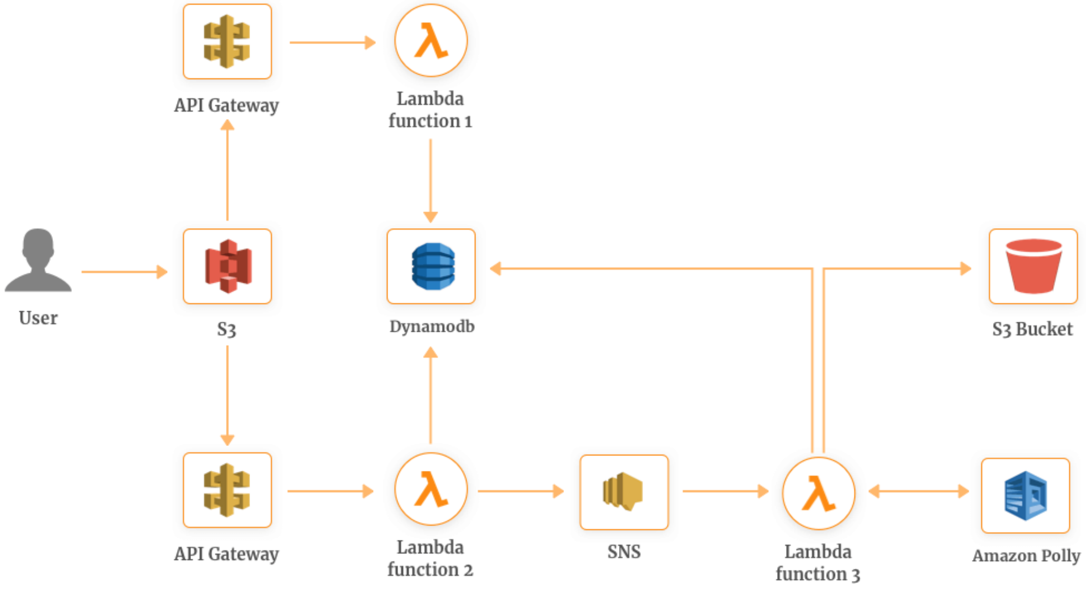
Conversational Experience - Alexa Integration to Fetch Loan Level Details

Presented by

Gaurav Kumar



Acknowledgement

- I would like to express my heartfelt gratitude to Tata Consultancy Services, **Ms. Sumy Kuriakose**(Academic Relationship Manager), **Mr. Samrat Guchhait**(Project Manager), **Mr. Suman Mukherjee**(Mentor) and **Mr. Manasij Dutta**(Mentor) for giving me the opportunity to work as a part of the team and help me gain immensely enriching professional experience. I would also like to thank my fellow project intern **Mr. Sayak Das** for his valuable help in doing this project.

Gaurav Kumar

Intern Id: 1984738

Agenda

- Developed a voice-based recommended system that recognizes the speech pattern of the user and extract relevant keywords and show contents for the same
- AWS Deployment Package in Python: Here I configured AWS lambda, Built the deployment package, Scheduled cron jobs with AWS cloudwatch

Introduction

- The overwhelming success of speech-enabled products like Amazon Alexa has proven that some degree of speech support will be an essential aspect of household tech for the foreseeable future. If you think about it, the reasons why are pretty obvious. Incorporating speech recognition into your Python application offers a level of interactivity and accessibility that few technologies can match.
- The accessibility improvements alone are worth considering. Speech recognition allows the elderly and the physically and visually impaired to interact with state-of-the-art products and services quickly and naturally—no GUI needed!

Basic Speech Recognition in Python

Using pyAudio & speechrecognition package

Implementation

Required Installations

The following must be installed:

1.Python Speech Recognition module:

```
sudo pip install SpeechRecognition
```

2.PyAudio: Command for linux users

```
sudo apt-get install python-pyaudio
python3- pyaudio
```

If the versions in the repositories are too old, install pyaudio using the following command

```
sudo apt-get install portaudio19-dev
python-all-dev python3-all-dev && sudo
pip install pyaudio
```

