Aman Yadav

Email: amanyadav18072000@gmail.com

**HIGH LEVEL DESIGN**

**SQL-SQOOP-DATA-INGESTION-ETL-PROJECT**

**Content**

[Abstract. 4](file:///C:\Users\amany\Desktop\Project%20Submission%20Steps\UGV_HLD_2.0v.docx#_TOC_250020)

1. [Introduction 5](file:///C:\Users\amany\Desktop\Project%20Submission%20Steps\UGV_HLD_2.0v.docx#_TOC_250019)
   1. Why this High-Level Design Document
   2. Scope
   3. Definition
2. [General Description. 6](file:///C:\Users\amany\Desktop\Project%20Submission%20Steps\UGV_HLD_2.0v.docx#_TOC_250015)
   1. Problem Statement
   2. Proposed Solution
   3. Data Requirement
   4. AWS RDS
   5. AWS EMR
   6. AWS EC2(Apache Airflow Installed)
3. [Design Details 10](file:///C:\Users\amany\Desktop\Project%20Submission%20Steps\UGV_HLD_2.0v.docx#_TOC_250005)
   1. [Process Flow. 10](file:///C:\Users\amany\Desktop\Project%20Submission%20Steps\UGV_HLD_2.0v.docx#_TOC_250004)
4. Conclusion 14

In Big data field when there is need to transfer the data from transaction databases to Hadoop ecosystem the best practice is by doing with Sqoop which can transfer the data from MySQL, PostgreSQL, Oracle, etc to Hadoop cluster and vice-versa.

**Abstract**

1. **Introduction**

###### Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* + - Present all of the design aspects and define them in detail
    - Describe the user interface being implemented
    - Describe the hardware and software interfaces
    - Describe the performance requirements
    - Include design features and the architecture of the project
    - List and describe the non-functional attributes like: o Security
      * Reliability
      * Maintainability
      * Portability
      * Reusability
      * Application compatibility
      * Resource utilization
      * Serviceability

##### Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

* 1. **Definitions**

*Terms**Description*

|  |  |
| --- | --- |
| **ETL** | **Extract Transformation Load** |
| **Database** | **Collection of Data** |
| **AWS** | **Amazon Web Service** |
| **Sqoop** | **SQL and Hadoop** |

### **General Description**

* 1. **Problem statement**

Use Sqoop to read data from SQL database and import it into Hadoop.

You need to build the following requirement:

1. Create SQL database at any cloud platform.
2. Design an Ecommerce database and store 10 GB record in SQL Database.
3. Use Sqoop to load data from SQL Database to Hadoop.
4. Schedule pipeline such a way that new data from Database can be transferred to Hadoop automatically on daily basis.
   1. **PROPOSED SOLUTION**

The solution proposed here is first set up the MySQL database on AWS RDS and connect with MySQL Workbench and create the e-commerce schema and dump the data or records after that set up the AWS EMR cluster having tools like Hadoop, Hive, Spark and Sqoop and then create another EC2 instance for Apache Airflow which schedule the entire code on daily basis and new updated or incremental data automatically get stored in Hadoop which will then use by downstream analyst or data scientist for model building.

