

# EBS Volume Backup Automation using Lambda, EventBridge and SNS (simple notification service)

## Introduction

This project automates the creation of EBS volume backups (snapshots) using AWS Lambda. The function is triggered automatically by Amazon EventBridge, and once the snapshot is created, a notification is sent to your email using Amazon SNS.

This ensures regular, reliable, hands-free backups without manual work.

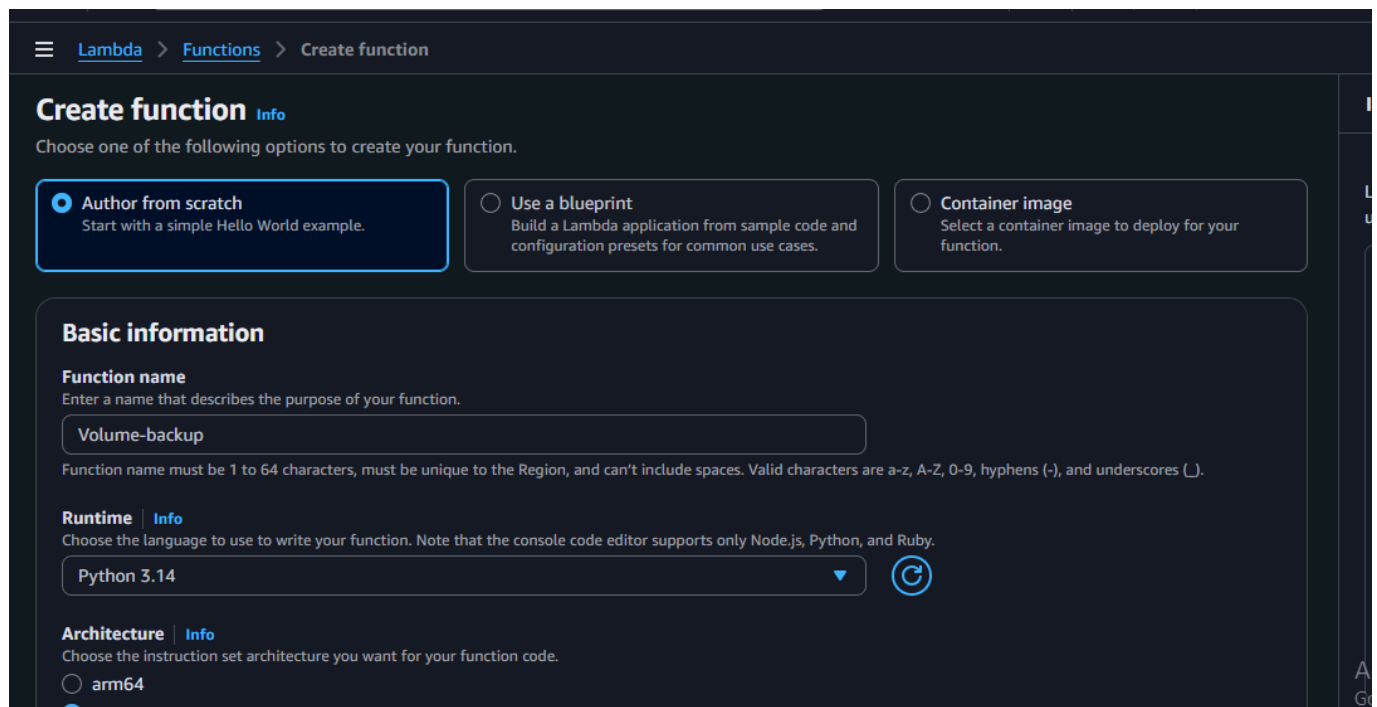
## Project Objective

- Automatically create EBS volume backups.
- Schedule backup using EventBridge (cron).
- Send email notifications via SNS.
- Reduce manual effort and prevent human error.

## Architecture

### Step 1: Create a Lambda Function

- Open AWS Lambda console -> Click create function -> Author from scratch



The screenshot shows the AWS Lambda 'Create function' page. The 'Author from scratch' option is selected. The 'Basic information' section is visible, showing the function name 'Volume-backup', runtime 'Python 3.14', and architecture 'arm64'.


**Create function** [Info](#)

Choose one of the following options to create your function.

- ☒ **Author from scratch**  
Start with a simple Hello World example.
- ☐ **Use a blueprint**  
Build a Lambda application from sample code and configuration presets for common use cases.
- ☐ **Container image**  
Select a container image to deploy for your function.