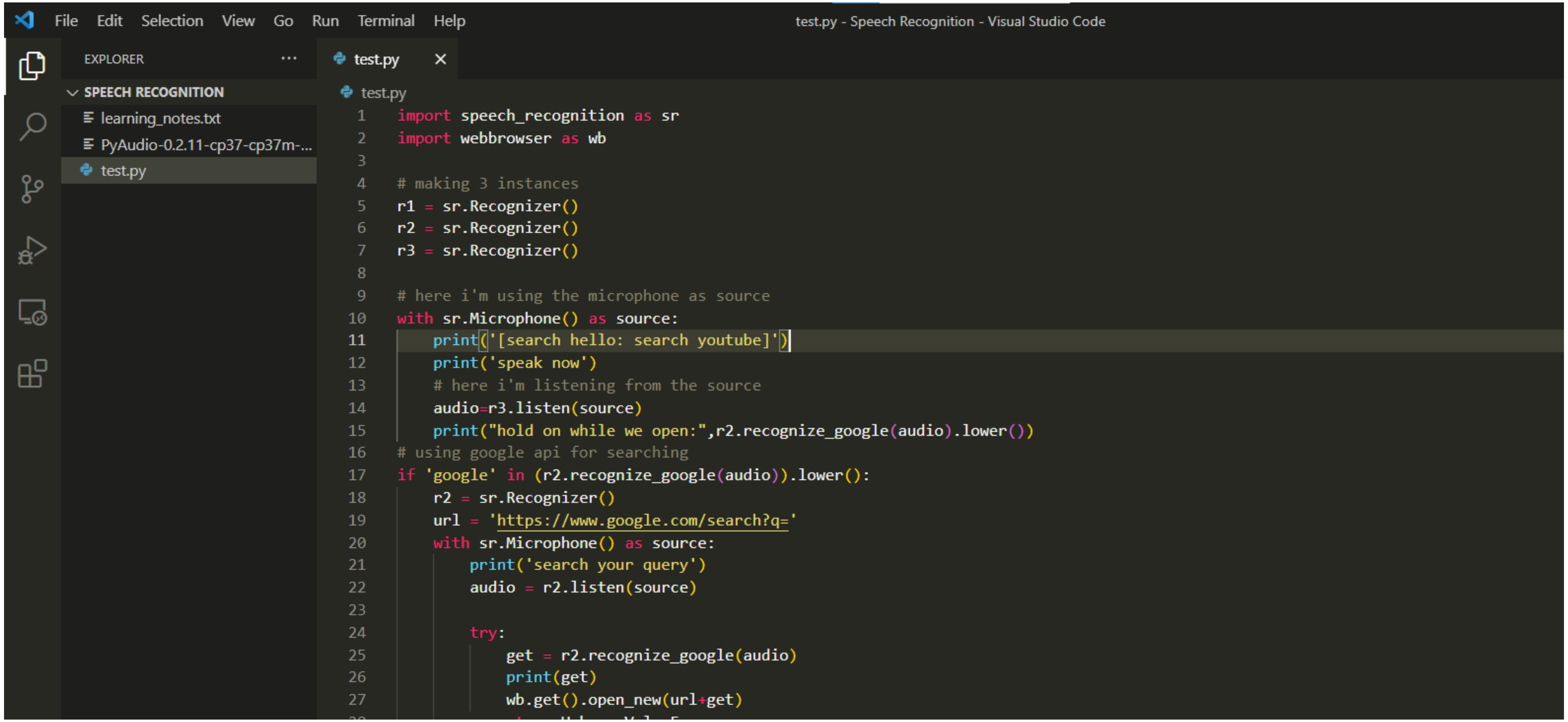
Speech Recognition Libraries

- CMU Sphinx
- Kaldi
- Speech Recognition
- wav2letter++

Speech Recognition Functions

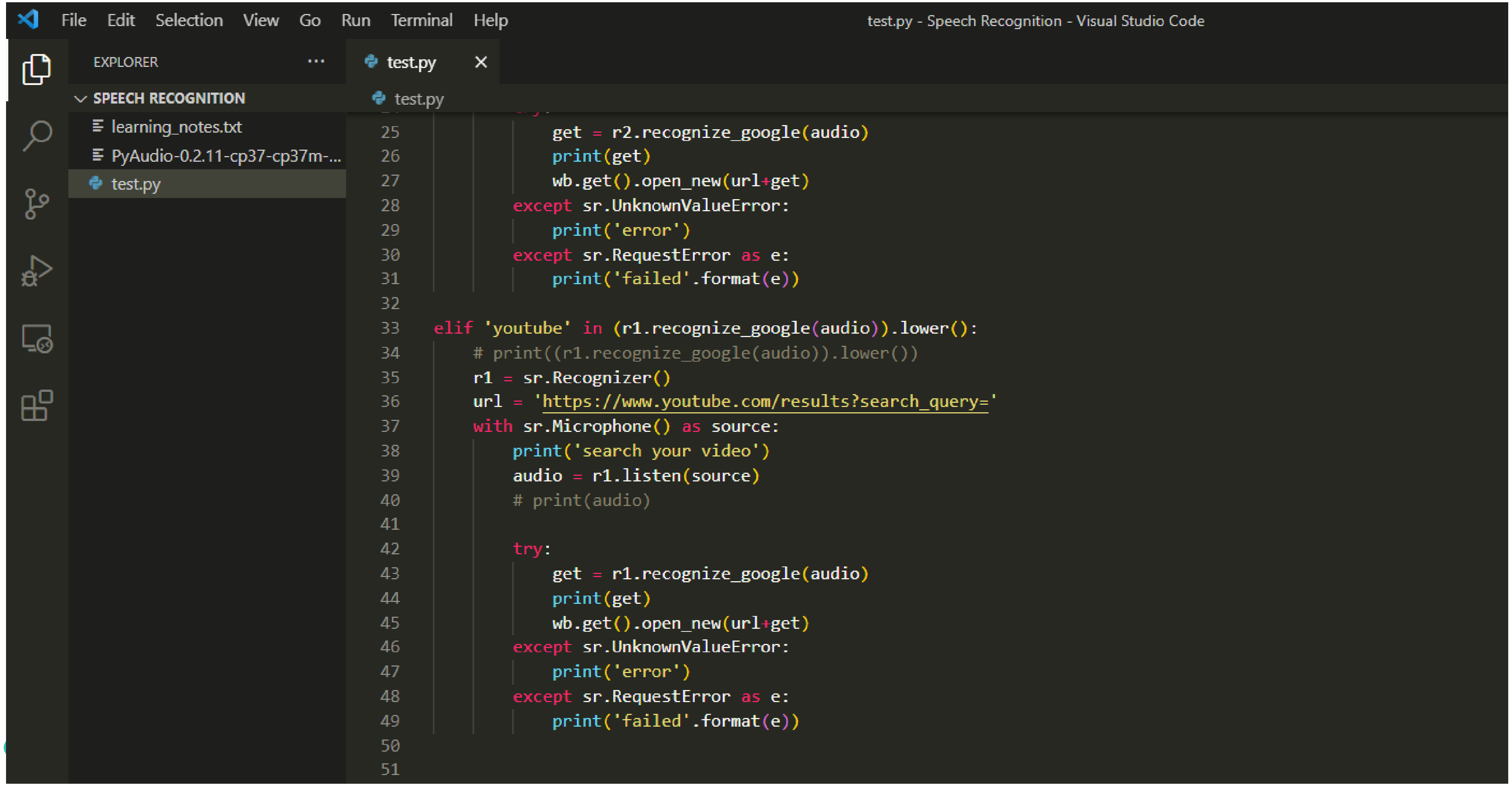
•recognize_bing()
•recognize_google()
•recognize_google_cloud()
•recognize_wit()

