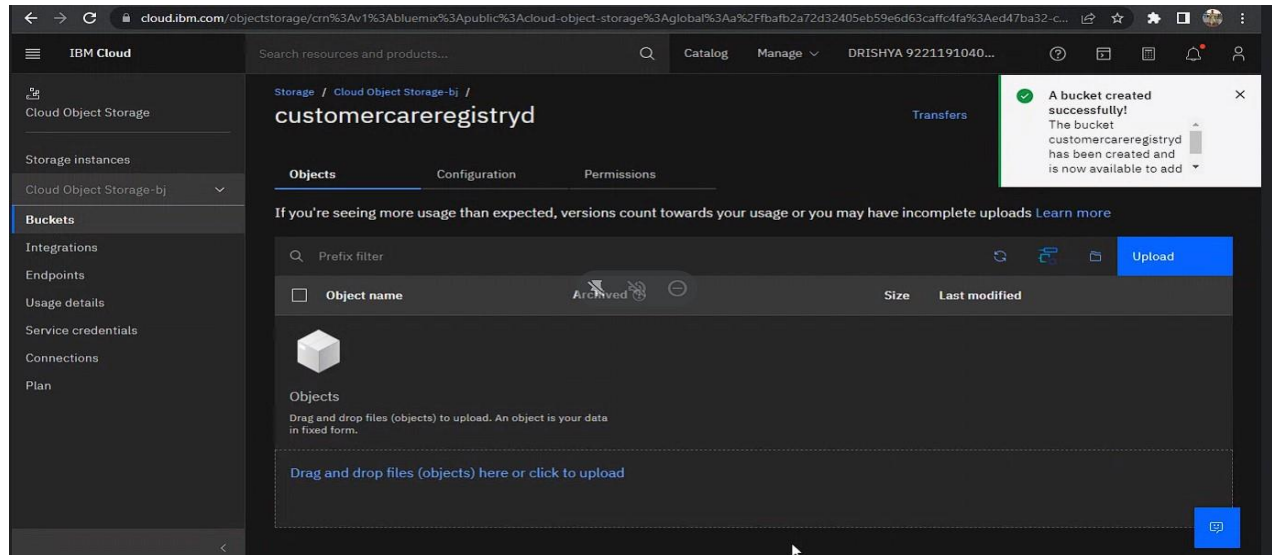
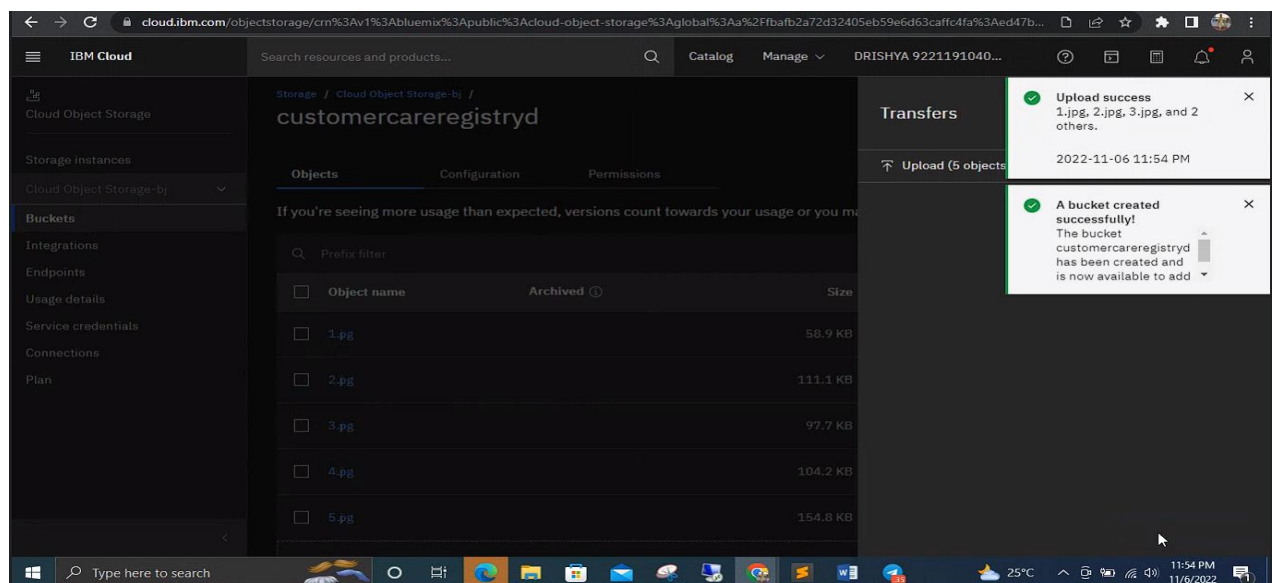


