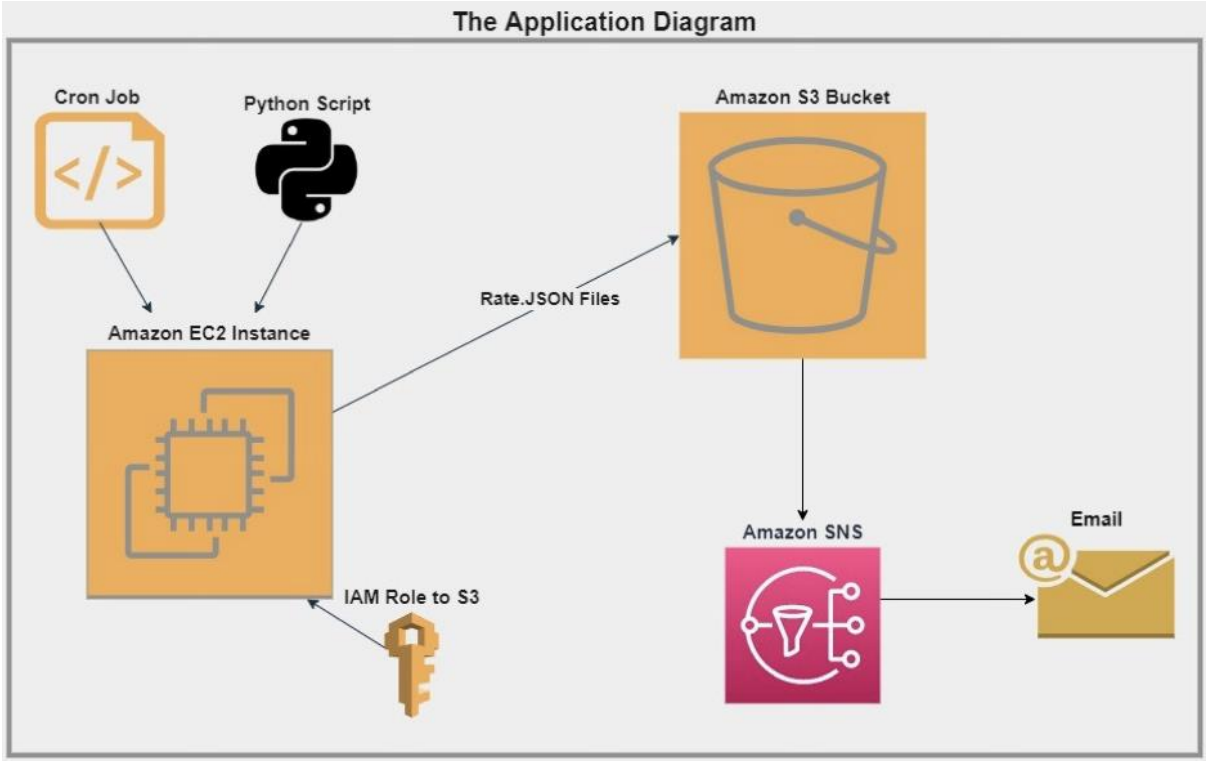
Creating an Exchange Rate API Hosted on AWS EC2 To Generate Hourly Exchange Rates

The client wants to have a file with the currency exchange rate generated every hour for their business need. The file should contain values of each currency every hour in a JSON format for their data analyst to use in creating a dashboard for their business. This will require a server to host the API job for this project and the API python code that will get the exchange rate. To get the updated exchange rate, a cron job was created to run the python code every hour. The output will be generated and saved as JSON files into an S3 bucket in US East 1 region. An SNS notification will be set up to send an email notification whenever the job is executed.



Steps to Complete the Project:

- The first thing we did was to spin an EC2 server on Amazon.
- Under the EC2 dashboard, click on launch instances on the top right corner. From the quick start, search and choose the EC2 AMI of your choice, for this project we are using the Amazon Linux 2 AMI. We are going to continue with the default configuration, under Security Group please ensure that SSH is allowed on port 22 as we will be connecting with the EC2 instance through the CLI command. Create a keypair or select your keypair if you already have one.
- Create an Amazon SNS topic and subscribe to the topic with the notification email where emails will be sent whenever the API is executed.
- An IAM role was created to grant EC2 permission to put object into the S3 bucket and also to allow the EC2 to send notifications through SNS.
- Connected to the EC2 instance through SSH using the keypair specified when creating the EC2.
- Created a folder in the EC2 instance home directory named “project2” and did a cd project2.

- Inside the project2 folder, created a folder named “data” where the file generated will written into and another folder named “scripts” where both the python and bash scripts will be saved.
- Note, the scripts created should be granted executable permission, run ‘sudo chmod 700’ on each script to grant ‘rwx’ for each of them.
- A python script was created to do an API call for the currency rate and save a copy of the json output into the designated S3 bucket called “cloudreality”.

The python code is below:

```
import os

import forex_python

import json

import boto3

from datetime import datetime

moment=datetime.today().strftime('%Y_%m_%d_%H_%M_%S')



from forex_python.converter import CurrencyRates

c = CurrencyRates()

rate=(c.get_rates('USD'))

#data = rate.json()

filename='home/ec2-user/project2/data/rate_'+moment+'.json'

print(f"Executed the exchange rate code and will save the created file with a timestamp as exchangerate_{moment}")

s3_client = boto3.client('s3')

with open(filename, 'w') as json_file:

        json.dump(rate, json_file)

json_file.close()

with open(filename, 'rb') as final:

        s3_client.upload_fileobj(final, 'cloudreality', f'exchangerate_{moment}.json')
print("Successfully executed, no error and file exported as JSON")

os.environ['AWS_DEFAULT_REGION'] = 'us-east-1'

sns_client = boto3.client('sns')

sns_client.publish(
```

TargetArn='arn:aws:sns:us-east-1:022357963194:ExchangeRate',

Message=f'Hello Team, \n The exchange rate for the hour file named exchangerate_{moment} has been successfully logged into CloudReality S3 bucket. The next exchange rate notification is in an hour, if none is received kindly take that as an indication of an error and act promptly. \n Thank you. \n Cloud Developer Carew Ayodeji')

- The above will output a message every time the python script is executed and print it into the terminal.
- To automate the execution of the python API job, a cron is created which will only run the python script every hour and send the output into a cronjob file created for logs.

0 * * * * /usr/bin/python3 /home/ec2-user/project2/scripts/rate.py >> /home/ec2-user/project2/data/cron.log 2>&1

- The cron job is initiated by typing ‘crontab -e’ into the terminal, clicking ‘i’ to insert into the crontab, and can paste the above bash to have it trigger the python API every 0 minutes of every hour, every day, every month and every year.
- Every time the API is executed, a log with the output is written into the cronjob file inside the /data directory.
- The code was executed by the cronjob on the hour and an email was received as shown below with the message from the rate.py to the group of recipients that subscribed to the SNS topic.
- Below is a picture of the email notifications received at every hour with the Exchange rate file from the automated API job. The message can always be customized as needed.