

# ASSIGNMENT 3

## CLOUD OBJECT & IBM WATSON

DATE	26.10.2022
TEAM ID	PNT2022TMID45905
PROJECT NAME	PERSONAL EXPENSES TRACER APPLICATION
MAXIMUM MARKS	10 MARKS

QUESTION 1 : Create a Bucket in IBM object storage

The screenshot shows the IBM Cloud Object Storage console. The left sidebar contains navigation links: Cloud Object Storage, Storage instances, Cloud Object Storage-Ib, Buckets (selected), Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main content area displays the details for a bucket named 'saravanan-ibm'. At the top, there are tabs for Archive rule, Quota enforcement, Retention policy, Lifecycle policy, and Cross-Origin Resource Sharing (CORS) rules. The 'Bucket details' section shows the following information:

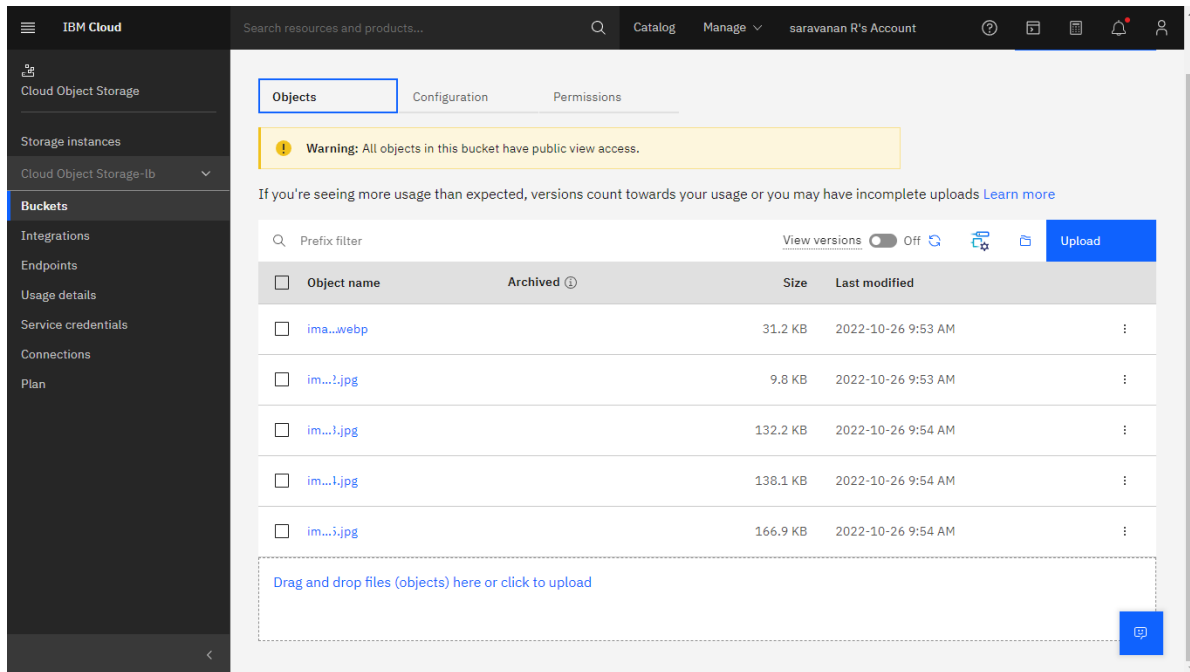
Bucket name	saravanan-ibm	Total bytes	478.2 KB
Service instance	cloud-object-storage	Resiliency	Regional
Total objects	5	Location	jp-tok
Storage class	Smart Tier ⓘ	Date created	2022-10-26 9:43 AM
Cloud Functions trigger	Disabled <a href="#">Learn more</a>		

Below the details, the 'Bucket instance CRN' is shown: crn:v1:bluemix:public:cloud-object-storage:global:a/94c1f17c3ff849c1bd2cad92342ab1eb:5d46267e-1634-4362-bdec-a882498ac7f4:bucket:saravanan-ibm. A 'Show less' link is available.

