

Spark Connect Sessions (Technical Preview)

Date published: 2020-07-30

Date modified: 2025-07-30



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

External IDE connectivity through Spark Connect-based sessions (Technical Preview).....	4
Configuring external IDE Spark Connect sessions.....	4
Sample code to connect to an external IDE Spark Connect session.....	7
Troubleshooting errors when working with an external IDE Spark Connect session.....	9

External IDE connectivity through Spark Connect-based sessions (Technical Preview)

You can learn what an external IDE Spark Connect session is, certain known limitations, and the supported Runtime component versions.

What an external IDE Spark Connect session is

A session is an interactive short-lived development environment for running Spark commands. A Spark Connect Session is a type of [Session](#) that exposes the Spark Connect interface. A Spark Connect Session allows you to connect to Spark from any remote Python environment.

Spark Connect allows you to connect remotely to the Spark clusters. Spark Connect is an API that uses the DataFrame API and unresolved logical plans as the protocol. The separation between client and server allows Spark and its open ecosystem to be leveraged from everywhere. It can be embedded in modern data applications, in IDEs and Notebooks. For more information about Spark Connect, identify the Spark version in your Virtual Cluster, and navigate to the relevant *Spark Connect Overview* page linked to that Spark version in the [Spark documentation](#).

Supported versions of Cloudera Runtime components

Ensure that you are using Spark 3.5.1 before you use Spark Connect Sessions.

Limitations

Spark Connect Sessions do not support the following:

- **Profile support:** Spark Connect does not support profiles in the configuration files even though the clients support "Profiles" in the configuration files.
- **PySpark:** In Spark 3.4, Spark Connect supports most PySpark APIs, including DataFrame, Functions, and Column. Some APIs, such as SparkContext and RDD are not supported. You can check which APIs are currently supported in the [Apache Spark API Reference](#) documentation. Supported APIs are labeled "Supports Spark Connect", so before migrating existing code to Spark Connect, you can check whether the APIs you are using are available. For more information, see the [Apache Spark documentation](#).
- **Scala:** In Spark 3.5, Spark Connect supports most Scala APIs, including Dataset, functions, Column, Catalog, and KeyValueGroupedDataset. For more information, see the [Apache Spark documentation](#).
- **User-Defined Functions (UDFs)** are supported, by default, for the shell and in standalone applications, with additional setup requirements.
- The majority of the Streaming API is supported, including DataStreamReader, DataStreamWriter, StreamingQuery, and StreamingQueryListener. For more information, see the [Apache Spark documentation](#).
- APIs, such as SparkContext and RDD are deprecated in all Spark Connect versions.

Configuring external IDE Spark Connect sessions

Learn about how to configure a Spark Connect Session with .

Before you begin

Before you create a Spark Connect Session, perform the following steps:

1. **Enable a service .**
2. **Create a Virtual cluster.** You must select All Purpose (Tier 2) in the Virtual Cluster option and Spark 3.5.1 as the Spark version.

Procedure

1. Perform the following steps on each user's machine:

- Create the `~/.cde/config.yaml` configuration file and add the `vcluster-endpoint` and `cdp-endpoint` parameters.

