

Migration of database to cloud services for Citi Bikes

Course:

MIS 730 Integrating IS Technologies

Under the guidance of Distinguished professor:

Jeffrey Nickerson

Team No:

01

Team Members:

Sandeep Moparthy

Saishyam Nandakumar

Prachi Yewale

Hetul Shah

Bo Zhang

Abdulrahman Aldharrab



Contents

Business Strategy

Business Scenarios

Alternative architectures

Class diagram

Deployment diagram

Hardware considerations

Sequence diagrams

Extending and Reuse

Screenshots of the prototype

Business Strategy

Citi bike is a public cycle sharing company that serves various parts of New York City and it consists of numerous fleets of beautifully designed bikes that are durable in nature and can be locked into a network of docking stations which are sited at regular intervals around the entire city.

The user can rent the bike from and return it to any station in the system creating an efficient network with many possible start and end points with a good combination of departure and arrival.

It is a privately owned company serving the New York city of Bronx, Manhattan, Queens as well as Jersey City, New Jersey and planning to expand to several more major cities in the United States. It is named after its lead sponsor CitiGroup.

Citibikes deals with travel data each day, every time a user rents the bike, it stores the information.

Bike related data: Bike ID, Start and end times, Trip ID

Trip related data: Duration of trip

Station related data: Latitude and longitude of the station, Station Name, Station ID, Station city

Now, that this information is currently stored in local machines, we in this document are proposing an extension to their existing system of data storage where instead of having the database in the storage devices, to migrate to a cloud solution which will enable other teams and users to view the analytics and also maintain data integrity at the same time.

Our motive is to help migrate citi bikes from local databases to cloud services to accommodate increasing loads of data for data security and will eventually help to analyse the trends in bike sharing across the United States.

The main functions of the system are as follows:

1. Map Based tracking
2. Analytics dashboard
3. Feedback system

The benefits of the solution are as follows:

- Increased storage
- New database design that accommodates data for citi bike across the United states
- A data warehouse that makes sure read only data available for the analytics team
- Data across the US can be used for analysis readily instead of waiting for data to come from different sources.

Business scenarios

Scenario 1

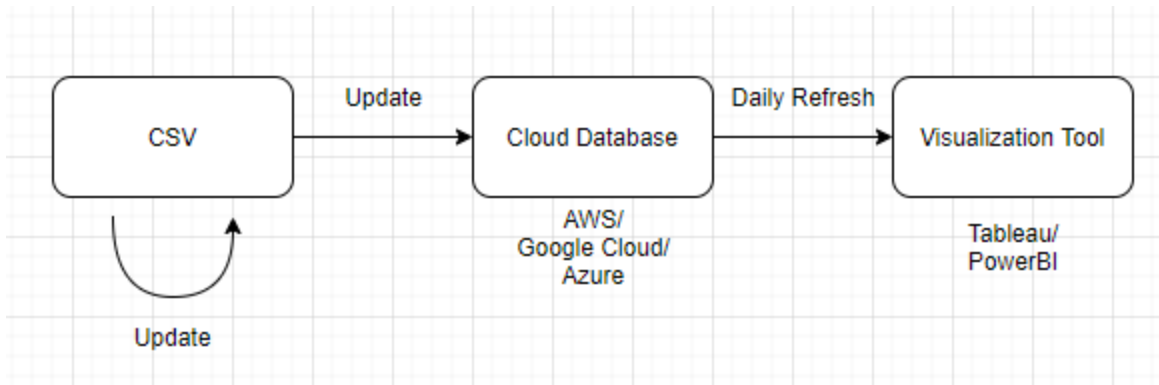
Citi Bikes is maintaining an on-premise CRM system at their backend that continuously collects ride data from customers such as Trip ID, start station, end station, duration of trip, start time, end time, bike id and other customer related information. With major changes in the GDPR (General Data Protection Regulation) act coming into effect, Citi Bike is facing challenges in applying advanced encryption over data since on-premise systems are more susceptible to data leaks . This pushed the Citi Bike to adopt a different approach of Migrating on-premise systems into cloud platforms which guarantees complete data security as well as Compliance. By moving its application to cloud, Citi Bike could leverage other additional powerful features of cloud such as modernizing the data, scaling up the data, reducing the infrastructure maintenance costs etc. This eventually helps Citi Bike not only in earning customer trust but also improving operational efficiency to great extent.

Scenario 2

By Leveraging the google cloud, Citi bikes will have better scalability as well as better response time. Thus, they can use these features to provide a Map based tracking system to their existing customer according to their membership type. They can connect their database to any analytical dishoarding tool and create the analytics and integrate it with their existing application to enhance user experience.

Alternative architectures

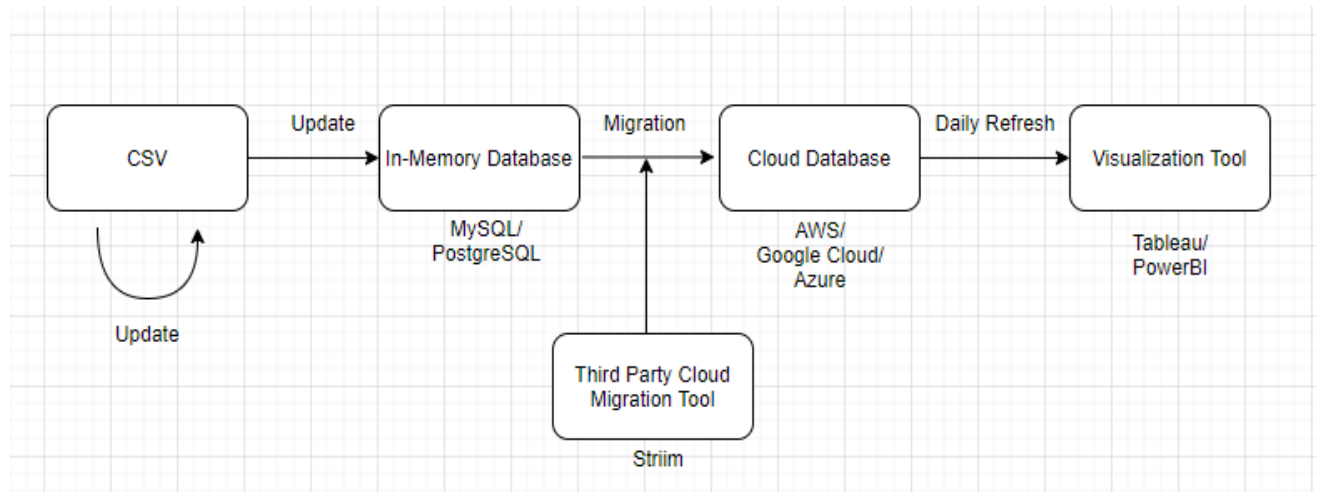
Alternative solution 1



Strength:

- There minimal work to be done and the database reflects the data acquired from different sources.
- Using an existing analytics tool helps in low investment
- Cloud database is usually fast for analysis
- Continuous ingestion of data is avoided because the database is running through out

