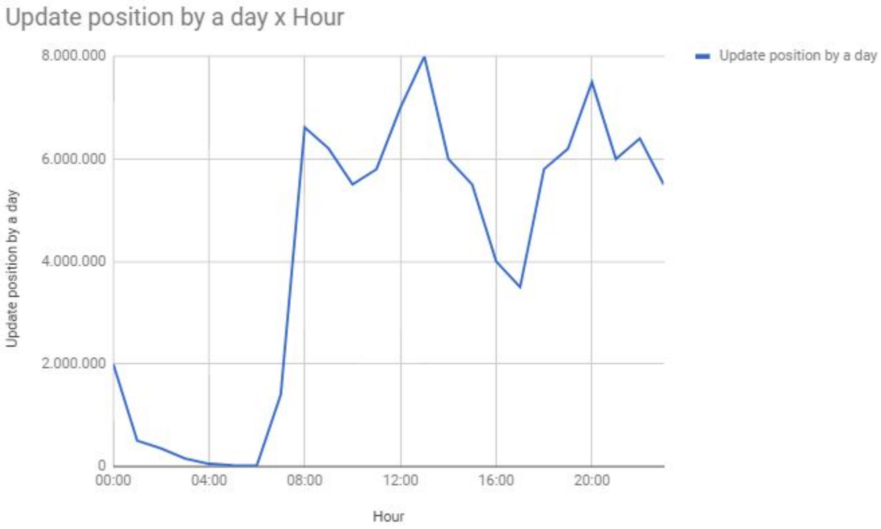
SkipTheDishes - Tech Challenge - Data Engineer

“Design a database solution to handle 100.000.000 Courier’s update/position per day and prepare the solution for a increase of 50% in the the next 6 months.”

My Suggestion:

- Considering that each update/position is 4kb of information;
- 100.000.000 = 381GB~/day = 11,17TB~/month
- Sample Graph Estimation - Update position by a day x Hour graph:



- According graph above, the database must be fast ready to process about 8.000.000 at 13:00 PM (cause I believe that is launch time), 8.000.000 per peak hour = 133.333~ events per minute = 2.222~ events per second ;
- I believe that DynamoDB would be better for this solution because it has multi-model, schemaless, scalable, auto-managed and provisioned read/write capability cloud database @ AWS;
- Considering that will be exponential growth we could:
    - First: Maybe configure auto-scaling (could be expensive, but you can do for a short time just to get the right behavior);
    - Second: in the next 6 months it will be 150.000.000 update/position per day, then, considering that peak is about 8.000.000 per hour and average 2.222 per

second, and it means that writes currently should be able write 2.222 (4kb) events per second, and 6 months next should be able write around 3.333 events per second and resulting 11.998.800 in a peak hour. However, DynamoDB must be able to write 3.333 events per second.

- It will works if you provision 3.333 units to write per second in a table that will modeled to able cover courier's system for the next 6 months.

Database Model:

Table: CouriersPosition

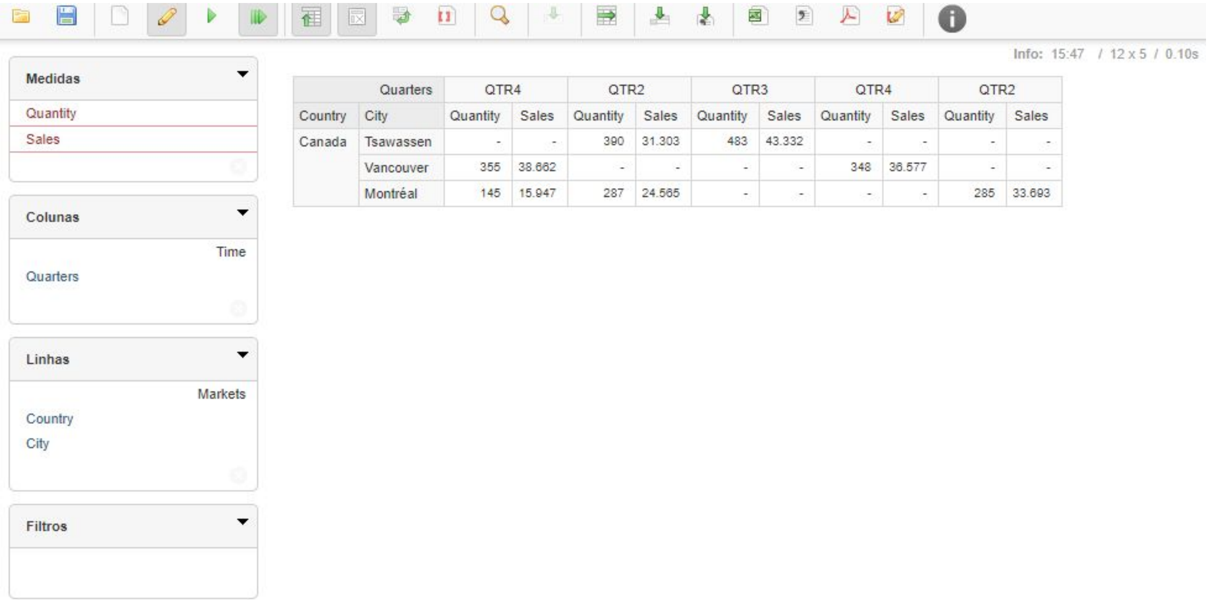
CourierId:Number *(Partition Key)*	PositionTimeStamp:String *(Sort Key)*	Area
1	2018-03-17T12:22:34Z	Eastman
2	2018-03-17T12:25:55Z	Eastman
5	2018-03-17T12:26:13Z	Parkland

"Suggests how to analyze the data to identify areas in the City that the Company could expand the service in the next 6 months."

You can use Redshift to copy DynamoDB's data, and you could perform SQL queries at Redshift like:

```
        SELECT Area, YEAR(PositionTimeStamp) AS EventsYear,
MONTH(PositionTimeStamp) AS EventsMonth, COUNT(*) as
EventsQuantity
                FROM CouriersPosition
                        WHERE PositionTimeStamp >= '2017-09-01'
        GROUP BY Area
        ORDER BY 2
```

The query above show you a just an example. You can granularity your data results to see wich area generates less events, the time is important, cause some areas the behavior could be change. You can use a OLAP engine like Pentaho Mondrian to see your data in multidimensional model (cubes), like images below using to show data with Saiku Analytics - A Pentaho Business Analytics Server’s plugin:



Info: 15:47 / 12 x 5 / 0.10s

Medidas

Quantity

Sales

Colunas

Time

Quarters

Linhas

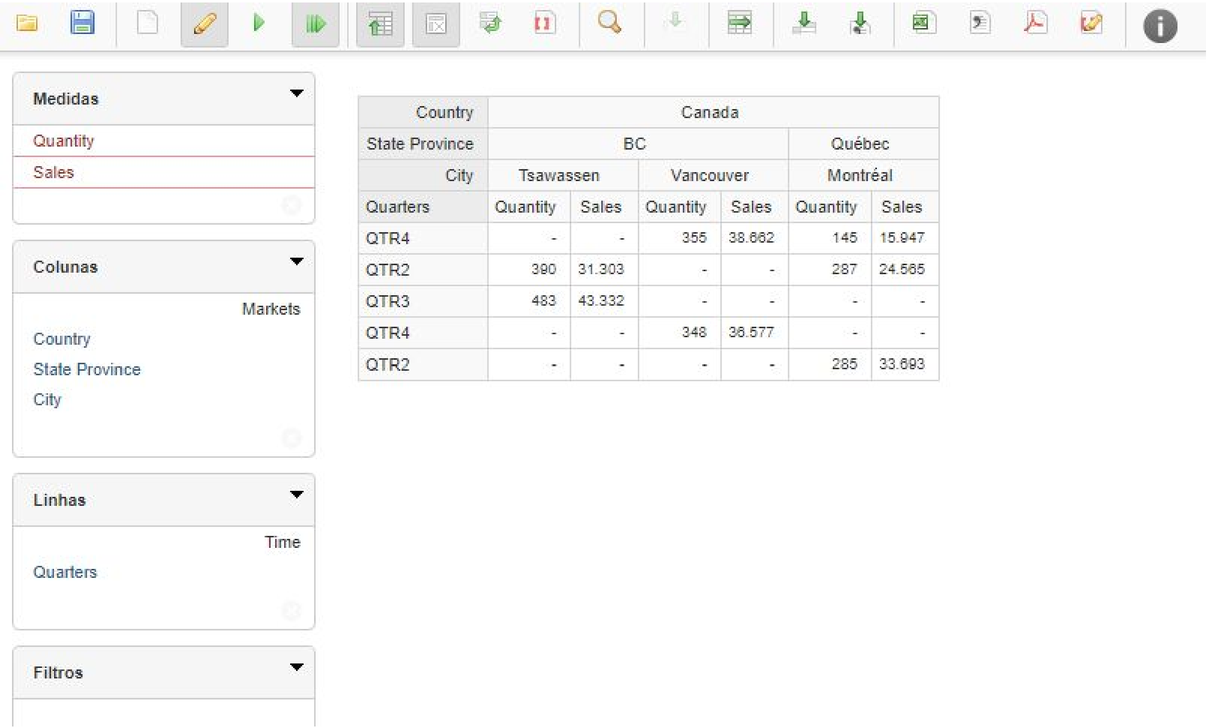
Markets

Country

City

Filtros

	Quarters	QTR4		QTR2		QTR3		QTR4		QTR2	
Country	City	Quantity	Sales	Quantity	Sales	Quantity	Sales	Quantity	Sales	Quantity	Sales
Canada	Tsawassen	-	-	390	31.303	483	43.332	-	-	-	-
	Vancouver	355	38.662	-	-	-	-	348	36.577	-	-
	Montréal	145	15.947	287	24.565	-	-	-	-	285	33.693



Medidas

Quantity

Sales

Colunas

Markets

Country

State Province

City

Linhas

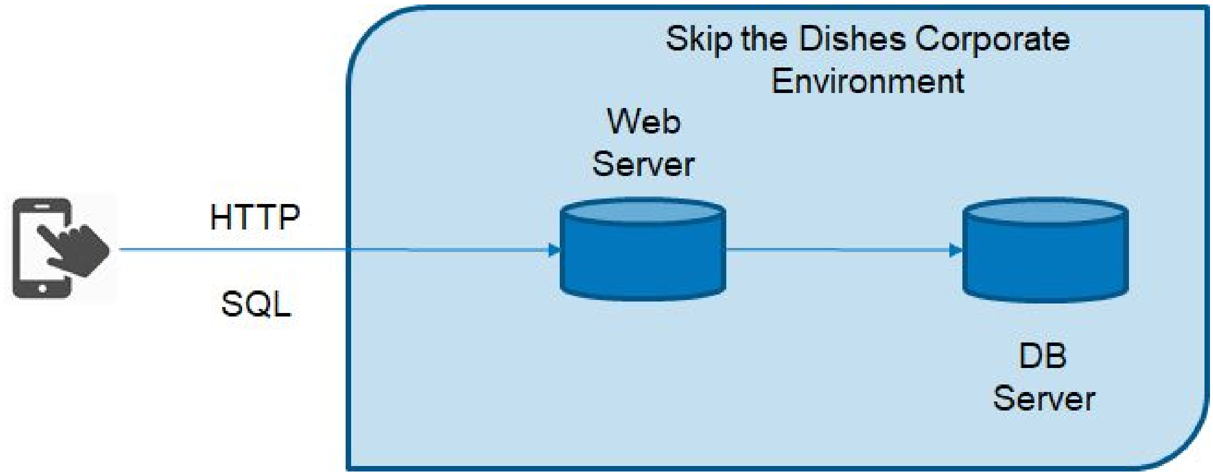
Time

Quarters

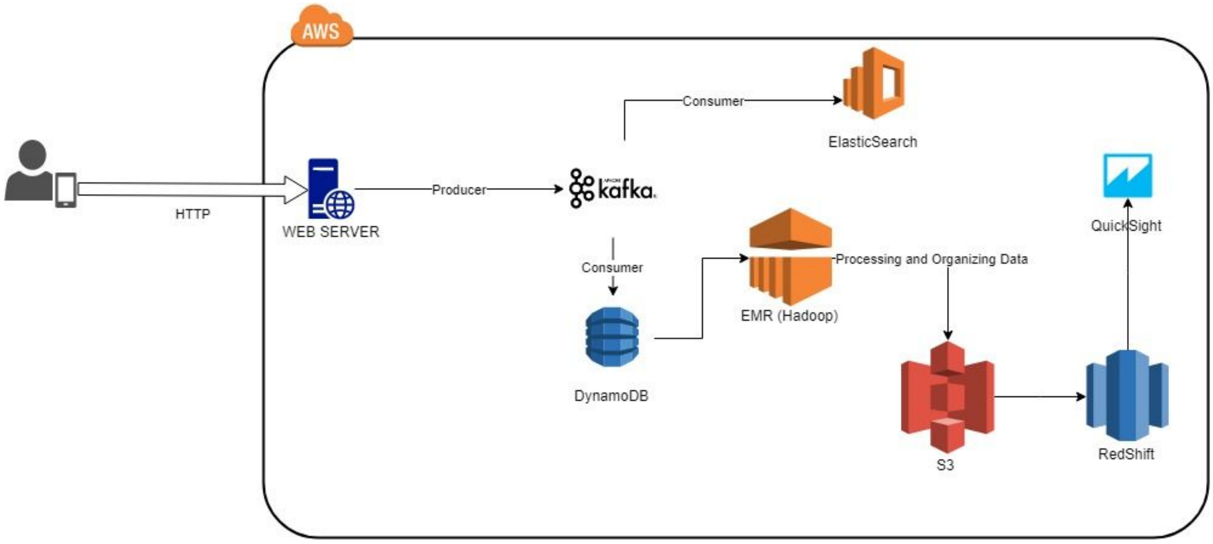
Filtros

Country	Canada					
State Province	BC				Québec	
City	Tsawassen		Vancouver		Montréal	
Quarters	Quantity	Sales	Quantity	Sales	Quantity	Sales
QTR4	-	-	355	38.662	145	15.947
QTR2	390	31.303	-	-	287	24.565
QTR3	483	43.332	-	-	-	-
QTR4	-	-	348	36.577	-	-
QTR2	-	-	-	-	285	33.693

"Consider Skip is moving their environment to the Cloud. Change the architecture diagram using the following technologies: Kafka, Hadoop, Elasticsearch, MQTT/AMQP, columnar databases."



My Suggestion



Kafka to stream events. You should keep the last 24 hours of events for a lot of consumers that wants to make near real-time decision or insert data into relational databases or DynamoDB.

DynamoDB to persist data.

Hadoop (EMR) to process distributed events in large scale, organizing data and record them into S3.

Redshift (Columnar Database) will get S3's organized data to datawarehouse purposes (S3DistCP).

QuickSight for analytics, reports and dashboards.

Elasticsearch to keep events to be able to full text search events (I believe that will work for segmentation issues).