Lab: Data Access in DSX

January 10th, 2018

Author: Elena Lowery elowery@us.ibm.com

Table of contents

Contents

Overview

In this lab you will learn how to access database and Hortonworks Data Platform (HDP) data sources in DSX. You will learn how to

- Define a database connection for a database that's supported via UI
- Connect to a database data source which is not supported in the UI
- Connect to a non-secure HDP cluster

Required software, access, and files

- To complete this lab, you will need access to a DSX Local cluster and HDP.
- You will also need to download and unzip this GitHub repository: https://github.com/elenalowery/DSX_Local_Workshop

Part 1: Set up a database

In this section we will set up a database in IBM Cloud so that we can test database access from DSX. If you already have an external database that you would like to use, you can skip this section.

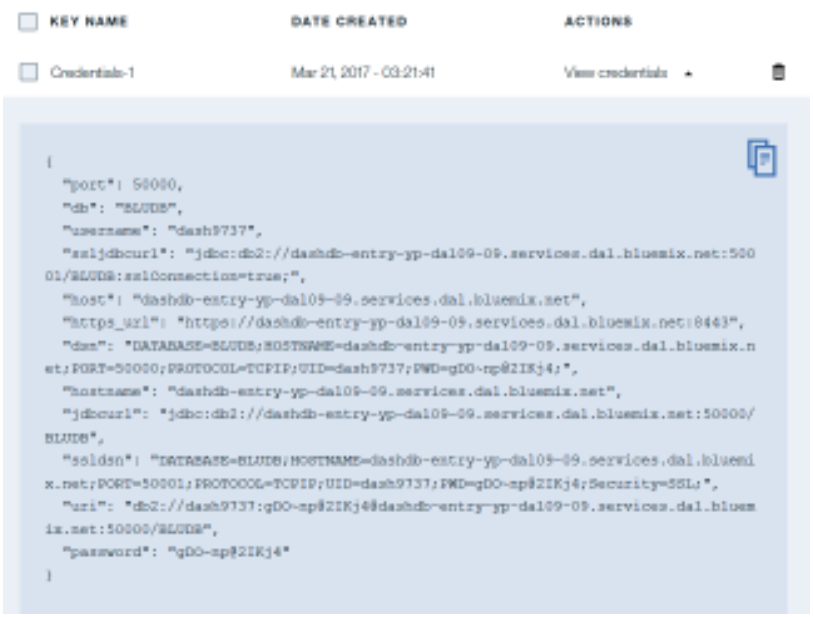
At this time the names *dashDB* and *DB2 on Cloud* are used interchangeably.

Note: If you are not able to create the DB2 on Cloud service, please ask the lab instructor for a pre-configured instance.

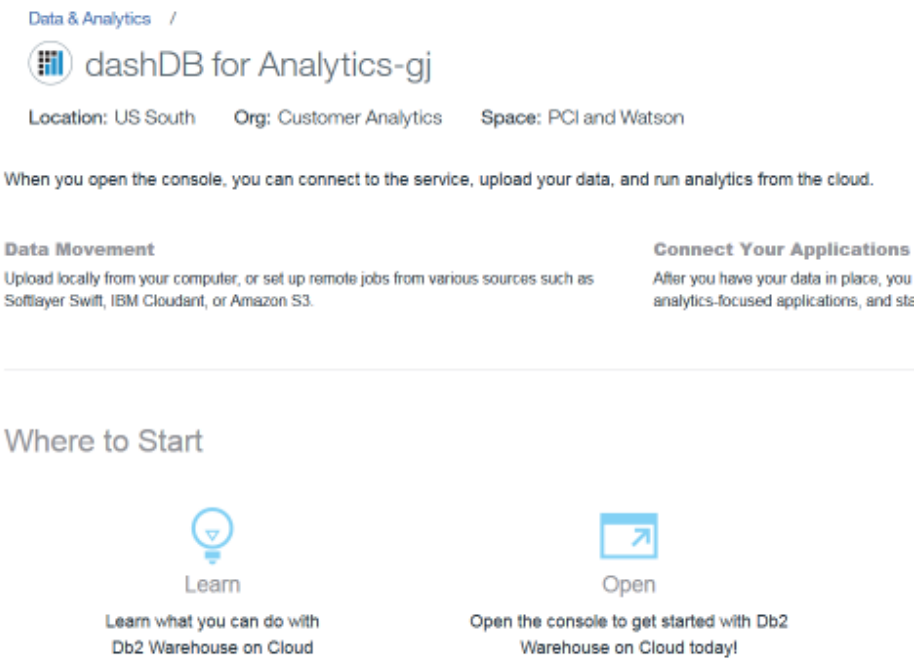
1. Create a DB2 on Cloud service in IBM Cloud