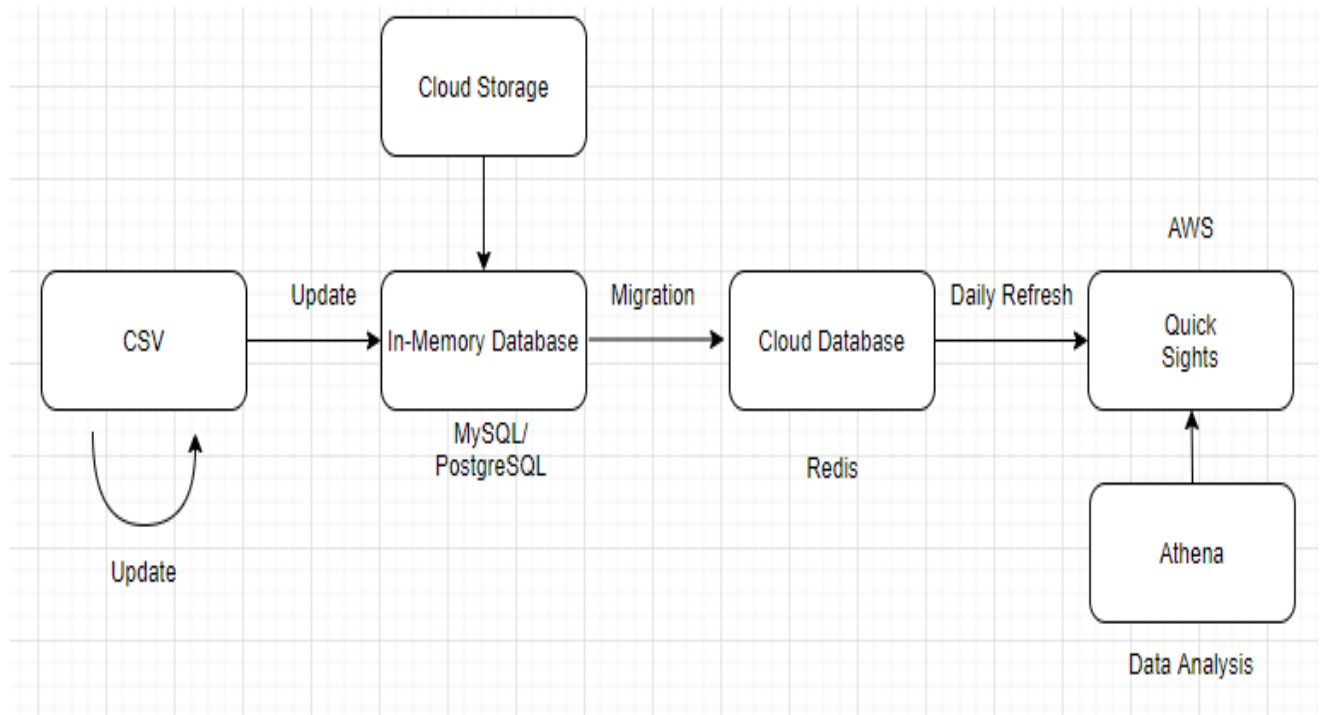
Alternate solution 2



Strengths:

- Migrate workloads to and from any environment very easily – physical, cloud & virtual
- Replicate data continuously to minimize downtime and enable fast cutover.
- Perform Health Checks frequently.
- Continuous Ingestion of data and delivery of data to cloud

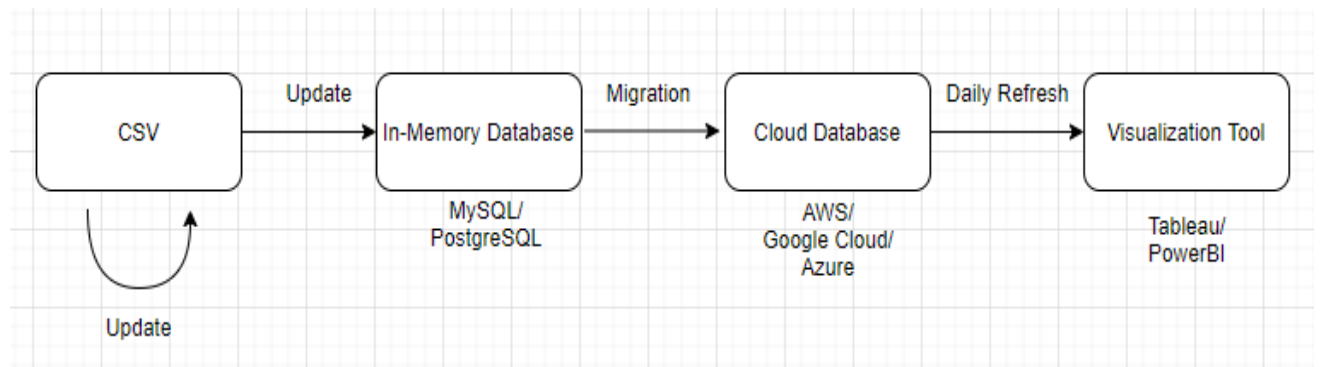
Alternate solution 3



Strengths:

- Local storage helps in times of need like if cloud services are suspended
- Storage is used for the purpose of backup and recovery
- Redis is strong for in-memory analysis which is low latency
- Integration of AWS services in AWS cloud make it a seamless experience for the user

Alternate solution 4



Strengths:

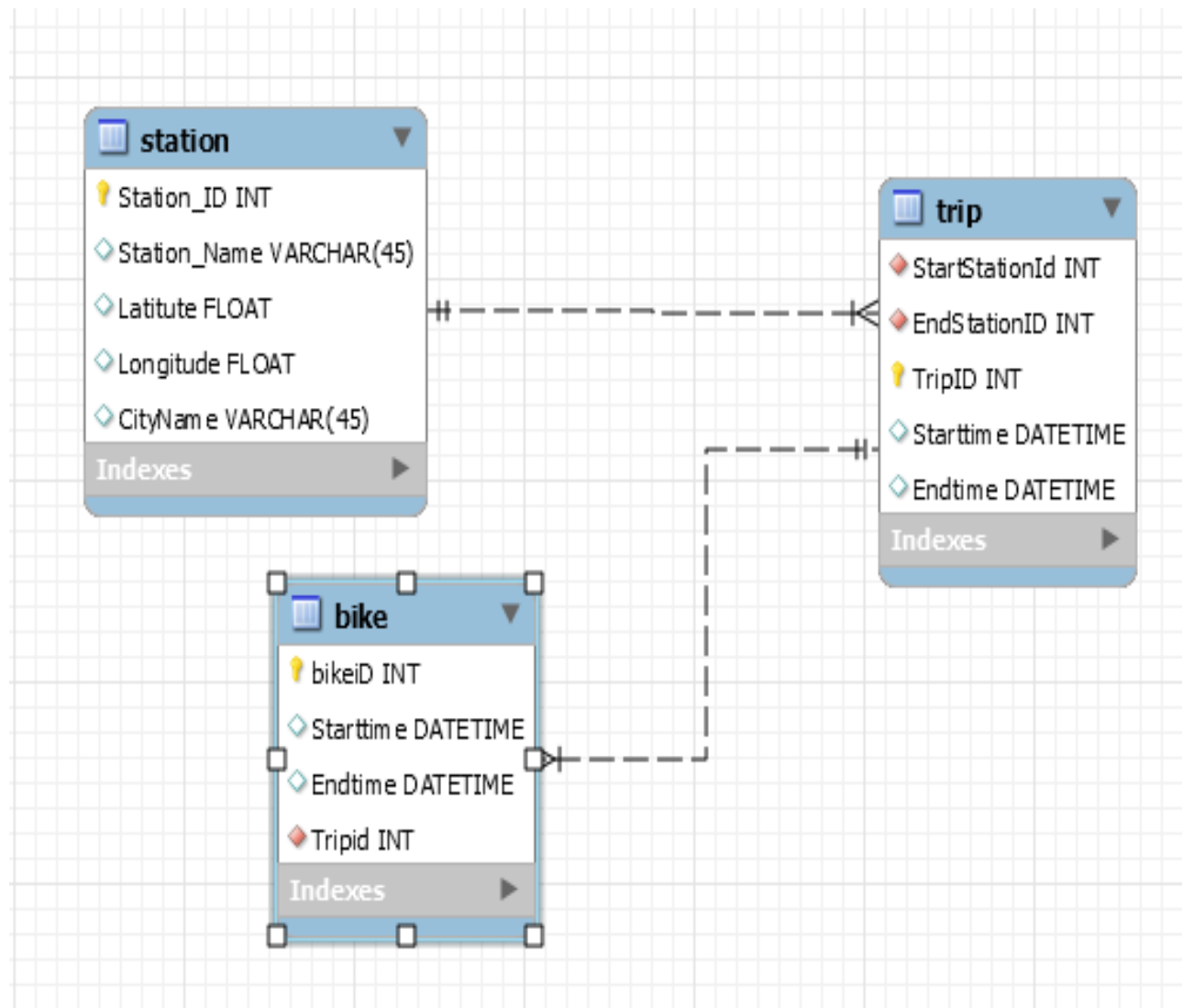
- Data is updated continuously

- Locally stored database to play around and tweak with operations and then upload onto the cloud for deployment
- Visualization tool Tableau that already exists is used

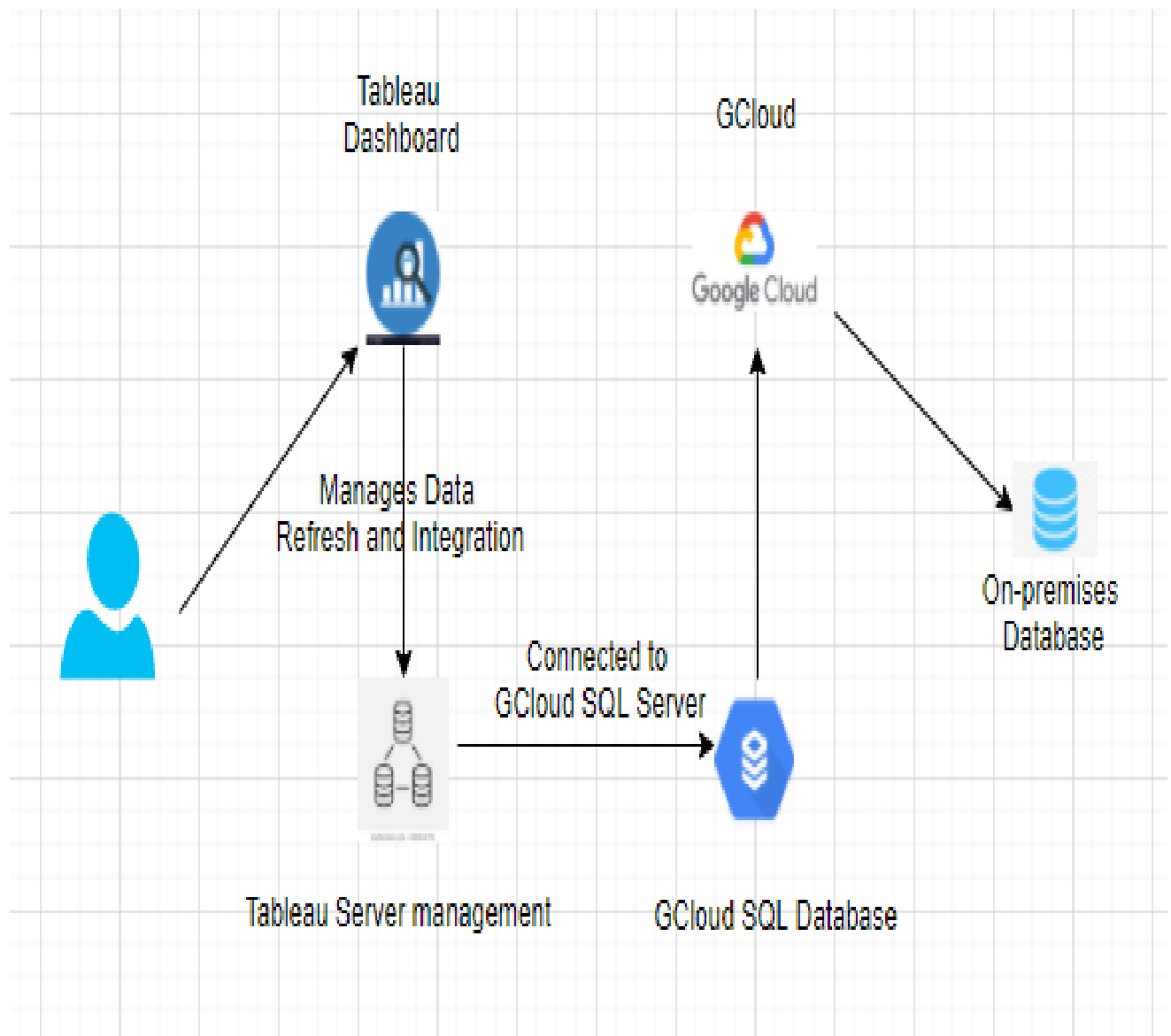
Reasons for picking this proposed solution:

- This solution has a new database design that accommodates data across the US.
- Making sure that data is available to teams for analytics and users for the dashboard.
- Data can be analyzed instantaneously without waiting for data to come from different sources.
- More importantly, the solution is low investment and can accommodate other features of the cloud services like in-memory data for faster analysis, add more CPU and add additional storage in future.

Class Diagram



Deployment Diagram



Discussion of the environment:

An instance is a virtual machine (VM) hosted on Google's infrastructure. MySQL database is running over the Google Cloud SQL instance which is the virtual machine running on the cloud. The underlying MySQL cloud database can be controlled through this virtual machine by passing commands to store, replicate, and manipulate the data. When an instance is created, an unique IP address will be assigned. This IP address will

be used to establish a connection link between the cloud and other external applications that want to access the cloud. Google Cloud SQL enables data to be replicated from Cloud SQL (master) instances to Cloud SQL (read-replica) instances or external (read-replica) MySQL instances. This instance is backed up automatically during the time frame of 11:00 AM — 3:00 PM (UTC-5) everyday by default across multiple regions. To ensure secure connection, specifying how you would like to connect to your database instance is necessary.

