

Setting Up Data Connections

Date published: 2020-07-16

Date modified: 2025-03-18

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

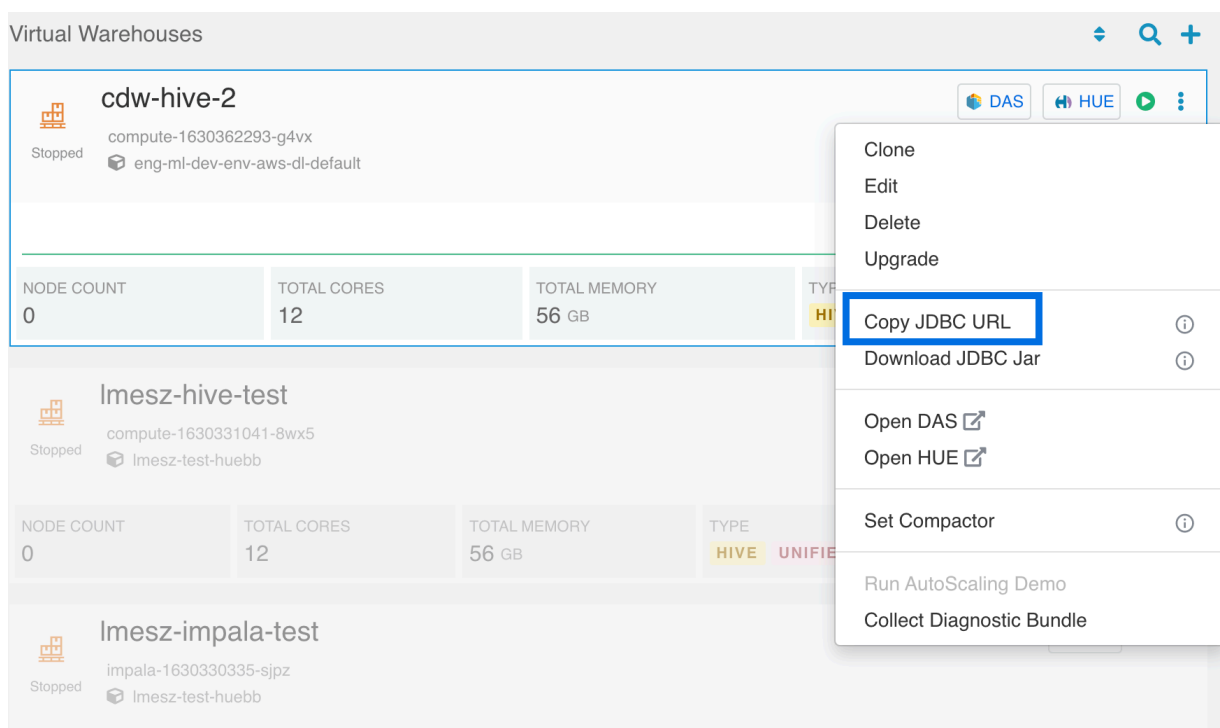
- Set up a Hive or Impala data connection manually.....4**
- Setting up a Spark data connection.....5**
- Setting up Amazon S3 data connection.....6**
- Setting up a data connection to Cloudera Data Hub.....6**
 - Set up a DataHub data connection using raw code.....8
- Data connection management.....8**
- Setting up a Custom Data Connection.....9**
 - Custom Data Connection Development..... 11
 - Developing and testing your first custom connection..... 11
 - Loading custom connections..... 17

Set up a Hive or Impala data connection manually

Data connections to Hive or Impala virtual warehouses within the same environment as the Cloudera AI Workbench are automatically discovered and configured. You can also set up a data connection manually, which works across Cloudera environments. Follow this procedure to set up a Hive or Impala data connection.

Procedure

1. Log into the Cloudera web interface and navigate to the Cloudera Data Warehouse service.
2. In the Cloudera Data Warehouse service, select Virtual Warehouses in the left navigation panel.
3. Select the options menu for the warehouse you want to access, and select Copy JDBC URL.




