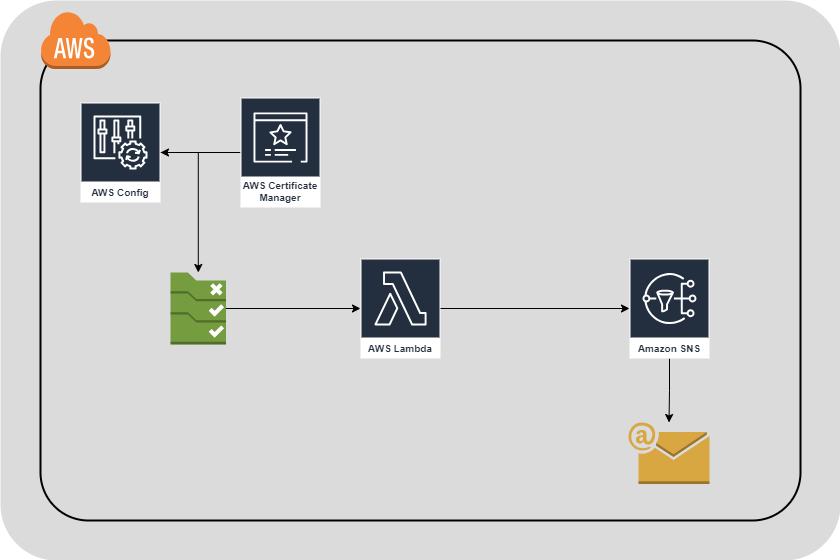
**ACM Custom Email Notification**

**Overview**

The **ACM Custom Email Notification** process is designed to monitor AWS **ACM Certificates** for expiry using **AWS Config** and **AWS Lambda**. A custom **AWS Config Rule** is created to track ACM certificate changes, and any detected configuration change triggers the Lambda function.

The Lambda function **analyzes the certificate details**, extracts the **expiry date**, and determines whether the certificate is expiring within **90, 60, or 30 days**. If the condition is met, the Lambda function **publishes a message** to an **SNS topic**, which **sends an email notification** to the **APS Operations Team**.

**Architecture Diagram**  


AWS Config Rule → AWS Lambda → SNS Topic → Email Notification

**Components and Workflow**

**1. AWS Config Rule**

* An **AWS Config Rule** is created to track ACM certificate configuration changes.
* When an ACM certificate changes (e.g., issuance, renewal, expiration update), **AWS Config** triggers the **Lambda function**.

**2. AWS Lambda Function**

* The **Lambda function** processes the **AWS Config event** and extracts details of the certificate, including:
  + **Account ID**
  + **AWS Region**
  + **Certificate ARN**
  + **Domain Name**
  + **Expiration Date**
* If the certificate’s expiry date falls within **90, 60, or 30 days**, the Lambda function:
  + 1. **Formats an email notification message**.
    2. **Publishes the message to an AWS SNS Topic**.

**3. AWS SNS Topic**

* The **SNS Topic (ACM-Certificate-Expiry)** is used to send the notification.
* The topic has an **email subscription for aps\_operations**, ensuring they receive the alert.

**Step-by-Step Implementation**

**Step 1: Create an AWS Config Rule**

1. **Go to AWS Config Console** → Click **Rules**.
2. Click **Add Rule** → Select **Custom Rule**.
3. Provide a **Rule Name** (e.g., acm-certificate-expiry-email).
4. Under **Trigger**, select:
   * **Event-based**
   * **Resource type: AWS::ACM::Certificate**
5. **Attach the Lambda function** (created in Step 2).
6. Click **Save Rule**.

**Step 2: Create a Lambda Function**

1. **Go to AWS Lambda Console** → Click **Create Function**.
2. Select **Author from Scratch**.
3. Enter function name: ACM\_Certificate\_Expiry.
4. Select **Runtime: Python 3.x**.
5. Under **Execution Role**, choose **Create a new role with basic permissions**.
6. Click **Create Function**.

**Lambda Code (Python)**

**Step 3: Configure SNS Topic**

1. **Go to AWS SNS Console** → Click **Create Topic**.
2. **Topic Name:** ACM-Certificate-Expiry.
3. **Choose Standard topic** → Click **Create**.

**Add Email Subscription**

1. **Click on the created SNS topic**.
2. Click **Create Subscription**.
3. Select **Protocol: Email**.
4. Enter **Email Address:** [**aps\_operations**](mailto:aps_operations@yourcompany.com).
5. Click **Create Subscription**.
6. **Confirm the subscription** from the email received.

**Testing the Process**

1. **Manually Trigger AWS Config Rule**:
   * Make an update to an ACM certificate.
   * AWS Config detects the change and triggers Lambda.
2. **Check Lambda Logs in CloudWatch**:
   * Go to **AWS Lambda Console** → Click on the function.
   * Navigate to **Monitor > Logs**.
   * Verify that the correct details are logged.
3. **Verify SNS Email Delivery**:
   * Ensure aps\_operations receives the alert.
   * Check **SNS Console > Topics > ACM-Certificate-Expiry > Delivery Status**.

**Permissions Required**

The Lambda function needs the following permissions:

{

"Effect": "Allow",

"Action": [

"sns:Publish",

"acm:ListCertificates",

"acm:DescribeCertificate",

"config:GetResourceConfigHistory"

],

"Resource": "\*"

}

**Conclusion**

This setup ensures that any changes to **ACM certificates** are detected via **AWS Config**, processed by **AWS Lambda**, and **emailed via SNS** when expiration is approaching.