

<b>Overview</b>	<b>2</b>
<b>Data sources</b>	<b>3</b>
Amazon Customer Reviews Dataset	3
Country	3
<b>Prerequisites and Environment Setup</b>	<b>3</b>
Apache Airflow	3
Postgres connection to AWS Redshift Database (dwh)	4
SSH connection AWS EMR Cluster (emr_ssh_connection)	4
Copy DAG files	5
AWS EMR Cluster	5
Setting up EMR Cluster	6
Copy source files	6
Amazon Redshift Database	6
AWS S3	7
<b>Architecture</b>	<b>8</b>
Storage	8
Landing Zone	8
Working Zone	8
Data Warehouse	9
Stage	9
Detail Data Store	10
Data Marts	11
Data Quality	11
ETL	11
Load Flowchart	12
<b>Update strategy</b>	<b>12</b>
<b>How to run</b>	<b>13</b>
Project configuration	13
airflow/review_project.cfg	13
etl/review_project.cfg	13
Configure initial dataset	14
Initial Data Loading	15
ETL Flow	15
Check data quality	15
<b>Project Files</b>	<b>15</b>
Airflow DAG	15

copy_initial_data_dag	15
copy_review_from_aws_S3	16
create_dwh_schema_dag	16
DWH_processing_dag	16
fill_dimensions_dag	16
<b>Testing the limits</b>	<b>16</b>
Initial data loading	16
Uploading ~700Mb of data	17
Scenarios	17
If the data was increased by 100x.	17
If the pipelines were run on a daily basis by 7am.	17
If the database needed to be accessed by 100+ people.	17
<b>Amazon Customer Reviews Dataset product categories</b>	<b>18</b>
<b>Analysis examples</b>	<b>20</b>

# Overview

Amazon sells tens and hundreds thousands of goods every day. After selling a customer can leave a review about the item purchased. All these reviews help other people to make the right choice. Also they can be used as an excellent source of data for marketing experts and other specialists from Amazon. Everyday customers generate huge amounts of data. So their analysis can be a bit complicated. Usually before the analysis data are cleared and transformed to the proper form. In many cases it means performing a lot of different operations and transformations.

The goal of this capstone project is building an analytical Data Warehouse. Customer reviews have been chosen as a subject of analysis.

The Data Warehouse can answer to the next questions:

- How do ratings vary with different options, for example verified purchases, marketplaces or product categories
- How have number of ratings changed over time
- Are ratings helpful

Also it is possible to take a look at the reviewer behavior or provide sentiment analysis.

## Data sources

### Amazon Customer Reviews Dataset

In a period of over two decades since the first review in 1995, millions of Amazon customers have contributed over a hundred million reviews to express opinions and describe their experiences regarding products on the Amazon.com website. Over 130+ million customer reviews are available to researchers as part of this dataset.

More: <https://registry.opendata.aws/amazon-reviews/> and full documentation: <https://s3.amazonaws.com/amazon-reviews-pds/readme.html>

### Country

Country dimension is provided in the JSON file:

s3://brutway-capstone-project/country/country-and-continent-codes-list.json

## Prerequisites and Environment Setup

The project has been developed using Apache Airflow, AWS S3, Amazon Redshift Database and AWS EMR.

Amazon Customer Reviews Dataset resides in the AWS S3 bucket in us-east-1 region.

Copying between S3 buckets can be possible if all the buckets are in the same region. So

Storage Area must be organized in us-east-1 region too. For performance improvement I resided the EMR cluster and Redshift Database in us-east-1 region too.

## Apache Airflow

Apache Airflow - is a platform to programmatically author, schedule and monitor workflows. One of the well known installations of Airflow is [puckel/docker-airflow](#). I have used this docker image for my workflows orchestration. But there are other possibilities for Apache Airflow deployment. For example AWS Cloud Formation or your own installation.

The next preliminary steps must be performed before Airflow using

### Postgres connection to AWS Redshift Database (dwh)

Conn Id *	dwh
Conn Type	Postgres
Host	review-dwh-cluster.c us-east-1.redshift.amazonaws.com
Schema	dwh
Login	awsuser
Password	
Port	5439

### SSH connection AWS EMR Cluster (emr\_ssh\_connection)

Before setting up the connection to the EMR Cluster the key pair from AWS EMR should be generated and copied to the Airflow installation.

Conn Id *	emr_ssh_connection
Conn Type	SSH
Host	ec2-54-245-47-58.us-east-1.compute.amazonaws.com
Username	hadoop
Password	
Port	22
Extra	<pre>{   "key_file": "/usr/local/airflow/.ssh/my_key_pair.pem",   "timeout": "10",   "compress": "false",   "no_host_key_check": "true",   "allow_host_key_change": "false" }</pre>

## Copy DAG files

All the files from `/de_capstone_project/airflow/dags` must be copied to the `/dags` folder inside the Airflow installation. All the files from `/de_capstone_project/airflow/plugins` must be copied to the `/plugins` folder inside the Airflow installation.

