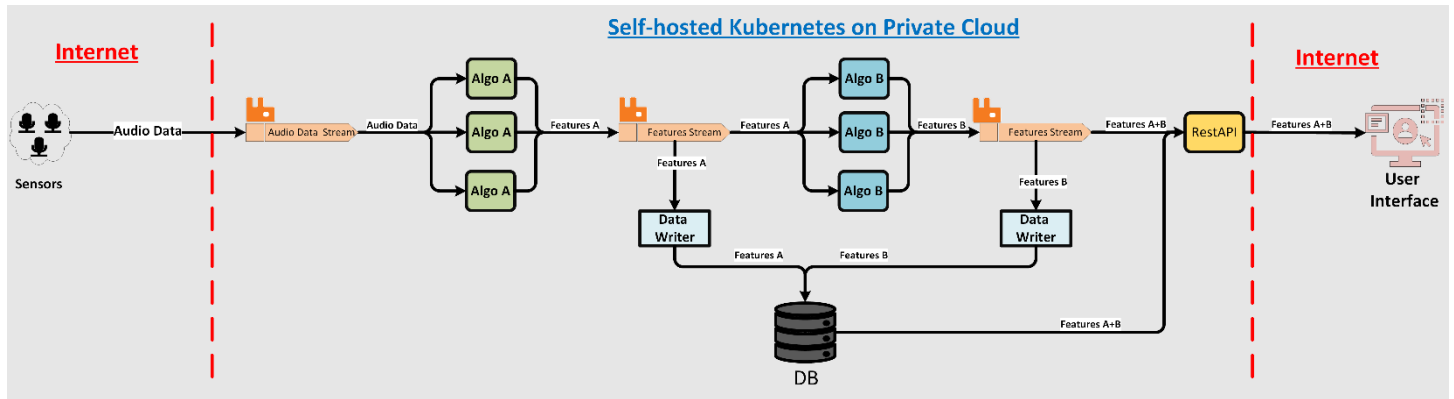


Automation Assignment – Audio Processing System

System Overview

You are presented with a schematic flow of an audio processing system. The system includes distributed sensors, message queues (RabbitMQ), Kubernetes pods running processing algorithms, asynchronous database writing, and a REST API serving external clients.



System Description:

1. **Sensors** distributed across the network record audio and securely transmit it to a dedicated stream queue on a message broker server (RabbitMQ) deployed within a Kubernetes infrastructure on a private cloud.
2. The audio data flowing into the audio queue is consumed by multiple pods running **Algorithm A** in a **load-balanced configuration**. Each pod retrieves only a portion of the messages from the queue, ensuring that all messages are processed collectively by the group of pods.
3. **Algorithm A** processes the audio and generates **Feature Type A**, which is then sent to another dedicated queue on the message broker – the **Features Stream**.
4. **Feature Type A** is consumed by **Algorithm B** (also using a load-balanced setup), which processes the data and generates **Feature Type B**, sending it back to a dedicated queue on the message broker.
5. The **DataWriter** process listens to all messages in the features queue and writes them asynchronously to the **database (DB)**.
6. The **RestAPI** process exposes the system's outputs (Feature Types A and B) to external clients as follows:
 - **For real-time data**, messages are retrieved from the features queue, stored locally for X minutes, and available to consumers as needed.
 - **For historical data**, features are retrieved from the database for the relevant time periods requested by consumers.
7. All consumers of the **RestAPI service** are located on the internet, outside the system's Kubernetes cluster.

Assumptions: All messages are in JSON format (audio and features).

Assignment Tasks

1. Test Design

Based on the system description, design a comprehensive test plan that includes:

a. Types of Tests

- Unit Tests
- Integration Tests
- Load & Performance Tests
- Security Tests
- Manual vs. Automated Tests

b. Coverage

Ensure your test plan covers all components:

- Sensors → RabbitMQ
- Algorithm A & B pods
- DataWriter → DB
- REST API (real-time and historical)
- External client access

c. Test Objectives

For each test type, explain:

- What is being tested?
- Why is it important?
- What are the expected outcomes?

d. Automation Strategy

- Which tests would you automate?
- What tools would you use (e.g., Pytest, Postman, JMeter, Selenium, etc.)?
- Provide example test scenarios or pseudocode.

e. CI/CD Integration

- How would you integrate your tests into a CI/CD pipeline?
- What tools or platforms would you use (e.g., Jenkins, GitHub Actions)?
- How would you handle test reporting and alerts?

(Optional)

- Include python code / diagrams / flowcharts to illustrate your testing strategy.
- Suggest monitoring or logging tools to support QA efforts (e.g., Prometheus, ELK stack).