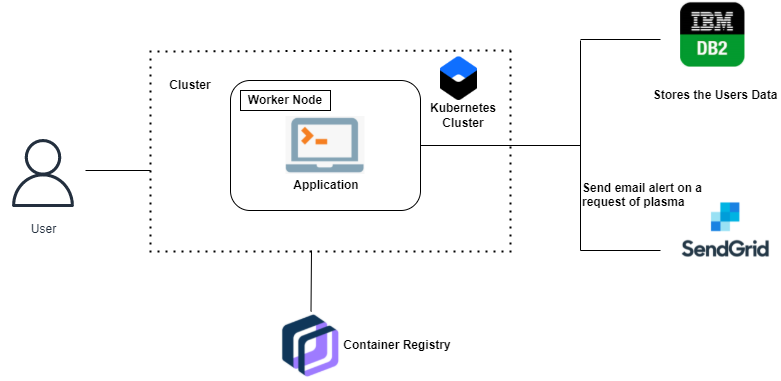
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 16 October 2022 |
| Team ID | PNT2022TMID12996 |
| Project Name | Project – Plasma Donor Application |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | User is presented with a website to interact with the platform, register, login, and place requests for plasma | HTML, CSS, JavaScript |
|  | Application Logic-1 | Login, Register as a patient or donor to the application | IBM DB2 |
|  | Application Logic-2 | Search for donors by entering details of plasma required | IBM DB2 |
|  | Application Logic-3 | Chatbot to help educate the users and navigate through the platform | IBM Watson Assistant |
|  | Database | Used for data appending and retrieval from backend server by users | MySQL |
|  | Cloud Database | Database Service on Cloud to store details about patients, donors, and history of requests and donations made | IBM DB2 |
|  | External API-1 | SendGrid is used to notify the donors if users have made requests for their specific plasma or compatible plasma | SendGrid |
|  | Infrastructure (Server / Cloud) | Deployed on container registry after containerizing image | Docker, Kubernetes, IBM Container Registry |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Flask framework of python is used to build the web application.  Kubernetes is used to containerize the application, deploy and maintain it. | Flask, Kubernetes |
|  | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Transport layer security, IBM Object Storage |
|  | Scalable Architecture | A 3-tier architecture is used, with different layers for the database, logic and presentation | Docker |
|  | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Kubernetes |
|  | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc. | Docker |