

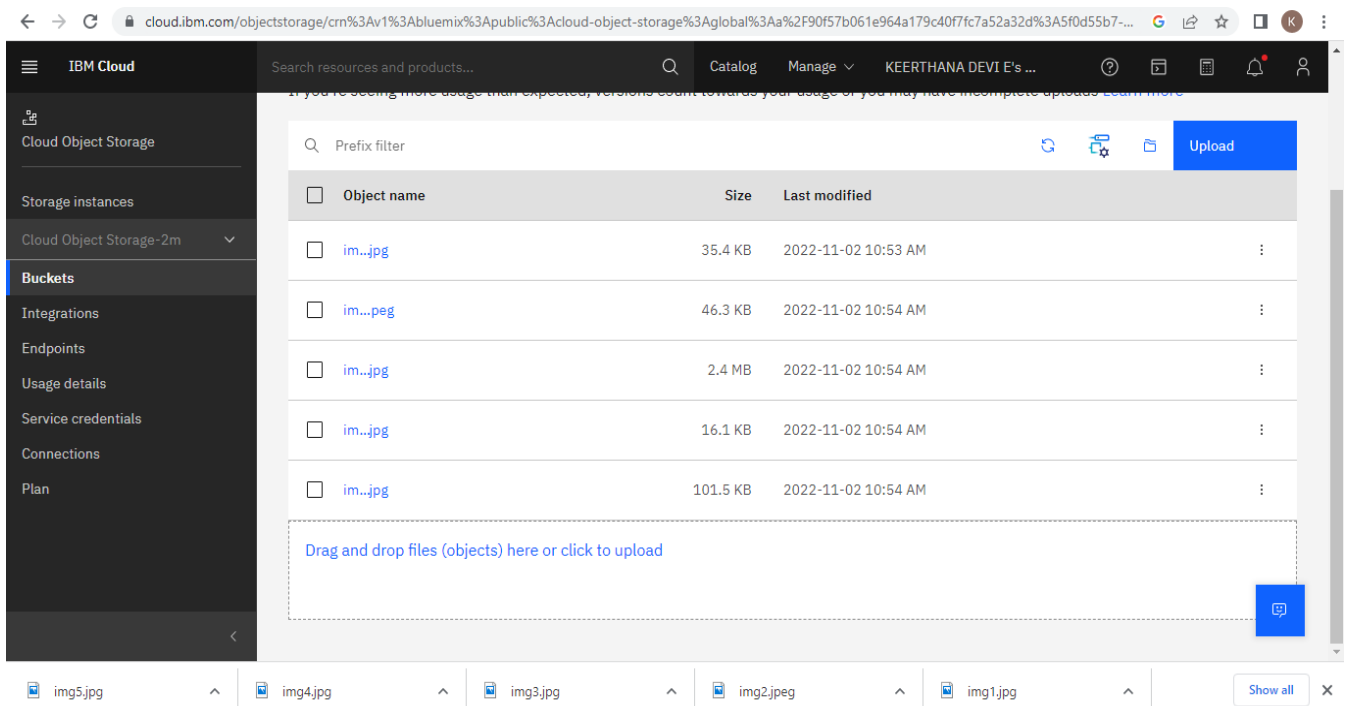
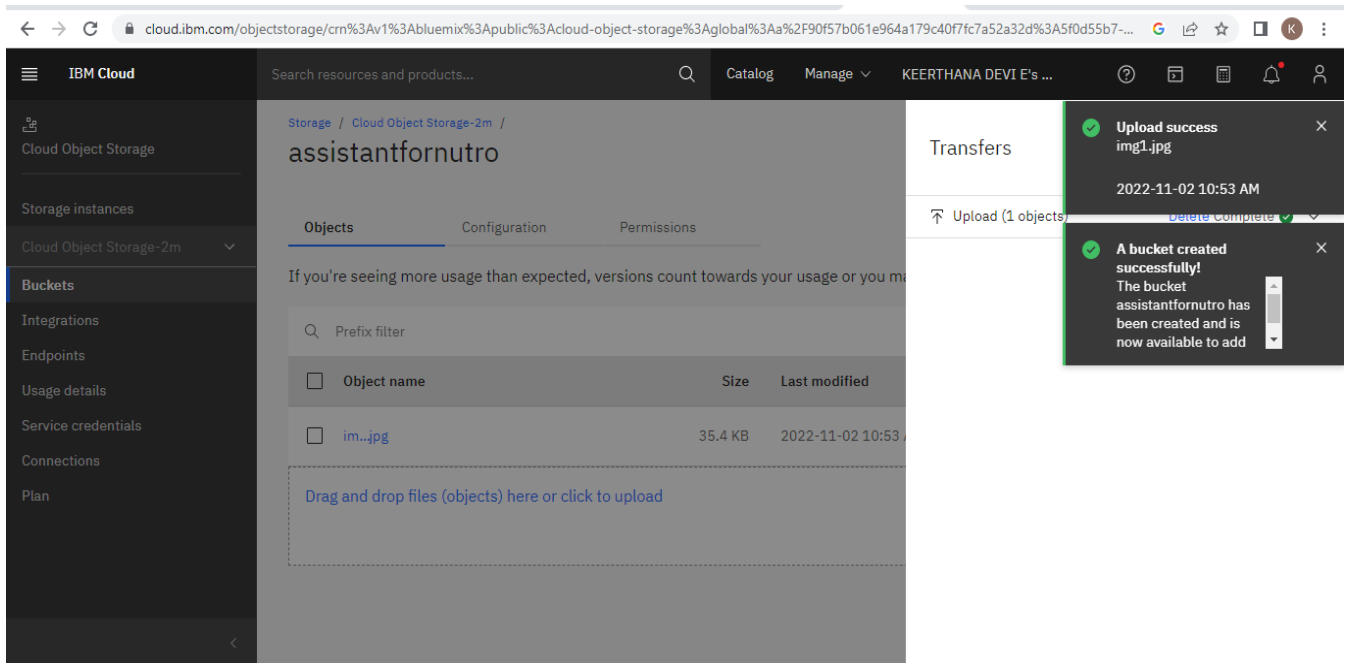
Assignment-3

