Automated Image Labeling using AWS Rekognition

Step 1: Set Up an Amazon S3 Bucket

1. **Log in to AWS Management Console**:
   - Go to the AWS Management Console.
2. **Create an S3 Bucket**:
   - Navigate to the S3 service.
   - Click on "Create bucket".
   - Enter a unique bucket name (e.g., image-labels-bucket).
   - Choose the appropriate region and configure settings as needed.
   - Click "Create bucket".

Step 2: Create an IAM Role

1. **Navigate to IAM**:
   - Go to the IAM service in the AWS Management Console.
2. **Create a Role**:
   - Click on "Roles" and then "Create role".
   - Choose the type of trusted entity as "AWS service" and select "EC2" for the service.
   - Click "Next: Permissions".
3. **Attach Policies**:
   - Attach the following policies:
     - AmazonRekognitionFullAccess
     - AmazonS3FullAccess
   - Click "Next: Tags", then "Next: Review".
4. **Name the Role**:
   - Give the role a name (e.g., RekognitionS3Role) and click "Create role".

Step 3: Install AWS CLI

1. **Download and Install AWS CLI**:
   - Install the AWS CLI on your system.
2. **Configure AWS CLI**:
   - Open your terminal or command prompt.
   - Run aws configure and provide your AWS Access Key ID, Secret Access Key, region, and output format.

Step 4: Write Python Code

1. **Install Required Libraries**:
   - Ensure you have boto3 and matplotlib installed.
2. **Create Python Script**:
   - Create a Python script (e.g., image_labels_generator.py) with the following code:

```
import boto3

import matplotlib.pyplot as plt

from PIL import Image, ImageDraw


def detect_labels(photo, bucket):

    client = boto3.client('rekognition', region_name='ap-south-1')


    # Detect labels with a maximum of 10 labels

    response = client.detect_labels(Image={'S3Object': {'Bucket': bucket, 'Name': photo}},
MaxLabels=10)

    return response['Labels']


def show_bounding_boxes(photo, bucket, labels):

    s3_client = boto3.client('s3')


    # Download the image from S3

    s3_client.download_file(bucket, photo, 'downloaded_image.jpg')

    image = Image.open('downloaded_image.jpg')


    img_width, img_height = image.size

    draw = ImageDraw.Draw(image)


    # Draw bounding boxes for each label with instances

    for label in labels:
```

```
        for instance in label.get('Instances', []):  # Use .get() to avoid KeyError if 'Instances' key is missing

            box = instance['BoundingBox']

            left = img_width * box['Left']

            top = img_height * box['Top']

            width = img_width * box['Width']

            height = img_height * box['Height']

            points = (

                (left, top),

                (left + width, top),

                (left + width, top + height),

                (left, top + height),

                (left, top)

            )

            draw.line(points, fill='#00d400', width=2)


    # Display the image with bounding boxes

    plt.imshow(image)

    plt.axis('off')

    plt.show()


def main():

    bucket = 'image-labels-bucket'  # Your S3 bucket name

    photo = 'Dog Image.jpg'         # Your image file name
```

```
    # Detect labels
    labels = detect_labels(photo, bucket)

    # Filter labels with high confidence
    high_confidence_labels = [label for label in labels if label['Confidence'] > 80]  # Adjust confidence threshold as needed

    # Print detected labels with high confidence
    print("Detected Labels with High Confidence:")
    for label in high_confidence_labels:
        print(f"Label: {label['Name']}, Confidence: {label['Confidence']}")

    # Show bounding boxes for high confidence labels
    show_bounding_boxes(photo, bucket, high_confidence_labels)

if __name__ == "__main__":
    main()
```

3. **Upload Images to S3**:
    - Upload images to the S3 bucket you created earlier.

Step 5: Run the Python Script

1. **Ensure Your IAM Role Has the Correct Permissions**:
    - Make sure the EC2 instance or the environment where you run this script has the necessary permissions to access Amazon Rekognition and S3.