

## Assignment-3

Date	10 October 2022
Team ID	PNT2022TMID30665
Project Name	News Tracker Application.

### 1. CREATE A BUCKET IN IBM OBJECT STORAGE.

The screenshot shows the IBM Cloud Object Storage console. The left sidebar contains a navigation menu with options: Cloud Object Storage, Storage instances, Cloud Object Storage-ig, 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
newbot	Yes	jp-tok	Smart Tier	2022-11-06 8:11 PM

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

Cloud Object Storage - IBM Cloud

new cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-ig

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Search resources and products...

Catalog

Manage

Vimala M's Account

Storage / Cloud Object Storage-ig /

newbot

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

<input type="checkbox"/>	Object name	Archived	Size	Last modified	
<input type="checkbox"/>	t... pg		26.1 KB	2022-11-06 8:12 PM	
<input type="checkbox"/>	t... pg		20.6 KB	2022-11-06 8:12 PM	
<input type="checkbox"/>	t... pg		15.8 KB	2022-11-06 8:12 PM	
<input type="checkbox"/>	t... pg		16.3 KB	2022-11-06 8:12 PM	
<input type="checkbox"/>	t... pg		17.3 KB	2022-11-06 8:13 PM	

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

26°C Cloudy

20:13 06-11-2022

Cloud Object Storage - IBM Cloud

new cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-ig

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Search resources and products...

Catalog

Manage

Vimala M's Account

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

Status: Enabled

Role for the group: Content Reader

As a Content Reader, one can read and list objects in the bucket.

Create access policy

Context-based restrictions

Firewall (legacy)

