

## Assignment-3

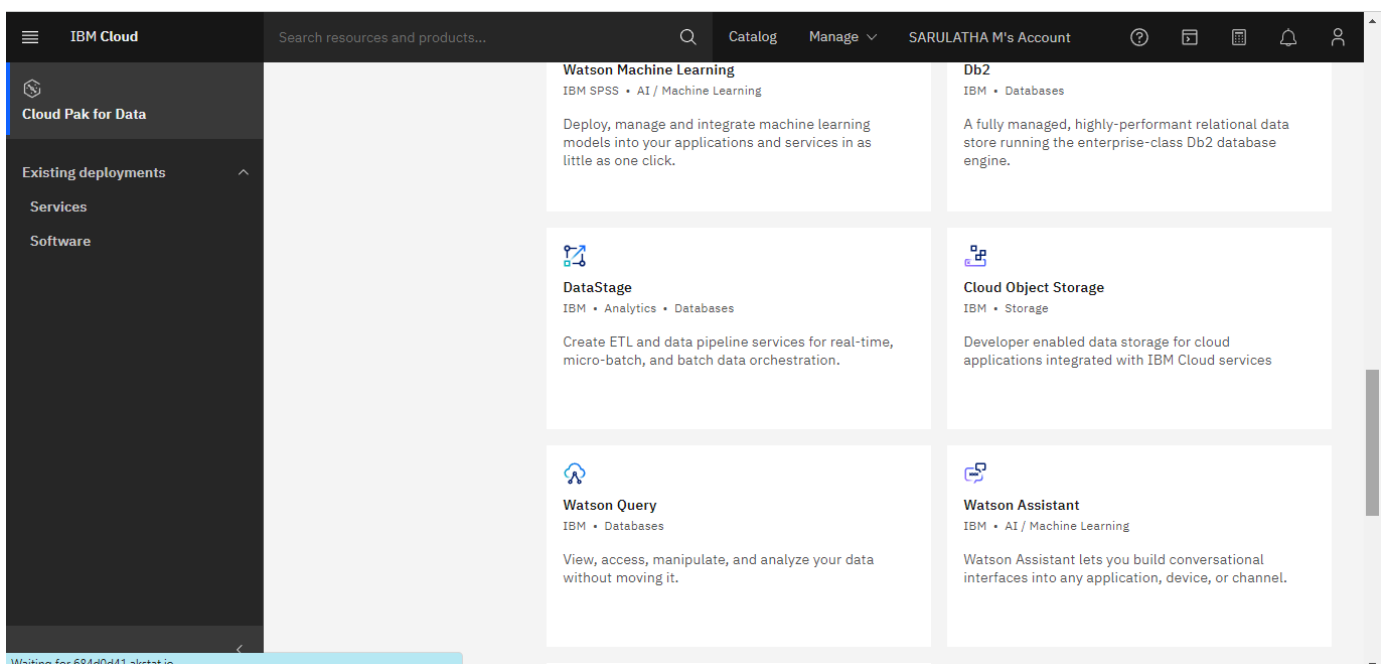
Name	Sarulatha M
Batch	B1-1M3E

1. Create a Bucket in IBM object storage.
2. Upload an 5 images to ibm object storage and make it public. write html code to displaying all the 5 images.
3. Upload a css page to the object storage and use the same page in your HTML code.
4. Design a chatbot using IBM Watson assistant for hospital. Ex: User comes with query to know the branches for that hospital in your city. Submit the web URL of that chat bot as a assignment.
5. Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.

