NAAN MUDHALVAN - IBM CLOUD BASED PROJECT

PART – PHASE 3

TITLE : IMAGE RECOGNITION SYSTEM USING IBM CLOUD VISUAL RECOGNITION.

PRESENTED BY

K. PAVITHRA

T. ABIRAMI

K. DHARANI

S. NIVETHA

INTRODUCTION:

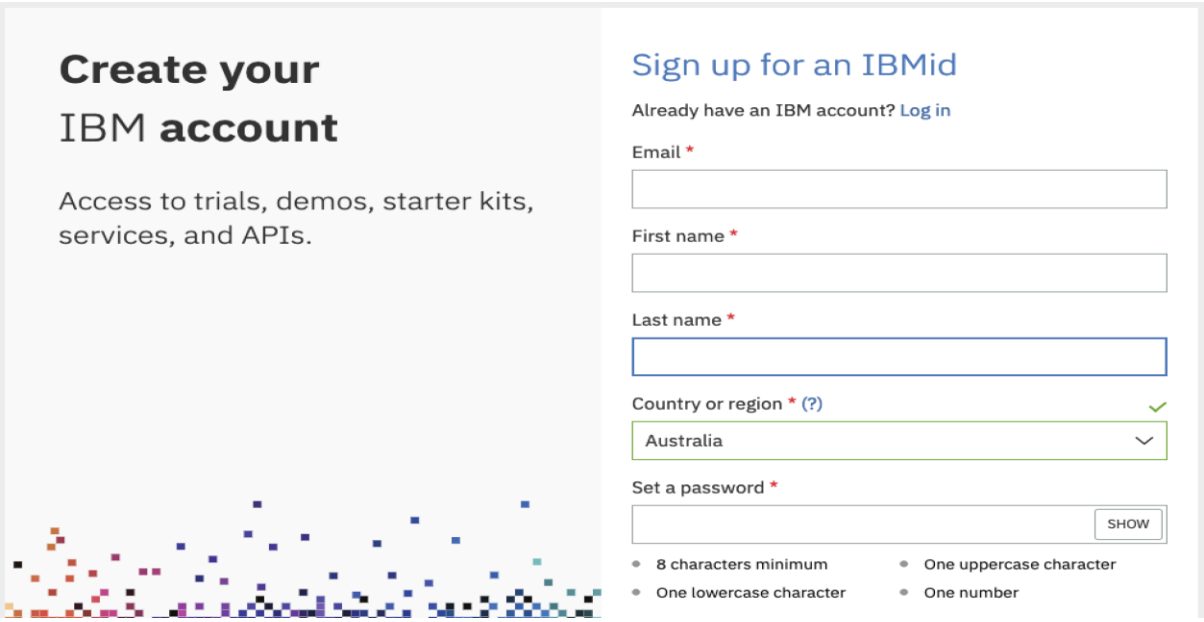
"Building an image recognition system using IBM Cloud Visual Recognition empowers developers and businesses to harness the capabilities of artificial intelligence for visual analysis. This technology allows you to train models to identify objects, scenes, and patterns within images, making it a valuable tool for various applications, from automating content moderation to enhancing user experiences.

IBM Cloud Visual Recognition offers a robust platform for creating custom image recognition models tailored to your specific needs. In this comprehensive guide, we will walk you through the process of setting up an IBM Cloud account, creating a Visual Recognition service instance, preparing your image dataset, training your model, and integrating it into your applications. With this system, you can provide your users with the ability to extract meaningful insights from images, enhance security, and automate tasks that rely on image analysis.

PROCEDURE:

Step 1: Create an IBM Cloud Account

If you don't already have an IBM Cloud account, you can sign up for a free account at IBM Cloud. Once you have an account, log in to your IBM Cloud dashboard.



Use the link given to signup or login to IBM cloud. https://cloud.ibm.com/registration

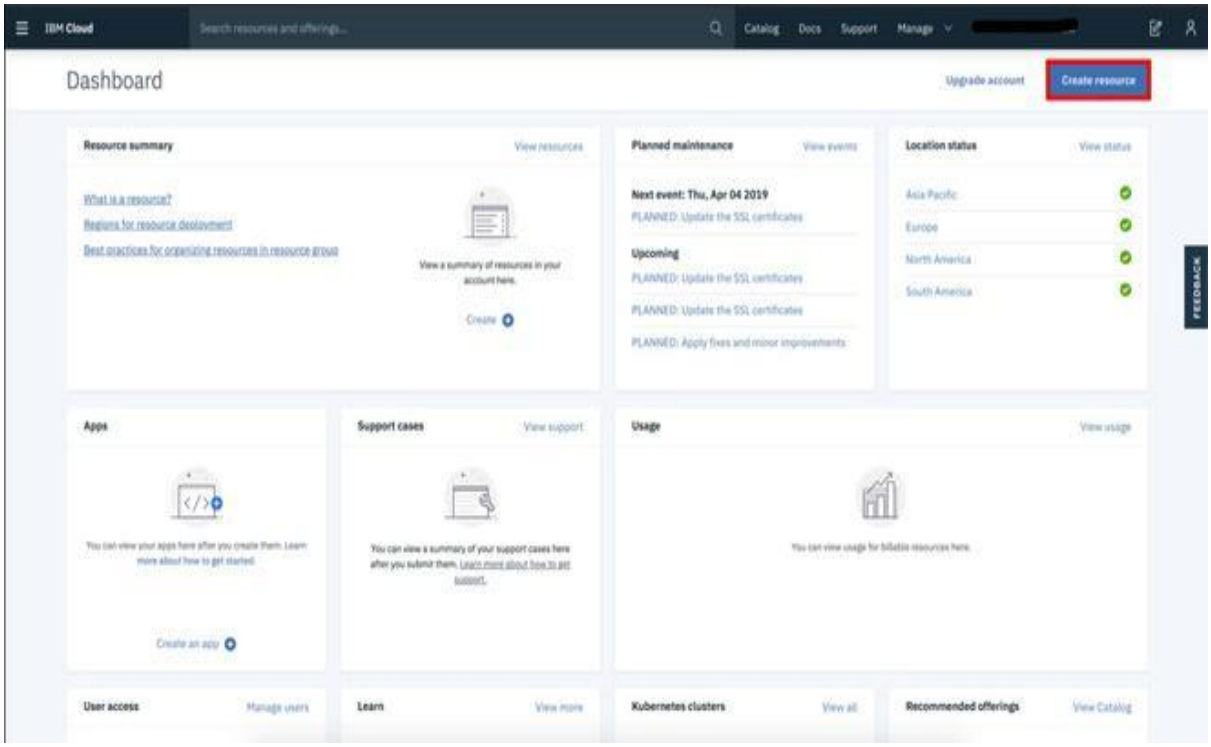
Step 2: Confirm Your Email Address

An email is sent to your email address to confirm your account. Go to your email account, and click on the “Confirm Account” link in the email that was sent to you.