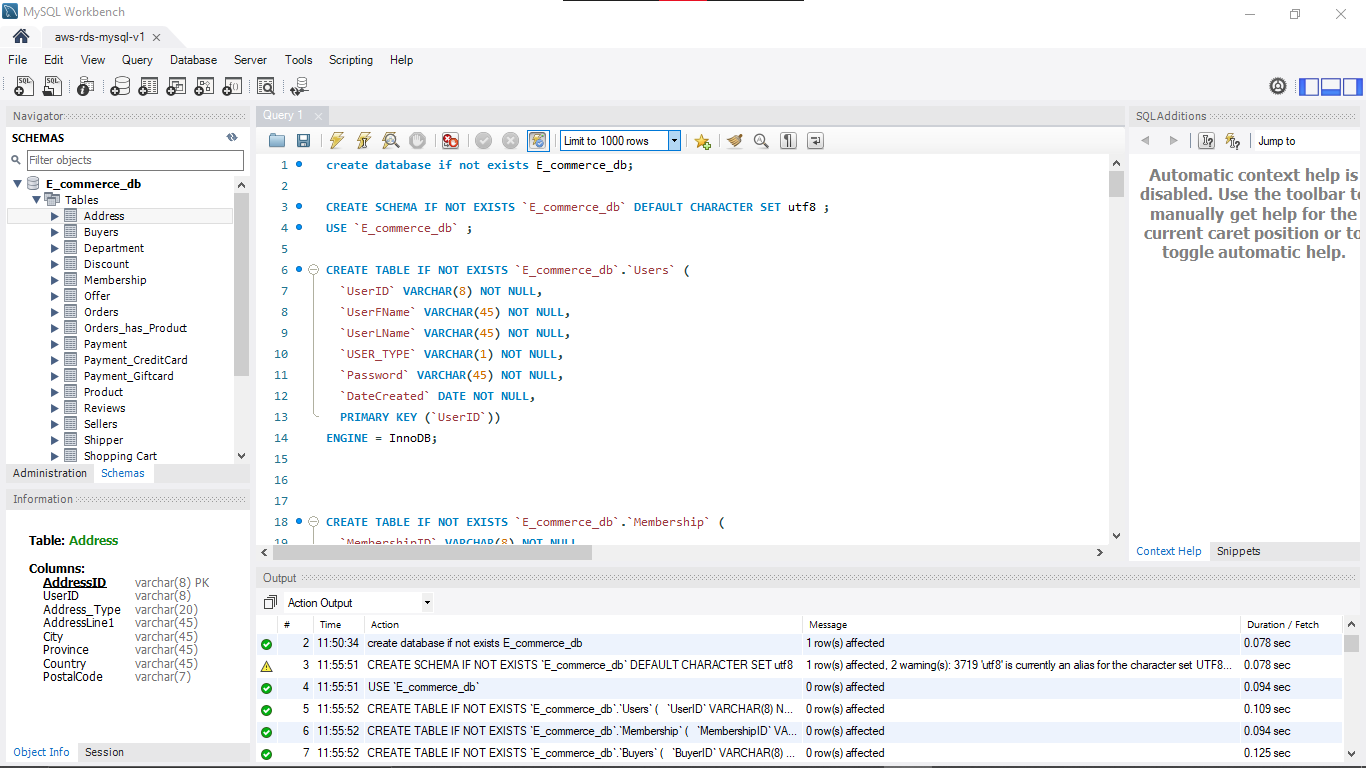
* 1. **Data Requirement**

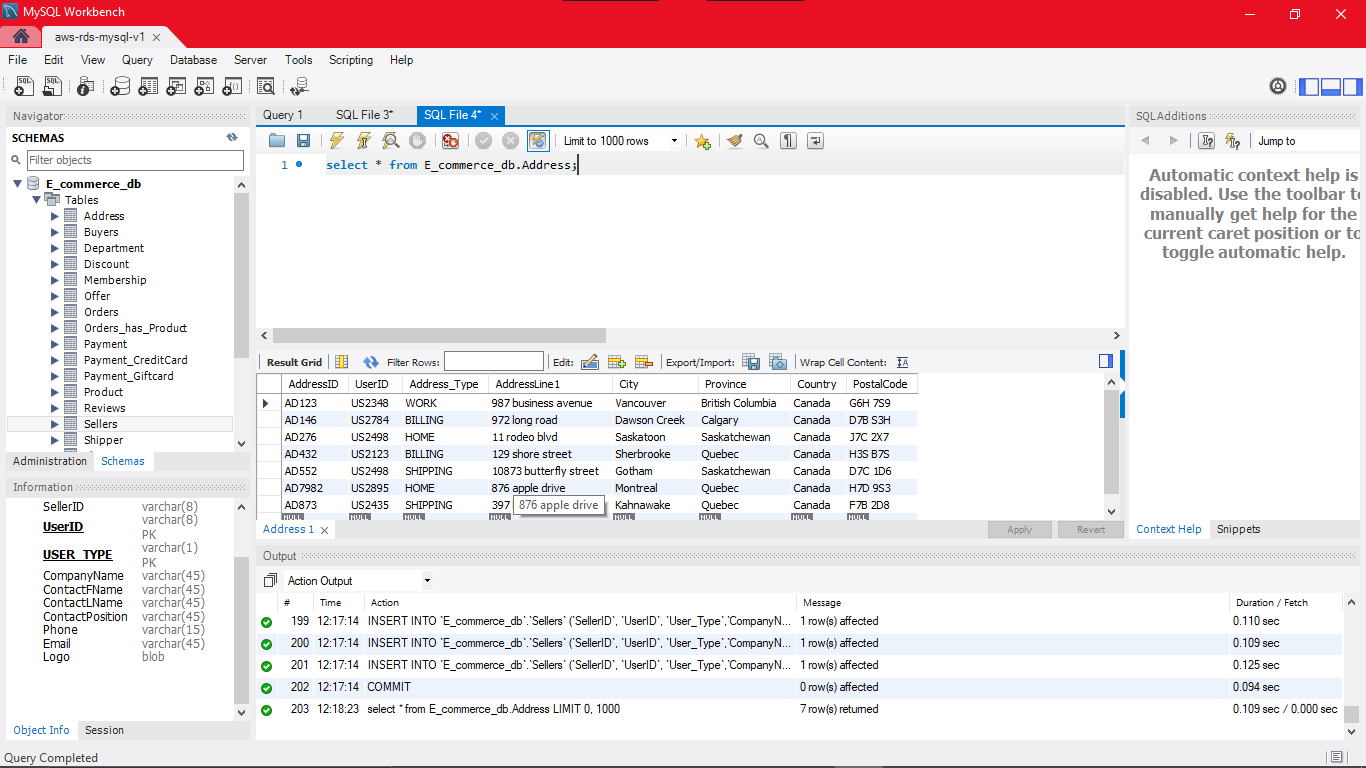
Here Data is in MySQL database which is in structured format i.e, rows and columns and this data dump at the time of e-commerce database creation having well defined schema.