## STEPS TO BE FOLLOWED

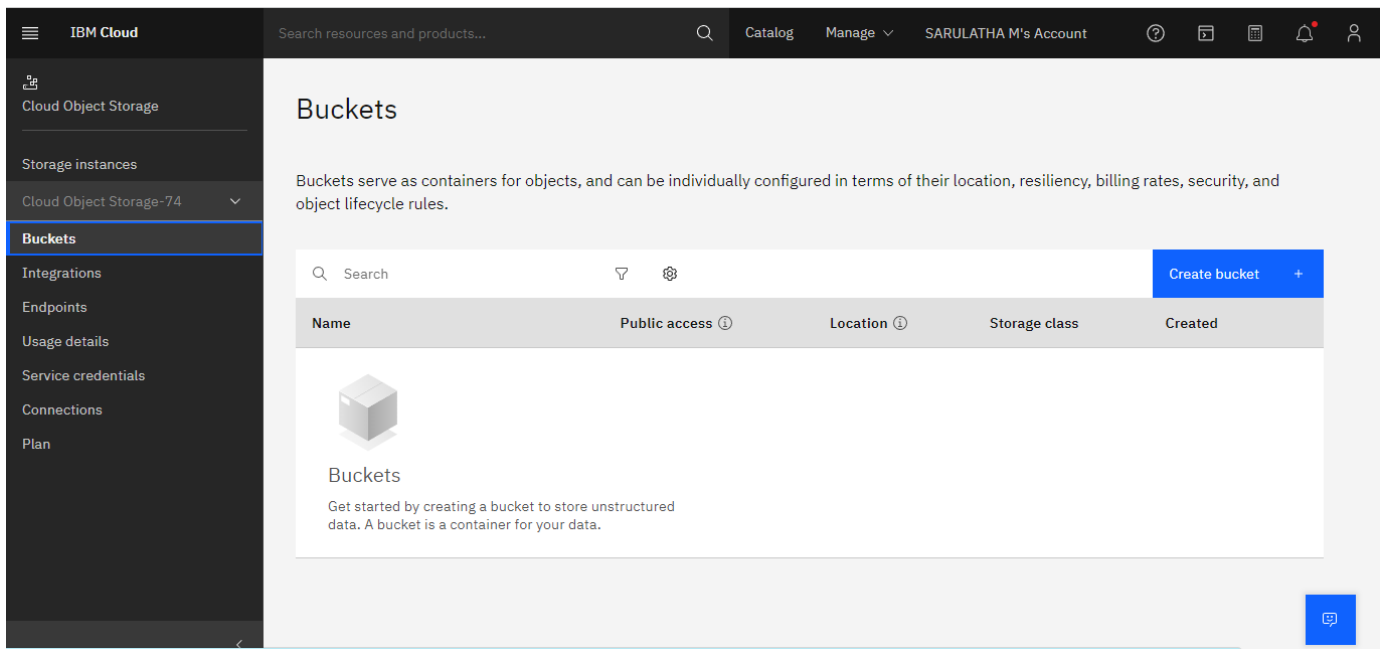
### 1. CREATE A BUCKET AND UPLOAD IMAGES

- a) Select create resource button
- b) Open *Object Storage* from the catalogue



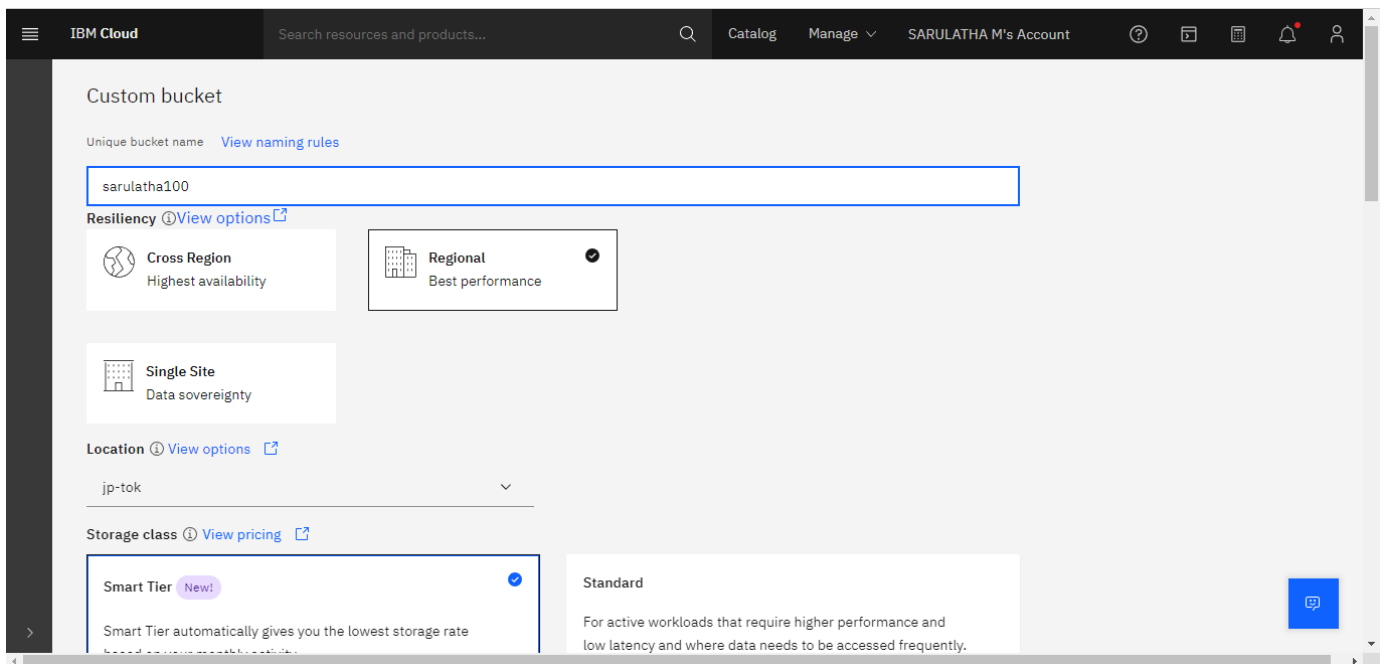
- c) Choose *IBM Cloud* as infrastructure, select the free plan, give a name and select create
- d) Go to active resource list and select the storage created

e) Select create bucket



f) Select customize your bucket, give a unique name for the bucket across all available buckets around the globe

- Resiliency – Regional
- Storage class – Smart Tier



IBM Cloud

Search resources and products...