Python Code Sample - 1



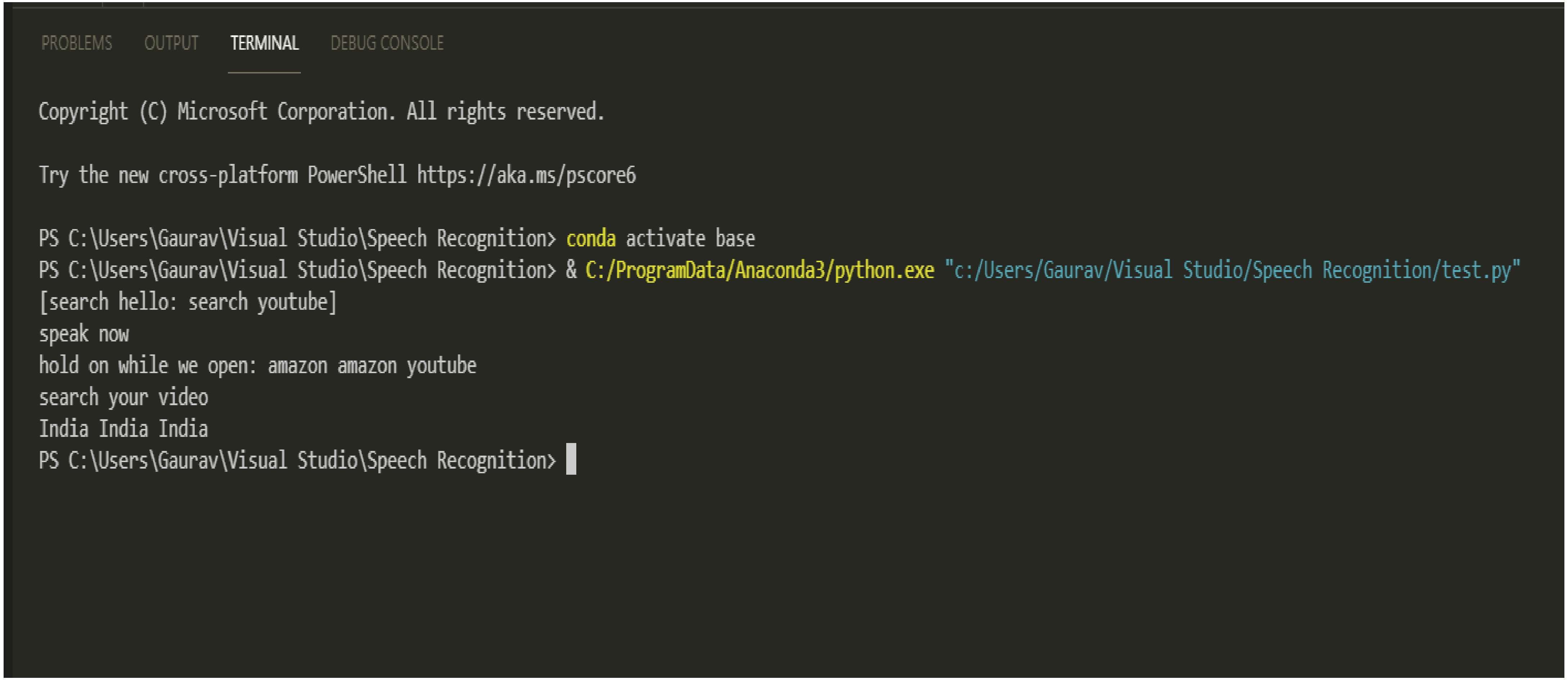
```python
import speech_recognition as sr
import webbrowser as wb

# making 3 instances
r1 = sr.Recognizer()
r2 = sr.Recognizer()
r3 = sr.Recognizer()

# here i'm using the microphone as source
with sr.Microphone() as source:
    print('[search hello: search youtube]')
    print('speak now')
    # here i'm listening from the source
    audio=r3.listen(source)
    print("hold on while we open:",r2.recognize_google(audio).lower())
# using google api for searching
if 'google' in (r2.recognize_google(audio)).lower():
    r2 = sr.Recognizer()
    url = 'https://www.google.com/search?q='
    with sr.Microphone() as source:
        print('search your query')
        audio = r2.listen(source)

        try:
            get = r2.recognize_google(audio)
            print(get)
            wb.get().open_new(url+get)
```

