

Test evidences

Project: Phsp-Dlg-Python-Test

Author: Paulo Henrique Silva Pinto

Date: August 30, 2020

Index

About.....	3
Procedure.....	3
1 - Deploying the code to AWS	3
Input.....	3
Expected	3
Output.....	3
2 – Subscribing to the SNS topic	6
Input.....	6
Expected	6
Output.....	6
3 – Uploading csv files.....	7
Input.....	7
Expected	7
Output.....	7
4 – Lambda Execution	8
Input.....	8
Expected	8
Output.....	8
5 – Data Validation.....	10
Input.....	11
Expected	11
Output.....	11
6 – Forcing an error.....	13
Input.....	13
Expected	13
Output.....	13

About

This document describes the procedure used to test the project and evidences its results.

Procedure

1 - Deploying the code to AWS

Input

The code was deployed to AWS by running:

➤ `.\infra\deploy.ps1 dev`

Expected

AWS resources created successfully.

Output

```
PS D:\Projetos\Repositórios\phsp-dlg-python-test\infra> .\deploy.ps1 dev

#--- Initiating CloudFormation Deployment ---#

Reformatting Python code.
Checking python code.

Verifying if bucket phsp-dlg-artifacts-dev exists.
make_bucket: phsp-dlg-artifacts-dev

Verifying Lambda layers:
[2020-08-30 23:55:16,386][INFO][botocore.credentials][load] Found credentials in environment variables.
[2020-08-30 23:55:16,894][INFO][root][main] Verifying if lambda layer awswrangler-layer-1.6.0-py3.6.zip exists....
[2020-08-30 23:55:17,682][INFO][root][main] Lambda layer does not exists. Downloading....
[2020-08-30 23:55:19,329][INFO][root][main] Uploading Lambda layer to s3....
[2020-08-30 23:57:22,417][INFO][root][main] Upload completed.

Building CloudFormation Template
Building function 'CsvToParquetFunction'
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource
Building function 'S3RawBucketEventNotificationFunction'
Running PythonPipBuilder:ResolveDependencies
Running PythonPipBuilder:CopySource

Build Succeeded

Built Artifacts : .aws-sam\build
Built Template : .aws-sam\build\template.yaml

Commands you can use next
=====
[*] Invoke Function: sam local invoke
[*] Deploy: sam deploy --guided
```

```
[*] Deploy: sam deploy --guided
```

```
Deploying with following values
```

```
=====
```

```
Stack name      : phsp-dlg-datalake-dev
```

```
Region         : us-east-1
```

```
Confirm changeset : False
```

```
Deployment s3 bucket : phsp-dlg-artifacts-dev
```

```
Capabilities    : ["CAPABILITY_IAM", "CAPABILITY_AUTO_EXPAND"]
```

```
Parameter overrides : {'StackName': 'phsp-dlg-datalake-dev', 'AwsRegion': 'us-east-1', 'Environment': 'dev', 'S3RawBucketName': 'phsp-dlg-datalake-raw-dev', 'S3AnalyticsBucketName': 'phsp-dlg-datalake-analytics-dev', 'S3ArtifactsBucketName': 'phsp-dlg-artifacts-dev', 'CsvToParquetSnsTopicName': 'csv-to-parquet-sns-topic-dev'}
```

```
Initiating deployment
```

```
=====
```

```
Uploading to 8802b6f1b80867332b68bb35023ee548 4439 / 4439.0 (100.00%)
```

```
Uploading to 0b287098ab7be87e189985d8265c4a8e 6689675 / 6689675.0 (100.00%)
```

```
Uploading to 05a4665c4d09461f9af2b16d8b5192d4.template 6784 / 6784.0 (100.00%)
```

```
Waiting for changeset to be created..
```

```
CloudFormation stack changeset
```

Operation	LogicalResourceId	ResourceType
+ Add	BucketNotificationRole	AWS::IAM::Role
+ Add	CsvToParquetFunction	AWS::Lambda::Function
+ Add	CsvToParquetRole	AWS::IAM::Role
+ Add	CsvToParquetSnsTopic	AWS::SNS::Topic
+ Add	GlueAnalyticsDatabase	AWS::Glue::Database
+ Add	RawCsvToParquetInvokePermission	AWS::Lambda::Permission
+ Add	RawCsvToParquetLambdaTrigger	Custom::LambdaTrigger
+ Add	S3AnalyticsBucket	AWS::S3::Bucket
+ Add	S3RawBucketEventNotificationFunction	AWS::Lambda::Function
+ Add	S3RawBucket	AWS::S3::Bucket
+ Add	WranglerLambdaLayer	AWS::Lambda::LayerVersion

```
Changeset created successfully. arn:aws:cloudformation:us-east-1:873206972289:changeSet/samcli-deploy1598842697/266e0abd-b2b7-47d0-a000-f776856cc9df
```

```
2020-08-30 23:58:26 - Waiting for stack create/update to complete
```

```
CloudFormation events from changeset
```