    - Login to Bluemix: bluemix.net
    - Search for "db2 on cloud" and create the service

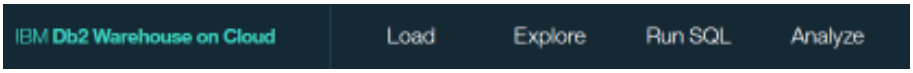
2. Lookup service credentials in Bluemix and save them in a notepad



4. Click Open to open the dashDB console

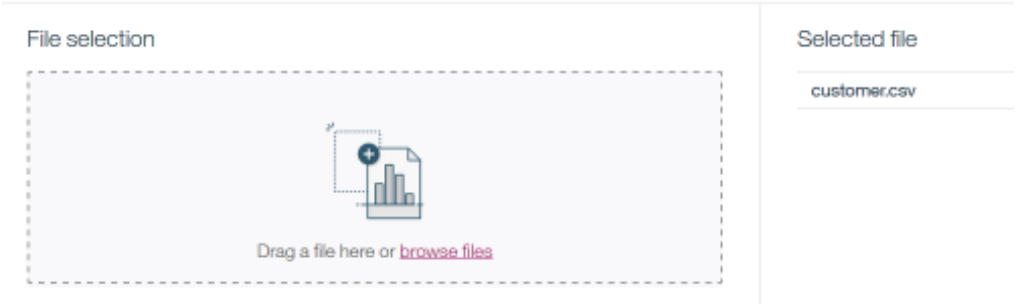


3. Click **Load**



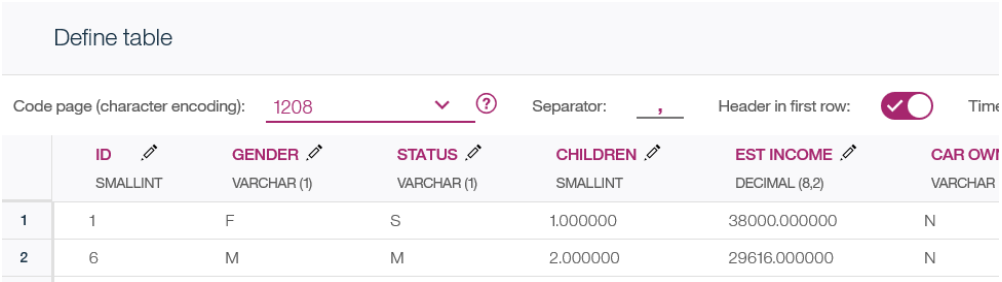
4. Click **Load Data**

5. Select **browse files** and navigate to the *data* folder of the unzipped GitHub repository. Select *customer.csv*. Click **Next**.

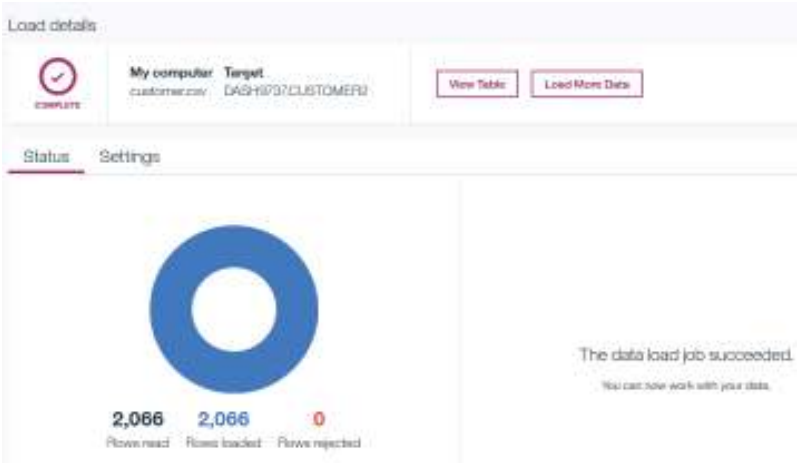


6. Select *Schema* (which will be different than the screenshot in your instance) and click **New Table**. Enter table name *CUSTOMER* and click **Create**. Click **Next**.

