Cloudera Data Engineering 1.5.5

Creating and managing Cloudera Data Engineering Sessions

Date published: 2020-07-30
Date modified: 2025-11-08

https://docs.cloudera.com/

Legal Notice

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

Creating sessions in Cloudera Data Engineering

A Cloudera Data Engineering session is an interactive short-lived development environment for running Spark commands to help you iterate upon and build your Spark workloads.

About this task

The commands that are run in a Cloudera Data Engineering session are called Statements. You can submit the Statements through the connect CLI command or the Interact tab in the Cloudera Data Engineering UI for a session. Python and Scala are the supported session types. Learn how to use Cloudera Data Engineering sessions using the user interface and CLI.

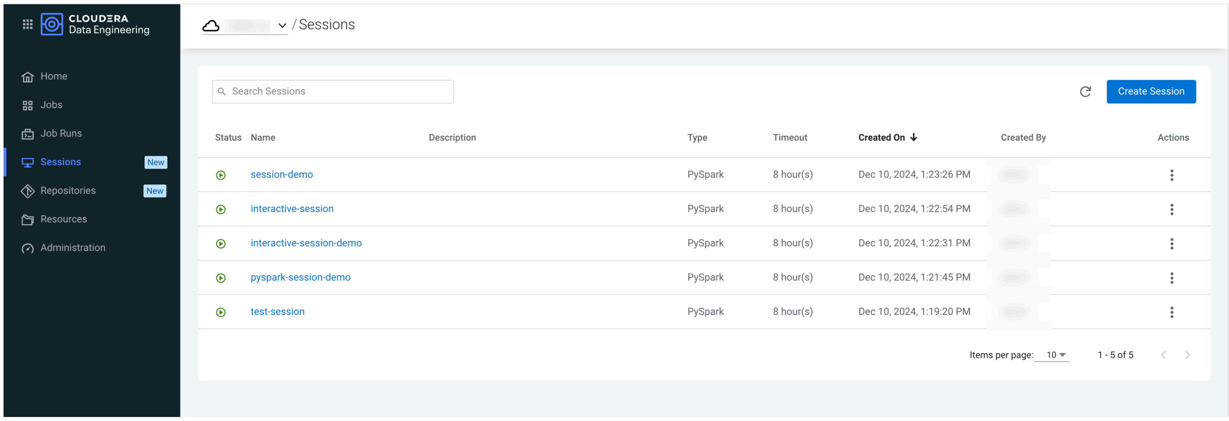
Before you begin

Important: You must create the All-Purpose cluster, initialize each cluster, and configure each user who need to run sessions before creating sessions.

In Cloudera Data Engineering, sessions are associated with virtual clusters. Before you can create a session, you must create a virtual cluster that can run it. For more information, see Creating virtual clusters.

Procedure

1. In the Cloudera console, click the Data Engineering tile. The Home page displays.
2. Click Sessions in the left navigation menu and then click Create Session.



Status	Name	Description	Type	Timeout	Created On ↓	Created By	Actions
▶	session-demo		PySpark	8 hour(s)	Dec 10, 2024, 1:23:26 PM	[redacted]	⋮
▶	interactive-session		PySpark	8 hour(s)	Dec 10, 2024, 1:22:54 PM	[redacted]	⋮
▶	interactive-session-demo		PySpark	8 hour(s)	Dec 10, 2024, 1:22:31 PM	[redacted]	⋮
▶	pyspark-session-demo		PySpark	8 hour(s)	Dec 10, 2024, 1:21:45 PM	[redacted]	⋮
▶	test-session		PySpark	8 hour(s)	Dec 10, 2024, 1:19:20 PM	[redacted]	⋮

Items per page: 10 1 - 5 of 5

3. Enter a Name for the session.
4. Select a Type, for example, PySpark, Scala, or Spark Connect.
5. Select a Timeout value.

