

IBM CLOUD ASSIGNMENT - 3

1. Create a Bucket in IBM object storage.

The screenshot shows the 'Custom bucket' configuration page in the IBM Cloud console. The 'Unique bucket name' field contains 'ibmcloudbucket1409'. A 'Bucket naming rules' box lists the following rules:

- Must be unique across the **whole** IBM Cloud Object Storage system
- Do not use any personal information (any part of a name, address, financial or security accounts or SSN)
- Must start and end in alphanumeric characters (3 to 63)
- Characters allowed: lowercase, numbers and nonconsecutive dots and hyphens

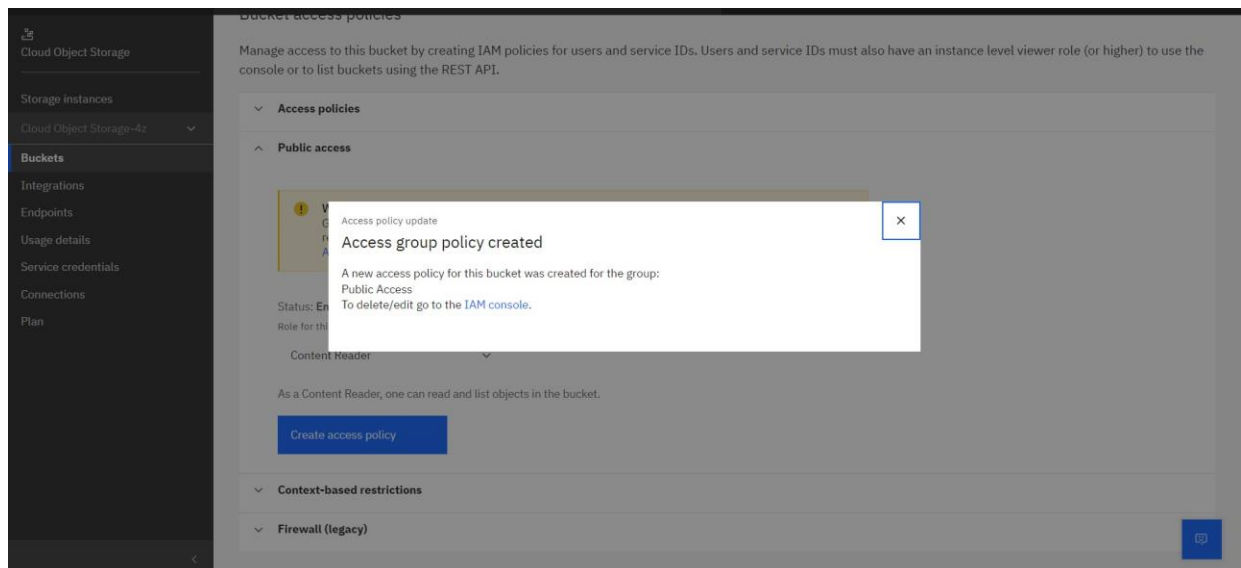
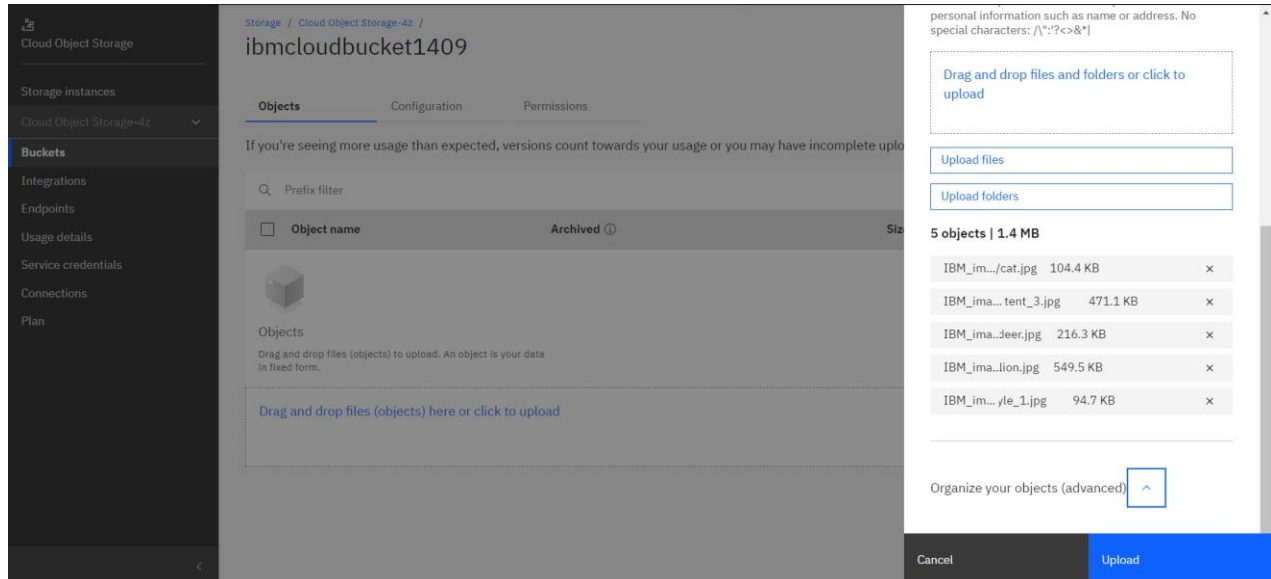
Under the 'Resiliency' section, three options are shown: 'Cross Region' (Highest availability), 'Regional' (Best performance, selected with a checkmark), and 'Single Site' (Data sovereignty). The 'Location' is set to 'jp-tok'. The 'Storage class' is set to 'Standard'.