4. Return to the Cloudera AI service. In Site Administration Data Connections , select New Connection.
5. Enter the connection name. You cannot have duplicate names for data connections within a workbench or within a given project.
6. Select the connection type:
 - a. Hive Virtual Warehouse
 - b. Impala Virtual Warehouse
7. Paste the JDBC URL for the data connection.

8. (Optional) Enter the Virtual Warehouse Name. This is the name of the warehouse in Cloudera Data Warehouse.

New Data Connection ✕

*** Name**

*** Type** ⓘ

 Hive Virtual Warehouse ▼

*** JDBC URL** ⓘ

Virtual Warehouse Name ⓘ

☒ **Available** ⓘ

CancelCreate

Results

The data connection is available to users by default. To change availability, click the Available switch. This switch determines if the data connection is displayed in Projects created within the workbench.

Setting up a Spark data connection

Spark data connections within the same environment as Cloudera AI are automatically discovered, but you can also set up a connection manually. Follow this procedure to set up a Spark data connection.

Procedure

1. In the Workbenches UI, select the link environment for the workbench you are using. This takes you to the Environments UI.

2. In Environments, select **Data Lake Cloud Storage** tabs.
3. Select the directory path shown for Hive Metastore External Warehouse, and copy it.
4. In Project Settings > Data Connections, click **New Connection**.
5. Enter a name for the connection.
6. Select the type: **Spark Data Lake**
7. Paste the value you copied in step 3 into **Datalake Hive Metastore External Warehouse Directory**.
8. Click **Create**.

Results

The data connection is available to users by default. To change availability, click the **Available** toggle.

Setting up Amazon S3 data connection

Amazon S3 object store connection is automatically created for Cloudera AI Workbenches to make it easier to connect to the data stored within the same environment. Other Data Connections can be configured to other S3 locations manually. You can also set up a connection manually.

About this task

Amazon S3 data connections are available only on AWS workbenches where RAZ (Ranger Authorization Service) is enabled to authorize connections to the environment's S3 buckets.

Procedure

1. In the **Cloudera** console, click the **Cloudera AI** tile.
The **Home** page displays.
2. Click **Site Administration** in the left navigation menu.
The **Site Administration** page displays.
3. In the Data Connections page, click **New Connection**.
The **New Data Connection** window is displayed.
4. In the Name field, enter a name for the connection.
5. In the Type drop-down list, select the type as **S3 Object Store**.
6. Click **Create**.

This data connection only supports per-bucket operations. For information on using the S3 data connection, and connection wrapper for the S3 boto client, see *Amazon Boto documentation*.

The data connection is available to users by default. To change availability, click the **Available** toggle. This switch determines if the data connection is displayed in Projects created within the workbench.

Related Information

[Amazon Boto documentation](#)

Setting up a data connection to Cloudera Data Hub

You can set up a data connection to the DataHub cluster. You can set up a connection using the **New Connection** dialog, or by using raw code inside your project. Both approaches are shown below.

Procedure

1. Log into the console web UI.

2. Depending on which connection you want to use, click on either **Connect to Hive** or **Connect to Impala** tile.
3. Ensure your Data Hub cluster name is correct in the popup.
4. Copy the JDBC URL string.
5. Now click on the **Build a Data Science Project** tile to log into the the Cloudera AI Workbench.
6. In **Site Administration Data Connections** , select **New Connection**.
7. Return to the Cloudera AI service. In **Site Administration Data Connections** , select **New Connection**.
8. Enter the connection name. You cannot have duplicate names for data connections within a workbench or within a given project.
9. Select the connection type:
 - a. **Hive Virtual Warehouse**
 - b. **Impala Virtual Warehouse**
10. Paste the JDBC URL for the data connection.
11. (Optional) Enter the Virtual Warehouse Name. This is the name of the warehouse in Cloudera Data Warehouse.

New Data Connection ✕


*

Name

*

Type

ⓘ

 Hive Virtual Warehouse ▼

*

JDBC URL

ⓘ

Virtual Warehouse Name

ⓘ

☒

Available

ⓘ

Cancel

Create

Results

The data connection is available to users by default. To change availability, click the Available switch. This switch determines if the data connection is displayed in Projects created within the workbench.

