

## Assignment-3

Date	10 October 2022
Team ID	PNT2022TMID30533
Project Name	CONTAINMENT ZONE ALERTING 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-cy, Buckets (selected), Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main content area is titled 'Buckets' and includes a description: 'Buckets serve as containers for objects, and can be individually configured in terms of their location, resiliency, billing rates, security, and object lifecycle rules.' Below this is a table of buckets. The table has columns: Name, Public access, Location, Storage class, and Created. A single bucket is listed with the name 'containmentzonealerting', public access 'Yes', location 'jp-tok', storage class 'Smart Tier', and creation time '2022-11-07 10:29 AM'. A 'Create bucket' button is visible in the top right of the bucket list area. The bottom of the image shows the Windows taskbar with the time 10:43 AM on 07-Nov-22.

Name	Public access	Location	Storage class	Created
containmentzonealerting	Yes	jp-tok	Smart Tier	2022-11-07 10:29 AM

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

WhatsApp IBM Cloud Object Storage - IBM Cloud

cloud.ibm.com/objectstorage/crn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F1dfe833178894e069...

IBM Cloud

Storage / Cloud Object Storage-cy /

containmentzonealerting

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
im... .png		71.9 KB	2022-11-07 10:30 AM

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

Activate Windows  
Go to Settings to activate Windows

Search Windows

10:33 AM  
07-Nov-22

WhatsApp IBM Cloud Object Storage - IBM Cloud

cloud.ibm.com/objectstorage/crn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F1dfe833178894e069...

IBM Cloud

Access policies

Public access

Warning: Granting Public access to this bucket will allow anyone to access the bucket. To revoke public access, remove

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)

Activate Windows  
Go to Settings to activate Windows

Search Windows

10:41 AM  
07-Nov-22

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

IBM Cloud

Cloud Object Storage

Storage Instances

Cloud Object Storage-cy

**Buckets**

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

### Buckets

Buckets serve as containers for objects, and can be individually configured in terms of their location, resiliency, billing rates, security, and object lifecycle rules.