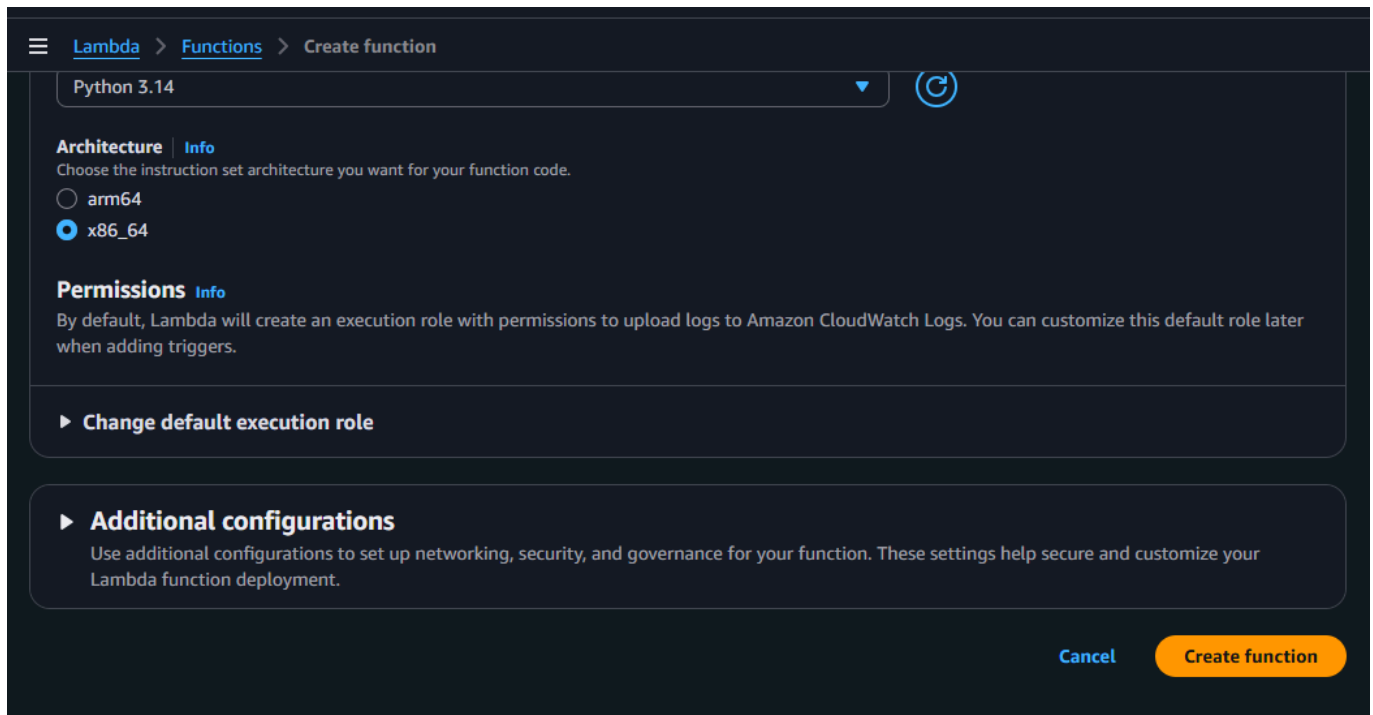
**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.  
  
