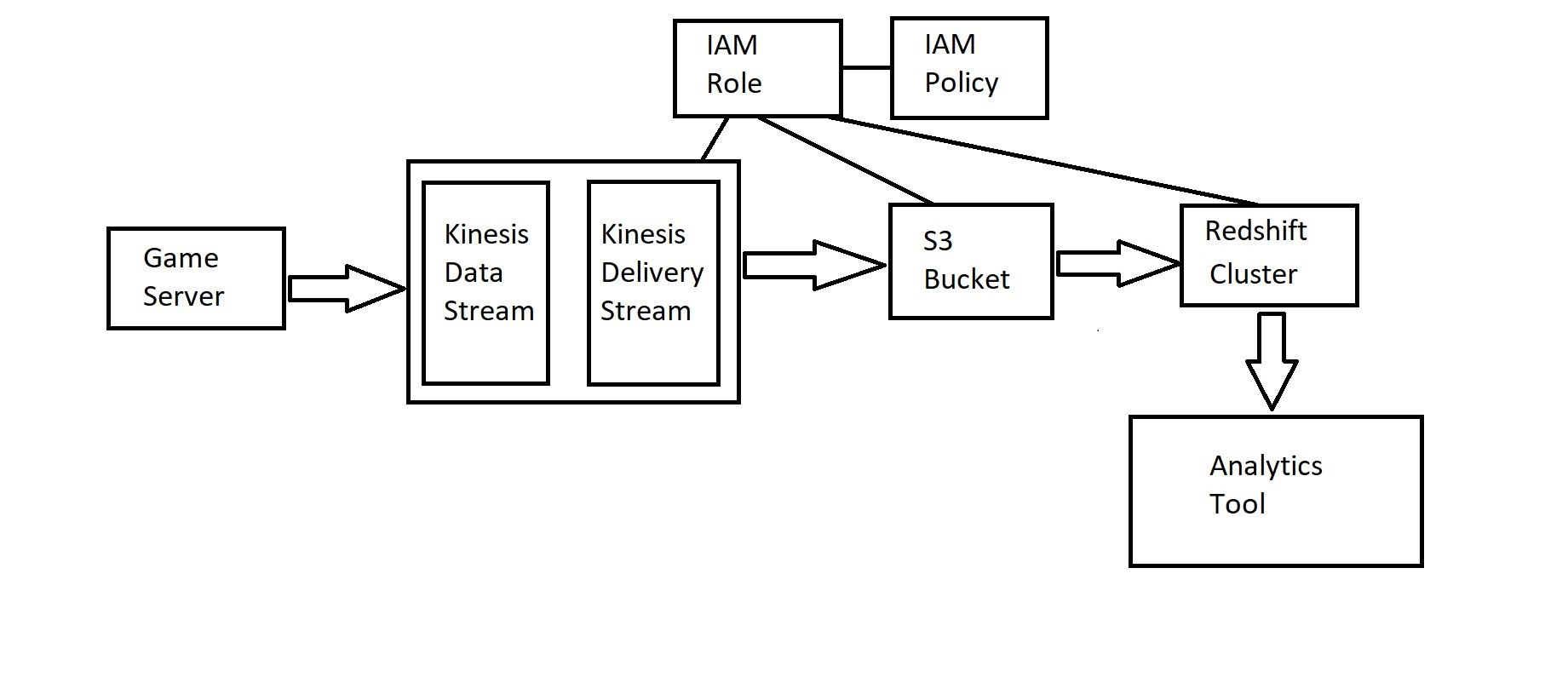


INFO7374 - Digital Marketing and Analytics

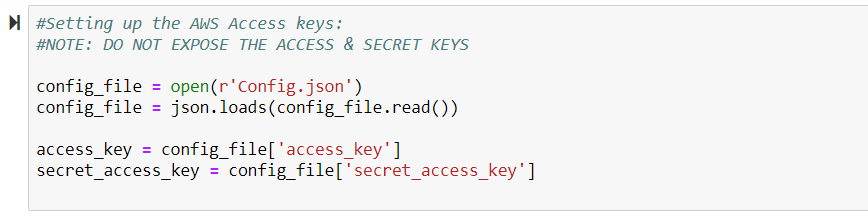
Assignment 1

**Architecture:**



**Setting up AWS access keys:**

* Set up the configuration using the AWS access key and secret access key

****

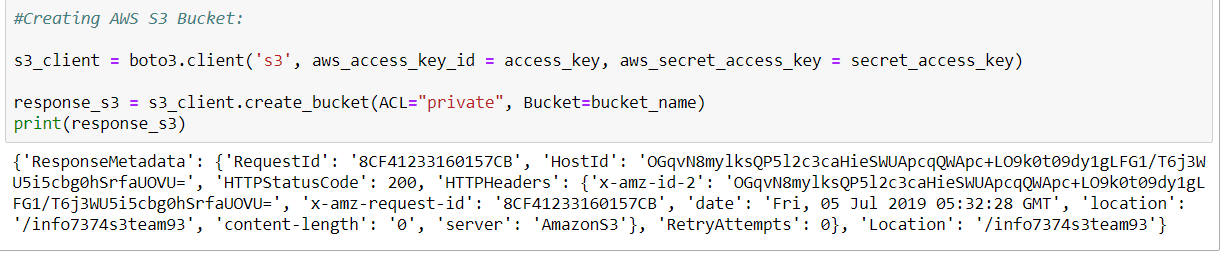
* Mention the names for the configuration parameters like bucket name, policy name, roles, database names and policies

