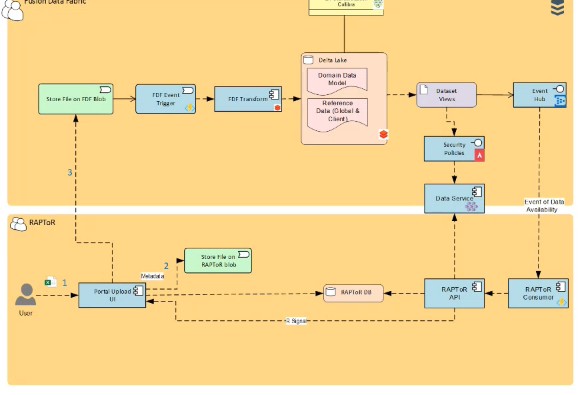
FUSION DATA FABRIC-RAPTOR APPLICATION FRONT END UPLOAD FILE PATTERN



Bottom Amber colour is Raptor

1)User is coming on to the Raptor Portal and they are uploading a file which can be a Excel ,CSV or any kind of file as these are mostly template driven

2) Clients like Goldman are contacted for getting the required data , once they get all the data they upload it in the front end portal named as Raptor where Raptor itself has its own mini Blob and it writes the data in to the Fusion data Fabric

3) Once the file is placed in FDF there is a **Event grid Trigger** Azure function and the clusters which are already alive there are going to be used for query executions and the data gets written to the Delta lake layer Schema . Schema evolution happens here.

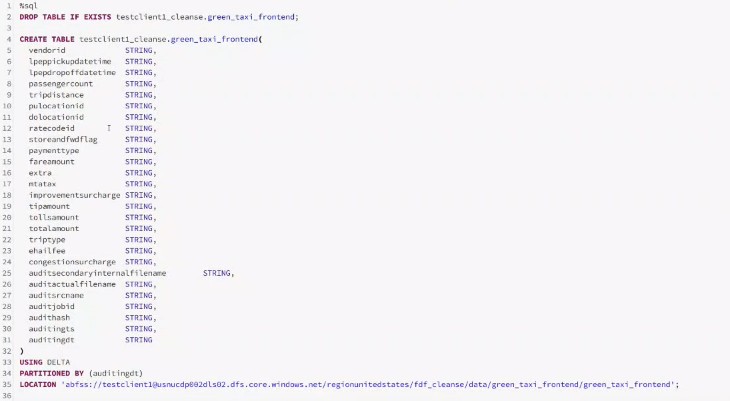
4) FDF transform job will write a notification message which will have details like client name .engagement name and file name . It doesnot contain the actual data

5) Any consumer Fusion data fabric can subscribe to the relevant event hubs. and once they are alerted they can basically call the Fusion data FabricAPI .

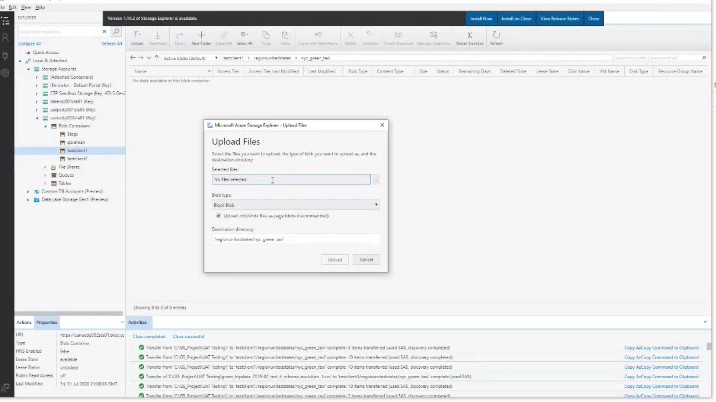
6) Here adapter has a azure function running which is eveny hub triggererd so as soon as a event comes raptor gets activated which call its api and data gets populated in to the Raptor Database

Coding :

Here is the code for creating a data table



Uploading a file in blob





AS here it is mentioned that job run has been completed and it took 26 seconds for its completion





