

## **Assignment 2: Automated S3 Bucket Cleanup Using AWS Lambda and Boto3**

**Objective:** To gain experience with AWS Lambda and Boto3 by creating a Lambda function that will automatically clean up old files in an S3 bucket.

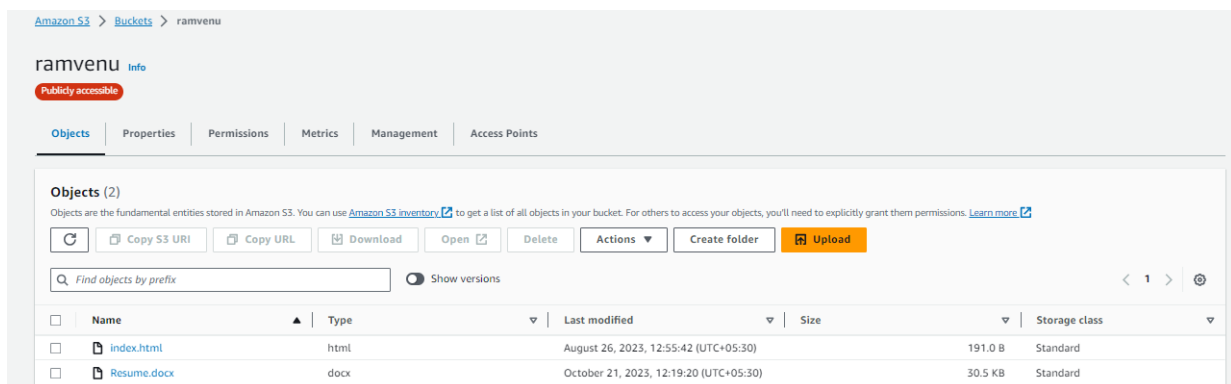
**Task:** Automate the deletion of files older than 30 days in a specific S3 bucket.

### **Instructions:**

#### **1. S3 Setup:**

- Navigate to the S3 dashboard and create a new bucket.
- Upload multiple files to this bucket, ensuring that some files are older than 30 days (you may need to adjust your system's date temporarily for this or use old files).

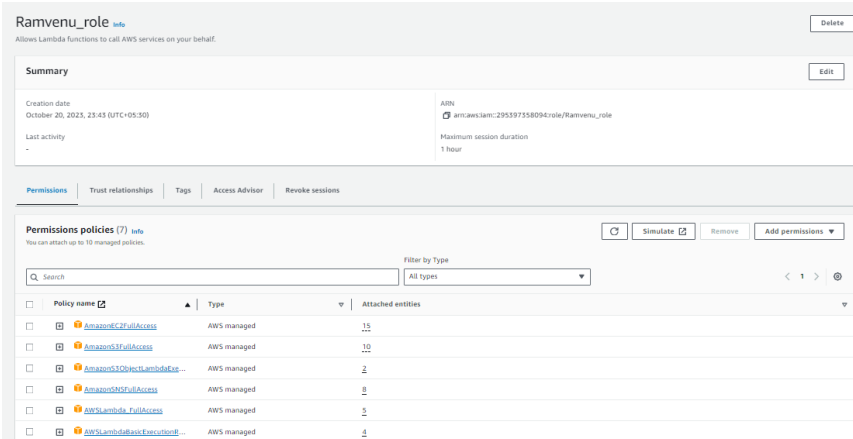
**Screenshot:** S3 bucket with files which are older than 30 days



#### **2. Lambda IAM Role:**

- In the IAM dashboard, create a new role for Lambda.
- Attach the 'AmazonS3FullAccess' policy to this role. (Note: For enhanced security in real-world scenarios, use more restrictive permissions.)

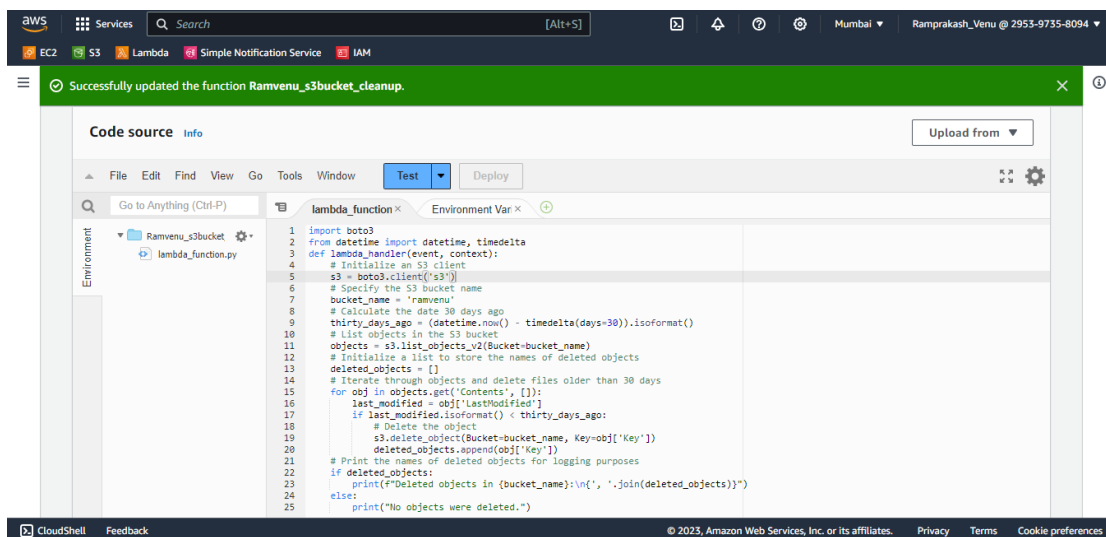
**Screenshot:** IAM Role created with all the access required to complete the tasks. ( one role for all assignment questions)