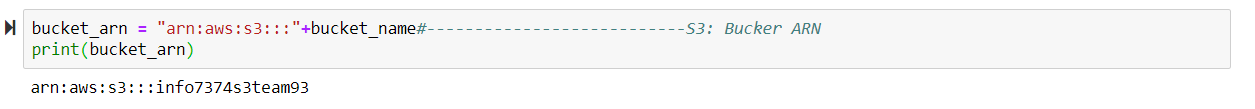
****

**AWS S3 Bucket:**

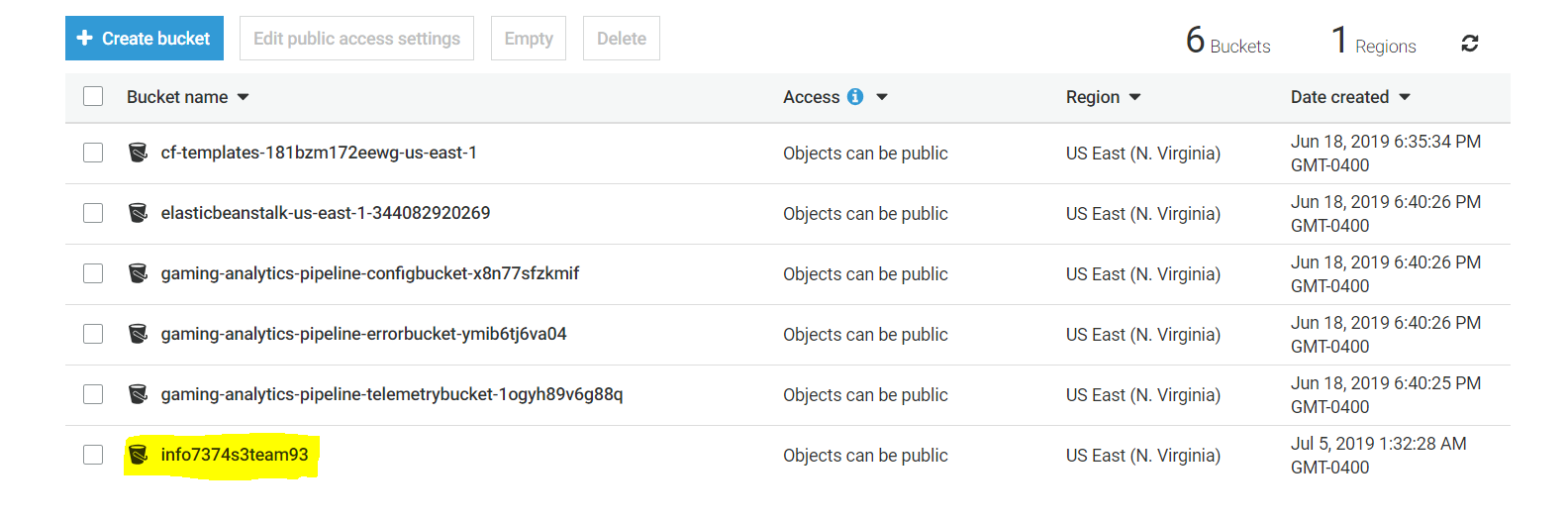
****

* To create an S3 Bucket with the help of python script, we require the AWS ACCESS KEY and AWS SECRET ACCESS KEY
* Provide a relevant name for the bucket.





* Go to the AWS console and under S3 services check if the S3 bucket is created.



**IAM:**



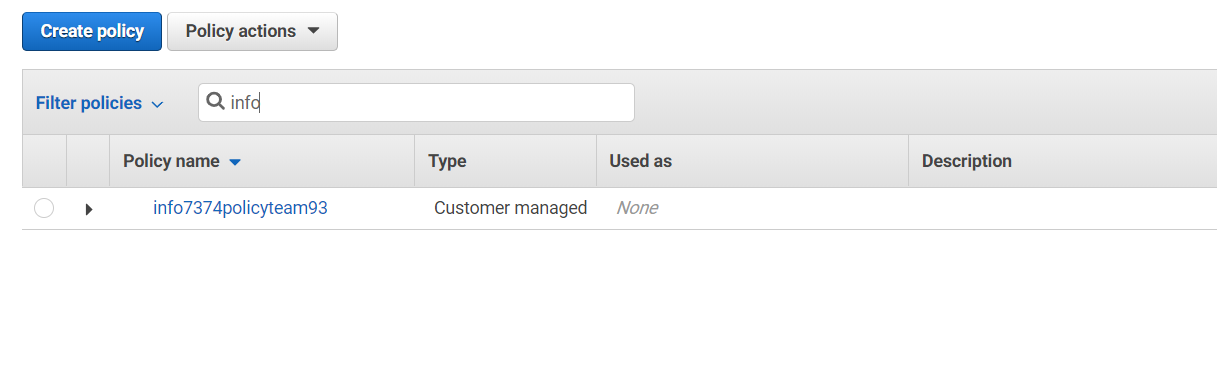
**IAM Policy:**

* Create an IAM policy for architectural access and we can check the IAM policy assigned too





