

## Assignment-3

|             |                                       |
|-------------|---------------------------------------|
| Date        | 10October2022                         |
| TeamID      | PNT2022TMID30621                      |
| ProjectName | SkillandJobRecommenderA<br>pplication |

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

Cloud Object Storage - IBM Cloud

Search resources and products...

Catalog Manage kowsika k's Account

Cloud Object Storage

Storage instances

Cloud Object Storage-dw

**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.

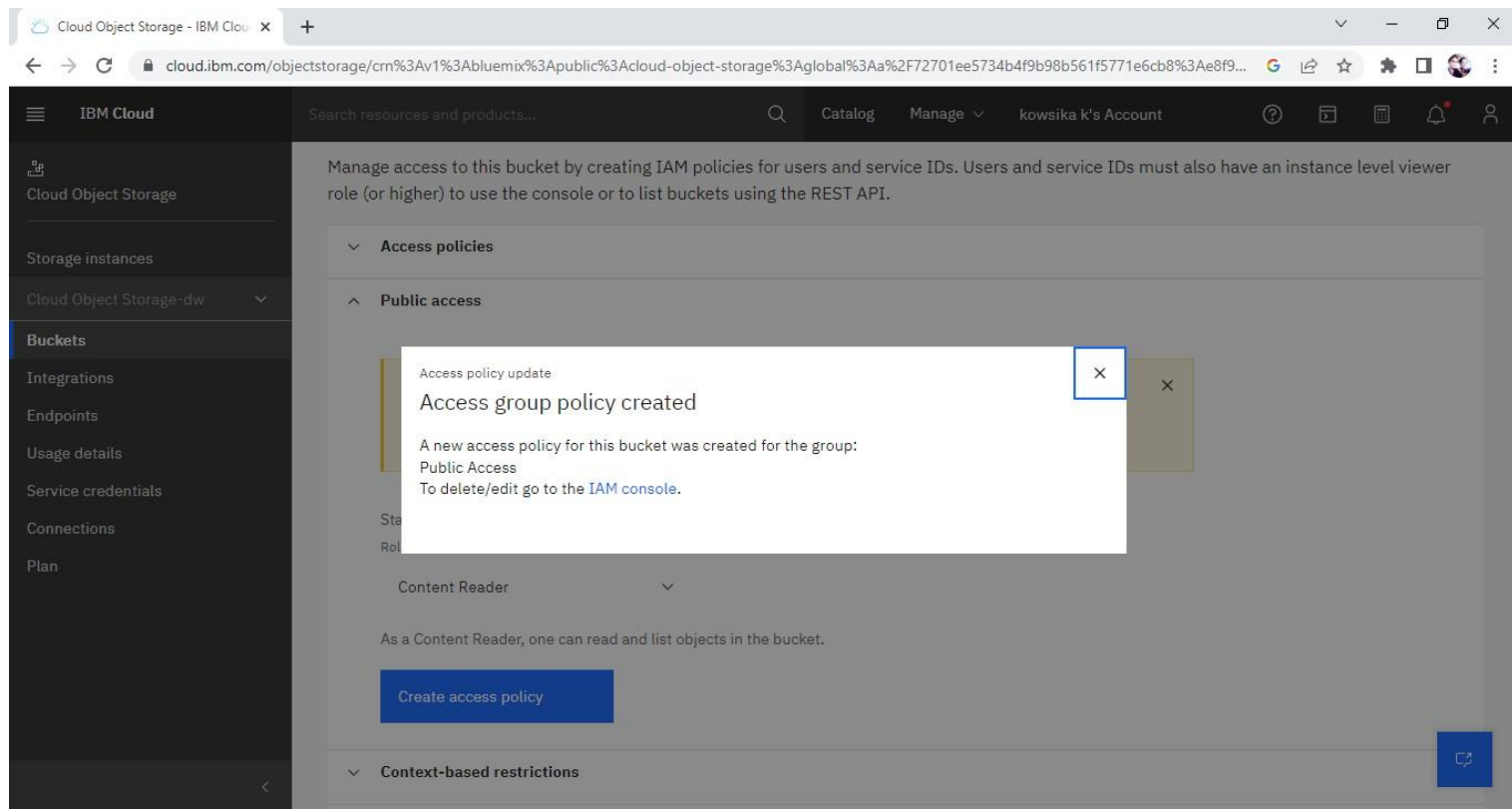
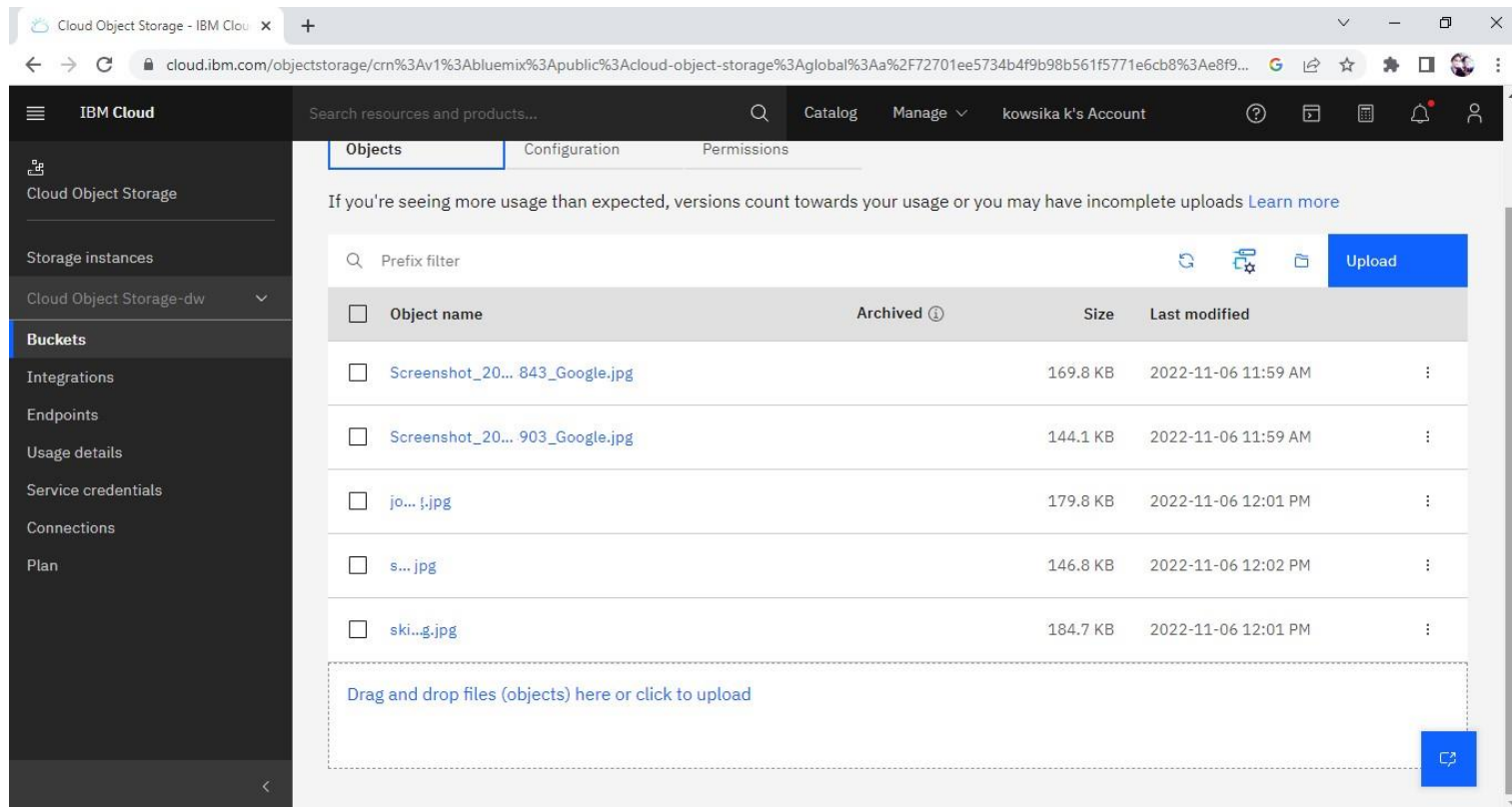
Search