This data is provided by Ineuron please go through: <https://astra.dev/ineuron>

**2.4 AWS RDS**

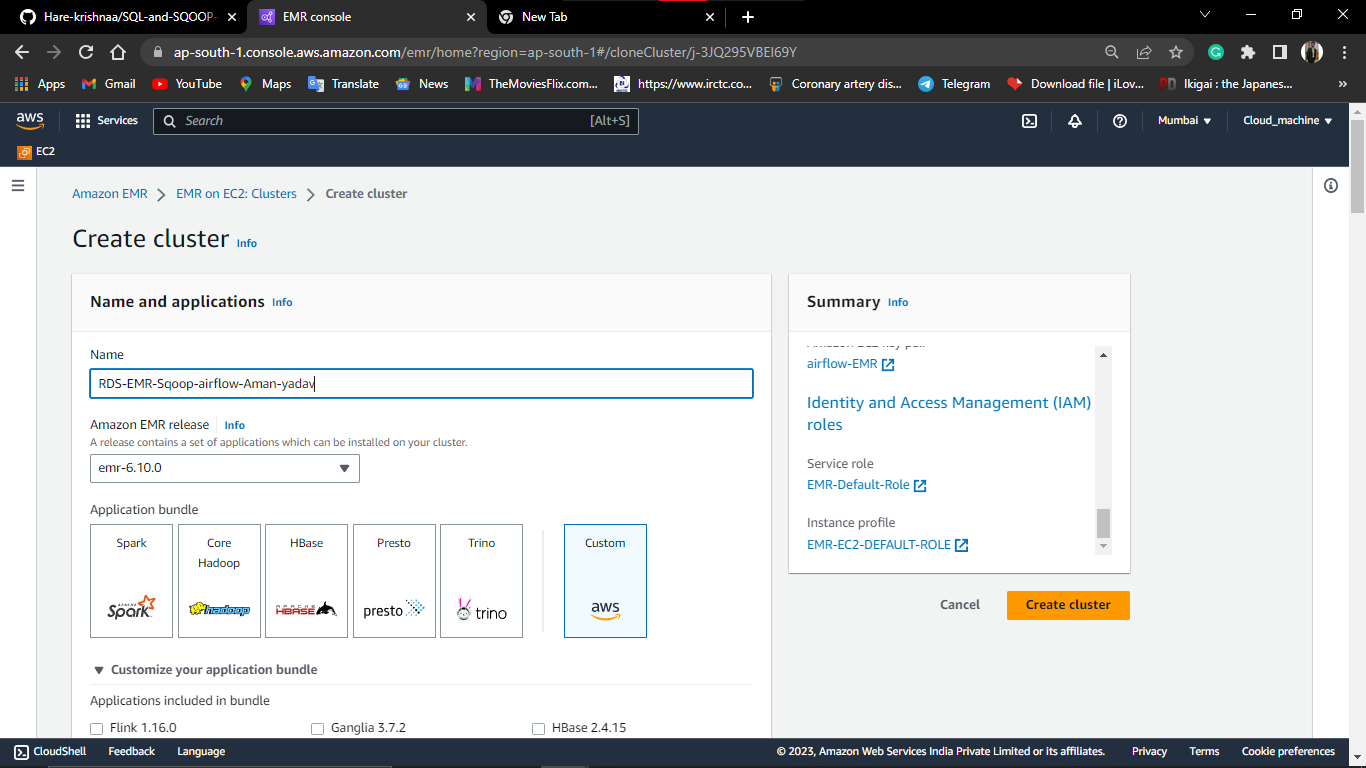
1. After setting up AWD RDS on cloud the next step is to create the E-commerce Schema which is in TableSchema.sql file.
2. In this step connect AWS RDS to your MySQL Workbench or SQL Workbench providing host, username and password which is available at the time of creating the AWS RDS cloud database.
3. Please Keep in mind that enable the “Publicly accessible”: Yes, otherwise it will not be connect to your workbench.
4. Now copy the all the commands from TableSchema.sql and paste into MySQL workbench and run it.
5. Now copy the records which is present in TableRecords.sql file and run it.
6. Just cross verify with any select query statement and see the result window.
7. **Our first step is Completed.**
8. **There are some snapshots of MySQL workbench showing the commands.**
9. **Images while creating the E-commerce database on MySQL Workbench.**