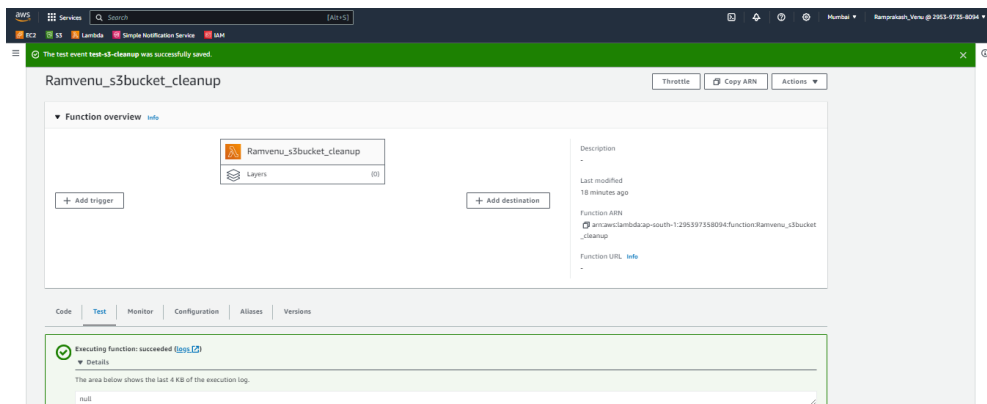


### 3. Lambda Function:

- Navigate to the Lambda dashboard and create a new function.
- Choose Python 3.x as the runtime.
- Assign the IAM role created in the previous step.
- Write the Boto3 Python script to:
  1. Initialize a boto3 S3 client.
  2. List objects in the specified bucket.
  3. Delete objects older than 30 days.
  4. Print the names of deleted objects for logging purposes.