Create bucket +

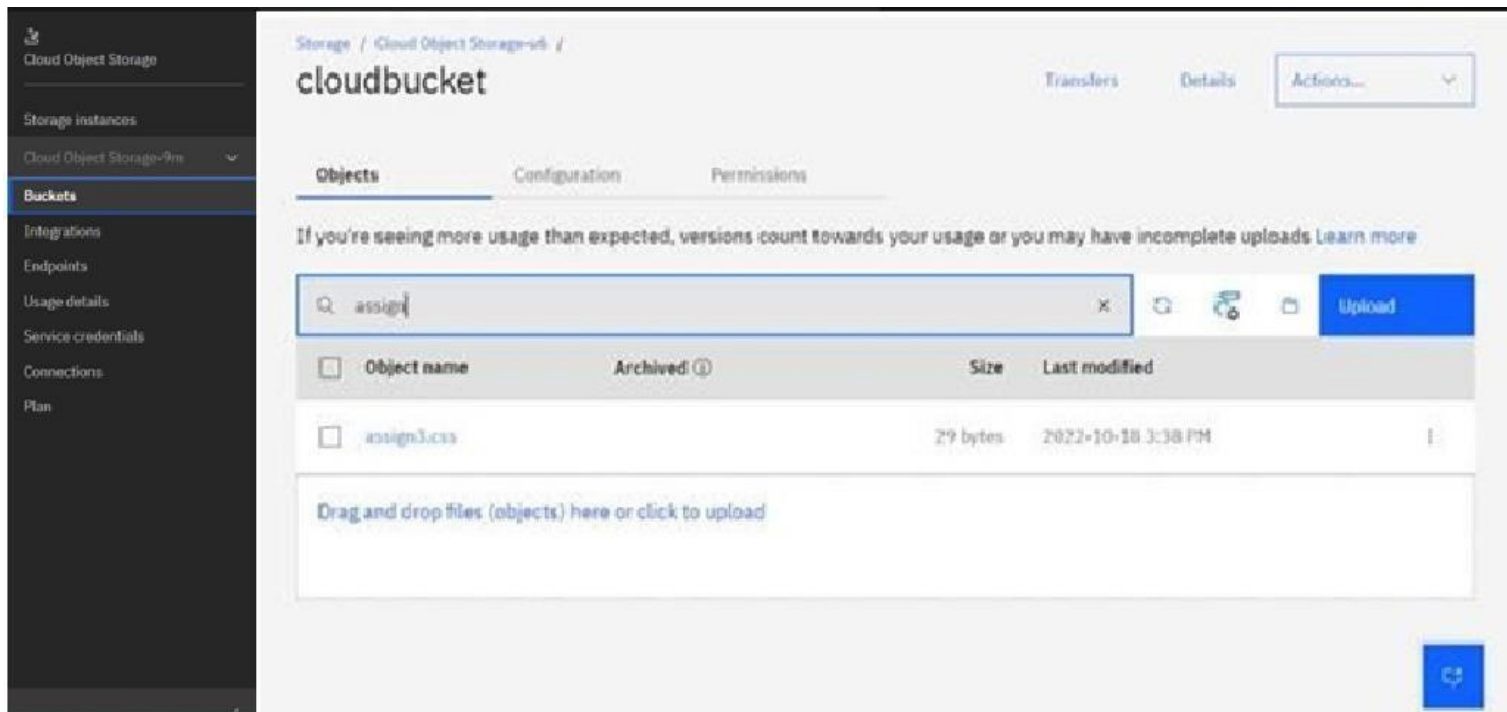
| Name      | Public access ⓘ | Location ⓘ | Storage class | Created             |
|-----------|-----------------|------------|---------------|---------------------|
| quick.123 | Yes             | jp-tok     | Smart Tier    | 2022-11-06 11:43 AM |

https://cloud.ibm.com/objectstorage/cn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F72701ee5734b4f9b98b561f5771e6cb8%3Ae8f928ba-d9e5-42ed-a074-e80ae3ee077f%3A7panelId=manage