First it establishes a jdbc connection



After that try to get the metadata details



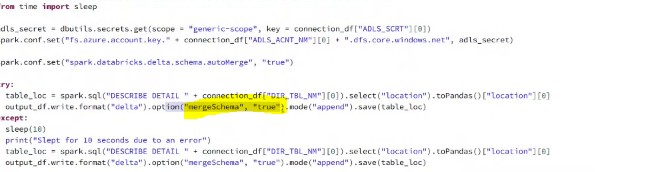
Checking whether the input file is csv or xls



Check whether the file is excel and if it is excel it converts the file to CSV. After converting to csv read the csv file as dataframe



And now we are considering the metadata and we are getting the list of scrubbing details which we need to execute in order to clean up the data set and dynamically allocate the column names and write the dataframe to the target table



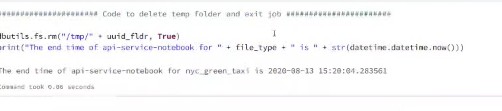
Once the Write happens successfully it sends out the notification to the eventhub



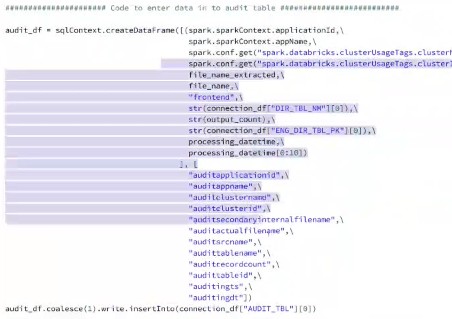


This is the actual job endtime from the flow diagram perspective. And the net steps are moving the raw file in to ADLS zone and writing the audit table and the temporary directory is deleted





writing the audit table



Deleting the temporary directory

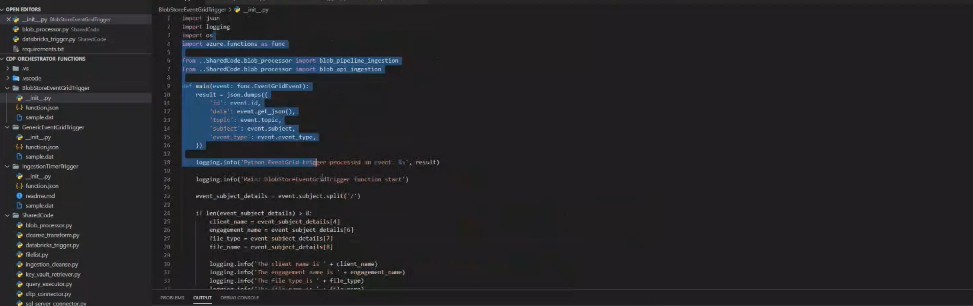
As stated before it took around 26 seconds



An average of 30 seconds is maintained for this job even if there are number of files

BLOB STORE EVENT GRID TRIGGER

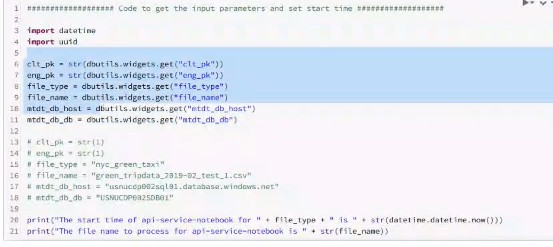
Entire message is json and we are extracting certain fields from the sybject sothe way the data comesin ,we know the parameter and their position as well



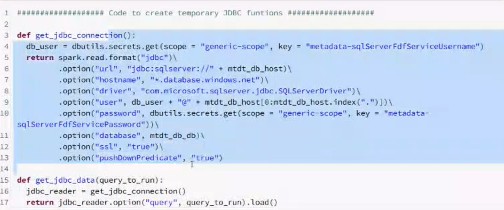
Whatever fileuploads need to happen will happen here according to the requirement and pipeleine ingestion extactly similar to SFTP process like if a particular user uploads a file in blob they are putting it under pipeline ingestion