The 'Endpoints' section explains that endpoints are used with credentials (keys, CRN, bucket name) to tell the service where to look for the bucket. It mentions that depending on the service or application location, one of the endpoint types should be used. A 'Show less' link is also present.

Under the 'Regular Endpoints' section, there is a blue button with a document icon.

QUESTION 2 : Upload an 5 images to IBM object storage and use the same page in your html code



index.html

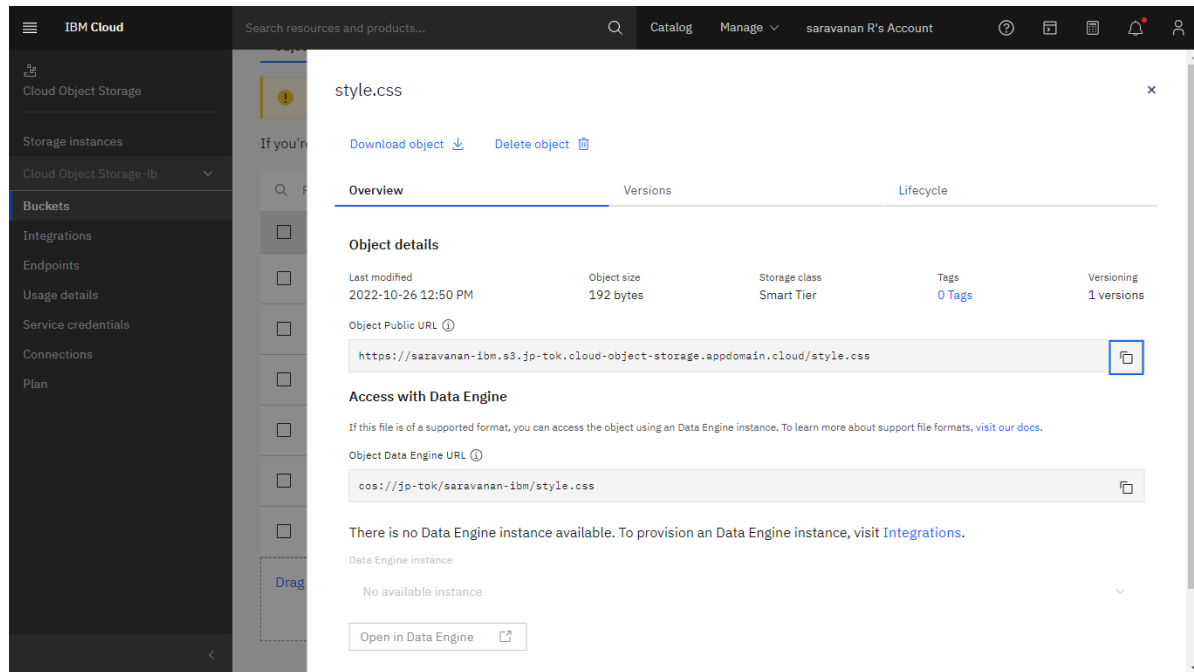
```
<html>
  <head><title>Jack Sparrow</title>
    <link rel="stylesheet"href="https://saravanan-ibm.s3.jp-tok.cloud-object-
storage.appdomain.cloud/style.css">
  </head>
  <body><center><h1> Jack Sparrow</h1></center>
    <div>
      
      
      
      
      
    </div>
  </body>
</html>
```

Output:

Jack Sparrow



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



style.css

```
body {  
    background-color: aqua;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    flex-direction: column;  
}  
  
img {  
    height: 240px;  
    border-radius: 10px;  
    margin: 10px;  
}
```

Output:

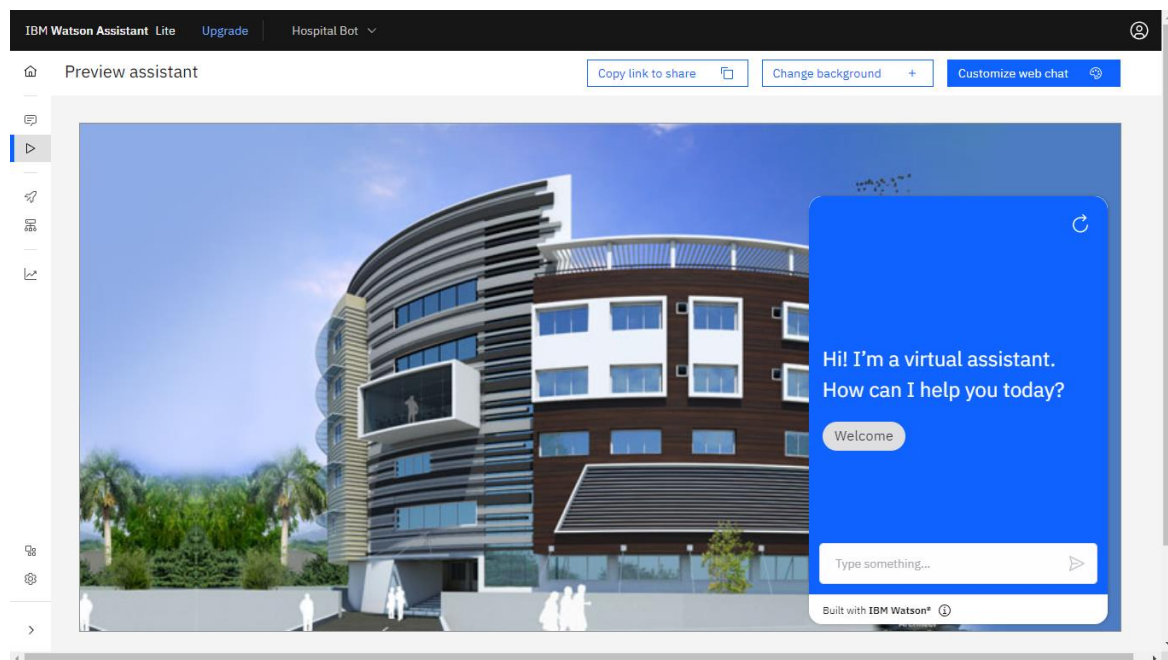


Question 4: Design a chatbot using IBM Watson assistant for hospital (Ex user come with query to know the branches for that hospital in your city. Submit web url of that chat bot as a assignment

WEB URL:

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-703de420-34e1-4702-83e3-0e688a602629%3A%3A1e009ed0-5011-433c-9f8f-cd65c25fe290&integrationID=81636b2f-25de-4689-9f85-749164bfa477&region=au-syd&serviceInstanceID=703de420-34e1-4702-83e3-0e688a602629>