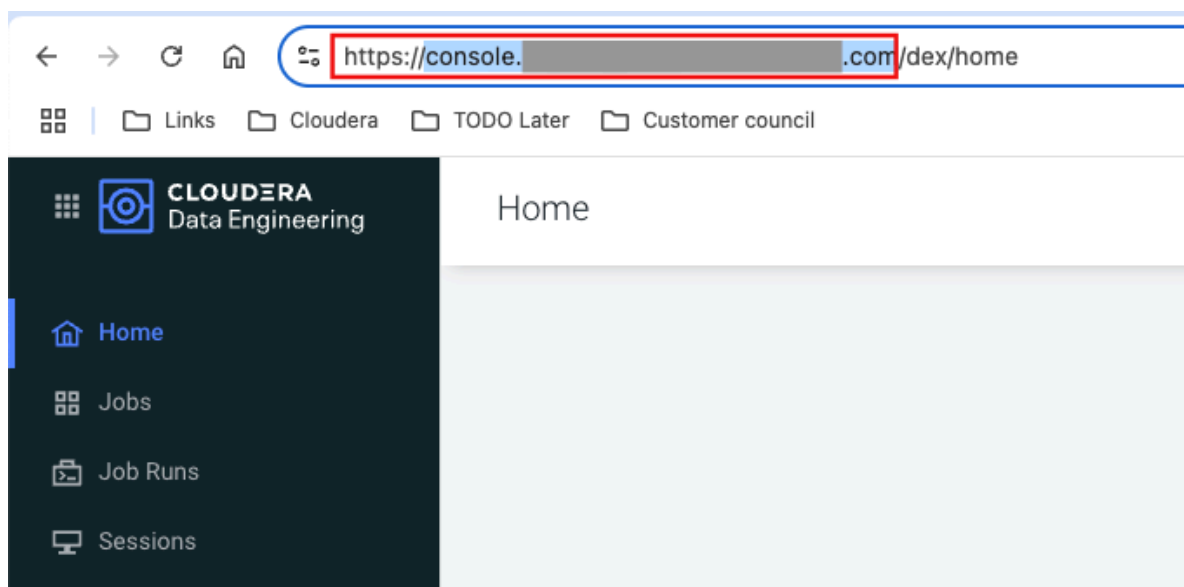
This allows the client machine to identify a virtual cluster.



Note: The `cdp-endpoint` value is the same as the console URL. From the console URL, copy the protocol (`https://`), the subdomain (`console`), the second-level domain, and the top-level domain (`.com`).

Example: `https://console.[***SECOND-LEVEL-DOMAIN***].com`

Figure 1: Getting the endpoint URL from the console URL



For more information, see [vcluster-endpoint](#) and [cdp-endpoint](#).

For example,

```
cdp-endpoint: https://console.cdp.apps.example.com
credentials-file: /Users/user1/.cde/credentials
vcluster-endpoint: https://ffws6v27.cde-c9b822vr.apps.example.com/dex/api/v1
```

- Create an [access key](#) and update the `credentials-file` parameter in the `~/.cde/config.yaml` configuration file with the path where the credentials file is located. This allows the client machine to acquire the short-lived access tokens.



Note: Access keys configured with the default profile are supported.

For example,

```
[default]
cdp_access_key_id=571ff....
cdp_private_key=dvbYd....
```

2. Create a Spark Connect Session using one of the following methods:



Note: You can interact with a Spark Connect session that only you have created.

- Using the UI: Create a new session as per [Creating Sessions in](#) but when you select the session type, select Spark Connect (Tech Preview) from the Type drop-down list.

The screenshot shows the 'Create Session' form in the Cloudera IDE. It is divided into two main sections: 'Session Details' and 'Compute Options'.

Session Details:

- Name ***: A text input field with the placeholder 'Session Name'.
- Type ***: A dropdown menu currently showing 'Spark Connect (Tech Preview)'.
- Description**: A text input field with the placeholder 'Describe the session'.
- Configurations (Optional)**: Two input fields labeled 'config_key' and 'config_value', with a plus icon to add more configurations.

Compute Options:

Configure the compute options for this session

Option	Slider Range	Current Value
Number Executors	1 to 40	1
Driver Cores	1 to 16	1
Executor Cores	1 to 16	1
Driver Memory (GB)	1 to 32	1
Executor Memory (GB)	1 to 32	1

At the bottom of the form are two buttons: 'Cancel' and 'Create'.

- [Using the CLI](#): Create a Spark Connect Session by running the following command:

```
cde session create --name [***SPARK-SESSION-NAME***] --type spark-connect
```

**Note:**

To get all the attributes of a cde session command, run the `cde session -h` command.

- On the Home page, click Sessions and then select the Spark Connect Session that you have created.
- Go to the Connect tab and download the required TAR file and PySpark TAR file as displayed on the screen.

The screenshot shows the Cloudera Data Engineering (CDE) interface for a session named 'test-connect-2'. The session is in an 'Available' status, created by 'cdpuser1' on Jan 31, 2024, at 2:34:35 PM. The 'Connect' tab is selected, displaying instructions for connecting to the session using a Python client.