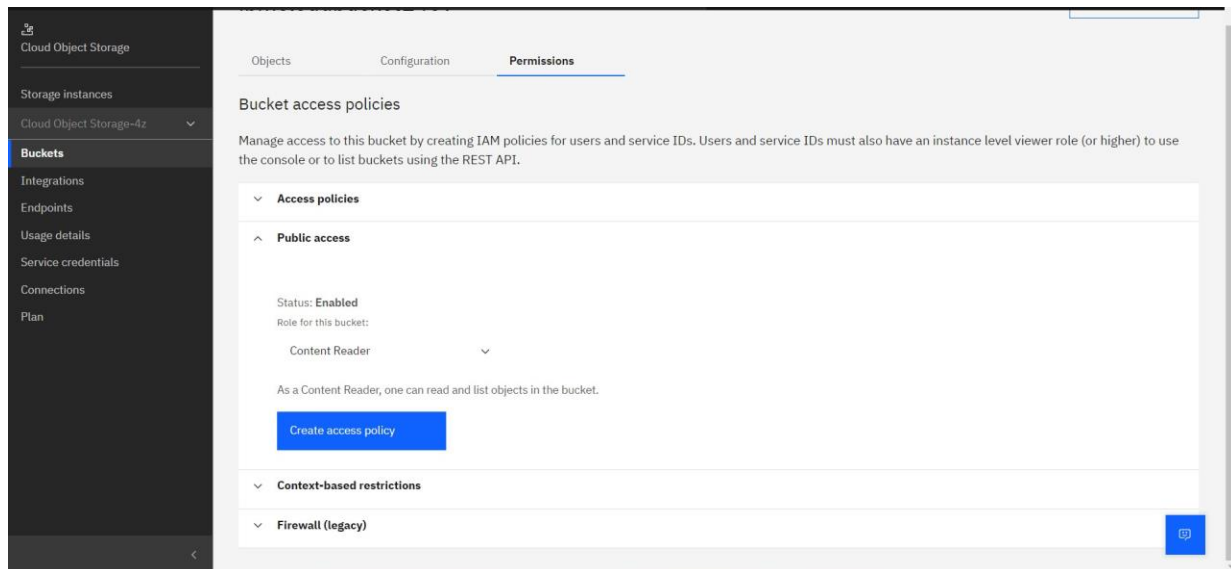
The screenshot shows the IBM Cloud console after the bucket has been created. A notification in the top right corner states: 'A bucket created successfully! The bucket ibmcloudbucket1409 has been created and is now available to add'. The main view shows the bucket 'ibmcloudbucket1409' with tabs for 'Objects', 'Configuration', and 'Permissions'. The 'Objects' tab is active, displaying a table with columns: Object name, Archived, Size, and Last modified. Below the table, there is a section for 'Objects' with a message: 'Drag and drop files (objects) to upload. An object is your data in fixed form.' and a button to 'Upload'.