## AWS EMR Cluster

I used `m5.xlarge` instance type with 3 instances (1 master and 2 core nodes)

Release: 5.29.0

Applications: Spark (Spark 2.4.4)

## Software configuration

**Release** emr-5.29.0 ⓘ

**Applications**

- ☐ Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.6, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- ☐ HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.6, Hue 4.4.0, Phoenix 4.14.3, and ZooKeeper 3.4.14
- ☐ Presto: Presto 0.227 with Hadoop 2.8.5 HDFS and Hive 2.3.6 Metastore
- ☒ Spark: Spark 2.4.4 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.2

☐ Use AWS Glue Data Catalog for table metadata ⓘ

## Hardware configuration

**Instance type** m5.xlarge ⓘ The selected instance type adds 64 GiB of GP2 EBS storage per instance by default. [Learn more](#)

**Number of instances** 3 (1 master and 2 core nodes)

## Setting up EMR Cluster

For connecting to a Redshift Database all the scripts use package psycopg2. And for connecting to S3 buckets use boto3.

To perform all the preliminary settings you should connect to the EMR Cluster and execute:

```
$ sudo yum install postgresql-libs
$ sudo yum install postgresql-devel
$ sudo pip-3.6 install psycopg2
$ pip-3.6 install boto3 --user
$ export PYSPARK_DRIVER_PYTHON=python3
$ export PYSPARK_PYTHON=python3
```

## Copy source files

All the files from /de\_capstone\_project/etl must be copied to the /home/hadoop/review\_project\_src folder inside the EMR Cluster.

## Amazon Redshift Database


I used 2 node cluster with Instance Types dc2.large

DC2

High performance with fixed local SSD storage

<input checked="" type="radio"/>	dc2.large	\$0.25/node/hour
	Storage: 160 GB/node	
<input type="radio"/>	dc2.8xlarge	\$4.80/node/hour
	Storage: 2.6 TB/node	

SSD



dc2.large

2 vCPU (gen 2)

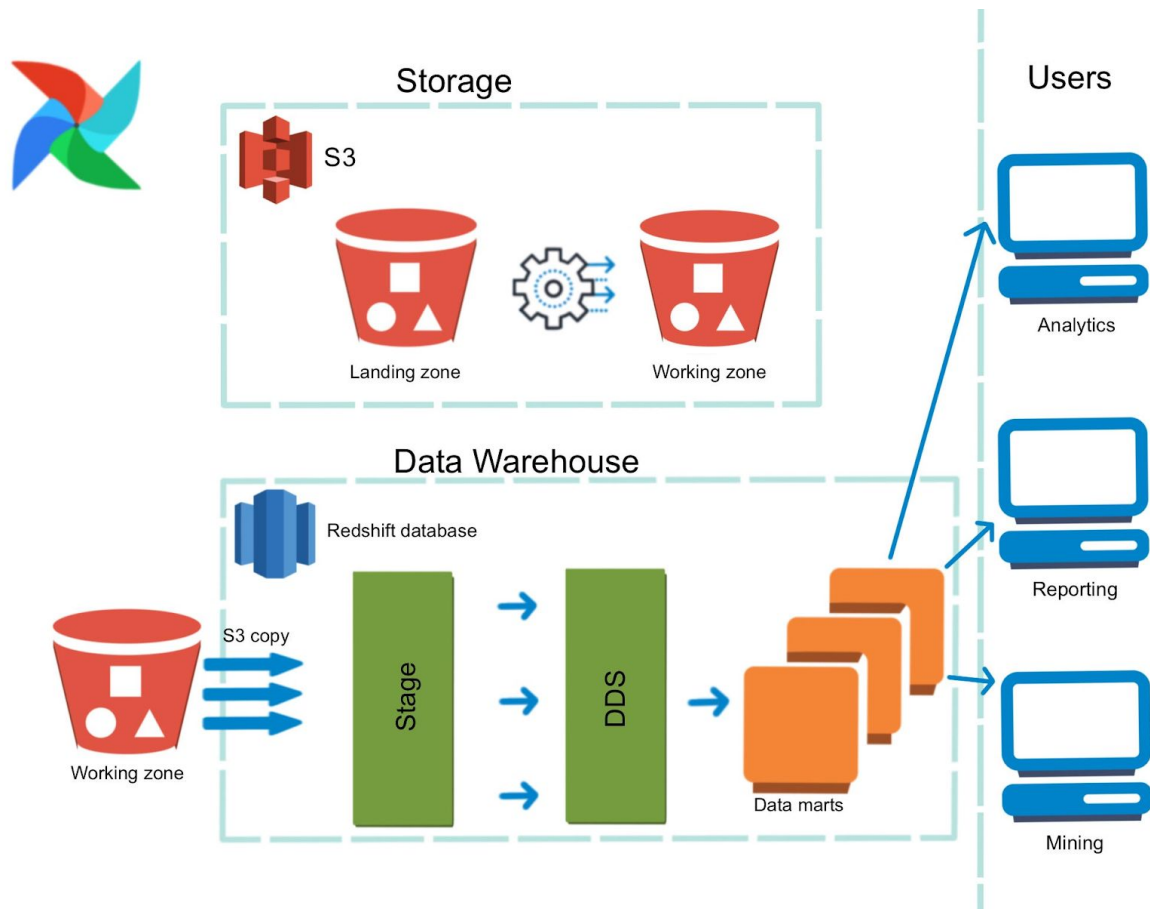
## AWS S3

Storage Area for the project resides in two AWS S3 buckets: landing zone and working zone

<input type="radio"/>	<a href="#">review-project-landing-zone</a>	US East (N. Virginia) us-east-1
<input type="radio"/>	<a href="#">review-project-working-zone</a>	US East (N. Virginia) us-east-1

Buckets names must be specified in the file `review_project.cfg`

# Architecture



## Storage

Storage Area has been carried out in two S3 buckets, one for source data and other for preprocessed data.