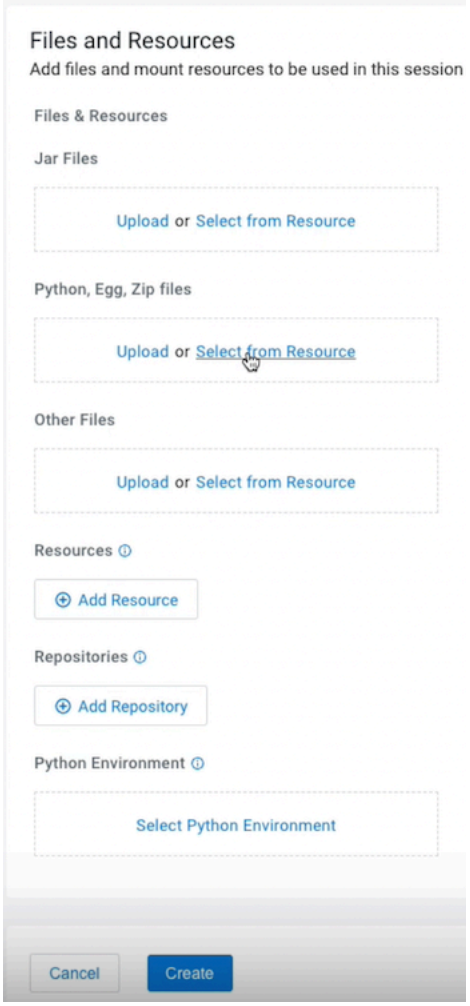
   The session will stop after the indicated time has passed.
6. Enter a Description for the session.
7. Enter the Configurations.

   Note: The Spark session is created during the job run or session creation. Most Spark configurations are not modifiable during runtime and has to be specified during job run or session creation. You can check if a configuration can be modified by using spark.conf.isModifiable. For example,

   ```
   spark.conf.isModifiable("spark.executor.memory")
   False
   ```
8. Click Data Connector drop-down list and select the name of the data connector from the list. The UI displays the storage information that is internally overwritten. For more information about how to add an Ozone data connector, see Adding Ozone data connector for Cloudera Data Engineering service.

9. Set the Compute options.
    - Optional: GPU Acceleration (Technical Preview): You can accelerate your session using GPUs. Click Enable GPU Accelerations checkbox to enable the GPU acceleration and configure selectors and tolerations if you want to run the job on specific GPU nodes. When you run this session, this particular session will request GPU resources.
10. In the **Files and Resources** section, you can upload Jar, Python, Egg, Zip, and other files. You can also add a resource, respositories, or a Python environment to be used in this session.

    Files that are uploaded to a session are stored in the app/mount directory.



Files and Resources

Add files and mount resources to be used in this session

Files & Resources

Jar Files

Upload or Select from Resource

Python, Egg, Zip files

Upload or Select from Resource

Other Files

Upload or Select from Resource

Resources

Add Resource

Repositories

Add Repository

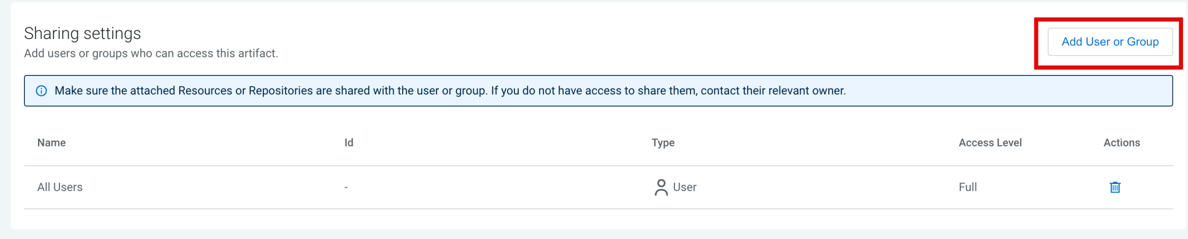
Python Environment

Select Python Environment

Cancel Create

11. Share the session with a user or group.

a) In the Sharing Settings section, click Add User or Group. The Add User or Group pop-up appears.