* Check in AWS Console under IAM services if the policy has been created

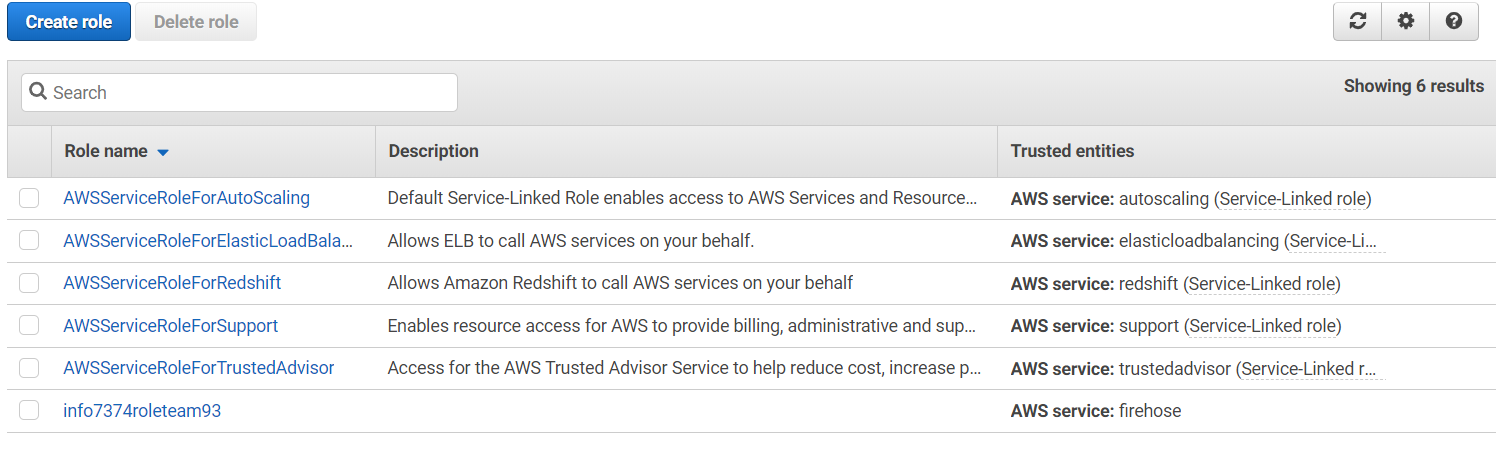


**IAM Role:**

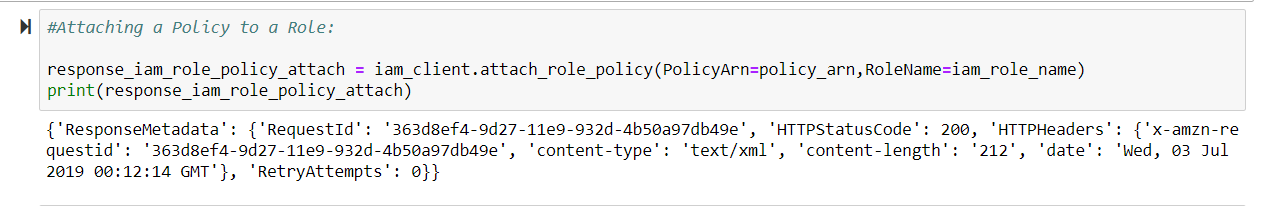
* Then create an IAM Role which is an AWS firehose service and assign it



* Check in the AWS console and confirm if the role has been created



* Now attach the policy to the role



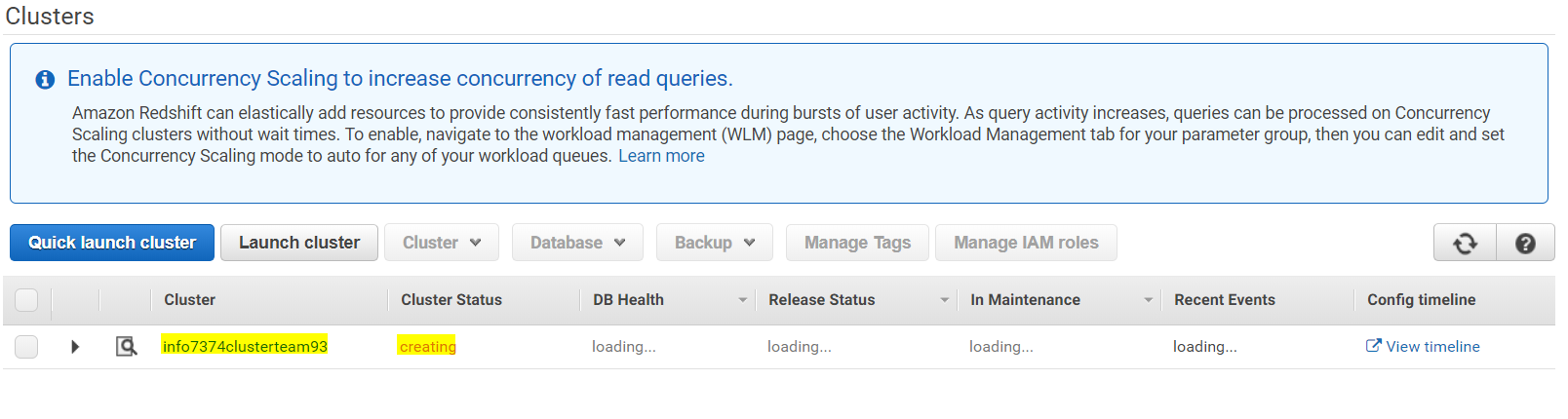
**AWS Redshift Cluster:**

****

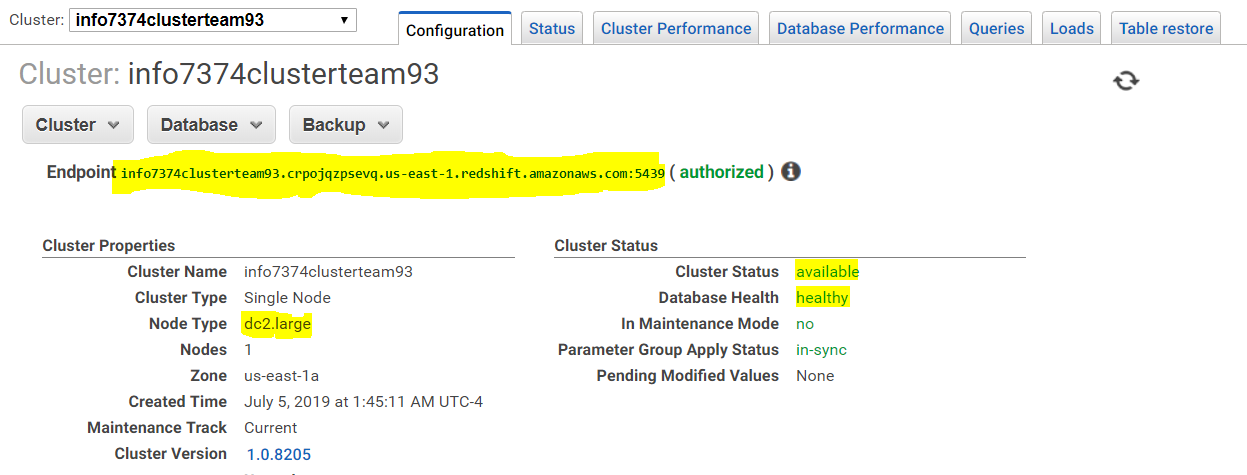
* Create a Redshift Cluster with the help of the python script using the necessary input parameters like the region name and the access keys
* Once the cluster is created then describe the cluster to get an endpoint



* It takes a while to create the cluster, as seen in the screenshot below the status us creating



* Once the AWS Redshift cluster is created open the cluster and we can see in the console that the status is **available**  and the database health is **healthy**
* We can also get the cluster endpoint details here