Set up a DataHub data connection using raw code

It is recommended to use the New Connection dialog to create a new data connection. If needed, you can also set up a data connection in your project code by using and adapting the following code snippet.

Example

```
from impala.dbapi import connect

#Example connection string:
# jdbc:hive2://my-test-master0.eng-ml-i.svbr-nqvp.int.cldr.work/;ssl=true;
transportMode=http;httpPath=my-test/cdp-proxy-api/hive

conn = connect(
    host = "my-test-master0.eng-ml-i.svbr-nqvp.int.cldr.work",
    port = 443,
    auth_mechanism = "LDAP",
    use_ssl = True,
    use_http_transport = True,
    http_path = "my-test/cdp-proxy-api/hive",
    user = "csso_me",
    password = "Test@123")
cursor = conn.cursor()
cursor.execute("select * from 3yearpop")

for row in cursor:
    print(row)
cursor.close()
conn.close()
```

Data connection management

There are a few things to keep in mind about data connections.

- Manage data connections in a workbench

At the workbench level, you can check the data connections that are available in a workbench. In Project Settings Data Connections, check that your desired data source is present. You can also set the availability for any discovered connections, if necessary.

- Manage data connections in a project

All the data connections that were available in the workbench when the project was created will be automatically created in the project as well. The available connections can then further be marked as unavailable if so desired. You can update any changes to the connections that were made at the workbench level by clicking Sync with workbench. Any changes made here only apply to your project.

- Data connection availability

Keep in mind these two scenarios for setting data connection availability.

1. If a workbench data connection is marked Unavailable, and you then create a project, the data connection will not appear in the project. If the connection is then changed to Available, and then the Sync with workbench button is clicked, the connection will appear in the project.
2. If a workbench data connection is marked Available, and you then create a project, the connection shows up. If the workbench data connection is then toggled to Unavailable, and you click Sync with workbench in the project, the data connection will remain available in the project.

Setting up a Custom Data Connection

Data connections that point to data sources outside of Cloudera or require custom configurations can be created and made available to end users with Custom Data Connections. These Python implementations of the Cloudera AI Data library are stored in the Data Connections Registry. Workbench users can track and connect to any data source and connection implementation a Cloudera AI Administrator makes available.

About this task

Consider the followings:

- Custom connections can only be created in projects created by the Administrator.
- The project source selection list in the Data Connection creation dialogue only displays projects created by the user.
- Team projects or projects with multiple collaborators will also not be displayed, only those directly created by the user.
- Custom connections at workbench level can only be edited by the creator, not other Administrator users. Attempts at editing workbench level custom connections will result in an error.

Before you begin

Before setting up a custom connection, you might want to create a dedicated Cloudera AI Team to collaborate on external connections. A good practice is to separate the connection code projects and and configure collaborators on the Team level to build and maintain the connection code.

Procedure

1. Develop your own custom data connection (see *Developing a Custom Data Connection*) in a Cloudera AI project, or clone an existing custom data connection files directory into a Cloudera AI project.
2. In Site Administration Data Connections , select New Connection.
3. Enter the connection name. You cannot have duplicate names for data connections within a workbench or within a given project.
4. Select the connection type: Custom Connection
5. Enter the Type Display name. This should be a descriptive label to help Cloudera AI project owners identify what this custom connection could be used for.
6. Select the Cloudera AI Project and Project directory which contains your custom connection implementation
 - a. Connection files must be in a directory and not in the root of your project.
 - b. A snapshot of all implementation files in the directory will be uploaded to the Cloudera AI Custom Data Connection registry located in the workbench.
 - c. These uploaded files are safe from any changes to the originating project. To make changes to the files, create a new custom data connection.

7. (Optional) Enter any custom parameters. These are available during a session and can be validated or overridden depending on the interface implementation for the custom data connection. Refer to the implementation of your custom data connection for specific details on required keys and values.