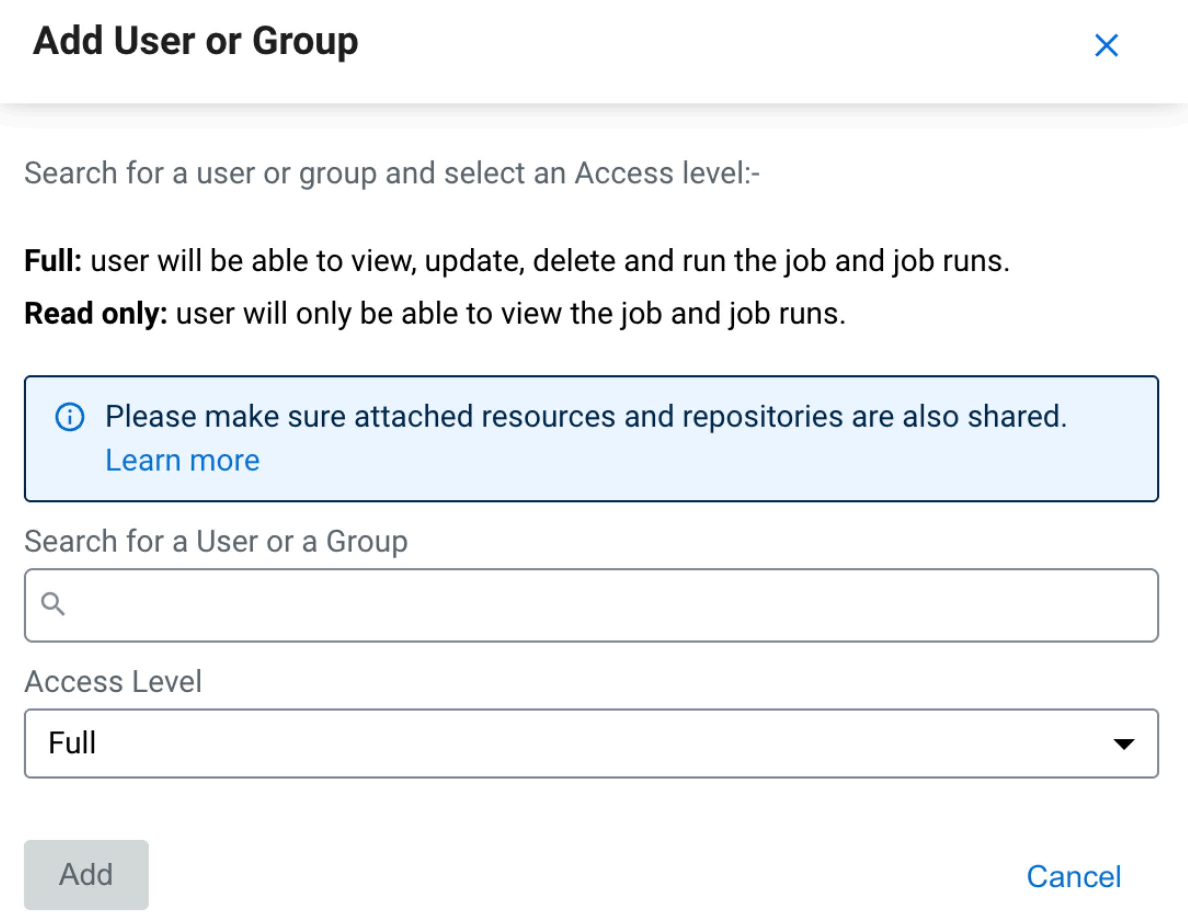
Sharing settings

Add users or groups who can access this artifact.

ⓘ Make sure the attached Resources or Repositories are shared with the user or group. If you do not have access to share them, contact their relevant owner.

Name	Id	Type	Access Level	Actions
All Users	-	User	Full	🗑

b) In the Search for a User or a Group field, type the user or group name and select the required user or group from the list.