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

Code File : display.html

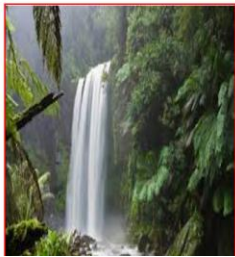
Uploading Images





Style file present in local machine

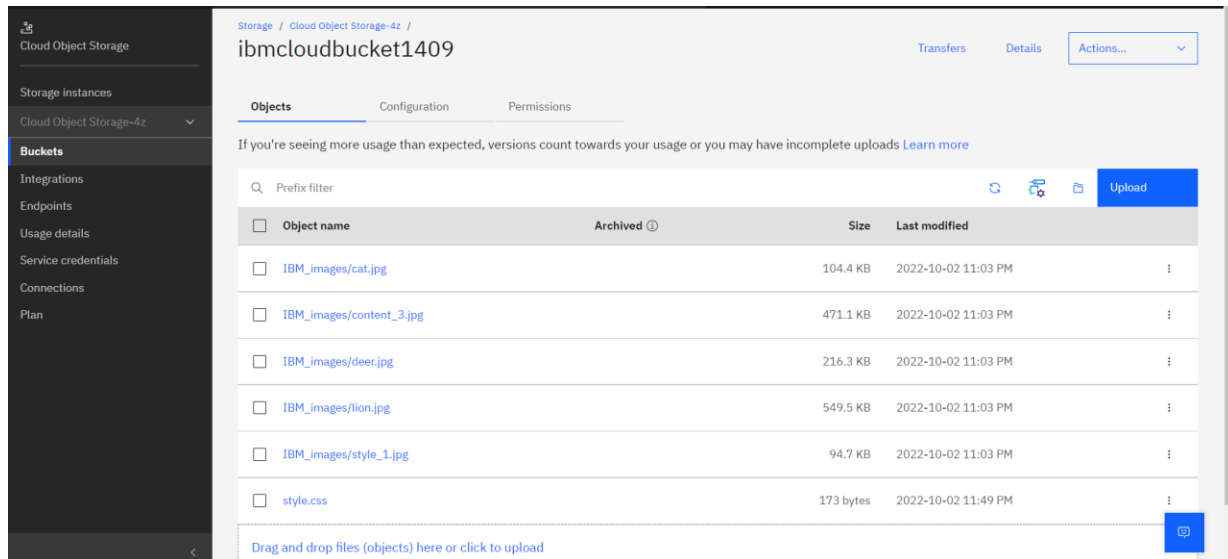
```
Assignments > Team Leader > Assignment3 > display.html > html > body > img
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>
5      |   Display Images
6    </title>
7    <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10   <h2>Images From Cloud Object Storage</h2>
11   </img>
12   </i
13   </img>
14   </img>
15   </img>
16 </body>
17 </html>
```



