

## Waq Ahmed

6 Followers

About

Follow

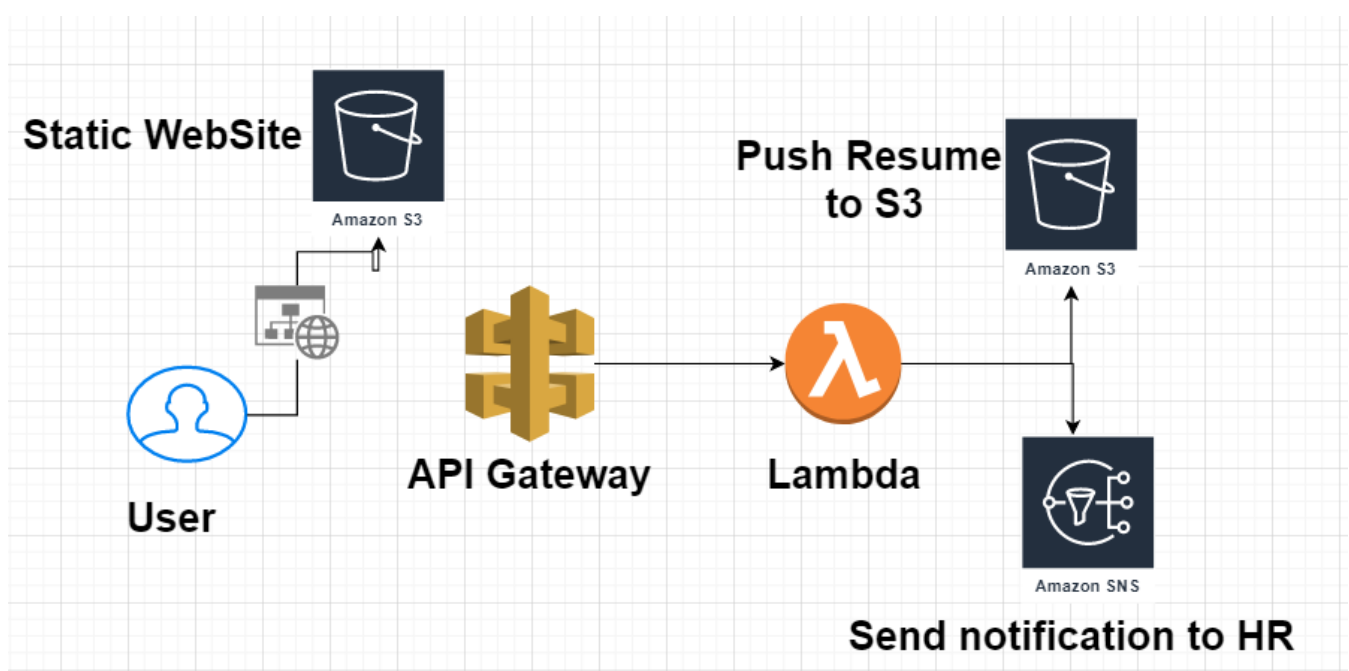


# Manage Job Application using AWS Free Service



Waq Ahmed Dec 3, 2021 · 3 min read

One of my friend (HR Executive) reached out to me with a problem. The issue was that they were getting the hundreds of Resume from candidates via email and it's hard for them to manage all those resume in his email. They want some solution for this problem. However, as a startup company they don't have any infrastructure or budget for this. After hearing this use case, I came up with an idea to use AWS Services in Free Tier to address this issue



High Level Architecture



time send a notification to HR about the job application received. Both API Gateway and Lambda provide 1 million call within free tier (which will be more than enough in this case) and S3 and SNS also fall in free tier.

## Static Website:

Create a new S3 bucket and enable public access and Static WebSite. Upload the index.html page to make your website available for all. Note: I'm not going to cover how to host a static website here. If you are new to AWS then refer [this document](#)

```
<html>
<head>
<title>Please upload your Resume< </title>
<center><h2>Upload your Resume to apply for the job </h2>

<br></br>
<form method="post" enctype="multipart/form-data",
action="https://ua6zsjvy23.execute-api.us-east-1.amazonaws.com/v1/upload">
  Select File:
  <input type="file" name="file"><br><br><br>
  <input type="submit" name="submit" value="Submit">
</head>
</center>
</form>
</html>
```