ResourceStatus	ResourceType	LogicalResourceId	ResourceStatusReason
CREATE_IN_PROGRESS	AWS::S3::Bucket	S3AnalyticsBucket	-
CREATE_IN_PROGRESS	AWS::IAM::Role	BucketNotificationRole	-
CREATE_IN_PROGRESS	AWS::Glue::Database	GlueAnalyticsDatabase	-
CREATE_IN_PROGRESS	AWS::S3::Bucket	S3RawBucket	-
CREATE_IN_PROGRESS	AWS::S3::Bucket	S3AnalyticsBucket	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::SNS::Topic	CsvToParquetSnsTopic	-
CREATE_IN_PROGRESS	AWS::S3::Bucket	S3RawBucket	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::LayerVersion	WranglerLambdaLayer	-
CREATE_IN_PROGRESS	AWS::IAM::Role	BucketNotificationRole	Resource creation Initiated
CREATE_COMPLETE	AWS::Glue::Database	GlueAnalyticsDatabase	-
CREATE_IN_PROGRESS	AWS::Glue::Database	GlueAnalyticsDatabase	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::SNS::Topic	CsvToParquetSnsTopic	Resource creation Initiated
CREATE_COMPLETE	AWS::Lambda::LayerVersion	WranglerLambdaLayer	-
CREATE_IN_PROGRESS	AWS::Lambda::LayerVersion	WranglerLambdaLayer	Resource creation Initiated
CREATE_COMPLETE	AWS::SNS::Topic	CsvToParquetSnsTopic	-
CREATE_COMPLETE	AWS::IAM::Role	BucketNotificationRole	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	S3RawBucketEventNotificationFunction	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Function	S3RawBucketEventNotificationFunction	-
CREATE_COMPLETE	AWS::Lambda::Function	S3RawBucketEventNotificationFunction	-
CREATE_COMPLETE	AWS::S3::Bucket	S3AnalyticsBucket	-
CREATE_COMPLETE	AWS::S3::Bucket	S3RawBucket	-
CREATE_IN_PROGRESS	AWS::IAM::Role	CsvToParquetRole	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::IAM::Role	CsvToParquetRole	-
CREATE_COMPLETE	AWS::IAM::Role	CsvToParquetRole	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	CsvToParquetFunction	-
CREATE_COMPLETE	AWS::Lambda::Function	CsvToParquetFunction	-
CREATE_IN_PROGRESS	AWS::Lambda::Function	CsvToParquetFunction	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Permission	RawCsvToParquetInvokePermission	Resource creation Initiated
CREATE_IN_PROGRESS	AWS::Lambda::Permission	RawCsvToParquetInvokePermission	-
CREATE_COMPLETE	AWS::Lambda::Permission	RawCsvToParquetInvokePermission	-
CREATE_IN_PROGRESS	Custom::LambdaTrigger	RawCsvToParquetLambdaTrigger	-
CREATE_COMPLETE	Custom::LambdaTrigger	RawCsvToParquetLambdaTrigger	-
CREATE_IN_PROGRESS	Custom::LambdaTrigger	RawCsvToParquetLambdaTrigger	Resource creation Initiated
CREATE_COMPLETE	AWS::CloudFormation::Stack	phsp-dlg-datalake-dev	-

```
Successfully created/updated stack - phsp-dlg-datalake-dev in us-east-1
```

```
PS D:\Projetos\Repositórios\phsp-dlg-python-test\infra>
```



phsp-dlg-datalake-dev

Delete

Update

Stack actions

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Overview

Stack ID

[arn:aws:cloudformation:us-east-1:073206972289:stack/phsp-dlg-datalake-dev/d3638450-eb35-11ea-9ed9-0e8a1126433d](#)