## Landing Zone

The Landing Zone is the area for storing files getting from the data sources. All the files are stored as is.

Landing Zone is cleared after processing and storing data at the Working Zone

## Working Zone

Working Zone is used for storing preprocessed files from the Landing Zone. A Spark Job is used for transforming source data and putting them to the Working Zone. One review file contains information about products, customers and marketplaces. So one review file is processed to several entities (fact and dimensions)

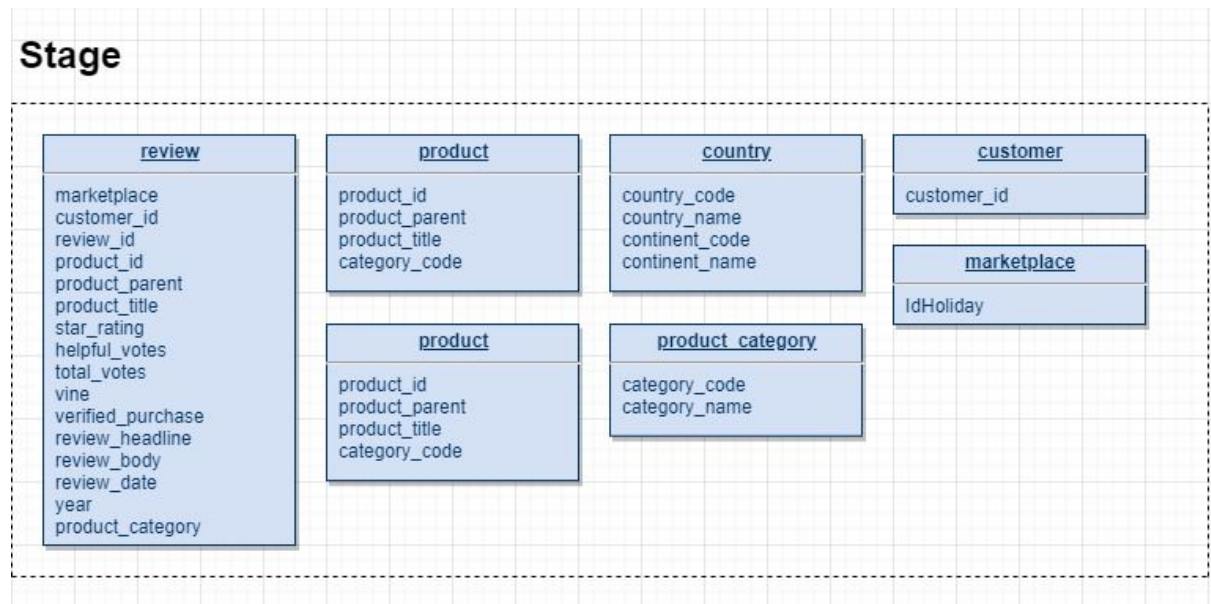


# Data Warehouse

Data Warehouse has been carried out using Amazon Redshift Database.

There are three main and one supplementary area in the Data Warehouse. They are made in different schemas