Search

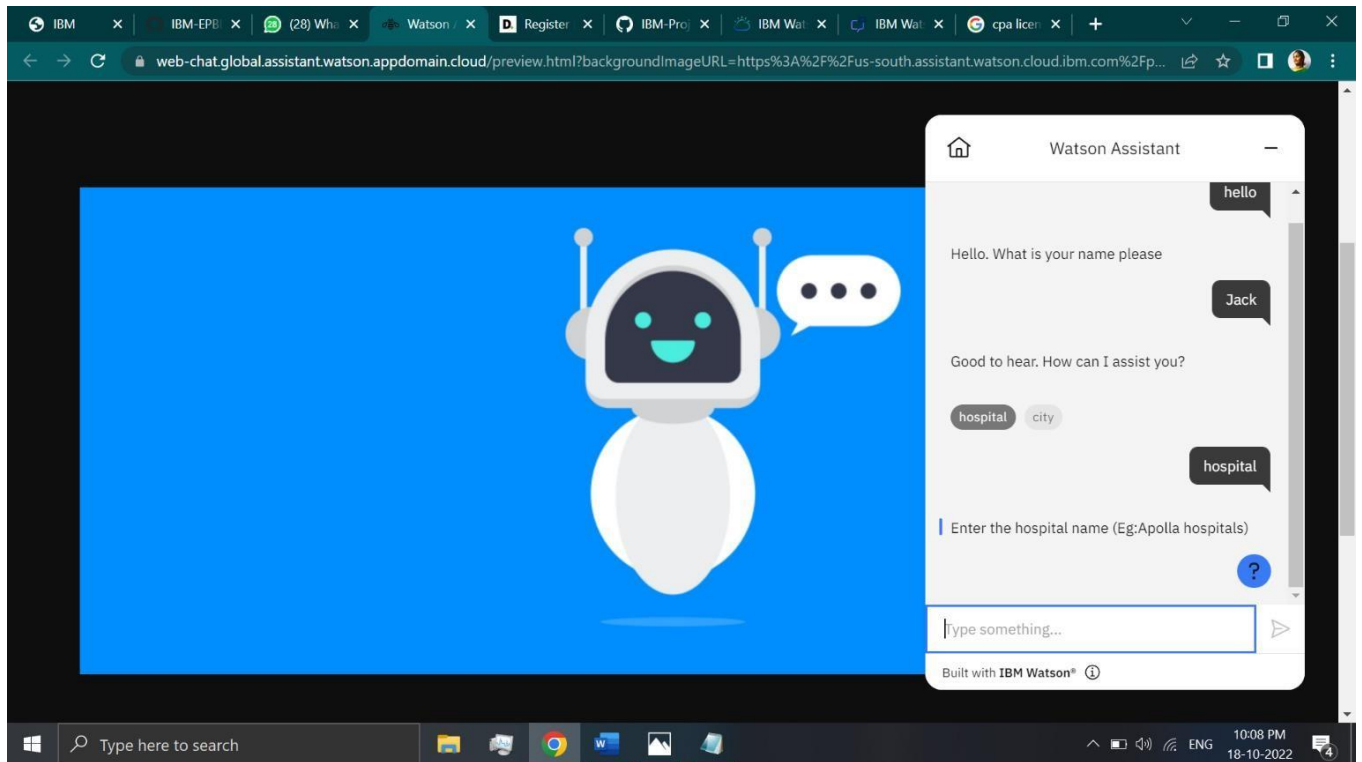
Create bucket +

Name	Public access ⓘ	Location ⓘ	Storage class	Created
<a href="#">containmentzonealerting</a>	Yes	jp-tok	Smart Tier	2022-11-07 10:29 AM

Activate Windows  
Go to Settings to activate Windows.

```
index.html - 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">
</head>
<script>
window.watsonAssistantChatOptions = {
  integrationID: "14b83b8f-3dfd-405f-9520-b550092892aa", // The ID of this integration.
  region: "us-south", // The region your integration is hosted in.
  serviceInstanceID: "6e95bee9-8d0b-49f6-8a2f-4125fb3a7945", // 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">
```

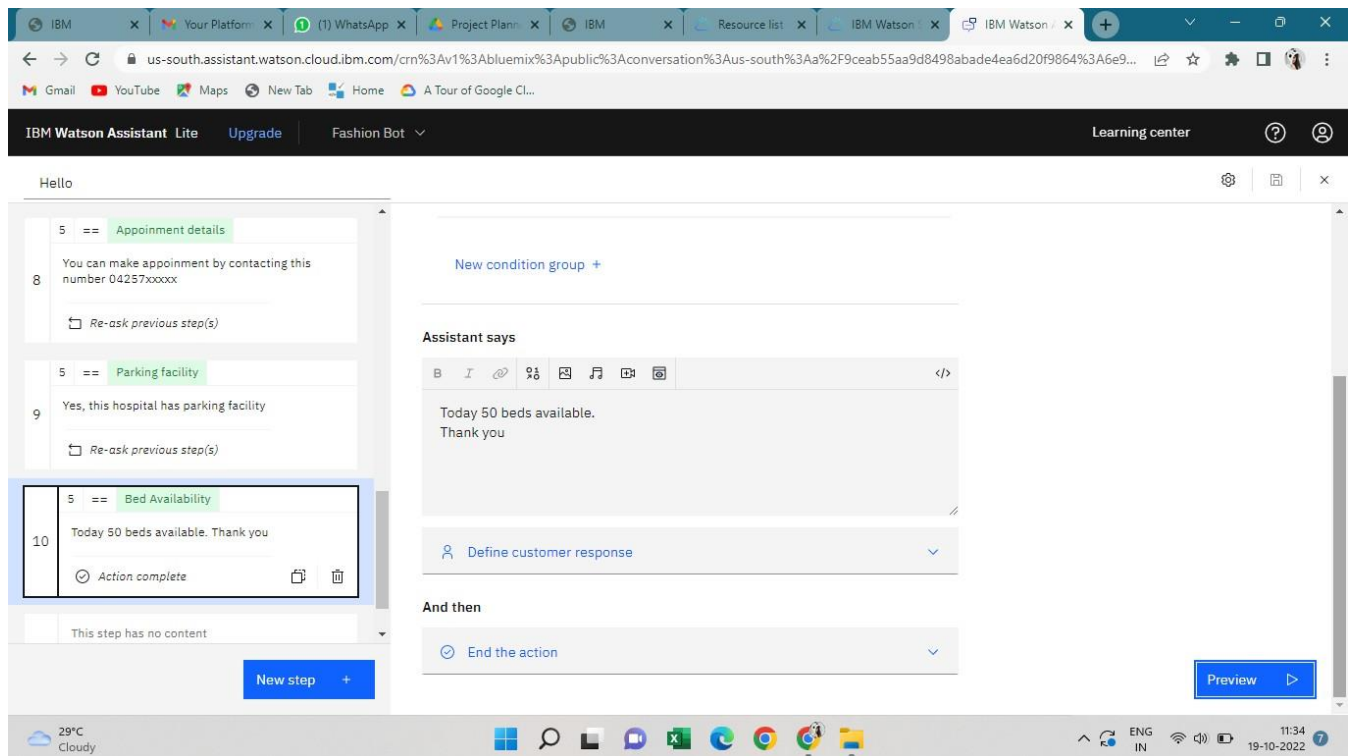
### 3. Design a chatbot using IBM Watson assistant for hospital.



