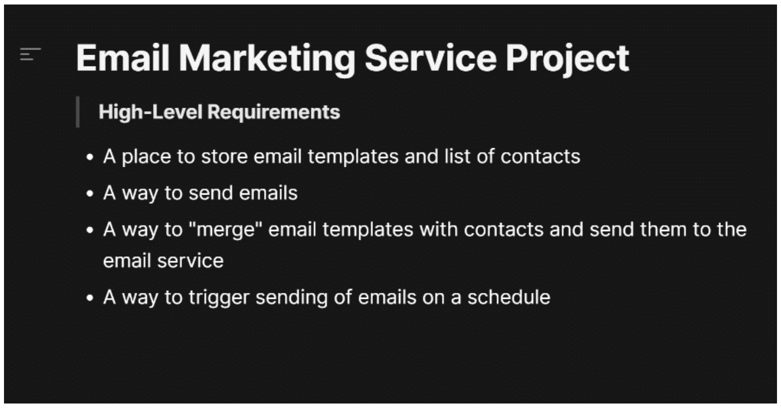
Introduction:

This project is supposed to set up a serverless architecture for an email marketing service. The architecture is comprised of AWS S3, SES, Lambda, and EventBridge. S3 is used to store the email templates and contacts. SES is used to send the emails to the designated addresses. Lambda is use to merge the email templates with the contacts and send them to the email service, and EventBridge is used to create a schedule where the event of sending these emails are triggered.

Requirements:

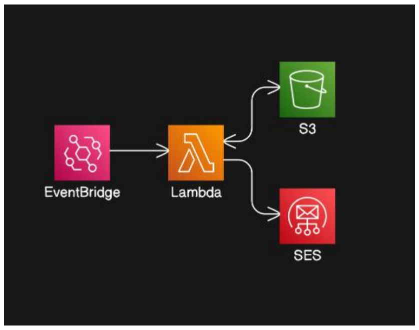


## Email Marketing Service Project

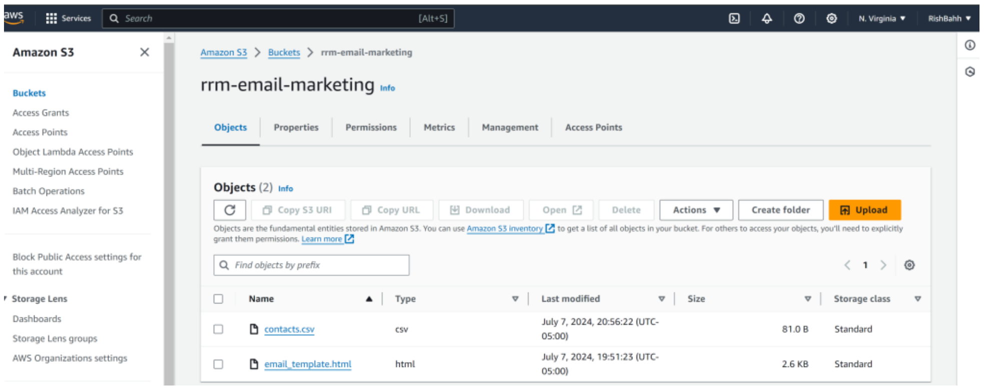
**High-Level Requirements**

- A place to store email templates and list of contacts
- A way to send emails
- A way to "merge" email templates with contacts and send them to the email service
- A way to trigger sending of emails on a schedule

Architecture of the Email Marketing Service:



S3 Configuration:



rrm-email-marketing Info

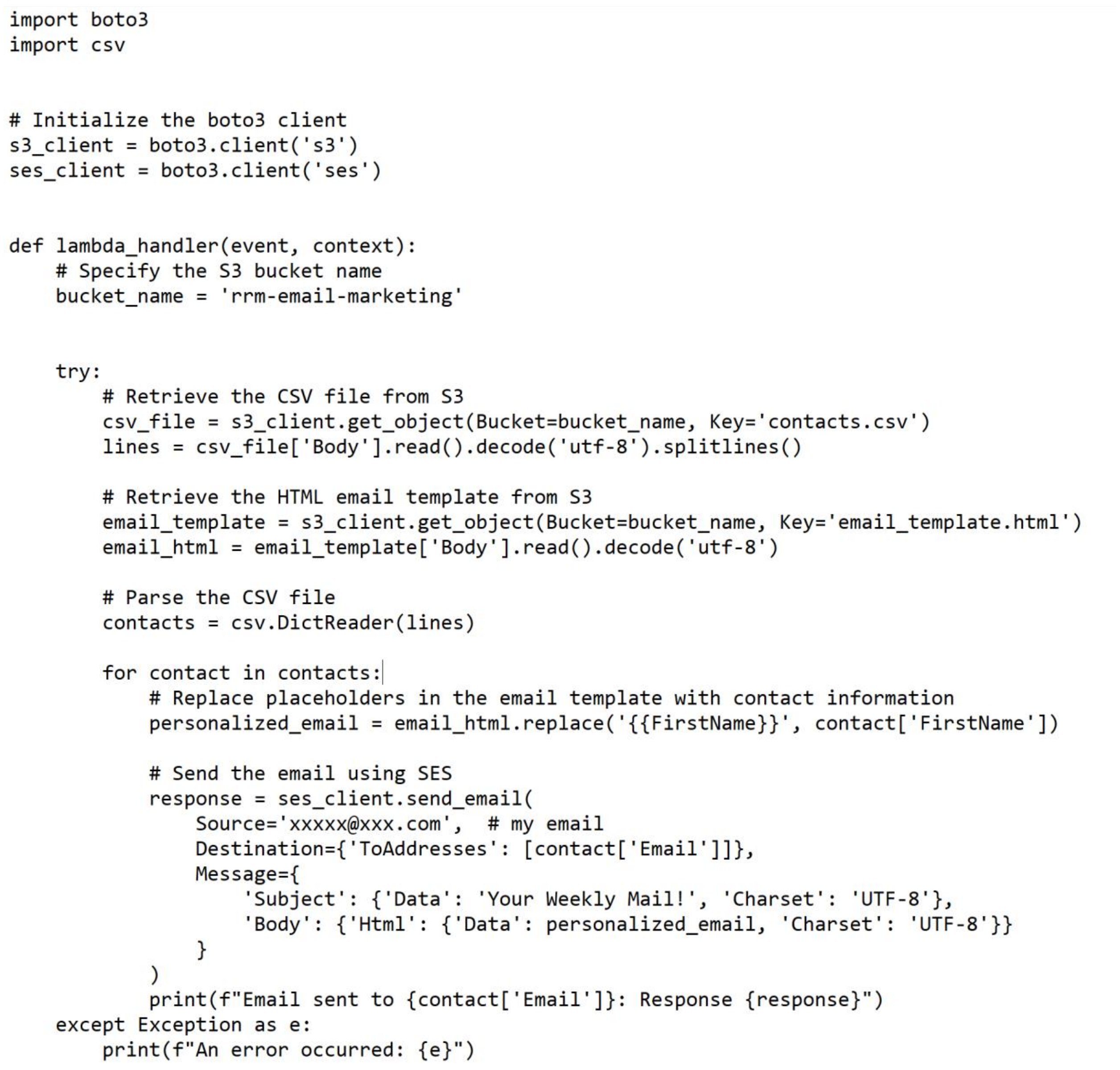
Objects (2) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

| Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|
| contacts.csv | csv | July 7, 2024, 20:56:22 (UTC-05:00) | 81.0 B | Standard |
| email_template.html | html | July 7, 2024, 19:51:23 (UTC-05:00) | 2.6 KB | Standard |