Date	10 October 2022
Team ID	PNT2022TMID20166
Project name	Nutrition Assistant Application

1. CREATE A BUCKET IN IBM OBJECT STORAGE.

The screenshot displays the IBM Cloud Object Storage interface in a web browser. The browser's address bar shows the URL: `cloud.ibm.com/objectstorage/cm%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F90f57b061e964a179c40f7c7a52a32d%3A5f0d55b7-...`. The IBM Cloud logo is in the top left corner. A search bar and navigation links (Catalog, Manage) are at the top. The left sidebar lists various services, with 'Cloud Object Storage' selected. Under 'Cloud Object Storage', 'Buckets' is highlighted. The main content area shows the details for a bucket named 'assistantfornutro'. A green notification box in the top right corner states: 'A bucket created successfully! The bucket assistantfornutro has been created and is now available to add'. Below the bucket name, there are tabs for 'Objects', 'Configuration', and 'Permissions'. A message indicates that if usage is higher than expected, versions count towards usage or uploads may be incomplete. The 'Objects' tab is active, showing a table with columns for 'Object name', 'Size', and 'Last modified'. Below the table, there is a section titled 'Objects' with instructions: 'Drag and drop files (objects) to upload. An object is your data in fixed form.' and a dashed box with the text 'Drag and drop files (objects) here or click to upload'. An 'Upload' button is visible in the top right of the objects section. The Windows taskbar at the bottom shows the search bar and several application icons, with the system clock indicating 10:49 AM on 11/2/2022.

2. Upload an 5 images to ibm object storage and make it public. write html code to displaying all the 5 images.



IBM Cloud

Search resources and products...

CatalogManageKEERTHANA DEVI E's ...