**Screenshot:** deploying Lambda codes and running it.

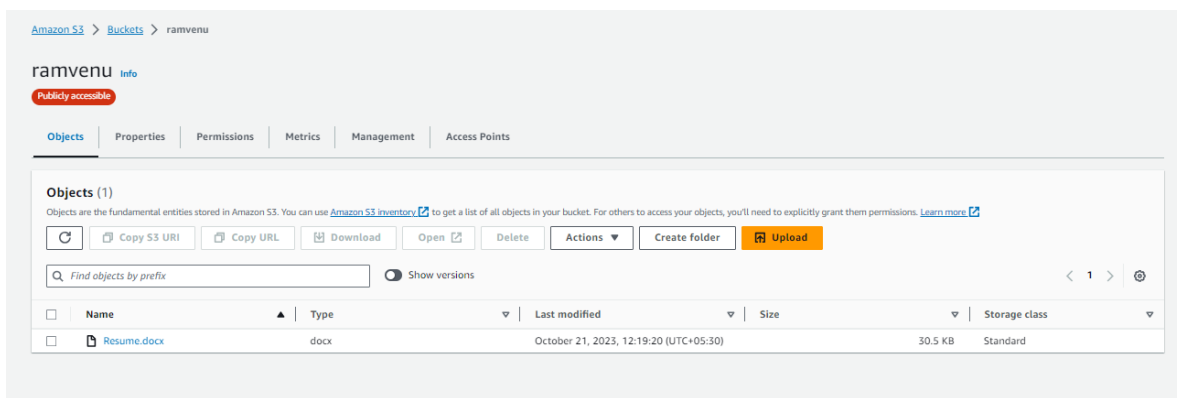




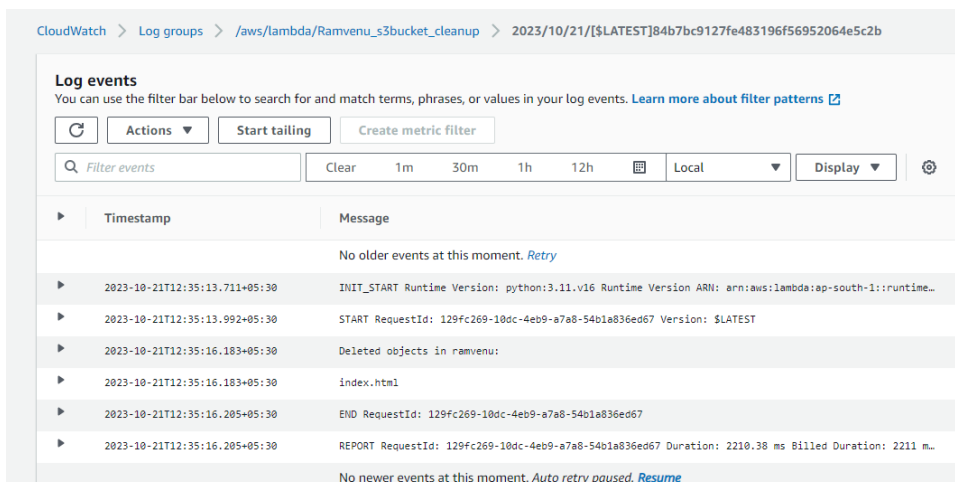
#### 4. Manual Invocation:

- After saving your function, manually trigger it.
- Go to the S3 dashboard and confirm that only files newer than 30 days remain.

**Screenshot:** S3 Bucket after clearing the file older than 30 days



**Screenshot:** Logfile indicating the task executed and the file name which was deleted



## Assignment 20: Load Balancer Health Checker

**Objective:** Design a Lambda function that checks the health of registered instances behind an Elastic Load Balancer (ELB) and notifies via SNS if any instances are unhealthy.

### Instructions:

1. Launch Two Ec2 instances , create Loadbalancer and add these Ec2 instance into aTarget Group

Screenshot: Ec2 instances Launched for the task

<input type="checkbox"/>	Ram_LBCheck_1	i-0afba499b4e034fee	Running	t2.micro	Initializing	No alarms
<input type="checkbox"/>	Ram_LBCheck_2	i-091593dba41cbef8a	Running	t2.micro	Initializing	No alarms

Screenshot: Load balancer Created

The screenshot shows the AWS Management Console 'Load balancers' page. It displays a table with one entry: 'Loadbalancer-RV' with a state of 'Active'. The page includes a search bar, a filter button, and a 'Create load balancer' button.

Screenshot: Target Group with two instances

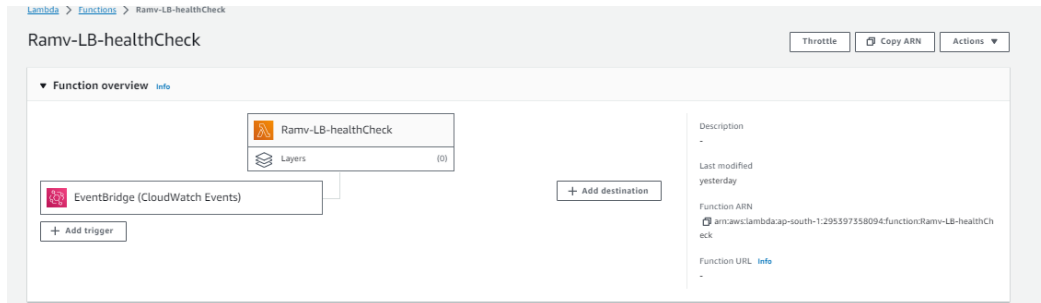
The screenshot shows the AWS Management Console 'Target-Group-Ramvenu' page. It displays details for the target group, including a table of registered targets. The table shows two instances, 'Ram\_LBCheck\_1' and 'Ram\_LBCheck\_2', both with a health status of 'unhealthy'.

2. Create a Lambda function.

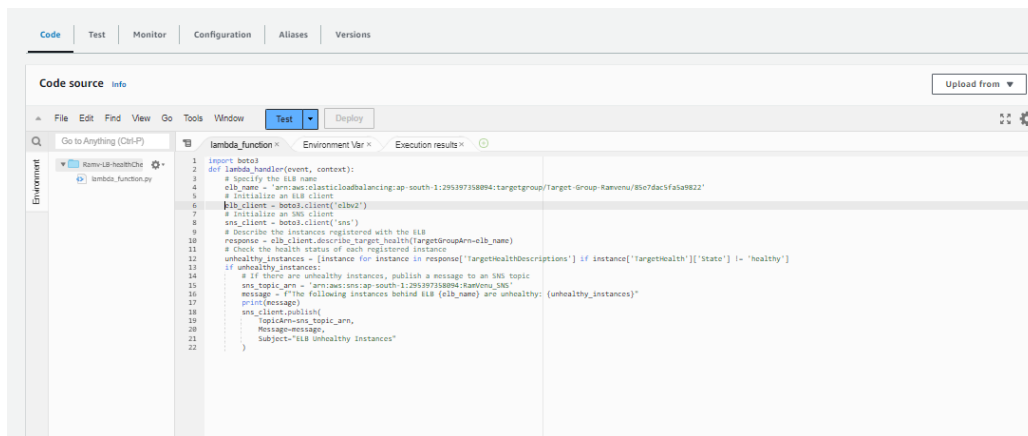
With Boto3, configure the function to:

Check the health of registered instances behind a given ELB. If any instances are found to be unhealthy, publish a detailed message to an SNS topic.

