Business Performance Dashboard

Overview:

The project involved developing a dashboard to track the performance and behavior trends of approx. 2000+ stores. The client is an experienced automotive care company that runs franchise operations around the globe. The project enabled the client to compare various metrics and forecast trends for the upcoming periods.

Client details:

Name: Confidential | **Industry:** Automotive | **Location:** USA

Technologies:

Python, Django, PostgreSQL, Celery, Redis, AWS

Project Description:

The client had multiple franchises that made it difficult for them to handle the merged data from sales, revenue, leads, and employees. As the process became cumbersome, it led to an increase in ETA for generating reports and metrics. After analyzing the client's need, team@Mindfire offered to develop an interactive web interface to help provide meaningful insights to business owners. Some salient features of the application:

- Customizable interface for the client and business owners with different hierarchies to view and edit the data. The web interface was implemented via Python and Django Web framework for dynamic webpage generation and extendibility.

- An integrated database to transfer real-time data from distributed business locations to a centralized repository. Since the physical stores were geographically distributed, it was difficult to access the systems for error checking and maintenance. Hence, a central logging system, AWS Cloud Watch was used to view logs without requiring a remote connection.

- The client can import various data from the centralized database like call logs and appointment requests to measure conversion rates.

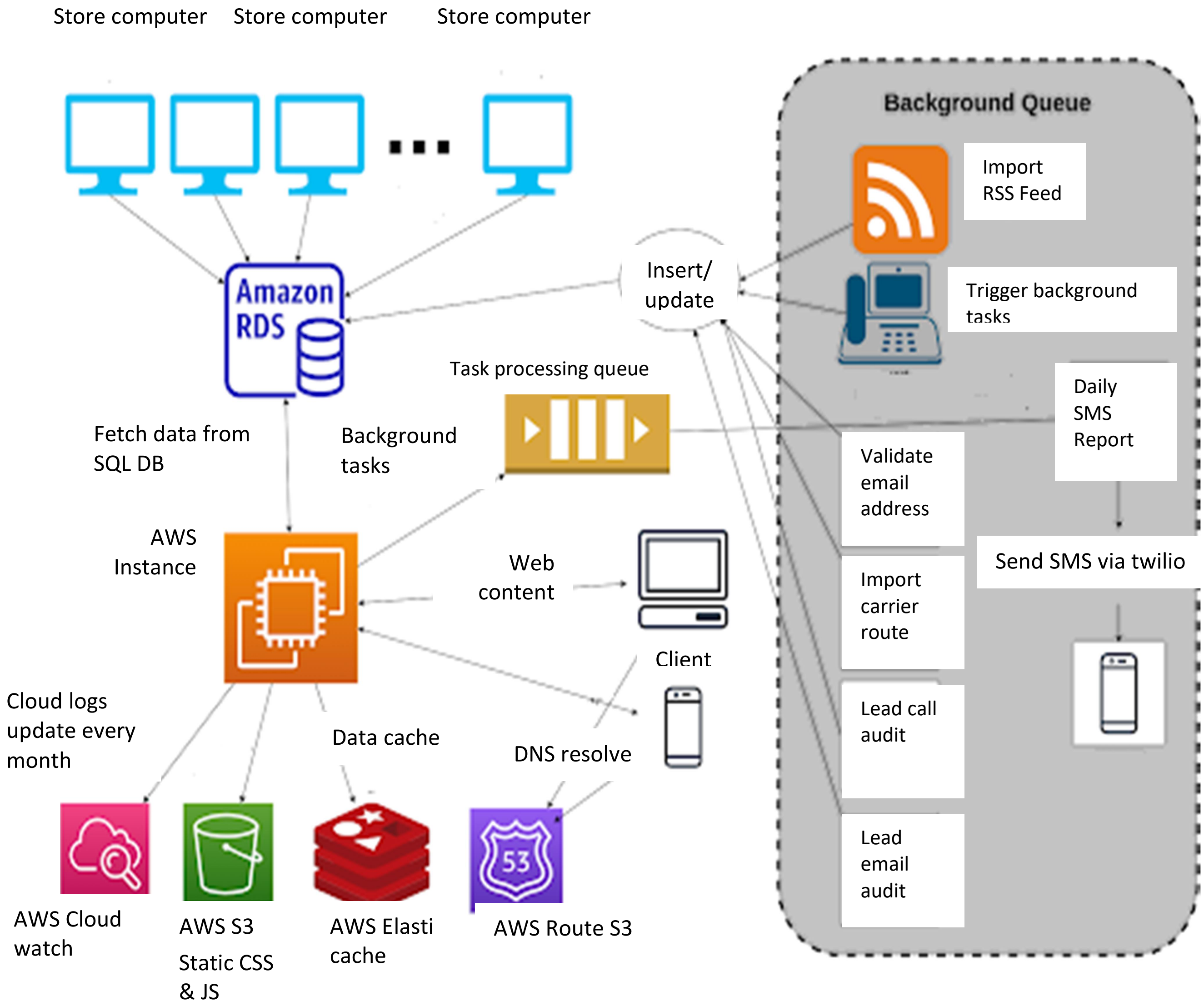
- The client can view real-time employee and customer data. They can also analyze call metrics to verify the effectiveness of calls for lead generation.

- The reports dashboard can generate insights based on rules shared by the client to track metrics and forecast trends. The platform also allowed business owners to view historic data and compare the performance of different aspects of the business.

- As there were multiple queries to be executed, this led to a decrease in efficiency. The team created indexes on key columns to optimize the queries for large date ranges.

- The team ensured that data of one metric would not affect another. It helped to maintain consistency in results across historical time frames.

Architecture:


Store computer
Store computer
Store computer
Amazon RDS
Fetch data from SQL DB
AWS Instance
Background tasks
Task processing queue
Insert/ update
Web content
Client
Cloud logs update every month
Data cache
DNS resolve
AWS Cloud watch
AWS S3
Static CSS & JS
AWS Elasti cache
AWS Route S3
Background Queue
Import RSS Feed
Trigger background tasks
Daily SMS Report
Validate email address
Send SMS via twilio
Import carrier route
Lead call audit
Lead email audit