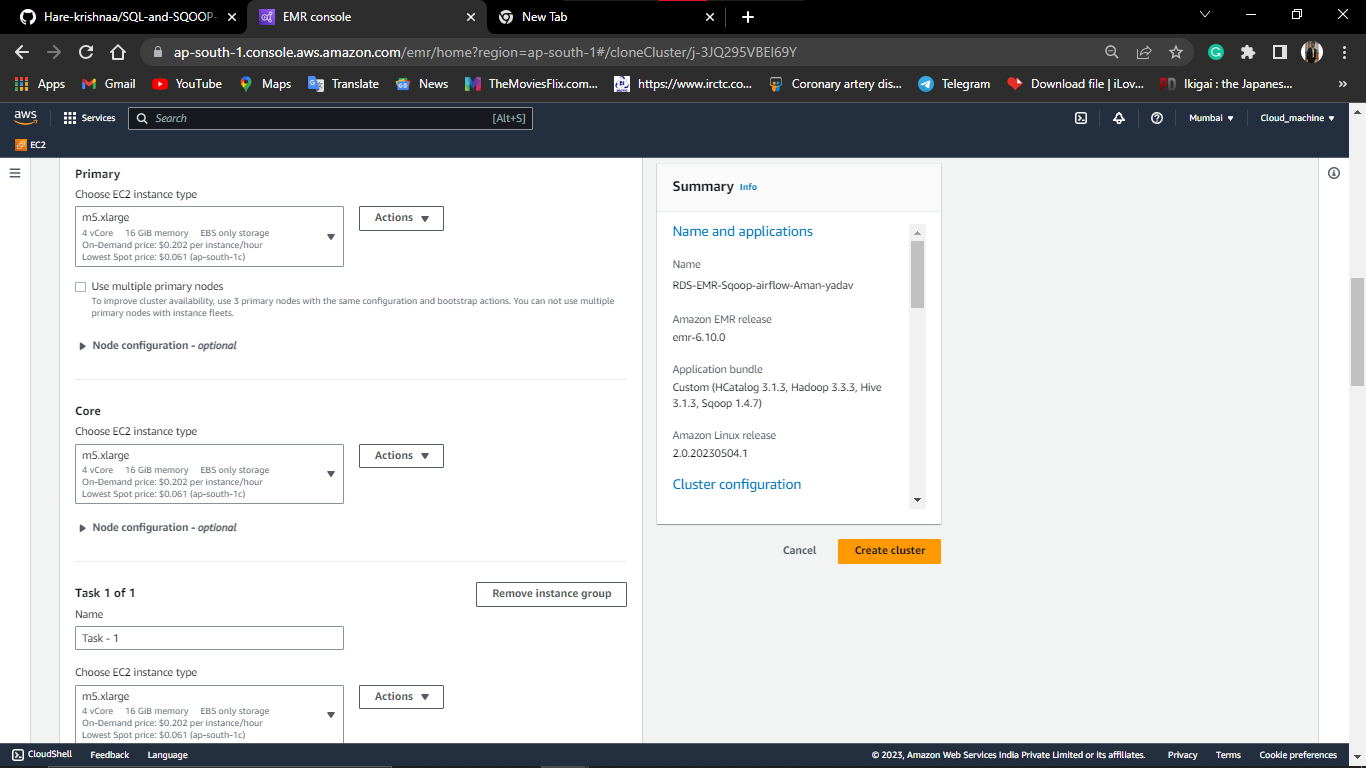




**2.5 AWS EMR**

1. Now after setting up the AWS RDS and create the E-commerce database followed by loading records, the next step is to create the EMR cluster on AWS.
2. In this project I have integrate EMR with Hadoop, Hive, Spark, Hcatalogs and Sqoop. Along with configuration I have also install the Mysql-connector.jar through Bootstrap action.
3. Images related to setting up the cluster: -





1. After start the cluster, configure the security groups i.e.
2. Add one SSH connection having port 22 and traffic Anywhere ipv4.
3. Add one MYSQL connection having port 3306 and traffic Anywhere ipv4.
4. Demo to create the EMR cluster:
5. Name: RDS-EMR-Sqoop-airflow-v2
6. Amazon EMR release: emr-6.10.0
7. Application bundle: Custom (HCATALOGS 3.1.3, Hadoop 3.3.3, Hive 3.1.3, Sqoop 1.4.7)
8. Amazon Linux release: 2.0.20230504.1

Cluster configuration

1. Instance groups: Primary (m5.xlarge), Core (m5.xlarge), Task (m5.xlarge)
2. Provisioning configuration

1. Core size: 1 instance

1. Task size: 1 instance

Networking

1. VPC: [vpc-044d52213...](https://ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#VpcDetails:VpcId=vpc-044d52213ae495055)
2. Subnet: [subnet-0125de...](https://ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#SubnetDetails:subnetId=subnet-0125deb2e18c1e3fe)
3. Primary node security group: [sg-02a003c441...](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SecurityGroup:groupId=sg-02a003c441db2404a)
4. Core node security group: [sg-0545a1c61c...](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SecurityGroup:groupId=sg-0545a1c61c571c6b0)

Cluster termination

Cluster termination: Terminate cluster after Idle time: 1 hour

Tags – *optional*

1. Number of tags: 1 tag

Security configuration and EC2 key pair – *optional*

1. Amazon EC2 key pair: [airflow-EMR](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#KeyPairs:key-pair-id=key-0cfd1a4461aff6918)

Identity and Access Management (IAM) roles

1. Service role: [EMR-Default-Role](https://console.aws.amazon.com/iamv2/home#roles/details/EMR-Default-Role)
2. Instance profile: [EMR-EC2-DEFAULT-ROLE](https://console.aws.amazon.com/iamv2/home#roles/details/EMR-EC2-DEFAULT-ROLE)