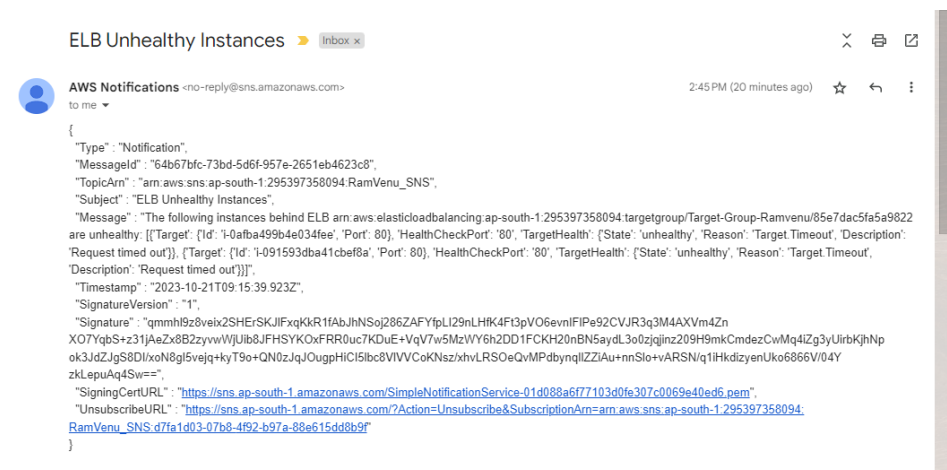
## Screenshot: Lambda function created



## Screenshot : Boto3 Code



## Screenshot : Detailed message with instance id notified on email



3. Set up a CloudWatch event to trigger this Lambda function every 10 minutes.

**Screenshot:** event trigger - Cron Job set to check health status every 10 mins and notify if instances is unheathy

Amazon EventBridge > Rules > LB-healthcheck-rule

LB-healthcheck-rule Edit Disable Delete CloudFormation Template

**Rule details**

Rule name LB-healthcheck-rule	Status <span>Enabled</span>	Event bus name default	Type Scheduled Standard
Description to check health of instance in LB every 10 mins	Rule ARN <a href="#">arn:aws:events:ap-south-1:295397358094:rule/LB-healthcheck-rule</a>	Event bus ARN <a href="#">arn:aws:events:ap-south-1:295397358094:event-bus/default</a>	

**Event schedule** Edit

Cron expression  
0/10 \* \* \* \* \*

Next 10 trigger date(s)  
Sat, 21 Oct 2023 09:40:00 UTC  
Sat, 21 Oct 2023 09:50:00 UTC  
Sat, 21 Oct 2023 10:00:00 UTC  
Sat, 21 Oct 2023 10:10:00 UTC  
Sat, 21 Oct 2023 10:20:00 UTC  
Sat, 21 Oct 2023 10:30:00 UTC  
Sat, 21 Oct 2023 10:40:00 UTC

UTC

**Screenshot :** SNS message on email for every 10 mins

AWS Notifications 4 ELB Unhealthy Instances - { "Type": "Notification", "MessageId": "762d5a82-3700-57b4-bd1..." 3:20 PM

ELB Unhealthy Instances Inbox x

**AWS Notifications** 2:45 PM (42 minutes ago) ☆  
{ "Type": "Notification", "MessageId": "64b67bfc-73bd-5d6f-957e-2651eb4623c8", "TopicArn": "arn:aws:sns:ap-south-1:295397358094:RamVenu\_SNS", "Subject": "..."

**AWS Notifications** 3:00 PM (28 minutes ago) ☆  
{ "Type": "Notification", "MessageId": "97e20c5b-9b61-5a52-a10c-51f607144b95", "Timestamp": "2023-10-21T09:30:29.627Z", "SignatureVersion": "1", "Signature": "..."

**AWS Notifications** 3:10 PM (18 minutes ago) ☆  
{ "Type": "Notification", "MessageId": "f69dbd80-1d32-532d-a3d0-e02d83ff5206", "Timestamp": "2023-10-21T09:40:29.507Z", "SignatureVersion": "1", "Signature": "..."

