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

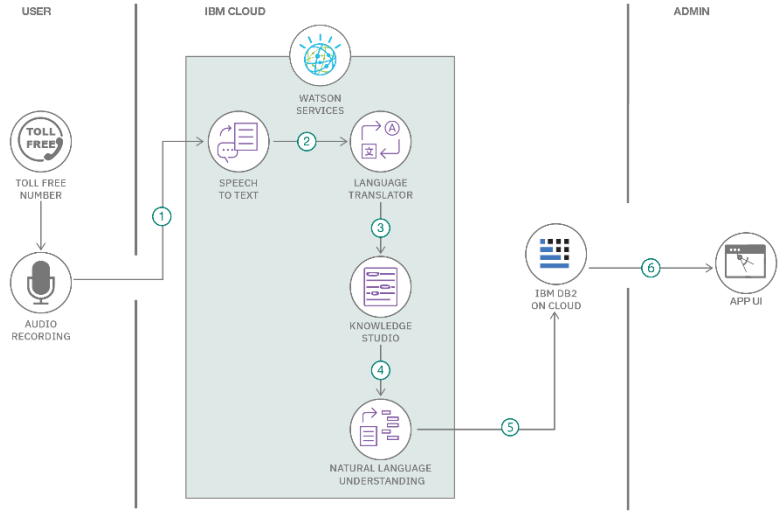
Date	28th October 2022
Team ID	1.2
Project Name	AI – Enhanced Intrusion Detection System
Maximum Marks	4 Marks

Technical Architecture:

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

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/



Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with the system through a web-based user interface. This UI provides access to intrusion detection alerts and system configurations.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	This component encompasses the core logic for detecting and analyzing network intrusions. It includes algorithms and processes for identifying anomalous network activities.	Java / Python
3.	Application Logic-2	The system integrates with IBM Watson's Speech to Text (STT) service to convert audio and voice data into text for analysis. It's a crucial part of processing spoken commands or instructions related to intrusion detection	IBM Watson STT service
4.	Application Logic-3	IBM Watson Assistant is used for building a chatbot-like interface for user interactions. It provides a natural language way to query the system, obtain intrusion reports, or configure the detection parameters.	IBM Watson Assistant
5.	Database	The database stores configuration data, detected intrusion records, and historical information related to network activities.	MySQL, NoSQL, etc.
6.	Cloud Database	A cloud-based database service used for storing additional data, backups, and for facilitating system scalability.	IBM DB2, IBM Cloudant etc.
7.	File Storage	This component covers file storage requirements such as storing configuration files, logs, and reports.	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	The system leverages external APIs, like the IBM Weather API, to enhance its functionality. For	IBM Weather API, etc.

		example, it can use weather data to improve intrusion detection by accounting for weather-related anomalies.	
9.	External API-2	Another external API, the Aadhar API, is used for identity verification and additional security measures when accessing the system.	Aadhar API, etc.
10.	Machine Learning Model	The system employs machine learning models, specifically an Object Recognition Model, to improve the accuracy of intrusion detection by identifying objects or patterns within network data.	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	The application can be deployed both on local servers and in cloud environments, including Cloud Foundry and Kubernetes. This ensures flexibility in deployment based on user preferences and scalability requirements.	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The project leverages various open-source frameworks to build a flexible and extensible intrusion detection system. These frameworks are essential for rapid development and integration of features and modules.	Technology of Open-source framework used, e.g., TensorFlow, scikit-learn, Django, Flask, etc.
2.	Security Implementations	The application places a strong emphasis on security. It implements various security measures to safeguard against unauthorized access and data breaches. These include cryptographic techniques such as SHA-256 for data integrity, encryption for sensitive data protection, IAM (Identity and Access Management) controls to manage user access, and adhering to OWASP (Open Web Application	SHA-256, Encryption, IAM Controls, OWASP Best Practices, Firewall Technologies, etc.

		Security Project) guidelines for web application security best practices. Additionally, firewalls are employed to protect the application from external threats.	
3.	Scalable Architecture	The architecture of the system is designed for scalability. It adopts a microservices-based architecture, allowing individual components to scale independently. This design choice ensures that the system can accommodate increased workloads by adding more microservice instances as required.	Microservices Architecture, Containerization (e.g., Docker, Kubernetes), Scalability Patterns and Strategies.
S.No	**Characteristics**	**Description**	**Technology**
4.	Availability	Ensuring high availability, the application utilizes load balancers to distribute traffic evenly across multiple servers. In the event of a server failure, requests are automatically rerouted to healthy servers, minimizing downtime. Distributed server configurations also contribute to increased availability.	Load Balancers (e.g., Nginx, HAProxy), Redundant Servers, Server Clustering, Failover Mechanisms, High Availability Architectural Patterns.
5.	Performance	To optimize performance, the application employs several design considerations. It is capable of handling a high number of requests per second, thanks to efficient coding practices and optimized algorithms. Caching mechanisms are in place to reduce the response time for frequently accessed data. Content Delivery Networks (CDNs) are utilized to accelerate the delivery of static assets, ensuring a smooth user experience.	Efficient Algorithm Design, Caching Strategies (e.g., Redis, Memcached), Load Testing and Performance Optimization, Integration with CDNs for Static Asset Delivery, Monitoring and Profiling Tools.

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d