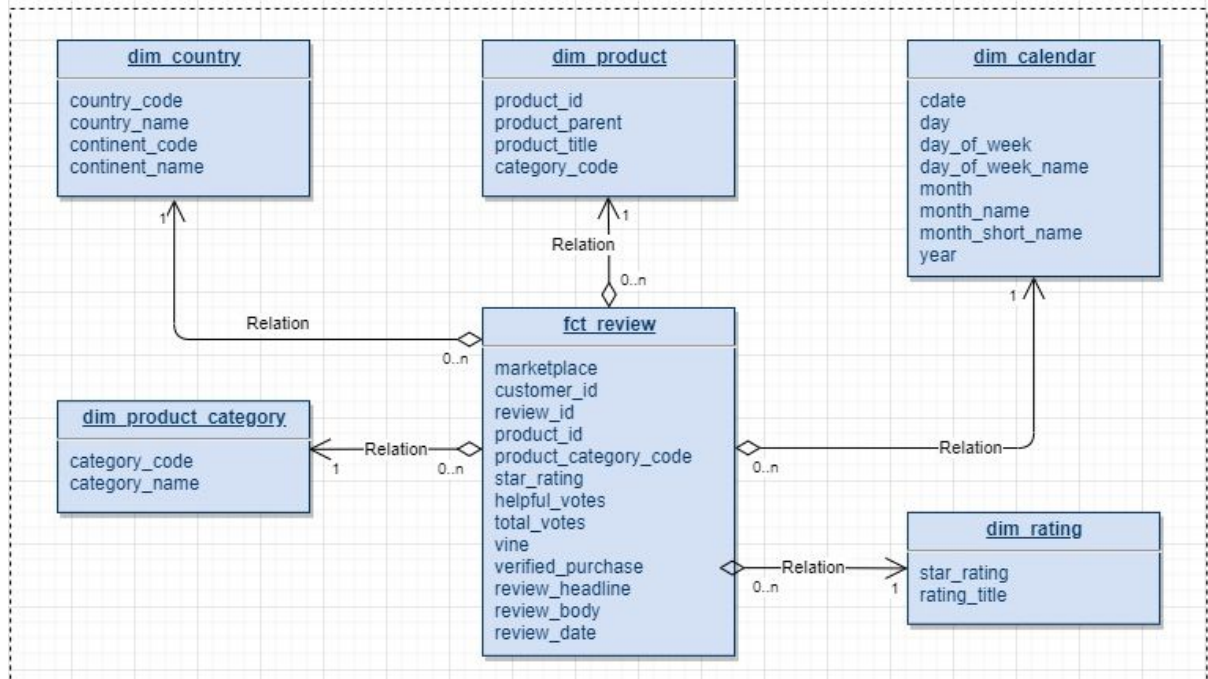
## Stage



All the data from the Working Zone are copied using copy operation from files to the Redshift tables as is. Before the copying all the tables in the Stage schema are truncated.

## Detail Data Store

### Detail Data Store (DDS)



Detail Data Store comprises a fact table and several dimensions tables. I used star schema for data modeling. DDS uses UPSERT strategy for data updating. Amazon Redshift does not support UPSERT operation so I used DELETE + INSERT instead.

## Data Marts

### Data Marts (DM)

<u>product_rating_day</u>
cdate
product_id
marketplace
star_rating_cnt_1
star_rating_cnt_2
star_rating_cnt_3
star_rating_cnt_4
star_rating_cnt_5
star_helpful_votes_cnt_1
star_helpful_votes_cnt_2
star_helpful_votes_cnt_3
star_helpful_votes_cnt_4
star_helpful_votes_cnt_5
rating_cnt
helpful_votes_cnt
verified_purchase_cnt
is_vine_member_cnt

Data Marts is a source of data for a team of analytics, marketing specialists and others. It contains daily aggregated information about products reviews.

## Data Quality

And the last area is Data Quality. ETL Pipeline performs different data quality checks and stores the result into the etl.check\_dq table