Step 3: Login to Your Account

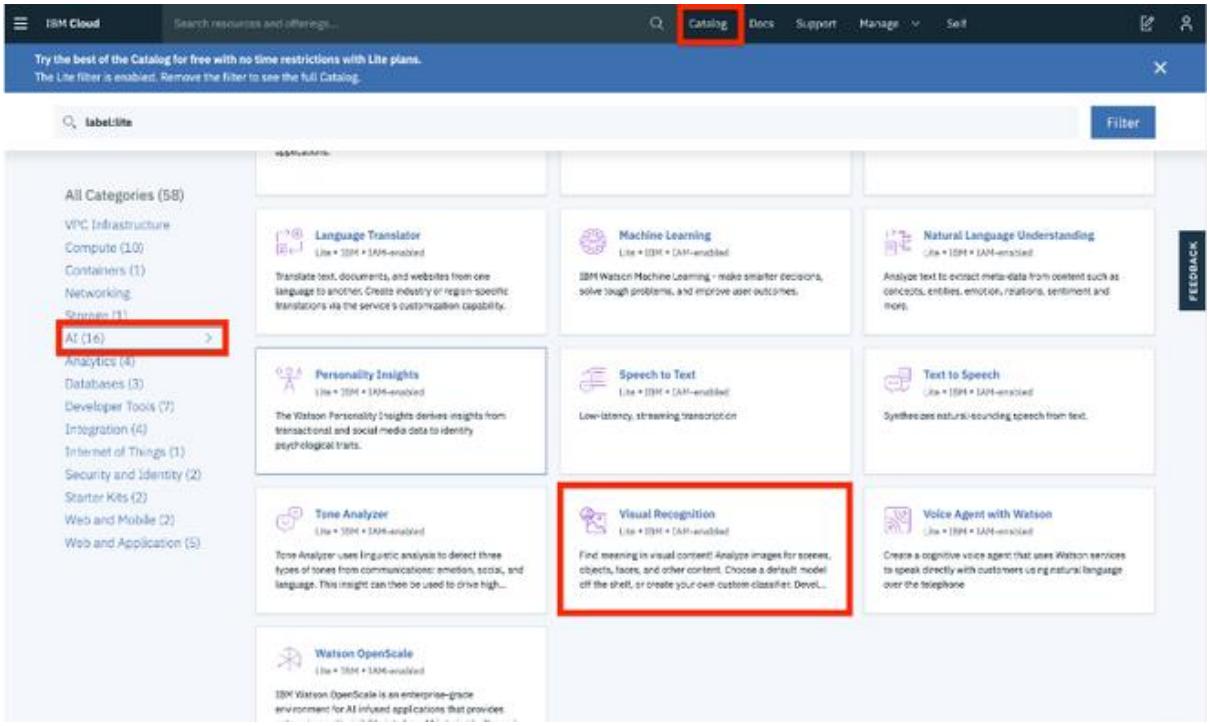
Step 4: Create a New Resource

On your dashboard page, click on the **Create a resource** on the top right to create a new source.