7. Leave the default values on the *Define Table* screen. Click **Next**. Then **Begin Load**.



8. If you want to verify that data has been loaded successfully, click **View Table**.



9. If you want to convert all sample notebooks to the database data sources, then repeat the data load steps for all files in the */data* directory.

Part 2: Configure database connection in DSX

In this section we will define a connection in the DSX UI and test it in a notebook.

1. Open a DSX Local project (for example, *DSX_Local_Workshop*) or create a new DSX Local project.

2. Click on **Data Sources**, then **add data source**

3. Enter data source name (for example, dashDB_DS) and fill out the required fields (which you saved from Service Credentials view in IBM Cloud).

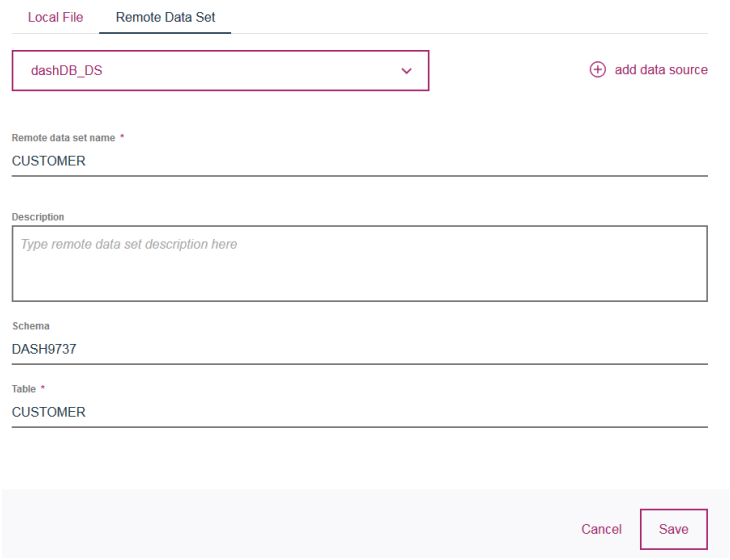
   Do not check the "Shared" checkbox. If you select it, then your credentials will be shared with collaborators on the project.

   Click **Create**.

4. Switch to the **Assets** view and scroll down to **Data Sets**. Click **add data set**. Select the created data source from the dropdown and enter the required fields.

   In our example we gave the same name to the data set as the table name – *CUSTOMER.* Your schema name will be different.

   Click **Save**.



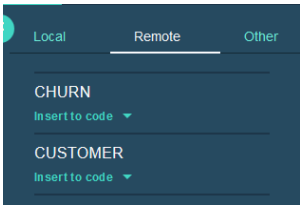
5. If you created tables from other CSV files, create the **Remote Data Source** for each of them.

Notice that the remote data sets are shown as tables.

Part 3: Test Database Connection

1. To test the connection, create a new Jupyter/Python notebook.

2. Click on the data icon, then **Remote** tab and select the **Insert to code** option for one of the remote data sources. You can test both Spark and Pandas data frames.



3. Run the code and make sure data is displayed

```
df1.head()
```

	ID	GENDER	STATUS	CHILDREN	EST_INCOME	CAR_OWNER	AGE	LONGDISTANCE	INTERNATIONAL	LOCAL	DROPPED	PAYMETHOD	LOCALBILLTYPE	LONGDISTANCEBILLTYPE	USAGE	RATEPLAN
0	886	F	M	1	34555.0	N	23.000000	25.77	0.00	8.84	0	CH	FreeLocal	Standard	34.61	4
1	887	F	M	2	11234.7	Y	49.046667	0.69	0.00	112.72	0	CC	Budget	Standard	113.42	2
2	889	F	M	2	36660.9	Y	39.960000	0.00	0.00	3.36	1	CH	Budget	Intnl_discount	3.36	3
3	891	F	M	2	68462.8	N	34.400000	24.39	5.93	60.51	0	CC	FreeLocal	Standard	90.84	1
4	894	F	M	2	72841.3	N	47.020000	10.36	0.00	72.16	0	CC	Budget	Standard	82.53	2

4. If you wish, change the sample notebooks that use .csv to use a database data source.

   Make sure to insert the correct data frame type.
   - *TelcoChurn* notebook uses Spark data frames
   - *CreditCardDefault* notebook uses Spark data frames

   In addition to generating the code, you may need to change variable names. Please check with the lab instructor if you need help with understanding how to modify the code.

