

**IMAGE RECOGNITION WITH IBM
CLOUD VISUAL RECOGNITION
PHASE 3**

OVERVIEW

Certainly! Here's an overview of the steps to build an image recognition system using IBM Cloud Visual Recognition and design a simple web interface for users to upload images and view AI-generated captions

IBM CLOUD VISUAL RECOGNITION

CREATE A IBM CLOUD

IBM Cloud

Verify identity

Apply code [Register without a code](#)

Enter the code that has been provided to you. You can apply the code to only one account, and it can't be removed.

Enter code

a83987c40f9b9a3e09c59c01a446322

Account ID: 9d53b18aa3454d160a0c86389fc9a5cc

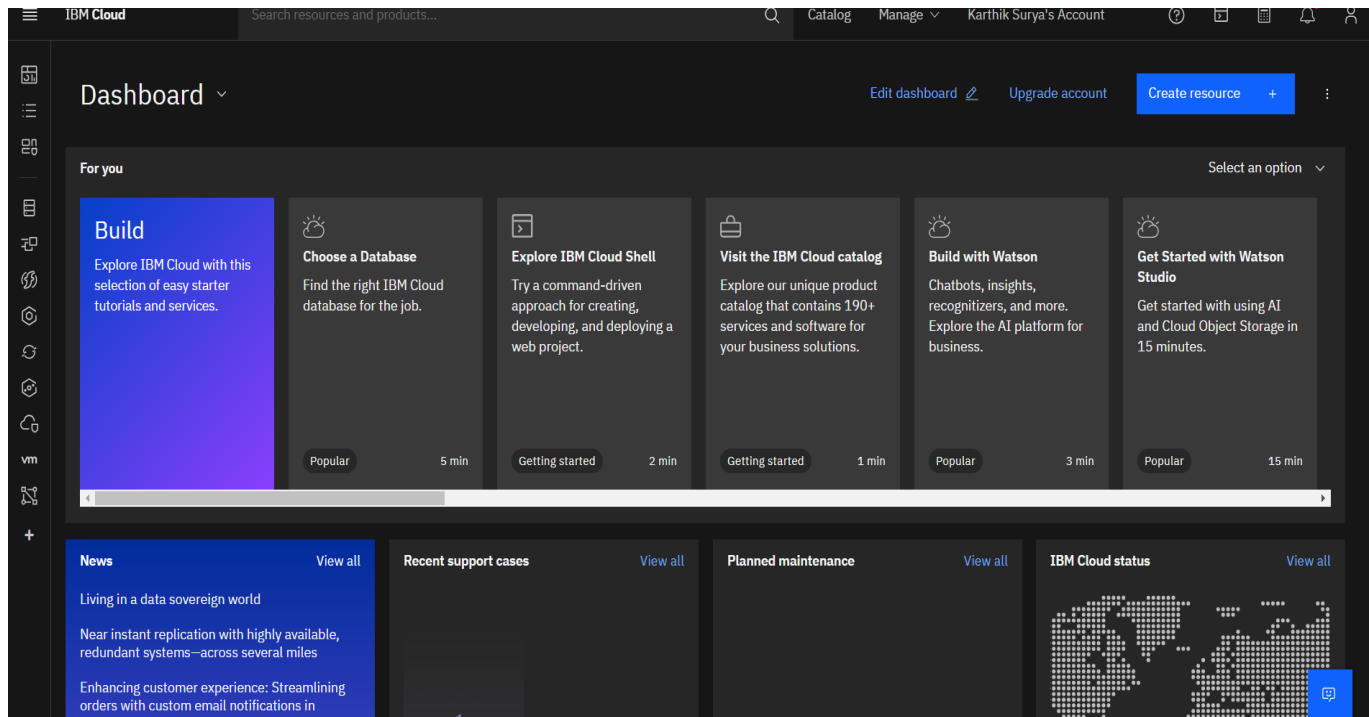
Secure your account

You're asked to verify your identity by entering credit card information.

