Using the Python Connector for Databricks SQL

Ryan Chynoweth

4 min read · Apr 25, 2022

Introduction

Databricks is the top platform for building and deploying **all** data applications: streaming data, machine learning, business intelligence, and ad-hoc analysis. Working within the Databricks ecosystem is easy and it provides most of the tools that data practitioners require, however, there are circumstances where a custom application is desired.

As an example, let us assume I am an appliance manufacturer and I am processing claim information. A customer submits a claim and we assign a technician to go out and complete the request. Once completed the technician will fill out a form to provide more information on the claim. Ultimately we would like to classify our claims into different categories and label the issue which allows us to understand ways we can improve our product or issue a recall for defective parts. Due to human nature, technicians may mislabel the tickets causing “dirty data”. Luckily our analysts are reviewing this information, but they need a way to correct (update) the label properly. In addition, we would like to avoid giving direct access to the tables for them to run DDL commands on. Our solution is to create a small client application to give them the ability to “reclassify” claims into the correct category. This application has a lower transactional SLAs, meaning the application does not require an OLTP database. For all purposes, this is an analytic workload that requires low latency and single row updates so that we are able to ensure high data quality for data scientists, analysts, and engineers who are creating data solutions.

Enter the Python Connector for Databricks SQL

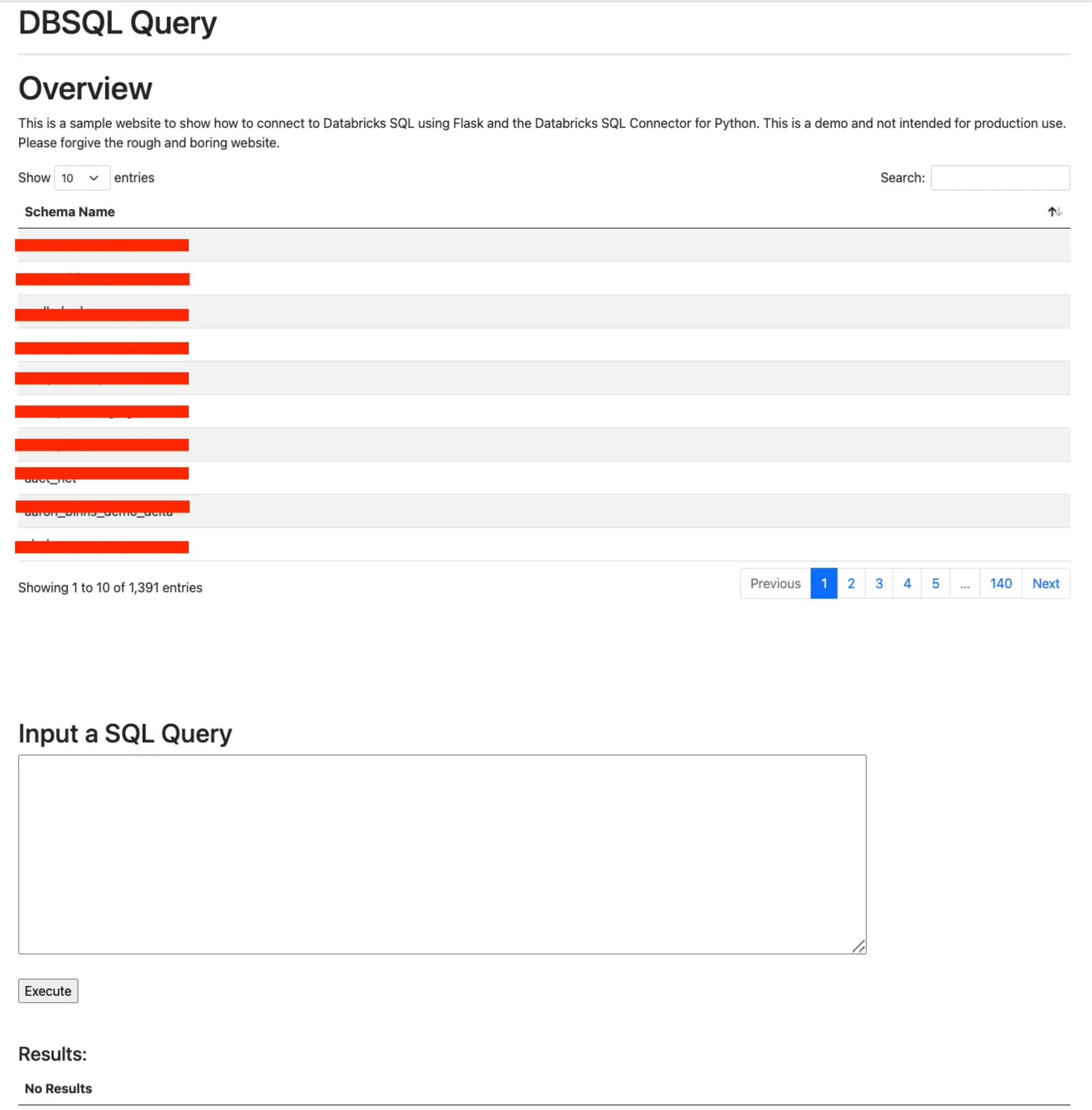
The Databricks SQL Connector for Python is a PyPi library which allows applications in Python to execute SQL commands directly on a Databricks Cluster or SQL Endpoint.

