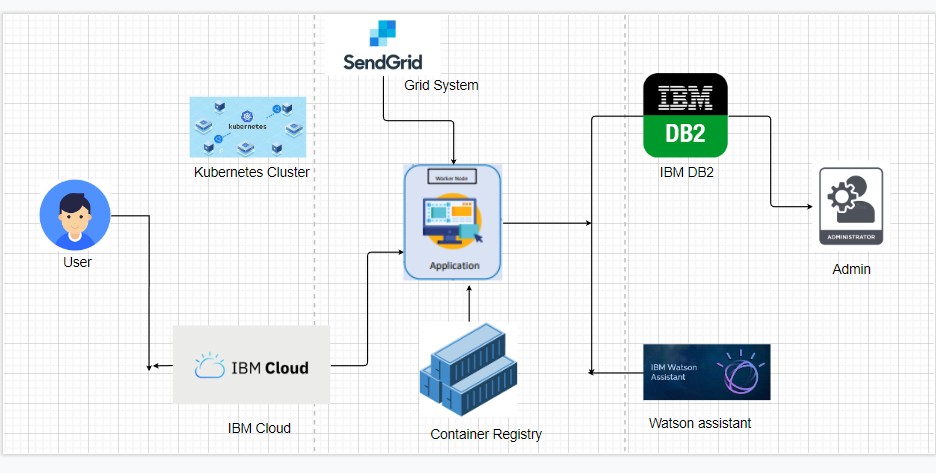
Project Design Phase-II Technology Stack (Architecture & Stack)

|  |  |
| --- | --- |
| Date | 8-NOVEMBER 2022 |
| Team ID | PNT2022TMID47056 |
| Project Name | Smart fashion recommender system |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N**  **o** | **Component** | **Description** | **Technology** |
| 1. | User Interface | user interacts with application e.g.  Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript /  Angular Js / React Js etc. |
| 2. | Requests | How web application communicates  with server | Javascript / Python |
| 3. | Application Logic-2 | The application includes login where user can login with their credentials and also supports registration where new users can be added. | Python |



|  |  |  |  |
| --- | --- | --- | --- |
| 4. | Watson chatbot | The application includes a chatbot which helps the user in  recommendation of products. | IBM Watson Assistant |
| 5. | IBM cloud DB2 | Details of customers and products are stored.  Data types are String, Numeric, Date, time, and timestamp distinct types.  Act\_ sortmem\_ limit, auto\_ del\_ rec \_ obj, auto\_ maint Configuration | MySQL, NoSQL, etc. |
| 6. | Cloud Db2 | A fully managed cloud database with  AI capabilities that keep our website running 24\*7 | IBM DB2, IBM Cloudant etc. |
| 7. | Kubernetes | Manage the complete process in the  stable state If any software crash it automatically restart the work | Kubernetes |
| 8. | Infrastructure (Server /  Cloud) | Application Deployment on Local  System / Cloud  Local Server Configuration: Cloud Server Configuration : | Local, Cloud Foundry,  Kubernetes, etc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N**  **o** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | Cloud Stack, Eucalyptus. Open  Nebula, App Scale, Docker | Docker |
| 2. | Security Implementations | Authentication and password management Accountability to authorize and monitor the use  anonymous accounts and to remove | Encryptions, Secured Authorization |
| 3. | Scalable Architecture | Handles large number users on demand | Container registry, Kubernetes |
| 4. | Availability | The application can be accessed at any time.The administrator needs to look up the stock availability in the  database | Docker |
| 5. | Performance | Speed up the webpage Site optimization based on data analysis. | Kubernetes |

**Table-2: Application Characteristics:**