Storage / Cloud Object Storage-74 /

Transfers

**sarulatha100**

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

Object name Archived Size Last modified

Objects

Drag and drop files (objects) to upload. An object is your data in fixed form.

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

Upload

A bucket created successfully! The bucket sarulatha100 has been created and is now available to add

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...

Catalog Manage SARULATHA M's Account

Cloud Object Storage

Storage instances

Cloud Object Storage-74

**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="#">1img</a>	No	jp-tok	Smart Tier	2022-11-18 5:58 AM
<a href="#">2img</a>	No	jp-tok	Smart Tier	2022-11-18 6:00 AM
<a href="#">3img</a>	No	jp-tok	Smart Tier	2022-11-18 6:01 AM
<a href="#">4img</a>	No	jp-tok	Smart Tier	2022-11-18 6:02 AM
<a href="#">5img</a>	No	jp-tok	Smart Tier	2022-11-18 6:03 AM
<a href="#">sarulatha100</a>	No	jp-tok	Smart Tier	2022-11-18 5:57 AM

IBM Cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-74

**Buckets**

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Search resources and products...

Catalog

Manage

SARULATHA M's Account

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
1img	Yes	jp-tok	Smart Tier	2022-11-18 5:58 AM
2img	Yes	jp-tok	Smart Tier	2022-11-18 6:00 AM
3img	Yes	jp-tok	Smart Tier	2022-11-18 6:01 AM
4img	No	jp-tok	Smart Tier	2022-11-18 6:02 AM
5img	No	jp-tok	Smart Tier	2022-11-18 6:03 AM
sarulatha100	No	jp-tok	Smart Tier	2022-11-18 5:57 AM

IBM Cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-74

**Buckets**

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Search resources and products...

Catalog

Manage

SARULATHA M's Account

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
1img	Yes	jp-tok	Smart Tier	2022-11-18 5:58 AM
2img	Yes	jp-tok	Smart Tier	2022-11-18 6:00 AM
3img	Yes	jp-tok	Smart Tier	2022-11-18 6:01 AM
4img	No	jp-tok	Smart Tier	2022-11-18 6:02 AM
5img	No	jp-tok	Smart Tier	2022-11-18 6:03 AM
sarulatha100	No	jp-tok	Smart Tier	2022-11-18 5:57 AM

Configuration

Access Policies

Access with Data ...

IAM Panel

IBM Cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-74

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Search resources and products...

Catalog

Manage

SARULATHA M's Account

Access policies

Public access

Warning: Granting Public access to this bucket will allow anyone to access the bucket. To

Enable public access

Granting public access will allow anyone to access this bucket and read all the objects in it. Are you sure you want to enable this functionality?

Cancel

Enable

Create access policy

Context-based restrictions

Firewall (legacy)

IBM Cloud

Cloud Object Storage

Storage instances

Cloud Object Storage-74

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Search resources and products...

Catalog

Manage

SARULATHA M's Account

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.

Public URL

Reference URL ⓘ

https://1img.s3.jp-tok.cloud-object-storage.appdomain.cloud

Create bucket +

Name	Reference URL ⓘ	Created
1img	https://1img.s3.jp-tok.cloud-object-storage.appdomain.cloud	2022-11-18 5:58 AM
2img	Yes jp-tok Smart Tier	2022-11-18 6:00 AM
3img	Yes jp-tok Smart Tier	2022-11-18 6:01 AM
4img	Yes jp-tok Smart Tier	2022-11-18 6:02 AM
5img	Yes jp-tok Smart Tier	2022-11-18 6:03 AM
sarulatha100	No jp-tok Smart Tier	2022-11-18 5:57 AM

IBM Cloud

Search resources and products...