Why is this cool?

Typically connections to databases, data warehouses, and even Databricks require drivers to be installed. At Databricks we have our drivers, but this is a dependency that at times can be more annoying than helpful (even if it is common for all databases). If you are building an application in Python then you are no longer required to use one of the drivers. Simply install our library and connect directly to Databricks to execute all your SQL commands.

Sample Application

The entire purpose of this article is to provide an example application that uses the Databricks SQL Connector for Python. In the introduction I used a manufacturing scenario to explain how custom applications are needed to connect to Databricks directly. So I built a very simple demo application using Python 3.8, flask 2.1.1, and databricks-sql-connector 1.0.2. The application is shown below. In short, it displays all your schemas and allows users to submit SQL queries against your Databricks instance.



DBSQL Query

Overview

This is a sample website to show how to connect to Databricks SQL using Flask and the Databricks SQL Connector for Python. This is a demo and not intended for production use. Please forgive the rough and boring website.

Schema Name
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]

Showing 1 to 10 of 1,391 entries

Input a SQL Query

Results:

No Results

This example is simple. I use a Databricks PAT token to connect to a SQL Endpoint and allow users to submit free formed SQL queries against that endpoint. Each time the user submits code it then displays results as needed below as a table. We also, by default, display all the schemas that are available to the user at the top of the application.

Most apps will likely require user authentication and validation of access rights (i.e. maybe a global PAT is not a great idea in production). As a simple example I hope you can see the value of this type of application.

Running the application

The source code is available in this GitHub repository. This solution was developed using an anaconda environment on my local machine and has been deployed in Azure using a Docker container.

Requirements:

- Docker (optional)
- Python 3.8 (other versions may work but I only tested on 3.8)
- flask 2.1.1
- databricks-sql-connector 1.0.2

The application requires a configuration file in order to supply some of the secrets required to connect to Databricks. It is recommended to save the configuration file as `app/app_config.conf` and should follow this syntax:

```
[DEV]
server_hostname=adb-1234567890123456.11.azuredatabricks.net
```

```
http_path=/sql/1.0/endpoints/1234567890123456
access_token=1234567890123456789012345678901234567890123456
```

To run the application locally via a command line host, complete the following (assuming you have the dependencies installed):

1. Clone the repository.
2. Open a command line and navigate to the `app` directory.
3. Run the following commands:

```
export FLASK_APP=main

export FLASK_ENV=development

flask run
```

4. Navigate to `127.0.0.1:5000`

To run the application using Docker, complete the following (assuming you have the dependencies installed):

1. Navigate to the base directory of the application i.e. `WebApp` . This is where the Docker file is located.
2. Run the following commands:

```
docker image build -t db_web_app .
docker run -p 5000:5000 -d db_web_app
```

3. Navigate to `localhost:5000` in a web browser. Note that the URL could be different depending on where you run the container. In this example it is running on my local computer.

Conclusion

Connecting to Databricks using Python is super simple. Let me know if you use the library and what solutions you are building with it.

Disclaimer: these are my own thoughts and opinions and not a reflection of my employer

Python Databricks Databricks Sql Databricks Connector

Written by Ryan Chynoweth

Edit profile

312 Followers

Senior Solutions Architect Databricks — anything shared is my own thoughts and opinions

More from Ryan Chynoweth

DELTA SHARING

en standard for secure data sh

dustry's first open protocol for secure data sharing, mak

ner organizations regardless of which computing platfor

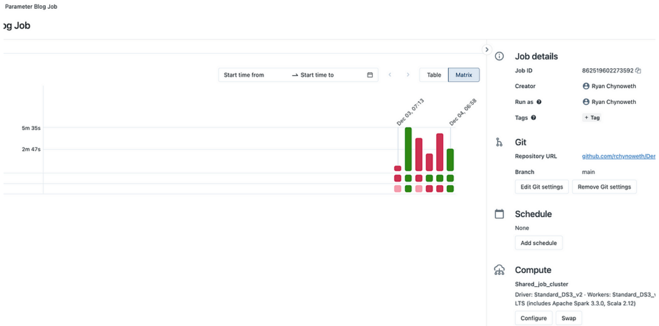
Ryan Chynoweth

Delta Sharing: An Implementation Guide for Multi-Cloud Architecture

Introduction

8 min read · 3 days ago

3 2



Parameter Blog Job

›g Job

Job details

Job ID 862519602273592

Creator Ryan Chynoweth

Run as Ryan Chynoweth

Tags

Git

Repository URL github.com/rchynoweth/Der

Branch main

Schedule

None

Compute

Shared_job_cluster

Driver: Standard_DS3_v2 · Workers: Standard_DS3_
LTS (includes Apache Spark 3.3.0, Scala 2.12)