**Upload an 5 images to ibm object storage and make it public. Writehtmlcodetodisplayingall the5images.**



2. Upload a css page to the object storage and use the same page in yourHTMLcode.



```

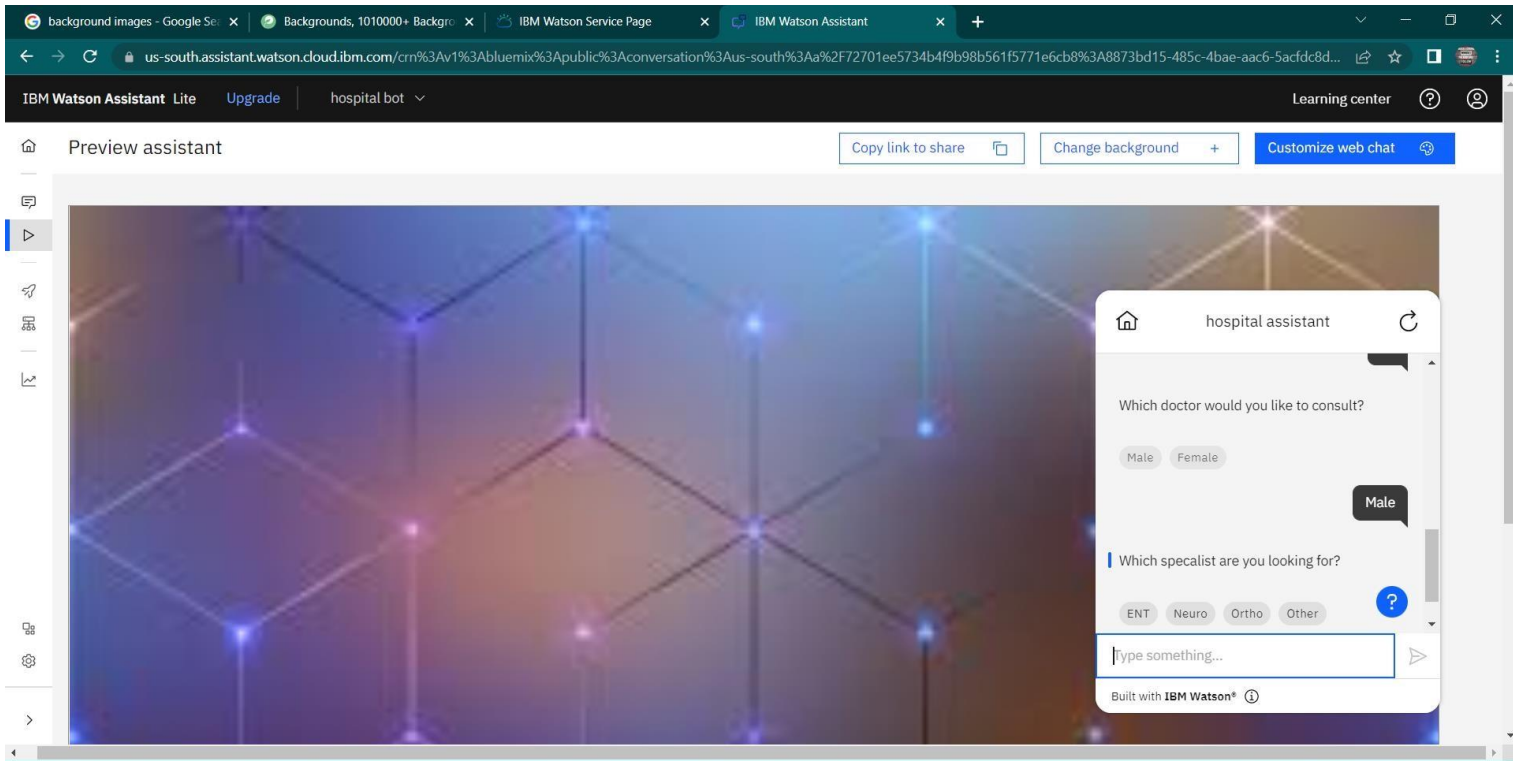
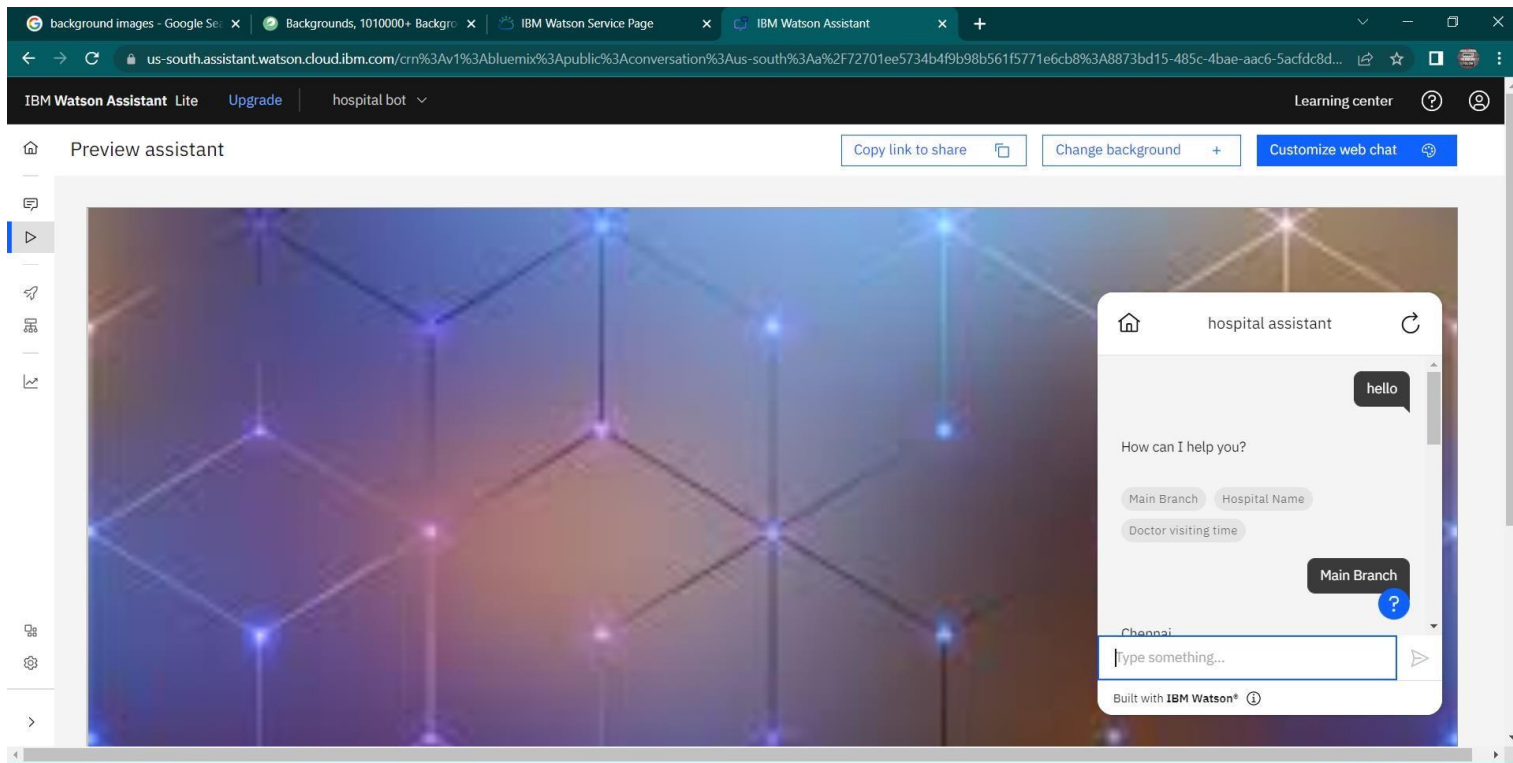
*Untitled - Notepad
File Edit Format View Help
<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8"/>
<metahttp-equiv="X-UA-Compatible"content="IE=edge"/>
<metaname="viewport"content="width=device-width,initial-scale=1.0"/>
<titleHome</title>
<linkrel="stylesheet" href="{url_for('redirect to',link='https://s3.jp-tok.cloud-object-storage.appdomain.cloud/cloudbucket/assign3.css')}}"type="text/css">
<script>
window.watsonAssistantChatOptions = {
  integrationID: "2468cc28-3ae1-4cb9-a012-3c197820ab0f", // The ID of this integration.
  region: "us-south", // The region your integration is hosted in.
  serviceInstanceID: "8873bd15-485c-4bae-aac6-5acfdc8d28a7", // 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>
<formaction="/uploader"method="POST"enctype="multipart/form-data">
<inputtype="text"placeholder="Enterfilename"name="filename"/>
<br/>
<inputtype="file"name="file"/>
<br/>
<br/>
<inputtype="submit"/>

</form>
<br/>
<br/>
<br/>
{%-forrowinfiles %}
<divstyle="border:1pxsolid#EFEFEF;margin:10px;">
<h3>Filename:{{row}}</h3>
</td>
</div>