Catalog Manage SARULATHA M's Account

Cloud Object Storage

\*ass3 - Notepad

```
File Edit Format View Help
<body>

<br>

<br>

<br>

<br>

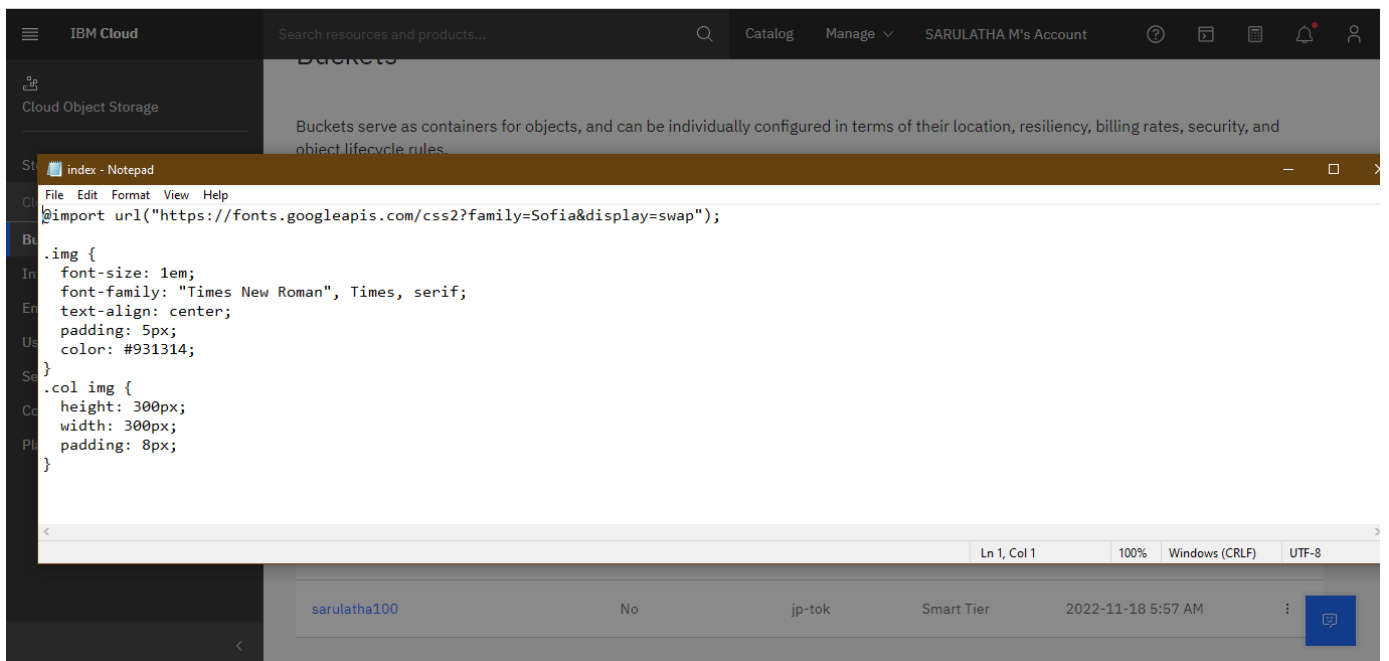
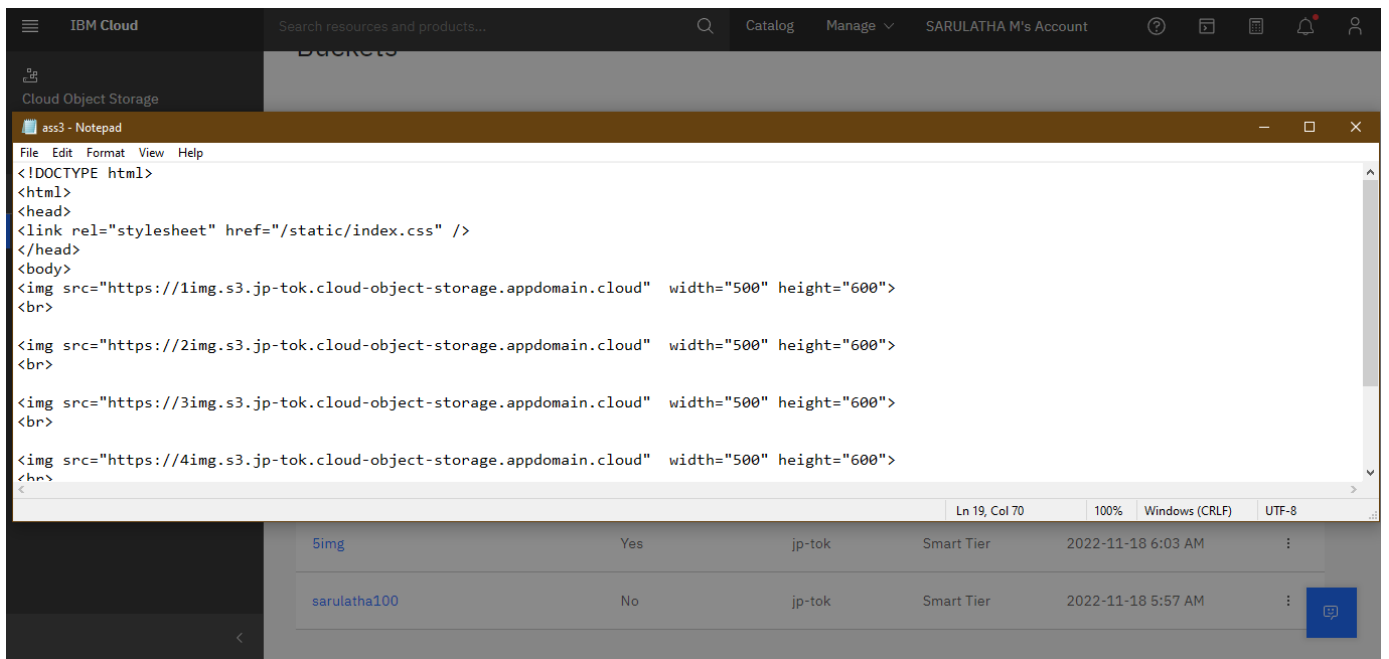
<br>
</body>
</html>
```

