

Name	Prasanth A
Roll No	SSNCE195001080
Date	28 October 2022
Team ID	PNT2022TMID53036
Project Name	Project - Personal Expense Tracker

Assignment - 3

Object Storage and Watson Chat Bot

Question

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.

Solutions

1. Create a Bucket in IBM object storage

- Create a object storage from the catalog in IBM Cloud
- Create a new bucket in the object storage
- An unique name should be given to new bucket

IBM Cloud

Search resources and products...

CatalogManagePrasanth A's Account

Cloud Object Storage

Storage instances

Cloud Object Storage-vb

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
test-bucket-assignment3	Yes	jp-tok	Smart Tier	2022-10-31 2:40 PM

27°C Cloudy

IBM Cloud

Search resources and products...

CatalogManagePrasanth A's Account

Cloud Object Storage

Storage instances

Cloud Object Storage-vb

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Storage / Cloud Object Storage-vb /

test-bucket-assignment3

TransfersDetailsActions...

ObjectsConfigurationPermissions

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"/>	im...peg		4.7 KB	2022-10-31 2:45 PM
<input type="checkbox"/>	im...peg		7.2 KB	2022-10-31 2:45 PM
<input type="checkbox"/>	im...peg		7.6 KB	2022-10-31 2:45 PM
<input type="checkbox"/>	im...peg		4.7 KB	2022-10-31 2:45 PM
<input type="checkbox"/>	im...peg		4.8 KB	2022-10-31 2:45 PM
<input type="checkbox"/>	sam....css		166 bytes	2022-10-31 2:52 PM

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

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

Adding 5 images

The screenshot shows the IBM Cloud Object Storage console. The left sidebar contains navigation links: Cloud Object Storage, Storage instances, Cloud Object Storage-vb, Buckets, Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main content area shows the bucket 'test-bucket-assignment3' with tabs for Objects, Configuration, and Permissions. The Objects tab is active, displaying a table of uploaded files. The table has columns for Object name, Archived, Size, and Last modified. There are 5 image files (im...peg) and 1 CSS file (sam...css) listed. The status bar at the bottom shows the system temperature as 27°C Cloudy and the time as 14:56 on 31-10-2022.

Object name	Archived	Size	Last modified
im...peg		4.7 KB	2022-10-31 2:45 PM
im...peg		7.2 KB	2022-10-31 2:45 PM
im...peg		7.6 KB	2022-10-31 2:45 PM
im...peg		4.7 KB	2022-10-31 2:45 PM
im...peg		4.8 KB	2022-10-31 2:45 PM
sam...css		166 bytes	2022-10-31 2:52 PM

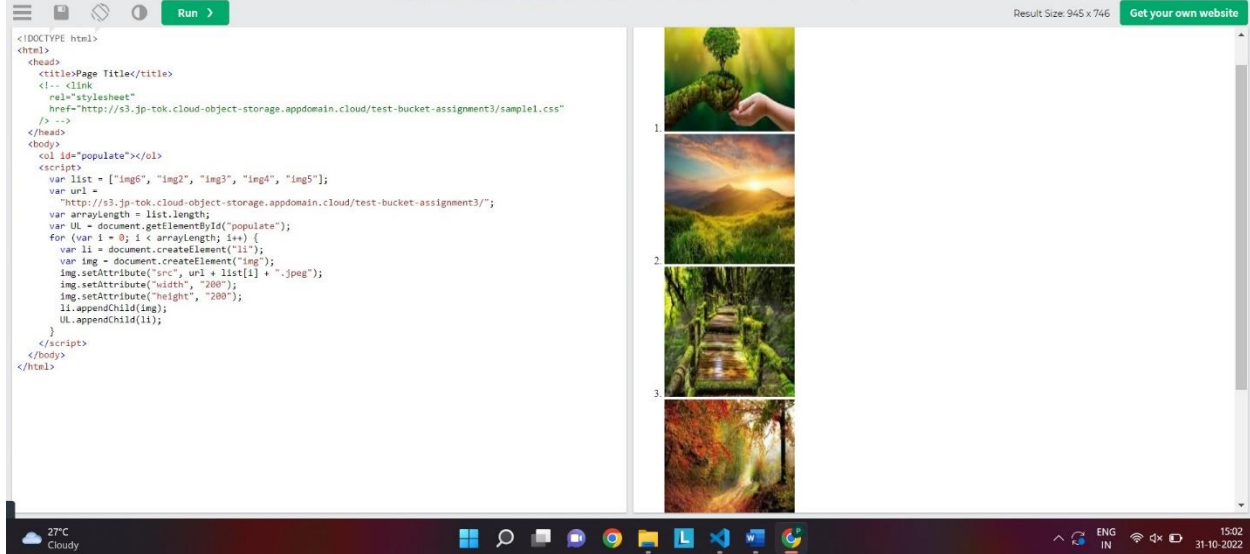
Making it to public

The screenshot displays the IBM Cloud console interface. On the left, a sidebar menu lists navigation options: Cloud Object Storage, Storage instances, Cloud Object Storage-vb, Buckets (highlighted), Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main content area is titled '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.'