**AWS Notifications** <no-reply@sns.amazonaws.com> 3:20 PM (8 minutes ago) ☆ ↩ ⋮  
to me ▾  
{  
 "Type": "Notification",  
 "MessageId": "762d5a82-3700-57b4-bd1d-072f962584a9",  
 "TopicArn": "arn:aws:sns:ap-south-1:295397358094:RamVenu\_SNS",  
 "Subject": "ELB Unhealthy Instances",  
 "Message": "The following instances behind ELB arn:aws:elasticloadbalancing:ap-south-1:295397358094:targetgroup/Target-Group-Ramvenu/85e7dac5fa5a9822 are unhealthy: [{ 'Target': { 'Id': 'i-0afba499b4e034fee', 'Port': 80 }, 'HealthCheckPort': '80', 'TargetHealth': { 'State': 'unhealthy', 'Reason': 'Target.Timeout', 'Description': 'Request timed out' } }, { 'Target': { 'Id': 'i-091593dba41cbef8a', 'Port': 80 }, 'HealthCheckPort': '80', 'TargetHealth': { 'State': 'unhealthy', 'Reason': 'Target.Timeout', 'Description': 'Request timed out' } } ]".  
 }

## Assignment 6: Auto-Tagging EC2 Instances on Launch Using AWS Lambda and Boto3

**Objective:** Learn to automate the tagging of EC2 instances as soon as they are launched, ensuring better resource tracking and management.

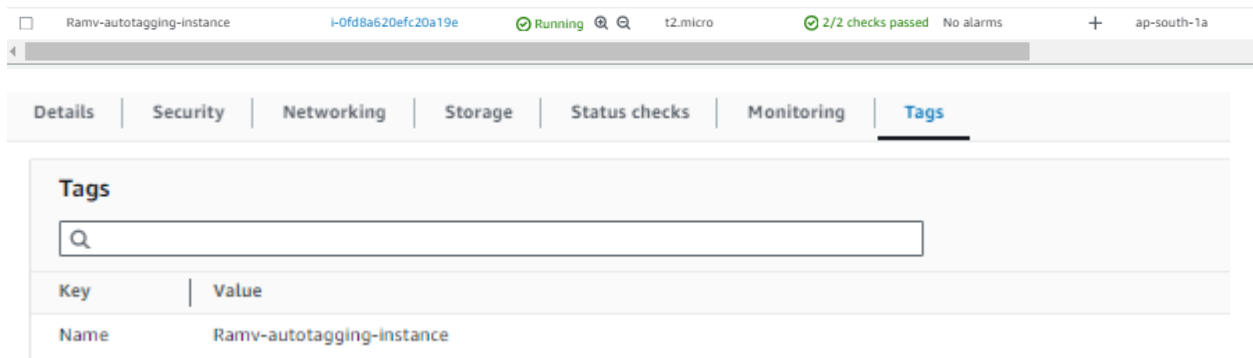
**Task:** Automatically tag any newly launched EC2 instance with the current date and a custom tag.

### Instructions:

#### 1. EC2 Setup:

- Ensure you have the capability to launch EC2 instances.

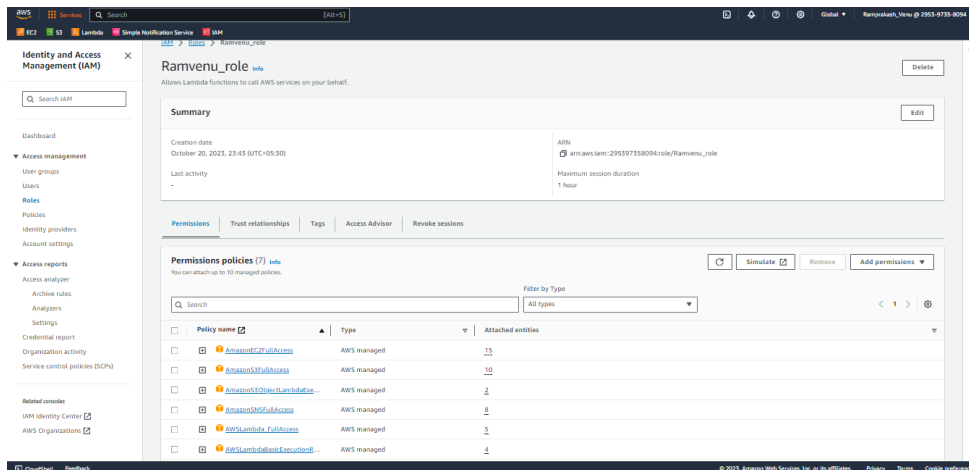
**Screenshot:** EC2 instance launched and tags before.