Ln 19, Col 70 100% Windows (CRLF) UTF-8

5img	Yes	jp-tok	Smart Tier	2022-11-18 6:03 AM	:
sarulatha100	No	jp-tok	Smart Tier	2022-11-18 5:57 AM	:

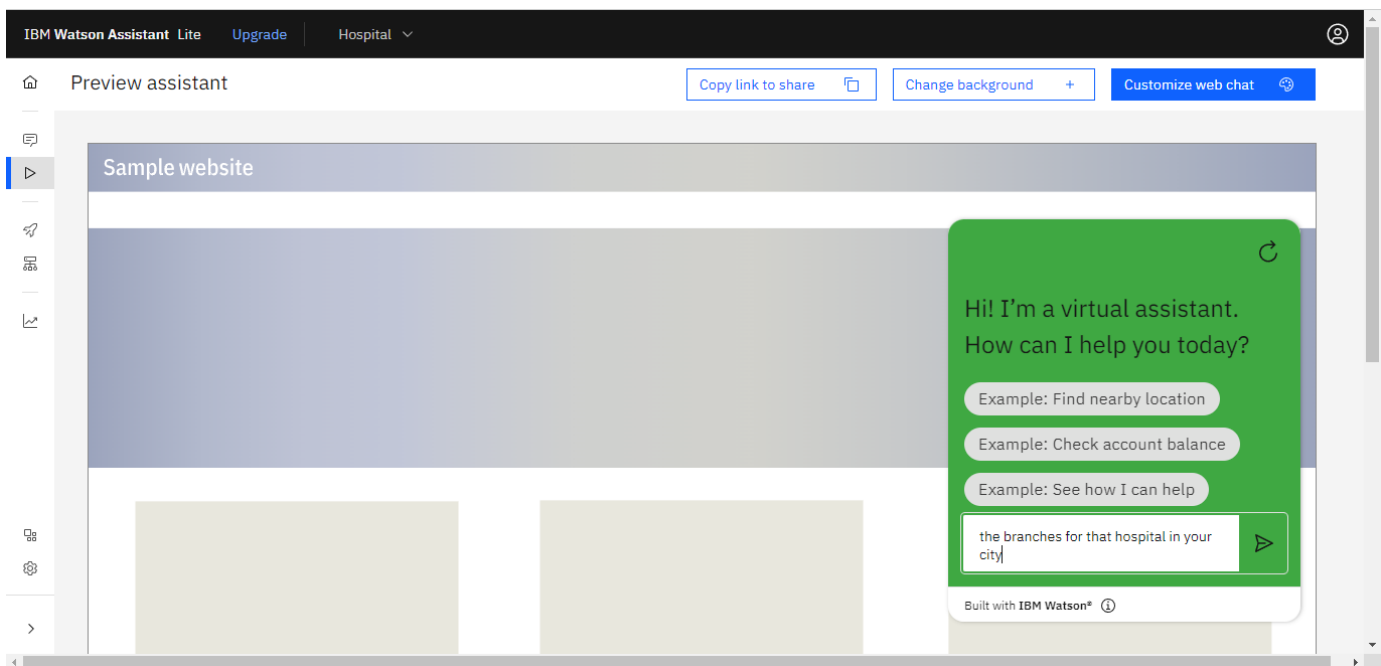
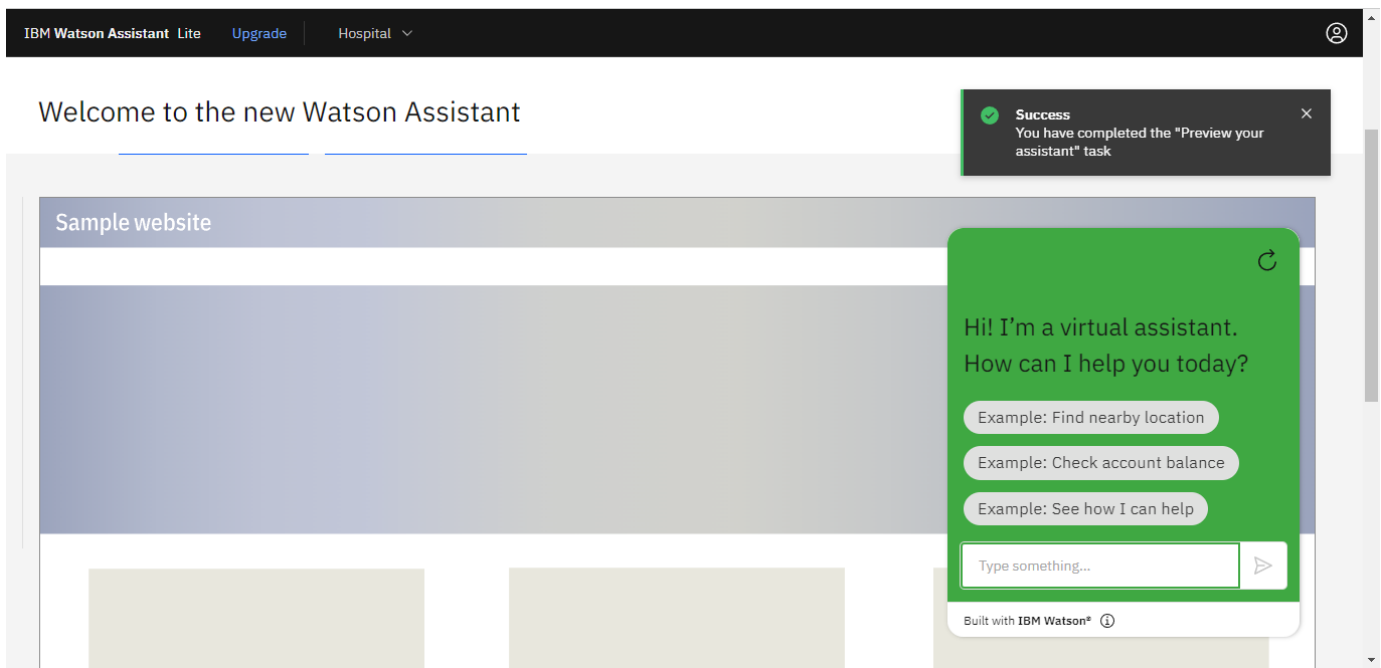