New Data Connection

X

*

Name

postgres_v1

*

Type

i

Custom Connection

▼

*

Type Display

Postgres

*

Project

i

Project Name

▼

Custom Parameters

i

DB_NAME

database

-

+

HOST_NAME

host

-

+

Available

i

Cancel

Create

8. Click Create.

Results

The data connection is now available to all users. To change availability, click the Available switch. This switch determines if the data connection is displayed in Projects created within the workbench. Refer to *Data connection management* for availability of your newly created custom connections in new and existing Cloudera AI Projects.

Related Information

[Data connection management](#)

Custom Data Connection Development

Custom data connections can be developed from within Cloudera AI Workbench and Python Sessions using the Cloudera AI Python Data Library and implementing the Cloudera AI Custom Connection Interface.

You can view CustomConnection interface help descriptions within in a session:

```
import cml.data_v1 as cmldata
help(cmldata.customconnection)
```

Alternatively, you can inspect the source content as follows:

```
import cml.data_v1 as cmldata
import inspect
print(inspect.getsource(cmldata.customconnection))
```

Your custom connection code must implement the CustomConnection interface for the cml.data_v1 library to load your module dynamically (see *Loading custom connections*)

Two functions are already implemented so that the Cloudera AI Python Data Library can dynamically load your Python module implementation and make custom parameters available in self.parameters. In most cases, you will not need to reimplement these:

1. `__init__(self, properties)`
2. `update_properties(self, properties)`

The rest of the interface functions are included as common functions that you may want to implement.

1. `get_base_connection(self)`
2. `get_pandas_dataframe(self, query)`
3. `get_cursor(self)`
4. `print_usage(self)`
5. `override_parameters(self)`

See *Developing and testing your first custom connection* for a simple example of how to implement these.

Related Information

[Loading custom connections](#)

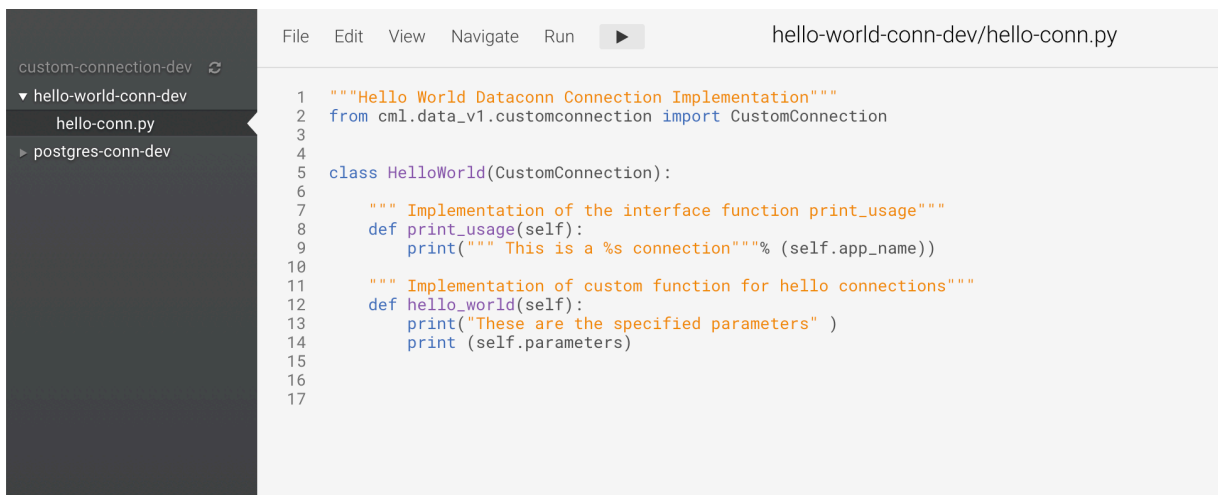
[Developing and testing your first custom connection](#)

Developing and testing your first custom connection

You can develop a custom connection in your own project, as shown in this example.

