

Developing a plasma donor application using Function-as-a-service in AWS

Aishwarya R Gowri

Jain University, Department of MCA, computer science

ABSTRACT

A plasma is a liquid portion of the blood, over 55% of human blood is plasma. Plasma is used to treat various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish antibodies that fights the infection. In this project plasma donor application is being developed by using AWS services. The services used are AWS Lambda, API gateway, DynamoDB, AWS Elastic Compute Cloud with the help of these AWS services, it eliminates the need of configuring the servers and reduces the infrastructural costs associated with it and helps to achieve serverless computing. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates were high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors.

Keywords: serverless, aws, plasma theory, covid19, dynamoDB, cloud

1. INTRODUCTION

Cloud computing helps in on-demand deliver of IT resources over the internet with pay-as-you-go pricing model where users have to pay only for the resource that they use. This helps to reduce the additional infrastructural cost and users can access technology services such as power, storage, compute, database, networking, analytics and also intelligence over the internet in order to offer flexible, innovation, and economies of scale. Users can run their infrastructure more efficiently and scale their business according to their requirement. Cloud deployment modules such as public cloud, private cloud, hybrid cloud and community cloud helps the users to choose the type of deployment options that are beneficial for their company. Cloud service models consists of software as a service (saas), platform as a service (paas) and infrastructure as a service (iaas).

In Software as a service a third party service providers will host the applications and make them available over the internet. Some requires purchasing of licenced version with involves huge cost and with the help of software as a service those applications can also be used without having to buy the licence of the software which is more cost effective. with the help of platform-as-a-service customers can run, develop and manage the applications without any complexity of building and maintaining the infrastructure which is associated with developing and launching the applications. Infrastructure as a service allows the enterprise to rent or lease the servers for compute and storage in cloud. Microsoft Azure.

Amazon Web Services (AWS), Microsoft azure, Google Cloud, IBM Cloud, Oracle, Salesforce, SAP are some of the cloud service providers. Amazon web services (AWS) is one of the leading cloud service providers. They offer mixture of infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS). Aws was launched in 2006 and it is one of the first company to introduce pay-as-you-go cloud computing model. Plasma is that the clear, straw-coloured liquid part of blood this is still once crimson blood cells, white blood cells, platelets and alternative cellular elements rectangular measure removed. it's the only largest element of human blood, comprising concerning fifty-five p.c, and carries water, salts, enzymes, antibodies and alternative proteins. A plasma donor has to pass health screening prior to each donation. Plasma is collected through a process known as plasmapheresis. During this process an automated device is used to separate the plasma from the blood. Once the plasma is collected red blood cells and other components will be returned to the donor. After plasma is collected it is tested for suitability for future manufacture, frozen and then held for 60 days prior to pooling.

A donor has to register to the website providing his details such as name, contact information (phone number and email id) along with donor's blood group and donor's plasma count. In this project the services used are AWS Lambda which will allow the users to run the code without managing or provisioning the servers, AWS API gateway is a fully managed service which makes it easy for a developer to create, publish monitor, secure, maintain APIs at any scale. It handles all the tasks which is involved in accepting and processing hundreds of Concurrent API calls along with traffic management, authentication, authorization and API version management. DynamoDB is a multi-master database used for storing the data, Amazon SNS is a messaging service from system-to-system and app-to-person communication.

1.1 Function as a Service (FaaS)

Function as a Service is a cloud computing model based on the serverless technologies and architectures. It provides a platform to the developers in order to run, deploy, and also manage the application functionalities without any complexity of building and maintaining the infrastructure that is associated with deploying the application. Applications developed using this model helps to achieve a serverless architecture.

2. LITERATURE REVIEW

[1]In this paper, the author has carried out analysis based on the opportunities presented by serverless computing. They emphasise that serverless services are more affordable approach for many network services and it is more user friendly as serverless approach will relieve the customers from the intricacies of deployment. These services will help to improve the new business opportunities. [2]Author conducted a survey of existing serverless platform in this paper from source projects, industry, academia, use cases, and key characteristics and has described the challenges and the open problems associated with it. Authors work presented a hands on experience of serverless technologies using different services from different cloud providers such as Amazon, Google, IBM, Microsoft Azure. [3]In this paper three demonstrators for IBM Bluemix OpenWhisk was presented. They exhibit even-based programming triggered by weather forecast data, speech utterances and Apple WatchOS2 application data. And also demonstrated a chatbot using IBM Bluemix OpenWhisk that calls on the IBM Watson services which include dates, weather, alarm services, news and music tutor.

[4]In this paper serverlessOS was designed. It comprises of components such as 1. desegregation model that leverages desegregation for abstraction but it will enable resources to move fluidly between servers for the performance. 2. The second key component is cloud orchestration layer which helps to manage fine-grained resource placement and allocation throughout the application lifetime with the help of global and local decision making 3. And the third component is an isolation capability which enforces data and resource isolation. [5]In this paper an efficient resource management system for serverless computing framework was proposed which aims to enhance resource with a focus on memory allocation among the containers and the design which was added on top of an open-source serverless platform, openLambda and it is based on allocation workloads and serverless functions memory needs events are triggered.

3. PROBLEM STATEMENT

During COVID 19 crisis the requirement for plasma increased drastically as there were no vaccinations found in order to treat the infected patients. In such situation it was very difficult to find the plasma donor, check whether the donor was infected previously and was recovered, and which donor is eligible to donate plasma was a challenging task. As the plasma therapy was one of the ways to treat the infected patients getting the donor details played a major role.

4. PROPOSED MODEL

The proposed method helps the users to check the availability of donors. A donor has to register to the website providing their details. The registered users can get the information about the donor count of each blood group. The database will have all the details such as name, email, phone number, infected status. Whenever a user requests for a particular blood group then the concerned blood group donors will receive the notification regarding the requirement. A Json code is written to store the information, to fetch the requested information in lambda.

5. ARCHITECTURE

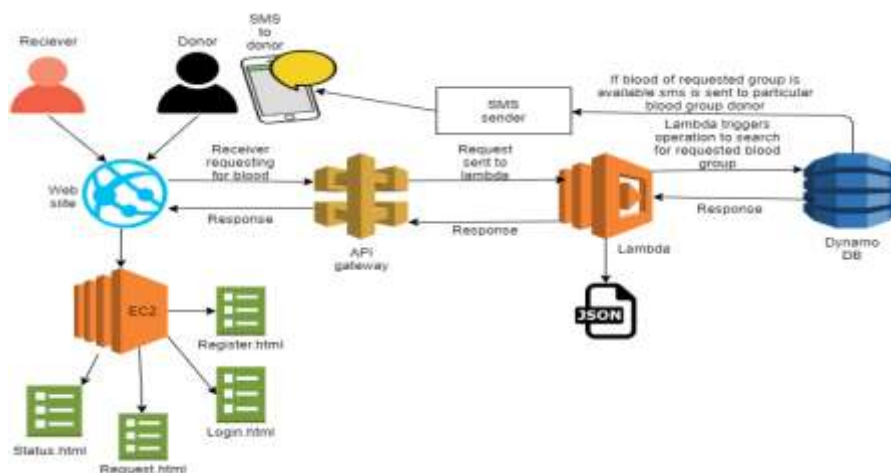


Figure 1: Blueprint of the proposed model.

A user will register by providing all his details such as user name, email id, phone number, location, password, infected status, and blood group. once the user has successfully registered to the web application they can view the availability of blood group and the count. once a user register to the web application an API request will be called which invokes one of the functions with which the user's details can be stored in dynamoDB. When API invokes the lambda function where the program will be written to store the details of the lambda to store the details of the registered users in dynamoDB.

When a user request for a particular blood group an API will invoke the lambda function and the lambda function will trigger operation and fetches the information of a particular blood group donor from the dynamodb and it will then fetch it back to the API and this API will display the information in the user interface. when a user requests for a particular blood group a request sms will be sent to the particular blood group donor.

6. FLOW OF MECHANISM

User will first register to the web application then will login to the application. If a user can successfully login, then he can check the availability of the blood group and can request for a particular blood group. once the user requests for a blood group then the concerned blood group donor will receive a request sms.

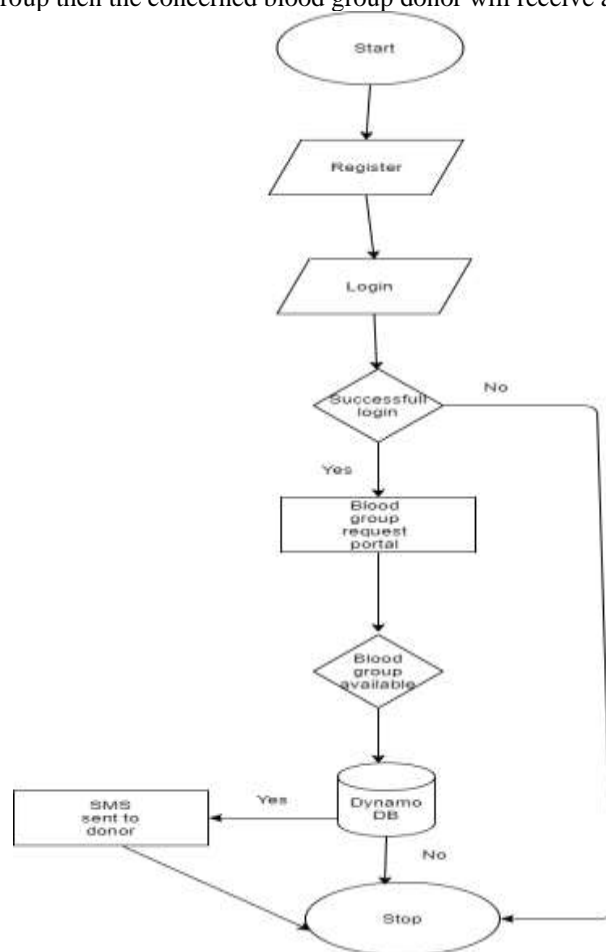


Figure 2: Data flow diagram

7. RESULT ANALYSIS



Figure 3: Registration page



Figure 4: successfully registered page

id	name	age	sex	blood_group	location
1	John Doe	35	Male	B+	New York
2	Jane Smith	28	Female	A-	Los Angeles
3	Mike Johnson	42	Male	O+	Chicago
4	Sarah Brown	31	Female	B-	San Francisco
5	David Wilson	25	Male	A+	Seattle

Figure 5: Dynamodb database table



Figure 6: Request message received by the donor

8. CONCLUSION

The efficient way of finding plasma donor for the infected people is implemented using the plasma donor website that is hosted on Aws platform. To ensure the smooth functioning of the website operations. I have hosted the website in aws platform to make sure the operations are running successfully Aws lambda function is used and to deploy the application AWS EC2 service is used.

9. FUTURE ENHANCEMENTS

Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures that many plasma donors can be added into the community.

Using elastic load balancer, it helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible downtime.

10. REFERENCES

- [1] R. C. Gojko Adzic, "Serverless computing: Economic and architectural impact," *ESEC/FSE*, 2017.
- [2] P. C. P. C. a. V. I. M. Yan, "Building a chatbot with serverless computing," *IBM watson research center*, 2016.
- [3] S. E. a. B. J. J. Short, "Cloud Event Programming Paradigms: Applications and Analysis," *9th IEEE International Conference on Cloud Computing (CLOUD)*, pp. pp. 400-406, 2017.
- [4] Z. Al-Ali, "Making Serverless Computing More Serverless," *IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. pp. 456-459, 2018., 2018.
- [5] A. S. a. S. Jindal, "EMARS: Efficient Management and Allocation of Resources in Serverless," *IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. pp. 827-830, 2018.