Description

Data Lake Foundation

Status

CREATE_COMPLETE

Status reason

-

Root stack

-

Parent stack

-

Created time

2020-08-30 23:58:20 UTC-0300

Deleted time

-

Updated time

2020-08-30 23:58:26 UTC-0300

Drift status

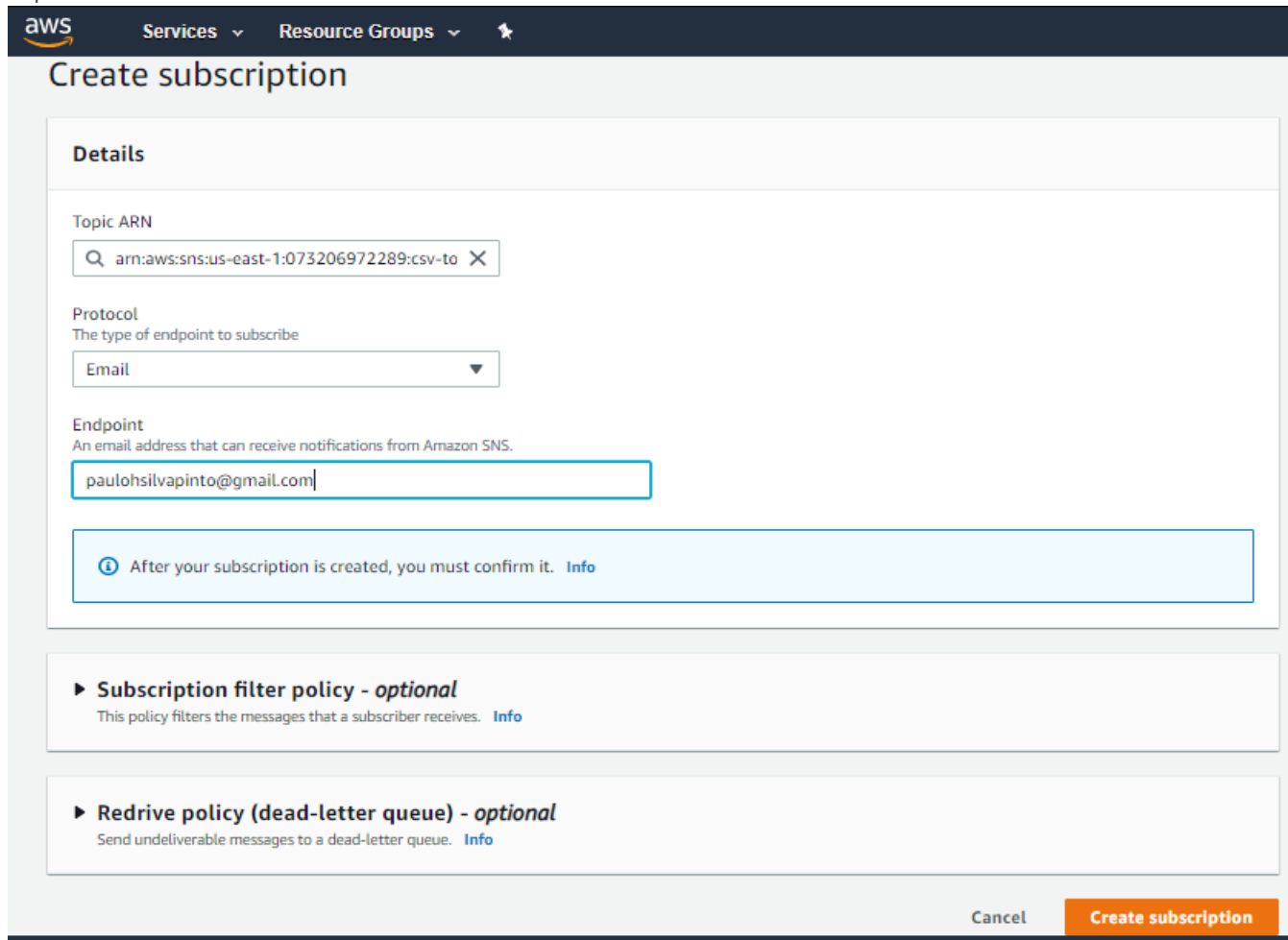
NOT_CHECKED

Last drift check time

-

2 – Subscribing to the SNS topic

Input

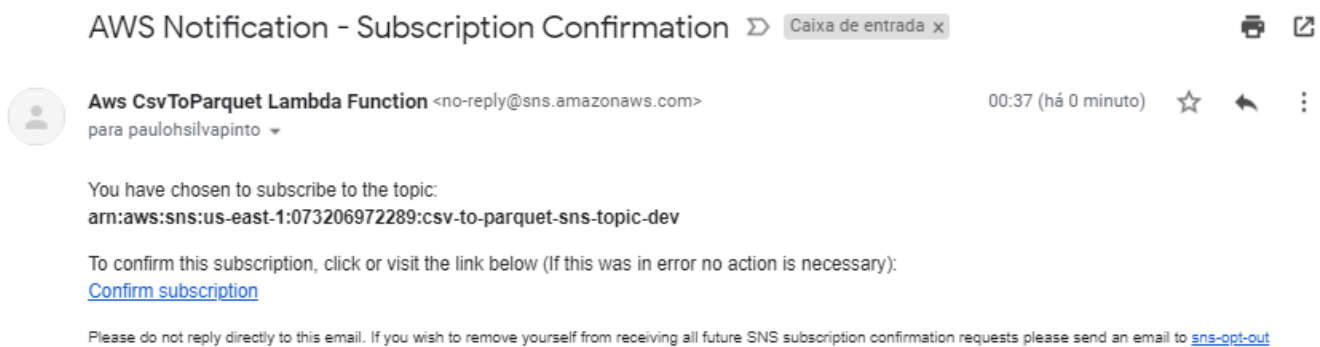


The screenshot shows the AWS console interface for creating a subscription. At the top, there's a navigation bar with the AWS logo, 'Services', and 'Resource Groups'. The main heading is 'Create subscription'. Below this, there's a 'Details' section with three input fields: 'Topic ARN' (containing 'arn:aws:sns:us-east-1:073206972289:csv-to'), 'Protocol' (set to 'Email'), and 'Endpoint' (containing 'paulohsilvapinto@gmail.com'). A blue information box states: 'After your subscription is created, you must confirm it. Info'. Below the details section, there are two optional policy sections: 'Subscription filter policy - optional' and 'Redrive policy (dead-letter queue) - optional'. At the bottom right, there are 'Cancel' and 'Create subscription' buttons.

Expected

Receiving the subscription confirmation e-mail.

Output



Subscription confirmed!

You have subscribed paulohsilvapinto@gmail.com to the topic:
csv-to-parquet-sns-topic-dev.

Your subscription's id is:

3 – Uploading csv files

Input

The following commands were executed:

- `cd ..\test-data\weather\`
- `.\upload_data_to_s3.ps1 dev`

Expected

- Files successfully uploaded to S3 raw bucket;
- Lambda Function triggered automatically by a S3 Event Notification.

Output

```
PS D:\Projetos\Repositórios\phsp-dlg-python-test\infra> cd ..\test-data\weather\  
PS D:\Projetos\Repositórios\phsp-dlg-python-test\test-data\weather> .\upload_data_to_s3.ps1 dev  
  
Loading weather data.  
Creating directories if does not exists.  
  
Directory: D:\Projetos\Repositórios\phsp-dlg-python-test\test-data\weather  
  
Mode                LastWriteTime         Length Name  
----                -  
d-----          8/30/2020  10:31 PM             incoming_data  
d-----          8/30/2020  10:31 PM             archived_data  
  
Detected development environment. Moving archived_data files to incoming_data.  
  
Metadata file was found.  
  
Loading file D:\Projetos\Repositórios\phsp-dlg-python-test\test-data\weather\incoming_data\weather.20160201.csv to phsp-dlg-datalake-raw-dev.  
