

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

| | |
|---------------|-------------------------------------------|
| Date | 21 October 2022 |
| Team ID | PNT2022TMID04841 |
| Project Name | Inventory Management System for Retailers |
| Maximum Marks | 4 Marks |

Technical Architecture:

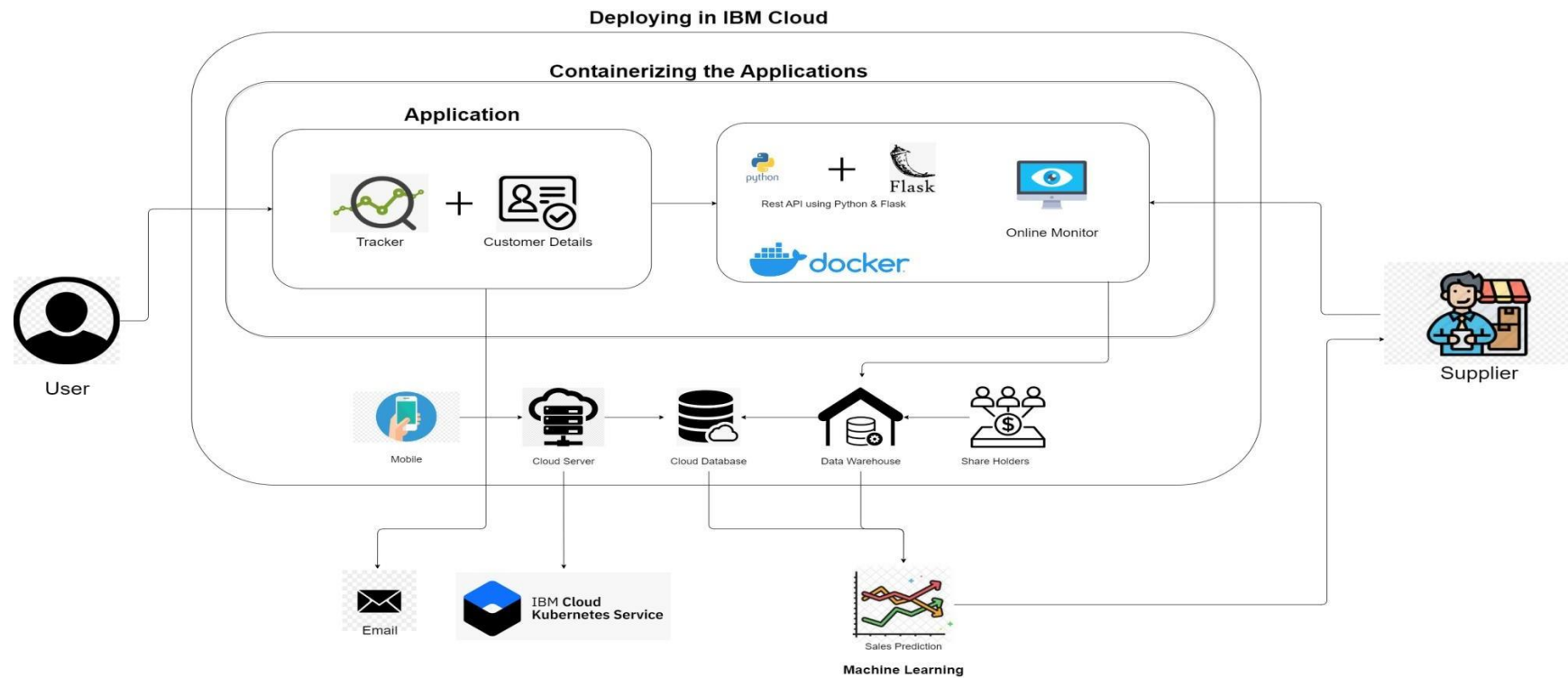


Table-1 : Components & Technologies:

| S.No | Component | Description | Technology |
|------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| 1. | User Interface | How user interacts with application e.g. Web UI, Mobile App, Chatbot, etc. | HTML, CSS, JavaScript, IBM Cloud Object Storage, Python-Flask, Kubernetes, Docker, IBM DB2, IBM Container Registry. |
| 2. | Application Logic | The logic for a process in the application | Python-Flask. |
| 3. | Database | Data Type Configuration etc. | MySQL, etc. |
| 4. | ChatBox | Chatbox for users to access help from a virtual assistant on the application. | IBM Watson Assistant |
| 5. | Cloud Database | Database Service on Cloud | IBM DB2 |
| 6. | File Storage | File storage requirements | IBM Cloud Object Storage |
| 7. | App Container | Contain the whole application in a single container. | Docker Container, IBM Container Registry |
| 8. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud Local Server Configuration: port 5000 Cloud Server Configuration : | Local, Cloud Foundry, Kubernetes. |
| 9. | Send Mail | To send emails when low stock is present in the inventory to retailers. | IBM SendGrid |

Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| 1. | Open-Source Frameworks | We use HTML, CSS, Bootstrap and Flask as the open source for our application | HTML, CSS, JavaScript, Bootstrap, Python-Flask. |
| 2. | Security Implementations | User log in and authentication are done to provide secure access to their account. | IBM Cloud Security, Cookies.. |
| 3. | Scalable Architecture | The system can be scalable easily by using these technologies as to optimize, improve and add new features, allocate sufficient bandwidth to allow more users at a time, etc. | Docker, Kubernetes Cluster |
| 4. | Availability | System availability is high as we make sure the unwanted database access is minimized through SQL and code optimization. | IBM Db2, IBM Container Registry |
| 5. | Performance | Deployment is easy and fast by containerizing the application. Providing fast access time and responsiveness by deploying the application in cloud. | Flask, Docker, IBM Db2. |