of the service descriptor to cloudera-scm:cloudera-scm with permission 644.
 - d. Restart the Cloudera Manager Server with the following command:

```
systemctl restart cloudera-scm-server
```

3. 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.

4. Download the CDS Powered by Apache Spark parcel, distribute the parcel to the hosts in your cluster, and activate the parcel. For instructions, see [Managing Parcels](#).
5. 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.
6. Return to the Home page by clicking the Cloudera Manager logo in the upper left corner.
7. Click the stale configuration icon to launch the Stale Configuration wizard and restart the necessary services.

Install the Livy for Spark 3 Service Descriptor

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 a service descriptor for the Livy for Spark 3 service.

1. Install the Livy for Spark 3 service descriptor into Cloudera Manager.
 - a. To download the service descriptor, click the [service descriptor](#) link for the version you want to install.
 - b. Log on to the Cloudera Manager Server host, and copy the Livy service descriptor to the location configured for service descriptor files.
 - c. Set the file ownership of the service descriptor to cloudera-scm:cloudera-scm with permissions set to 644.
 - d. Restart the Cloudera Manager Server with the following command:

```
systemctl restart cloudera-scm-server
```

2. 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.

3. Download the CDS Powered by Apache Spark parcel, distribute the parcel to the hosts in your cluster, and activate the parcel. For instructions, see [Managing Parcels](#).
4. 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.
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.

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 Powered by Apache Spark

With CDS 3 Powered by Apache Spark, 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/canary_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.

CDS 3 Powered by Apache Spark Version and Download Information

The following sections provide links to the parcel and service descriptor files for CDS 3.

CDS 3 versions available for download

Table 1: Available CDS Versions

Version	Custom Service Descriptors	Parcel Repository
3.0.1.3.0.7110.0-81	- SPARK3_ON_YARN-3.0.1.3.0.7110-81.jar - LIVY_FOR_SPARK3-0.6.0.3.0.7110.0-81.jar	https://archive.cloudera.com/p/spark3/3.0.7110.0/parcels/

Using the CDS 3 Powered by Apache Spark Maven Repository



Important: CDS 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 Powered by Apache Spark, 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 3.0 Powered by Apache Spark Maven Artifacts

The following tables lists the groupId, artifactId, and version required to access the artifacts for CDS 3 Powered by Apache Spark:

CDS 3.0 Maven Artifacts

The following pom fragment shows how to access a CDS 3.0 artifact from a Maven POM.

```
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-core_2.12</artifactId>
  <version>3.0.1.3.0.7110.0-81</version>
  <scope>provided</scope>
</dependency>
```

The complete artifact list for this release follows.

Project	groupId	artifactId	version
Apache Spark	org.apache.spark	spark-assembly_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-avro_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-catalyst_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-core_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-cypher_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-graph-api_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-graph_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-graphx_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-hadoop-cloud_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-hive-thriftserver_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-hive_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-kubernetes_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-kvstore_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-launcher_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-mllib-local_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-mllib_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-network-common_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-network-shuffle_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-network-yarn_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-parent_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-repl_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-sketch_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-sql-kafka-0-10_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-sql_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-streaming-kafka-0-10-assembly_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-streaming-kafka-0-10_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-streaming_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-tags_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-token-provider-kafka-0-10_2.12	3.0.1.3.0.7110.0-81
	org.apache.spark	spark-unsafe_2.12	3.0.1.3.0.7110.0-81

Project	groupId	artifactId	version
	org.apache.spark	spark-yarn_2.12	3.0.1.3.0.7110.0-81