BLOB PIPELINE INJECTION

It is a counter part over SFTP. Here we have our BLOB API query to figure out which job needs to be executed and which notebook needs to be executed and after these query executions databricks will be executed and with the existing parameters job gets started on the databricks side. Since it is a already fixed job there is no need of cluster creation . the below code what it does here is parematers like client primarykey,engagement,primary key filetype,file name SQL server ,metadata,only daily hist and DB database are captured here



when the above things are done implementing JDBC is the next step



It fires up the query and it gets the entire details like hubdetails, event hub named access keys,mentioned in below screenshot . It is similar to the SFTP process just slight columns here and there . Files are passed to the corresponding locations and from there using the metadata file gets loaded in to the corresponding table

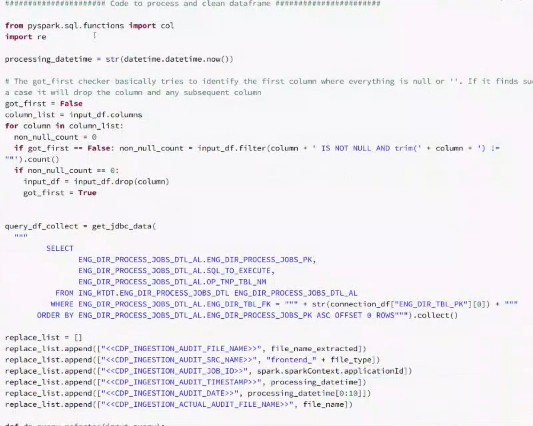


Next is to convert the excel file in to csv multiple sheets

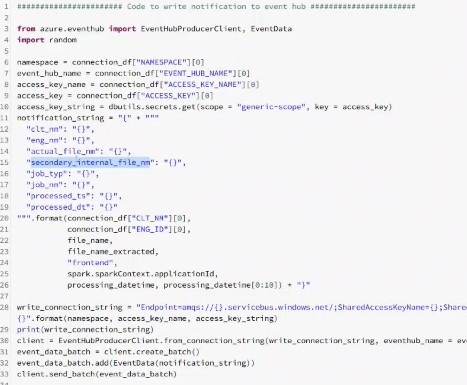


Convert the csv fie in to a dataframe

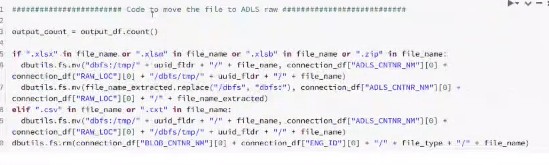


Scrubbing the code and cleaning the dataframe 

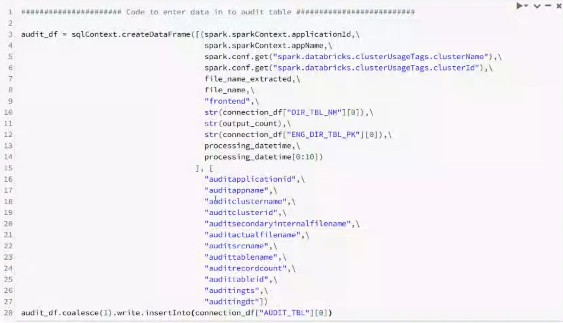
Here is a code to write to write notification to eventhub



Here is a code to move data to ADLS



Here is a code to write the data to the audittable

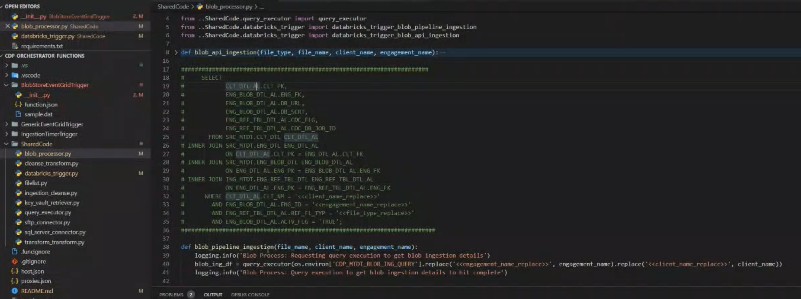


Code to delete temp folder



It connects the database and run the below sql queries in the code alone and no cdc as of now





\* Getting the data from the corresponding folder and load it in to the corresponding table respectively

\* Converting the excel files like xls,xlsx to csv

\* Ignoring null values and special characters for standerzing

\* Transformation performed using sparksql

\* Loading the requested data in to the audit table