### Check DQ (ETL)

<u>check_dq</u>
id
cdate
check_type
check_name
query
param
row_number

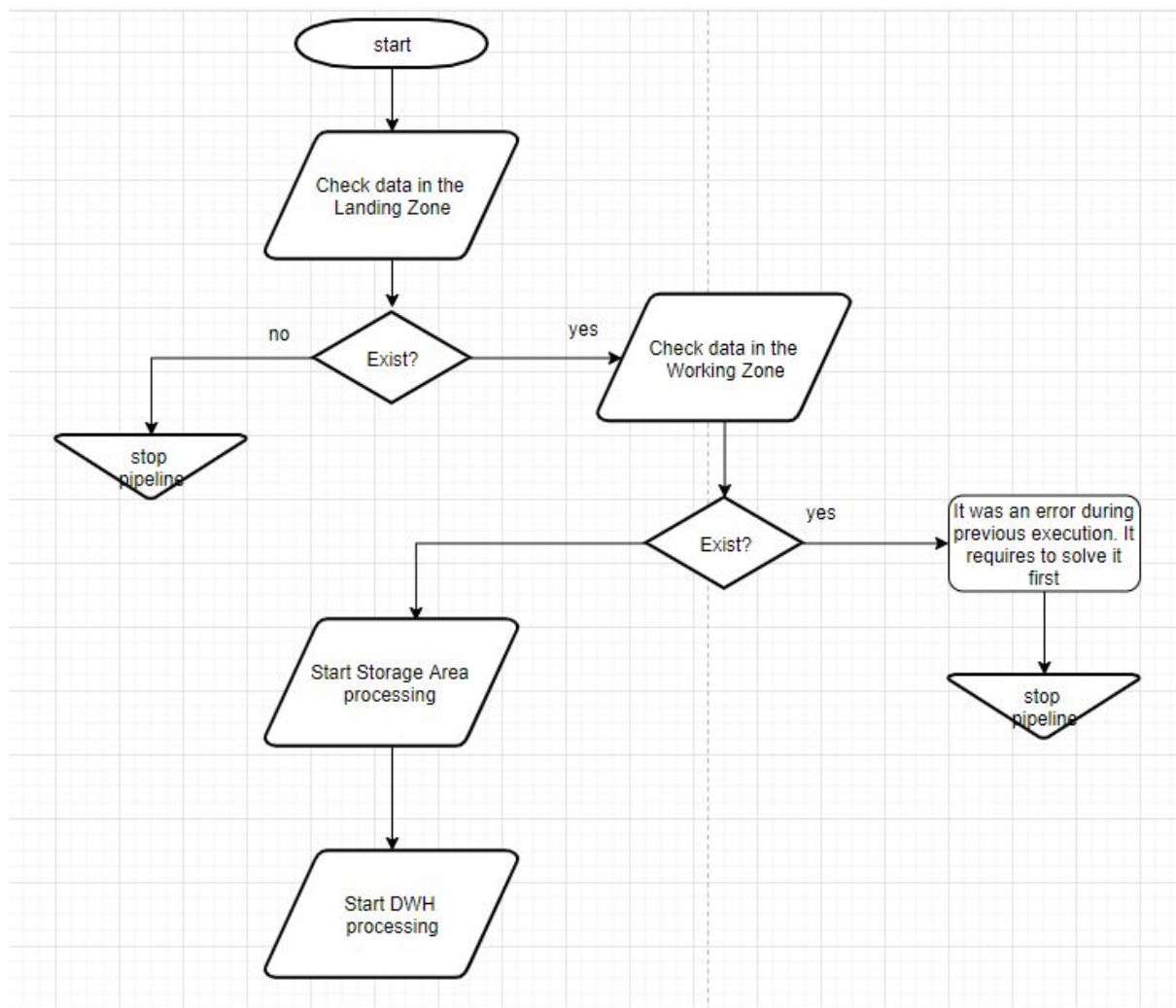
## ETL

ETL pipeline comprises of several parts:

- Initiate Data Warehouse
  - Create DWH structure (schemas and tables)

- Initial Data Loading. Copy an initial subset of data
- Fill in fixed dimension: dim\_calendar and dim\_rating tables
- DWH Control Flow:
  - Transform data from the Landing Zone to the Working one
  - Copy data from Working Zone to Stage in the DWH
  - Transform data from Stage to DDS
  - Calculate Data Marts
  - Check data quality

## Load Flowchart



## Update strategy

This analytical DataWarehouse suppose batch updating. It isn't necessary to implement Kappa Architecture. So ETL pipeline is started everyday at 07:00

# How to run

## Project configuration

Projects configuration is provided in two slightly different files with the same name.

### airflow/review\_project.cfg

# Please provide your own KEY and SECRET KEY

[AWS]

KEY =

SECRET =

REGION = us-east-1

# This section is unchangeable

[S3\_DATA\_SOURCE]

BUCKET\_REVIEW = amazon-reviews-pds

BUCKET\_COUNTRY = brutway-capstone-project

# Please create two S3 buckets and write down their names into this section

[S3\_STORAGE]

LANDING\_ZONE = review-project-landing-zone

WORKING\_ZONE = review-project-working-zone

# An amount of initial dataset (detailed explanation will be provided below)

# possible options

[INITIAL\_LOADING]

REVIEW\_INITIAL\_DATA = moderate

### etl/review\_project.cfg

# Please provide your own KEY and SECRET KEY

[AWS]

KEY = AKIAS53S2P6WJPOQWAF7

SECRET = rQxKF1htnwH67q0ui+5NUzynQhyRxCsYHImMmTJT

REGION = us-east-1

# Please create two S3 buckets and write down their names into this section

[S3\_STORAGE]

LANDING\_ZONE = review-project-landing-zone

WORKING\_ZONE = review-project-working-zone

# Connection parameters to the Redshift cluster

[CLUSTER]

```
HOST = 'review-dwh-cluster.____u.us-east-1.redshift.amazonaws.com'
DB_NAME = 'dwh'
DB_USER = 'awsuser'
DB_PASSWORD = 'pwd_'
DB_PORT = 5439
```

```
[IAM_ROLE]
ROLE_ARN =
```

## Configure initial dataset

It is possible to set up an amount of initial Reviews. Amazon Customer Reviews Dataset consists of 43 product categories that take more than 43 Gb of space. So it will be quite expensive to load all of them. So I provided special settings for controlling the amount of data. The dictionary of possible options is provided in the file `/de_capstone_project/airflow/plugins/helpers/review_project_initial_data.py`

---

```
product_category_filter_a_category = ["Luggage"]
product_category_filter_small      = ["Digital_Video_Games", "Gift_Card",
"Digital_Software"]
product_category_filter_moderate   = ["Digital_Video_Games", "Gift_Card",
"Digital_Software", "Mobile_Electronics", "Major_Appliances",
"Personal_Care_Appliances", "Software", "Home_Entertainment"]
product_category_filter_big        = []
product_category_all               = []

product_category_filter = {"category": product_category_filter_a_category,
                           "small": product_category_filter_small,
                           "moderate":
product_category_filter_moderate,
                           "big": product_category_filter_big,
                           "all": product_category_all
                           }
```

---

So you can choose among the next options: category, small, moderate, big or all.

And this option should be setting up in the file

`/de_capstone_project/airflow/dags/review_project.cfg` in the section

```
[INITIAL_LOADING]
```

```
REVIEW_INITIAL_DATA = moderate
```

The full list of product categories is described in section [Amazon Customer Reviews Dataset product categories](#)

## Initial Data Loading

Before starting data loading the Data Warehouse model should be created. So please run the following DAGs as follows:

1. create\_dwh\_schema\_dag
2. copy\_initial\_data\_dag
3. fill\_dimensions\_dag

And then run DWH\_processing\_dag

## ETL Flow

The DAG “fill\_dimensions\_dag” starts all the necessary steps for data processing. It is scheduled to run everyday at 07:00.

