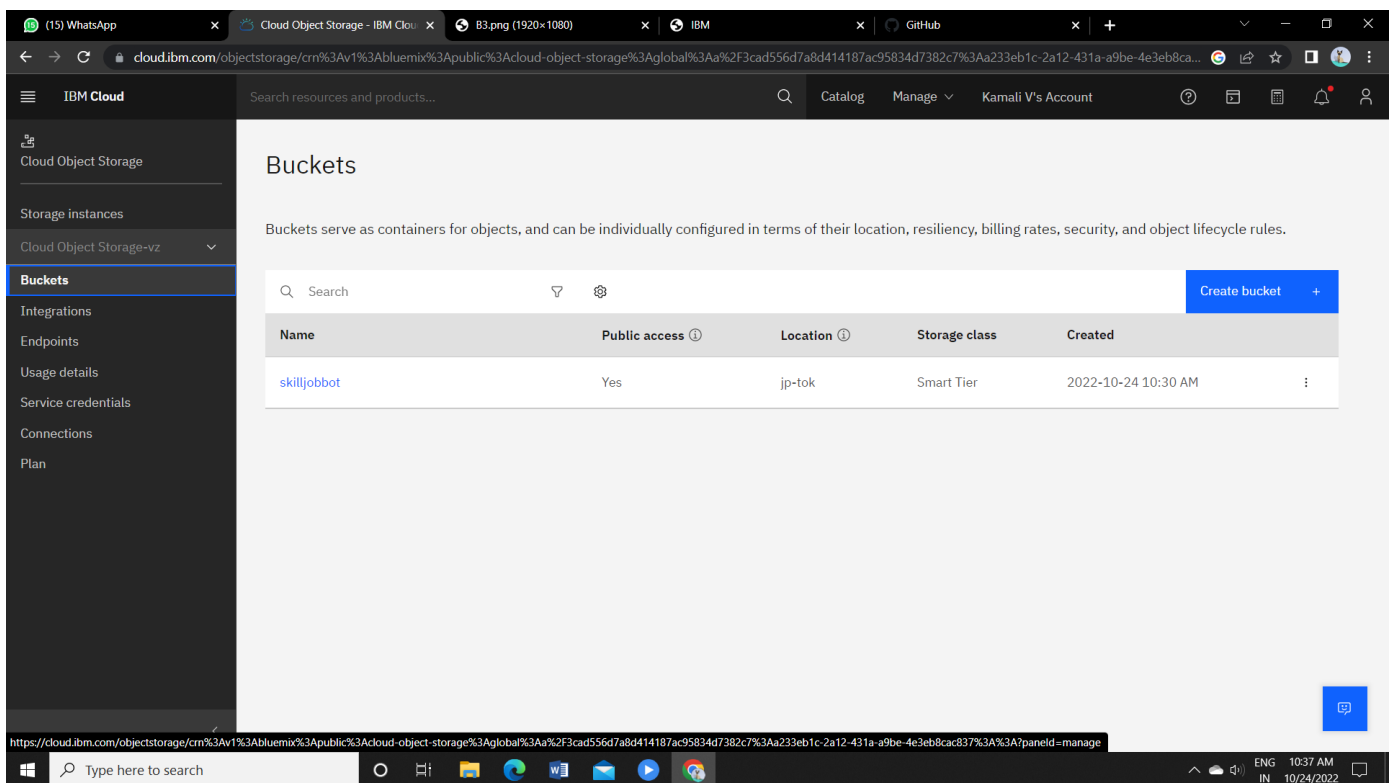


Assignment-3

Date	10 October 2022
Name	Kamali.V
Team ID	PNT2022TMID30531
Project Name	Skill and Job Recommender Application

1. CREATE A BUCKET IN IBM OBJECT STORAGE.



The screenshot shows the IBM Cloud Object Storage interface. The left sidebar contains navigation links: Cloud Object Storage, Storage instances, Cloud Object Storage-vz, Buckets (selected), Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main content area is titled 'Buckets' and includes a search bar, a 'Create bucket' button, and a table of existing buckets.

Name	Public access	Location	Storage class	Created
skilljobbot	Yes	jp-tok	Smart Tier	2022-10-24 10:30 AM

**Upload an 5 images to ibm object storage and make it public.
Write html code todisplaying all the 5 images.**

IBM Cloud

Search resources and products...

Storage / Cloud Object Storage-vz / skilljobbot

Transfers Details Actions...