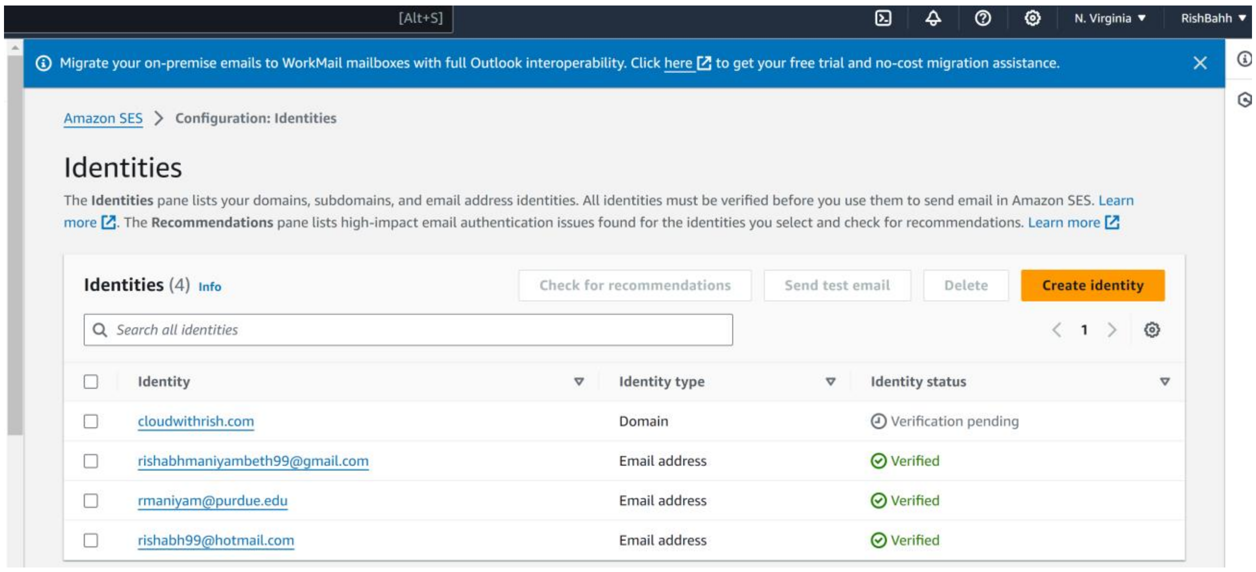
contacts.csv and email_template.html file adapted from
https://www.youtube.com/watch?v=hK0ztmWCBA8