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

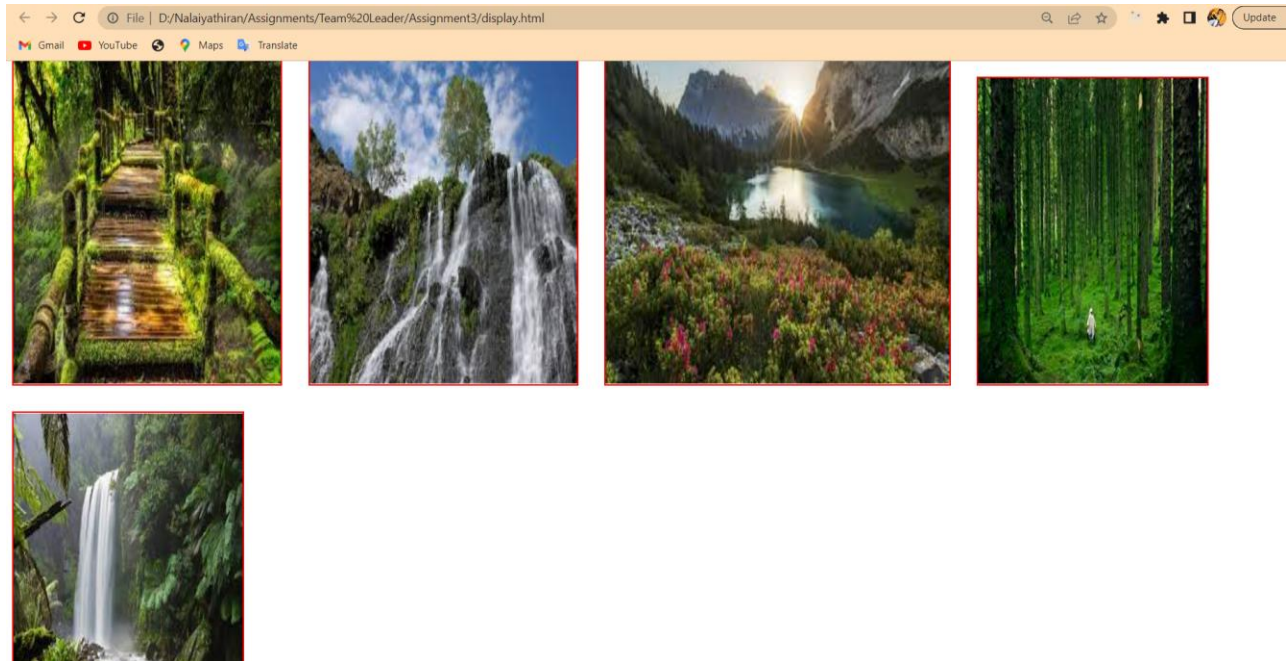
Code File : display_que3.html

Uploading Style page to cloud object storage



```
Assignments > Team Leader > Assignment3 > display.html > html > body > img
1  !DOCTYPE html>
2  html>
3  head>
4    <title>
5      Display Images
6    </title>
7    <link rel="stylesheet" href="style.css">
8  /head>
9  body>
10   <h2>Images From Cloud Object Storage</h2>
11   </img>
12   </i
13   </img>
14   </img>
15   </img>
16 /body>
17 /html>
```

Using the style page from cloud object storage



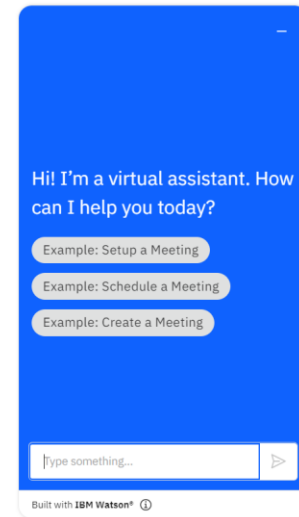
4.Design a chatbot using IBM Watson assistant for hospital. Ex: User comes with query to know the branches for that hospital in your city. Submit the web URL of that chat bot as a assignment.

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageUrl=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-f122de51-ed15-497c-96da-2441907f6219%3A%3A1e032c38-9f84-4b97-8d18-ddfb4c10fc4c&integrationID=4fabf571-b0a6-4a0b-b6b9-e6f2293c0bac®ion=au-syd&serviceInstanceID=f122de51-ed15-497c-96da-2441907f6219>

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

Code File : display_watson.html

ChatBot Page



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>
5     Display Images
6   </title>
7   <link rel="stylesheet" href="style.css">
8   <script>
9     window.watsonAssistantChatOptions = {
10      integrationID: "ebffc6c0-c92f-4cf9-98a2-363b64e22002", // The ID of this integration.
11      region: "au-syd", // The region your integration is hosted in.
12      serviceInstanceID: "f122de51-ed15-497c-96da-2441907f6219", // The ID of your service instance.
13      onLoad: function(instance) { instance.render(); }
14    };
15    setTimeout(function(){
16      const t=document.createElement('script');
17      t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion ||
18      'latest') + "/WatsonAssistantChatEntry.js";
19      document.head.appendChild(t);
20    });
21   </script>
22 </head>
23 <body>
24   <h1>ChatBot Page </h1>
25 </body>
26 </html>
```