ASSIGNMENT-03

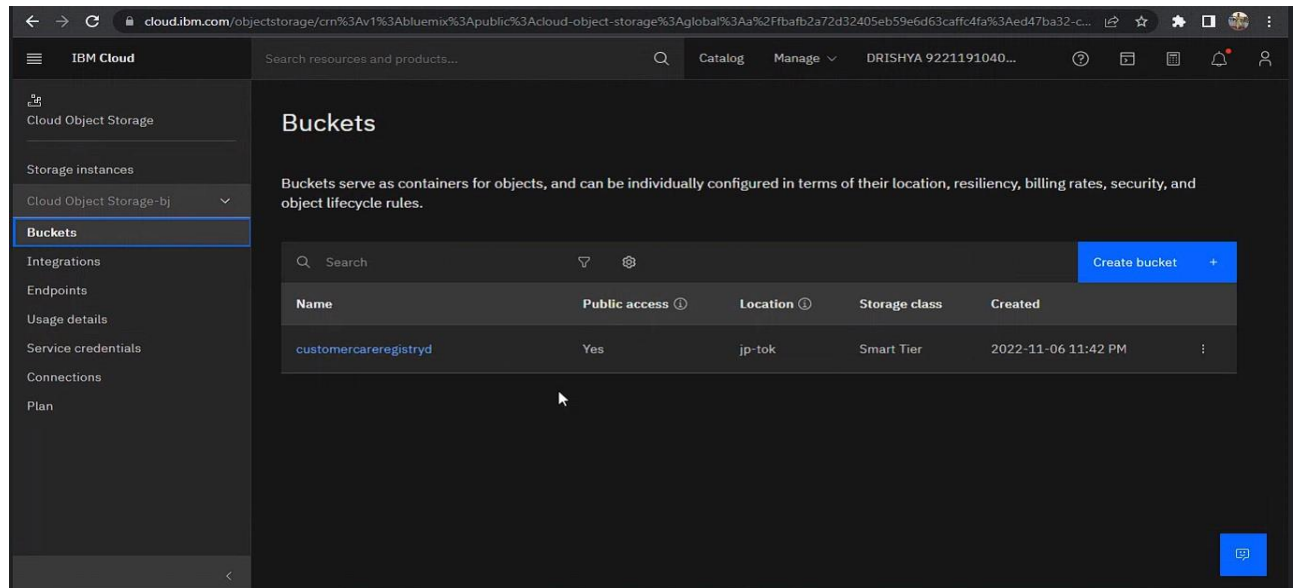
1. Create a Bucket in IBM object storage.



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



3. Enabling private access



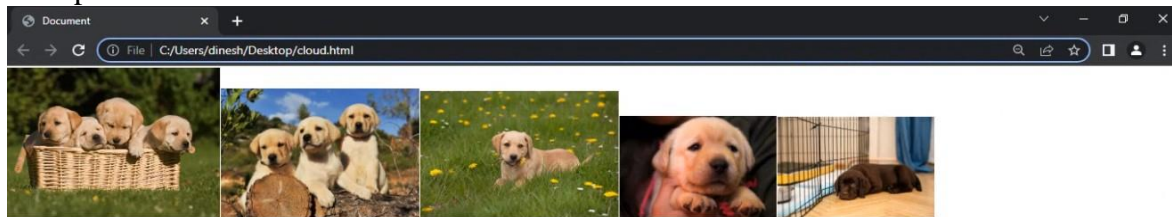
The screenshot displays the IBM Cloud Object Storage interface. On the left, a sidebar contains navigation links: Cloud Object Storage, Storage instances, Cloud Object Storage-bj, Buckets (highlighted), Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main content 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 with a search bar and a 'Create bucket' button. The table has five columns: Name, Public access, Location, Storage class, and Created. One bucket is listed: 'customeracareregistryd' with 'Yes' for Public access, 'jp-tok' for Location, 'Smart Tier' for Storage class, and '2022-11-06 11:42 PM' for Created. A blue chat icon is in the bottom right corner.

Name	Public access	Location	Storage class	Created
customeracareregistryd	Yes	jp-tok	Smart Tier	2022-11-06 11:42 PM