upload: weather\incoming_data\weather.20160201.csv to s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160201.csv  
  
Loading file D:\Projetos\Repositórios\phsp-dlg-python-test\test-data\weather\incoming_data\weather.20160301.csv to phsp-dlg-datalake-raw-dev.  
upload: weather\incoming_data\weather.20160301.csv to s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160301.csv  
  
Archiving uploaded data.  
  
weather data was loaded successfully.  
  
PS D:\Projetos\Repositórios\phsp-dlg-python-test\test-data\weather> |
```

Files successfully uploaded to S3 raw bucket:

phsp-dlg-datalake-raw-dev

Overview

 Type a prefix and press Enter to search. Press ESC to clear.

Versions

US East (N. Virginia)



Viewing 1 to 2

<input type="checkbox"/> Name ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/> weather.20160201.csv	Aug 31, 2020 12:41:19 AM GMT-0300	10.0 MB	Standard
<input type="checkbox"/> weather.20160301.csv	Aug 31, 2020 12:41:42 AM GMT-0300	10.8 MB	Standard

Lambda Function triggered automatically by a S3 Event Notification:

▶	Timestamp	Message
		There are older events to load. Load more.
▶	2020-08-31T00:41:40.452-03:00	START RequestId: cfc7f215-c1d9-4561-9f31-84357792d9d3 Version: \$LATEST
▶	2020-08-31T00:41:40.452-03:00	OpenBLAS WARNING - could not determine the L2 cache size on this system, assuming 256k
▶	2020-08-31T00:41:41.887-03:00	[2020-08-31 03:41:41,887][INFO][root][parse_event] Event received: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-east-1',
▶	2020-08-31T00:41:41.887-03:00	[2020-08-31 03:41:41,887][INFO][root][handler] #--- Starting processing file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160201.csv.
▶	2020-08-31T00:42:02.932-03:00	START RequestId: 8f9797e8-edca-4fb3-94aa-287241350e87 Version: \$LATEST
▶	2020-08-31T00:42:02.936-03:00	[2020-08-31 03:42:02,936][INFO][root][parse_event] Event received: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-east-1',
▶	2020-08-31T00:42:02.936-03:00	[2020-08-31 03:42:02,936][INFO][root][handler] #--- Starting processing file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160301.csv.

4 – Lambda Execution

Input

The input was a S3 Event Notification sent automatically to the Lambda Function.

Expected

- Lambda Function finishes execution without errors;
- Files successfully saved to S3 analytics bucket, as parquet, compressed and partitioned by observation_date;
- Table created on AWS Glue Data Catalog with expected data types;
- SNS notification sent to E-mail.

Output

Lambda Function finishes execution without errors:

2020-08-31T00:41:40.452-03:00	START RequestId: cfc7f215-c1d9-4561-9f31-84357792d9d3 Version: \$LATEST
2020-08-31T00:41:40.452-03:00	OpenBLAS WARNING - could not determine the L2 cache size on this system, assuming 256k
2020-08-31T00:41:41.887-03:00	[2020-08-31 03:41:41,887][INFO][root][parse_event] Event received: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-east-1',
2020-08-31T00:41:41.887-03:00	[2020-08-31 03:41:41,887][INFO][root][handler] #--- Starting processing file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160201.csv.
2020-08-31T00:41:41.887-03:00	[2020-08-31 03:41:41,887][INFO][root][get_s3_object_metadata] Retrieving object metadata.
2020-08-31T00:41:41.945-03:00	[2020-08-31 03:41:41,945][INFO][botocore.credentials][load] Found credentials in environment variables.
2020-08-31T00:41:42.246-03:00	[2020-08-31 03:41:42,245][INFO][root][get_s3_object_metadata] Metadata: {'custom-cast': '{"ObservationDate":"date", "WindDirection":"int", "WindSpeed":"in
2020-08-31T00:41:42.248-03:00	[2020-08-31 03:41:42,247][INFO][root][get_partition_cols] Identifying partition columns.
2020-08-31T00:41:42.248-03:00	[2020-08-31 03:41:42,248][INFO][root][get_partition_cols] Partition Columns: ['observation_date']
2020-08-31T00:41:42.248-03:00	[2020-08-31 03:41:42,248][INFO][root][_read_csv_cloud] Extracting csv file from s3.
2020-08-31T00:41:42.300-03:00	[2020-08-31 03:41:42,300][INFO][botocore.credentials][load] Found credentials in environment variables.
2020-08-31T00:41:43.486-03:00	[2020-08-31 03:41:43,486][INFO][root][cast_df_columns] Casting dataframe columns.
2020-08-31T00:41:43.742-03:00	[2020-08-31 03:41:43,742][INFO][root][str_columns_to_upper] Applying upper to string columns.
2020-08-31T00:41:44.225-03:00	[2020-08-31 03:41:44,224][INFO][root][add_etl_metadata_to_df] Adding ETL metadata.
2020-08-31T00:41:44.248-03:00	[2020-08-31 03:41:44,239][INFO][root][normalize_column_name] Normalizing column names.
2020-08-31T00:41:44.248-03:00	[2020-08-31 03:41:44,240][INFO][root][replace_nan_values] Replacing NaN values to None.
2020-08-31T00:41:44.639-03:00	[2020-08-31 03:41:44,638][INFO][root][_save_to_s3_as_parquet] Saving dataframe to s3.
2020-08-31T00:41:51.727-03:00	[2020-08-31 03:41:51,727][INFO][root][_save_to_s3_as_parquet] Successfully saved dataframe to s3 on s3://phsp-dlg-datalake-analytics-dev/databases/analyti
2020-08-31T00:41:51.727-03:00	[2020-08-31 03:41:51,727][INFO][root][publish_success_to_sns] Publishing success to arn:aws:sns:us-east-1:073206972289:csv-to-parquet-sns-topic-dev
2020-08-31T00:41:51.826-03:00	[2020-08-31 03:41:51,825][INFO][root][handler] #--- Finished loading file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160201.csv. --
2020-08-31T00:41:51.840-03:00	END RequestId: cfc7f215-c1d9-4561-9f31-84357792d9d3