Python Code Sample - 2



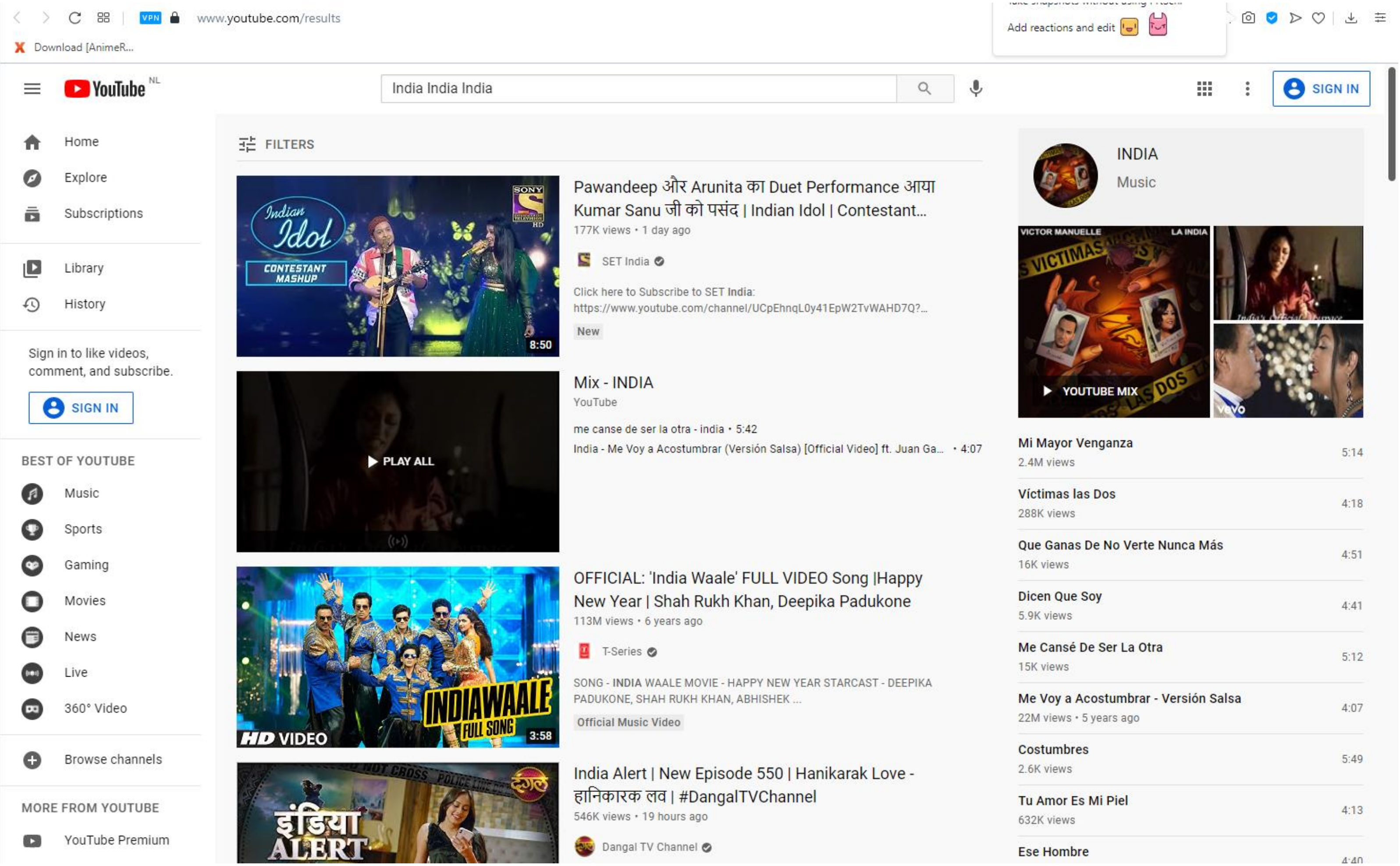
```python
            get = r2.recognize_google(audio)
            print(get)
            wb.get().open_new(url+get)
        except sr.UnknownValueError:
            print('error')
        except sr.RequestError as e:
            print('failed'.format(e))

elif 'youtube' in (r1.recognize_google(audio)).lower():
    # print((r1.recognize_google(audio)).lower())
    r1 = sr.Recognizer()
    url = 'https://www.youtube.com/results?search_query='
    with sr.Microphone() as source:
        print('search your video')
        audio = r1.listen(source)
        # print(audio)

        try:
            get = r1.recognize_google(audio)
            print(get)
            wb.get().open_new(url+get)
        except sr.UnknownValueError:
            print('error')
        except sr.RequestError as e:
            print('failed'.format(e))
```

Python Code Output - 1



```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Gaurav\Visual Studio\Speech Recognition> conda activate base
PS C:\Users\Gaurav\Visual Studio\Speech Recognition> & C:/ProgramData/Anaconda3/python.exe "c:/Users/Gaurav/Visual Studio/Speech Recognition/test.py"
[search hello: search youtube]
speak now
hold on while we open: amazon amazon youtube
search your video
India India India
PS C:\Users\Gaurav\Visual Studio\Speech Recognition>
```

Python Code Output - 2



Sign in to like videos, comment, and subscribe.

Pawandeep और Arunita का Duet Performance आया Kumar Sanu जी को पसंद | Indian Idol | Contestant...
177K views • 1 day ago
SET India
Click here to Subscribe to SET India: https://www.youtube.com/channel/UCpEhnqL0y41EpW2TvWAHD7Q?...
New
8:50

Mix - INDIA
YouTube
me canse de ser la otra - india • 5:42
India - Me Voy a Acostumbrar (Versión Salsa) [Official Video] ft. Juan Ga... • 4:07

OFFICIAL: 'India Waale' FULL VIDEO Song |Happy New Year | Shah Rukh Khan, Deepika Padukone
113M views • 6 years ago
T-Series
SONG - INDIA WAALE MOVIE - HAPPY NEW YEAR STARCAST - DEEPIKA PADUKONE, SHAH RUKH KHAN, ABHISHEK ...
Official Music Video
3:58