Cloud Object Storage

Storage instances

Cloud Object Storage-2m

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Public access

Warning:

Granting Public access to this bucket will allow anyone to access the bucket. To revoke public access, remove the "Public access" policy from this bucket within [Access groups](#)

Learn more

Access policy update

Access group policy created

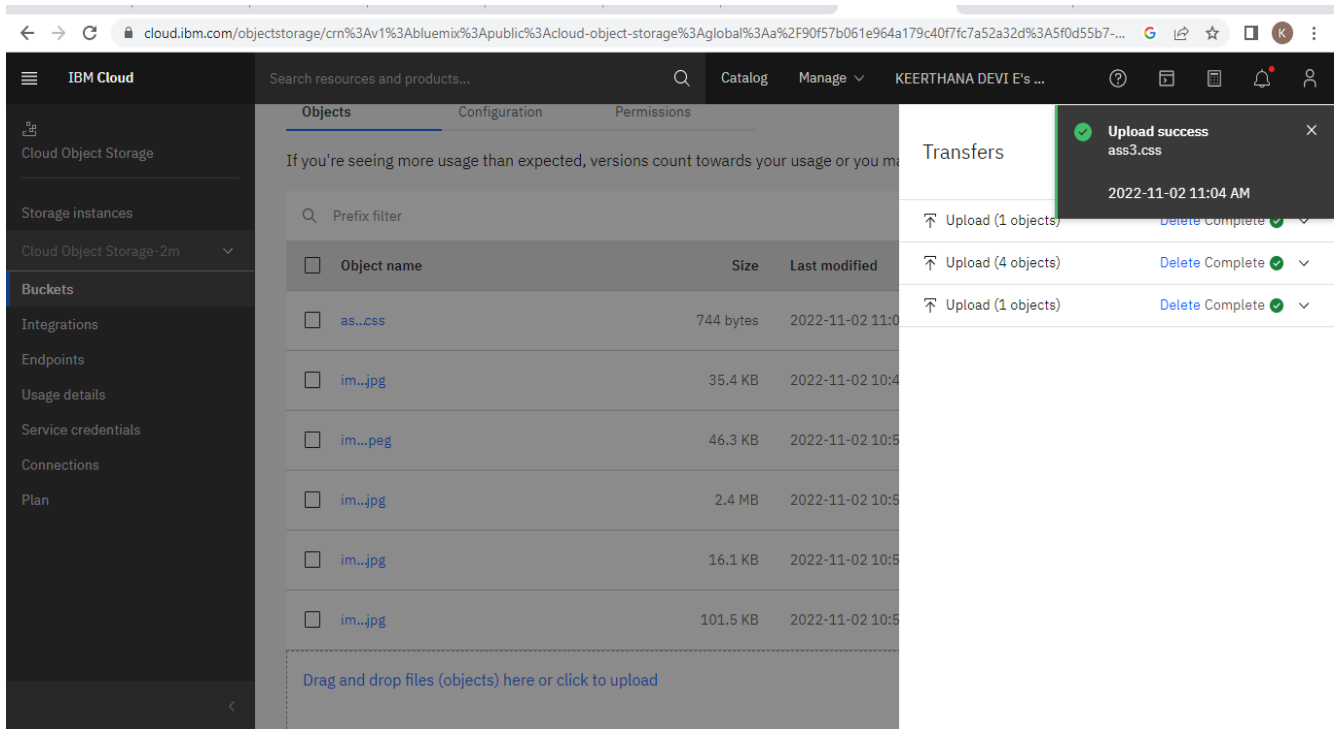
A new access policy for this bucket was created for the group:
Public Access
To delete/edit go to the [IAM console](#).