3. Upload a css page to the object storage and use the same page in your HTML code.

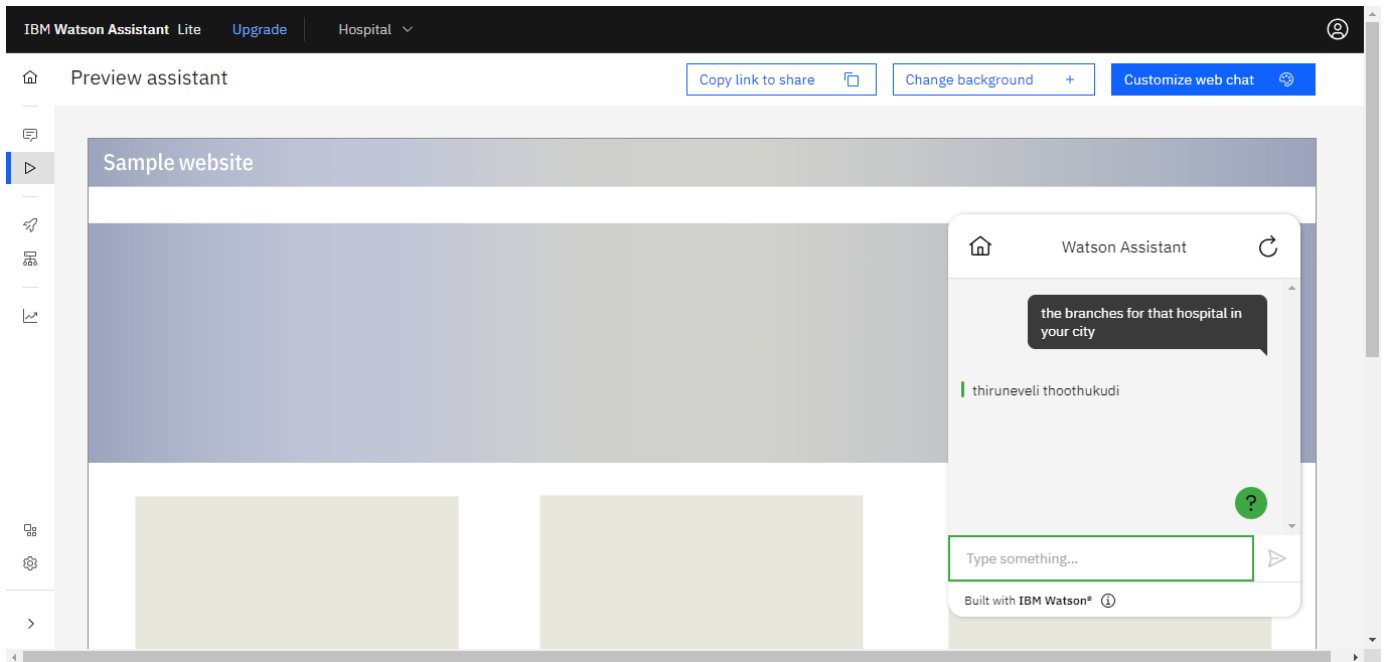


#### 4. Design a chatbot using IBM Watson assistant for hospital

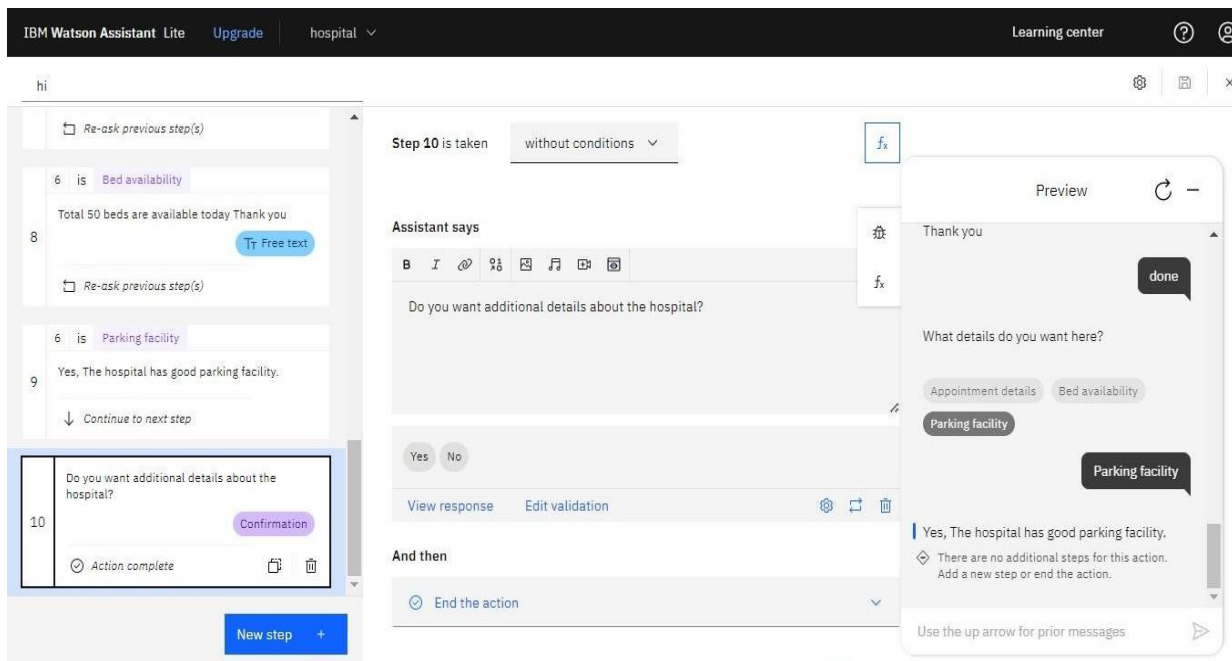
- Select create resource button