Objects Configuration Permissions

If you're seeing more usage than expected, versions count towards your usage or you may have incomplete uploads [Learn more](#)

Prefix filter

Upload

Object name	Archived ⓘ	Size	Last modified
B1....		152.7 KB	2022-10-24 10:32 AM
B2....		192.2 KB	2022-10-24 10:32 AM
B3ng		190.3 KB	2022-10-24 10:34 AM
B5ng		133.5 KB	2022-10-24 10:34 AM
B6ng		2.4 MB	2022-10-24 10:35 AM

Drag and drop files (objects) here or click to upload

IBM Cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-vz

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Type here to search

ENG 1035 AM IN 10/24/2022

IBM Cloud

Search resources and products...

Bucket access policies

Manage access to this bucket by creating IAM policies for users and service IDs. Users and service IDs must also have an instance level viewer role (or higher) to use the console or to list buckets using the REST API.

Access policies

Public access

Access policy update

Access group policy created

A new access policy for this bucket was created for the group: Public Access

Status: Enabled

Role for this policy: Content Reader

To delete/edit go to the [IAM console](#).

Create access policy

Context-based restrictions

Firewall (legacy)

IBM Cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-vz

Buckets

Integrations

Endpoints

Usage details

Service credentials

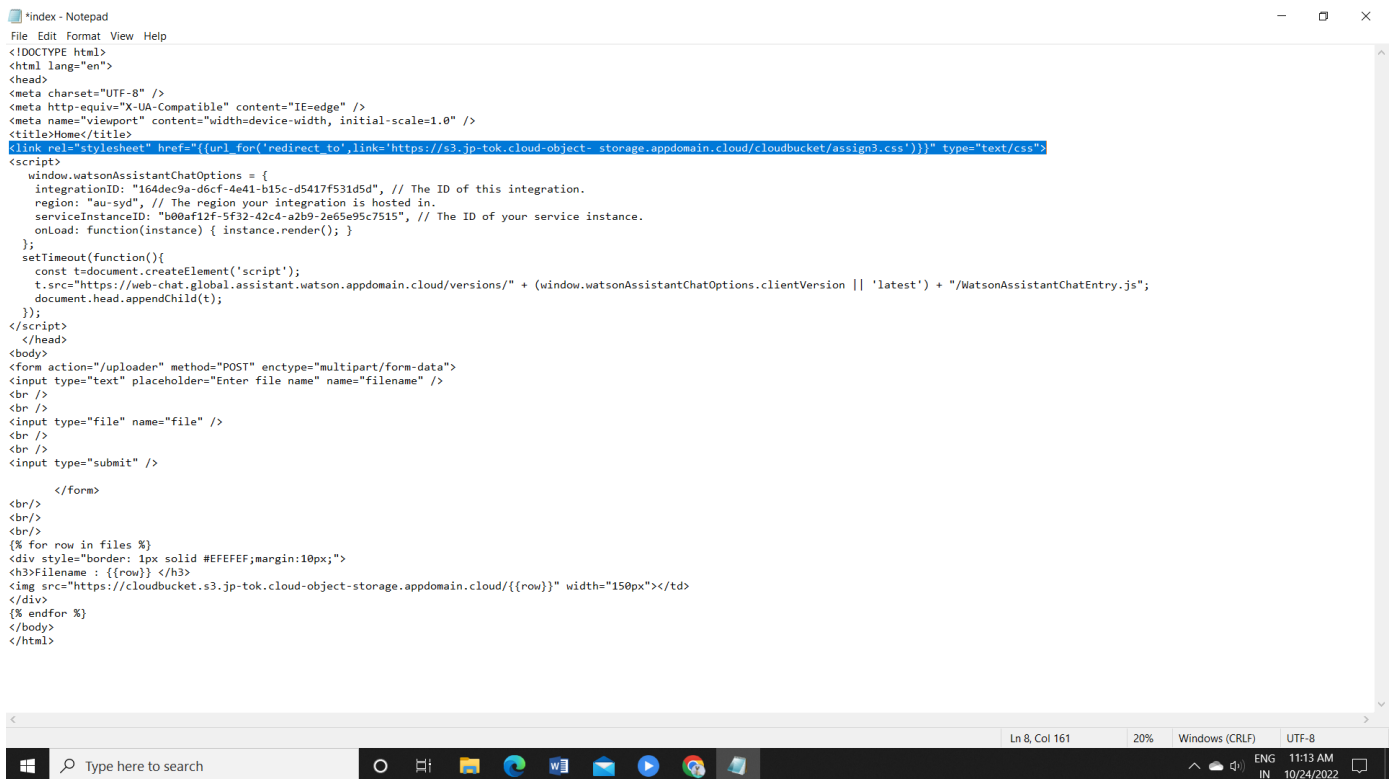
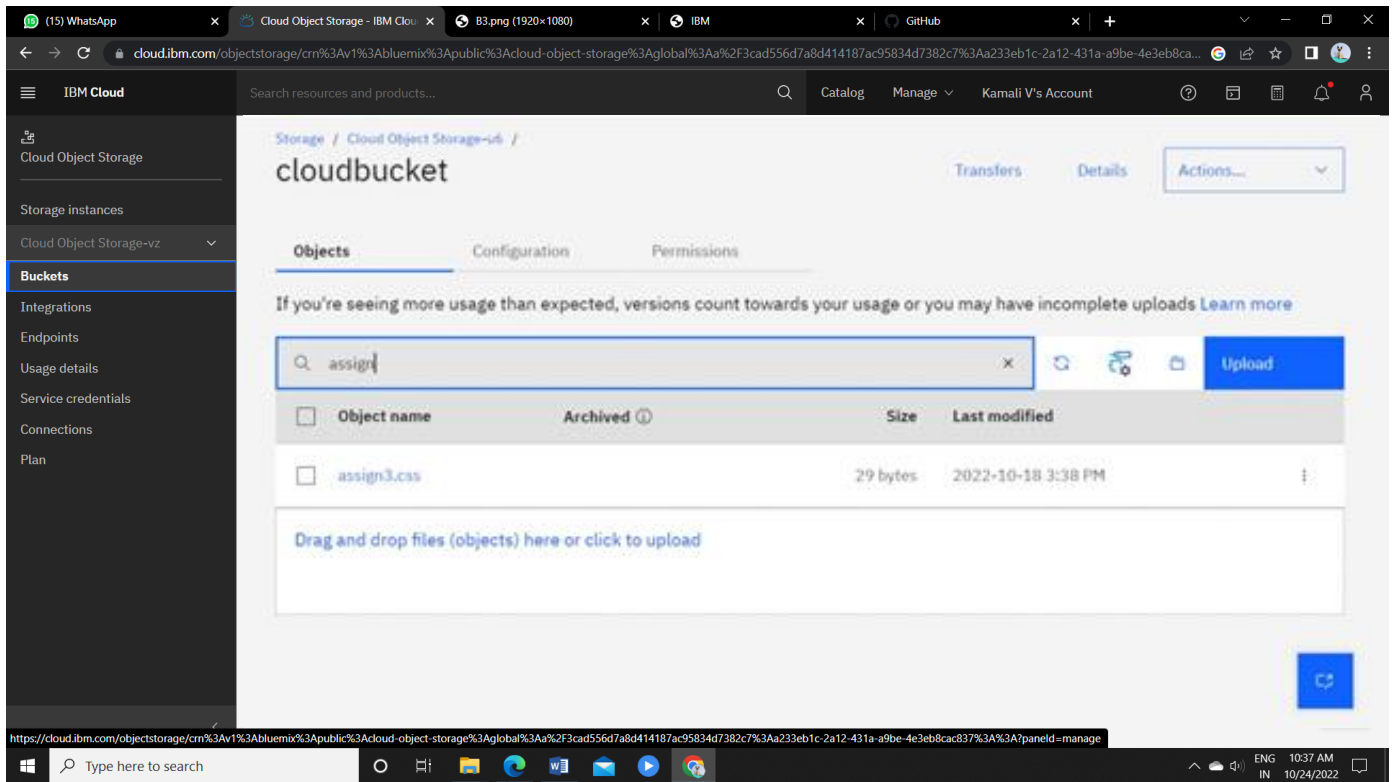
Connections