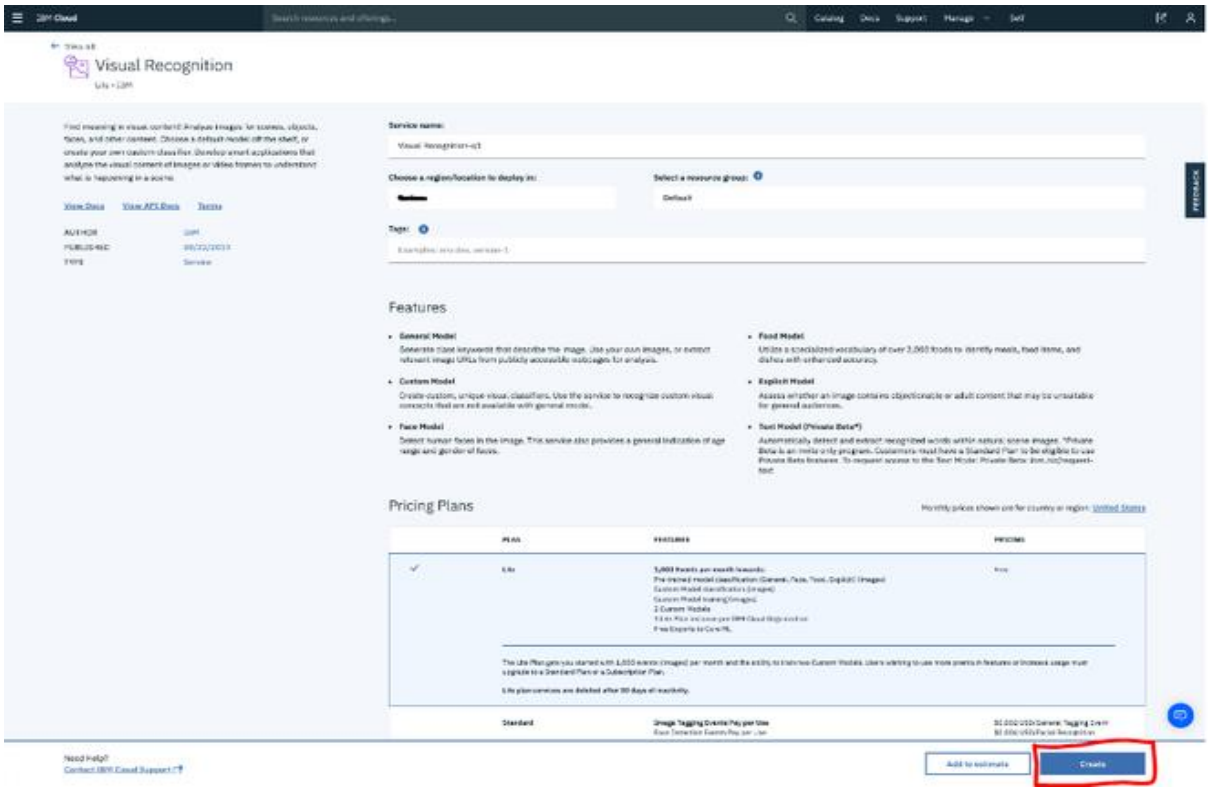


Step 5: Create a Visual Recognition and Watson Studio Resource

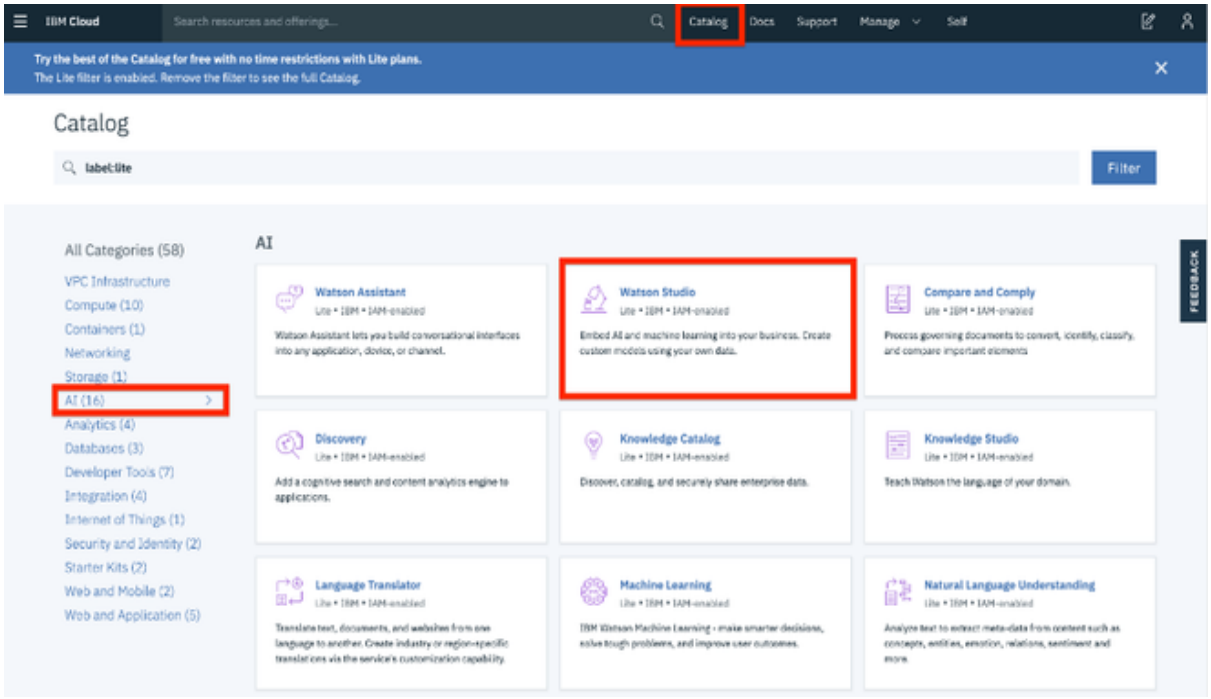
On the Catalog page, select the **AI** category from the left pane, and then select the **Visual Recognition** resource.



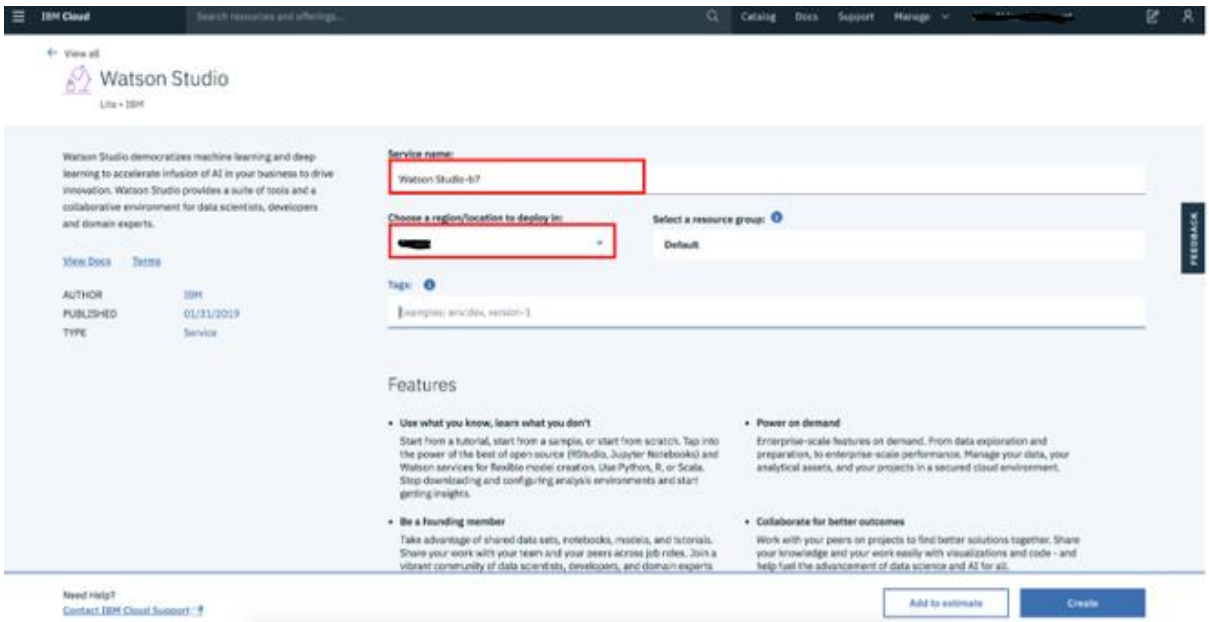
On the next page, you will get to name your service instance and choose your region. Click on the arrow to reveal the drop-down menu of regions. Make sure to select the region that is closest to you. Then scroll down and make sure that the **lite** plan is selected, and click the **Create** button.



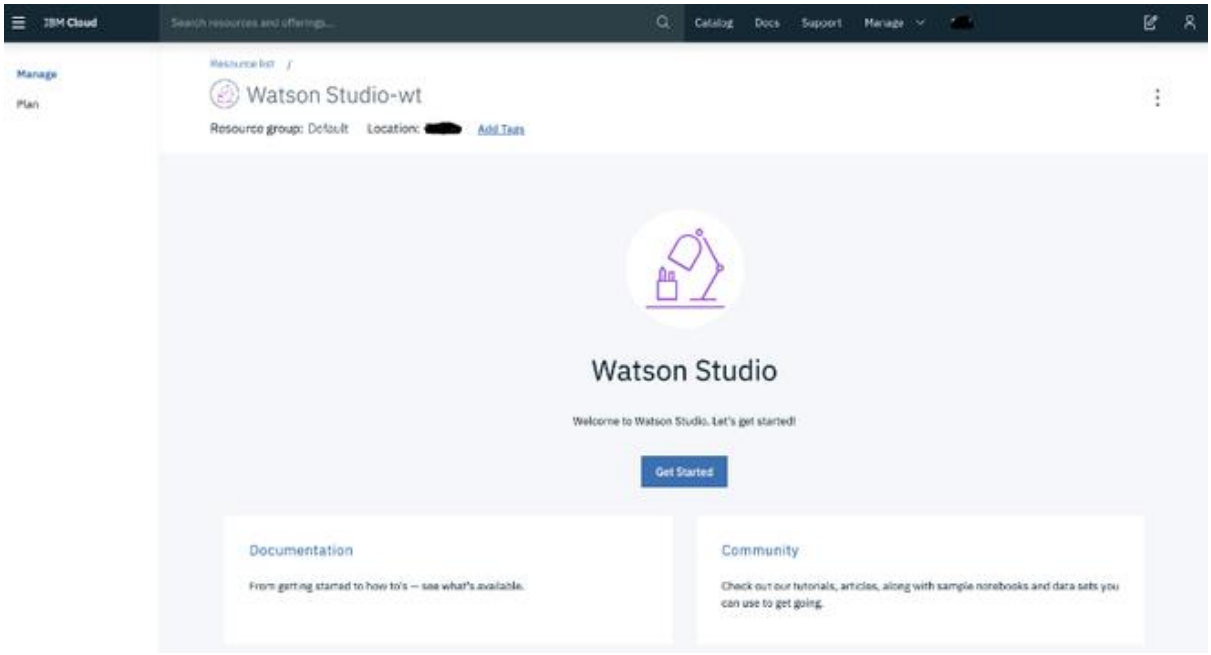
On the Catalog page, select the **AI** category from the left pane, and then select the **Watson Studio** resource.



On the next page, you will get to name your service instance and choose your region. Click on the arrow to reveal the drop-down menu of regions. Make sure to select the region that is closest to you. Then scroll down and make sure that the **lite** plan is selected, and click the Create button.



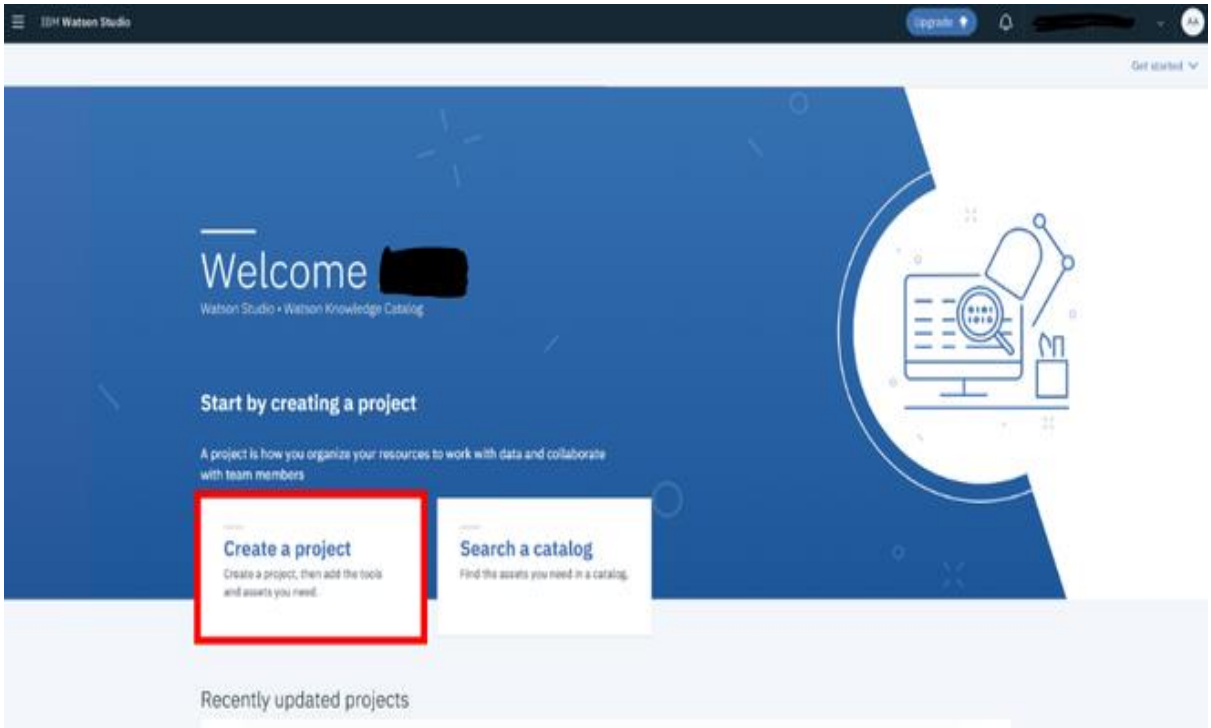
On the next page, click the **Get Started** button to start using Watson Studio.



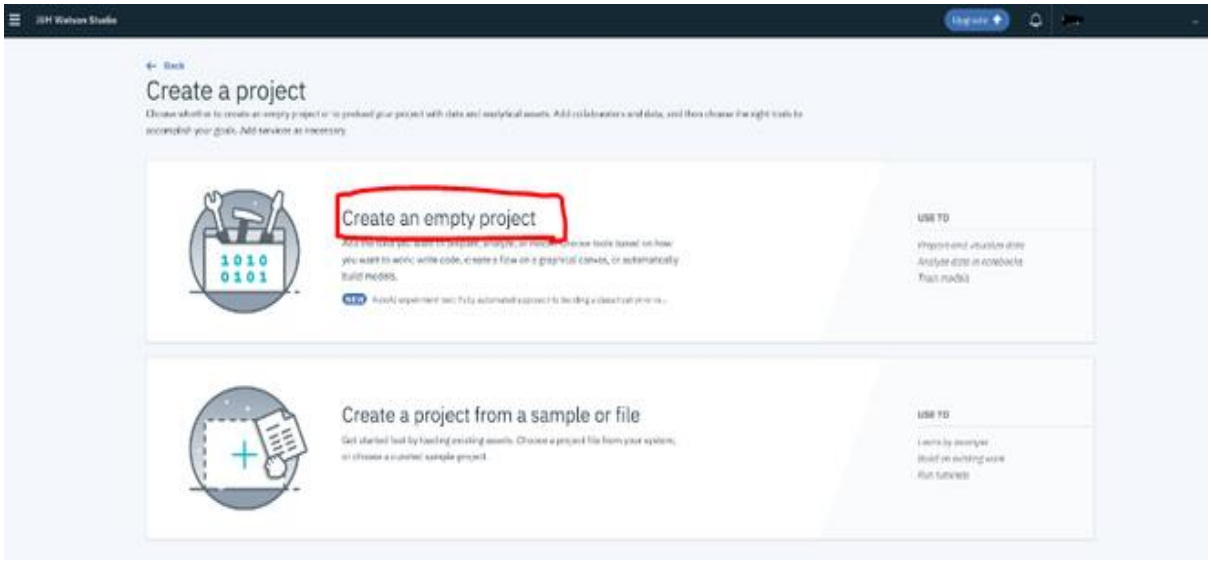
Once the provisioning process is complete, click the **Get Started** button to start using Watson Studio.

Step 6: Create a Project

Once you land on the IBM Watson Studio main page, start by **creating a project.**



Choose create empty project.



Step 7: Set up Your Project

Gather and Prepare Your Image Data

Before you can start training your image recognition model, you need a dataset of labeled images for each category or class you want to recognize. Make sure your image dataset is properly labeled and organized.

When gathering and preparing image data for building an image recognition system using IBM Cloud Visual Recognition, you can use your own images or consider using IBM Cloud Object Storage to store and manage your data. Here's a step-by-step guide on how to gather and prepare image data using IBM Cloud's Object Storage:

1. Define Your Categories:

- Begin by defining the categories or classes that you want your model to recognize. These will serve as the labels for your images. For example, if you're building a system to recognize different types of fruits, your categories could be "Apples," "Bananas," "Oranges," etc.

When gathering and preparing image data for building an image recognition system using IBM Cloud Visual Recognition, you can use your own images or consider using IBM Cloud Object Storage to store and manage your data. Here's a step-by-step guide on how to gather and prepare image data using IBM Cloud's Object Storage:

1. Define Your Categories:

- Begin by defining the categories or classes that you want your model to recognize. These will serve as the labels for your images. For example, if you're building a system to recognize different types of fruits, your categories could be "Apples," "Bananas," "Oranges," etc.

2. Collect Image Data:

- Gather a substantial number of images for each category. The number of images you collect will depend on the complexity of the recognition task, but aim for a minimum of several hundred images per category.

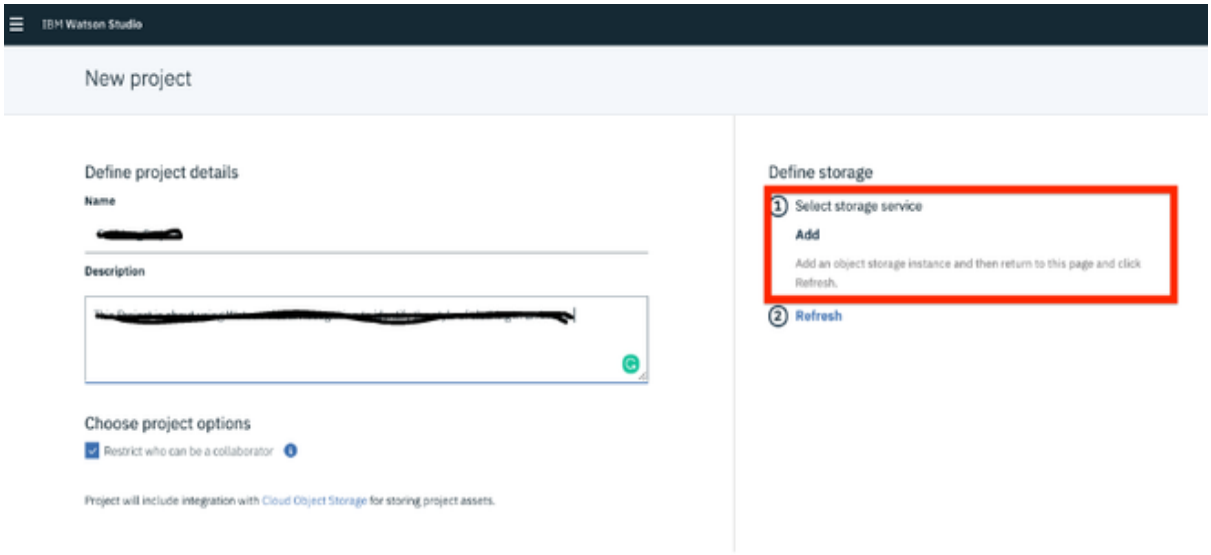
3. Organize Image Data:

- Organize your image data into separate folders, with each folder representing a category. You can label your image files using a consistent naming convention that includes the category label. For example, "apple001.jpg," "apple002.jpg," "banana001.jpg," and so on.

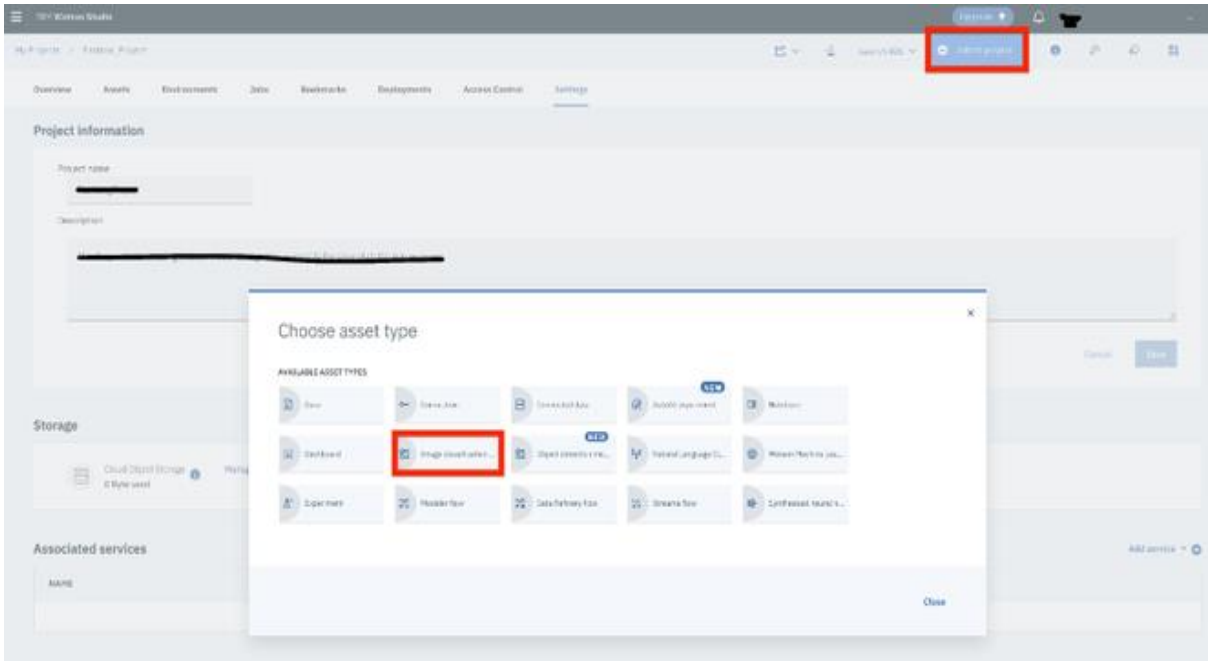
4. Upload Data to IBM Cloud Object Storage:

- Log in to your IBM Cloud account and navigate to the IBM Cloud Object Storage service.
- Create a new bucket in IBM Cloud Object Storage to store your image data. You can create separate buckets for training, validation, and testing data if desired.
- Upload your organized image data to the corresponding buckets in the Object Storage. You can use the web interface to upload files or use tools like **ibmcloud** CLI or a third-party object storage explorer for large datasets.

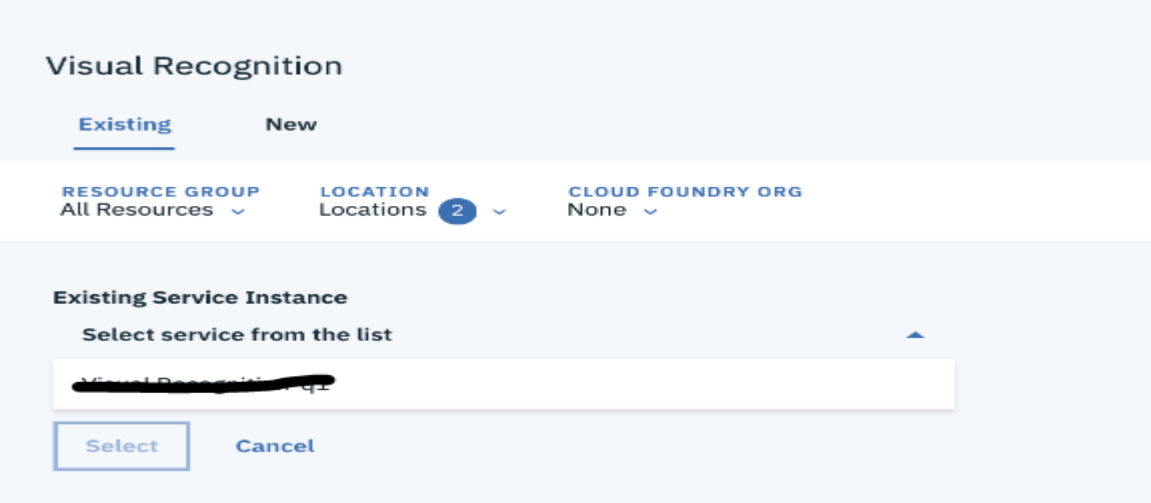
fill in some project details and click **Create**. The IBM Cloud Object Storage, which provides you storage for your images, should be automatically created for you but if not created you can click on add button below the heading on top right part of screen. if it is already created it will show you name and will not let you create a new one.



Create the storage service with the **lite** plan and click on **create** button. Go to **Add to project** and choose **Image classification**.

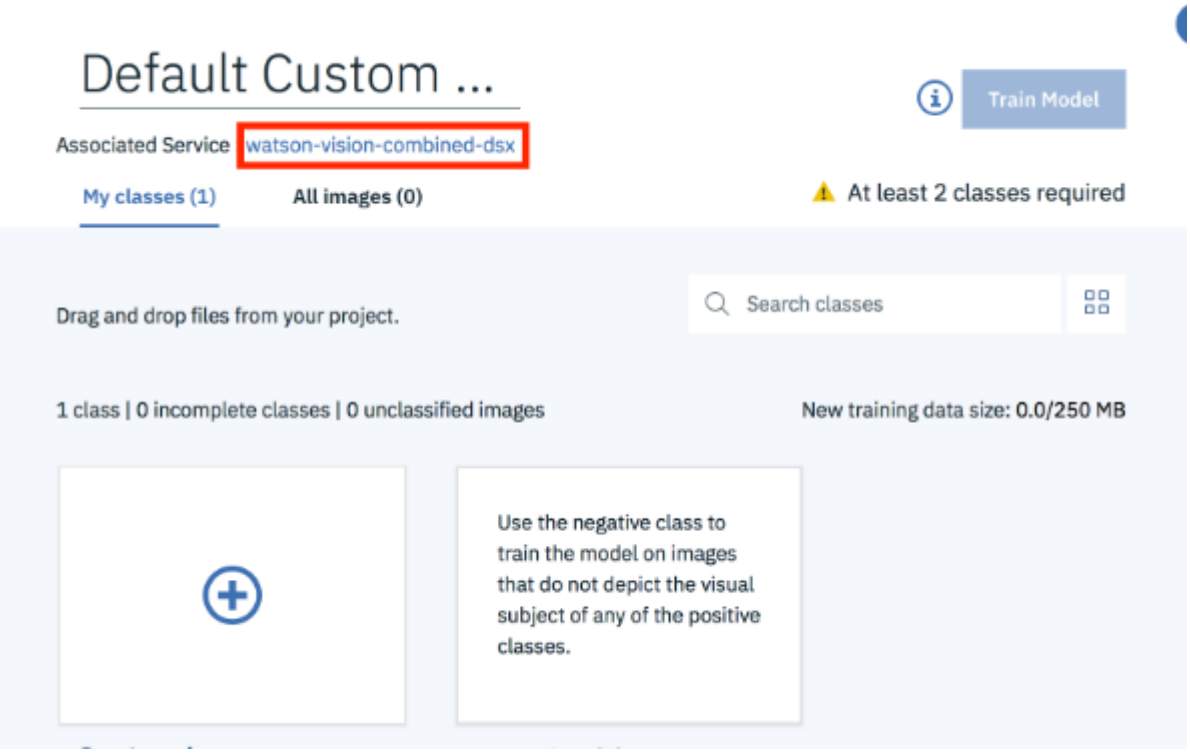


Then provision a visual recognition service.



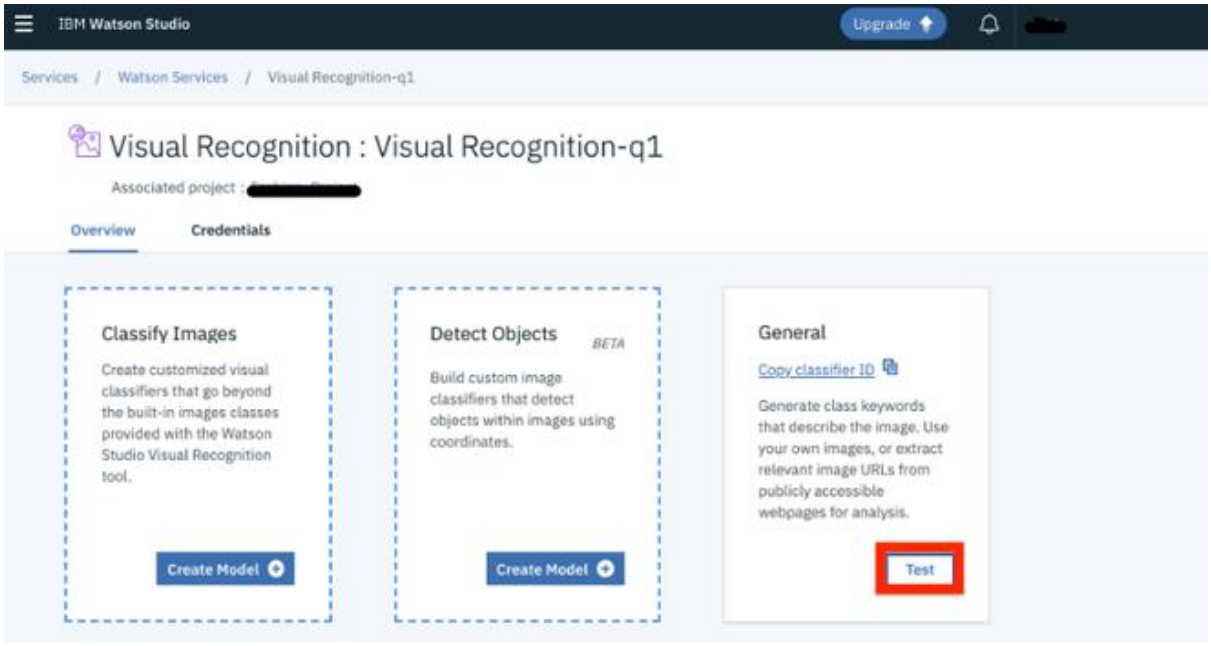
Step 8: Selecting Built-in Models for Watson Visual Recognition

After creating your project, by default, you will land on the page where you can perform some advanced tasks but we will skip this for now and use the built-in models. To access the built-in models, click on the name of the service, as seen in the red box below:



Step 9: Choose the General model.

Now you can see all the built-in image classification models that IBM Watson provides! Let's try the **General** model.

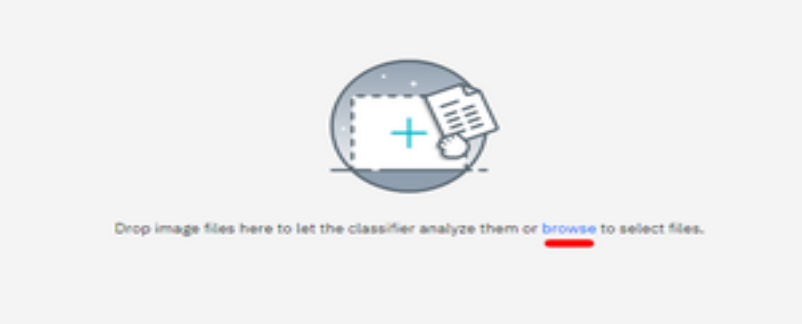


Step 10: Try out the General model

To test the General model, click on **Test** tab on top of screen.

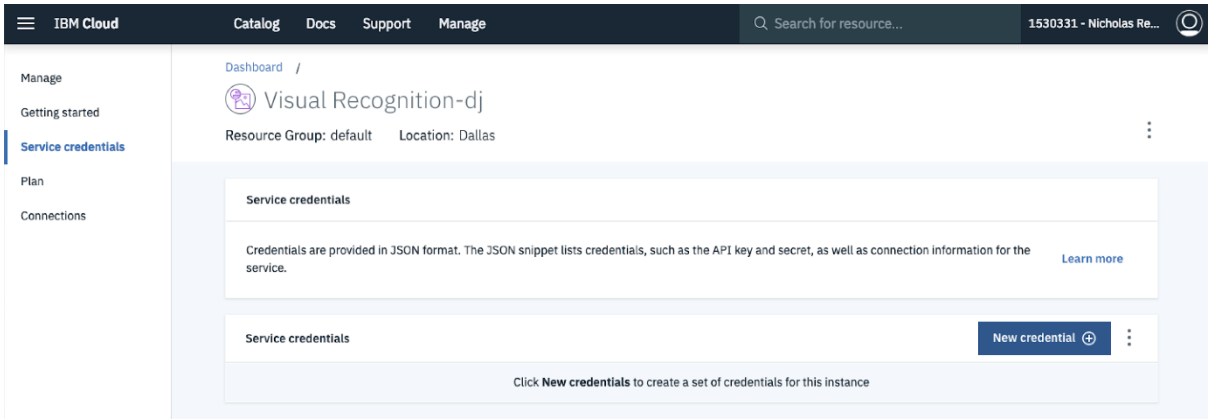
Step 11: Upload Your Images!

Now you can upload any images you'd like by clicking on **Browse**. Remember that you will have to upload all the images that you want to test in a single go as once you upload images then you cannot add more images one by one. you will have to go back and upload all images including the new ones.

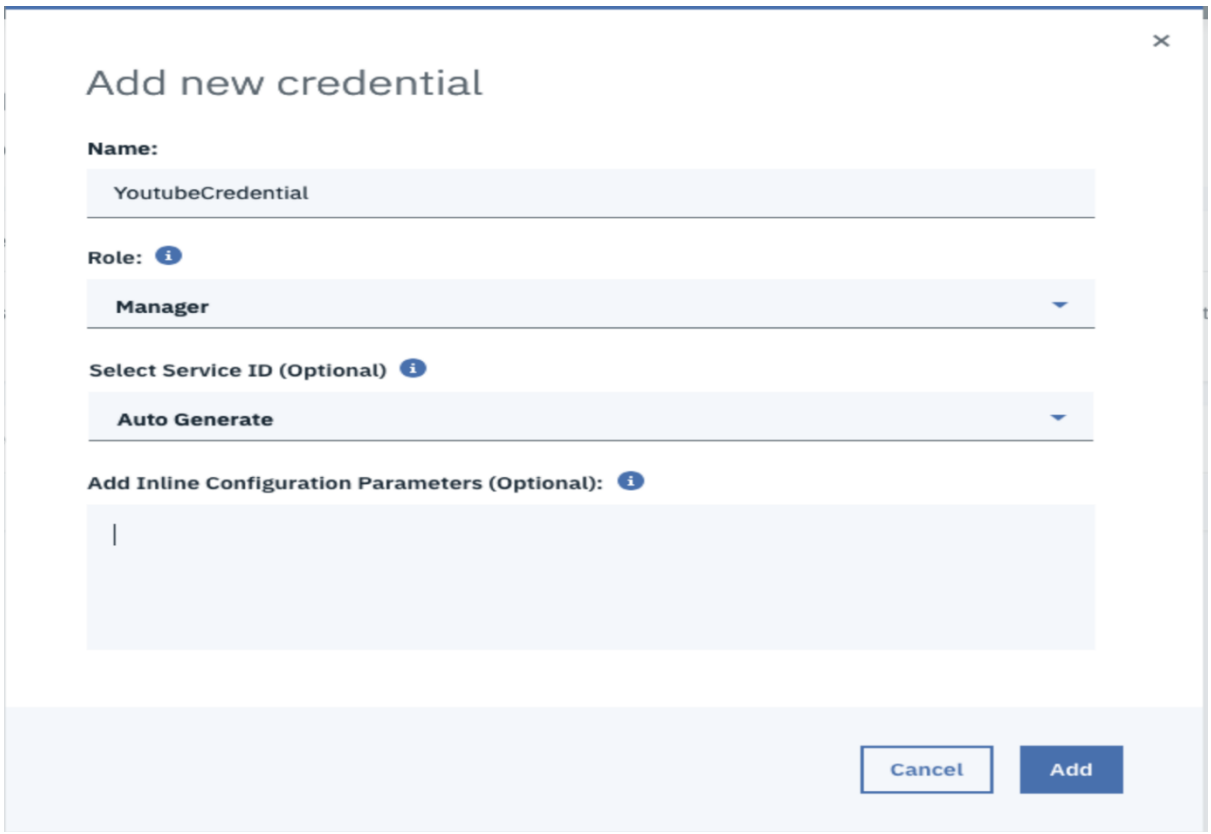


STEP 12 – GENERATE API CREDENTIALS

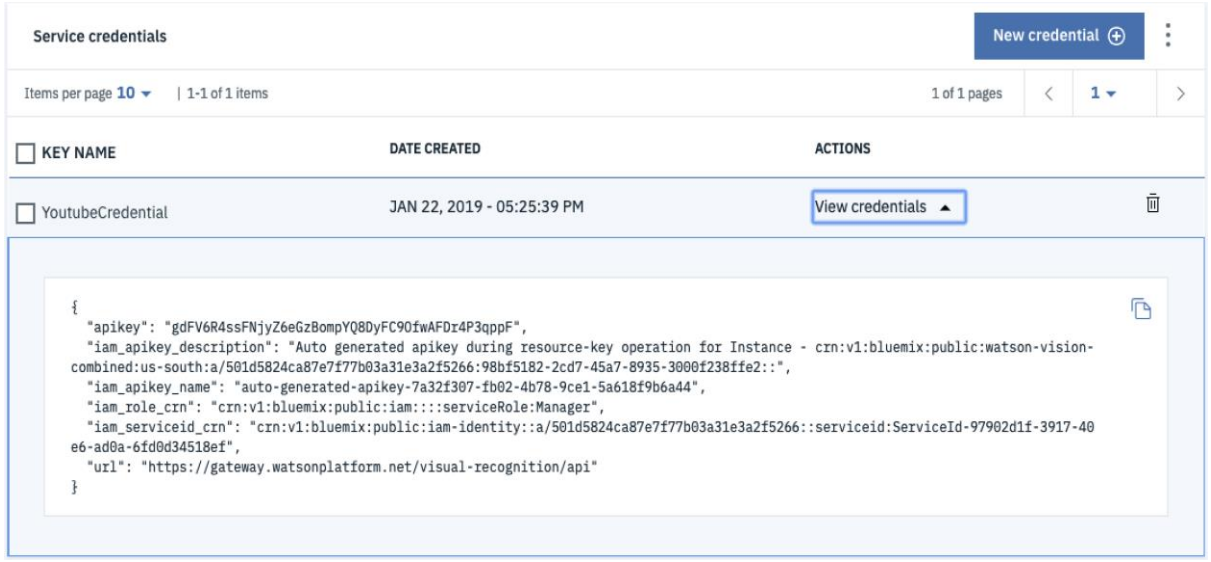
The Visual Recognition API uses token-based IAM to authenticate requests. The IAM credentials can be generated once you've created your service by going to Service Credential then Selecting New Credential.



Once there, create a new credential by entering a name, changing the role to Manager and selecting Auto Generate (optional) for the Select Service ID then hitting Add.



This will generate new credentials which will allow you to use the API when accessing it using Python later on. Click View Credentials and copy the API key.



STEP 13 – CLASSIFYING GENERAL IMAGES

```
# URL that you want to classify

url =
'https://cdn.shopclues.com/images/thumbnails/18729/320/320/78279966PC1427
5391591432900471.jpg'



# Call classify method from service

classes_result = visual_recognition.classify(url=url).get_result()



# Pretty print JSON result

print(json.dumps(classes_result, indent=2))
```

The response can be pretty-printed using json.dumps and should look similar to the result shown below. The image used was a basic desktop computer that looked something like this:

Looking at the classes returned you can see that the classifier accurately classified the image as a desktop computer.

```
# Check classes returned

classes_result['images'][0]['classifiers'][0]['classes']
```

```
# Expected results

[{'class': 'desktop computer',

  'score': 0.959,

  'type_hierarchy': '/machine/computer/digital computer/personal computer/desktop computer'},

 {'class': 'personal computer', 'score': 0.977},

 {'class': 'digital computer', 'score': 0.977},

 {'class': 'computer', 'score': 0.984},

 {'class': 'machine', 'score': 0.984},

 {'class': 'system', 'score': 0.77},

 {'class': 'coal black color', 'score': 0.901}]
```

USER INTERFACE:

Design a simple web service interface where users can upload images and view the AI generated captions.

Creating a simple web interface for users to upload images and view AI-generated captions can be achieved using HTML, CSS, and JavaScript. Below is a basic example of how you can design such an interface:

HTML (index.html):

```
<!DOCTYPE html>
<html>
<head>
    <title>Image Caption Generator</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <h1>Image Caption Generator</h1>
    <input type="file" id="imageInput" accept="image/*" />
    <button id="generateCaption">Generate Caption</button>
    <img id="uploadedImage" src="" alt="Uploaded Image" />
    <div id="captionResult">
        <h2>Generated Caption:</h2>
        <p id="captionText"></p>
      </div>
    <script src="script.js"></script>
</body>
</html>
```

CSS (styles.css):

```
body {
    text-align: center;
    font-family: Arial, sans-serif;
}

h1 {
    margin-top: 20px;
}
input[type="file"] {
    display: none;
}
button {
    padding: 10px 20px;
    background-color: #007BFF;
    color: white;
    border: none;
    cursor: pointer;
}
img {
```

```
    max-width: 100%;
    max-height: 400px;
    display: none;
}
#captionResult {
    margin-top: 20px;
    display: none;
}
#captionText {
    font-style: italic;
}
```

JavaScript (script.js):

```
document.getElementById("generateCaption").addEventListener("click", function() {
    const imageInput = document.getElementById("imageInput");
    const uploadedImage = document.getElementById("uploadedImage");
    const captionResult = document.getElementById("captionResult");
    const captionText = document.getElementById("captionText");
  // Ensure an image is selected
    if (imageInput.files.length === 0) {
        alert("Please select an image to generate a caption.");
```

```
        return;
    }
const file = imageInput.files[0];
    const reader = new FileReader();
 reader.onload = function(event) {
        uploadedImage.src = event.target.result;
        uploadedImage.style.display = "block";

        // You can now send the uploaded image to your AI service for caption generation.
        // Replace the following line with your actual AI integration code.
        const generatedCaption = "A beautiful image.";
 captionText.textContent = generatedCaption;
        captionResult.style.display = "block";
    };
 reader.readAsDataURL(file);
});

// You can also add functionality to clear the uploaded image and caption if needed.
```

This code provides a simple web interface for users to upload an image, generate a caption (currently hardcoded), and display the result. It uses HTML for the structure, CSS for styling, and JavaScript for interaction. To make this code functional with AI-generated captions, you'll need to integrate your AI service's capabilities where indicated in the JavaScript code. This typically involves making an API request to the AI service for caption generation based on the uploaded image.

CONCLUSION:

In this journey, it's important to recognize that the success of your image recognition system depends not only on the technology but also on the quality of your data. The larger, more diverse, and well-labeled your dataset is, the better your model's performance is likely to be.

Ultimately, IBM Cloud Visual Recognition offers a flexible and scalable platform for developing custom image recognition solutions. Whether you're building applications for content moderation, visual search, or object identification, this technology opens the door to a wide range of AI-driven possibilities. By following best practices and continuously refining your model, you can harness the power of image recognition to enhance user experiences and provide valuable insights to your users.