Lambda Python Code:



```python
import boto3
import csv


# Initialize the boto3 client
s3_client = boto3.client('s3')
ses_client = boto3.client('ses')


def lambda_handler(event, context):
    # Specify the S3 bucket name
    bucket_name = 'rrm-email-marketing'


    try:
        # Retrieve the CSV file from S3
        csv_file = s3_client.get_object(Bucket=bucket_name, Key='contacts.csv')
        lines = csv_file['Body'].read().decode('utf-8').splitlines()

        # Retrieve the HTML email template from S3
        email_template = s3_client.get_object(Bucket=bucket_name, Key='email_template.html')
        email_html = email_template['Body'].read().decode('utf-8')

        # Parse the CSV file
        contacts = csv.DictReader(lines)

        for contact in contacts:
            # Replace placeholders in the email template with contact information
            personalized_email = email_html.replace('{{FirstName}}', contact['FirstName'])

            # Send the email using SES
            response = ses_client.send_email(
                Source='xxxxx@xxx.com',  # my email
                Destination={'ToAddresses': [contact['Email']]},
                Message={
                    'Subject': {'Data': 'Your Weekly Mail!', 'Charset': 'UTF-8'},
                    'Body': {'Html': {'Data': personalized_email, 'Charset': 'UTF-8'}}
                }
            )
            print(f"Email sent to {contact['Email']}: Response {response}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

## SES Configuration:



Migrate your on-premise emails to WorkMail mailboxes with full Outlook interoperability. Click here to get your free trial and no-cost migration assistance.

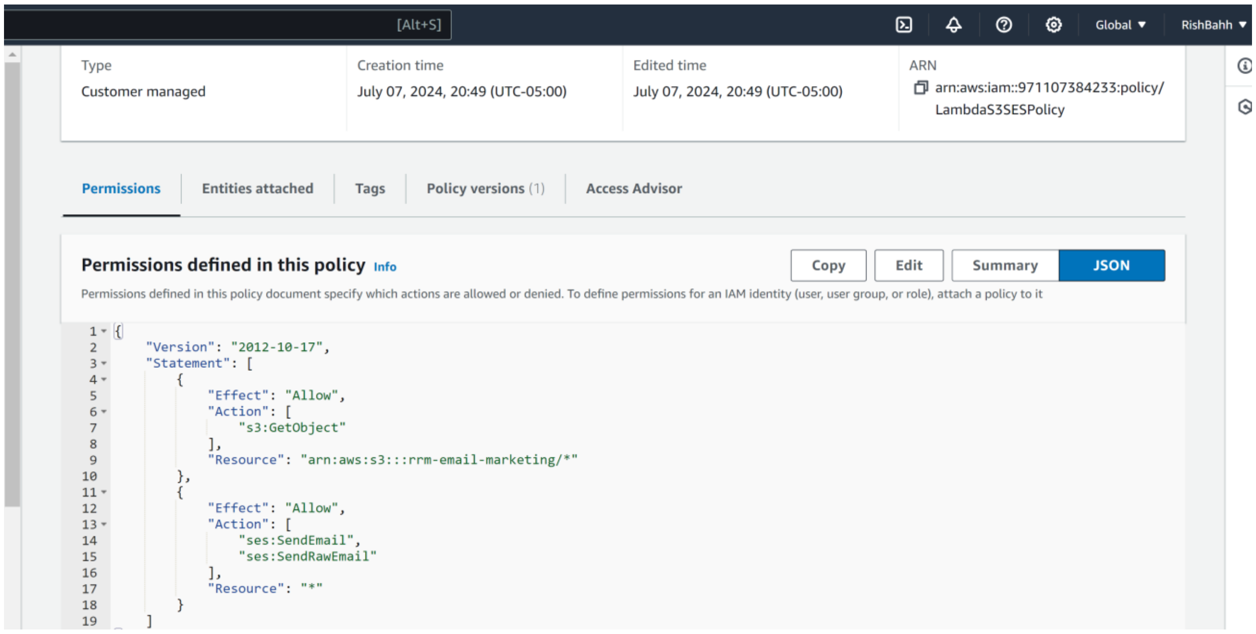
Amazon SES > Configuration: Identities

Identities

The **Identities** pane lists your domains, subdomains, and email address identities. All identities must be verified before you use them to send email in Amazon SES. Learn more. The **Recommendations** pane lists high-impact email authentication issues found for the identities you select and check for recommendations. Learn more

| Identity | Identity type | Identity status |
|---|---|---|
| cloudwithrish.com | Domain | Verification pending |
| rishabhmaniyambeth99@gmail.com | Email address | Verified |
| rmaniyam@purdue.edu | Email address | Verified |
| rishabh99@hotmail.com | Email address | Verified |

## IAM Policy for Lambda:



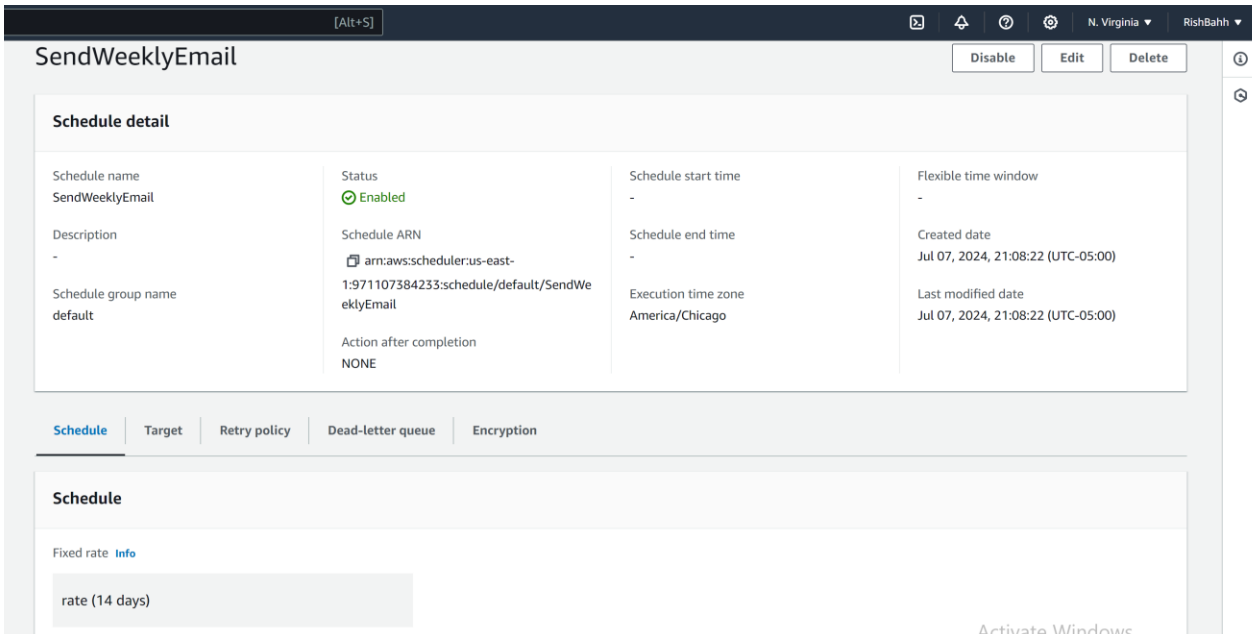
| Type | Creation time | Edited time | ARN |
|---|---|---|---|
| Customer managed | July 07, 2024, 20:49 (UTC-05:00) | July 07, 2024, 20:49 (UTC-05:00) | arn:aws:iam::971107384233:policy/LambdaS3SESPolicy |

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::rrm-email-marketing/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:SendEmail",
                "ses:SendRawEmail"
            ],
            "Resource": "*"
        }
    ]
```

EventBridge Configuration:



SendWeeklyEmail

Disable Edit Delete

**Schedule detail**

Schedule name
SendWeeklyEmail

Description
-

Schedule group name
default

Status
Enabled

Schedule ARN
arn:aws:scheduler:us-east-1:971107384233:schedule/default/SendWeeklyEmail

Action after completion
NONE

Schedule start time
-

Schedule end time
-

Execution time zone
America/Chicago

Flexible time window
-

Created date
Jul 07, 2024, 21:08:22 (UTC-05:00)

Last modified date
Jul 07, 2024, 21:08:22 (UTC-05:00)

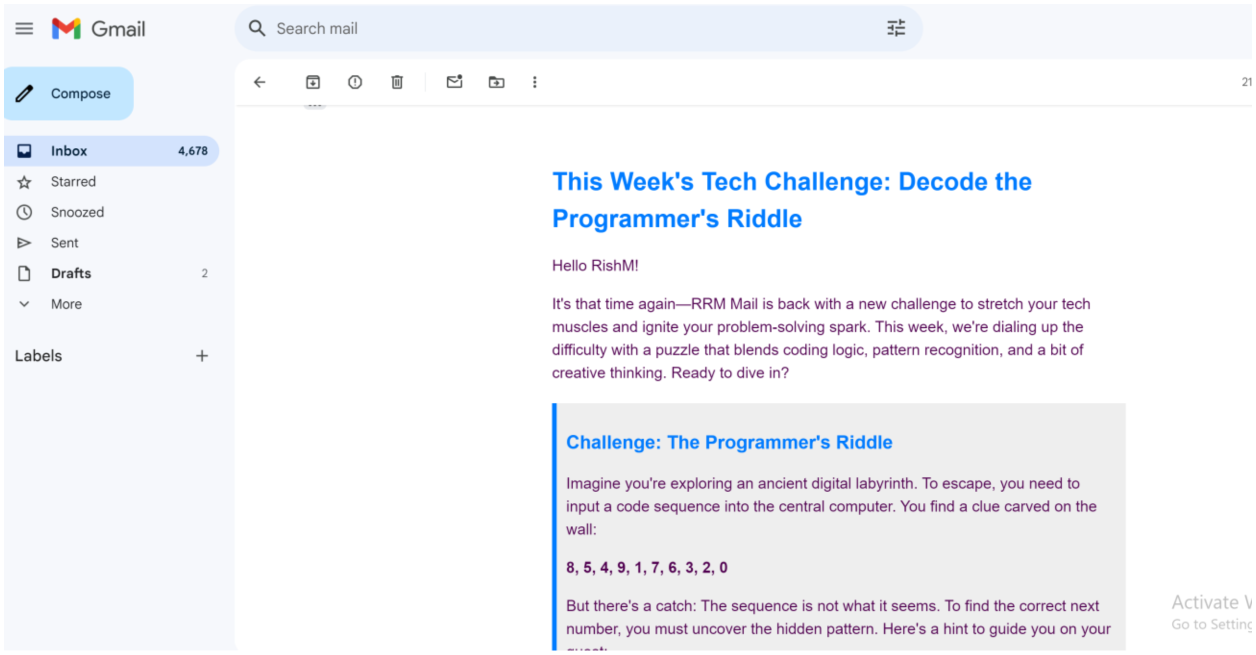
Schedule | Target | Retry policy | Dead-letter queue | Encryption

**Schedule**

Fixed rate Info

rate (14 days)

Result:



**This Week's Tech Challenge: Decode the Programmer's Riddle**

Hello RishM!

It's that time again—RRM Mail is back with a new challenge to stretch your tech muscles and ignite your problem-solving spark. This week, we're dialing up the difficulty with a puzzle that blends coding logic, pattern recognition, and a bit of creative thinking. Ready to dive in?

**Challenge: The Programmer's Riddle**

Imagine you're exploring an ancient digital labyrinth. To escape, you need to input a code sequence into the central computer. You find a clue carved on the wall:

**8, 5, 4, 9, 1, 7, 6, 3, 2, 0**

But there's a catch: The sequence is not what it seems. To find the correct next number, you must uncover the hidden pattern. Here's a hint to guide you on your