Create access policy

Context-based restrictions

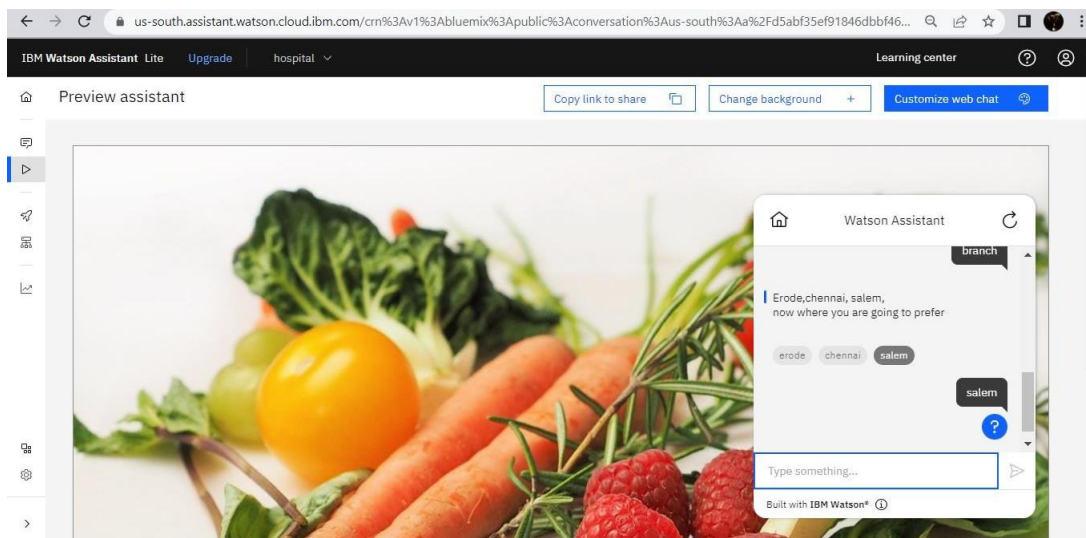
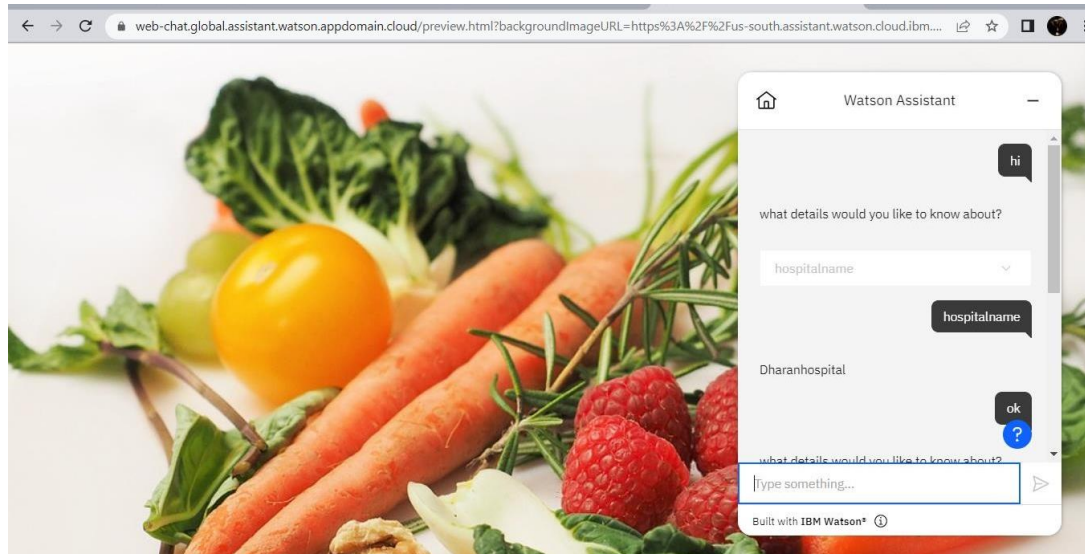
Firewall (legacy)

2. Upload a css page to the object storage and use the same page in your HTML code.



```
index - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Home</title>
<link rel="stylesheet" href="{url for('redirect to',link='https://s3.jp-tok.cloud-object-storage.
appdomain.cloud/cloudbucket/assign3.css')}" type="text/css">
<script>
window.watsonAssistantChatOptions = {
  integrationID: "8c0d878c-d3b4-4a3e-b63a-01b255dc7d6e", // The ID of this integration.
  region: "us-south", // The region your integration is hosted in.
  serviceInstanceID: "398c4defa-af81-42f3-badd-b3b31e98f373", // The ID of your service instance.
  onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
  const t=document.createElement('script');
  t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.
clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
  document.head.appendChild(t);
});
</script>
</head>
<body>
<form action="/uploader" method="POST" enctype="multipart/form-data">
  <input type="text" placeholder="Enter file name" name="filename" />
  <br />
  <input type="file" name="file" />
  <br />
  <input type="submit" />
</form>
<br/>
<br/>
```