Issues related Virtual Machine/Instance:

- Connectivity issues might occur when connected through a private IP address as it requires additional APIs and permissions.
- SSL encryption is recommended when using Public IP to connect to your instance. The server Certificate Authority (CA) certificate is required in SSL connections. Connectivity issues will occur if certificate is expired
- If instance reaches the maximum storage amount allowed, write operation to the database will be failed.
- When Cloud SQL restarts an instance due to maintenance events, Read and write requests from clients using encrypted connections fail and return an error message

Hardware / Software / Cloud usage specifications

- 64 bit Operating system: Windows 7/Windows 8/Windows 10/Windows Server 2008 R2/2012/2016 (Clean build recommended)
- NET Framework 3.5 (for SQL Server Express) and 4.5
- Recommended system specification - Primary Server:
time.
- 2GHz 8 Core Processor or better
- 200+GB Disk space
- 16+GB Memory
- Recommended system specification - Secondary Server(s):
- 2GHz 4 Core Processor or better
- 100GB Disk space
- 8+GB Memory

When any tools are used for cloud migration, that tool should satisfy following software requirements to support cloud migration

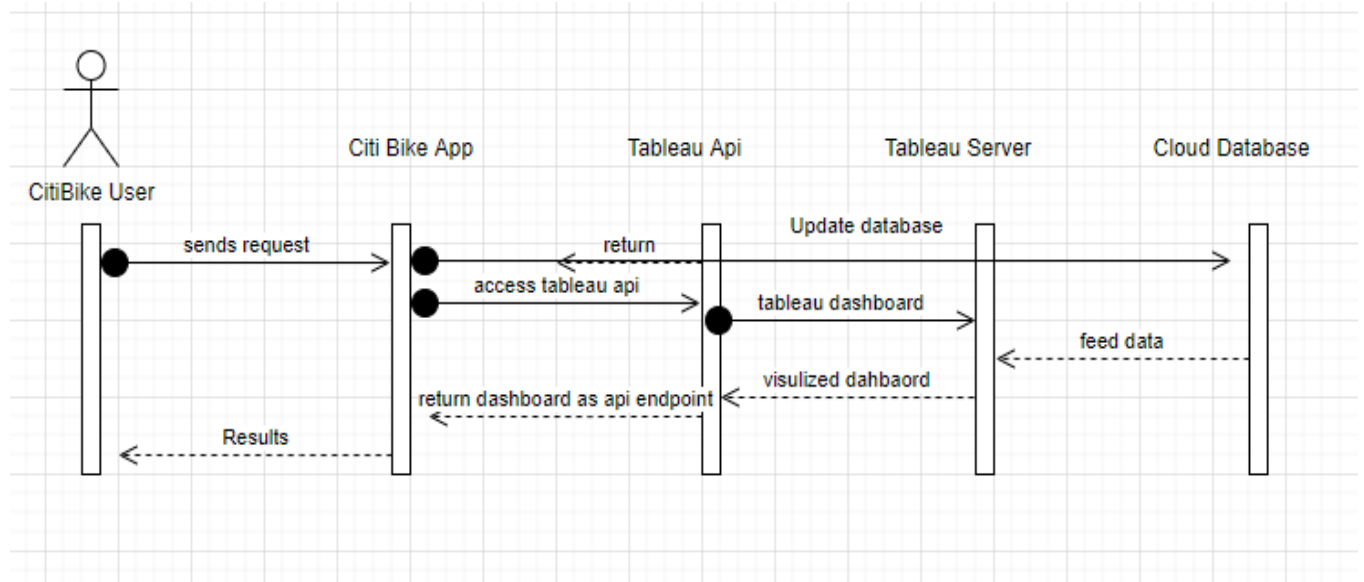
- Microsoft .NET Framework version 4.7.2
- PowerShell 3.0/4.0

If migration managers are used, it should satisfy following requirements

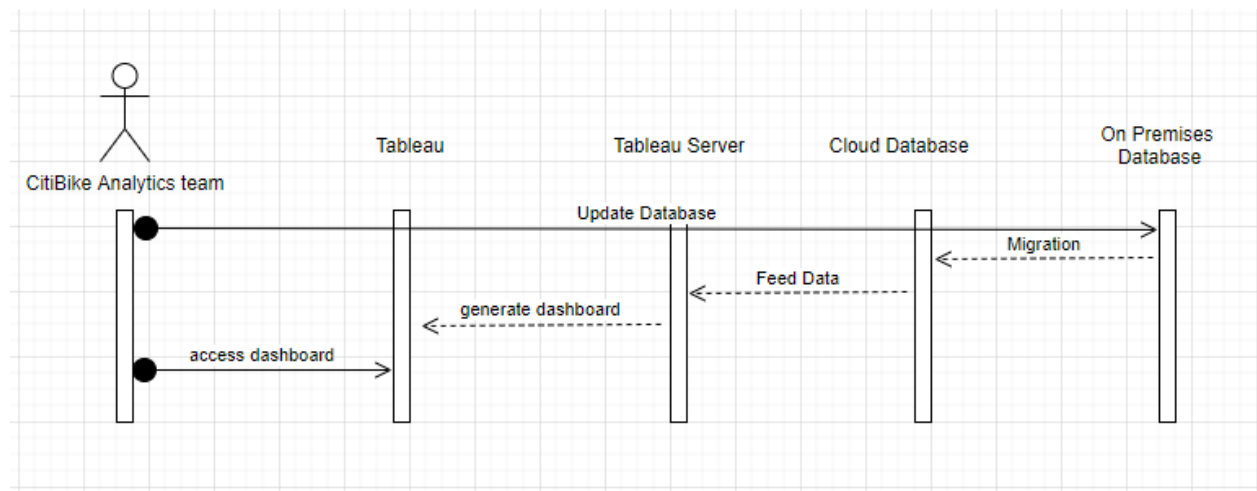
- Connection speed must be at least: 4Mbps Down and 0.6Mps Up with a latency <100ms.
- Network : 100 Base-T Ethernet or Wireless N

Sequence Diagrams

Use case 1



Use case 2



Steps for cloud migration and tableau connection

Migration of CSV data to local db:

Data source sample:

A	B	C	D	E	A	B	C	D	E
Station_ID	Station Name	Latitude	Longitude	City name	StartStationID	EndStationID	StartTime	EndTime	Duration
3212	Christ Hospital	40.73478582	-74.05044364	NYC	3212	3207	10/1/2015 0:16	10/1/2015 0:22	6
3207	Oakland Ave	40.7376037	-74.0524783	NYC	3207	3212	10/1/2015 0:27	10/1/2015 0:39	12
3193	Lincoln Park	40.7246051	-74.07840595	NYC	3193	3193	10/1/2015 0:32	10/1/2015 1:18	45
3199	Newport Pkwy	40.7287448	-74.0321082	NYC	3199	3187	10/1/2015 0:34	10/1/2015 0:39	5
3183	Exchange Place	40.7162469	-74.0334588	NYC	3183	3192	10/1/2015 0:40	10/1/2015 0:49	9
3198	Heights Elevator	40.74871595	-74.0404433	NYC	3198	3215	10/1/2015 0:41	10/1/2015 0:47	6
3206	Hilltop	40.7311689	-74.0575736	NYC	3206	3195	10/1/2015 0:43	10/1/2015 0:46	2
3197	North St	40.752559	-74.044725	NYC					
3213	Van Vorst Park	40.71848892	-74.04772663	NYC					

A	B	C	D
BikeID	StartTime	EndTime	TRIPID
24470	10/1/2015 0:16	10/1/2015 0:22	1
24481	10/1/2015 0:27	10/1/2015 0:39	2
24628	10/1/2015 0:32	10/1/2015 1:18	3
24613	10/1/2015 0:34	10/1/2015 0:39	4
24668	10/1/2015 0:40	10/1/2015 0:49	5
24644	10/1/2015 0:41	10/1/2015 0:47	6

Steps to migrate the above CSV files into local database:

Step 1: Open MySQL workbench and create a local database in it.