Plan

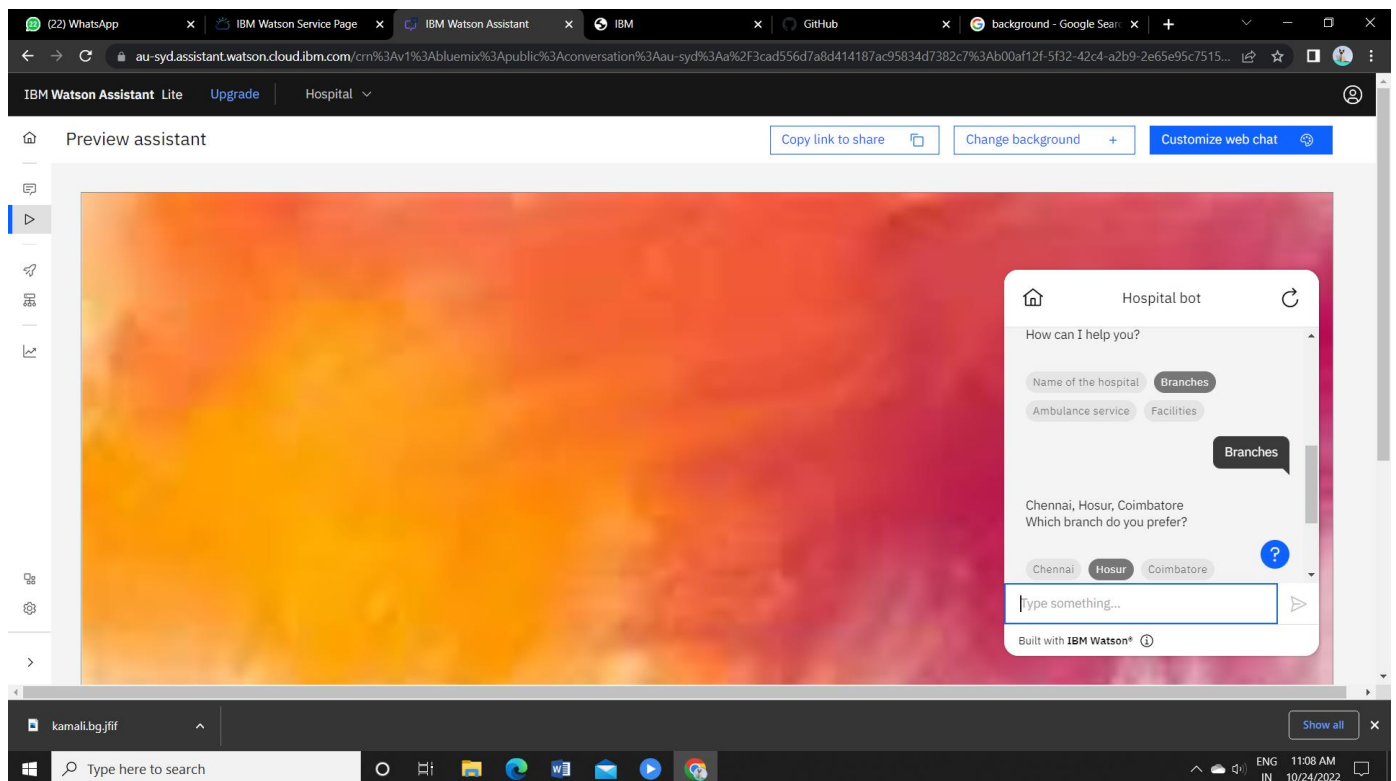
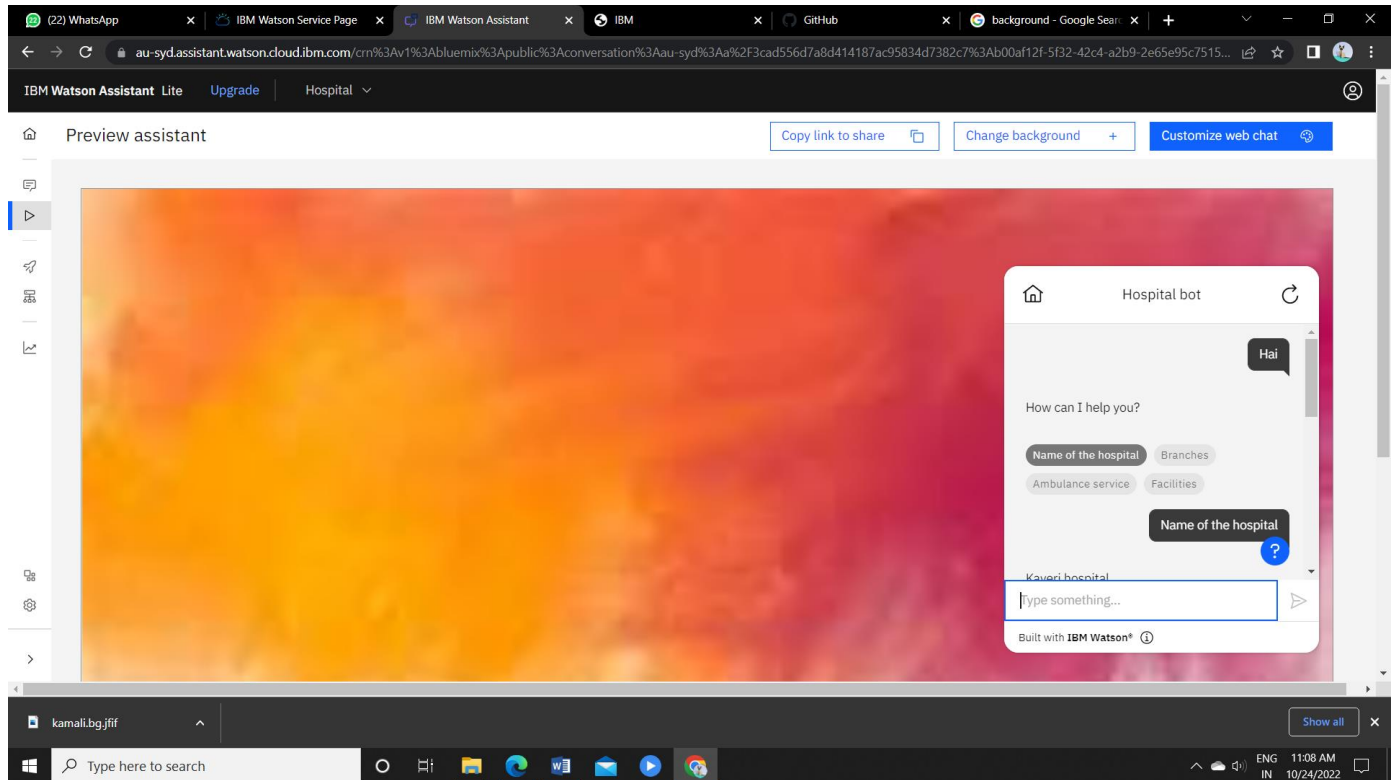
Type here to search

