**CPSC-531 Adv. Database Management Systems**

**Topic – Big Data Analytics for Insurance Company**

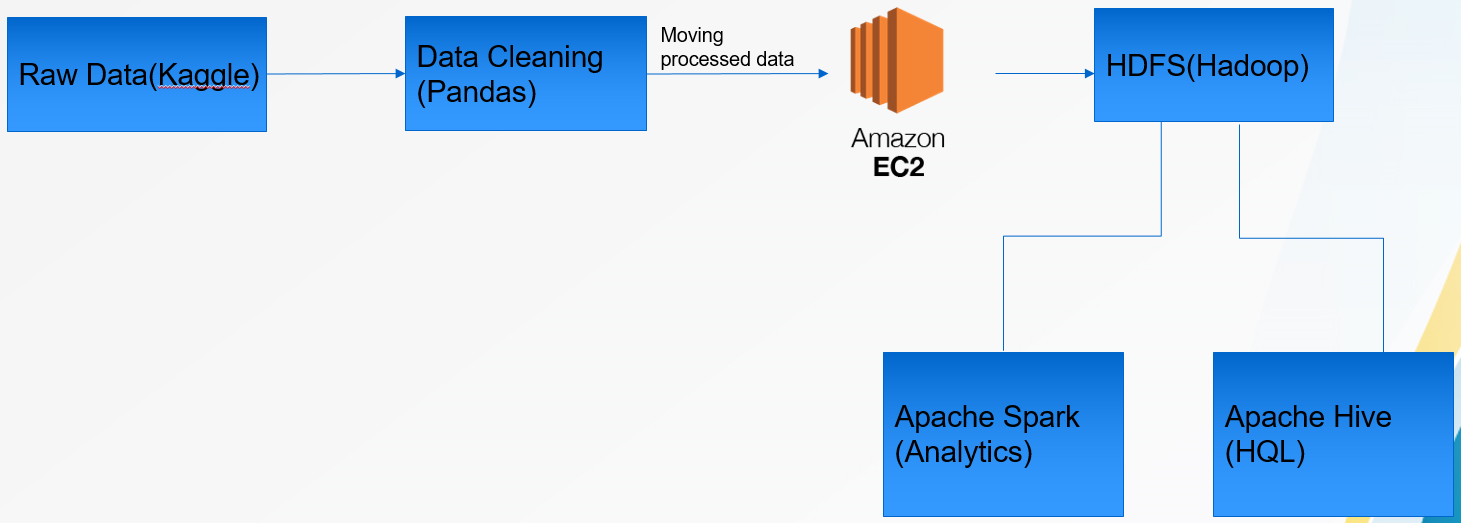
**Functionalities** –

We performed the big data analytics for Insurance Company based on the following use cases which help the company to analyze their revenue –

1. Which sourcing channel has generated the Max revenue for the company?
2. Which sourcing channel has got best customers with best possible premium policies (application underwriting score vs. sourcing channel).

Our project will perform the analysis based on this use cases and perform the analytics using Apache Spark.

**Architectural Diagram –**



Architectural flow of project -

1. Data Gathering (Raw data) from kaggle.
2. Data cleaning of raw data using pandas.
3. Moving data from local machine to hadoop clusters in AWS EC-2 instances.
4. Loading data from hdfs AWS EC2 instances into Apache Spark for analytics.
5. Loading data from hdfs AWS EC2 instances into Apache Hive for analytics

Technologies Used to implement our project –

1. AWS EC2
2. Pandas for data cleaning
3. Apache hadoop
4. Apache Hive
5. Apache Spark- Analytics
6. Stand-alone configuration using Apache Ambari (Horton Works Sandbox)

Configuration required for setting up AWS EC-2 instances –

Our team had set-up clusters on AWS EC-2 free tier.

**Steps to set-up EC-2 instances –**

**IP-Address for every node set-up on AWS -**

**Name Node** ec2-54-189-101-173.us-west-2.compute.amazonaws.com

**SNN** ec2-18-237-128-174.us-west-2.compute.amazonaws.com

**DataNode1** ec2-54-149-20-91.us-west-2.compute.amazonaws.com

**DataNode2** ec2-35-165-118-140.us-west-2.compute.amazonaws.com

**Copying the .pem file to every node -**

1. ssh -i AWSEC2.pem ubuntu@ec2-54-189-101-173.us-west-2.compute.amazonaws.com nn
2. ssh -i AWSEC2.pem ubuntu@ec2-18-237-128-174.us-west-2.compute.amazonaws.com snn
3. ssh -i AWSEC2.pem ubuntu@ec2-54-149-20-91.us-west-2.compute.amazonaws.com d1
4. ssh -i AWSEC2.pem ubuntu@ec2-35-165-118-140.us-west-2.compute.amazonaws.com d2

**Configuring hadoop - hdfs on AWS EC-2:**

1. install java sudo apt-get -y install openjdk-8-jdk-headless
2. wget link.taz
3. tar xvzf filename
4. mv extracted filename hadoop --->
5. sudo vi ~/.bashrc --> no need to change

**Exporting it to the PATHS:**

1. export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64
2. export HADOOP\_HOME=/home/ubuntu/hadoop
3. export HADOOP\_CONF=$HADOOP\_HOME/conf
4. export PATH=$PATH:$JAVA\_HOME:$HADOOP\_HOME/bin
5. source ~/.bashrc

**Commands to connect data nodes and name node**

1. ssh-keygen
2. copy key generated to other nodes from name node
3. scp -i AWSEC2.pem /home/ubuntu/.ssh/id\_rsa.pub ubuntu@ec2-18-237-128-174.us-west-2.compute.amazonaws.com:/home/ubuntu/.ssh/id\_rsa.pub (snn,d1,d2 from nn)
4. cat ~/.ssh/id\_rsa\_pub >> ~/.ssh/authorized\_keys (all nodes )
5. vi ~/.ssh/config

After configuring the Name Node, Secondary Name Node and Data Nodes commands to run the services hdfs and yarn:

1. ssh -i AWSEC2.pem ubuntu@ec2-35-88-101-162.us-west-2.compute.amazonaws.com
2. hdfs namenode -format
3. $HADOOP\_HOME/sbin/start-dfs.sh
4. $HADOOP\_HOME/sbin/start-yarn.sh

Apache Ambari – Stand-Alone clusters –

**Steps to set up Apache Ambari –**