Add User or Group

Search for a user or group and select an Access level:-

Full: user will be able to view, update, delete and run the job and job runs.

Read only: user will only be able to view the job and job runs.

ⓘ Please make sure attached resources and repositories are also shared. Learn more

Search for a User or a Group

Access Level

Full

c) Select Full or Read Only depending on the access you want to provide from the Access Level drop-down list.

d) Click Add.

12. Click Create.

The Connect tab displays a list of connectivity options available to interact with the session. The Interact tab allows you to interact with the session, and becomes available once the session is running.

13. To delete a session, open the session and click Delete.

Note: Deleting a session results in the termination of an active session and the loss of any attached logs and details.

Interacting with a Session in Cloudera Data Engineering

Once a Session is created in Cloudera Data Engineering, you can begin your interaction.

About this task

You can also interact with files that were uploaded to a Session. Those files are stored in the app/mount directory.

Procedure

1. In the Cloudera console, click the Data Engineering tile. The Home page displays.
2. Click Sessions in the left navigation menu.
3. Click the Interact tab.
4. In the Input box, enter your commands.
5. Click Run. A history of executed commands displays in the sequence of their execution.

Connecting Sessions with the CDE CLI

Once a Session is created in Cloudera Data Engineering, you can download the CDE CLI from the user interface then use the terminal on your local machine to set up the interactive session using the CDE CLI.

Procedure

1. In the Cloudera console, click the Data Engineering tile. The Home page displays.
2. Click Sessions in the left navigation menu.
3. Click the Connect tab.
4. Download the CDE CLI.
5. Use the CDE CLI to run and connect interactive commands:
   a) Copy the command that displays.
   b) Paste the command in the CDE CLI and continue to interact with your session.

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

⚠ **Important:** External IDE connectivity through Spark Connect-based sessions is supported from Cloudera Data Engineering 1.5.5 CHF1 or higher versions.

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

Supported versions of Cloudera Runtime components

Ensure that you are using the following software versions of the Cloudera Runtime components before you use Spark Connect Sessions:

- Spark 3.5
- Cloudera Runtime 7.3.1

Supported Spark Connectors

The following Spark Connectors are supported with the previously listed Runtime component versions:

- Hive
- HDFS
- Hive tables Parquet storage
- Hive tables ORC storage
- Ranger - table-level access controls

Limitations

Spark Connect Sessions do not support the following:

- Profile support: Spark Connect does not support profiles in the configuration files even though the Cloudera Data Engineering clients support "Profiles" in the configuration files.
- Documentation links within the Spark Connect UI point to incorrect documents.
- Session creation allows a mix of uppercase and lowercase letters in the session names. However, using uppercase letters causes Spark Connect Sessions to connect incorrectly. As a workaround, use only lowercase letters in session names.
- Access control support: Spark Connect Sessions do not support access control. After a session is created, anyone with access to the virtual cluster can connect to it.
- 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 Cloudera Data Engineering.

Before you begin

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

1. Create a Cloudera Data Engineering Service.
2. Create a Cloudera Data Engineering Virtual cluster. You must select All Purpose (Tier 2) in the Virtual Cluster option and Spark 3.4.1 as the Spark version.
3. Initialize the virtual cluster.
4. Configure Hadoop Authentication.

5. If you are using an OpenShift cluster, then run the following command:

```
$ oc -n openshift-ingress-operator annotate ingresscontrollers/default i
ngress.operator.openshift.io/default-enable-http2=true
```