- Open *Watson Assistant* from the catalogue
- Choose the lite plan and create the bot
- Launch the assistant and select the actions tab to add new actions to the bot



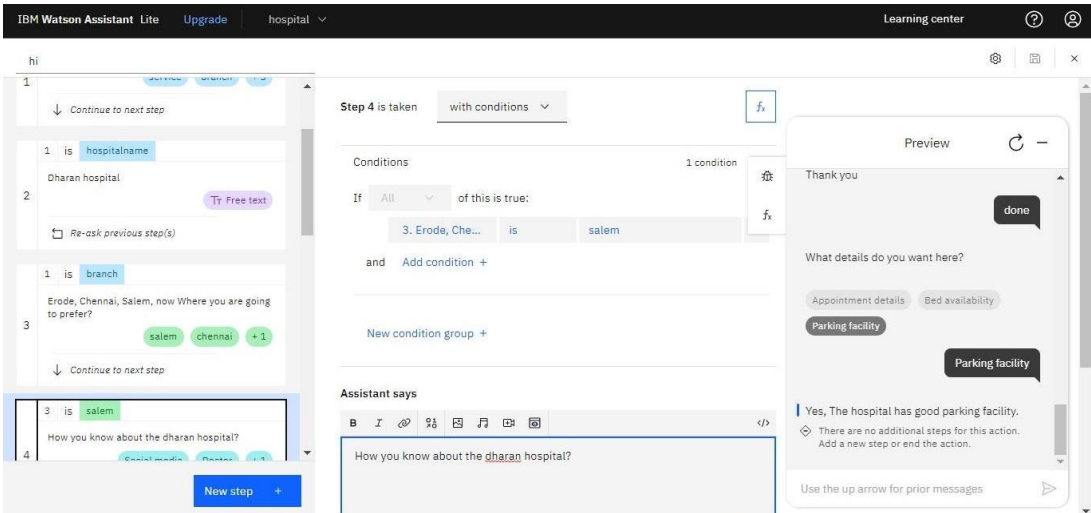
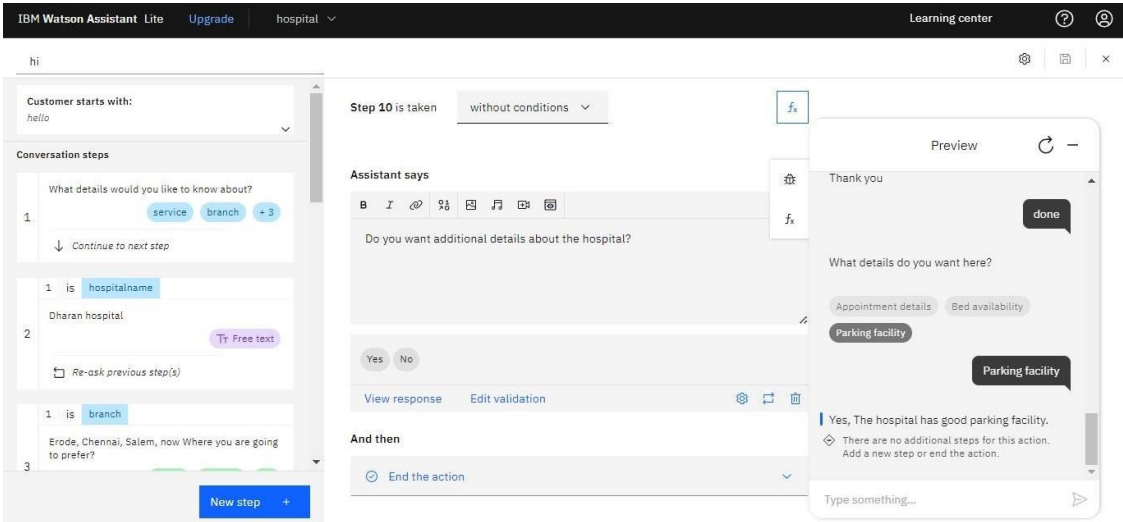


