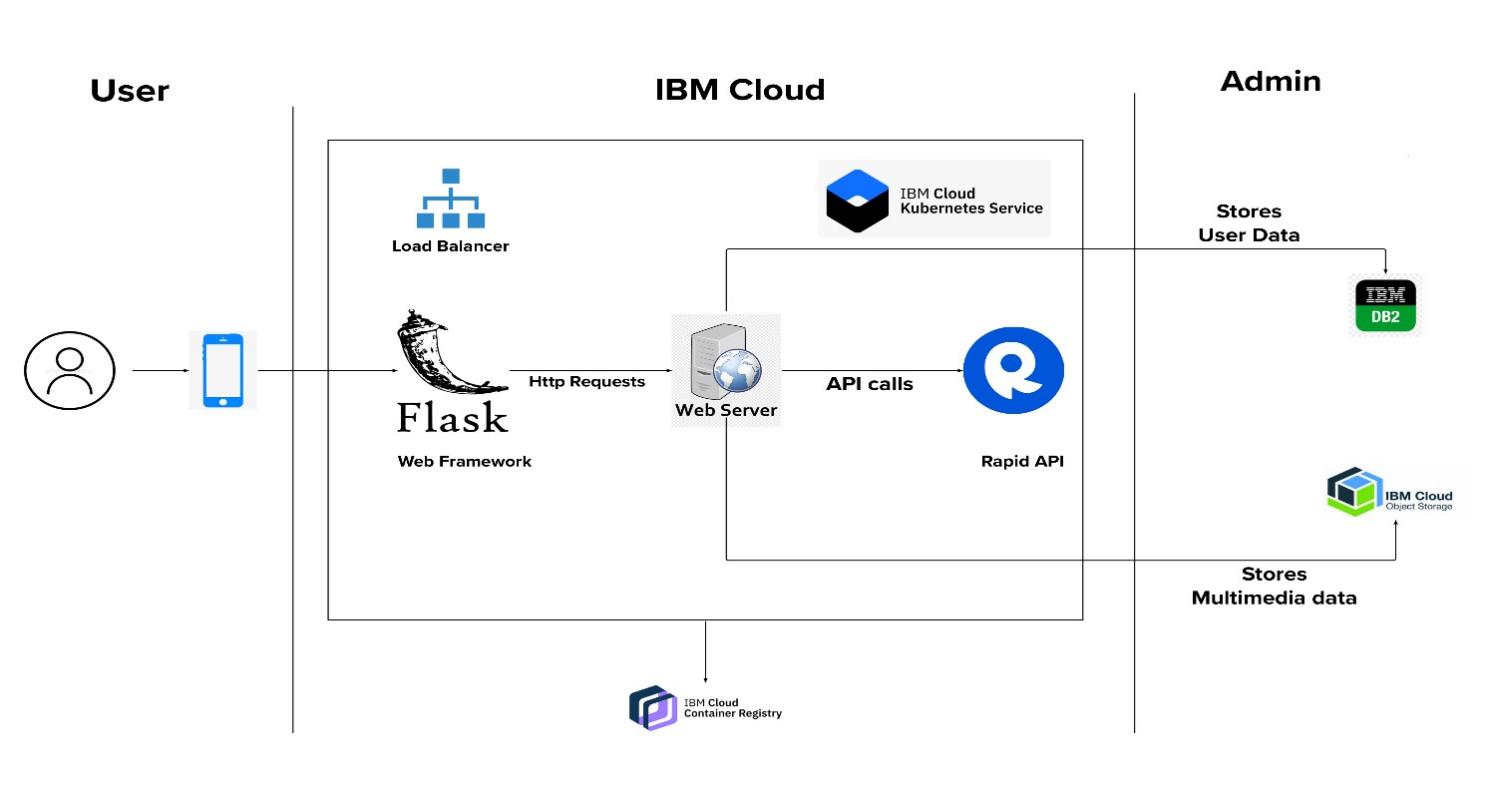
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 24 May 2023 |
| Team ID | NM2023TMID07188 |
| Project Name | Journey to Sustainability :Decoding the C02 Impact of Travel |

**Technical Architecture:**



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How user interacts with the application e.g. Web UI. | HTML, CSS, JavaScript |
| 2. | Backend(web framework) | The server side of the application handles data processing and communication with the front end. | Flask, Python |
| 3. | Database | The storage and management of data used by the application. | SQL |
| 4. | Cloud Database | Database service on cloud | IBM DB2 |
| 5. | Cloud platform | The cloud-based infrastructure for hosting and deploying the application. | IBM cloud |
| 6. | Authentication | The mechanism for verifying the identity of users and granting access to the application. | IBM App ID |
| 7. | External API | Purpose of External API used in the application is to fetch memes based on the keyword and display it to the user. | Rapid API(Humor-Jokes-and-Memes) |
| 8. | Deployment | The process of deploying the application to a production environment. | Docker, Kubernetes |
| 9. | Version control | The system for managing and tracking changes to the application's source code. | GitHub |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | Django, Flask, ReactJS | Python, MySQL, HTML, CSS, Javascript |
| 2. | Security Implementations | * Regular Security Audits and Updates * Role based Access control * Secure authentication * Input validation | Encryption, Authentication and  Authorization, Web application firewalls,  Access Control,  Security Monitoring and logging. |
| **S.No** | **Characteristics** | **Description** | **Technology** |
|  |  | * Secure file updates * Privacy considerations |  |
| 3. | Scalable Architecture | * Presentation Tier:   This tier manages user interface. Uses load balancer to distribute incoming traffic across multiple web servers. CDN helps in caching and delivery of resources such as images javascript etc.   * Application Tier:   This tier is meant for business logic and application processing. The components used are Application servers, API gateways, micro services.   * Data Tier:   This tier manages storage and retrieval of data(memes). Components used are Database servers, distributed file storage. | Web server: IBM WebSphere Application server, Content Delivery Network(CDN), IBM Load Balancer for IPv4/ IPv6.        Application server: Python(Flask), IBM Cloud Monitoring with Sysdig, IBM API connect.      IBM DB2, IBM Cloud Object Storage,  IBM Cloud Kubernetes Service |
| 4. | Availability | Redundancy and Failure, Cloud infrastructure, Traffic distribution, Scalability | IBM cloud load balancer, IBM Cloud Availability Monitoring, IBM cloud Internet Services. |
| 5. | Performance | Caching and Content Delivery, Content Optimization , Scalability and load handling, Server side performance. | Content Delivery Network, HAProxy – Balances HTTP and TCP traffic. |