OUTPUT

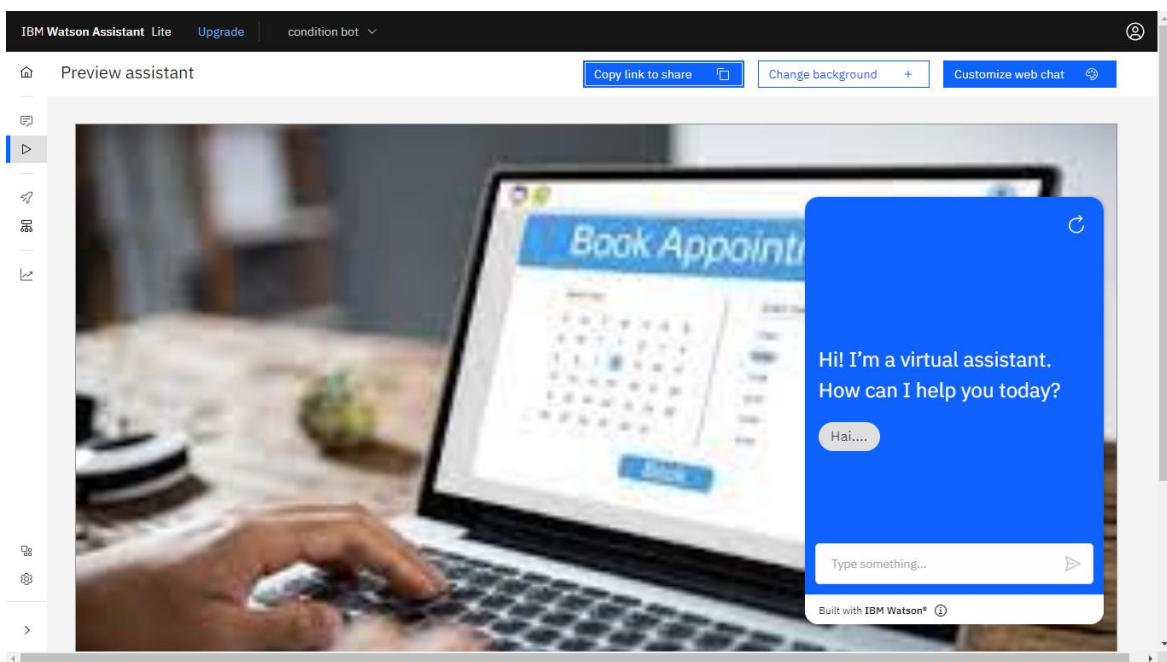


QUESTION 5 : Create Watson Assistant Service with 10 steps and use 3 condition in it. Load the script in HTML page

## WEB URL:

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageUrl=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-703de420-34e1-4702-83e3-0e688a602629%3A%3Afedb1c2e-49a9-4bc7-9a6a-18b0f92bf499&integrationID=f7b7bbbe-d846-4e5b-be6b-25b97f5613c7&region=au-syd&serviceInstanceID=703de420-34e1-4702-83e3-0e688a602629>

## OUTPUT:



## Script:

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "f7b7bbbe-d846-4e5b-be6b-25b97f5613c7", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "703de420-34e1-4702-83e3-0e688a602629", // 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>
```

## OUTPUT:

The screenshot displays a web browser window with two main components. On the left is a form titled "Appointment Details" with a green header. The form includes fields for "Appointment for\*" (set to "Coffee"), "Appointment Description:" (with the text "I wish to get an appointment to skype for resolving a software problem."), "Date\*" (format "mm/dd/yyyy"), "Time\*" (format "--:--"), and "How Long??(Minutes)" (with radio buttons for 30, 60, 90, and more). A green button at the bottom is labeled "Request For Appointment". On the right is a chat window titled "Watson Assistant". The chat history shows a user message "ss@gmail.com" and a system response: "Let's confirm your appointment. I'll need to authenticate you first. What's the email address associated with the appointment?", followed by "Thanks. I've found you in our system! You currently have an appointment scheduled for Oct 26, 2022 at 12:00:06 PM. Would you like to confirm this appointment?". There are two response buttons: "Yes, confirm this appointment" and "No, I'd like to make a modification". A search bar at the bottom of the chat window contains the text "Type something...".



ibm watson assistant.html

C:/Users/Rasith/Desktop/New%20folder%20(5)/ibm%20watson%20assistant.html

2 Appointment Details

Appointment for\*:  
Coffee

Appointment Description:  
I wish to get an appointment to skype for resolving a software problem.

Date\*:  
mm/dd/yyyy

Time\*:  
--:--

How Long??(Minutes)  
☒ 30 ☐ 60 ☐ 90 ☐ more

Request For Appointment

Watson Assistant

Cancel this appointment

Change this appointment

Change this appointment

Alright, let's get your appointment updated.

How long would you like to meet for?

15 minutes

30 minutes

1 hour

1 hour

What day works best for you?

Choose a date (mm/dd/yyyy)

mm/dd/yyyy

Type something...

Built with IBM Watson®