2020-08-31T00:42:02.932-03:00	START RequestId: 8f9797e8-edca-4fb3-94aa-287241350e87 Version: \$LATEST
2020-08-31T00:42:02.936-03:00	[2020-08-31 03:42:02,936][INFO][root][parse_event] Event received: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-east-1',
2020-08-31T00:42:02.936-03:00	[2020-08-31 03:42:02,936][INFO][root][handler] #--- Starting processing file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160301.csv.
2020-08-31T00:42:02.936-03:00	[2020-08-31 03:42:02,936][INFO][root][get_s3_object_metadata] Retrieving object metadata.
2020-08-31T00:42:03.031-03:00	[2020-08-31 03:42:03,031][INFO][root][get_s3_object_metadata] Metadata: {'custom-cast': '{"ObservationDate":"date", "WindDirection":"int", "WindSpeed":"in
2020-08-31T00:42:03.033-03:00	[2020-08-31 03:42:03,033][INFO][root][get_partition_cols] Identifying partition columns.
2020-08-31T00:42:03.034-03:00	[2020-08-31 03:42:03,034][INFO][root][get_partition_cols] Partition Columns: ['observation_date']
2020-08-31T00:42:03.034-03:00	[2020-08-31 03:42:03,034][INFO][root][_read_csv_cloud] Extracting csv file from s3.
2020-08-31T00:42:03.063-03:00	[2020-08-31 03:42:03,063][INFO][botocore.credentials][load] Found credentials in environment variables.
2020-08-31T00:42:04.241-03:00	[2020-08-31 03:42:04,240][INFO][root][cast_df_columns] Casting dataframe columns.
2020-08-31T00:42:04.463-03:00	[2020-08-31 03:42:04,463][INFO][root][str_columns_to_upper] Applying upper to string columns.
2020-08-31T00:42:04.984-03:00	[2020-08-31 03:42:04,983][INFO][root][add_etl_metadata_to_df] Adding ETL metadata.
2020-08-31T00:42:04.999-03:00	[2020-08-31 03:42:04,999][INFO][root][normalize_column_name] Normalizing column names.
2020-08-31T00:42:04.999-03:00	[2020-08-31 03:42:04,999][INFO][root][replace_nan_values] Replacing NaN values to None.
2020-08-31T00:42:05.321-03:00	[2020-08-31 03:42:05,320][INFO][root][_save_to_s3_as_parquet] Saving dataframe to s3.
2020-08-31T00:42:12.893-03:00	[2020-08-31 03:42:12,892][INFO][root][_save_to_s3_as_parquet] Successfully saved dataframe to s3 on s3://phsp-dlg-datalake-analytics-dev/databases/analyti
2020-08-31T00:42:12.893-03:00	[2020-08-31 03:42:12,893][INFO][root][publish_success_to_sns] Publishing success to arn:aws:sns:us-east-1:073206972289:csv-to-parquet-sns-topic-dev
2020-08-31T00:42:12.970-03:00	[2020-08-31 03:42:12,969][INFO][root][handler] #--- Finished loading file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160301.csv. --
2020-08-31T00:42:12.980-03:00	END RequestId: 8f9797e8-edca-4fb3-94aa-287241350e87
2020-08-31T00:42:12.980-03:00	REPORT RequestId: 8f9797e8-edca-4fb3-94aa-287241350e87 Duration: 10045.16 ms Billed Duration: 10100 ms Memory Size: 512 MB Max Memory Used: 279 MB

Files successfully saved to S3 analytics bucket, as parquet, compressed and partitioned by observation_date:

Services
Resource Groups
🌟
🔔
paulohsilvapinto @ paulohsilva...
Global

[Amazon S3](#)
>
[phsp-dlg-datalake-analytics-dev](#)
>
[databases](#)
>
[analytics_db](#)
>
[tbl_weather](#)
>
[observation_date=2016-02-01](#)

phsp-dlg-datalake-analytics-dev

Overview

Type a prefix and press Enter to search. Press ESC to clear.

Upload

Create folder

Download

Actions

Versions

Hide

Show

US East (N. Virginia)

Viewing 1 to 1

<input type="checkbox"/> Name	Last modified	Size	Storage class
<input type="checkbox"/> 6ce43a8e7c2c4b1483fc0f5fb4662136.snappy.parquet	Aug 31, 2020 12:41:46 AM GMT-0300	39.1 KB	Standard

Table created on AWS Glue Data Catalog with expected data types:

AWS Glue | Services | Resource Groups | **tbl_weather** | Last updated 31 Aug 2020 | Table | Vers

[Edit table](#) [Delete table](#) [View partitions](#) [Compare ve](#)

Name	tbl_weather
Description	
Database	analytics_db
Classification	parquet
Location	s3://phsp-dlg-datalake-analytics-dev/databases/analytics_db/tbl_weather/
Connection	
Deprecated	No
Last updated	Mon Aug 31 00:42:12 GMT-300 2020
Input format	org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat
Output format	org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat
Serde serialization lib	org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe
Serde parameters	serialization.format 1
Table properties	projection.observation_date.format yyyy-MM-dd compressionType snappy projection.enabled false typeOfData file

AWS Glue | Services | Resource Groups | **tbl_weather** | Last updated 31 Aug 2020 | Table | Vers

Column name	Data type	Partition key	Comment
1	forecast_site_code	bigint	
2	observation_time	bigint	
3	wind_direction	bigint	
4	wind_speed	bigint	
5	wind_gust	bigint	
6	visibility	bigint	
7	screen_temperature	double	
8	pressure	bigint	
9	significant_weather_code	bigint	
10	site_name	string	
11	latitude	double	
12	longitude	double	
13	region	string	
14	country	string	
15	dl_creation_date	date	
16	dl_source_file	string	
17	observation_date	date	Partition (0)

SNS notification sent to E-mail:

Aws CsvToParquet La. | SUCCESS - weather.20160301.csv - Csv to parquet succeeded. - The f...

Aws CsvToParquet La. | SUCCESS - weather.20160201.csv - Csv to parquet succeeded. - The file s3://ph...