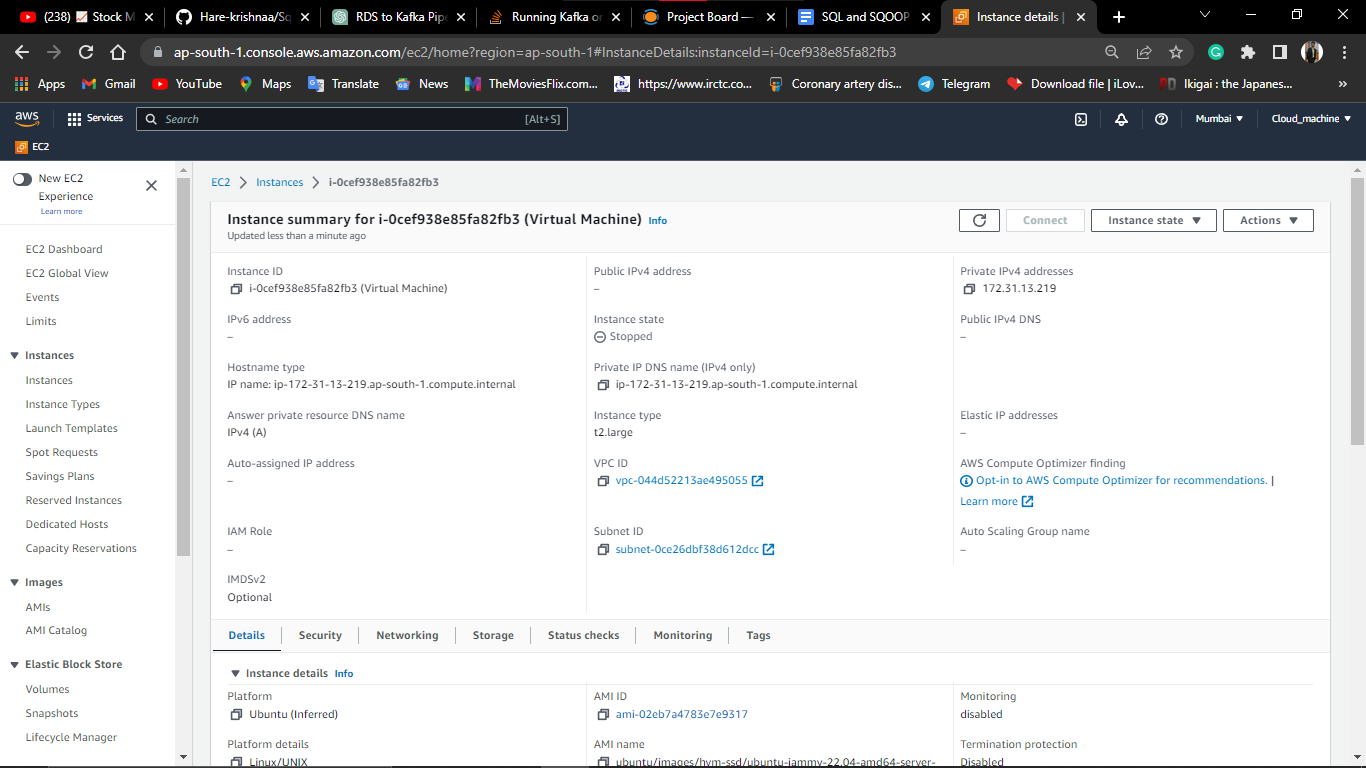
Note: Please do not copy exact same details just take a reference to this for creating your cluster.

1. Here In IAM roles I have add two policies i.e, AWS S3 full access and AWS EMR full access.

**2.6 AWS EC2(Apache Airflow)**

Steps to Creates a AWS EC2 and installed the airflow:

1. Create the AWS EC2 having t2.micro system configuration and add the IAM role.
2. Now make a SSH connection using Putty and add the SSH security group for successful connection.
3. Here some images during creating the instance:



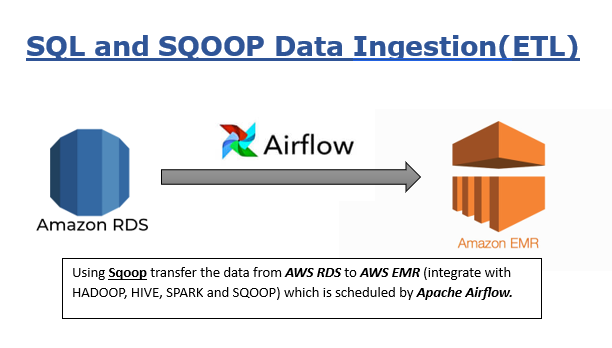
1. Following steps to install the python and Apache airflow:

* 1. sudo apt-get update
  2. sudo apt install python3-pip
  3. sudo pip install apache-airflow
  4. sudo pip install pandas

1. Now just run the “airflow standalone” and a demon process will start.

**3 Design Detail**

**3.1 Process Flow**



**4. Conclusion**

After successfully run all the amazon services and establish the connection to your local machine. I have run the Apache airflow and schedule the Sqoop job on daily basis so that whatever new data came in AWS RDS (MySQL database) it will automatically run the command and a new part file will creates in Hadoop.

Hence A simple Sqoop pipeline is working properly.