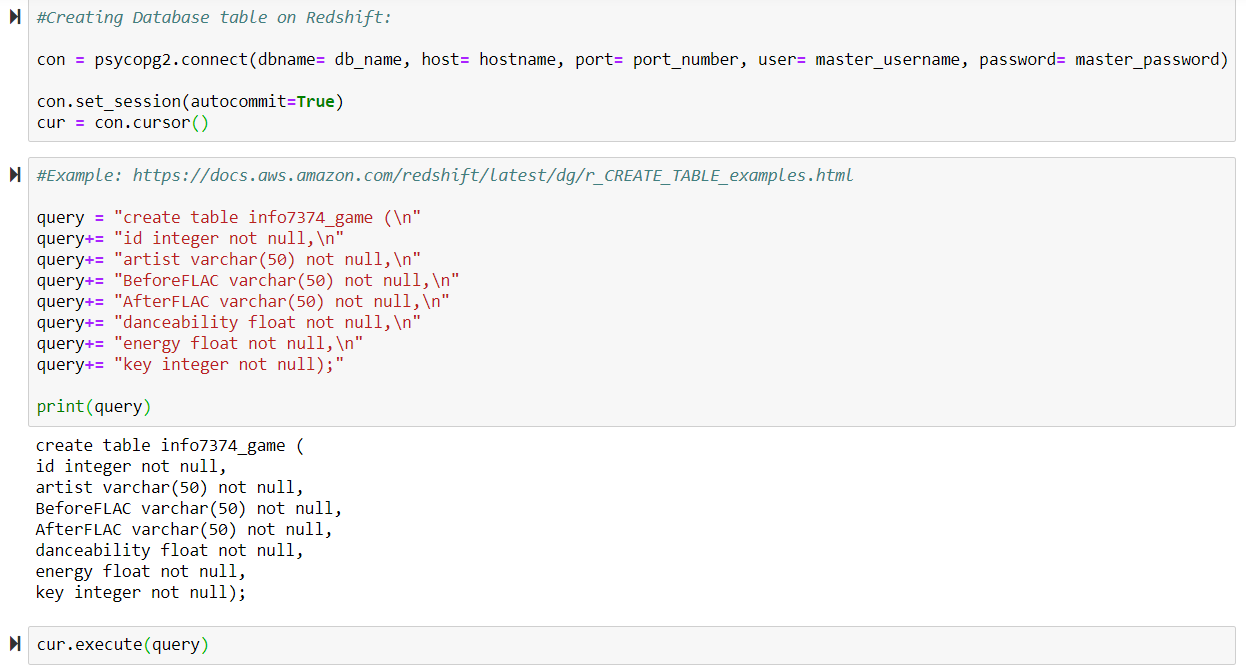


**Schema:**

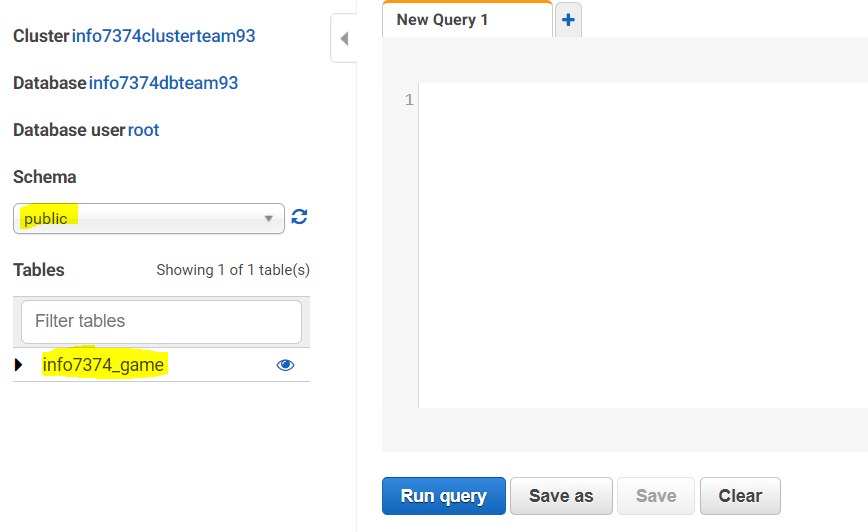
* The Schema of our assignment is:
* Create a database and then create the required tables based on the schema

**Database:**

* Create a Database using the python script below in which our cleansed data will be stored



* Now go to AWS Redshift services in the console and under the cluster check if the table is created.

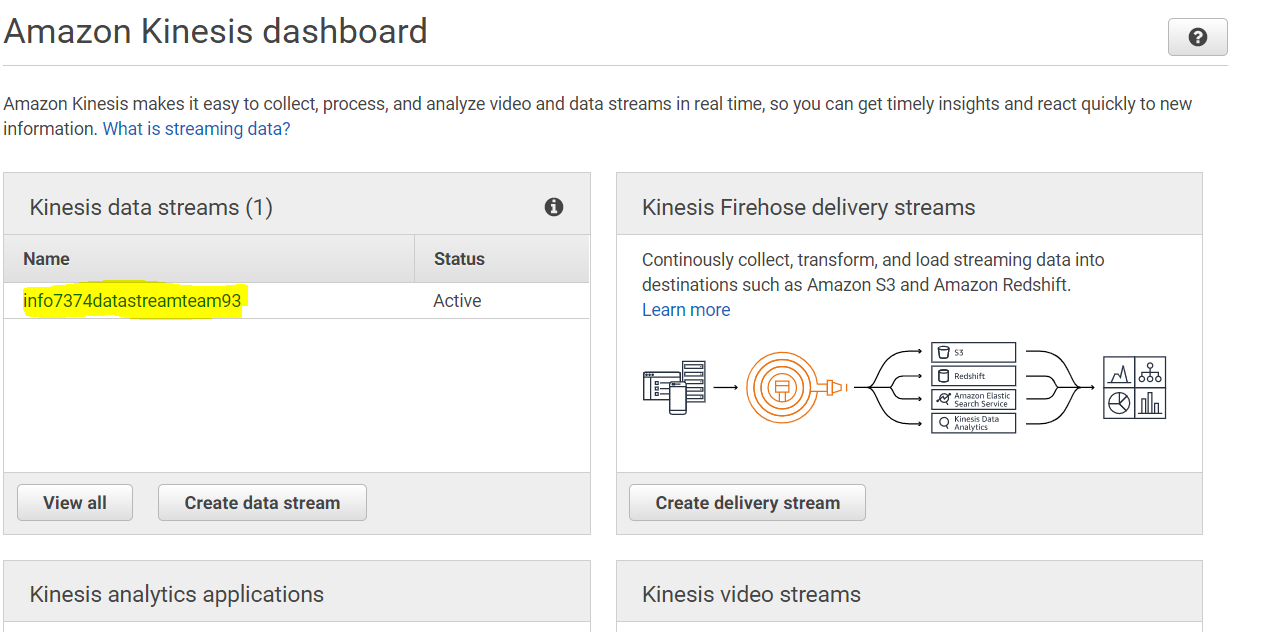


**Kinesis Data Stream:**

* Create a Kinesis Data Stream

****

* Verify in the AWS kinesis dashboard services if the data stream is created.