India Alert | New Episode 550 | Hanikarak Love - हानिकारक लव | #DangalTVChannel
546K views • 19 hours ago
Dangal TV Channel

INDIA
Music

Mi Mayor Venganza 5:14
2.4M views

Víctimas las Dos 4:18
288K views

Que Ganas De No Verte Nunca Más 4:51
16K views

Dicen Que Soy 4:41
5.9K views

Me Cansé De Ser La Otra 5:12
15K views

Me Voy a Acostumbrar - Versión Salsa 4:07
22M views • 5 years ago

Costumbres 5:49
2.6K views

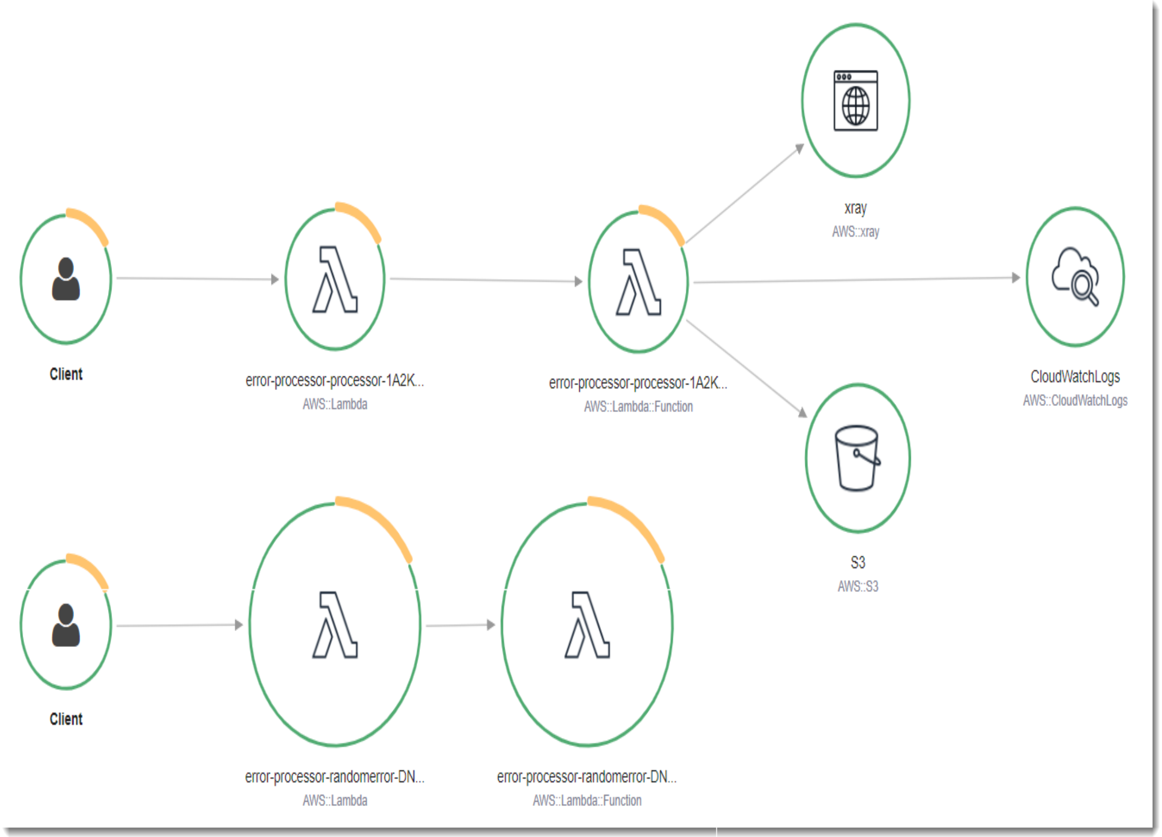
Tu Amor Es Mi Piel 4:13
632K views

Ese Hombre

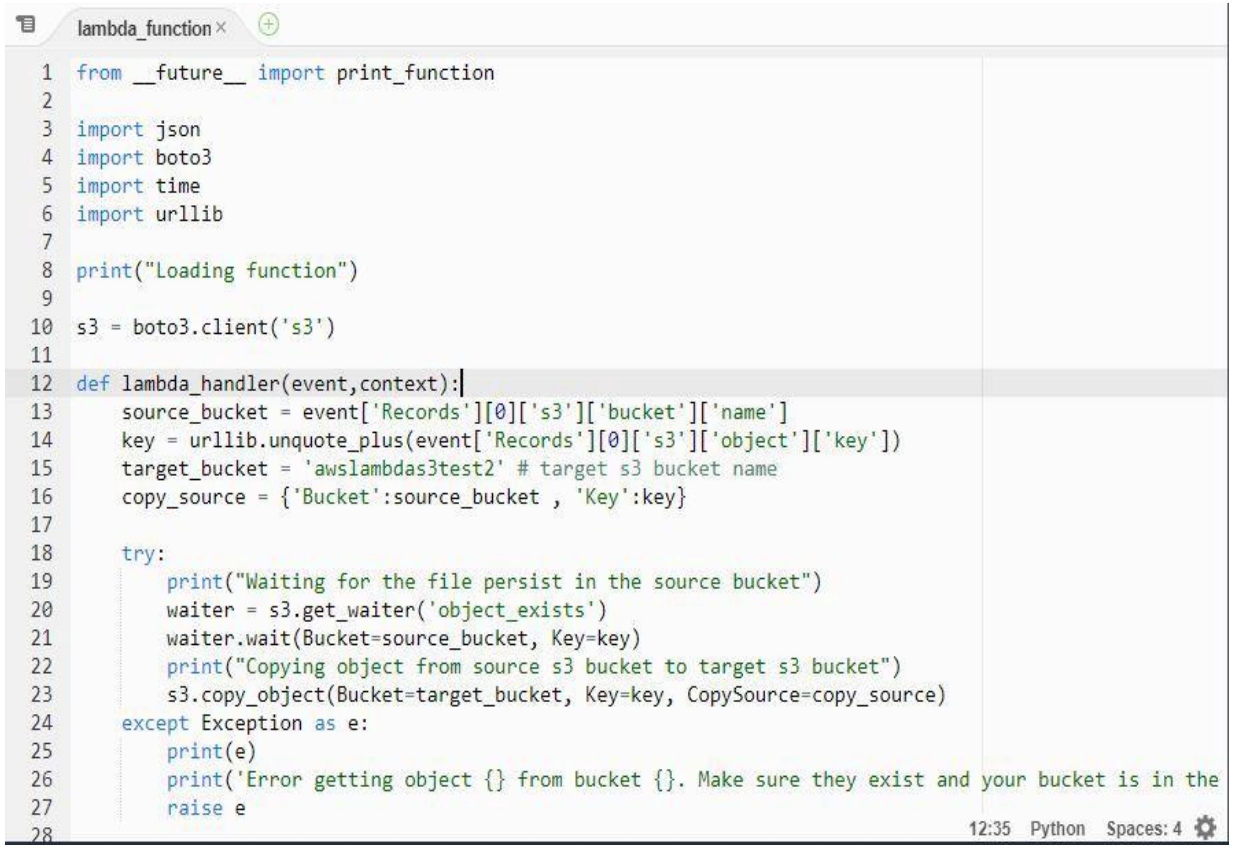
AWS Deployment package in Python

Using Amazon AWS Lamda, CRON Jobs, Cloudwatch

Amazon AWS Lamda - 1



xray
AWS::xray
Client
error-processor-processor-1A2K...
AWS::Lambda
error-processor-processor-1A2K...
AWS::Lambda::Function
CloudWatchLogs
AWS::CloudWatchLogs
S3
AWS::S3
Client
error-processor-randomerror-DN...
AWS::Lambda
error-processor-randomerror-DN...