Procedure

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

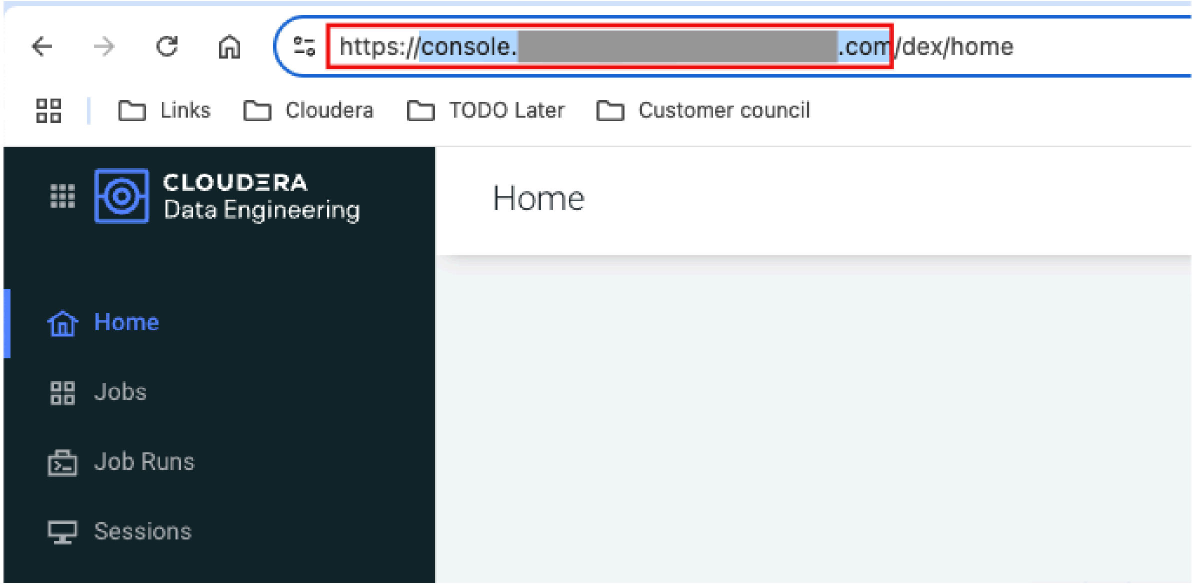
a) Create the ~/.cde/config.yaml configuration file and add the vcluster-endpoint and cdp-endpoint parameters.

This allows the client machine to identify a virtual cluster.