****

**Kinesis Delivery Stream:**

* Create a Kinesis Delivery Stream (Firehose) with the help of the python script mentioned below

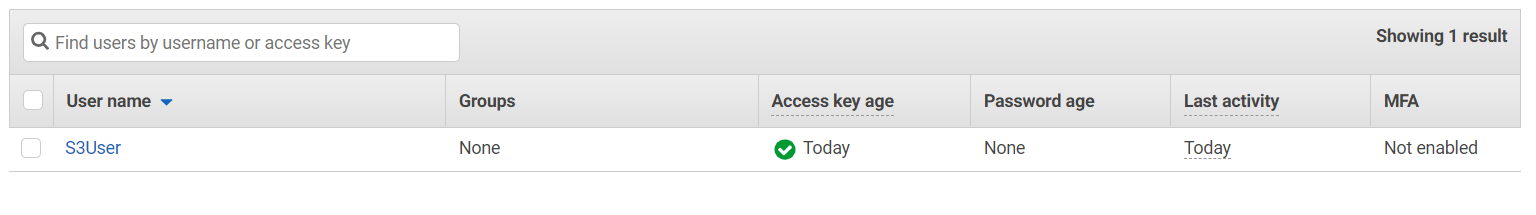


**Data:**

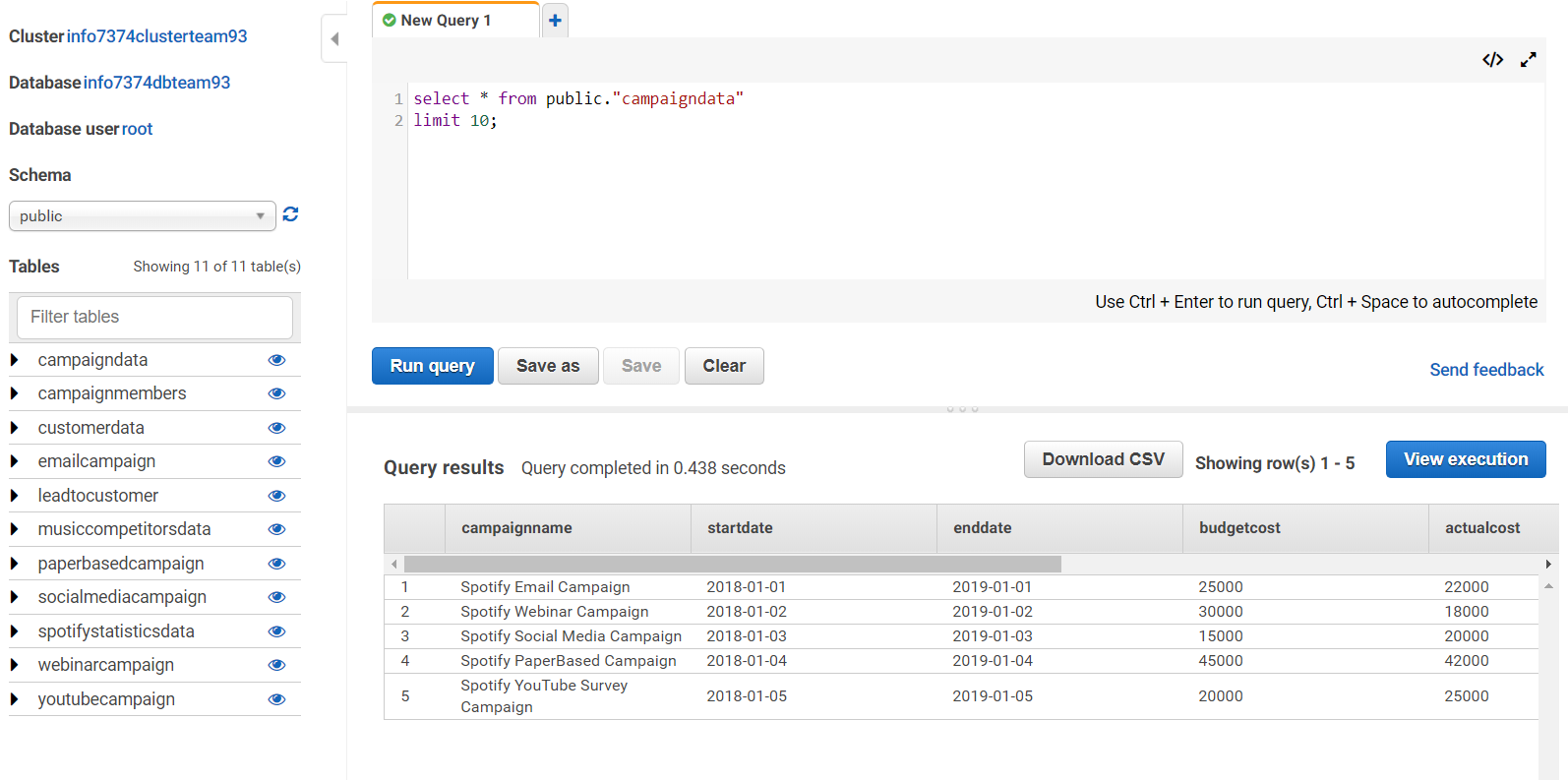
* Now pull in the data into S3 bucket using the kinesis stream

**S3 USER :**

Create a S3 User and provide the respective permissions for S3, EC2, Redshift and Query editor



Query editor:



CREATE TABLE emailcampaign(

months date not null,

leads integer not null,

leadstocustomer integer not null);

CREATE TABLE paperbasedcampaign(

months date not null,

leads integer not null,

leadstocustomer integer not null);