## Check data quality

All data quality checks are provided in etl/helpers/check\_data\_quality\_config.py. It is possible to set up any checks anytime.

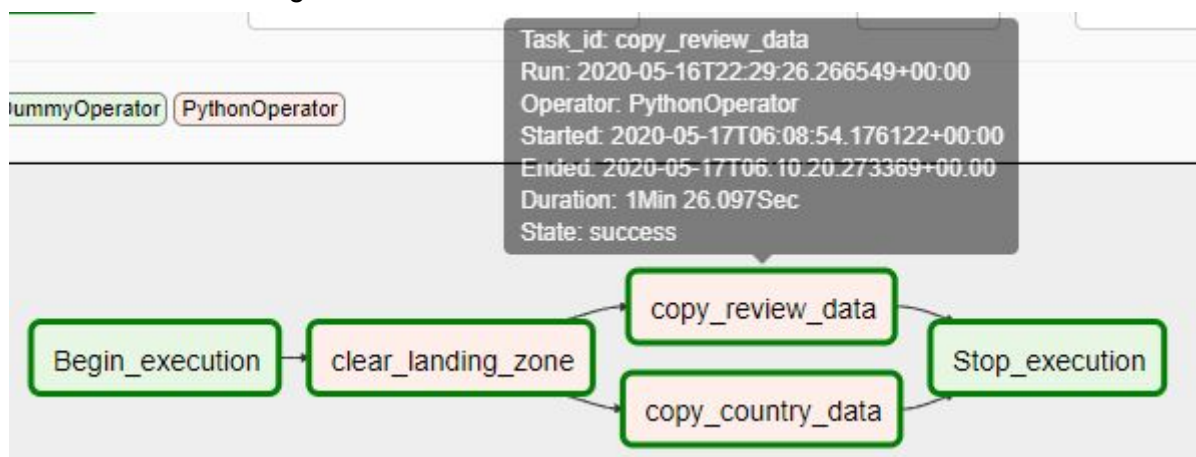
## Project Files

### Airflow DAG

#### copy\_initial\_data\_dag

File: copy\_initial\_data\_dag.py

Description: This DAG copy Initial Customer Reviews dataset and Country files from the sources to the Working Zone



## copy\_review\_from\_aws\_S3

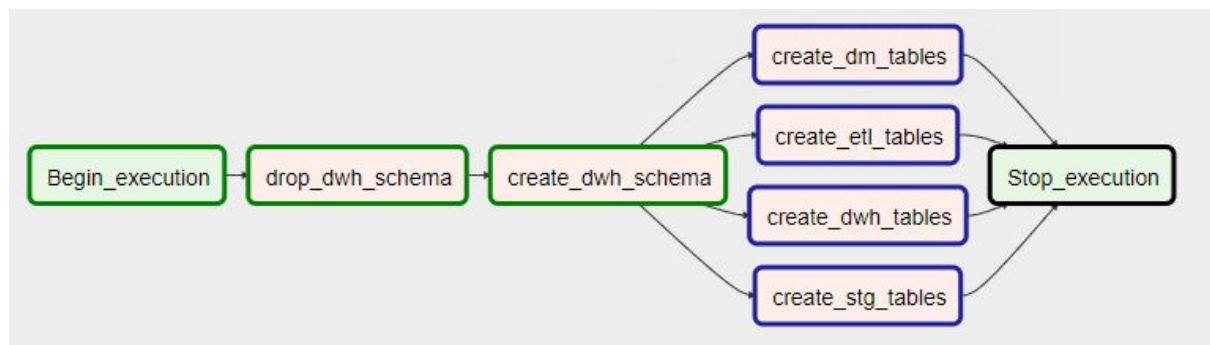
File: copy\_review\_from\_aws\_dag.py

Description: This DAG copy only Customer Reviews dataset from the S3 amazon-reviews-pds to the Working Zone. The type of Dataset should be setted in the review\_project.cfg in the section [INITIAL\_LOADING]

## create\_dwh\_schema\_dag

File: create\_dwh\_schema\_dag.py

Description: Create DataWarehouse model



## DWH\_processing\_dag

File: dwh\_control\_flow\_dag.py

Description: Run ETL pipeline



## fill\_dimensions\_dag

File: fill\_fixed\_dimensions\_dag.py

Description: Fill in two fixed dimensions: Calendar and Rating



## Testing the limits

### Initial data loading

I chose several Product\_Categories (moderate set of data) that take about 580Mb of data.