Access policy update

Access group policy created

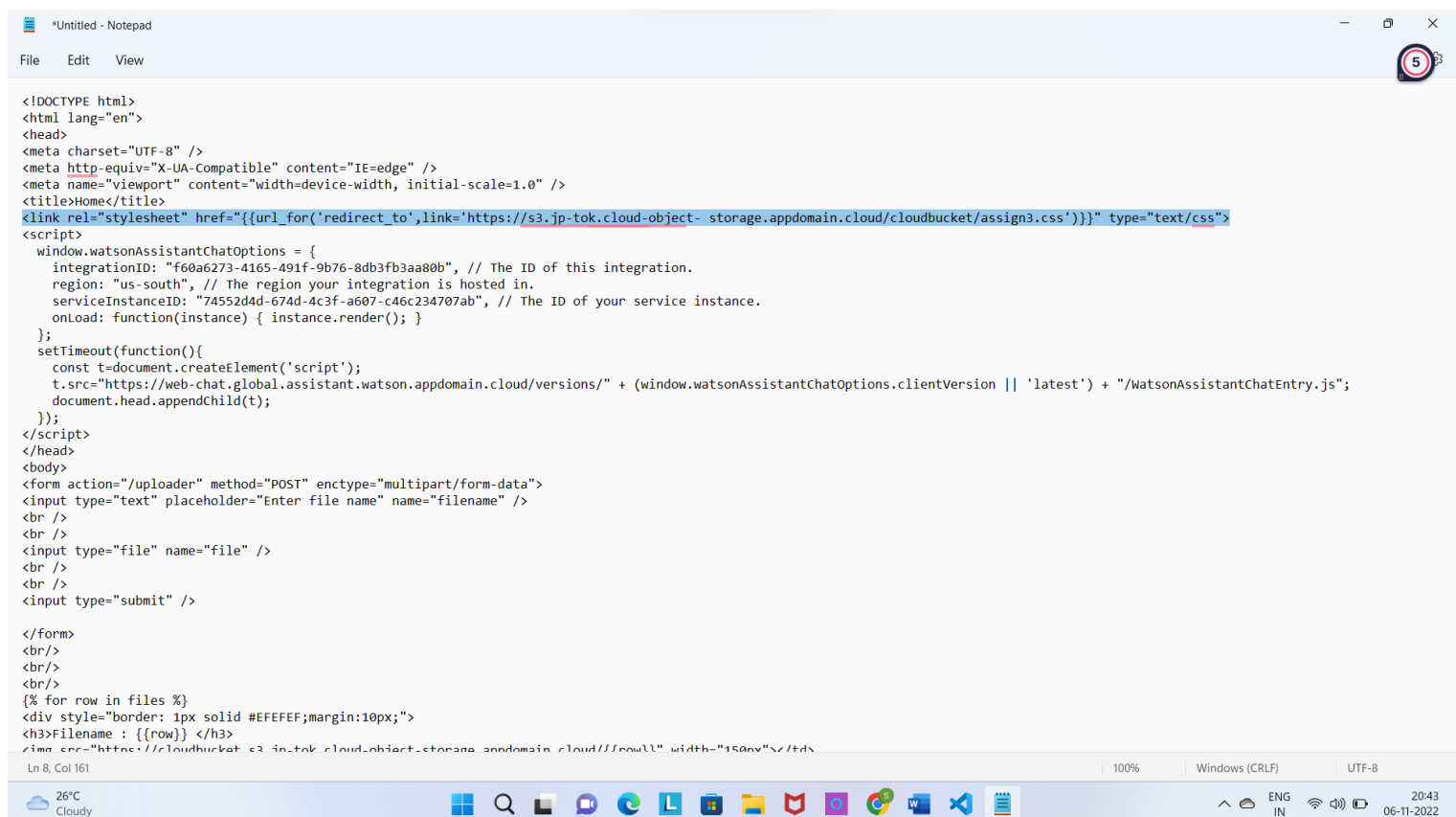
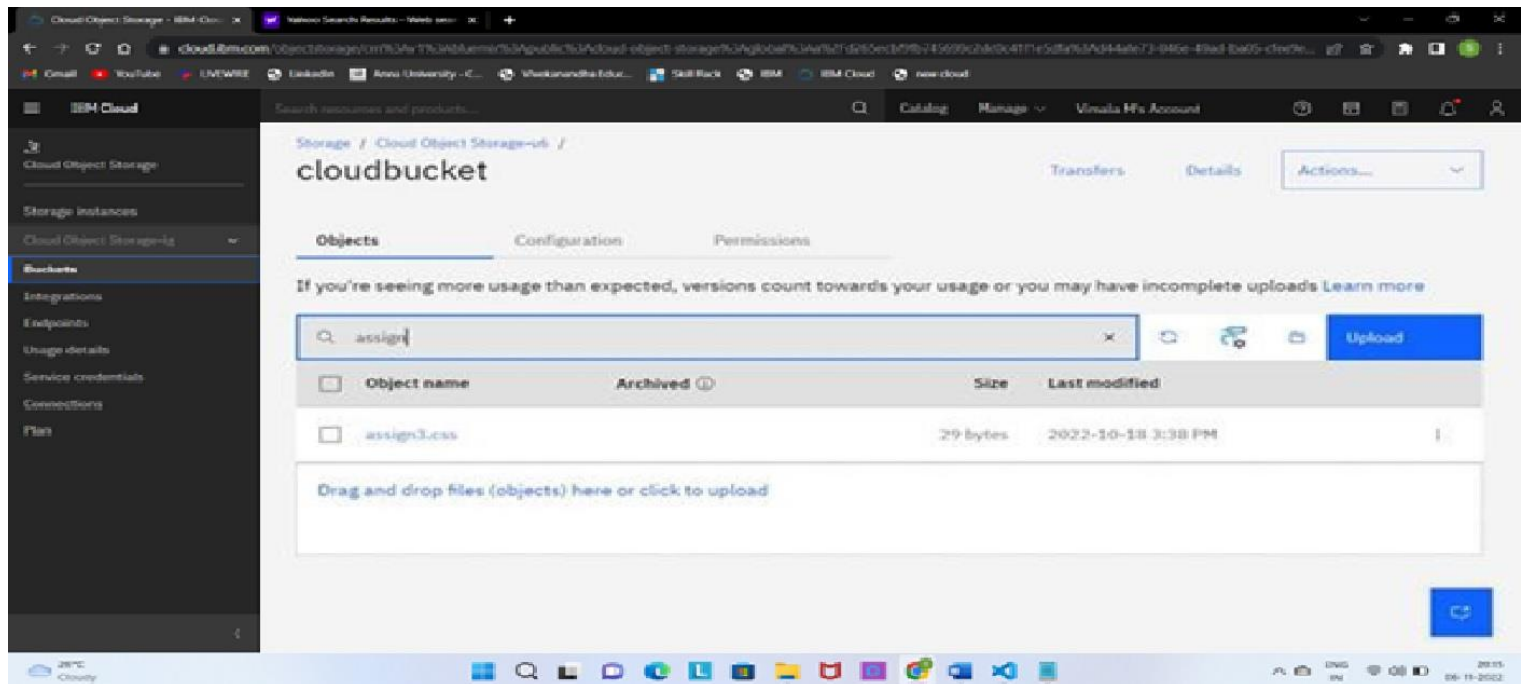
A new access policy for this bucket was created for the group: Public Access

To delete/edit go to the IAM console.

