Smart Home Transformation with IBM Cloud Functions for IoT Data Processing

Phase – 4 Document Submission

Team Members:

- 110121205029 – Mohamed Shaaheen. A
- 110121205014 – Hasheer Ahamed. M
- 10121205020 – Mohamed Aarkif. S
- 110121205026 – Mohamed Rilwan. J
- 110121205034 – Mohammed Faiz. J
- 110121205050 – Sheejan. J

Project: Serverless IOT Data Processing

Phase – 4: Development Part 2

Step 1: Implement Real-Time Data Processing

a. Set Up Real-Time Triggers:

- Within your IBM Cloud Functions project, create triggers that respond to incoming data in real-time. These triggers can be based on events like new data arriving from the IoT platform.

b. Develop Real-Time Processing Functions:

- Write serverless functions that perform real-time processing on the incoming data. This could include tasks like filtering, aggregation, transformation, or even running machine learning models for immediate insights.

  Example (for IBM Cloud Functions in Python):

  def main(params):

  # Real-time processing logic

  # …

return { result: "Real-time processing complete" }

Step 2: Implement Automation Routines

a. Define Automation Rules:

- Determine the conditions under which automated routines should be triggered. This could be based on specific data thresholds, time-based events, or any other relevant criteria.

b. Create Automation Functions:

- Develop serverless functions that encapsulate the logic for automated routines. These functions will be triggered based on the defined conditions.

  Example (for IBM Cloud Functions in JavaScript):

  function main(params) {

  // Automation routine logic

  // …

  return { result: "Automation routine executed" };

  }

Step 3: Set Up IBM Cloud Object Storage

a. Create Object Storage Instance:

- Navigate to the IBM Cloud Dashboard and create a new instance of IBM Cloud Object Storage.

b. Define Storage Buckets:

- Within the Object Storage instance, set up storage buckets to organize and store processed data. Define access policies as needed.

Step 4: Store Processed Data

a. Integrate Data Storage Functions:

- In your IBM Cloud Functions project, create functions responsible for storing the processed data. These functions should take the processed data and save it to the designated IBM Cloud Object Storage bucket.

  Example (for IBM Cloud Functions in Node.js):

  function main(params) {

  // Store processed data in Object Storage

  // …

  return { result: "Data stored successfully" };

  }

b. Configure Authentication for Object Storage:

- Ensure that your serverless functions have the necessary credentials and permissions to interact with the IBM Cloud Object Storage instance.

Step 5: Test Real-Time Processing, Automation, and Data Storage

a. Simulate Real-Time Events:

- If real devices are not available, simulate events that trigger real-time processing and automation routines.

b. Verify Data Storage:

- Confirm that the processed data is being correctly stored in the designated IBM Cloud Object Storage bucket.

c. Monitor Automation Execution:

- Monitor the execution of automated routines to ensure they are triggered and executed according to the defined rules.