AWS::Lambda::Function

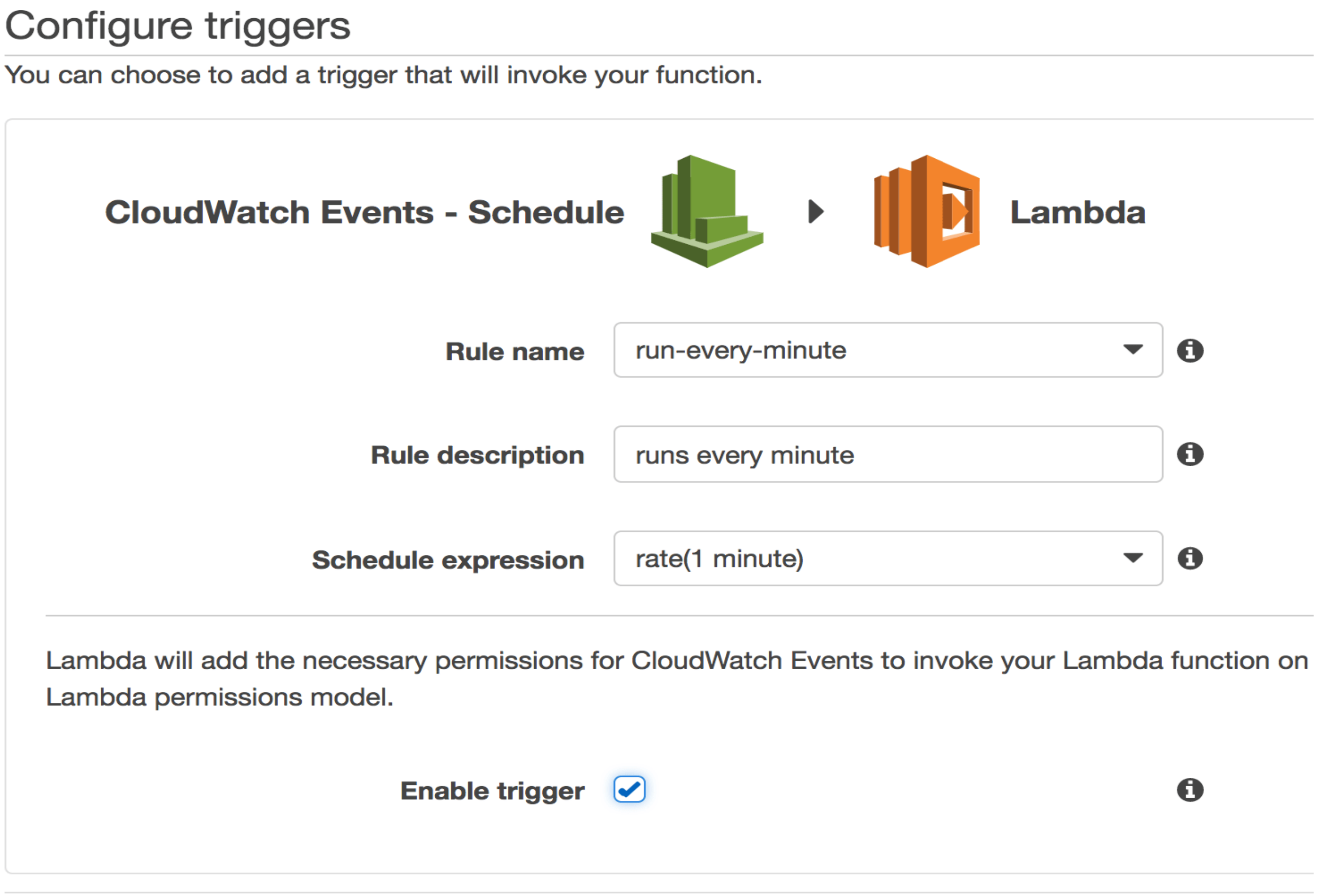


```
from __future__ import print_function

import json
import boto3
import time
import urllib

print("Loading function")

s3 = boto3.client('s3')

def lambda_handler(event,context):
    source_bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.unquote_plus(event['Records'][0]['s3']['object']['key'])
    target_bucket = 'awslambdas3test2' # target s3 bucket name
    copy_source = {'Bucket':source_bucket , 'Key':key}

    try:
        print("Waiting for the file persist in the source bucket")
        waiter = s3.get_waiter('object_exists')
        waiter.wait(Bucket=source_bucket, Key=key)
        print("Copying object from source s3 bucket to target s3 bucket")
        s3.copy_object(Bucket=target_bucket, Key=key, CopySource=copy_source)
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the
        raise e
```