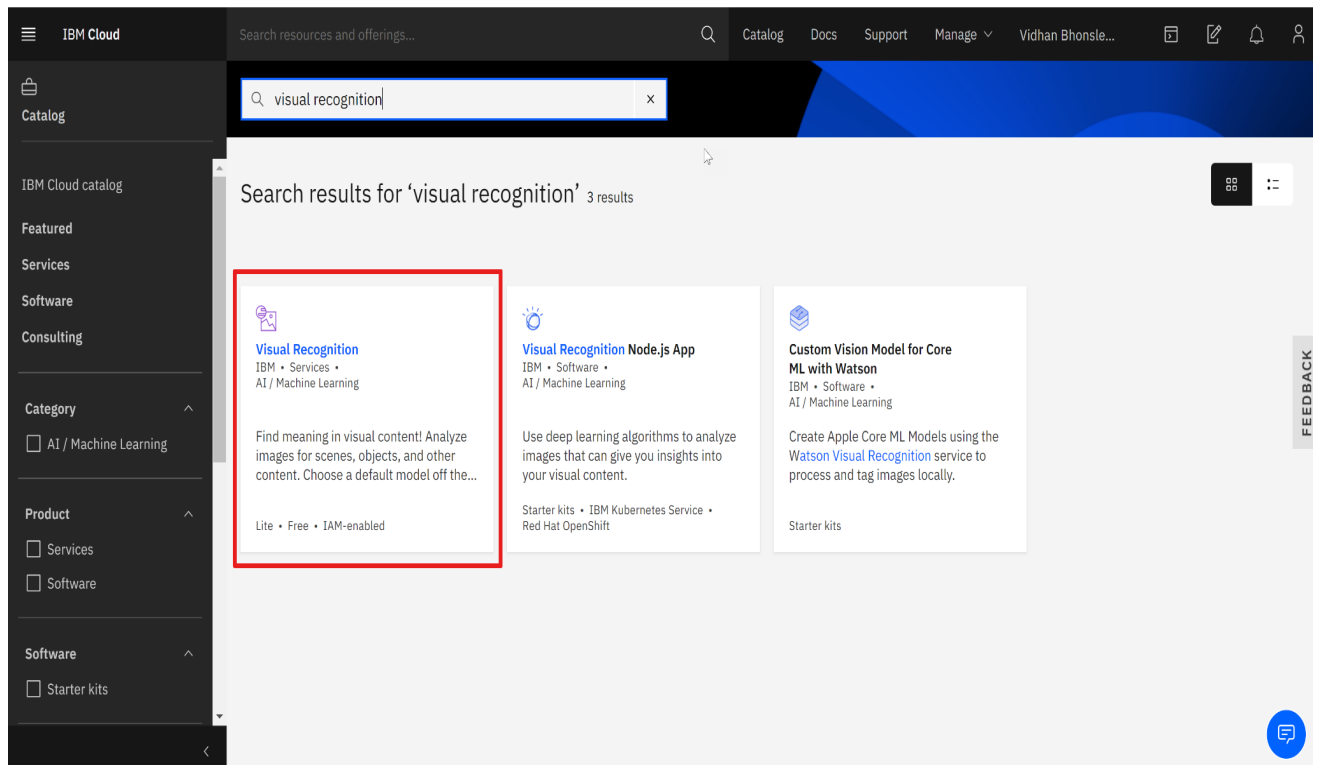
Create your account for free

Access over 40 always-free products. You won't be charged for any usage below the IBM cloud free tier limits.

Create account



VISUAL RECOGNITION AND API KEY



```
watsonvr:
  - name: "Account 1"
    credentials:
      authType: "BASIC"
      apiKey: "2Abcdefghijk_lmn37oP0qrSTUvwXZmUUpa-6E4AGjt2"
      apiVersion: "v3"
      versionDate: "2018-03-19"
      loggingFlag: "true"
    endpoint: {}
```

CODE TO CREATE WEBSITE

STEP 1 CREATE A HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Image Recognition</title>
</head>
<body>
  <h1>Image Recognition</h1>
  <input type="file" id="imageInput" accept="image/*">
  <button id="analyzeButton">Analyze Image</button>
  <p>AI-generated Caption: <span id="caption"></span></p>
  <img id="uploadedImage" style="display: none;" />
  <script src="script.js"></script>
</body>
</html>
```

STEP 2: Set Up JavaScript to Call the Visual Recognition API

Create a JavaScript file (script.js) to handle user interactions and call the Visual Recognition API using the API key.

```
document.getElementById('analyzeButton').addEventListener('click', () => {
  const fileInput = document.getElementById('imageInput');
  const image = fileInput.files[0];
```

```

if (!image) {
  alert('Please select an image to analyze.');
```

```

  return;
}

const formData = new FormData();
formData.append('images_file', image);

// Call the Visual Recognition API
analyzeImage(formData);
});

async function analyzeImage(imageData) {
  // Replace with your Visual Recognition API key
  const apiKey = 'YOUR_API_KEY';

  const url =
`https://api.us-south.visual-recognition.watson.cloud.ibm.com/instances/YOUR_INSTANCE_ID/v3/classify?version=2018-03-19`;

  try {
    const response = await fetch(url, {
      method: 'POST',
      headers: {
        'Authorization': `Basic ${btoa(`apikey:${apiKey}`)}`,
      },
      body: imageData,
    });

    if (response.ok) {
      const result = await response.json();
      displayCaption(result);
    } else {
      alert('Error analyzing the image. Please try again.');
```

```

    }
  } catch (error) {
    console.error('An error occurred:', error);
  }
}

function displayCaption(result) {
  const caption = result.images[0].classifiers[0].classes[0].class;
  const captionElement = document.getElementById('caption');
  captionElement.textContent = caption;
}

```

```

}
// Replace with your Visual Recognition API key
const apiKey = 'YOUR_API_KEY';

// Event listener for the "Analyze Image" button
document.getElementById('analyzeButton').addEventListener('click', () => {
  const fileInput = document.getElementById('imageInput');
  const image = fileInput.files[0];

  const formData = new FormData();
  formData.append('images_file', image);

  // Use the IBM Watson Visual Recognition SDK to analyze the image
  // Include code here to send the image to the Visual Recognition service
  // and display the AI-generated caption in the 'caption' span.
});

```