## AWS Lambda function:

Create a new lambda function with Python 3.8 Runtime. Attach the execution role with the function to allow lambda to execute and send notification using SNS. Add following Python code to lambda function

```
import json
import base64
import boto3
import email

def lambda_handler(event, context):
    s3 = boto3.client("s3")

    client = boto3.client('sns')
```



```
try:
    content_type = event["headers"]["Content-Type"]
except:
    content_type = event["headers"]["content-type"]
# concat Content-Type: with content_type from event
ct = "Content-Type: " + content_type + "\n"

# parsing message from bytes
msg = email.message_from_bytes(ct.encode() + post_data)

# checking if the message is multipart
print("Multipart check : ", msg.is_multipart())

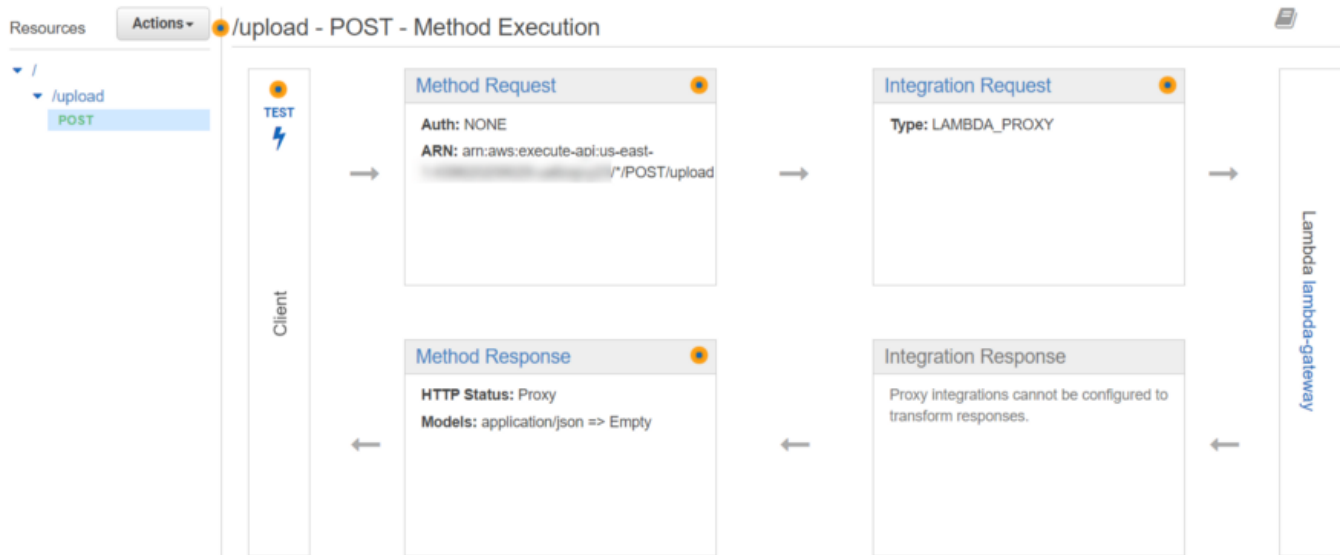
# if message is multipart
if msg.is_multipart():
    multipart_content = {}
    # retrieving form-data
    for part in msg.get_payload():
        # checking if filename exist as a part of content-
disposition header
        if part.get_filename():
            # fetching the filename
            file_name = part.get_filename()
            multipart_content[
                part.get_param("name", header="content-disposition")
            ] = part.get_payload(decode=True)

# filename from form-data
#file_name = json.loads(multipart_content["Metadata"])
["filename"]
# u uploading file to S3
s3_upload = s3.put_object(
    Bucket="aws-bucket-to-store-Resume", Key=file_name,
Body=multipart_content["file"]
)

#Send email to HR
topic_arn = 'arn:aws:sns:us-east-
1:4396202xxxx:Resume_Manager'
message = 'A job application has been received. Please check
the Bucket for all Resume'
client.publish(TopicArn=topic_arn,Message=message)

# on upload success
return {"statusCode": 200, "body": json.dumps("Thank you for
your intrest in this job. We will get back to you soon")}
else:
    # on upload failure
    return {"statusCode": 500, "body": json.dumps("Upload
failed!")}
```

“POST” to allow user send Resume using REST-API. Integrate the API with Lambda function and enable lambda proxy



API Gateway Configuration

Make sure to go to API Gateway → Setting → Binary Media Type and set it to ‘multipart/form-data’

## AWS SNS:

Create a new SNS topic and subscribe to the email address you wish you receive the emails. Make sure to add the ARN to lambda code

**Details**

Name

Resume\_Manager

ARN

arn:aws:sns:us-east-1: Resume\_Manager

Type

Standard

Display name

-

Topic owner

Subscriptions

Access policy

Delivery retry policy (HTTP/S)

Delivery status logging

Encryption

Tags

Subscriptions (1)

Edit

Delete

Request confirmation

Confirm subscription

Create subscription

Q Search

< 1 > ⚙

ID	Endpoint	Status	Protocol
07f153b7-951a-45c8-87cb-d7078	forwardtowaqas@gmail.com	Confirmed	EMAIL

---

[Open in app](#)



---

Please hit the clap icon if you like this article or leave your feedback.

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