## 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-6e95bee9-8d0b-49f6-8a2f-4125fb3a7945%3A%3Ab437fcda-6135-46e5-9e62-faa901cdce2d&integrationID=14b83b8f-3dfd-405f-9520-b550092892aa&region=us-south&serviceInstanceID=6e95bee9-8d0b-49f6-8a2f-4125fb3a7945>

## 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 displays the IBM Watson Assistant Lite interface. On the left, a conversation flow is visible with steps 1 through 4. Step 3 is highlighted, showing a text input field with the placeholder "Enter the hospital name (Eg: Apollo hospitals)" and a "Free text" button. The right panel shows the configuration for Step 3, which is set to "with conditions". Under the "Conditions" section, two conditions are listed: "2. Good to he... is hospital" and "2. Good to he... is city". The "Assistant says" section shows the output text "Enter the hospital name (Eg: Apollo hospitals)".

The screenshot displays the IBM Watson Assistant Lite interface. On the left, a conversation flow is visible with steps 4 through 6. Step 5 is highlighted, showing a text input field with the placeholder "Hospital is 30km away from your location" and a "Continue to next step" button. The right panel shows the configuration for Step 5, which is set to "with conditions". Under the "Conditions" section, one condition is listed: "4. Here are th... is Distance". The "Assistant says" section shows the output text "Hospital is 30km away from your location".

Hello



4 == Appointment details

Step 8 is taken with conditions

f<sub>1</sub>

7 number 04257xxxx

If of this is true:

4. Here are th... is Parking facility

X

and Add condition +

Newvcondi\$on group +

Thank you.

Assistant says

B I @ % & | | | | | </>

New step +

Yes, this hospital has parking facility

Preview



29°C Cloudy



ENG IN 11:16 19-10-2022

## 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: "14b83b8f-3dfd-405f-9520-b550092892aa", // The ID of this integration.

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

        serviceInstanceID: "6e95bee9-8d0b-49f6-8a2f-4125fb3a7945", // 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']
```

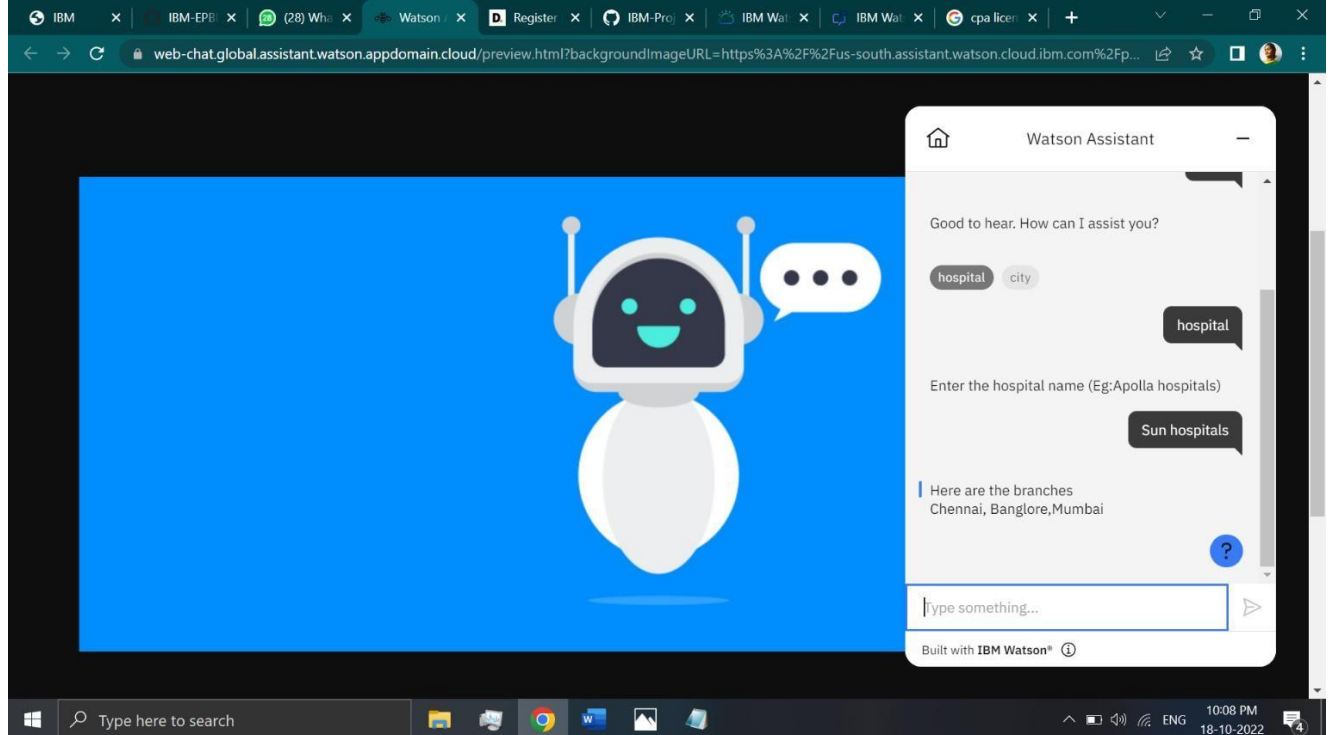
```
try:
```

```
    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)
```



