## Parse data in a given dataset

Expected Time - 2 hours approx

Consider a scenario where we received files with multiple formats, and the goal is to import the data into a relational database.

Can you come with a solution that helps to parse the files and convert them into tabular format?

|  |  |  |  |
| --- | --- | --- | --- |
| Column01 | Column02 | Column03 | Column04 |
| Field value | Field value | Field value | Field value |
| Field value | Field value | Field value | Field value |



Additional instructions

* Please use python; just native libraries and only .py files are required
* The expected output is three files in .txt format, using tab as delimiter

Please find the solutions below



## System design

Expected Time - 3 hours approx

Consider the scenario of an offline recommender system. A user is going to navigate one of the many online stores distributed around the globe. And based on their cookie id, we are going to provide a product recommendation.

i.e. A user in Thailand with cookie id 383DFF4949 visit the HP local store, and we recommend the product P0001

Assumptions:

1.- The HP store will call our API using the cookie id as a parameter, and we return the product id in a json format.

2.- We have over 20 online stores distributed around the globe

3.- Recommendations are provided by the data scientist team daily in a txt file dropped in an S3.

4.- Use has many services as required

5. Expected output will be a ppt, doc or pdf diagram.

Please use the official icons provided by AWS

<https://aws.amazon.com/architecture/icons/>

Please find the recommendation system below



## Design pattern

Expected Time - 1 hour approx

Consider the scenario of ingesting data from an SQS into an RDS Aurora Postgres

Please review the four different architectures and briefly describe the ideal scenario for each one.

1.- SQS->Lambda->RDS

2.- SQS->Lambda->RDS proxy->RDS

3.- SQS->Lambda->Kinesis->Lambda->RDS

4.- SQS->Lambda->Kinesis->Lambda->RDS proxy->RDS

Please find reviews below

1. A scheduled Lambda function is executed in predefined intervals and can consume all the SQS messages that were produced during that specific interval. Once it processes all the messages, it can create a batch upload using API calls. The Lambda function will contain code to access the SQS and to execute upload to RDS
2. RDS Proxy helps us to manage a large number of connections from Lambda to an RDS database by establishing a warm connection pool to the database. Your Lambda functions can scale to meet your needs and use the RDS Proxy to serve multiple concurrent application requests. This reduces the CPU and Memory requirements for your database, and eliminates the need for connection management logic the code.
3. For every message that SQS generates the lambda code will read the messages and pass it to Kinesis stream from which multiple lambdas/ single lambda can read data and store in RDS, this will improve bottleneck created by single lambda from SQS to RDS
4. This is a improvement over design in step 3 by adding RDS proxy we can hadle multiple connection pool to DB, sometimes it will be a overkill since Kinesis also helps us manage multiple connections along with limited storage buffer.