Amazon AWS Lamda - 2

- The function runtime passes a context object to the handler, in addition to the invocation event. The context object contains additional information about the invocation, the function, and the execution environment. More information is available from environment variables.
- Your Lambda function comes with a CloudWatch Logs log group. The function runtime sends details about each invocation to CloudWatch Logs. It relays any logs that your function outputs during invocation. If your function returns an error, Lambda formats the error and returns it to the invoker.
- The `lambda_function` file exports a function named `lambda_handler` that takes an event object and a context object. This is the handler function that Lambda calls when the function is invoked. The Python function runtime gets invocation events from Lambda and passes them to the handler. In the function configuration, the handler value is `lambda_function.lambda_handler`.
- Each time you save your function code, the Lambda console creates a deployment package, which is a .zip file archive that contains your function code. As your function development progresses, you will want to store your function code in source control, add libraries, and automate deployments. Start by creating a deployment package and updating your code at the command line.

Schedule expressions using rate or cron - 1



Configure triggers

You can choose to add a trigger that will invoke your function.

Rule name: run-every-minute

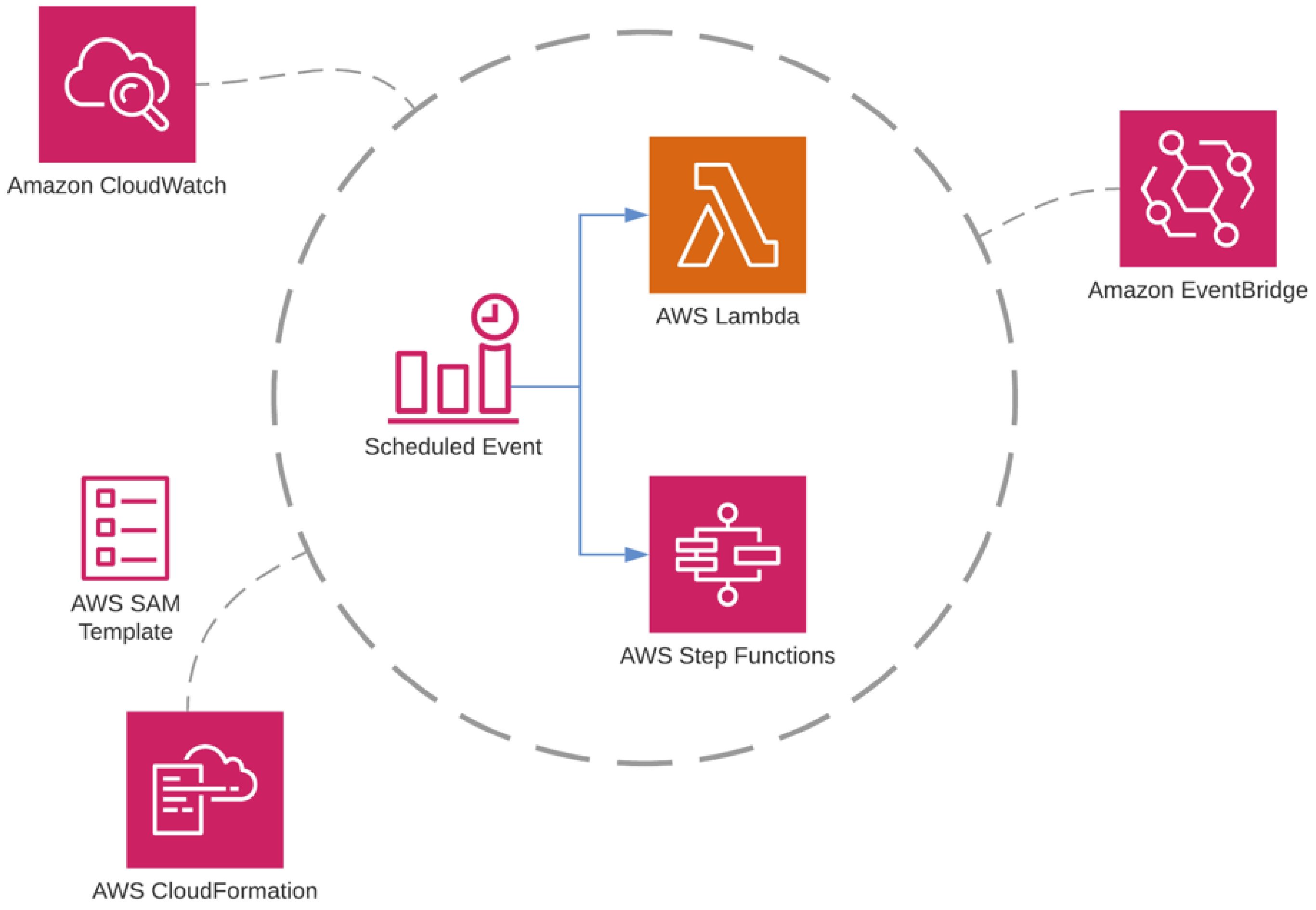
Rule description: runs every minute

Schedule expression: rate(1 minute)

Lambda will add the necessary permissions for CloudWatch Events to invoke your Lambda function on Lambda permissions model.