Under the 'Access policies' section, the 'Public access' policy is expanded. A yellow warning box states: 'Warning: Granting Public access to this bucket will allow anyone to access the bucket. To revoke public access, remove the "Public access" policy from this bucket within Access groups'. Below this, the 'Status' is 'Enabled' and the 'Role for this bucket' is 'Content Reader'. A description explains: 'As a Content Reader, one can read and list objects in the bucket.' A blue 'Create access policy' button is visible.

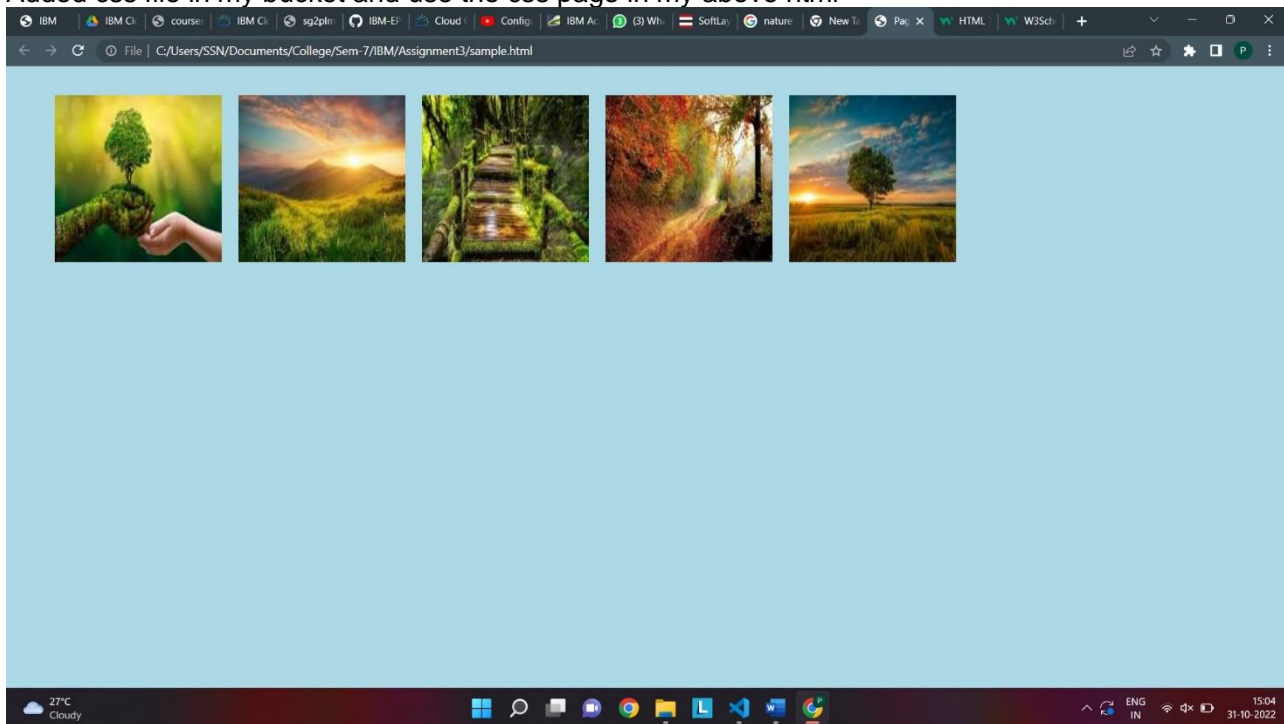
Below the 'Public access' section, there are expandable sections for 'Context-based restrictions' and 'Firewall (legacy)'. The bottom of the screen shows a Windows taskbar with the date 31-10-2022 and time 14:56.

