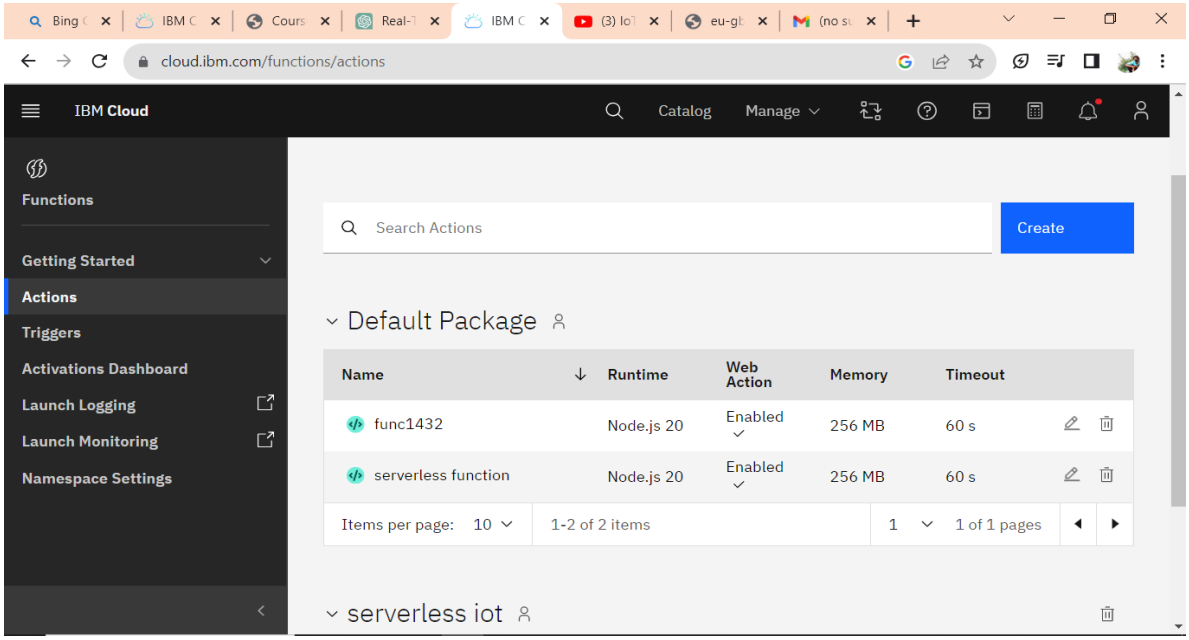
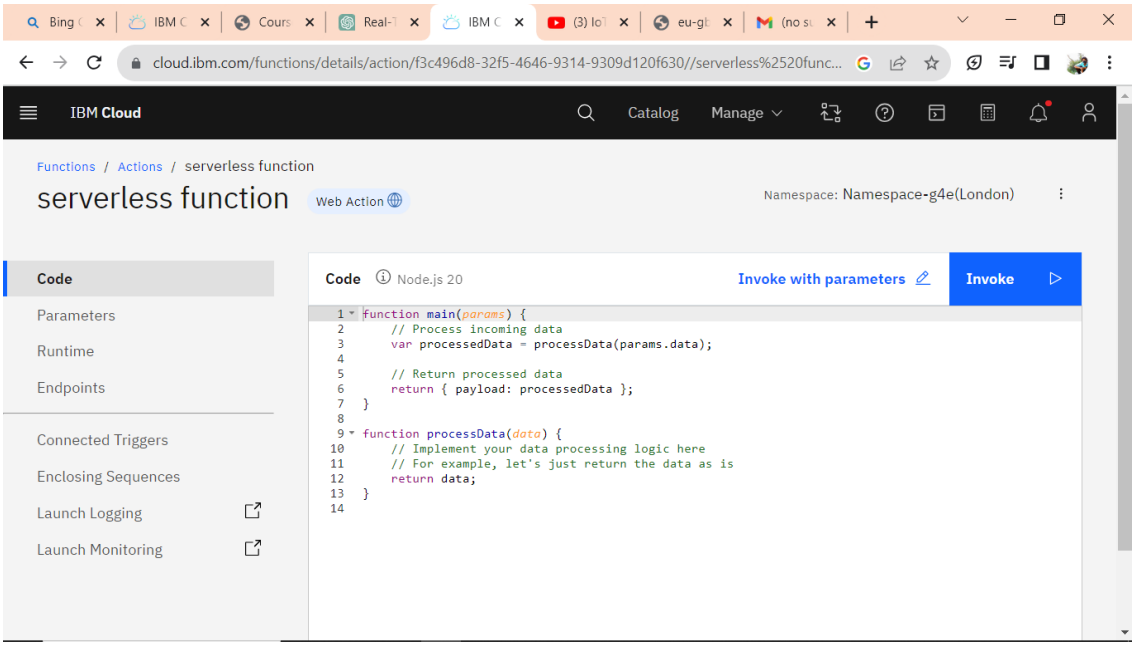
**TECHNOLOGY**:-cloud application development
**PROJECT:-** Serverless IOT data processing
**TITLE:-** To use IBM Cloud Functions to process data and trigger automated routines.Store processed data in IBM Cloud Object Storage for analysis.