Enable trigger ☑

Schedule expressions using rate or cron - 2



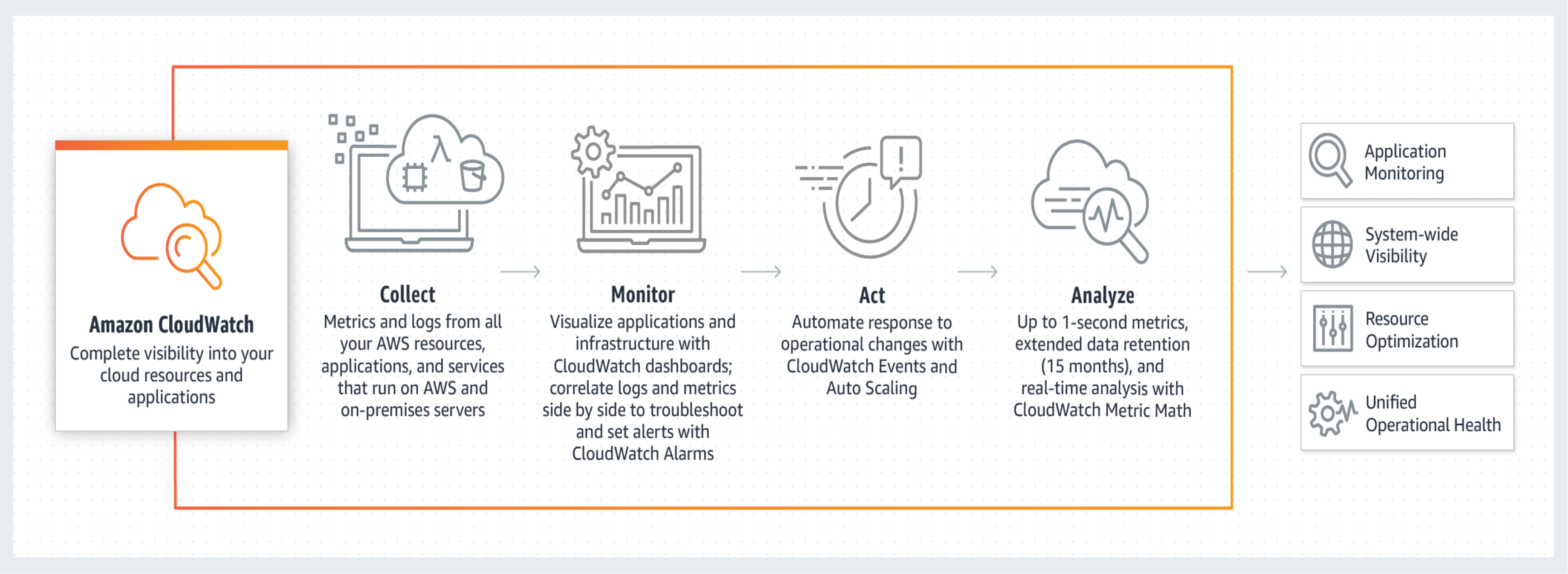
Schedule expressions using rate or cron - 3

Cron jobs are usually used to schedule commands at a specific time. You can use them for tasks like running backups, monitoring the status of the system, or running system maintenance tasks.

Cron jobs are a helpful utility for system administrators. And when you are administering a system in the cloud, cron jobs are still very useful – you still have to do a lot of administrative tasks on your systems.

One way of running cron jobs in the cloud is to use a function as a service (FaaS), like Lambda in the AWS ecosystem.

Amazon CloudWatch - 1



Amazon CloudWatch
Complete visibility into your cloud resources and applications

Collect
Metrics and logs from all your AWS resources, applications, and services that run on AWS and on-premises servers

Monitor
Visualize applications and infrastructure with CloudWatch dashboards; correlate logs and metrics side by side to troubleshoot and set alerts with CloudWatch Alarms

Act
Automate response to operational changes with CloudWatch Events and Auto Scaling

Analyze
Up to 1-second metrics, extended data retention (15 months), and real-time analysis with CloudWatch Metric Math

- Application Monitoring
- System-wide Visibility
- Resource Optimization
- Unified Operational Health

Amazon CloudWatch - 2

Amazon CloudWatch is a monitoring and management service that provides data and actionable insights for AWS, hybrid, and on-premises applications and infrastructure resources. With CloudWatch, you can collect and access all your performance and operational data in form of logs and metrics from a single platform.

This allows you to overcome the challenge of monitoring individual systems and applications in silos (server, network, database, etc.). CloudWatch enables you to monitor your complete stack (applications, infrastructure, and services) and leverage alarms, logs, and events data to take automated actions and reduce Mean Time to Resolution (MTTR). This frees up important resources and allows you to focus on building applications and business value.

Thank you

Gaurav Kumar

Intern Id: 1984738