```

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



### WebURL forAssistant:

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fussouth.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-7df6f34f-0d17-4036-a906-b500d7c4b278%3A%3A8a7038d1-5e0c-4f5f-b1eb-725f3b78e5de&integrationID=3ed15b4c-58ce-4a91-837f4e050b7c23fb&region=us-south&serviceInstanceID=7df6f34f-0d17-4036-a906-b500d7c4b278>

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

The screenshot displays the IBM Watson Assistant configuration interface in a web browser. The browser's address bar shows the URL: `us-south.assistant.watson.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Aconversation%3Aus-south%3Aa%2Fd09ee29f829344568851734f3a63206d%3Abb064674-31ff-4033-83f7-67e4d865...`. The interface is titled "IBM Watson Assistant Lite" and includes a "Hospital bot" dropdown menu. The main content area is divided into two panels. The left panel shows a flowchart with 10 steps. Step 5 is "Male", Step 6 is "Which specialist are you looking for?" with options "Cardiologist", "Ortho", and "+ 2", and a "Re-ask previous step(s)" button. Step 7 is "Please mention your preferred specialist!" with a "Free text" input field. Step 8 is "At what time would you like to visit the doctor?" with a "Time" input field. Step 9 is "We will let you know your appointment details in few minutes." The right panel shows the configuration for Step 7, which is "with conditions". It displays a single condition: "If All of this is true: 6. Which specialist ... is Other". Below the conditions, there is a "New condition group +" button. The "Assistant says" section shows the text "Please mention your preferred specialist!" with a red underline under "preferred". A "Preview" button is located at the bottom right of the interface.

## Included 3 conditions in steps:

The screenshot displays the IBM Watson Assistant configuration interface. On the left, a 'Conversation steps' panel shows a sequence of steps. Step 2 is highlighted with a blue border and contains the text 'ABC Hospital' and a 'Free text' input field. Below it, Step 3 is visible with the text 'Chennai' and another 'Free text' input field. Step 4 is also visible with the text 'Between 10am to 4pm.' and a 'Free text' input field. A 'New step +' button is at the bottom of the steps list.

The main area shows the configuration for 'Step 2 is taken'. It is set to 'with conditions'. The 'Conditions' section shows a single condition: 'If All of this is true: 1. How can I help yo... is Hospital name'. Below this, there is an 'Assistant says' section with a text area containing 'ABC Hospital' and a 'Preview' button.