#### 2. Lambda IAM Role:

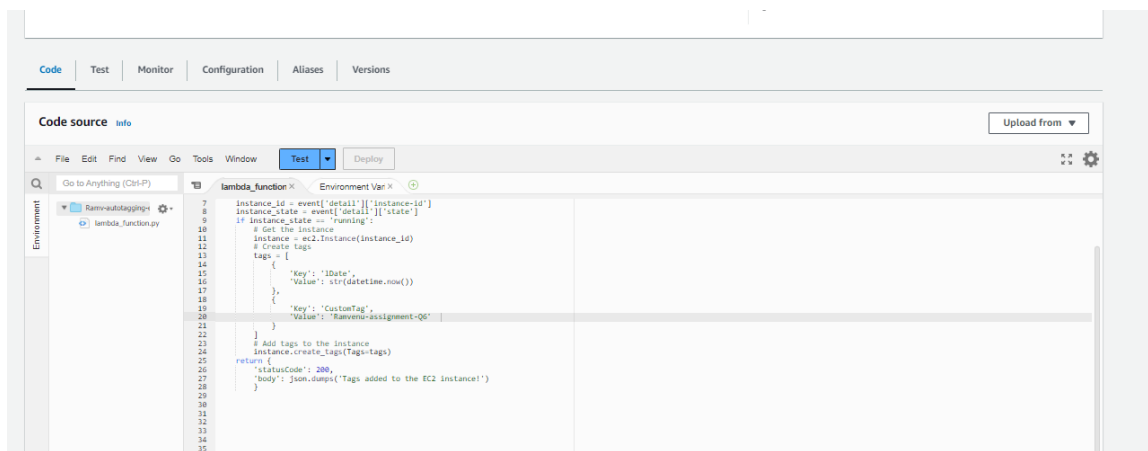
- In the IAM dashboard, create a new role for Lambda.
- Attach the 'AmazonEC2FullAccess' policy to this role.

**Screenshot:** IAM Role created with all the access required to complete the tasks. ( one role for all assignment questions)



### 3. Lambda Function:

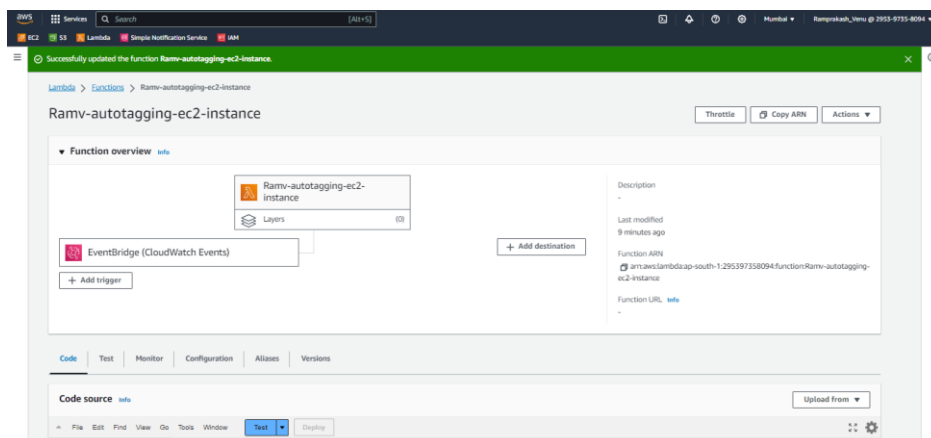
- Navigate to the Lambda dashboard and create a new function.
- Choose Python 3.x as the runtime.
- Assign the IAM role created in the previous step.
- Write the Boto3 Python script to:
  1. Initialize a boto3 EC2 client.
  2. Retrieve the instance ID from the event.
  3. Tag the new instance with the current date and another tag of your choice
  4. Print a confirmation message for logging purposes.



### 4. CloudWatch Events:

- Set up a CloudWatch Event Rule to trigger on the EC2 instance launch event.
- Attach the Lambda function as the target.

**Screenshot:** Triggers configured for lambda function





ec2-tag-trigger-ramv

Edit Disable Delete CloudFormation Template

**Rule details** info

Rule name ec2-tag-trigger-ramv	Status Enabled	Event bus name default	Type Standard
Description tags-update	Rule ARN arn:aws:events:ap-south-1:295397358094:rule/ec2-tag-trigger-ramv	Event bus ARN arn:aws:events:ap-south-1:295397358094:event-bus/default	

Event pattern Targets Monitoring Tags

**Event pattern** info Edit

```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["EC2 Instance State-change Notification"],
4   "detail": {
5     "state": ["running"],
6     "instance-id": ["i-0fd8a620efc20a19e"]
7   }
8 }
```

Copy

## 5. Testing:

- Launch a new EC2 instance.
- After a short delay, confirm that the instance is automatically tagged as specified.

## Screenshot: Tags successfully updated

Ramv-autotagging-instance i-0fd8a620efc20a19e Running t2.micro 2/2 checks passed No alarms + ap-south-1a

Details Security Networking Storage Status checks Monitoring **Tags**

**Tags**

Q

Key	Value
Name	Ramv-autotagging-instance
CustomTag	Ramvenu-assignment-Q6
lDate	2023-10-22 06:41:55.630519

## Assignment 15: Monitor EC2 Instance State Changes Using AWS Lambda, Boto3, and SNS

**Objective:** Automatically monitor changes in EC2 instance states and send notifications whenever an instance is started or stopped.