Step 2: Import all the CSV files as tables inside the created database.

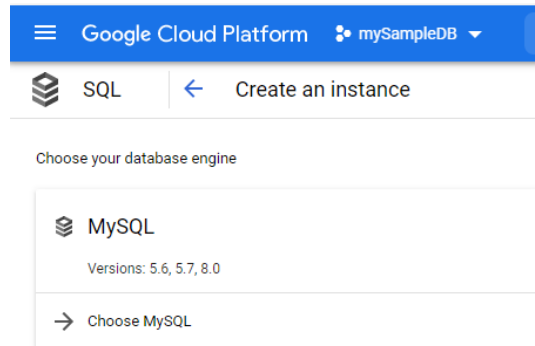
Step3: Choose data type for each column and define necessary constraints like Primark key-Foreign key relationship, Unique, Not null etc.

Now all the csv files are loaded into a local database in the form of multiple tables that are related with each other using primary key and foreign key relationship.

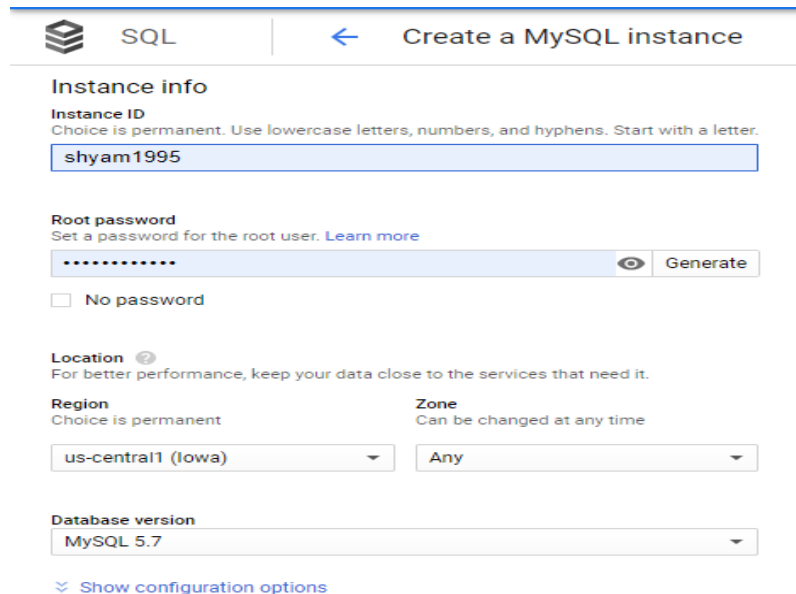
Step 4: Download SQL dump file from local database server so that the sql dump can be used as a source for migrating the existing database in local server into Google cloud database.

Steps to Migrate local database into cloud:

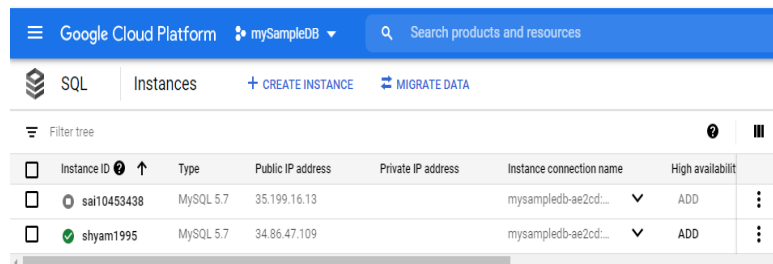
Step1: Go to Google cloud SQL console to create Google cloud SQL instance



Step 2: Configure your MySQL instance to be created.

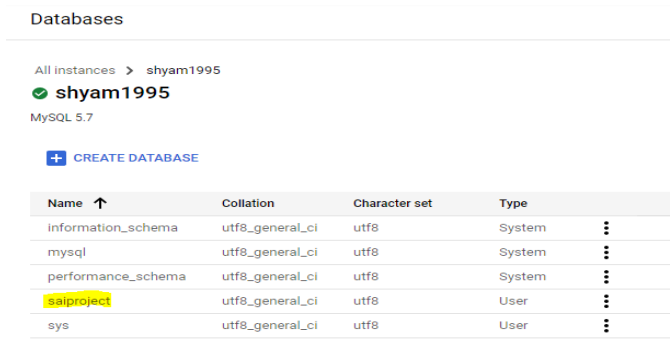


Step 3: Check if the instance got created.



Step 4: Make a note of the public IP address of Google cloud Instance created by you. This IP address will be used later when activating google cloud console.

Step 5: Create a database so that on premise data can be migrated to cloud databases.

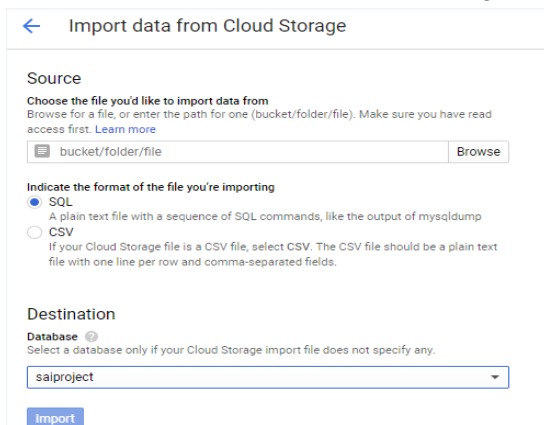


Name	Collation	Character set	Type
information_schema	utf8_general_ci	utf8	System
mysql	utf8_general_ci	utf8	System
performance_schema	utf8_general_ci	utf8	System
saiproject	utf8_general_ci	utf8	User
sys	utf8_general_ci	utf8	User

Step 6: Create a bucket for you and upload the sql dump file inside that bucket.

Step 7: Activate the cloud shell in order to access the created database. Then create a table inside the database so that data from the local can be migrated into this table.

Step 8: Import the sql dump file to migrate the local database into google cloud database



← Import data from Cloud Storage

Source

Choose the file you'd like to import data from
Browse for a file, or enter the path for one (bucket/folder/file). Make sure you have read access first. [Learn more](#)

bucket/folder/file

Indicate the format of the file you're importing

☒ SQL
A plain text file with a sequence of SQL commands, like the output of mysqldump

☐ CSV
If your Cloud Storage file is a CSV file, select CSV. The CSV file should be a plain text file with one line per row and comma-separated fields.

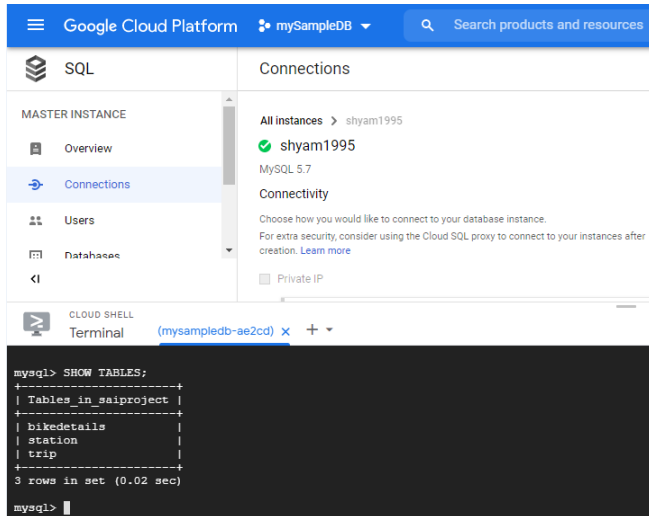
Destination

Database
Select a database only if your Cloud Storage import file does not specify any.

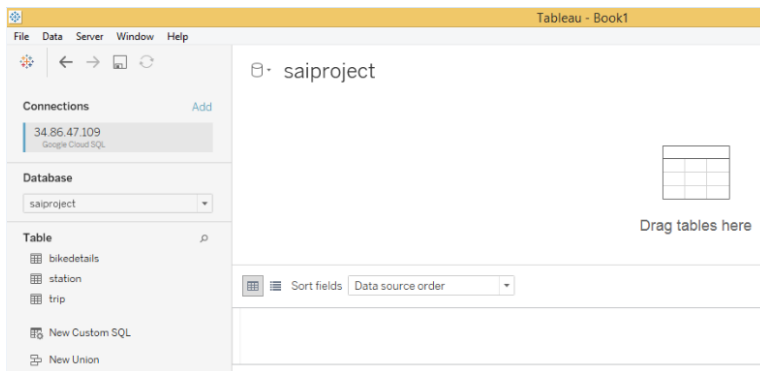
saiproject

Step 9: Database will be imported into Google cloud database from local server.

Step 10: In the cloud shell, access the database and check if data has been migrated into the database.



Step 11: Connect Tableau with the cloud database.



Step 12: Visualizing the data and finding interesting insights about the data to help the organization in decision making.

Extending and Reuse

- Moving into a Multi Cloud environment can be the future scope for Citi Bikes.
- Multi Cloud strategies adopt different cloud storage and cloud management platforms for different applications and departments in the organization.
- The workload on each application is balanced on multiple Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) vendors.
- Multicloud also provides a secure cloud migration approach since multiple vendors provide a multi-layered cloud security solution for each application. system

Screenshots of the system

Sample data in CSV files:

File 1 (Station):

A	B	C	D	E
Station_ID	Station Name	Latitude	Longitude	City name
3212	Christ Hospital	40.73478582	-74.05044364	NYC
3207	Oakland Ave	40.7376037	-74.0524783	NYC
3193	Lincoln Park	40.7246051	-74.07840595	NYC
3199	Newport Pkwy	40.7287448	-74.0321082	NYC
3183	Exchange Place	40.7162469	-74.0334588	NYC
3198	Heights Elevator	40.74871595	-74.0404433	NYC
3206	Hilltop	40.7311689	-74.0575736	NYC
3197	North St	40.752559	-74.044725	NYC
3213	Van Vorst Park	40.71848892	-74.04772663	NYC

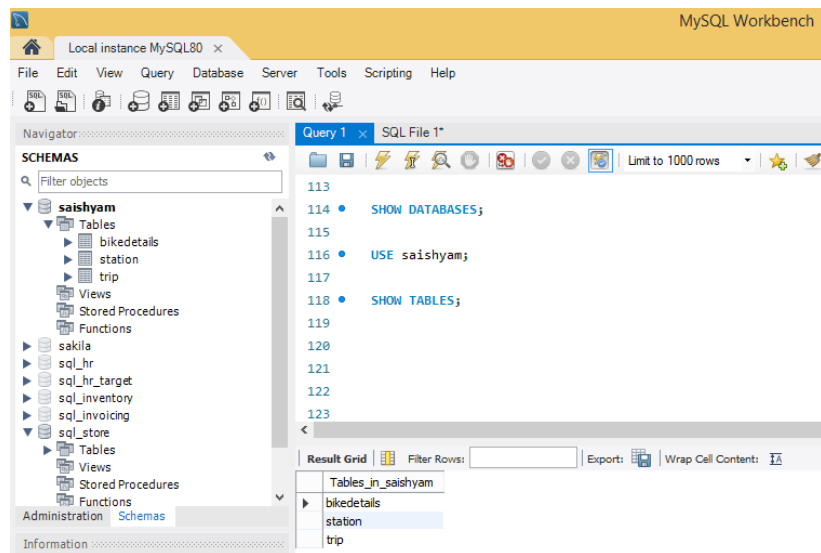
File 2 (Trip):

A	B	C	D	E
StartStationID	EndStationID	StartTime	EndTime	Duration
3212	3207	10/1/2015 0:16	10/1/2015 0:22	6
3207	3212	10/1/2015 0:27	10/1/2015 0:39	12
3193	3193	10/1/2015 0:32	10/1/2015 1:18	45
3199	3187	10/1/2015 0:34	10/1/2015 0:39	5
3183	3192	10/1/2015 0:40	10/1/2015 0:49	9
3198	3215	10/1/2015 0:41	10/1/2015 0:47	6
3206	3195	10/1/2015 0:43	10/1/2015 0:46	2

File 3 (BikeDetails):

A	B	C	D
BikeID	StartTime	EndTime	TRIPID
24470	10/1/2015 0:16	10/1/2015 0:22	1
24481	10/1/2015 0:27	10/1/2015 0:39	2
24628	10/1/2015 0:32	10/1/2015 1:18	3
24613	10/1/2015 0:34	10/1/2015 0:39	4
24668	10/1/2015 0:40	10/1/2015 0:49	5
24644	10/1/2015 0:41	10/1/2015 0:47	6

After Migration of csv files into local on-premise Database



Viewing contents of table after migration in local database:

119

120 • `Select * from station LIMIT 5;`

121

122

123

Result Grid

Filter Rows:

Edit:

Export/Imp

Station_ID	Station Name	Latitude	Longitude	City name
147	Greenwich St & Warren St	40.71542197	40.72533993	LA
152	Warren St & Church St	40.71473993	40.78414472	LA
173	Broadway & W 49 St	40.76068327	40.7311689	LA
224	Spruce St & Nassau St	40.71146364	40.70569254	LA
225	W 14 St & The High Line	40.74195138	40.72759597	NYC

120 • `Select * from trip LIMIT 5;`

121

122

123

Result Grid

Filter Rows:

Edit:

Export/Imp

StartStationID	EndStationID	StartTime	EndTime	Duration	TRIPID
3212	3207	10/1/15 0:16	10/1/15 0:22	6	1
3207	3212	10/1/15 0:27	10/1/15 0:39	12	2
3193	3193	10/1/15 0:32	10/1/15 1:18	45	3
3199	3187	10/1/15 0:34	10/1/15 0:39	5	4
3183	3192	10/1/15 0:40	10/1/15 0:49	9	5

120 • `Select * from BikeDetails LIMIT 5;`

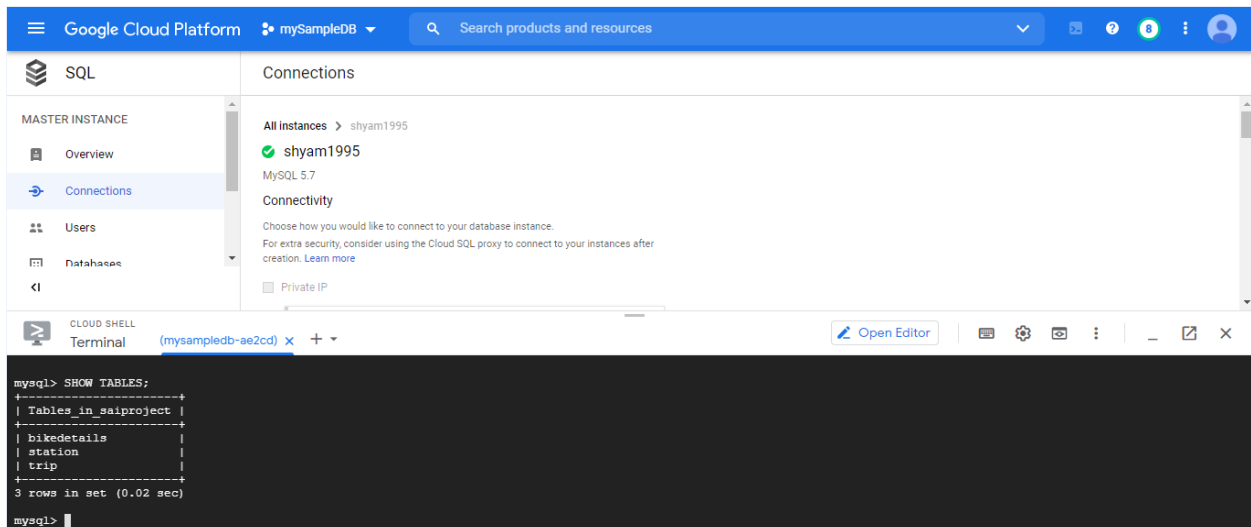
Result Grid

Filter Rows:

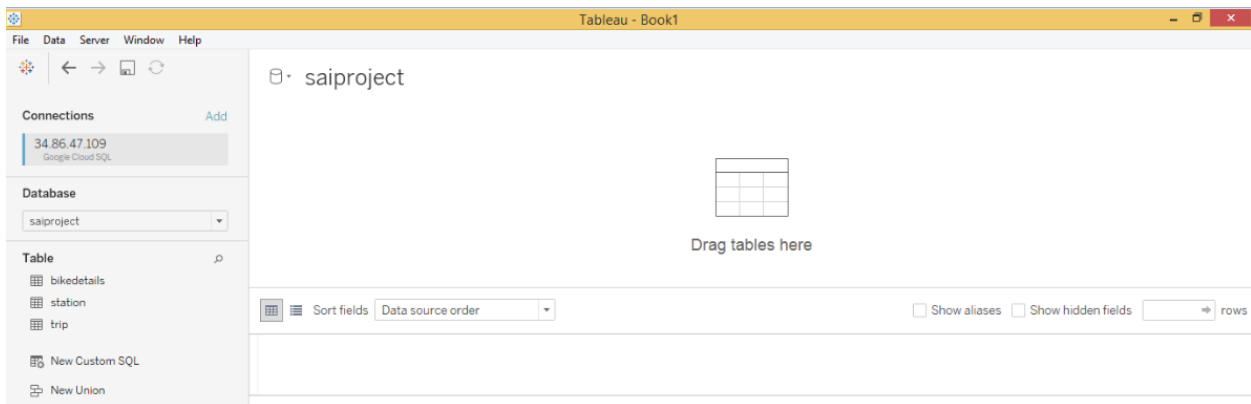
Edit:

BikeID	StartTime	EndTime	TRIPID
14552	10/1/2015 17:31	10/1/2015 17:42	366
14632	10/1/2015 16:13	10/1/2015 16:17	338
14705	10/1/2015 17:39	10/1/2015 17:51	369
14717	10/1/2015 17:27	10/1/2015 17:38	364
14786	10/1/2015 17:48	10/1/2015 17:52	374

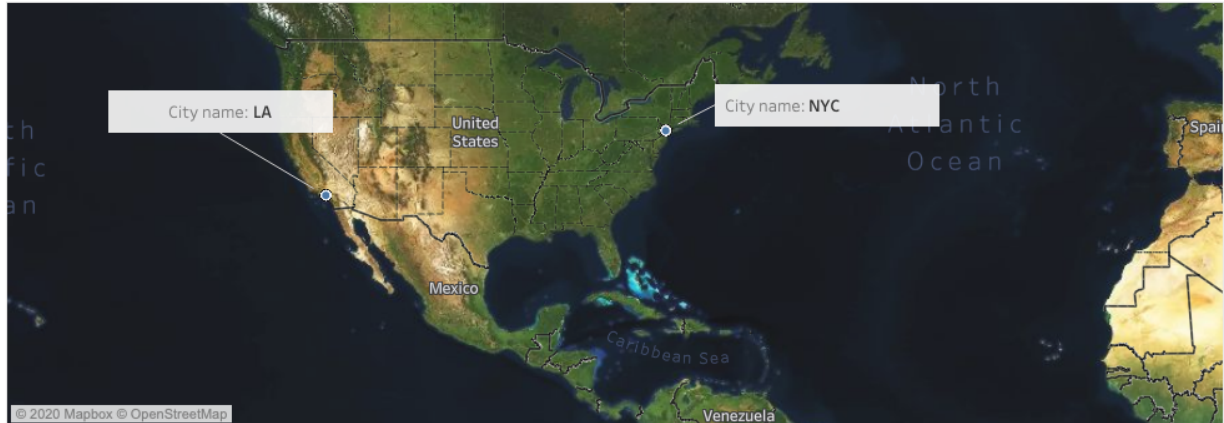
Migrating Local database into Google Cloud SQL database:



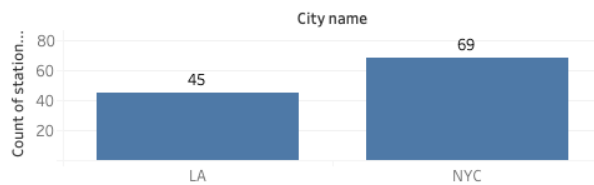
Connecting Tableau to Google Cloud database:



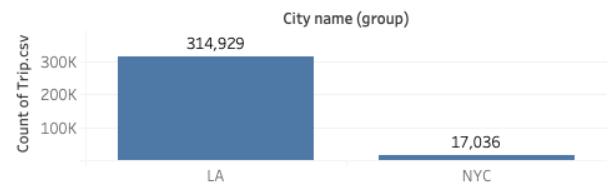
Dashboards from tableau for analysis:



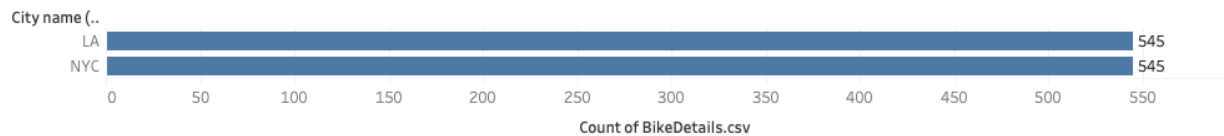
number of stations in each city



number of trips in each city

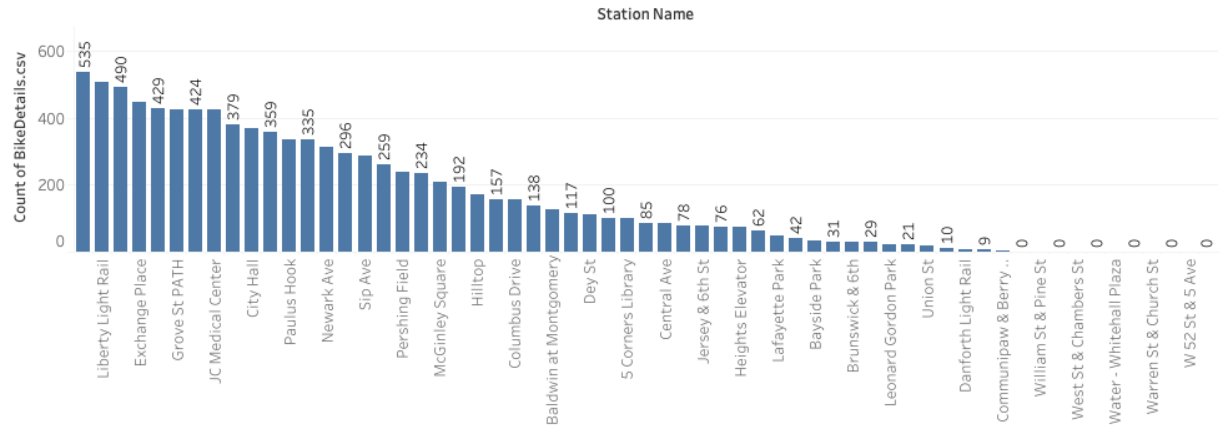


number of bikes in each city

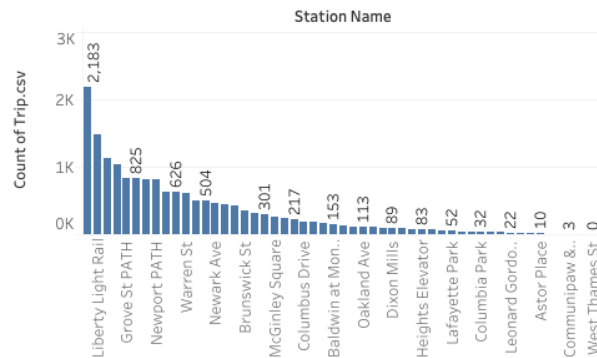


- Dashboard 1 is the overview page.
- Users can observe the number of stations in each city and number trips per city.

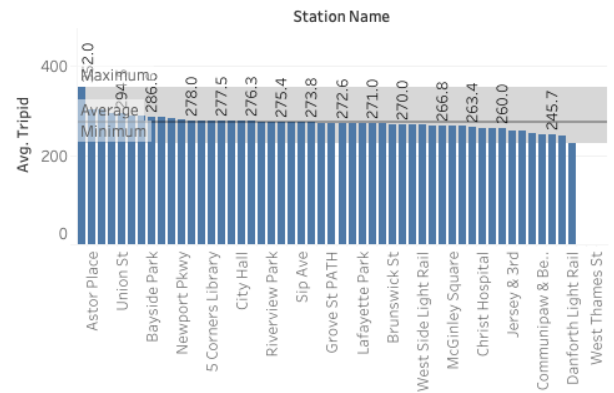
Number of bikes at each station



Number of reported trips per station - availability of bikes at highest trip stations is better for business



Average number of trips per station - Customer behaviour based on avg trip time

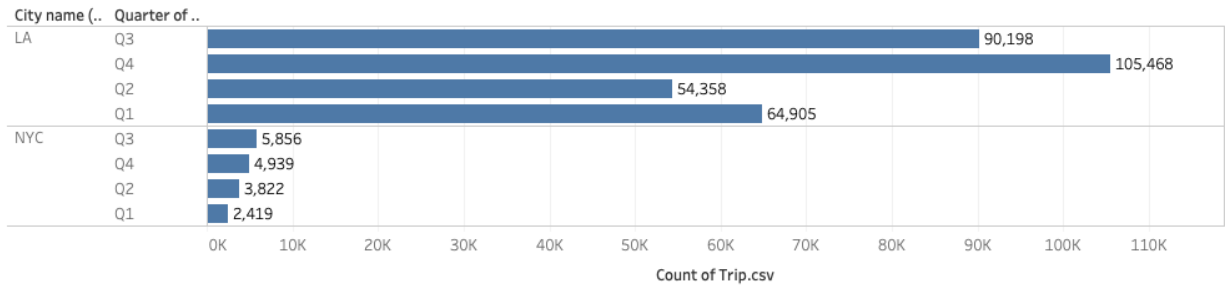


Dashboard 2:

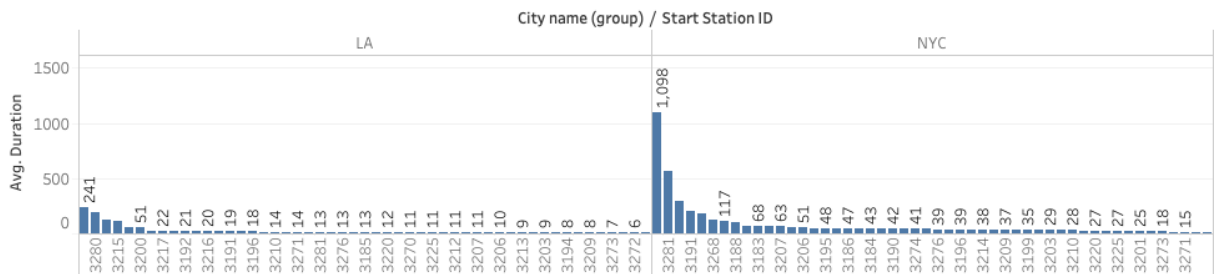
- Analyze number of bikes at each station
- Have more bikes available at the top busy stations

Understand what is the average trip time and identify the customer types. Eg, Recreation, commute

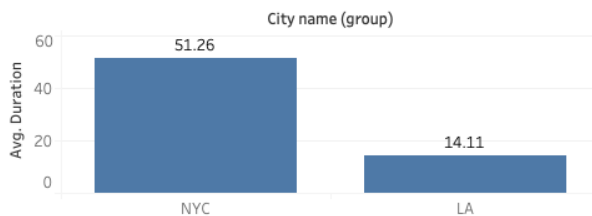
Trips per quarter



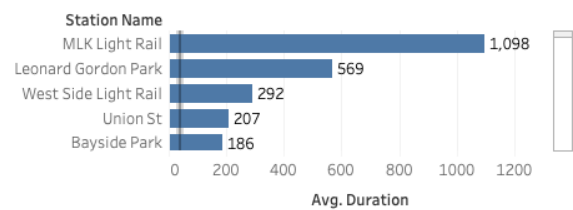
Avg duration per station ID - have more bikes supplied to these spots



Avg duration of a trip



Average duration of trip per station



Dashboard 3:

- Trips per quarter to understand which quarter business is good or bad
- Average duration per station to manipulate customer behaviour
- Average duration of trips per city for overall supply and marketing
- Average duration per station name for convenience.

Conclusion:

Making data available to users is one of the main purposes of our solution. At the same time, however, ensuring the integrity of data is of key importance. The migration of database to cloud that we adopted would reduce cost, improve scalability, and significantly reduce the risk of a cyber incident that could derail Citi Bikes. Moving to the cloud would allow Citi Bikes to seamlessly connect systems together and improve efficiency with all business services. The team would also work productively and securely from anywhere with no downtime.