26°C Cloudy

20:15 06-11-2022

## 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.

IBM Watson Assistant Lite Upgrade Hospital bot Learning center

Preview assistant

Copy link to share Change background Customize web chat

Watson Assistant

How can I help you?

Branches Visitor time Specialist

Visitor time

4.00 pm to 6.00 pm

How can I help you?

Branches Visitor time Specialist

Type something...

Built with IBM Watson®

26°C Cloudy

ENG IN 20:37 06-11-2022

IBM Watson Assistant Lite Upgrade Hospital bot Learning center

Preview assistant

Copy link to share Change background Customize web chat

Watson Assistant

4.00 pm to 6.00 pm

How can I help you?

Branches Visitor time Specialist

Specialist

ENT Pediatrician Orthologist

General

Type something...

Built with IBM Watson®

26°C Cloudy

ENG IN 20:37 06-11-2022

## 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-74552d4d-674d-4c3f-a607-c46c234707ab%3A%3Af3eb4fa9-52e0-476b-810a-1e6e37a4d84c&integrationID=f60a6273-4165-491f-9b76-8db3fb3aa80b&region=us-south&serviceInstanceID=74552d4d-674d-4c3f-a607-c46c234707ab>

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

The screenshot displays the IBM Watson Assistant console interface. The top navigation bar includes 'IBM Watson Assistant', 'Upgrade', and 'Hospital bot'. The main workspace shows a script with 10 steps. Step 10 is highlighted, showing the action 'Do you want to go for Pharmacy?' with a 'Free text' input type. The script is configured with 'without conditions' and 'End the action'. A preview window on the right shows the chat flow: 'Greet customer [default]' with the message 'Welcome, how can I assist you?'. The user input 'Hey' is recognized, and the assistant responds with 'How can I help you?'. The preview window also shows buttons for 'Branches', 'Visitor time', and 'Specialist'. The bottom status bar indicates '26°C Cloudy' and the time '20:35 06-11-2022'.

## Included 3 conditions in steps:

The screenshot displays the IBM Watson Assistant configuration interface. On the left, a conversation flow diagram shows steps 5 through 10. Step 5 is highlighted, showing a condition 'is Female' and a response 'Dr.sha'. Step 7 is also highlighted, showing a condition 'Continue to next step'. The main panel on the right shows the configuration for Step 7, which is titled 'Step 7 is taken with conditions'. It lists one condition: '5. Which doctor wo... is Female'. The 'Assistant says' section shows the response 'Dr.sha'. A 'Preview' button is visible at the bottom right.

The screenshot displays the IBM Watson Assistant configuration interface. On the left, a conversation flow diagram shows steps 4 through 8. Step 4 is highlighted, showing a condition 'ENT Pediatrician +2' and a response 'Continue to next step'. Step 5 is also highlighted, showing a condition 'Male Female' and a response 'Continue to next step'. Step 6 is highlighted, showing a condition 'is Male' and a response 'Dr.Raj'. The main panel on the right shows the configuration for Step 6, which is titled 'Step 6 is taken with conditions'. It lists one condition: '5. Which doctor wo... is Male'. The 'Assistant says' section shows the response 'Dr.Raj'. A 'Preview' button is visible at the bottom right.

Hey

Continue to next step

1 is Branches

This step has no content

2

Chennai Ariyalur + 1

Re-ask previous step(s)

1 is Visitor time

3

4.00 pm to 6.00 pm

Re-ask previous step(s)

1 is Specialist

This step has no content

4

ENT Pediatrician + 2

Continue to next step

Which doctor would I want to consult?

Male Female

New step +

Step 3 is taken with conditions

fx

Conditions

1 condition

If All of this is true:

1. How can I help yo... is Visitor time

and Add condition +

New condition group +

Assistant says

B I [icons] </>

4.00 pm to 6.00 pm

Preview

Define customer response

## 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: "f60a6273-4165-491f-9b76-8db3fb3aa80b", // The ID of this integration.
  region: "us-south", // The region your integration is hosted in.
  serviceInstanceID: "74552d4d-674d-4c3f-a607-c46c234707ab", // 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>
<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(files)
```

```
    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)
```