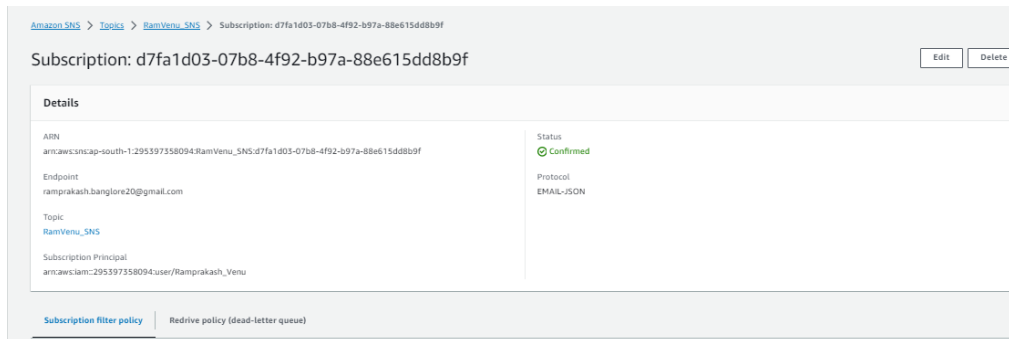
**Task:** Set up a Lambda function that listens to EC2 state change events and sends SNS notifications detailing the state changes.

### Instructions:

#### 1. SNS Setup:

- Navigate to the SNS dashboard and create a new topic.
- Subscribe to this topic with your email.

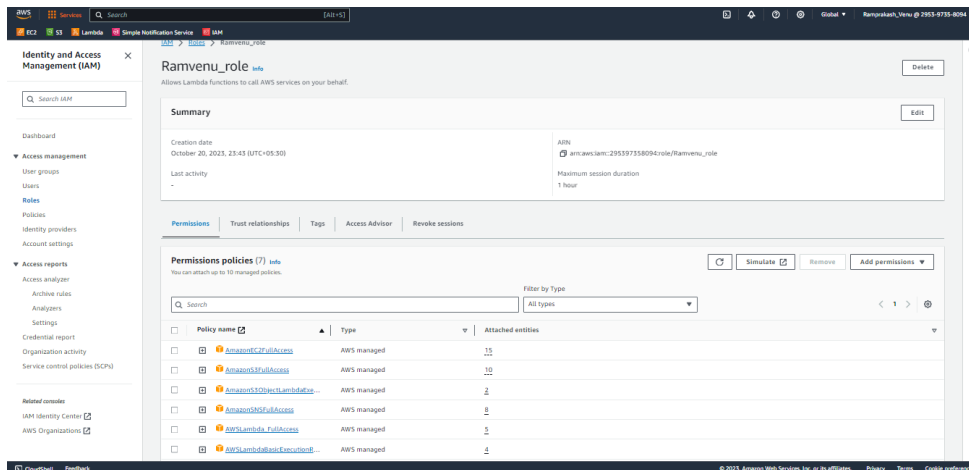
#### Screenshot : SNS dashboard



#### 2. Lambda IAM Role:

- Create a role with permissions to read EC2 instance states and send SNS notifications.

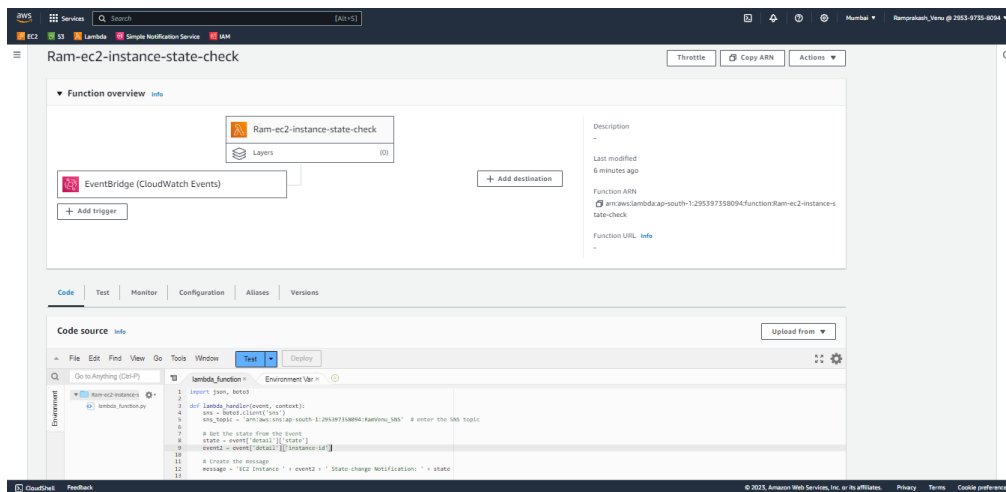
**Screenshot:** IAM Role created with all the access required to complete the tasks. ( one role for all assignment questions)



#### 3. Lambda Function:

- Create a function and assign the above IAM role.
- Use Boto3 to:
  1. Extract details from the event regarding the EC2 state change.
  2. Send an SNS notification with details about which EC2 instance changed state and the new state (e.g., started, stopped).