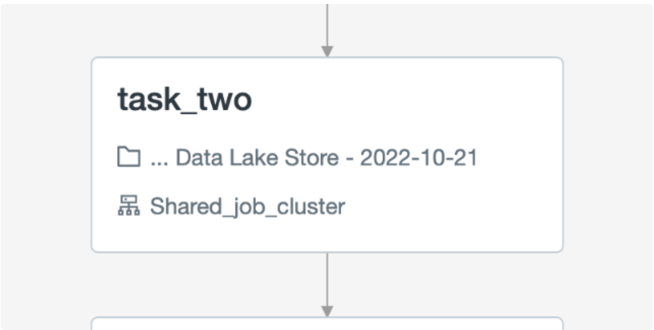
Ryan Chynoweth

Task Parameters and Values in Databricks Workflows

Databricks provides a set of powerful and dynamic orchestration capabilities that are...

11 min read · Dec 7, 2022

50 3



task_two

... Data Lake Store - 2022-10-21

Shared_job_cluster

Ryan Chynoweth

Converting Stored Procedures to Databricks

Special thanks to co-author Kyle Hale, Sr. Specialist Solutions Architect at Databricks.

14 min read · Dec 29, 2022

Ryan Chynoweth

Recursive CTE on Databricks

Introduction

3 min read · Apr 20, 2022

116 5

33

See all from Ryan Chynoweth

Recommended from Medium

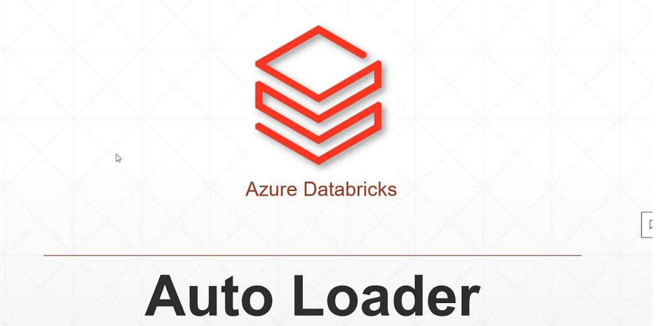
Daan Rademaker

Do-it-yourself, building your own Databricks Docker Container

In my previous LinkedIn article, I aimed to persuade you of the numerous advantages o...

7 min read · Oct 16, 2023

26



SIRIGIRI HARI KRISHNA in Towards Dev

Auto Loader

Autoloader simplifies reading various data file types from popular cloud locations like...

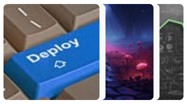
5 min read · Dec 9, 2023

4 1

Lists

Coding & Development

11 stories · 355 saves



Predictive Modeling w/ Python

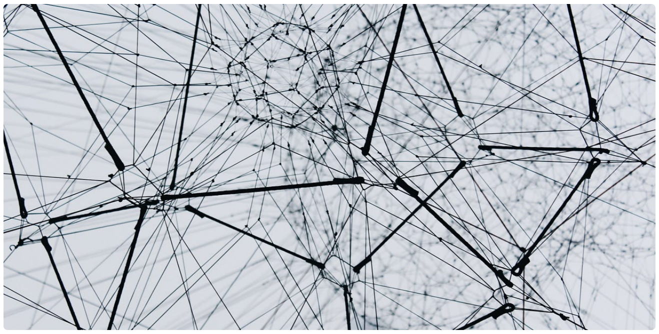
20 stories · 749 saves

Practical Guides to Machine Learning

10 stories · 865 saves

ChatGPT

23 stories · 371 saves



Subham Khandelwal in Dev Genius

PySpark—Optimize Joins in Spark

Shuffle Hash Join, Sort Merge Join, Broadcast joins and Bucketing for better Join...

8 min read · 4 days ago

65

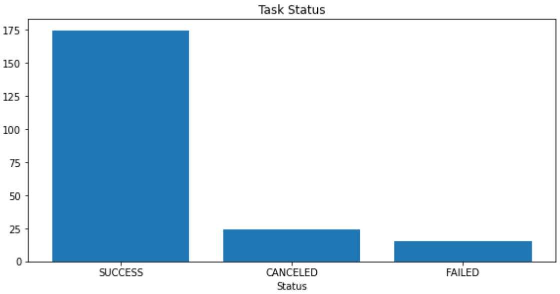
Pranavk

Python Loading Dataframe in SQL Server using BCP

Introduction

3 min read · Oct 11, 2023

55



Shahid haider

Databricks Job Runs Data monitoring Tool

Manasreddy

How to pass: Databricks Data Engineer Professional Certification

This script is designed to fetch and process job runs data from an Azure Databricks...

5 min read · Oct 14, 2023

13

Conquering the Databricks Data Engineer Professional Exam: A Definitive Guide

3 min read · Aug 24, 2023

8

See more recommendations