To implement real-time data processing, automation, and storage using IBM Cloud Functions and IBM Cloud Object Storage, follow these steps:

**Step 1:** Set up IBM Cloud Functions

1.Log in to your IBM Cloud account.
2.Navigate to the IBM Cloud Functions dashboard.
3.Create a new action (function) by clicking on “Create” -> “Action”.
4.Choose a unique name for your function and select the runtime (e.g., Node.js, Python).



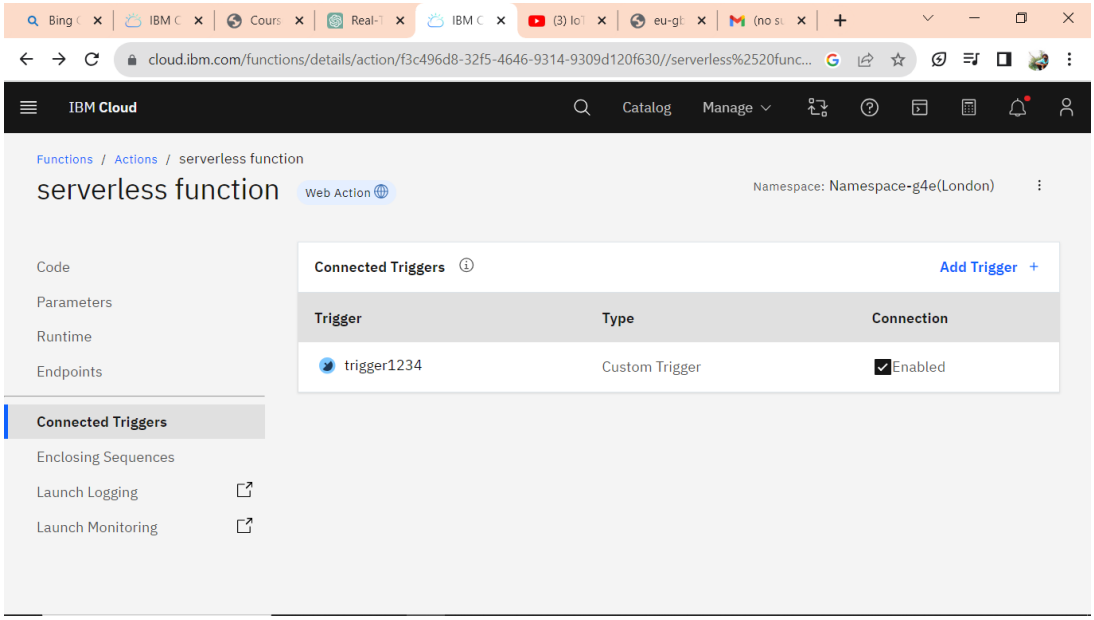



5.In the code editor, write your function for processing data. This function will be triggered in real-time when new data arrives. Here's a simple example of a function in Node.js that processes incoming data:

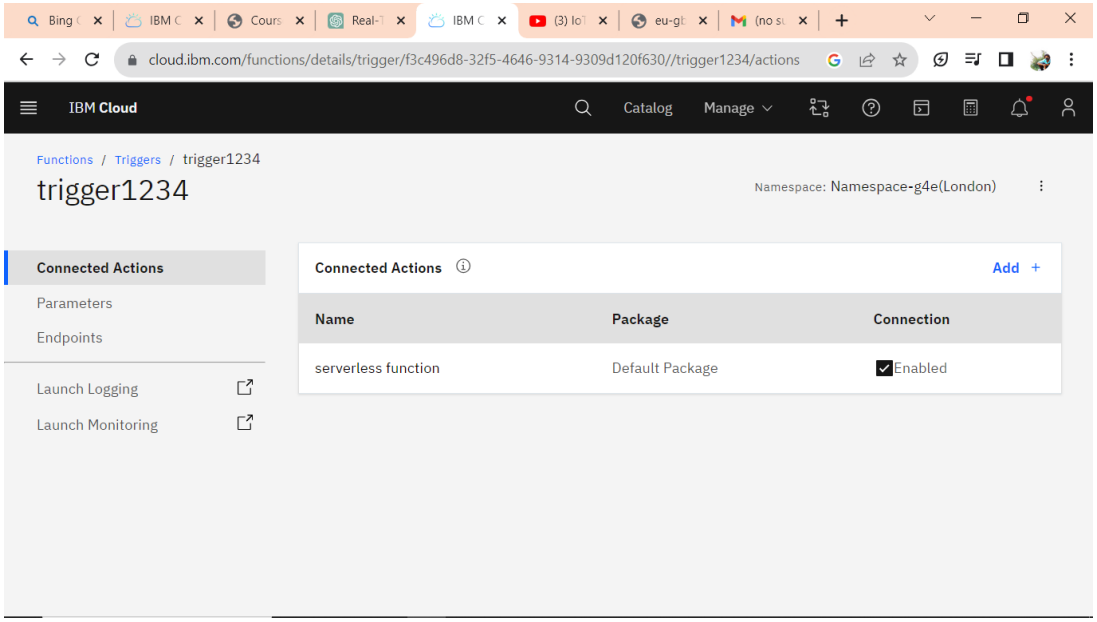
```
function main(params) {
    // Process incoming data
    var processedData = processData(params.data);

    // Return processed data
    return { payload: processedData };
}

function processData(data) {
    // Implement your data processing logic here
    // For example, let's just return the data as is
    return data;
}
```