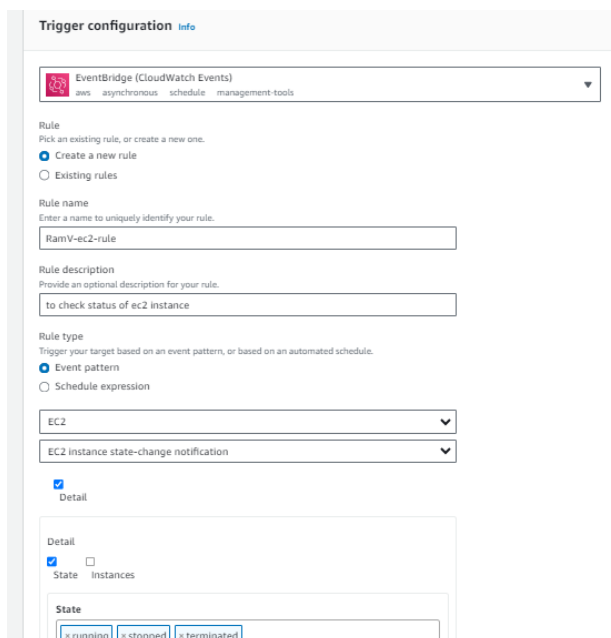
## Screenshot: Lambda Function



## 4. EC2 Event Bridge (formerly Cloud Watch Events):

- Set up an Event Bridge rule to trigger your Lambda function whenever an EC2 instance state changes.

## Screenshot: EventBridge rule configuration



RamV-ec2-rule

Edit

Disable

Delete

CloudFormation Template

Rule details

Rule name

RamV-ec2-rule

Description

to check status of ec2 instance

Status

Enabled

Rule ARN

arn:aws:events:ap-south-1:295397358094:rule/RamV-ec2-rule

Event bus name

default

Event bus ARN

arn:aws:events:ap-south-1:295397358094:event-bus/default

Type

Standard

Event pattern

Targets

Monitoring

Tags

Event pattern

```

1 {
2   "source": ["aws:ec2"],
3   "detail-type": ["EC2 Instance State-change Notification"],
4   "detail": {
5     "state": ["running", "stopped", "terminated"],
6     "instance-id": ["i-0afba499b4e034fee"]
7   }
8 }

```

Copy

## 5. Testing:

- Start or stop one of your EC2 instances.
- Confirm you receive an SNS notification about the state change.

**Screenshot:** State of Ec2 instance Ram\_LBcheck\_1 before testing

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	Grp-11-web-app-backend	i-0d1c22748f0802c71	Stopped	t2.micro	–	No alarms	ap-south-1b	–
<input type="checkbox"/>	–	i-0d9b4b365af025b8a	Stopped	t2.micro	–	No alarms	ap-south-1b	–
<input type="checkbox"/>	Boto3-Grp-12	i-0d39ae77e7cc59eb1	Stopped	t3.micro	–	No alarms	ap-south-1a	–
<input checked="" type="checkbox"/>	Ram_LBCheck_1	i-0afba499b4e034fee	Running	t2.micro	–	No alarms	ap-south-1a	ec2-13-127-196-110.ap...
<input type="checkbox"/>	Ram_LBCheck_2	i-091593dba41cbef8a	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-0-98-115.ap-so...

**Screenshot:** State changed from running to stopped.

<input type="checkbox"/>	Ram_LBCheck_1	i-0afba499b4e034fee	Stopped	t2.micro	–	No alarms	ap-south-1a	–	–
<input type="checkbox"/>	Ram_LBCheck_2	i-091593dba41cbef8a	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-0-98-115.ap-so...	65.0.98.115

**Screenshot:** Notified through email about state change an related information

AWS Notifications

<no-reply@sns.amazonaws.com>

to me

1:39 PM (3 minutes ago)

☆

↶

⋮

{

"Type": "Notification",

"MessageId": "008263d3-9aa9-57cd-ab2a-3e7e13230470",

"TopicArn": "arn:aws:sns:ap-south-1:295397358094:RamVenu\_SNS",

"Message": "EC2 Instance i-0afba499b4e034fee State-change Notification: stopped",

"Timestamp": "2023-10-21T08:09:46.905Z",

"SignatureVersion": "1",

"Signature": "c3BMq0tOP5DXIF2DXVejhrg5VXe+hKryX02735SZspoQqaPt8bdaVAeaZwyYWF5xYoHeFM1OCLNrap/Hh5VK3wEVEfl8Yjsd13mzMbUrUvwW3bnryrw8e4oAA2sxue/AzNpAMc0mvZhgbNzqNAGENRPxHgi0hgLH9jk46kiU8L6eERR4FMgV1qQgnhTdXuyjmOsaJVhPX7Q8ue/ugEokDeXGCOfoQpQK4+TYZ7gXKDaMwSCbbAUDts+vcW/wBoHVTRRismo4ifDUWEumdK4UKTWn8ISuBliffTWJRjoeJQYm2jNu5GWM0MJl6m9bezd31V7R24SpXYai4Lz/rJa114Q==",

\*\*\*

↶ Reply

↷ Forward