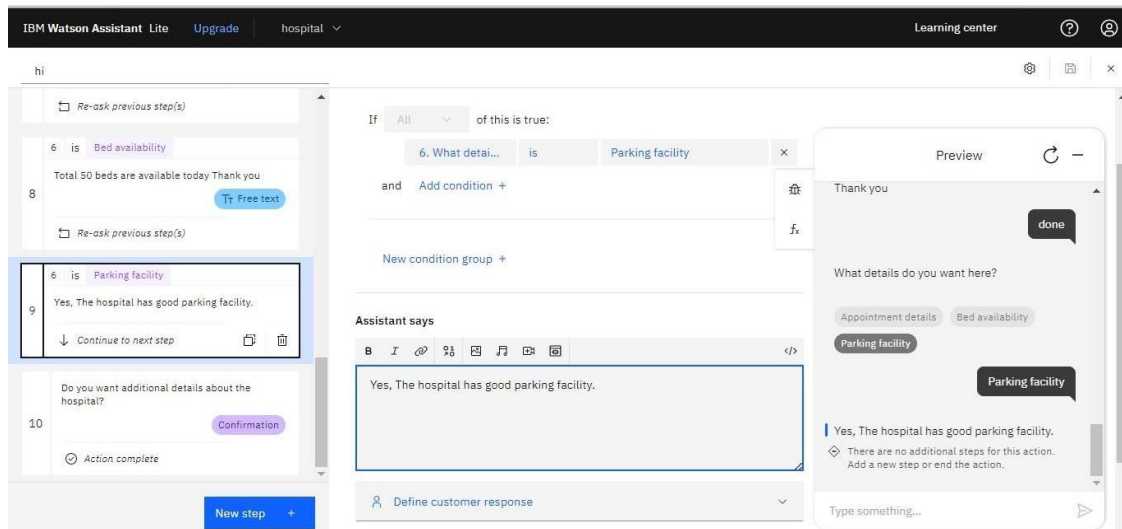
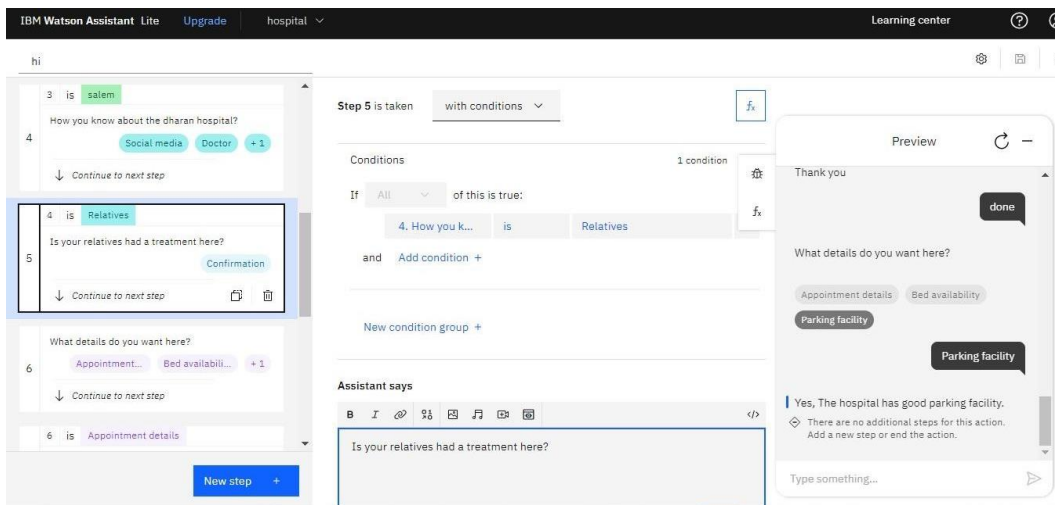


5. Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.



Included 3 conditions in steps:





## 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)
```