The screenshot displays the IBM Watson Assistant configuration interface for a later step. On the left, the 'Conversation steps' panel shows Step 7 highlighted with a blue border. It contains the text 'Please mention your preferred specialist!' and a 'Free text' input field. Below it, Step 8 is visible with the text 'At what time would you like to visit the doctor?' and a 'Time' input field. Step 9 is also visible with the text 'We will let you know your appointment details in few minutes.'

The main area shows the configuration for 'Step 7 is taken'. It is set to 'with conditions'. The 'Conditions' section shows a single condition: 'If All of this is true: 6. Which specialist ... is Other'. Below this, there is an 'Assistant says' section with a text area containing 'Please mention your preferred specialist!' and a 'Preview' button.

IBM Watson Assistant

us-south.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aus-south%3Aa%2Fd09ee29f829344568851734f3a63206d%3Abb064674-31ff-4033-83f7-67e4d865...

IBM Watson Assistant Life Upgrade Hospital bot Learning center

Hello

6 is Other

Please mention your preferred specialist!

7

Free text

Continue to next step

8

At what time would you like to visit the doctor?

Time

Continue to next step

8 is 01:00 PM

We will let you know your appointment details in few minutes.

9

Free text

Continue to next step

9 is Defined

Your appointment is fixed at 1.30pm.

10

Action complete

New step +

Step 9 is taken with conditions

Conditions1 condition

IfAll of this is true:

8. At what time would you like to visit the doctor? is at 01:00 PM

andAdd condition +

New condition group +

Assistant says

B I Link Image Audio Video Code

We will let you know your appointment details in few minutes.

User enters free text

Edit response Edit validation

Preview



## Index.html

```
<!DOCTYPEhtml>

<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-
    objectstorage.appdomain.cloud/cloudbucket/assign3.css')} }" type="text/css">
  <script> window.watsonAssistantChatOptions = { integrationID: "2468cc28-3ae1-4cb9-
  a012-3c197820ab0f", // The ID of this integration.
    region: "us-south", // The region your integration is hosted in. serviceInstanceID: "8873bd15-485c-
  4bae-aac6-5acfdc8d28a7", // 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 filename" 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-
objectstorage.appdomain.cloud"
COS_API_KEY_ID=""
COS_INSTANCE_CRN=""

```

```

cos =
    ibm_boto3.resource("s3", ibm_api_key_id=COS_A
PI_KEY_ID, ibm_service_instance_id=COS_INST
ANCE_CRN, config=Config(signature_version="oa
uth"), endpoint_url=COS_ENDPOINT
)
app = Flask(name)

```

```

@app.route('/')
def
index():
    try:
        files =

```

```

cos.Bucket('cloudbucket').objects.all()files_n
ames=[]
for file in
    files:files_names.append(file
    .key)print(file)
    print("Item: {0} ({1} bytes)".format(file.key,
file.size))returnrender_template('index.html',files=files_na
mes)

```

```

exceptClientErrorasbe: print("CLIENT
ERROR:
{0}\n".format(be))returnrender_template('in
dex.html')
exceptExceptionase:
    print("Unable to retrieve bucket contents:
{0}".format(e))returnrender_template('index.html')

```

```

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

```

```

        file_threshold=1024* 1024* 15
    transfer_config
ibm_boto3.s3.transfer.TransferConfig(multipart_thre
shold=file_threshold,

```

```

        multipart_chunksize=part_size
    )

content=f.read() cos.Object('cloudbucket',
    name_file).upload_fileobj(Fileobj=io.BytesIO
        (content),Config=transfer_config
    )
returnredirect(url_for('index'))

```

```

exceptClientErrorasbe: print("CLIENT
    ERROR:
    {0}\n".format(be))returnredirect(url_for('ind
    ex'))

```

```

exceptExceptionase:
    print("Unable to complete multi-part upload:
    {0}".format(e))returnredirect(url_for('index'))

```

```

if name == 'main' :
    app.run(host='0.0.0.0',port=8080,debug=True)

```