Copying from sources to Landing Zone: 1m 30s  
Processing data from Working Zone to Landing: 4m 17s

```
select cdate, check_name, query, "row_number" from etl.check_dq order by  
cdate desc, query asc limit 30
```

1	May 17, 2020, 6:34:12 AM	Check records count	select count(*) from dds.dim_country;	255
2	May 17, 2020, 6:20:56 AM	Check records count	select count(*) from dds.fct_review;	1771182
3	May 17, 2020, 6:20:56 AM	Check records count	select count(*) from dm.product_rating_day;	1281982
4	May 17, 2020, 6:20:55 AM	Check records count	select count(*) from dds.dim_product_category;	8
5	May 17, 2020, 6:20:55 AM	Check records count	select count(*) from dds.dim_rating;	5
6	May 17, 2020, 6:20:54 AM	Check records count	select count(*) from dds.dim_product;	282730
7	May 17, 2020, 6:20:53 AM	Check records count	select count(*) from dds.dim_country;	255
8	May 17, 2020, 6:20:53 AM	Check records count	select count(*) from dds.dim_customer;	1771182

## Uploading ~700Mb of data

Copying from sources to Landing Zone: 1m 30s  
Processing data from Working Zone to Landing: 5m 15s

1	May 17, 2020, 6:34:16 AM	Check records count	select count(*) from dm.product_rating_day;	3396701
2	May 17, 2020, 6:34:15 AM	Check records count	select count(*) from dds.dim_rating;	5
3	May 17, 2020, 6:34:15 AM	Check records count	select count(*) from dds.fct_review;	4107362
4	May 17, 2020, 6:34:14 AM	Check records count	select count(*) from dds.dim_product_category;	11
5	May 17, 2020, 6:34:13 AM	Check records count	select count(*) from dds.dim_customer;	3811269
6	May 17, 2020, 6:34:13 AM	Check records count	select count(*) from dds.dim_product;	988720
7	May 17, 2020, 6:34:12 AM	Check records count	select count(*) from dds.dim_country;	255

## Scenarios

If the data was increased by 100x.

Most of the workload is data processing at the Storage area while data is transformed from Landing Zone to Working Zone. I think the next steps can be tested:

- review repartition strategy (Spark RDD)
- Increase EMR cluster size

Also I would suggest to review data distribution in the Redshift Database.

If the pipelines were run on a daily basis by 7am.

DAG is scheduled to run at 07:00 everyday. So it is ok for batch processing

If the database needed to be accessed by 100+ people.

It is possible to reduce the concurrency limit for Redshift Database. But it is also possible to use BI tools that allow to extract data and store them into the internal storage for further analysis.

# Amazon Customer Reviews Dataset product categories

The table below describes Product Categories which are stored in Reviews Dataset. Please keep in mind the Size of a Dataset while project configuring

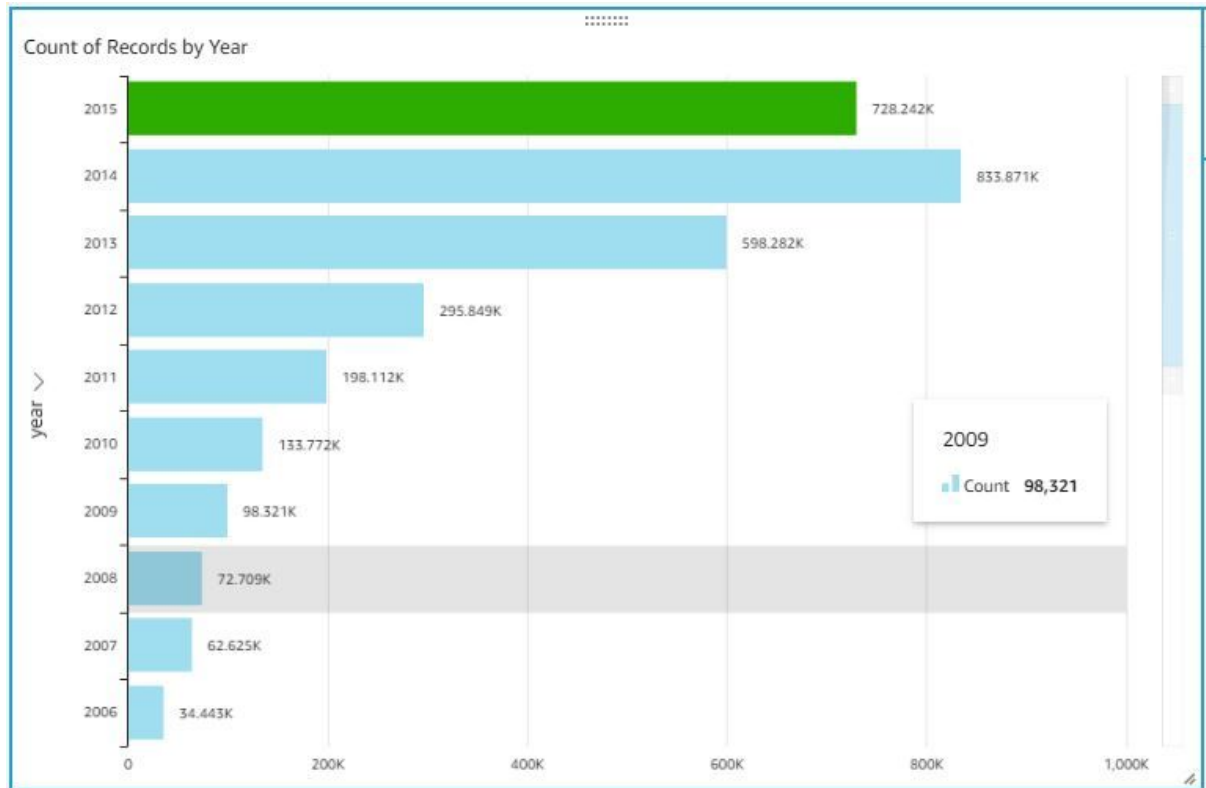
Product Category	Unit	Size
Apparel	MiB	1151.8
Automotive	MiB	809.8
Baby	MiB	491.1
Beauty	MiB	1271.2
Books	GiB	10
Camera	MiB	621.8
Digital_Ebook_Purchase	MiB	6005.9
Digital_Music_Purchase	MiB	389.1
Digital_Software	MiB	27.2
Digital_Video_Download	MiB	783.4
Digital_Video_Games	MiB	38.3
Electronics	MiB	958.7
Furniture	MiB	199.4
Gift_Card	MiB	16.8
Grocery	MiB	552.3
Health_&_Personal_Care	MiB	1393.3
Home	MiB	1508.6
Home_Entertainment	MiB	265
Home_Improvement	MiB	689.9
Jewelry	MiB	340.1

