Serverless Architecture

Case Study

Improving scalability & availability using serverless architecture

A healthcare technology startup reduces infrastructural costs for their platform by using next generation architecture

Executive Summary

Next generation challenges require next generation solutions. A healthcare technology startup wanted to create a highly scalable data gathering mechanism which could leverage the power of data science and machine learning to create predictive insights. Great Software Laboratory created a robust serverless architecture which not only reduced the resources required to develop and maintain the product, but it also gave our client the benefit of having a highly modular and flexible product which conformed to regional regulatory and data protection mechanisms with minimal effort.

Overview

Our customer is a healthcare technology startup who has created a healthcare app that leverages machine learning models to diagnose health issues. The product engineering of this solution had two aspects.

1. The data related to health parameters had to be collected globally. We developed an app to collect this data. The features of the data and meta data were to be extracted and the ML model would use it for training. This activity continues to happen on a large scale across the globe. Other apps from the ecosystem would also need to plug into the platform to access the data.

2. The final product showed the result to the user. This app with built in ML models, would need to respond really fast under all scenarios. Customers request health checks randomly all around the world and there are no fixed patterns to identify this demand.

Challenge

Due to the complexity and unique requirements of the product, the backend architecture had multiple facets.

1. Modern architecture
The architecture had to be modular and service based for the entire pipeline which included data gathering, processing, feature engineering, and the prediction engine. The product and services had to be accessible across the globe at any point of time.

2. Quick scalability
The surge in traffic is unpredictable. Keeping infrastructure optimized for the peak would be mean heavy infrastructure costs. However, quick results were equally important for the user.

3. Continuous availability
The tools and services had to remain up and running at all times. This would require a robust architecture which could handle loads continuously.

4. Integration

Individual components of the solution were to be created using multiple best suited languages and tools. These disparate tools had to work together in confluence. We knew right from beginning that we would need an interdisciplinary team with expertise in multiple languages.

5. Data protection

The application would handle and process a lot of personal health data. It was crucial to adhere to local, regional and global regulatory & law enforcement frameworks to maintain & protect the data and privacy of individuals.

6. Costs

While catering to all these needs, like every startup, it was very important not to waste resources in endless experimentation. We had to develop the optimal solution with minimal investment.

Solution

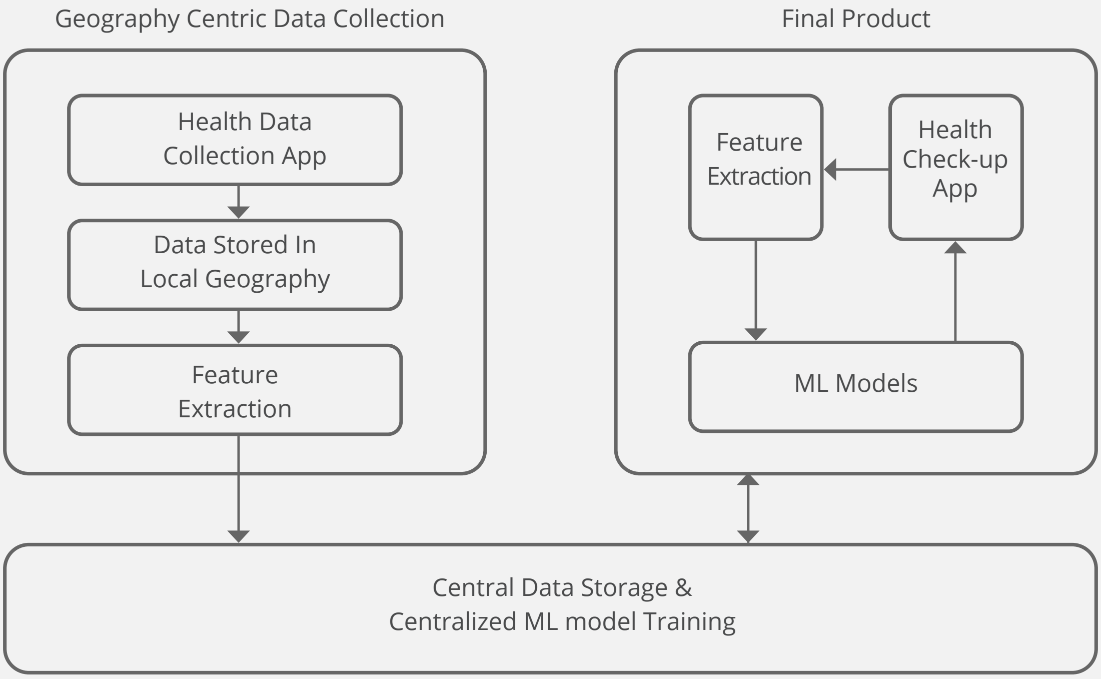
Considering these complex requirements, GS Lab decided to use AWS serverless components, microservices and SAM (serverless application architecture), for the product. Even though this architecture had been around for a couple of years, not many had used it in real products.

We used the following AWS components:

- The lambda function for almost all components (Lambda versioning is powerful when we have ML models as lambda).
- Kinesis service for feature extraction of the health data.
- AWS cloud formation templates
- S3 for secure storage
- RDS PostgreSQL for database needs.

***Serverless computing** is an execution model where the cloud provider runs and manages allocation of servers and resources by itself. In an AWS environment, it uses the Lambda function. It is completely dynamic. The resources are called upon as needed and one does not have to pre-purchase or allocate resources. The pricing model is based on actual usage making it a ‘pay per use’ model. Serverless computing simplifies scaling, capacity planning and maintenance while reducing the burden on ops.*

GS Lab decoupled ML model training & ML infused product architectures. This not only saved costs but also complied with data related government regulations.


Geography Centric Data Collection
Health Data Collection App
Data Stored In Local Geography
Feature Extraction
Final Product
Feature Extraction
Health Check-up App
ML Models
Central Data Storage & Centralized ML model Training

Solution architecture

Challenges faced by early adopters

- *Every new technology has a learning curve when it comes to finding the optimal approach. The development team needs to invest time to understand a new technology when the broader community has not adopted it en masse.*
- *We also faced challenges with language specific object/data initialization. A lot of experimentation and deep troubleshooting was required to solve issues such as these were not encountered by any previous adopter.*
- *We also factored in the technical limitations of the Lambda service when it comes to deploying large artifacts during the design phase.*

Impact

Reduced Costs

Improved Scalability

Architecture Extended To Platform

1. Traditional ops would have needed four DevOps engineers. The newer structure needs only one. This indicates how ops-light our solution is.

2. Serverless architecture resulted in approximately 40% lesser infrastructure costs with medium resource utilization. Higher savings can be achieved for higher loads.

3. We created a completely scalable product which not only had high availability, but also generated quick results for the user.

4. Our serverless architecture proved to be so efficient that we decided to make it extensible and implement it across the platform which allowed other apps/tools to plug in to the system.

5. Our solution made it very easy for the client to adhere to all regional and global data protection & other regulatory frameworks. Compliance to all legal requirements needed minimal effort due to the modular nature of the architecture.

6. The application used best-in-class tools and languages to develop individual components. Serverless architecture made it possible to integrate these disparate systems to work together seamlessly.

7. We also realized other savings around DevOps as patches or upgrades don’t need to e applied to EC2 machines.

Great Software Laboratory (GS Lab) has been the technology partner of choice to 100+ organizations across North America, Europe and Asia-Pacific for over 16 years. Leveraging our expertise in 130+ tools & technologies, we have created 300+ 'first-of-its-kind' solutions to real-world problems. Our 'Beyond code' philosophy ensures that we not only push boundaries of existing technologies but also try out newer problem solving approaches to keep our customers one step ahead of their competitors. Our global team of 1200+ employees is adept at creating 'real value' at each stage of the customer growth journey, right from proof-of-concepts to completely scaled up products. For more information about our solutions & offerings, please visit www.gslab.com

gslab™

www.gslab.com