CREATE TABLE webinarcampaign(

months date not null,

leads integer not null,

leadstocustomer integer not null);

CREATE TABLE socialmediacampaign(

months date not null,

leads integer not null,

leadstocustomer integer not null);

CREATE TABLE youtubecampaign(

months date not null,

leads integer not null,

leadstocustomer integer not null);

CREATE TABLE spotifystatisticsdata(

state varchar(10) not null,

freeusersubscirption integer not null,

onemonthsubscirption integer not null,

threemonthsubscirption integer not null,

sixmonthsubscirption integer not null,

familysubscirption integer not null,

yearsubscirption integer not null,

studentsubscirption integer not null);

CREATE TABLE spotify.campaigndata(

campaignname varchar(200) not null,

startdate date not null,

enddate date not null,

budgetcost integer not null,

actualcost integer not null,

type varchar(200) not null,

status varchar(200) not null,

leadsgainedpercent decimal(4,2)not null);

create table spotify.MusicCompetitorsData(

competitor varchar(200) not null,

monthlyfeeforpremium decimal(3,2) not null,

freeOption BOOLEAN,

freeTrialPeriod integer,

musicLibrarySize integer,

maximumBitrate integer,

familySharing decimal(4,2),

studentDiscount BOOLEAN,

studentDiscountamount decimal(3,2),

usmilitaryDiscount BOOLEAN,

offlineListening varchar(200) not null,

radioStations BOOLEAN,

podcasts BOOLEAN,

musicVideos BOOLEAN,

musicLockerFunc BOOLEAN,

flacProvided BOOLEAN,

revenueperyear integer);

create table customerdata(

name varchar(200) not null,

email varchar(200)not null,

subscription varchar(200),

premium varchar(200),

profit decimal(8,3),

usage varchar(200),

flac varchar(200),

customer varchar(200),

state varchar(10));