ENG 1037 AM IN 10/24/2022

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



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%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-b00af12f-5f32-42c4-a2b9-2e65e95c7515%3A%3A8be13166-e79a-4347-85c7-200df25a7f18&integrationID=164dec9a-d6cf-4e41-b15c-d5417f531d5d®ion=au-syd&serviceInstanceID=b00af12f-5f32-42c4-a2b9-2e65e95c7515>

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

The screenshot displays the IBM Watson Assistant Lite interface. On the left, a vertical list of 10 steps is shown, with the 10th step, 'Others', selected. The main area on the right shows the assistant's response to the selected step: 'Kindly mention your preferred specialist.' Below this, a text input field is visible with the placeholder 'User enters free text'. The interface includes a 'Preview' button at the bottom right and a 'New step' button at the bottom left. The top of the browser window shows several open tabs, including 'WhatsApp', 'IBM Watson Service Page', 'IBM Watson Assistant', 'B3.png', 'IBM', and 'GitHub'.

Included 3 conditions in steps:

This screenshot shows the IBM Watson Assistant configuration interface. On the left, the 'Conversation steps' panel displays a sequence of steps. Step 1 asks 'How can I help you?' with a 'Name of the ...' input and a '+2' branch. Step 2 is highlighted, showing a condition '1 is Name of the hospital' and a response 'Kaveri hospital'. Step 3 asks 'Chennai, Hosur, Coimbatore Which branch do you prefer?' with 'Chennai' and 'Hosur' inputs and a '+1' branch. The main panel shows 'Step 2 is taken' with a condition '1. How can I help you... is Name of the hospital'. The 'Assistant says' panel shows the response 'Kaveri hospital'. A 'Preview' button is visible at the bottom right.

This screenshot shows the IBM Watson Assistant configuration interface for Step 3. The 'Conversation steps' panel on the left shows Step 3 highlighted, with a condition '1 is Branches' and a response 'Chennai, Hosur, Coimbatore Which branch do you prefer?'. The main panel shows 'Step 3 is taken' with a condition '1. How can I help you... is Branches'. The 'Assistant says' panel shows the response 'Chennai, Hosur, Coimbatore Which branch do you prefer?'. A 'Preview' button is visible at the bottom right.

IBM Watson Assistant LiteUpgradeHospital

Hai

What kind of facilities do you need?

5

Bed availabilityParking faci...+ 1

Continue to next step

5 is Ambulance services

6

Available 24/7

Continue to next step

5 is Bed availability

7

91 beds are available. Do you need AC beds?

Confirmation

Re-ask previous step(s)

5 is Parking facilities

8

We have wallet parking.

Free text

Continue to next step

New step +

Step 7 is takenwith conditions

Conditions1 condition

If All of this is true:

5. What kind of faci... is Bed availability

and Add condition +

New condition group +

Assistant says

91 beds are available. Do you need AC beds?

YesNo

Preview

Type here to search

11:05 AM10/24/2022

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: "164dec9a-d6cf-4e41-b15c-d5417f531d5d", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "b00af12f-5f32-42c4-a2b9-2e65e95c7515", // 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 />

      <br />

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

      <br />

      <br />

      <input type="submit" />

    </form>

  </body>

</html>
```



```

</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)
```