3. Design a chatbot using IBM Watson assistant for hospital.



Web URL for Assistant:

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-398c4efa-af81-42f3-badd-b3b31e98f373%3A%3A6299a82c-671e-4181-a618-b8c6251f725a&integrationID=8c0d878c-d3b4-4a3e-b63a-01b255dc7d6e®ion=us-south&serviceInstanceID=398c4efa-af81-42f3-badd-b3b31e98f373>

4. Create Watson assistant service with 10 steps and use 3 conditions in it. Load thatscript in HTML page.

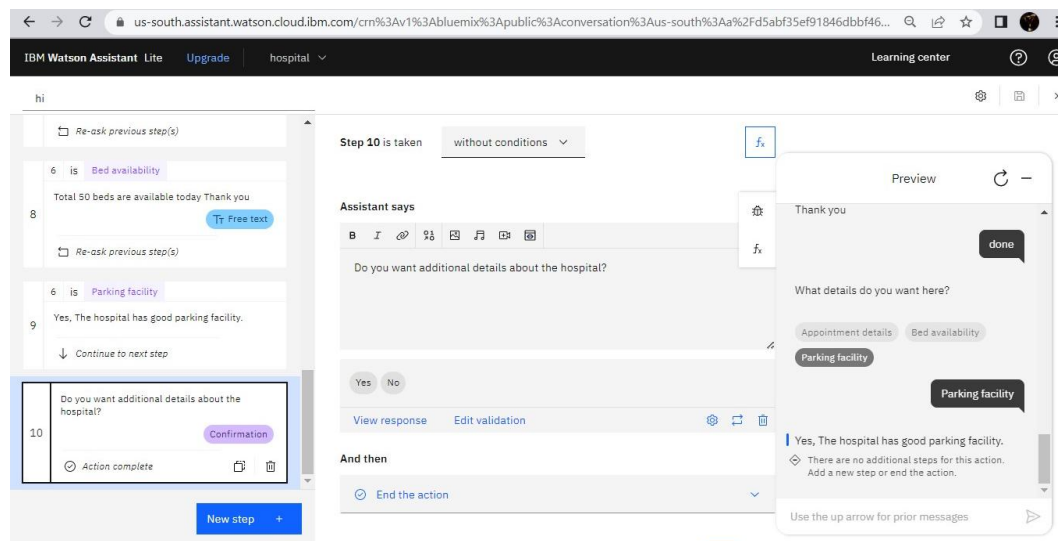


Figure 1. 10 steps of conversation

Included 3 conditions in steps:

The screenshot shows the IBM Watson Assistant interface for a conversation named "hospital". The left sidebar displays the "Conversation steps" list. Step 10 is highlighted, showing a "Continue to next step" action. The main panel shows the configuration for Step 10, which is set to "without conditions". The "Assistant says" section contains a text prompt: "Do you want additional details about the hospital?". Below this, there are "Yes" and "No" buttons. The "And then" section shows an "End the action" option. A "Preview" window on the right shows the conversation flow, including a "Thank you" message and a "Parking facility" button.

The screenshot shows the IBM Watson Assistant interface for the same "hospital" conversation. The left sidebar displays the "Conversation steps" list. Step 4 is highlighted, showing a "Continue to next step" action. The main panel shows the configuration for Step 4, which is set to "with conditions". The "Conditions" section shows a single condition: "If All of this is true: 3. Erode, Che... is salem". The "Assistant says" section contains a text prompt: "How you know about the dharan hospital?". Below this, there are "Social media" and "Review" buttons. The "And then" section shows a "Continue to next step" option. A "Preview" window on the right shows the conversation flow, including a "Thank you" message and a "Parking facility" button.

us-south.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aus-south%3Aa%2Fd5abf35ef91846dbbf46...

IBM Watson Assistant Lite Upgrade hospital Learning center

hi

3 is **salem**

4 How you know about the dharan hospital?

4 is **Relatives**

5 Is your relatives had a treatment here?

6 What details do you want here?

6 is **Appointment details**

Step 5 is taken with conditions

Conditions

If All of this is true:

4, How you k... is Relatives

and Add condition +

New condition group +

Assistant says

Is your relatives had a treatment here?