create table spotify.campaignmembers1(

name varchar(200) not null,

state varchar(200),

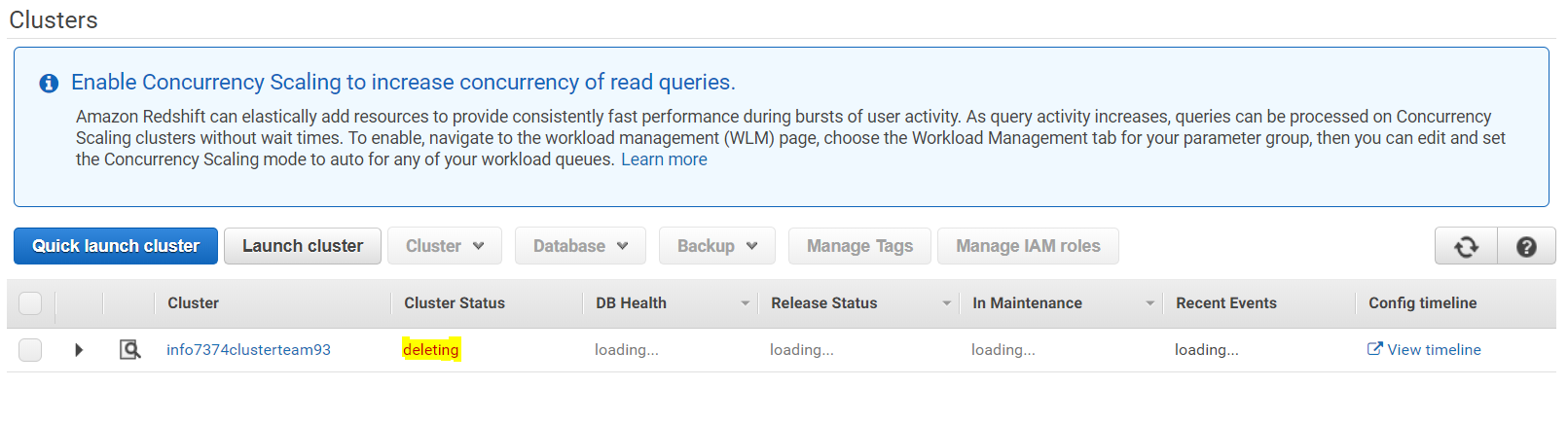
email varchar(200),

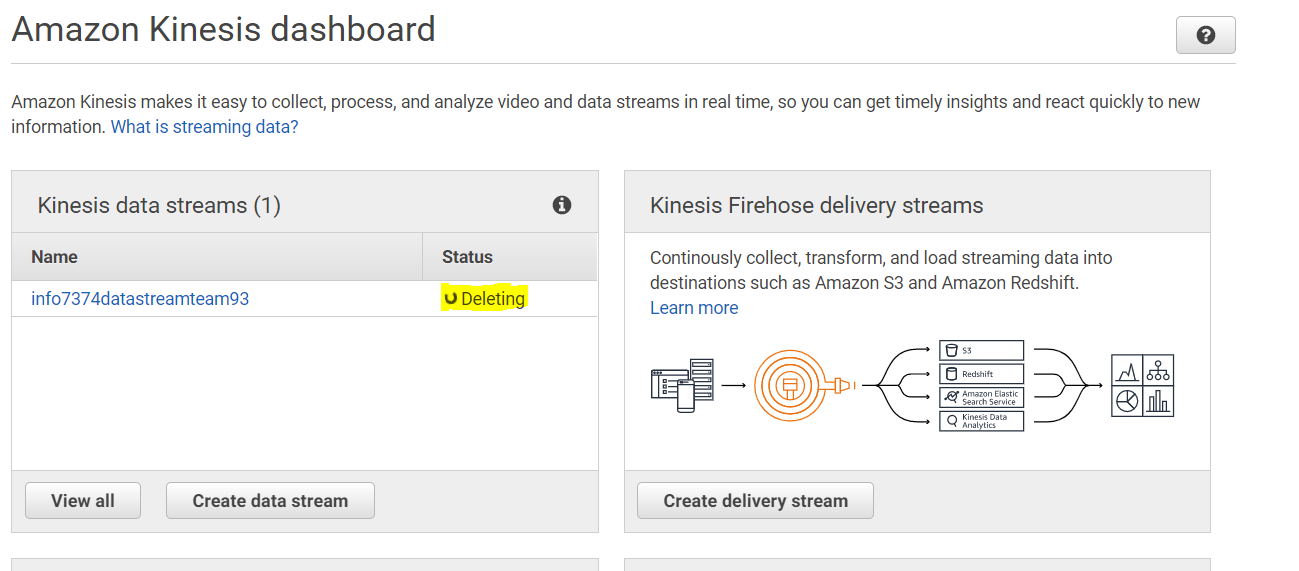
leadstatus varchar(200),

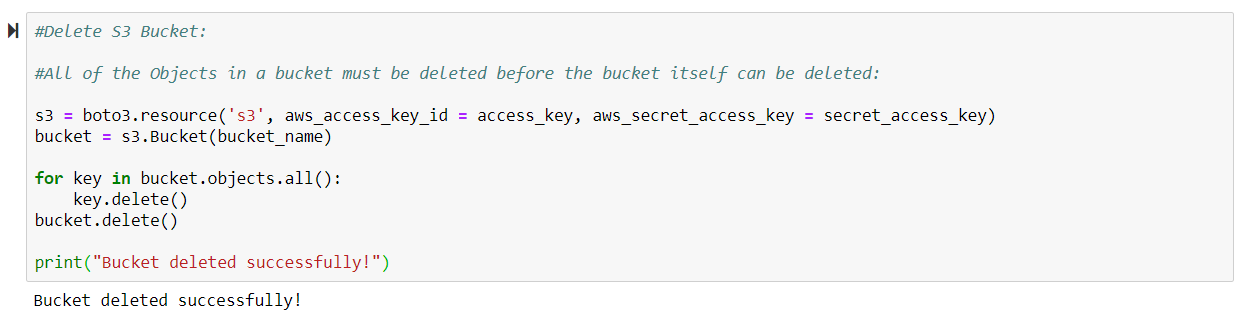
spotifyuser BOOLEAN,

campaigntype varchar(200));





