

Authenticate between Spark and SQL pool in Azure Synapse Analytics

Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it's likely that if you try this out you will encounter issues in your system.

Another way to authenticate is using SQL Authentication, instead of Azure Active Directory (Azure AD) with the Azure Synapse Apache Spark Pool to Synapse SQL connector.

Currently, the Azure Synapse Apache Spark Pool to Synapse SQL connector does not support a token-based authentication to a dedicated SQL pool that is outside of the workspace of Synapse Analytics. In order to establish and transfer data to a dedicated SQL pool that is outside of the workspace without Azure AD, you would have to use SQL Authentication.

To read data from a dedicated SQL pool outside your workspace without Azure AD, you use the Read API. The Read API works for Internal tables (Managed Tables) and External Tables in the dedicated SQL pool.

The Read API looks as follows when using SQL Authentication:

```
val df = spark.read.  
option(Constants.SERVER, "samplews.database.windows.net").  
option(Constants.USER, <SQLServer Login UserName>).  
option(Constants.PASSWORD, <SQLServer Login Password>).  
sqlanalytics("<DBName>.<Schema>.<TableName>")
```

The parameters it takes in are:

- **Constants.Server:** specify the URL of the server
- **Constants.USER:** SQLServer Login UserName
- **Constants.PASSWORD:** SQLServer Login Password
- **DBName:** the name of the database.
- **Schema:** the schema definition such as dbo.
- **TableName:** the name of the table you want to read data from.

In order to write data to a dedicated SQL Pool, you use the Write API. The Write API creates the table in the dedicated SQL pool, and then uses Polybase to load the data into the table that was created.

The Write API using SQL Auth looks as follows:

1
2
3
4
5

1

```
df.write.  
option(Constants.SERVER, "samplews.database.windows.net").  
option(Constants.USER, <SQLServer Login UserName>).  
option(Constants.PASSWORD, <SQLServer Login Password>).  
sqlanalytics("<DBName>.<Schema>.<TableName>", <TableType>)
```

The parameters it takes in are:

- **Constants.Server:** specify the URL of the server
- **Constants.USER:** SQLServer Login UserName
- **Constants.PASSWORD:** SQLServer Login Password
- **DBName:** the name of the database.
- **Schema:** the schema definition such as dbo.
- **TableName:** the name of the table you want to read data from
- **TableType:** specification of the type of table, which can have two values.
Constants.INTERNAL - Managed table in dedicated SQL pool
Constants.EXTERNAL - External table in dedicated SQL pool