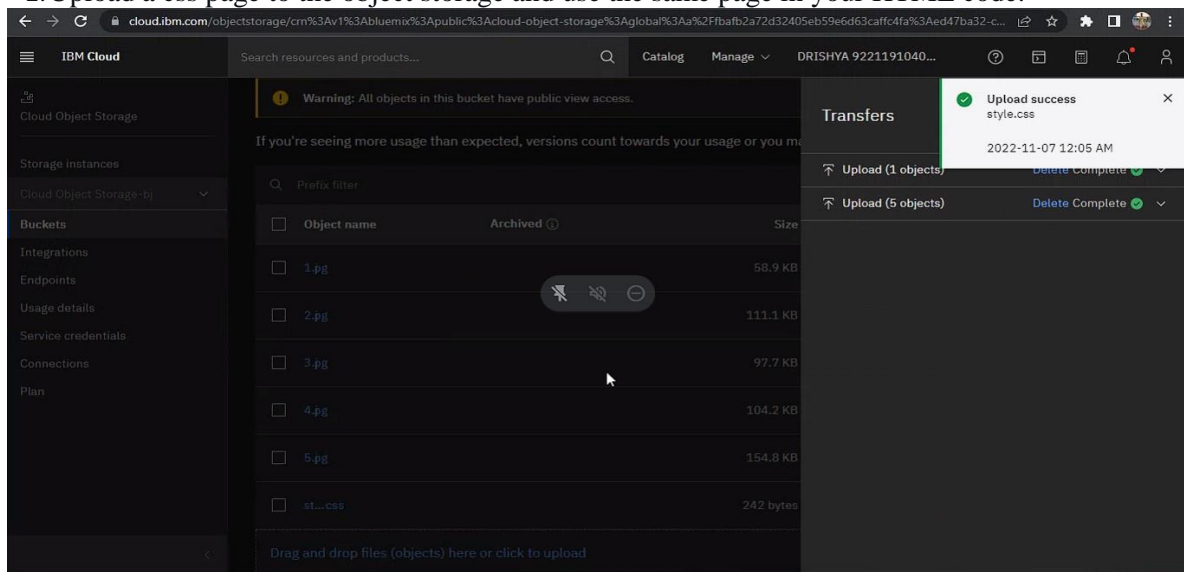
Html code:

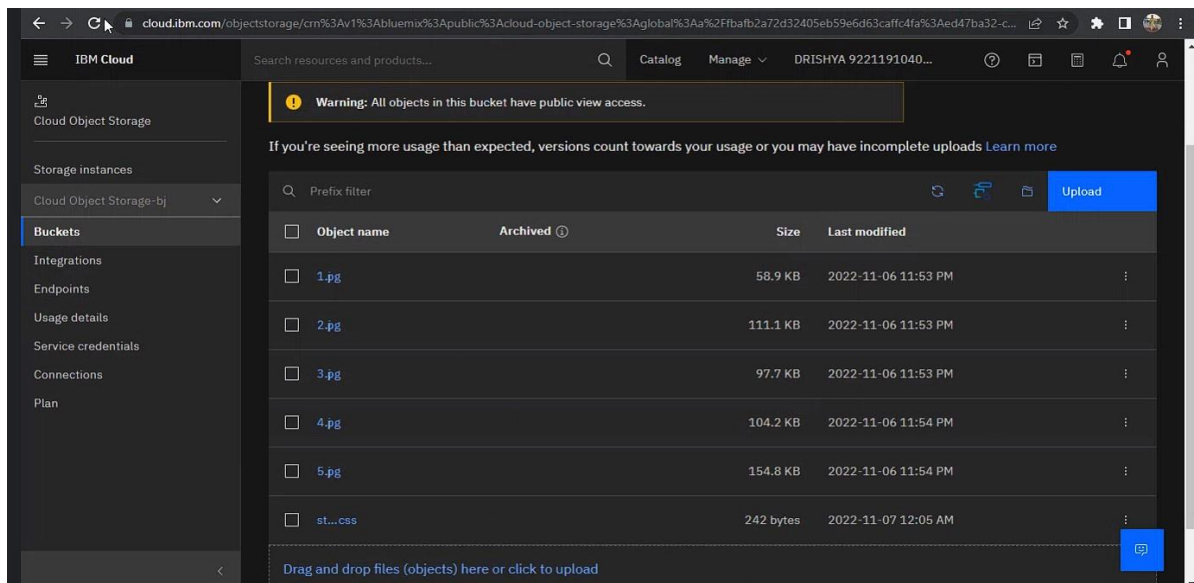
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <link rel="stylesheet"
href="https://customercareregistryd.s3.jp-tok.cloud-object-
storage.appdomain.cloud/style.css" />
    <title>Document</title>
  </head>
  <body>
    <div class="img-container">
      
      
      
      
      
    </div>
  </body>
</html>
```

Output:



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





Css:

```
.img-container
{
  display: flex;
  flex-direction: row;
  gap: 2rem;
  flex-wrap: wrap;
}
.img-container img
{
  width: 300px;
  height: 300px;
  border-radius: 10px;
}
```

3.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-27413bf2-e2c4-4422-bb6d-30a6b2f75a3c%3A%3Ac343695-711a-41aa-9ef0-d01b567b4f36&integrationID=bc3bbd30-2788-42fc-810f-6ee967ad6490®ion=au-syd&serviceInstanceID=27413bf2-e2c4-4422-bb6d-30a6b2f75a>

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

Html code:

```
<!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" />
    <link rel="stylesheet" href="https://customercareregistryd.s3.jp-tok.cloud-
object-storage.appdomain.cloud/style.css" />
    <title>Document</title>
  </head>
  <body>
    <div class="img-container">
      
      
      
      
      
    </div>
    <script>
      window.watsonAssistantChatOptions = {
        integrationID: "bc3bbd30-2788-42fc-810f-6ee967ad6490", // The ID of this
integration.
        region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "27413bf2-e2c4-4422-bb6d-30a6b2f75a3c", // 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>
  </body>
</html>
</body>
</html>
```

Output:

