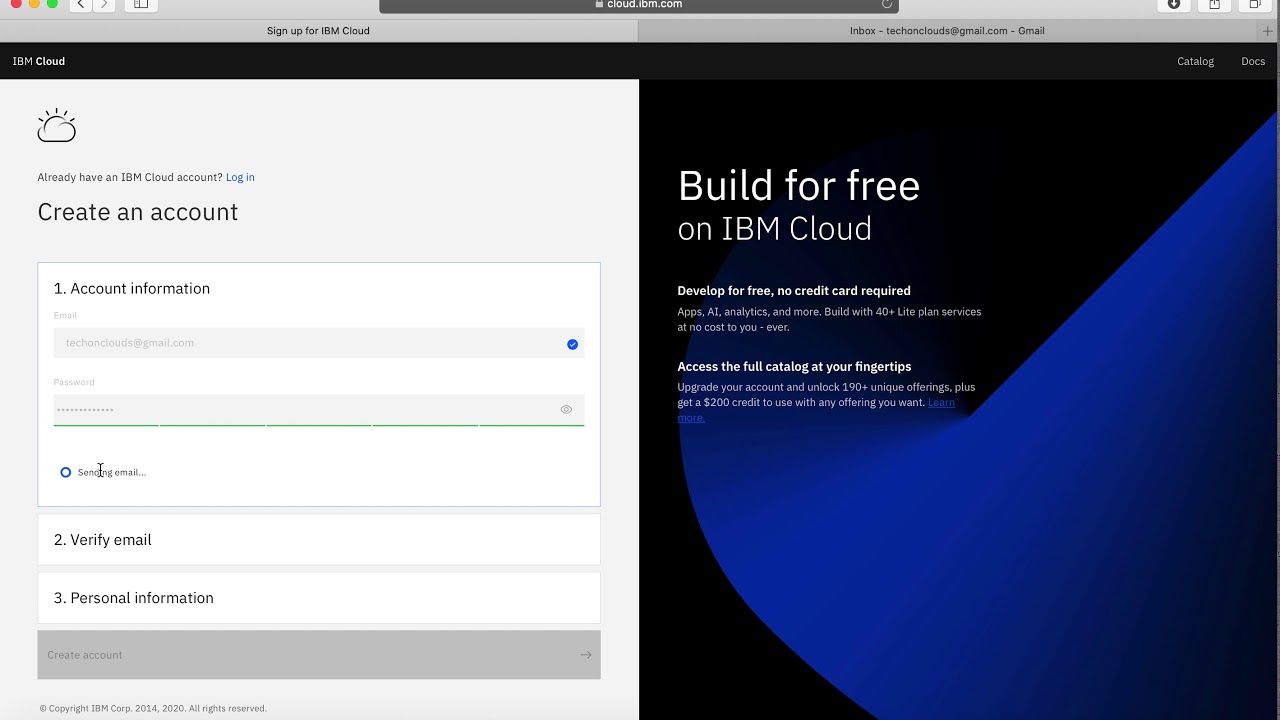
**IMAGE RECOGNITION USING IBM CLOUD VISUAL RECOGNITION**

**Given problem statement:**

* + Create an IBM Cloud account, set up the Visual Recognition service and obtain API keys.
  + Design a simple web interface where users can upload images and view the AI-generated captions.

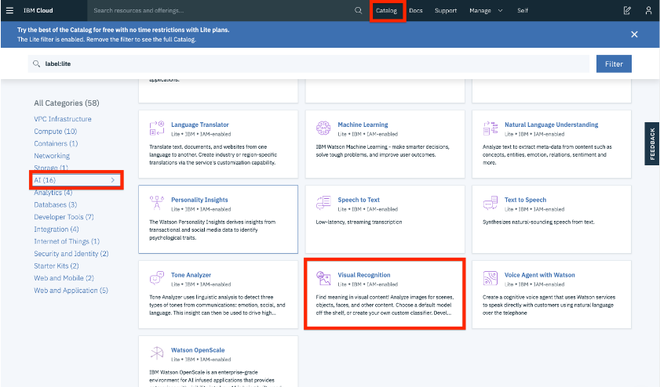
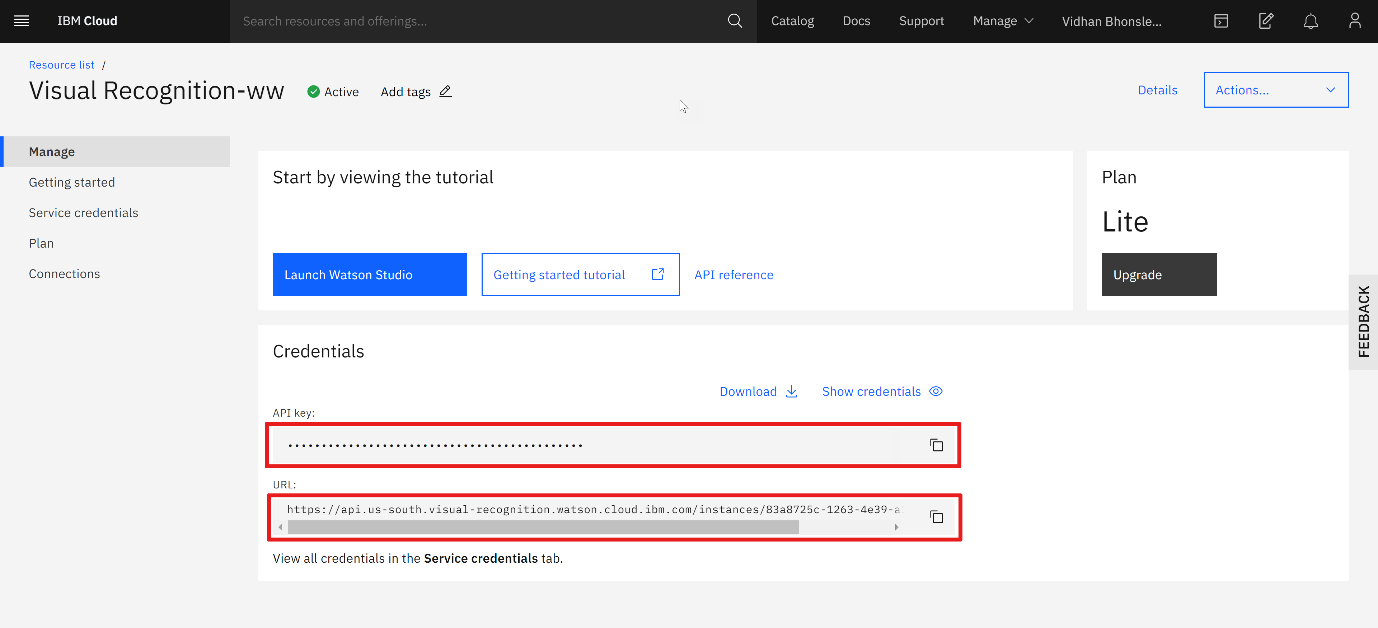
Step 1: Create an IBM account

* Enter your email address, and create a password.
* Complete any necessary verification steps.
* Provide additional information.
* Read and accept the terms and conditions of using IBM Cloud services.
* Click on the "Create Account"



Step 2: Set up of Visual recognition service and obtain API keys

* Go to the IBM Cloud Catalog.
* In the catalog, search for "Visual Recognition".
* Click on the Visual Recognition service to configure it.
* Provide a unique name for your service instance.
* Choose a region for your service (e.g., Dallas, London).
* Once the service instance is created, go to the service dashboard.
* Find the "Manage" tab, and from there, you can get your API key and the service URL.
* Explore the API documentation provided by IBM Cloud for Visual Recognition.



Step 3: Design of simple web interface

* Navigate to the "Catalog" and find the "Visual Recognition" service.
* Once the Visual Recognition service is created, obtain the API key and endpoint from the IBM Cloud dashboard.
* Install Necessary Libraries: Install Flask for building the web application

pip install Flask

* Install the IBM Watson SDK for Python.



* Create a Flask web application with routes for handling image uploads and classification results.

from flask import Flask, render\_template, request

from ibm\_watson import VisualRecognitionV4

from ibm\_watson.visual\_recognition\_v4 import FileWithMetadata

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/upload', methods=['POST'])

def upload():

# Handle image upload and send it to Visual Recognition

# Use the obtained API key and endpoint

# Parse the classification result

return render\_template('result.html', result=result)

* Design HTML templates for the home page (index.html) and the result page (result.html). Include a form for image upload on the home page.

<!-- index.html -->

<html>

<body>

<h1>Image Classification</h1>

<form action="/upload" method="post" enctype="multipart/form-data">

<input type="file" name="file">

<input type="submit" value="Upload">

</form>

</body>

</html>

<!-- result.html -->

<html>

<body>

<h1>Classification Result</h1>

<p>{{ result }}</p>

</body>

</html>

* In the Flask route for image upload, use the IBM Watson Visual Recognition API to classify the uploaded image.

@app.route('/upload', methods=['POST'])

def upload():

file = request.files['file']

visual\_recognition = VisualRecognitionV4(

version='2021-09-18',

iam\_apikey='your\_api\_key',

url='your\_endpoint'

)

with open(file.filename, 'rb') as image\_file:

classes = visual\_recognition.classify(

images\_file=FileWithMetadata(image\_file)

).get\_result()

# Parse the classification result

result = parse\_result(classes)

return render\_template('result.html', result=result)

• Run your Flask application and test it locally.

flask run