SUCCESS - weather.20160301.csv - Csv to parquet succeeded. | Caixa de entrada x



Aws CsvToParquet Lambda Function <no-reply@sns.amazonaws.com>
para paulohsilvapinto

00:42 (há 19 minutos) | ☆ | ↶ | ⋮

The file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160301.csv was loaded successfully into S3 Analytics and is accessible via Athena on table tbl_weather.

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

5 – Data Validation

Input

Queries executed via Athena.

Expected

- 93,255 rows in 2016-02 and 101,442 rows in 2016-03;
- Strings in uppercase;
- Well formatted numbers;
- Well formatted dates.

Output

93,255 rows in 2016-02 and 101,442 rows in 2016-03:

New query 1

```
1 SELECT
2   DATE_TRUNC('MONTH', observation_date) AS month
3   , count(1) as qty_registers
4 FROM "analytics_db"."tbl_weather"
5 group by DATE_TRUNC('MONTH', observation_date)
```

Run query

Save as

Create ▾

(Run time: 1.15 seconds, Data scanned: 0 KB)

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Results

	month ▾	qty_registers ▾
1	2016-02-01	93255
2	2016-03-01	101442

Well formatted numbers:

```
1 SELECT
2   forecast_site_code, observation_time, wind_direction, wind_speed, wind_gust, visibility, screen_temperature, pressure, significant_weather_code, latitude, longitude
3 FROM "analytics_db"."tbl_weather" limit 10;
```

Run query

Save as

Create ▾

(Run time: 0.98 seconds, Data scanned: 191.52 KB)

Format query

Clear

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Results

	forecast_site_code ▾	observation_time ▾	wind_direction ▾	wind_speed ▾	wind_gust ▾	visibility ▾	screen_temperature ▾	pressure ▾	significant_weather_code ▾	latitude ▾	longitude ▾
1	3002	0	11	14		50000	4.8	992	7	80.749	-0.854
2	3005	0	10	11		14000	3.9	993	9	80.139	-1.183
3	3008	0	10	13		40000	5.7	994	-99	59.53	-1.03
4	3017	0	9	14		16000	4.6	994	7	58.954	-2.9
5	3023	0	11	19		17000	6.4	991	9	57.358	-7.397
6	3026	0	8	11		50000	5.2	993	8	58.214	-6.325
7	3031	0	4	6		29000	2.1	994	8	57.725	-4.896
8	3034	0	8	6		50000	4.9	993	8	57.859	-5.636
9	3037	0	8	17		1200	5.2	993	11	57.257	-5.809
10	3039	0	8	18	36		-0.2		-99	57.42	-5.69

Strings in uppercase:

New query 1New query 2+

```
1 SELECT
2   distinct site_name, region, country
3 FROM "analytics_db"."tbl_weather";
```

Run querySave asCreate (Run time: 1.29 seconds, Data scanned: 233.06 KB)

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Results

14	NORTH (3802)	SOUTH WEST ENGLAND	ENGLAND
15	NORTHOLT (3872)	LONDON & SOUTH EAST ENGLAND	ENGLAND
16	ODIHAM (3761)	LONDON & SOUTH EAST ENGLAND	ENGLAND
17	MANSTON (3797)	LONDON & SOUTH EAST ENGLAND	ENGLAND
18	THORNEY ISLAND (3872)	LONDON & SOUTH EAST ENGLAND	
19	SHOREHAM (3876)	LONDON & SOUTH EAST ENGLAND	ENGLAND
20	ALTNAHARRA SAWS (3044)	HIGHLAND & EILEAN SIAR	SCOTLAND
21	AVIEMORE (3063)	HIGHLAND & EILEAN SIAR	SCOTLAND
22	ORLOCK HEAD (99018)	NORTHERN IRELAND	
23	KESWICK (3212)	NORTH WEST ENGLAND	ENGLAND
24	WALNEY ISLAND (3214)	NORTH WEST ENGLAND	ENGLAND
25	CROSBY (3316)	NORTH WEST ENGLAND	ENGLAND
26	REDESDALE CAMP (3230)	NORTH EAST ENGLAND	

Well formatted dates:

```
1 SELECT
2   distinct observation_date
3 FROM "analytics_db"."tbl_weather" limit 10;
```

Run querySave asCreate (Run time: 1.04 seco

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Results

	observation_date
6	2016-02-11
1	2016-02-16
9	2016-02-19
5	2016-02-21
8	2016-02-22
7	2016-03-03

6 – Forcing an error

Input

CSV file uploaded with invalid value for the data type specified.

Specified Integer and received String.

```
ForecastSiteCode,ObservationTime,ObservationDate,WindDirection,WindSpeed,WindGust,Visibility,ScreenTemperature,Pressure,3002,0,2016-02-01T00:00:00,12,8,,abcde,2.10,997,8,BALTASOUND (3002),60.7490,-0.8540,Orkney & Shetland,SCOTLAND3005,0,2016-02-01T00:00:00,10,2,,35000,0.10,997,7,LERWICK (S. SCREEN) (3005),60.1390,-1.1830,Orkney & Shetland,SCOTLAND3008,0,2016-02-01T00:00:00,8,6,,50000,2.80,997,-99,FAIR ISLE (3008),59.5300,-1.6300,Orkney & Shetland,
```

Expected

- Lambda failure;
- SNS notification sent to E-mail.

Output

Lambda failure:

```
2020-08-31T01:37:58.227-03:00    START RequestId: 740b5d9a-c826-4cf2-8eac-baa5604fcd13 Version: $LATEST
2020-08-31T01:37:58.231-03:00    [2020-08-31 04:37:58,231][INFO][root][parse_event] Event received: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-east-1',
2020-08-31T01:37:58.231-03:00    [2020-08-31 04:37:58,231][INFO][root][handler] #--- Starting processing file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160201.csv.
2020-08-31T01:37:58.231-03:00    [2020-08-31 04:37:58,231][INFO][root][get_s3_object_metadata] Retrieving object metadata.
2020-08-31T01:37:58.300-03:00    [2020-08-31 04:37:58,300][INFO][root][get_s3_object_metadata] Metadata: {'custom-cast': '{"ObservationDate":"date", "WindDirection":"int", "WindSpeed":"in
2020-08-31T01:37:58.302-03:00    [2020-08-31 04:37:58,302][INFO][root][get_partition_cols] Identifying partition columns.
2020-08-31T01:37:58.302-03:00    [2020-08-31 04:37:58,302][INFO][root][get_partition_cols] Partition Columns: ['observation_date']
2020-08-31T01:37:58.302-03:00    [2020-08-31 04:37:58,302][INFO][root][_read_csv_cloud] Extracting csv file from s3.
2020-08-31T01:37:59.211-03:00    [2020-08-31 04:37:59,210][INFO][root][cast_df_columns] Casting dataframe columns.
2020-08-31T01:37:59.427-03:00    [2020-08-31 04:37:59,427][ERROR][root][cast_df_columns] Could not cast column Visibility to int.
2020-08-31T01:37:59.427-03:00    [2020-08-31 04:37:59,427][INFO][root][publish_error_to_sns] Publishing error to arn:aws:sns:us-east-1:073206972289:csv-to-parquet-sns-topic-dev
2020-08-31T01:37:59.537-03:00    object cannot be converted to an IntegerDtype: TypeError Traceback (most recent call last): File "/var/task/main.py", line 54, in handler s3_object_meta=s
object cannot be converted to an IntegerDtype: TypeError
Traceback (most recent call last):
  File "/var/task/main.py", line 54, in handler
    s3_object_meta=s3_object_meta
  File "/var/task/main.py", line 260, in cast_df_columns
    raise err
  File "/var/task/main.py", line 233, in cast_df_columns
    'int64')
  File "/opt/python/pandas/core/generic.py", line 5698, in astype
    new_data = self._data.astype(dtype=dtype, copy=copy, errors=errors)
  File "/opt/python/pandas/core/internals/managers.py", line 582, in astype
    return self.apply("astype", dtype=dtype, copy=copy, errors=errors)
  File "/opt/python/pandas/core/internals/managers.py", line 442, in apply
    applied = getattr(b, f)(**kwargs)
  File "/opt/python/pandas/core/internals/blocks.py", line 625, in astype
    values = astype_nansafe(vals1d, dtype, copy=True)
  File "/opt/python/pandas/core/dtypes/cast.py", line 821, in astype_nansafe
    return dtype.construct_array_type()._from_sequence(arr, dtype=dtype, copy=copy)
  File "/opt/python/pandas/core/arrays/integer.py", line 354, in _from_sequence
    return integer_array(scalars, dtype=dtype, copy=copy)
  File "/opt/python/pandas/core/arrays/integer.py", line 135, in integer_array
    values, mask = coerce_to_array(values, dtype=dtype, copy=copy)
  File "/opt/python/pandas/core/arrays/integer.py", line 218, in coerce_to_array
    raise TypeError(f"{values.dtype} cannot be converted to an IntegerDtype")
TypeError: object cannot be converted to an IntegerDtype
```

Copy

SNS notification sent to E-mail:

ERROR - weather.20160201.csv - Csv to parquet failed. Caixa de entrada x



Aws CsvToParquet Lambda Function <no-reply@sns.amazonaws.com>
para paulo@silvapinto

01:35 (há 6 minutos) ☆ ↩ ⋮

The file s3://phsp-dlg-datalake-raw-dev/csv_to_analytics/weather/weather.20160201.csv could not be loaded into S3 Analytics. Please check the logs for more details.

Error:
object cannot be converted to an IntegerDtype

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe: