CLOUD BASED PLASMA DONOR APPLICATION

Submitted by

A. KALYANI

Assistant Professor

Department of Computer Applications

PSG COLLEGE OF TECHNOLOGY

COIMBATORE 641004

Email : akk.mca@psgtech.ac.in

CONTENTS

CLOUD BASED PLASMA DONOR APPLICATION

1. Introduction

1.1 Brief Description of the Project

Any new crisis needs new measures and practices. During covid-19 the world faced the crisis for blood donors to safeguard the infected people. Finding out the exact donor and reaching them was a big challenge during this period. Here comes a simple cloud based solution to connect the donors and recipients.

1.2 Purpose

The main objective of this project is to connect the recipient to the exact donors. An online application like this is very much useful during covid-19 like pandemic period. This cloud based application makes it usable anywhere , anytime , using any device. The SMS message is sent to the donor on request of a particular blood group .

The major objective of doing it as a cloud application is to make it cost effective with less investment on infrastructure.

2. Literature Survey

2.1 Existing problem

During any ailment blood is the major need. Covid-19 taught the entire world the importance of blood. Even though there are blood banks run by individuals and by hospitals , reaching the needy person on time is a big problem. There are many blood donor web applications which are not scalable and they cannot withstand storing voluminous data of donors list.

2.2 Proposed solution

My proposed solution is to build a cloud based blood donor application. It uses the cloud approach by deploying the application in the cloud making use of the cloud database viz., IBM DB2

3. Project overview

3.1 Project Flow

The prototype of the proposed system is given in Fig. 1

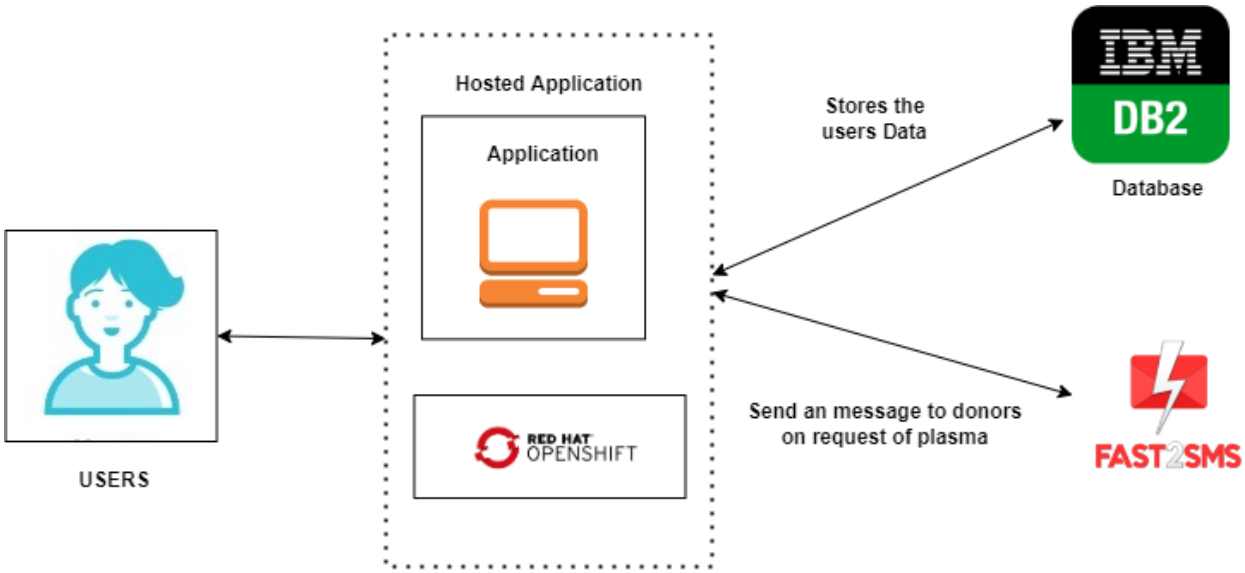


Fig 1. Proposed Plasma Donor Application

Users need to register an account and login to the application. Once the user logins, he will have a dashboard to view the total number of donors and count of people with specific blood groups. Users will have the option to request the blood. Once the user requests, all the people with that blood group will be notified with an SMS.

3.2 Hardware and Software Configuration

The configuration of hardware and software used for designing this project :

Hardware Specification: Windows 10 PC or above

Software Specification:

Coding language used - Python 3.7

IDE - Visual studio Code 2022

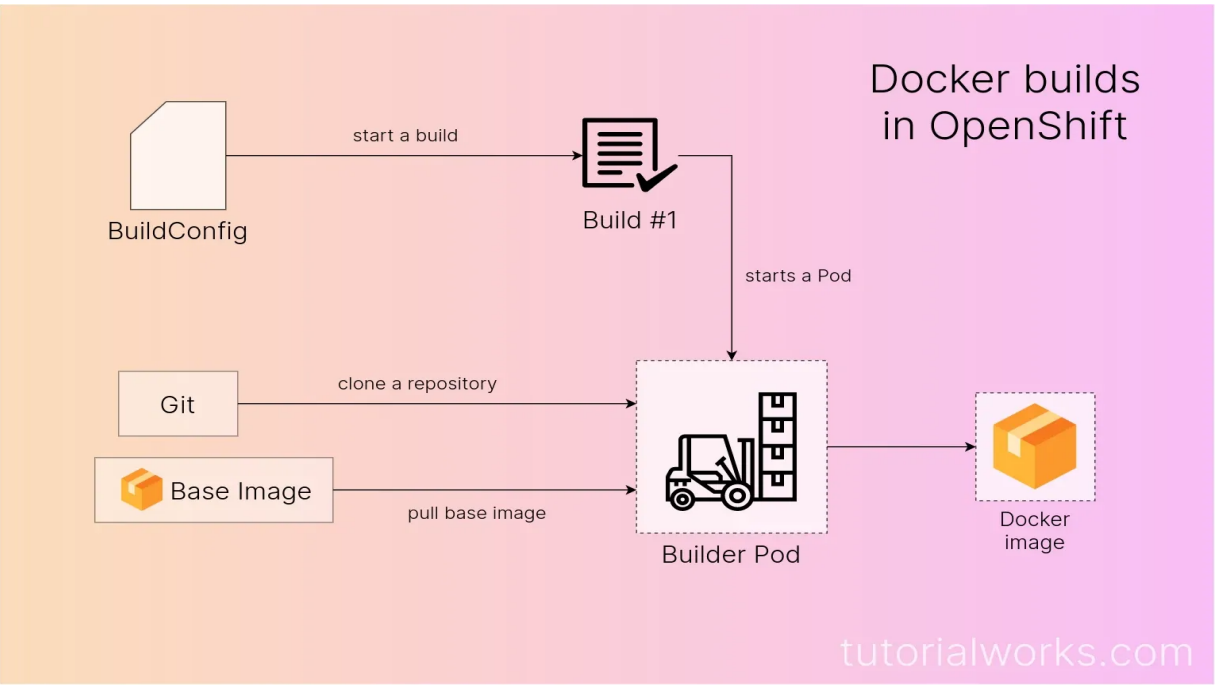
Public repository to upload code - Personal Github account

To access cloud database - IBM cloud account

Deployment environment - Redhat openshift account to work in sandbox

4. Implementation

4.1 Building and Testing the Application



The application was cloned from the given public repository to the local machine. The necessary credentials were modified as per the service credentials in my IBM cloud account . Then the code was tested by executing in the local machine. The following issues were found and solved.

Issues met:

- Could not access IBM DB2 through our college LAN. The issue was solved by using my phone as a hotspot for the college PC.
- Database creation error. As directed by my mentor, I solved it by creating a new schema.

- While executing the python ‘app.py’ , “ requests not found “ error was encountered. It

was solved by installing the requests package using the command “pip install requests” . Then the statement “import requests” was added in the fourth line of the ‘app.py’ file.

4.3 Tools and Technology used:

- Red Hat OpenShift Container :

Red Hat OpenShift is a cloud-based Kubernetes platform that helps developers build applications. It delivers a single, consistent Kubernetes platform anywhere that Red Hat Enterprise Linux runs. The platform ships with a user-friendly console to view and manage all your clusters

OpenShift gives organizations the ability to build, deploy, and scale applications faster both on-premises and in the cloud. It also protects your development infrastructure at scale with enterprise-grade security.

Being a Paas , it could be used with any other web services too.

OpenShift is a great platform to use for building and shipping cloud-native applications.

It also supports the developers by making the development and testing workflow of the applications much easier by ensuring that the developers do not have to worry about switching between the physical and the virtual servers whenever required. Thus, it helps to increase the productivity and efficiency of the existing application workflow with reduced maintenance costs.

- **Docker**

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

It is used for building and seamlessly integrating legacy projects enabling organizations to achieve high-velocity innovations. It encourages the concept of DevOps methodology through CI/CD (Continuous Integration/Continuous Development). Thus, the developers can integrate their code into a shared repository as early as possible and deploy it quickly and efficiently. Thus, the local development setup behaves like a live server. It comes up with integrated developer tools. Also, the virtual machine image is openly accessible and shareable. The applications developed on it can be reused and are shareable. It is open-source and available on Github.

It reduces the setup cost on the part of the customers and increases the efficiency and the productivity of the existing application workflow as both are open-source technologies. This also ensures the scalability of the existing application workflow. Also, as both of the technologies form an integral part of the cloud platform, they can be used independently.

Steps followed for Build config :

Basic OpenShift Docker BuildConfig example (from Git)

The application is built by importing Dockerfile from a Git repository along with other source files.

Fig 2 Import from Git

It allows one to manage everything easily by keeping all the build scripts in the same repository as your code. Using Git repository gives the advantage of version control of the docker files too.

5. Result

The final output of the application built is shown as screenshots in figures 5 to

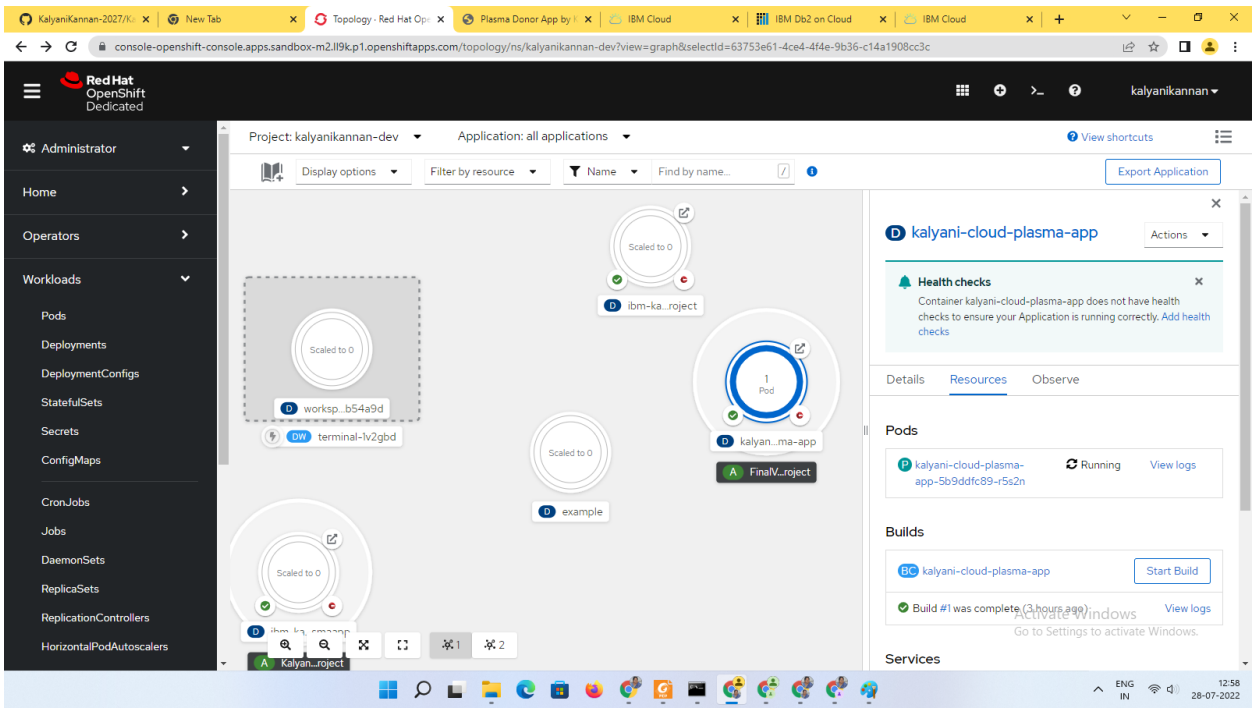


Fig 3 Redhat Openshift Console topology screen

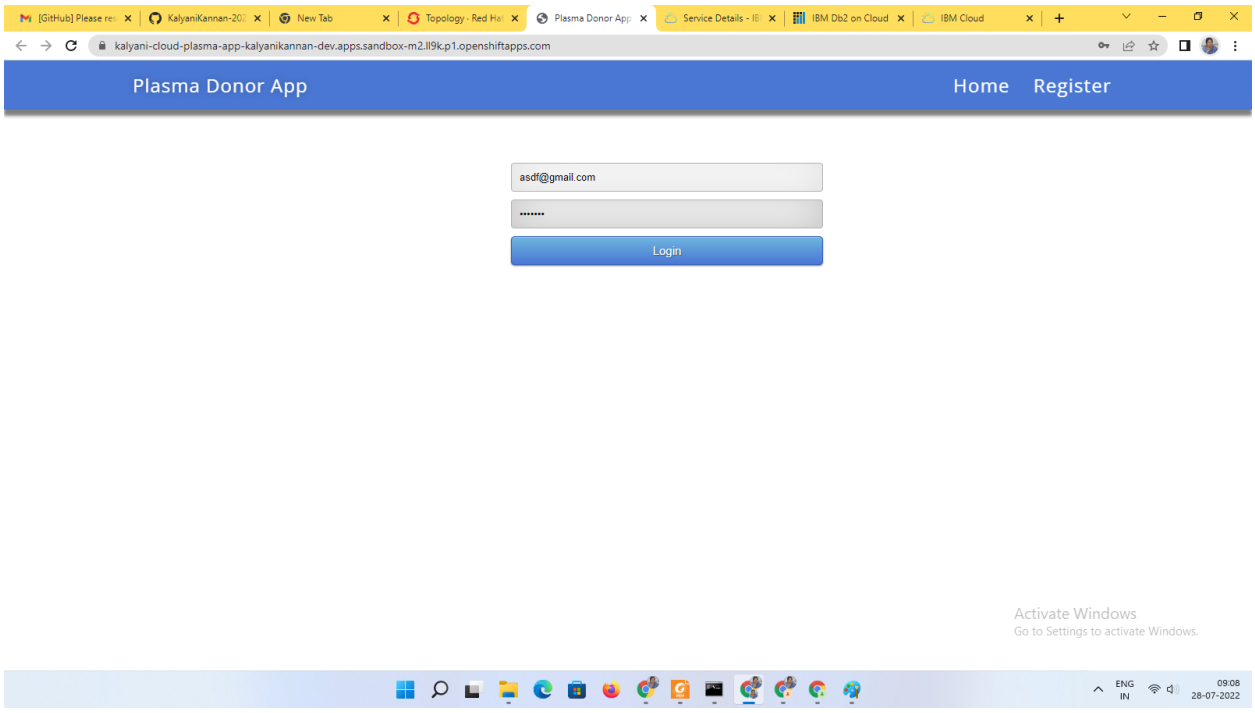


Fig 4 Home page of the Plasma Donor App

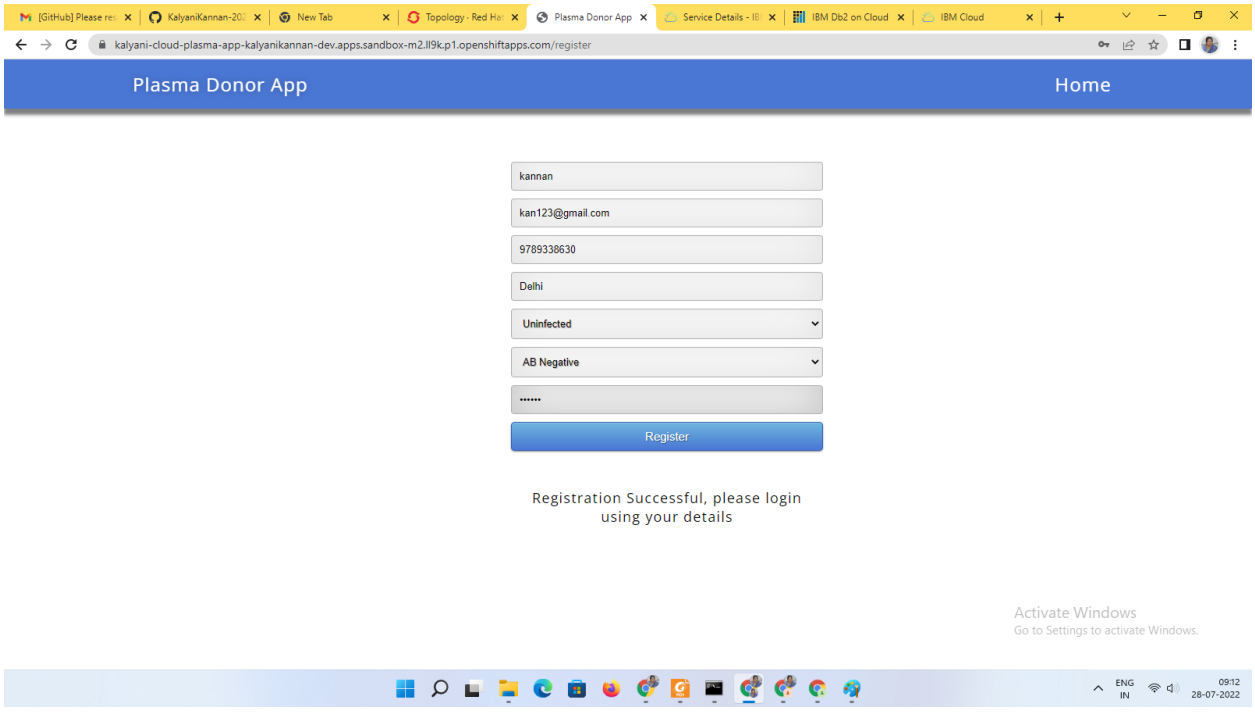


Fig 5 Registration page of the Plasma Donor App

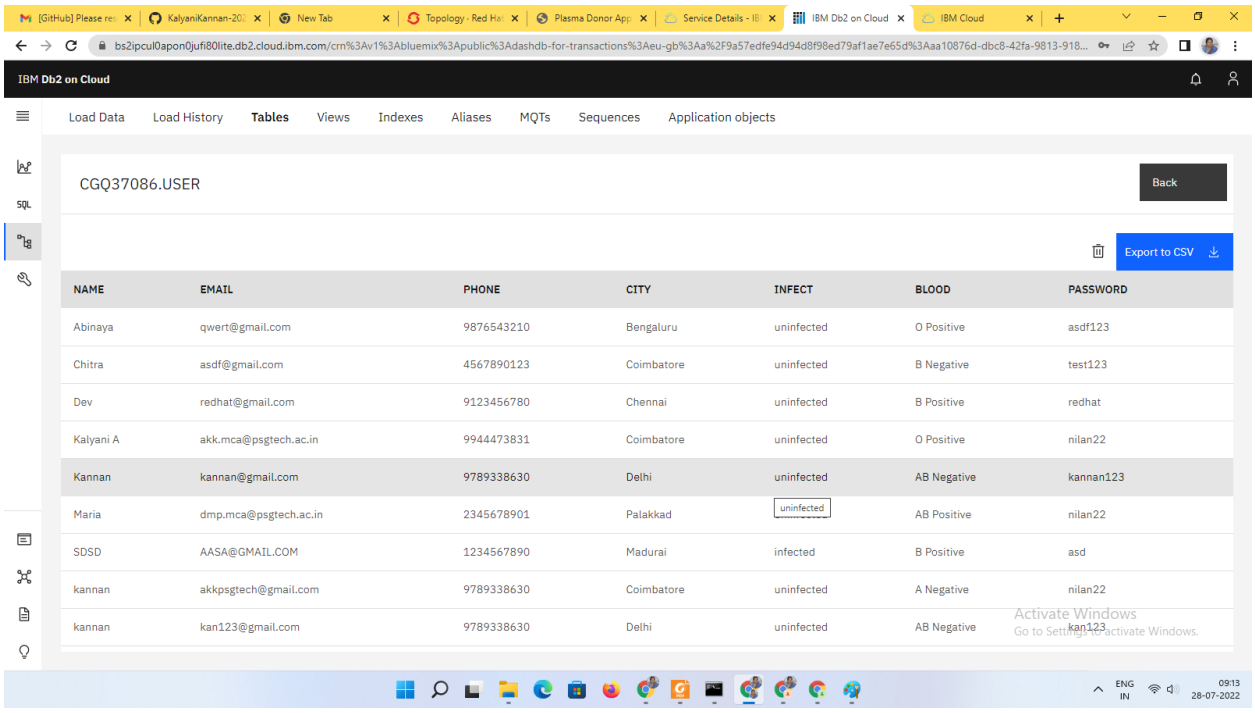


Fig 6 Database screen

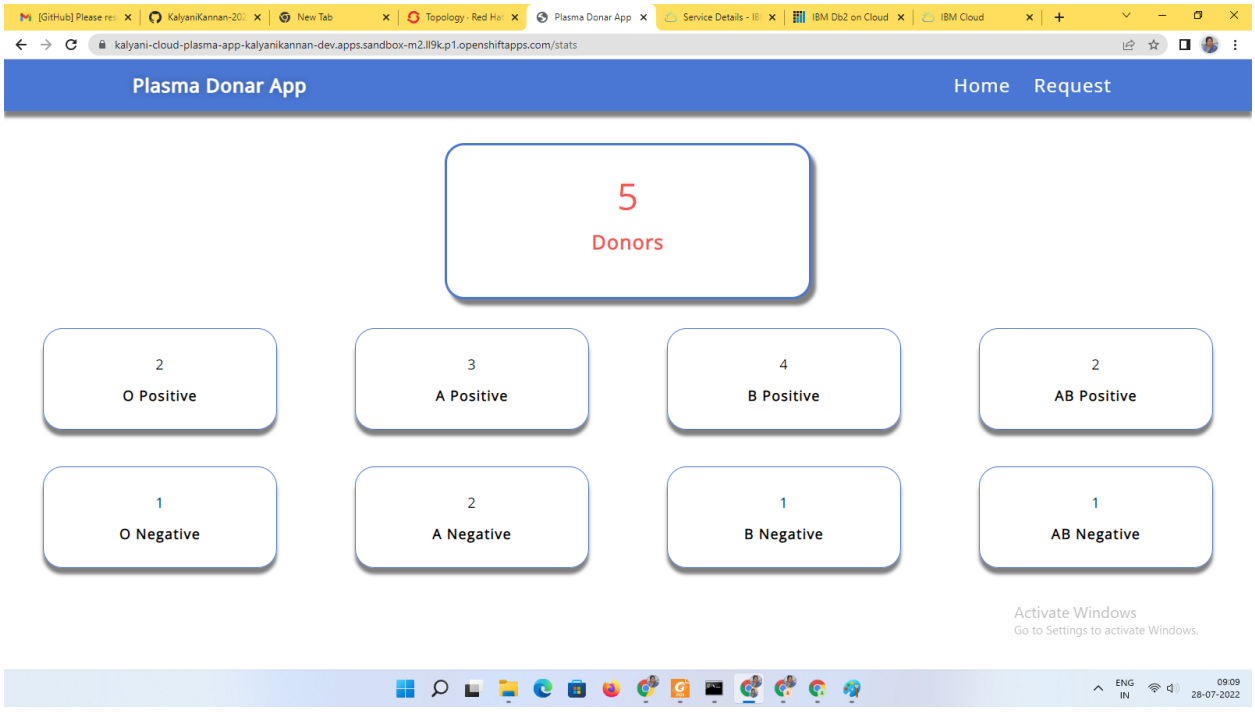


Fig 7 Available Donors page

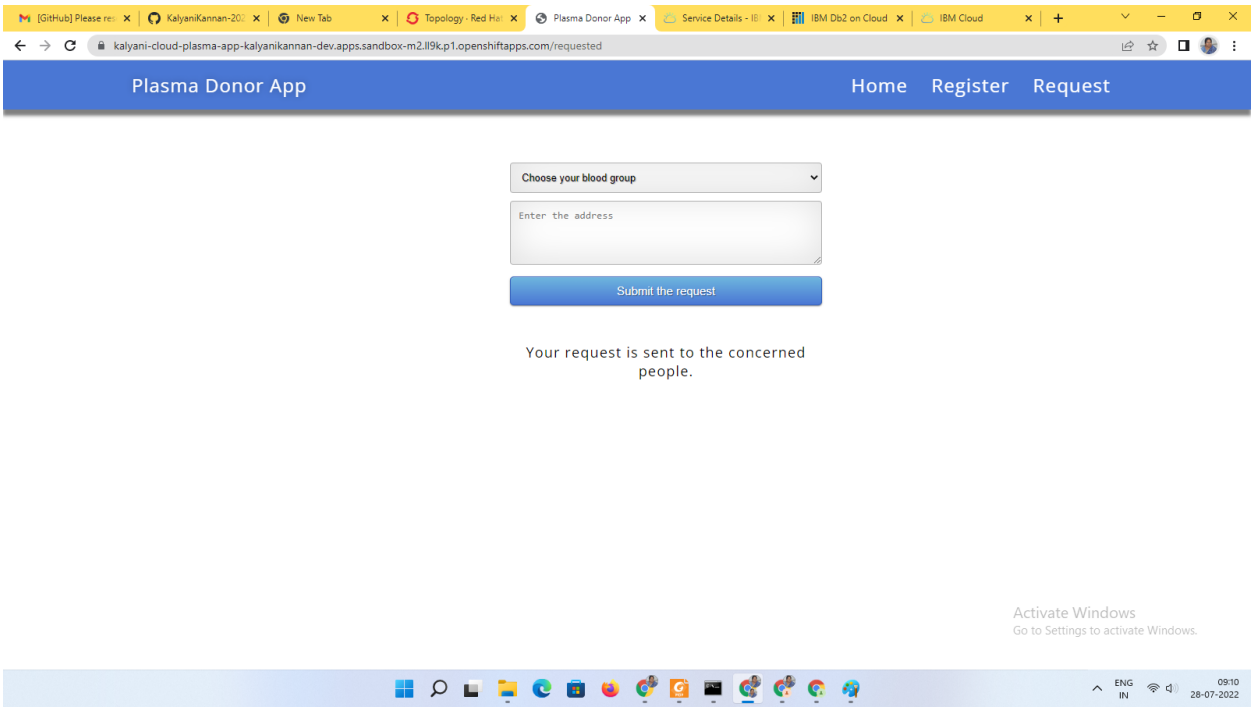


Fig 8 Request confirmation page

6. Merits of the proposed Application

This plasma donor application is very useful for any people in need of blood in their crisis. Being a cloud based application, scalability is much higher.

7. Application

This application could be used by any individual or in hospitals or by blood banks.

Conclusion

Implementing this application in a sandbox environment made a good start for deploying such kind of social cause applications in CI/CD approach.

9. Future Scope

Notifications to the donor through SMS/WhatsApp/email could be done in future and added to this application in a very simple way by using the CodeReady Workspaces. Implementation of this ‘plasma application’ using the CI/CD approach gives rise to easy addition of new features. It also reduces the gap between the development and operations team.

Acknowledgement

I thank the IBM team, AICTE , our Institution and HOD for providing me with such an opportunity to undertake this project-build-athon. Had a good learning experience and got good insights into the CI/CD approach .

***** END OF REPORT *****