Display the 5 images obtained from my newly created bucket



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

Added css file in my bucket and use the css page in my above html



Code Files

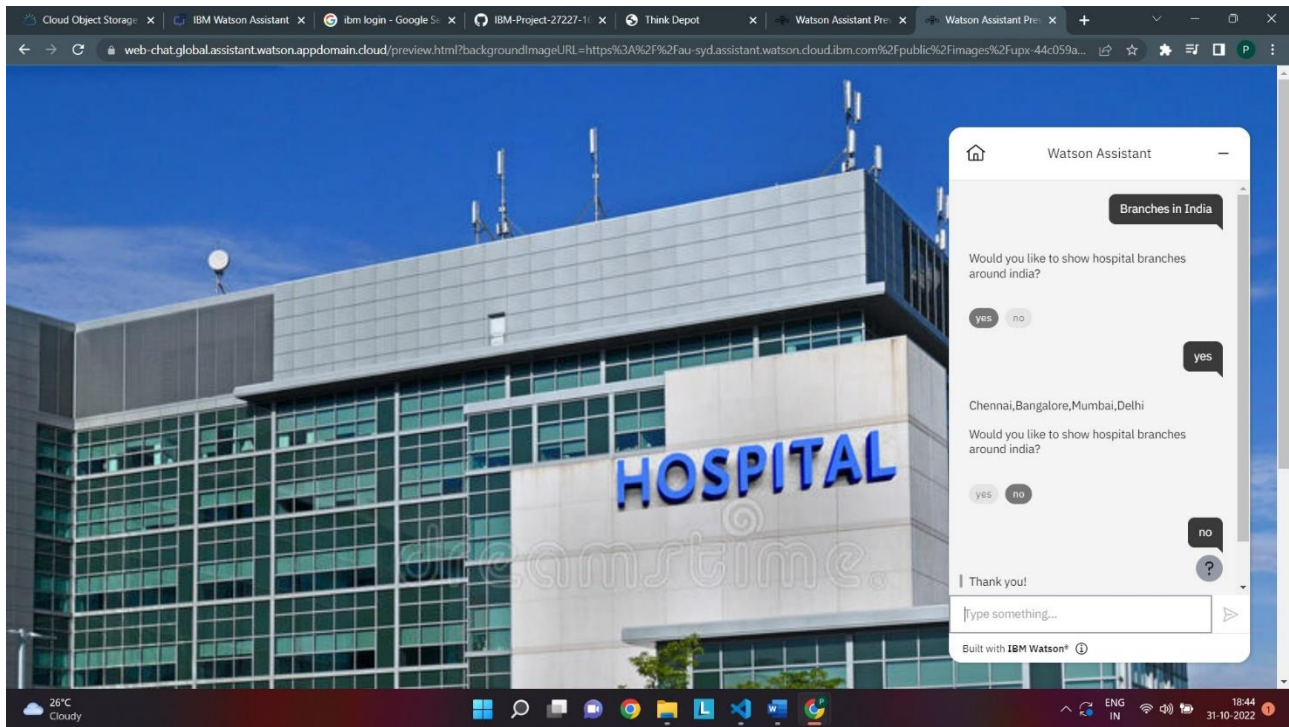
index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link
      rel="stylesheet"
      href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/test-bucket-
assignment3/sample1.css"
    />
  </head>
  <body>
    <ol id="populate"></ol>
    <script>
      var list = ["img6", "img2", "img3", "img4", "img5"];
      var url =
        "http://s3.jp-tok.cloud-object-storage.appdomain.cloud/test-bucket-
assignment3/";
      var arrayLength = list.length;
      var UL = document.getElementById("populate");
      for (var i = 0; i < arrayLength; i++) {
        var li = document.createElement("li");
        var img = document.createElement("img");
        img.setAttribute("src", url + list[i] + ".jpeg");
        img.setAttribute("width", "200");
        img.setAttribute("height", "200");
        li.appendChild(img);
        UL.appendChild(li);
      }
    </script>
  </body>
</html>
```

sample1.css

```
body {  
  display: -webkit-flex;  
  background-color: lightblue;  
  
  display: flex;  
}  
  
li {  
  padding: 10px;  
  display: inline-block;  
}
```

4. Design a chatbot using IBM Watson assistant for hospital. Ex: User comes with a query to know the branches for that hospital in your city. Submit the web URL of that chat bot as an assignment.



Web URL - [hospital-chat-bot](#)

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

Created a watson assistant service with 10 steps and used 3 conditions in it

Actions Screenshot

The image displays three screenshots of a Watson Assistant conversation flow for booking an appointment, showing steps 1 through 10.

First Screenshot (Steps 1-3):

- Step 1:** "Sure. I can make it for you !" with a "Continue to next step" button.
- Step 2:** "Please provide your name." with a "Free text" input field and a "Continue to next step" button.
- Step 3:** "Great! Select mail or phone number to reach you." with "Mobile" and "Email" buttons and a "Continue to next step" button.

Second Screenshot (Steps 4-6):

- Step 4:** "Great! Enter your mobile number." with a "Number" input field and a "Continue to next step" button.
- Step 5:** "Great! Please enter your email id." with a "Regex" input field and a "Continue to next step" button.
- Step 6:** "The available bookings are today and tomorrow. Please select anyone." with "Tomorrow" and "Today" buttons and a "Continue to next step" button.

Third Screenshot (Steps 7-10):

- Step 7:** "Sure! Your appointment is yet to be booked. Please confirm." with "Yes" and "No" buttons and a "Continue to next step" button.
- Step 8:** "Your appointment is booked successfully!" with a "Continue to next step" button.
- Step 9:** "No problem! Your booking is cancelled." with a "Continue to next step" button.
- Step 10:** "Thank You!" with a "Action complete" status and icons for a document and a trash can.

HTML page is created locally and loaded that script in the page

Page Screenshot