Note: The cdp-endpoint value is the same as the Cloudera console URL. From the Cloudera 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 Cloudera endpoint URL from the Cloudera 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
```

b) 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.

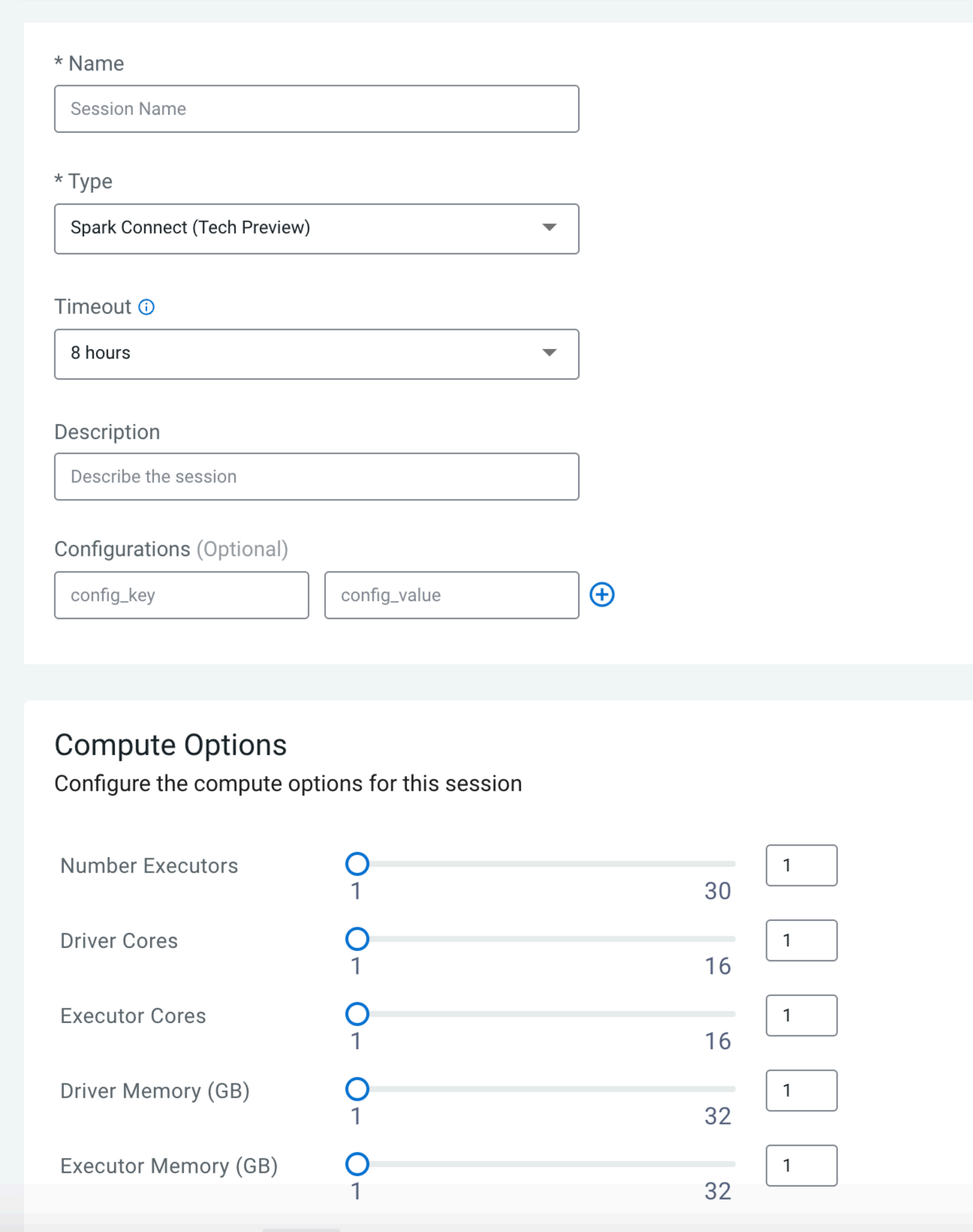
Note: Access keys configured with the default profile are supported.

For example,

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

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