Procedure

1. Create a new directory `hello-world-conn-dev` and file `hello-conn.py` to contain your custom data connection files.



```

File Edit View Navigate Run ▶ hello-world-conn-dev/hello-conn.py
1 """Hello World Dataconn Connection Implementation"""
2 from cml.data_v1.customconnection import CustomConnection
3
4
5 class HelloWorld(CustomConnection):
6
7     """ Implementation of the interface function print_usage"""
8     def print_usage(self):
9         print(""" This is a %s connection"""% (self.app_name))
10
11     """ Implementation of custom function for hello connections"""
12     def hello_world(self):
13         print("These are the specified parameters" )
14         print (self.parameters)
15
16
17
  
```

2. Implement your custom data connection in `hello-conn.py`. (There must only be one class which implements `CustomConnection` in this directory.)

```

"""Hello World Custom Connection Implementation"""
from cml.data_v1.customconnection import CustomConnection

class HelloWorld(CustomConnection):

    """ Implementation of the interface function print_usage"""
    def print_usage(self):
        print(""" This is a %s connection"""% (self.app_name))

    """ Implementation of custom function for hello connections"""
    def hello_world(self):
        print("These are the specified parameters" )
        print (self.parameters)
  
```

3. Test your custom data connection locally (for more information see *Loading custom connections*)

```

test_params = {"PARAM_1": "foo", "PARAM_2": "bar"}
import cml.data_v1 as cmldata

conn = cmldata.get_custom_connection_from_local(
    "hello-world-conn-dev",
    "my-hello-connection",
    test_params)
conn.print_usage()
  
```

```
conn.hello_world()
```

← Project >_ Terminal Access ☰ Data ✎ Clear ⚡ Interrupt ■ Stop Sessions ▾

Untitled Session  Running

By CDEP CREATED ACCOUNT — Session — 2 vCPU / 4 GiB Memory — a few seconds ago

Session Logs

✖ Collapse  Share  Export PDF

```
> test_params = {"PARAM_1": "foo", "PARAM_2": "bar"}
> conn = cmldata.get_custom_connection_from_local(
    "hello-world-conn-dev",
    "my-hello-connection",
    test_params)

Loaded custom connection my-hello-connection

> conn.print_usage()

This is a my-hello-connection connection

> conn.hello_world()

These are the specified parameters
{'PARAM_1': 'foo', 'PARAM_2': 'bar'}
```