Part 4: Configure HDFS and Hive connections in DSX

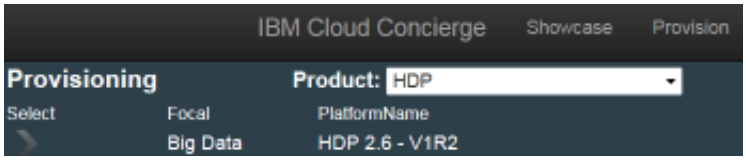
In this section we will define a non-secure connection to HDFS and Hive in DSX and access Hadoop data sources in a notebook.

Note: It's possible to configure a secure connection (see official documentation). We are using a non-secure connection because we can quickly set it up in a demo environment.

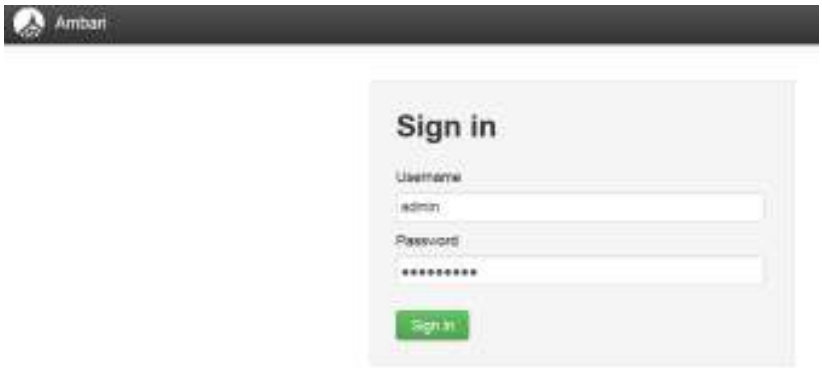
These instructions for loading data to HDFS and Hive in HDP are specific to IBM Cloud Concierge and Fyre environments. If you're working in a different environment, please check with the Hadoop administrator.

Please note that both, DSX Local and HDP must be in the same environment (Cloud Concierge or Fyre) when setting up the unsecure connection. Port 8020, which is used for unsecure connection, is not open, and the only way to get to the systems is via private IP.

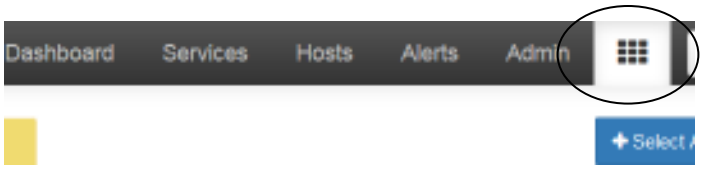
1. If the HDP instance was not provided, log it to **Cloud Concierge** (https://demo.ibmcloud.com) provision an HDP image.



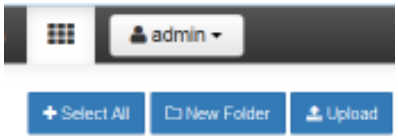
2. Log in to HDP: Login to Ambari (HDP admin console): *https://<hostname>:8080* with *admin/IBMDem0s!*



3. Select Files View from the menu in the right corner.



4. In the Cloud Concierge environment, by default the admin user can only upload to /tmp directory. Navigate to the */tmp* directory and upload any .csv file that we used in the previous labs (for example, files used for *Telco Churn*, *Credit Card Default*, or *Data Science for Automotive* use cases), for example, *churn.csv* and *customer.csv* that are used in the Telco Churn notebook.