**Step 2**: Set up Triggers and Rules

1.In the IBM Cloud Functions dashboard, navigate to “Triggers” -> “Create”.
2.Choose a unique name for your trigger and configure it based on your real-time data source.

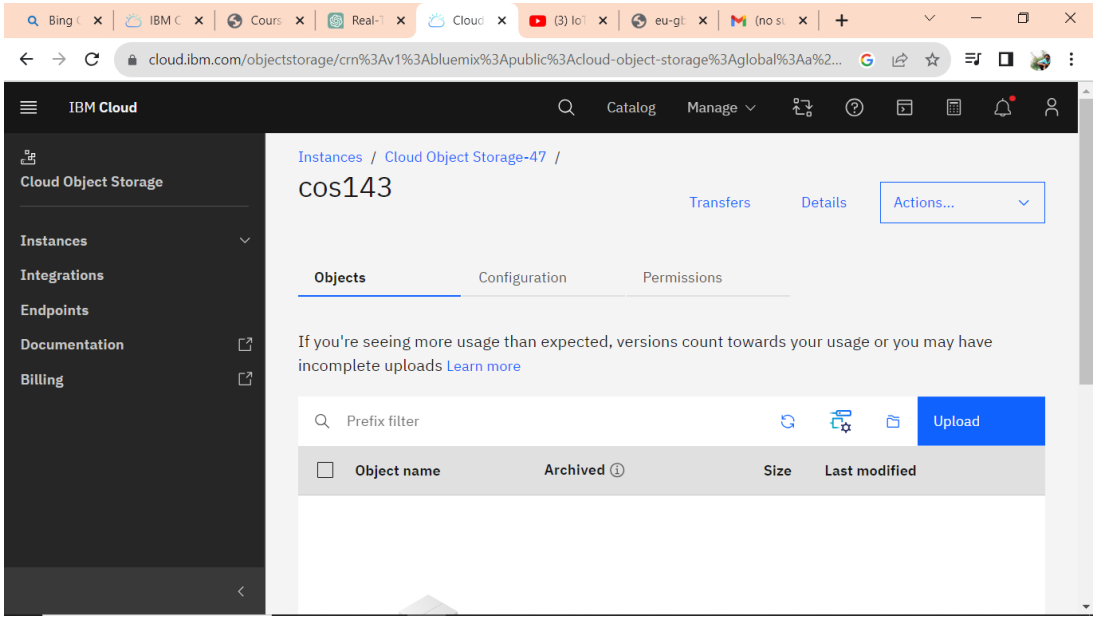



3.Once the trigger is created, go to “Rules” -> “Create”.
4.Connect your previously created action (function) and trigger.




**Step 3:** Set up IBM Cloud Object Storage

1.Log in to your IBM Cloud account.
2.Navigate to the IBM Cloud Object Storage dashboard1.
3.Create a new bucket by clicking on “Create bucket”. Choose a unique name for your bucket1.




4.Configure public access settings based on your requirements2.
**Step 4:** Store Processed Data Modify your cloud function to store the processed data in the object storage bucket:

```
const { IamTokenManager } = require('ibm-watson/auth');
const { S3 } = require('ibm-cos-sdk');

// Initialize IBM COS SDK with your credentials
const cos = new S3({
    endpoint: '<endpoint>',
    apiKeyId: '<api-key>',
    ibmAuthEndpoint: 'https://iam.cloud.ibm.com/identity/token',
    serviceInstanceId: '<service-instance-id>',
});
```

```
// Initialize token manager
const tokenManager = new IamTokenManager({
    apikey: '<api-key>'
});

async function main(params) {
    // Process incoming data
    var processedData = processData(params.data);

    // Store processed data in IBM Cloud Object Storage
    await storeDataInObjectStorage(processedData);

    // Return processed data
    return { payload: processedData };
}

function processData(data) {
    // Implement your data processing logic here
    // For example, let's just return the data as is
    return data;
}

async function storeDataInObjectStorage(data) {
    const token = await tokenManager.getToken();

    const params = {
        Bucket: '<bucket-name>',
        Key: '<object-key>',
        Body: JSON.stringify(data),
```

```
        ACL: 'public-read',
        ContentType: 'application/json'
    };

    return cos.putObject(params).promise();
}
```