Connect with Spark Connect
Interact with this session using the python client. This requires a few steps outlined below. More information can be found in our documentation [\[link\]](#)

Step 1 : Configure

- Download the required CDE Tarball file to work with your Spark Connect Session
- Download the Pyspark 3.4 tarball required by the CDESparkConnectSession package
- Install in your python environment by running

```
pip install <cde connect tarball>
pip install <pyspark tarball>
```

- Configure the python package

Step 2 : Connect

- Use the package to connect with this session

```
from cde import CDESparkConnectSession
spark = CDESparkConnectSession.builder.sessionName('test-connect-2').get()
```

Need help? [Follow our detailed guide \[link\]](#)

**Note:**

- The Copy Link option can be used to retrieve a URL and download the client using cURL.
 - The PySpark TAR file version must be same as the Virtual Cluster's Spark version.
- Create a new Python virtual environment or use your existing one and install the TAR file after activating your Python virtual environment.

```
python3 -m venv cdeconnect
. cdeconnect/bin/activate

pip install [***CDECONNECT_TARBALL***]
pip install [***PYSPARK_TARBALL***]
```

Sample code to connect to an external IDE Spark Connect session

After configuring Spark Connect Sessions, learn how you can run the CLI commands from a remote Python host to connect to a session and execute Spark SQL commands through an example.

You can use the following sample code to connect to the Spark Connect session. Use the spark variable to interact with Spark as you connect to the jobs or sessions.

```
> python
Python 3.9.13 (main, Jul 29 2022, 12:22:24)
[Clang 13.0.0 (clang-1300.0.27.3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from cde import CDESparkConnectSession
>>> spark = CDESparkConnectSession.builder.sessionName('connect-session').get()
>>> spark.version
'3.4.1.1.20.7180.0-33'
>>> spark.sql("use retaildb").show()
++
||
++
++

>>> spark.sql("select * from products_external").show()
+-----+-----+-----+-----+-----+
|product_id|product_category_id|product_name|product_description|product_price|product_image|
+-----+-----+-----+-----+-----+
|1|2|Quest Q64 10 FT. ...| |59.98|http://images.acm...|
|2|2|Under Armour Men'...| |129.99|http://images.acm...|
|3|2|Under Armour Men'...| |89.99|http://images.acm...|
|4|2|Under Armour Men'...| |89.99|http://images.acm...|
|5|2|Riddell Youth Rev...| |199.99|http://images.acm...|
|6|2|Jordan Men's VI R...| |134.99|http://images.acm...|
|7|2|Schutt Youth Recr...| |99.99|http://images.acm...|
|8|2|Nike Men's Vapor ...| |129.99|http://images.acm...|
|9|2|Nike Adult Vapor ...| |50.0|http://images.acm...|
|10|2|Under Armour Men'...| |129.99|http://images.acm...|
|11|2|Fitness Gear 300 ...| |209.99|http://images.acm...|
|12|2|Under Armour Men'...| |139.99|http://images.acm...|
|13|2|Under Armour Men'...| |89.99|http://images.acm...|
|14|2|Quik Shade Summit...| |199.99|http://images.acm...|
|15|2|Under Armour Kids...| |59.99|http://images.acm...|
|16|2|Riddell Youth 360...| |299.99|http://images.acm...|
|17|2|Under Armour Men'...| |129.99|http://images.acm...|
|18|2|Reebok Men's Full...| |29.97|http://images.acm...|
|19|2|Nike Men's Finger...| |124.99|http://images.acm...
```



```
|      20 |      2 | Under Armour Men' ... |
| 129.99 | http://images.acm... |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
only showing top 20 rows
```

Troubleshooting errors when working with an external IDE Spark Connect session

While working with the Spark Connect Sessions in , you might encounter errors. Learn how you can troubleshoot those errors.

Condition

If the session is killed or the driver exits due to an error when the code is being executed, Spark Connect shows the following error.

```
pyspark.errors.exceptions.connect.SparkConnectGrpcException: <_MultiThreaded
Rendezvous of RPC that terminated with:
  status = StatusCode.UNKNOWN
  details = "Stream removed"
  debug_error_string = "UNKNOWN:Error received from peer {grpc_message:"Str
eam removed", grpc_status:2, created_time:"2024-01-31T13:28:23.35214+05:30"}
"
>
```

Remedy

Procedure

Check the actual error from the session driver logs using [UI](#) or [CDE CLI](#).