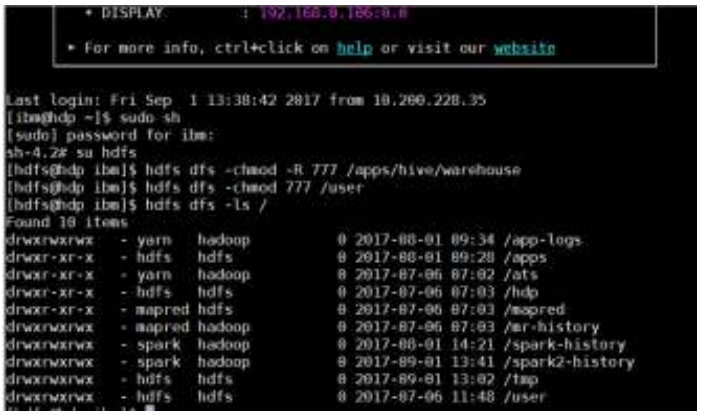
Name	Size	Last Modified	Owner	Group	Permission
ambari-qa	--	2017-11-22 14:03	ambari-qa	hdfs	drwx------
churn.csv	19.6 kB	2018-01-09 16:54	admin	hdfs	-rw-r--r--
customer.csv	273.0 kB	2018-01-09 16:54	admin	hdfs	-rw-r--r--

5. If you would like to test connectivity to Hive, then complete the steps to create Hive tables.

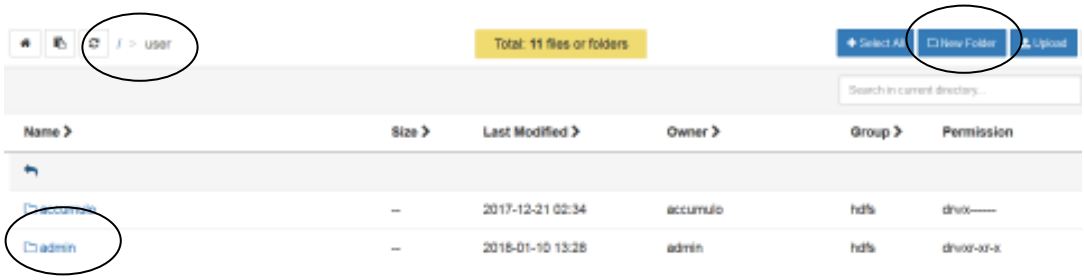
   If you are working as admin, you need to make sure that you have permissions to create folders and that */user/admin* folder exists

   Run these commands (from ssh) to allow admin to create folders in HDFS

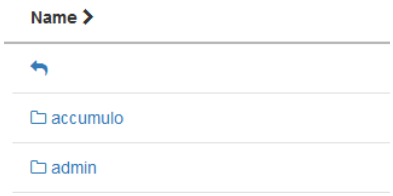
   - su hdfs
   - hdfs dfs –chmod –R 777 /apps/hive/warehouse
   - hdfs dfs –chmod 777 /user



