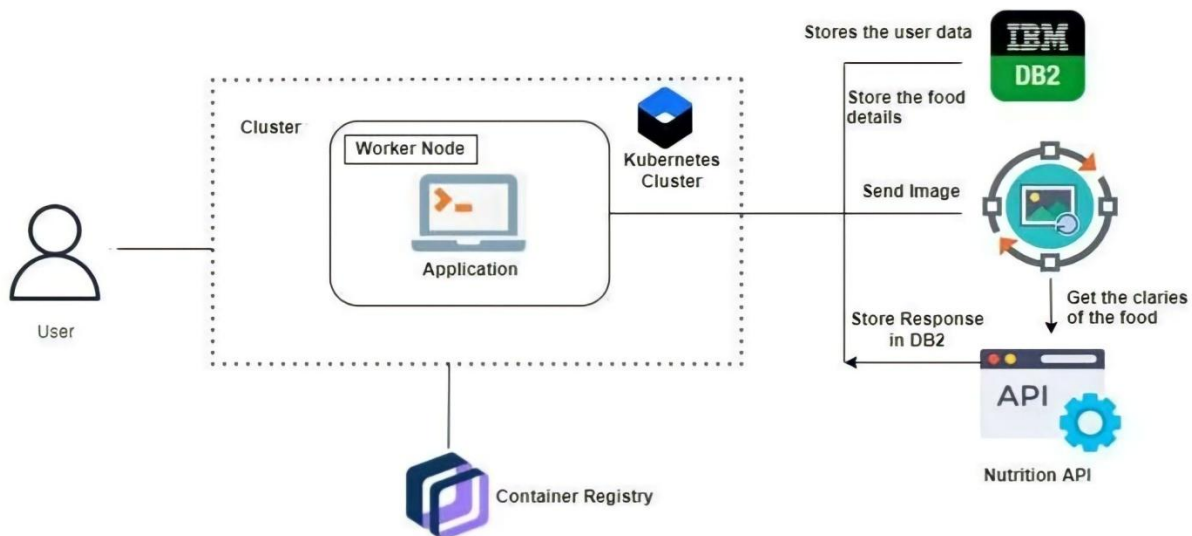


## Project Design Phase-II

### Technology Stack(Architecture & Stack)

|               |                                 |
|---------------|---------------------------------|
| Date          | 14 October 2022                 |
| Team ID       | PNT2022TMID45649                |
| Project Name  | Nutrition Assistant Application |
| Maximum Marks | 4 Marks                         |

### Technical Architecture:



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

| S.No. | Component         | Description   | Technologies                         |
|-------|-------------------|---|--------------------------------------|
| 1.    | User Interface    | How user interacts with application e.g. Web UI, Mobile App, Chatbot etc. | HTML, CSS, Javascript, Flask, Python |
| 2.    | Application Logic | Logic for a process in the application.                                   | Python                               |

|           |                        |   |   |
|-----------|------------------------|---|---|
| <b>3.</b> | Database               | Data Type, Configurations etc.  | MySQL   |
| <b>4.</b> | Cloud Database         | Database Service on Cloud.  | IBM DB2, IBM Cloudant   |
| <b>5.</b> | File Storage           | File storage requirements.  | IBM Block Storage or Other Storage Service or Local File system           |
| <b>6.</b> | External SendGrid      | API-1-<br>The SendGrid service will be used to alert users of various notifications etc as defined by the user.   | SendGrid  |
| <b>7.</b> | External NutritionAPI  | API-2-<br>The service will be used for image recognition.   | NutritionAPI  |
| <b>8.</b> | Machine Learning Model | Pre trained model available through the API to recognize food items.  | Object Recognition Model  |
| <b>9.</b> | Deployment             | Application Deployment on Local System / Cloud<br>Local Server Configuration: It will run on the local server/client side to allow user to interact with Web UI. Cloud Server Configuration: It will be hosted on the cloud for the user to user. This is done through containerization of the application using Docker, stored in the container registry, and will be run by Kubernetes. | IBM Cloud Registry, IBM Cloud Object Storage, IBM DB2, Docker, Kubernetes |

**Table-2 : Application Characteristics:**

| <b>S.No.</b> | <b>Characteristics</b>   | <b>Description</b>   | <b>Technologies</b>  |
|--------------|--------------------------|--|--|
| <b>1.</b>    | Open-Source Frameworks   | List the open-source frameworks used   | Python flask   |
| <b>2.</b>    | Security Implementations | List all the security / access controls implemented, use of firewalls etc                      | SHA-256, Encryptions, IAM Controls   |
| <b>3.</b>    | Scalable Architecture    | Justify the scalability of architecture (3 – tier, Micro-services)                             | IBM DB2, IBM Cloud Object Storage, Kubernetes  |
| <b>4.</b>    | Availability             | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) |
| <b>5.</b>    | Performance              | Performance depends on the availability of compute power in the cloud.                         | IBM Cloud Object Storage, Kubernetes, Docker Images, IBM DB2, SendGrid                         |