>

4. In Site Administration Data Connections , select New Connection, and fill in the fields as shown below.

New Data Connection ✕

*** Name**

*** Type** ⓘ

🔗 Custom Connection ▼

*** Type Display**

*** Project** ⓘ

Hello World Dev ▼

*** Connection Files**

hello-world-conn-dev ▼

Custom Parameters ⓘ

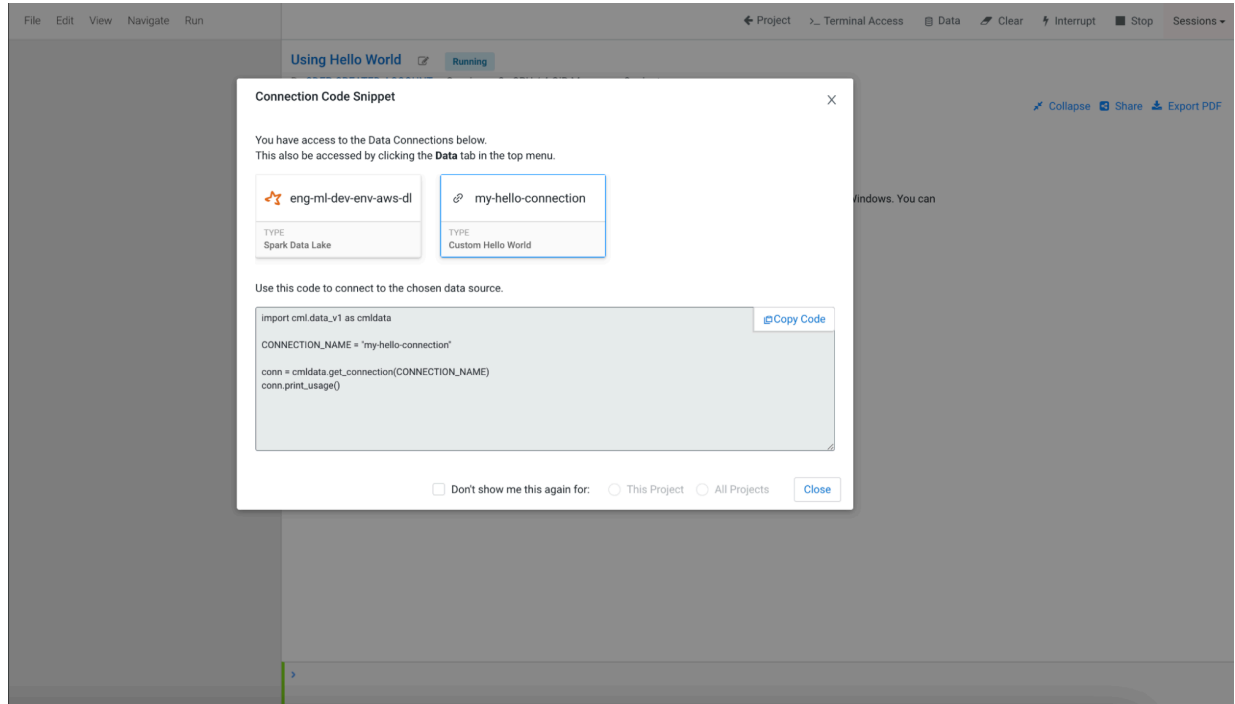
PARAM_1	foo	-	+
PARAM_2	bar	-	+

☒ **Available** ⓘ

CancelCreate

5. See *Data connection management* to set the availability of your newly created custom connection in new and existing Cloudera AI projects.

6. Create a project, and start a session to use your completed Hello World connection.



Note: A default custom connection code snippet is automatically displayed to the user.

[← Project](#) [> Terminal Access](#) [Data](#) [Clear](#) [Interrupt](#) [Stop](#) [Sessions ▾](#)

Using Hello World

Running

By [CDEP CREATED ACCOUNT](#) — Session — 2 vCPU / 4 GiB Memory — 2 minutes ago

[Session](#) [Logs](#) [Collapse](#) [Share](#) [Export PDF](#)

```
> import cml.data_v1 as cmldata
> CONNECTION_NAME = "my-hello-connection"
> conn = cmldata.get_connection(CONNECTION_NAME)

Loaded custom connection my-hello-connection

> conn.print_usage()

This is a my-hello-connection connection
```

>



Note: When exporting custom connection files, git is recommended. Alternatively, download the custom connection directory only, instead of all project files.

Related Information

[Loading custom connections](#)

[Data connection management](#)

[Using data connection snippets](#)

[Using data connection snippets](#)

Loading custom connections

You can instantiate a local connection for testing, using the name of your custom connection directory, a sample connection name, and an optional dictionary of parameters. This local connection object can then be used to test and implement functions in your custom connection.

The following code sample loads custom connection package directories in the same way that the Cloudera AI Data Library imports a registered custom connection when called with `get_connection`:

```
get_custom_connection_from_local(package_name, connection_name, parameters={})
```

Returns a Cloudera AI Custom Data Connection object. For testing in-development Data Connection code.

Parameters:

- `package_name` (str): The accessible package name containing custom connection code to load.
- `connection_name` (str): The connection name to be used in Custom Connection loading.
- `parameters` (dict of str: str): Mapping of custom parameter keys and values that will be loaded by the custom connection code.

Return:

A custom connection object that implements `cml.data_v1.customconnection.CustomConnection`

Usage: `conn = load_custom_connection_source("myconndir", {"HOSTNAME": "my.instance.host.com"})`

Note: When you make changes to your custom connection file, `get_custom_connection_from_local` will dynamically re-import the contents, so the latest code on disk is instantiated for the local connection.

To load any created data connection that is available in the Cloudera AI project, use the `get_connection` instruction.

```
get_connection(dataconnection_name, parameters=None)
```

Usage: `conn = get_connection(connection_name)`

If the specified connection is of type “Custom” a snapshot of the Custom Data Connection files specified at the time of Connection creation is imported.