Navigate to the File view and create *admin* folder in */user*



6. Navigate to the File View (click on the ) and create admin folder in */user*



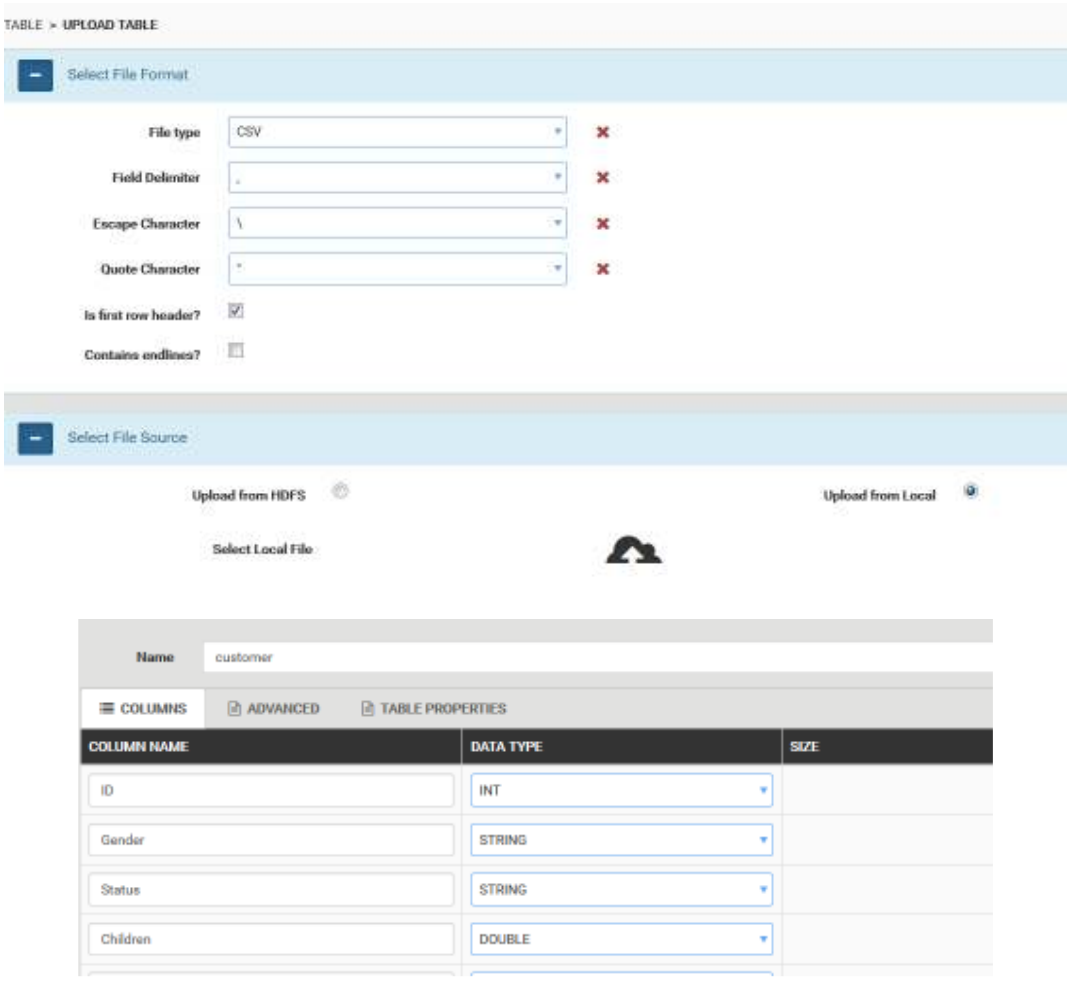
If you don't complete this step, you're likely to see this error: https://community.hortonworks.com/questions/80603/hdfs020-could-not-write-file-useradminhivejobshive.html

7. Switch to **Hive View 2.0**

8. Click on **Tables**. Click the **+** icon, then select **Upload Table**

9. Review/modify settings. If you are using one of the files from the sample notebooks, the first column is a header.

   After selecting the file, verify that all data types are correct in the **Preview**. Click **Create** and don't refresh the page (it'll show status).



You have finished loading sample data into Hive.

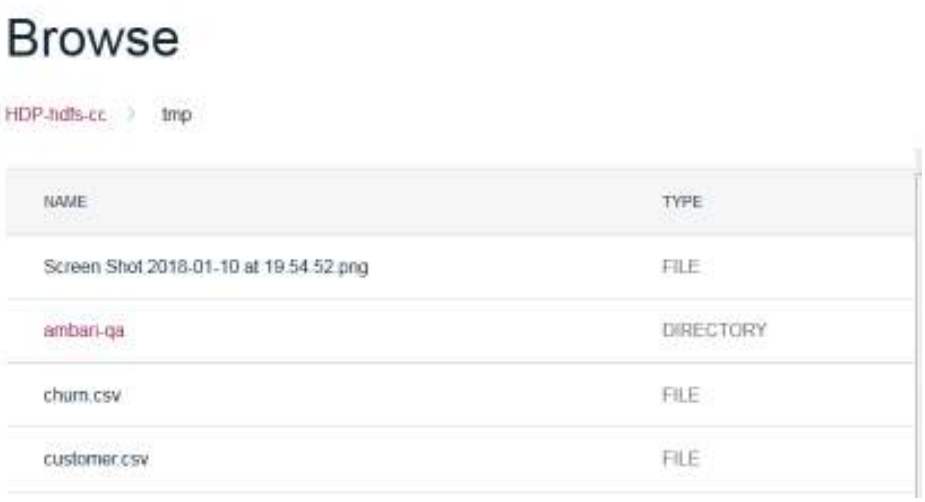
10. Now we are ready to define HDFS and Hive connections in DSX. Navigate to **Project** view for one of your projects (for example, *DSX_Local_Workshop*) and select **Data Sources**.