Kitchen	MiB	1291.4
Lawn_and_Garden	MiB	665.9
Luggage	MiB	78.4
Major_Appliances	MiB	33
Mobile_Apps	MiB	901
Mobile_Electronics	MiB	30.3
Music	MiB	2790.2
Musical_Instruments	MiB	264.9
Office_Products	MiB	704.2
Outdoors	MiB	618.7
PC	MiB	2115.2
Personal_Care_Appliances	MiB	24.3
Pet_Products	MiB	697
Shoes	MiB	895.5
Software	MiB	134.2
Sports	MiB	1224.1
Tools	MiB	453.7
Toys	MiB	1209.5
Video	MiB	215.8
Video_DVD	MiB	2972.8
Video_Games	MiB	661.4
Watches	MiB	226.1
Wireless	MiB	2362.1
(Empty)	(empty)	
Summary		38093.4

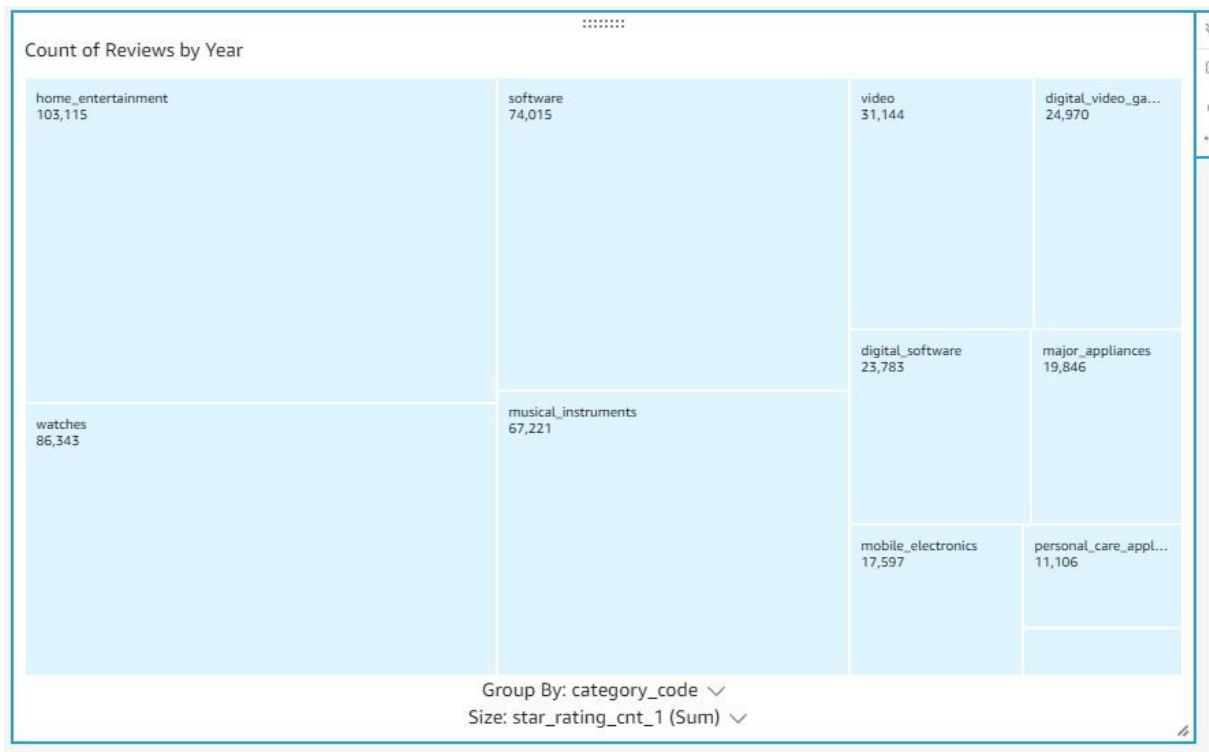
# Analysis examples

I used Amazon QuickSight for providing some examples

Count of reviews per Year



Number of 1-star rating per Product\_Category



## Data Model Sample