Preview

Thank you

What details do you want here?

Appointment details Bed availability

Parking facility

Parking facility

Yes, The hospital has good parking facility.

There are no additional steps for this action. Add a new step or end the action.

Type something...

us-south.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aus-south%3Aa%2Fd5abf35ef91846dbbf46...

IBM Watson Assistant Lite Upgrade hospital Learning center

hi

6 is **Bed availability**

8 Total 50 beds are available today Thank you

6 is **Parking facility**

9 Yes, The hospital has good parking facility.

10 Do you want additional details about the hospital?

Confirmation

Action complete

Re-ask previous step(s)

Re-ask previous step(s)

6. What detai... is Parking facility

and Add condition +

New condition group +

Assistant says

Yes, The hospital has good parking facility.

Define customer response

Preview

Thank you

What details do you want here?

Appointment details Bed availability

Parking facility

Parking facility

Yes, The hospital has good parking facility.

There are no additional steps for this action. Add a new step or end the action.

Type something...

Index.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Home</title>

    <link rel="stylesheet" href="{ {url_for('redirect_to',link='https://s3.jp-tok.cloud-
object-storage.appdomain.cloud/cloudbucket/assign3.css') } }" type="text/css">

    <script>

      window.watsonAssistantChatOptions = {

        integrationID: "8c0d878c-d3b4-4a3e-b63a-01b255dc7d6e", // The ID of this integration.

        region: "us-south", // The region your integration is hosted in.

        serviceInstanceID: "398c4efa-af81-42f3-badd-b3b31e98f373",

        // The ID of your service      instance.

        onLoad: function(instance) { instance.render(); }

      };

      setTimeout(function(){

        const t=document.createElement('script');

        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +

        (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

        document.head.appendChild(t);

      });

    </script>

    setTimeout(function(){

      const t=document.createElement('script');

      t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +

      (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

      document.head.appendChild(t);

    });
```

```
</script> </head>
```

```
<body>
```

```
<form action="/uploader" method="POST" enctype="multipart/form-data">
```

```
<input type="text" placeholder="Enter file name" name="filename" />
```

```
<br />
```

```
<br />
```

```
<input type="file" name="file" />
```

```
<br />
```

```
<br />
```

```
<input type="submit" />
```

```
</form>
```

```
<br/>
```

```
<br/>
```

```
<br/>
```

```
{% for row in files %}
```

```
<div style="border: 1px solid #EFEFEF;margin:10px;">
```

```
<h3>Filename : { {row}} </h3>
```

```
</td>
```

```
</div>
```

```
{% endfor %}
```

```
</body>
```

```
</html>
```

App.py

```
import io

from flask import Flask, redirect, url_for, render_template, request

import ibm_boto3

from ibm_botocore.client import Config, ClientError

COS_ENDPOINT="https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID=""
COS_INSTANCE_CRN=""


cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

app=Flask(__name__)


@app.route('/')
def index():
    try:
        files = cos.Bucket('cloudbucket').objects.all()

        files_names = []

        for file in files:
```

```

        files_names.append(file.key)
        print(file)
        print("Item: {0} ({1} bytes)".format(file.key, file.size))
        return render_template('index.html',files=files_names)

except ClientError as be:
    print("CLIENT ERROR: {0}\n".format(be))
    return render_template('index.html')
except Exception as e:
    print("Unable to retrieve bucket contents: {0}".format(e))
    return render_template('index.html')

@app.route('/uploader',methods=['POST'])
def upload():
    name_file=request.form['filename']
    f = request.files['file']
    try:
        part_size = 1024 * 1024 * 5

        file_threshold = 1024 * 1024 * 15

        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,

            multipart_chunksize=part_size
        )

        content = f.read()

```

```
cos.Object('cloudbucket', name_file).upload_fileobj(
    Fileobj=io.BytesIO(content),
    Config=transfer_config
)
return redirect(url_for('index'))
except ClientError as be:
    print("CLIENT ERROR: {0}\n".format(be))
    return redirect(url_for('index'))

except Exception as e:
    print("Unable to complete multi-part upload: {0}".format(e))
    return redirect(url_for('index'))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```