11. Click **add data source**. Select *HDFS - HDP* from the dropdown. Replace *IP addresses* (IP address of your HDP cluster) in the other fields, but don't change the ports.

    Click **Create**.

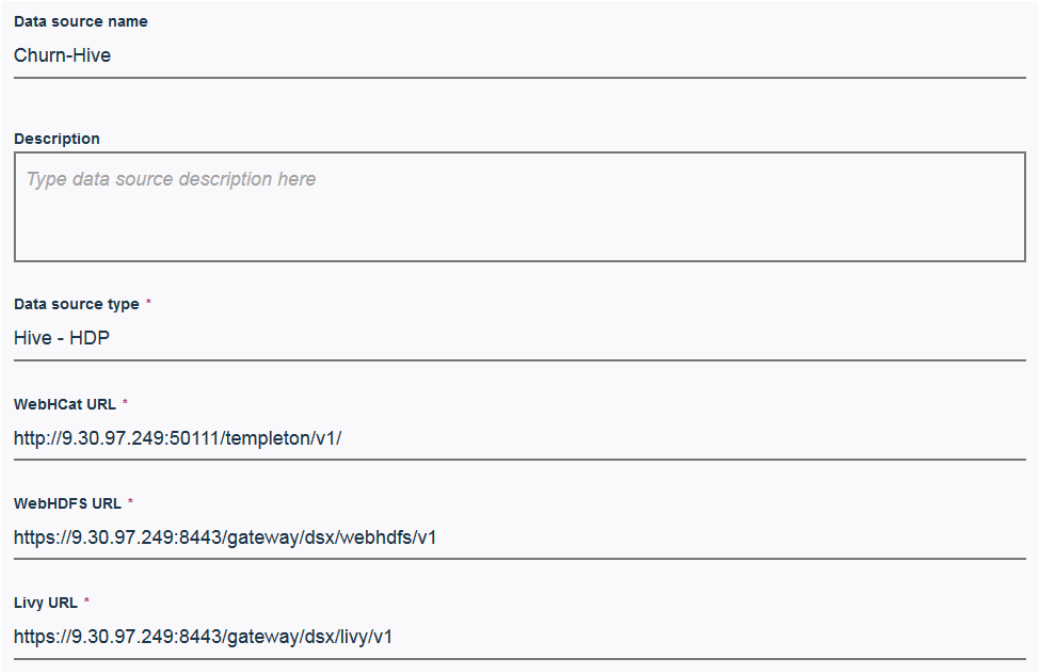
12. Navigate back to the **Data sources** view in the project and click on the created data source.

13. Click **add data set**, then browse to the file you uploaded to HDFS. Select the file, click **Open**, then click **Create**.



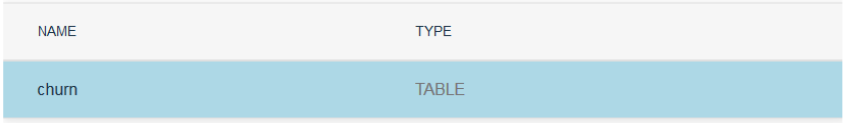
14. Click **add data source**. Select *Hive - HDP* from the dropdown. Replace *IP addresses* (IP address of your HDP cluster) in the other fields, but don't change the ports.

    Click **Create**.



15. Navigate back to the **Data sources** view in the project and click on the created data source.

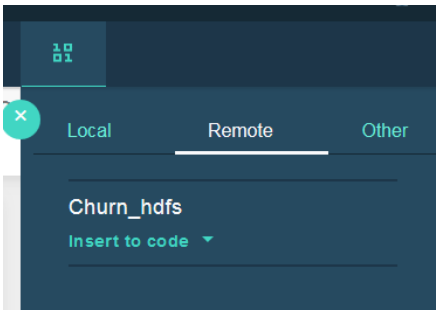
16. Click **add data set**, then browse to the file you uploaded to Hive. Select the file, click **Open**, then click **Create**.



Part 5 : Test HDFS and Hive connection

1. Create a new notebook and use code generation similar to database access code generation to test connectivity.

   *Hint: make sure to select a the **Remote** tab on data tab*



2. If you loaded data sources for one of the sample notebooks, try modifying notebooks to use HDFS and Hive data sources.