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



* Name

Session Name

* Type

Spark Connect (Tech Preview)

Timeout

8 hours

Description

Describe the session

Configurations (Optional)

config_key config_value

Compute Options

Configure the compute options for this session

Number Executors	1	30	1
Driver Cores	1	16	1
Executor Cores	1	16	1
Driver Memory (GB)	1	32	1
Executor Memory (GB)	1	32	1

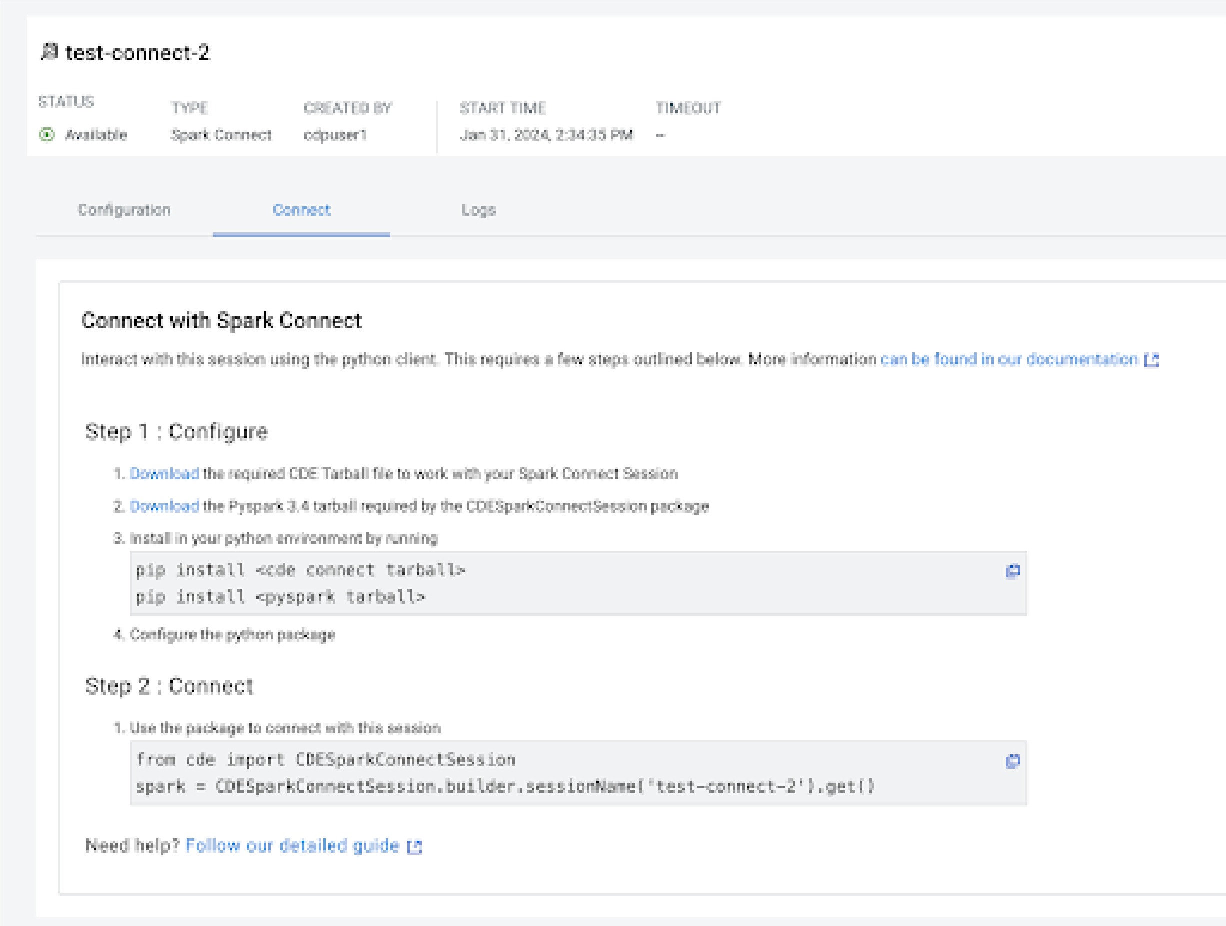
- Using the CLI: Create a Spark Connect Session by running the following command:

```
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.

3. On the Cloudera Data Engineering Home page, click Sessions and then select the Spark Connect Session that you have created.
4. Go to the Connect tab and download the required Cloudera Data Engineering TAR file and PySpark 3.4 TAR file as displayed on the screen.



test-connect-2

STATUS	TYPE	CREATED BY	START TIME	TIMEOUT
Available	Spark Connect	cdpuser1	Jan 31, 2024, 2:34:35 PM	--

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

Step 1 : Configure

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

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

4. Configure the python package

Step 2 : Connect

1. 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

Note: The Copy Link option can be used to retrieve a URL and download the client using cURL.

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.

5. 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***]
```

6. If you have used the self-signed certificates while Initializing the virtual cluster, then you must configure the certificates for the Cloudera Data Engineering Virtual Cluster, Spark Connect gRPC server, and the control plane hosts to be trusted. Append all the certificates belonging to those hosts to the Python "certifi cacerts ca" truststore.

Usually, the path of the truststore is venv/lib/python3.7/site-packages/certifi/cacert.pem. For trusting gRPC connections, export the following variable:

Note: If you do not have self-signed certificates, manually create cacerts.pem file.

```
# In bash_profile or terminal
export GRPC_DEFAULT_SSL_ROOTS_FILE_PATH=venv/lib/python3.7/site-packages/c
ertifi/cacert.pem

# In a Jupyter notebook use the inbuilt %env magic
%env GRPC_DEFAULT_SSL_ROOTS_FILE_PATH=~/<path-to-cert>
```

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 Cloudera Data Engineering 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').ge
t()
>>> 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|pro
duct_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 Cloudera Data Engineering, 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.

Creating cacerts.pem file

You can create a cacerts.pem file including all the self-signed certificates in it.

About this task

You can collect all the self-signed certificates of the control plane, virtual cluster, Spark Connect, and Cloudera Data Engineering service hostnames and add all of them to a cacerts.pem file.

Procedure

1. Identify the control plane hostname.

   The control plane hostname is the host name present in the URL used to access the on premises control plane. For example, if `https://console-cdp.apps.example.cloudera.com` is the control plane URL, then `console-cdp.apps.example.cloudera.com` is the control plane hostname.
2. Identify the virtual cluster hostname as follows:
   a) Select the Cloudera Data Engineering service containing the virtual cluster that you want to activate.
   b) Click Cluster Details.
   c) Click JOBS API URL to copy the URL to your clipboard.
   d) Paste the URL into a text editor to identify the endpoint host.

      For example, if the copied URL is `https://dfdj6kgx.cde-2cdxw5x5.apps.ecs-demo.example.com/dex/api/v1`, then the endpoint host is `dfdj6kgx.cde-2cdxw5x5.apps.ecs-demo.example.com`.
3. Identify the Spark Connect hostname: Using the virtual cluster hostname identified in the preceeding step, prefix "`sc-`" to this hostname to obtain the Spark Connect hostname. For example, if the virtual cluster hostname is `dfdj6kgx.cde-2cdxw5x5.apps.ecs-demo.example.com`, then the Spark Connect hostname is `sc-dfdj6kgx.cde-2cdxw5x5.apps.ecs-demo.example.com`.
4. Identify the Cloudera Data Engineering service hostname as follows:
   a) Select the Cloudera Data Engineering service containing the virtual cluster that you want to activate.
   b) Click Cluster Details.
   c) Click Grafana Charts to copy the URL to your clipboard.
   d) Paste the URL into a text editor to identify the endpoint host.

      For example, if the copied URL is `https://service.cde-2cdxw5x5.apps.ecs-demo.example.com/grafana/249u4dnkfnkdf` then the endpoint host is `service.cde-2cdxw5x5.apps.ecs-demo.example.com`.
5. Run the following commands to add all the certificates into a single cacerts.pem file:

```
openssl s_client -connect [***CONTROL PLANE HOSTNAME***]:443 2>/dev/null
 </dev/null |  sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' >> c
acerts.pem
openssl s_client -connect [***VIRTUAL CLUSTER HOSTNAME***]:443 2>/dev/null
 </dev/null |  sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' >> c
acerts.pem
openssl s_client -connect [***SPARK CONNECT HOSTNAME***]:443 2>/dev/null
 </dev/null |  sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' >> c
acerts.pem
openssl s_client -connect [***CDE SERVICE HOSTNAME***]:443 2>/dev/null </
dev/null |  sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' >> cacer
ts.pem
```

Viewing logs for Cloudera Data Engineering Sessions

Once a Session is created in Cloudera Data Engineering, you can view logs of the Spark driver and executors from the Session.

Procedure

1. In the Cloudera console, click the Data Engineering tile. The Home page displays.
2. Click Sessions in the left navigation menu.
3. Click the Session name that you want to view the logs for.
4. Click the Logs tab.
5. In the Select Log Type drop-down, select the Driver or Executor that you want to view the logs for.
6. Click Download All Logs to save your log files.

Viewing Spark UI for Cloudera Data Engineering Sessions

You can view the Spark UI for a Cloudera Data Engineering active Session for troubleshooting purposes.

About this task

Procedure

1. In the Cloudera console, click the Data Engineering tile. The Home page displays.
2. Click Sessions in the left navigation menu.
3. Select the Virtual Cluster from the drop-down menu at the top.
4. Select the Active/Running sessions that you wan to troubleshoot.
5. Click the Spark UI tab.