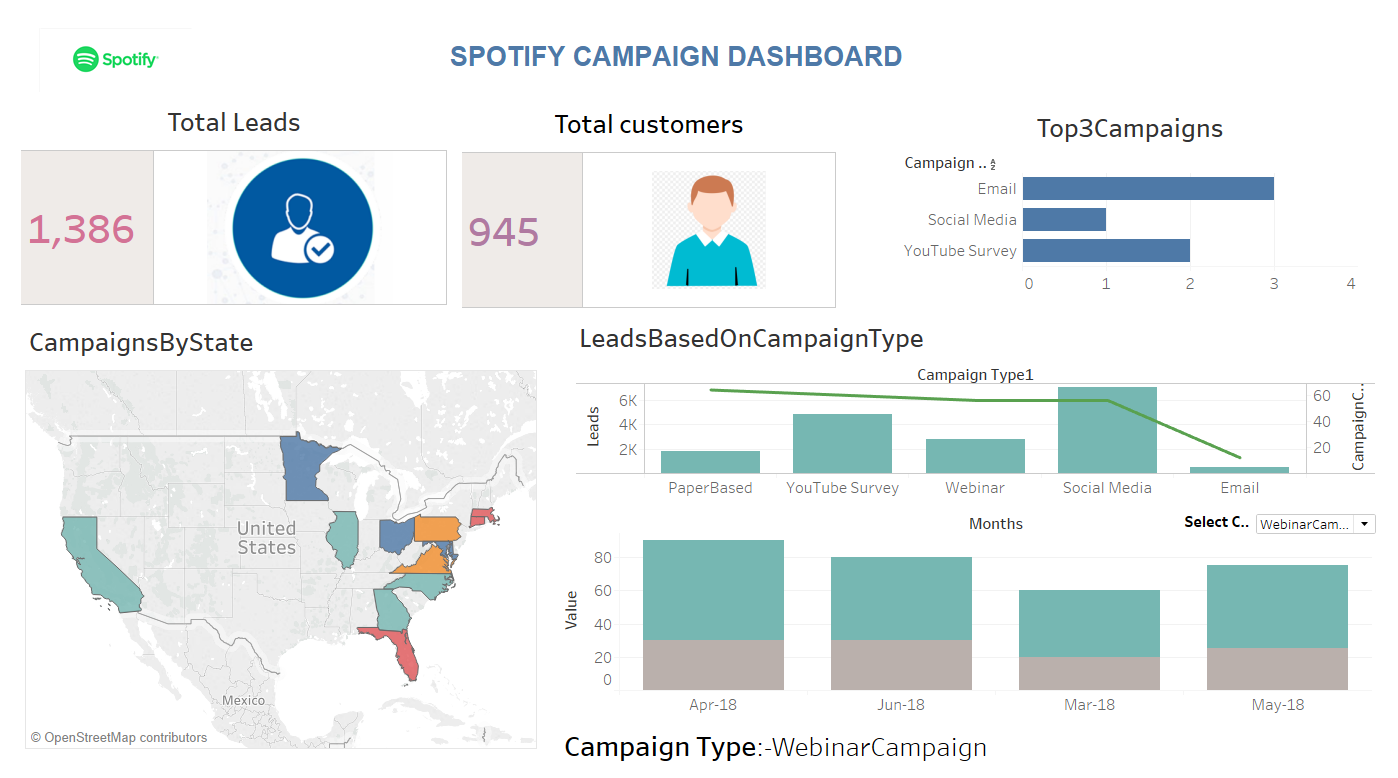




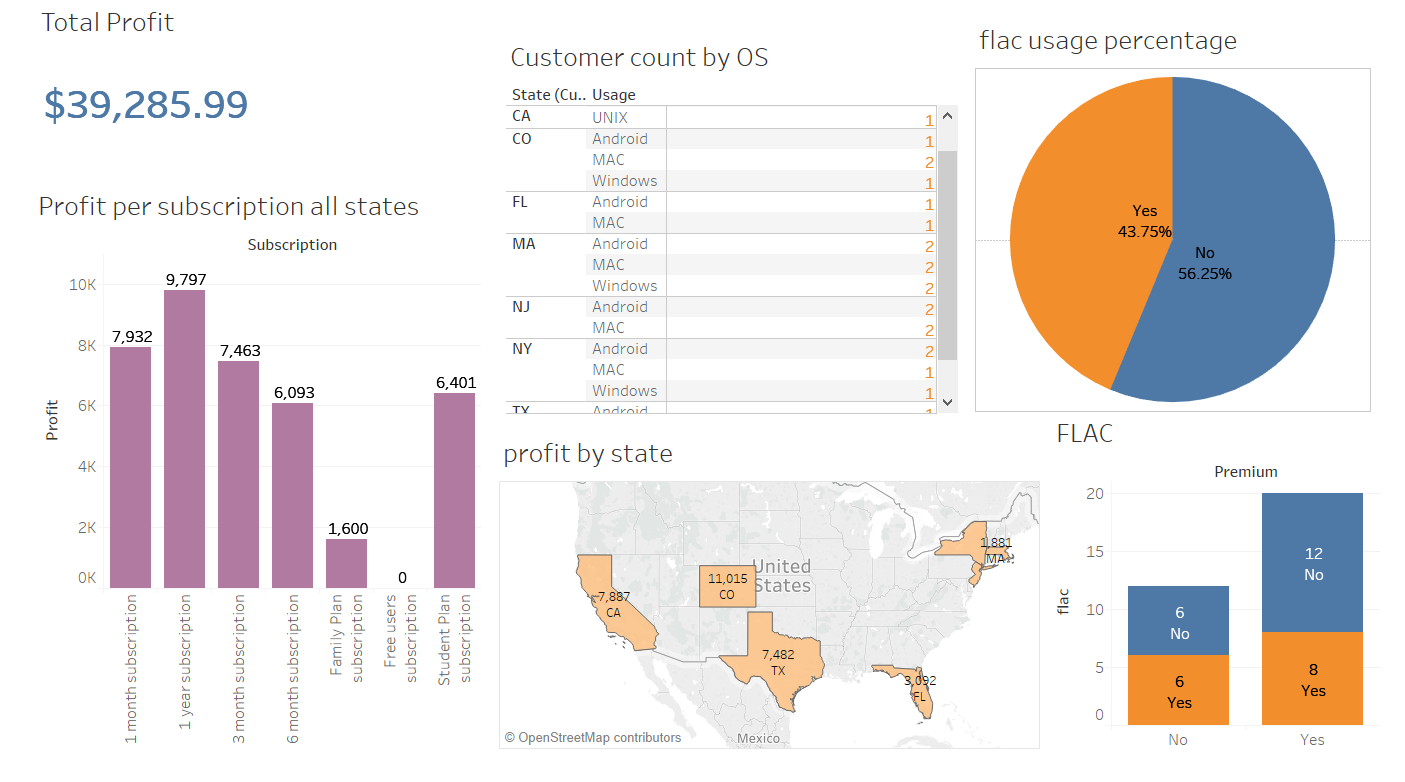
**DATA VISUALIZATION:**

Using Tableau data visualization was done and dashboards were created for each use case.

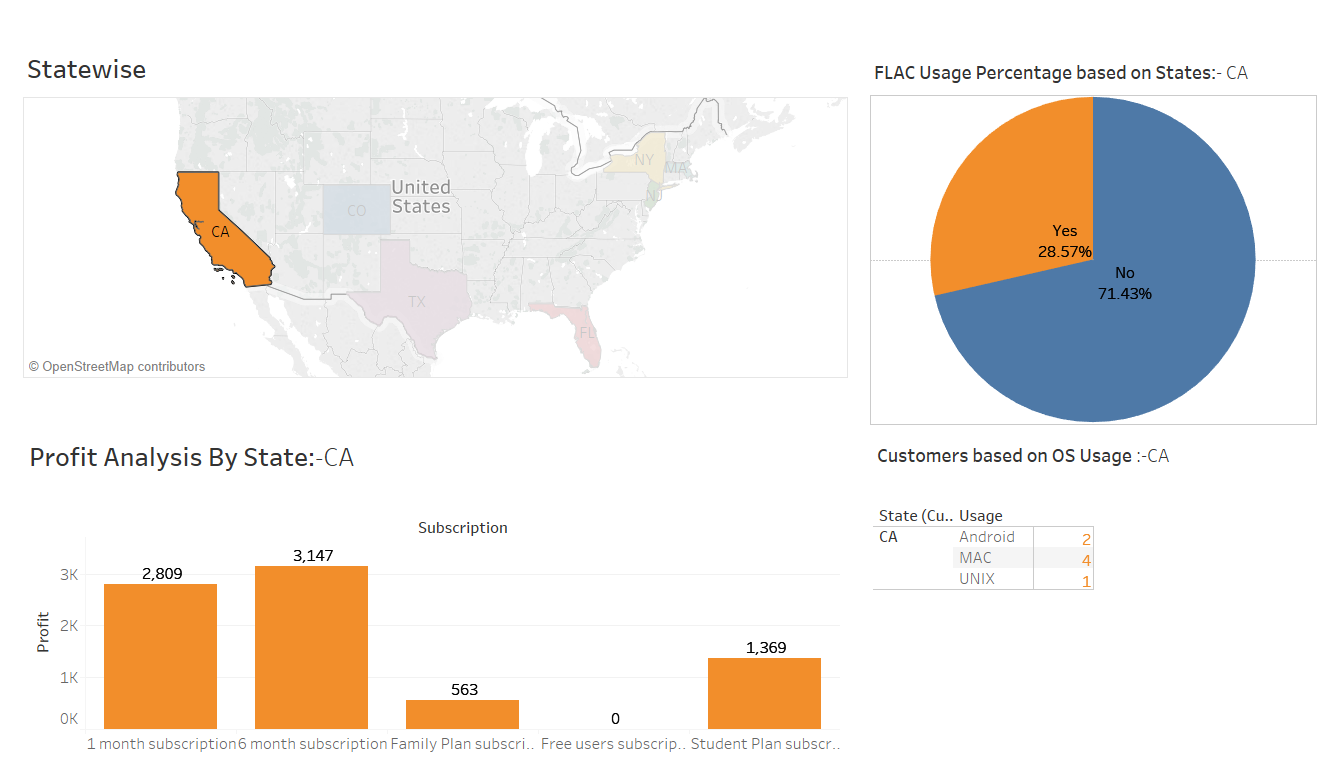
**Use case 1:**

****

**Use case 2:**

****

**Use case 3:**

****

**Use case 4:**