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

**Runtime** [Info](#)  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
 

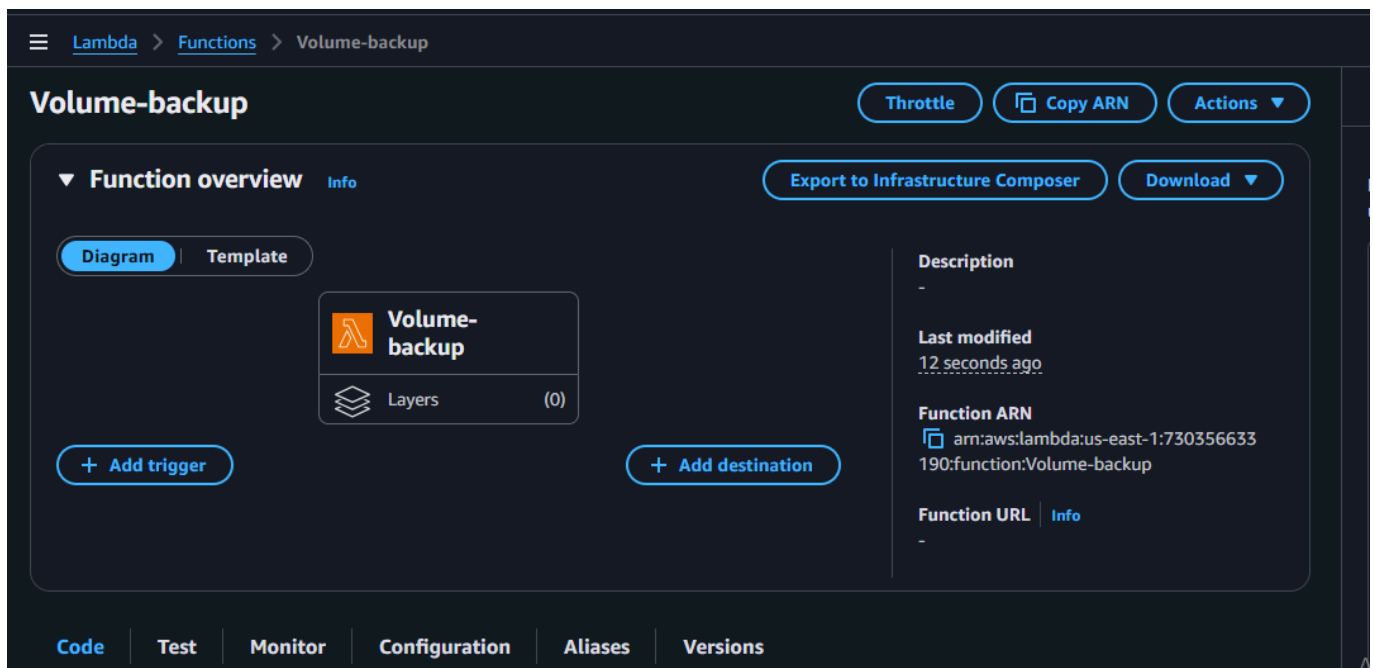
**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  
☐ arm64

- Runtime → Python 3.14
- Architecture → x86\_64
- Default execution role
- Click Create Function



The screenshot shows the 'Create function' page in the AWS Lambda console. The breadcrumb navigation at the top reads 'Lambda > Functions > Create function'. Below this, there's a dropdown menu for the runtime, currently set to 'Python 3.14'. To the right of the dropdown is a circular refresh icon. The main content area is divided into sections: 'Architecture' with an 'Info' link, a description 'Choose the instruction set architecture you want for your function code.', and two radio button options: 'arm64' and 'x86\_64', with 'x86\_64' being selected. Below this is the 'Permissions' section with an 'Info' link, a description 'By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.', and a link 'Change default execution role'. At the bottom of the main content area is a link 'Additional configurations' with a description 'Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.' At the bottom right of the page are two buttons: 'Cancel' and 'Create function'.

Here, Successfully created Lambda Function



The screenshot shows the 'Volume-backup' function page in the AWS Lambda console. The breadcrumb navigation at the top reads 'Lambda > Functions > Volume-backup'. The page title is 'Volume-backup'. On the right side, there are three buttons: 'Throttle', 'Copy ARN', and 'Actions'. Below these are two more buttons: 'Export to Infrastructure Composer' and 'Download'. The main content area is divided into two sections: 'Function overview' and 'Description'. The 'Function overview' section has two tabs: 'Diagram' and 'Template', with 'Diagram' being selected. Below the tabs is a diagram showing the function 'Volume-backup' with a stack of layers below it, labeled 'Layers (0)'. There are two buttons: '+ Add trigger' and '+ Add destination'. The 'Description' section has a 'Description' field with a minus sign, a 'Last modified' field with the value '12 seconds ago', a 'Function ARN' field with the value 'arn:aws:lambda:us-east-1:730356633:190:function:Volume-backup', and a 'Function URL' field with a minus sign and an 'Info' link. At the bottom of the page are six tabs: 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions', with 'Code' being selected.

Step 2: Add Code for Creating EBS Snapshot

```
import boto3

def lambda_handler(event, context):
    regions = []
    # Get a list of all regions
    ec2_client = boto3.client('ec2')
    regions_response = ec2_client.describe_regions()

    for region in regions_response['Regions']:
        regions.append(region['RegionName'])
```

```

snapshots_created = []

# Iterate through each region
for region in regions:
    print(f"Processing region: {region}")
    ec2 = boto3.client('ec2', region_name=region)

    # Get all volumes in 'in-use' state
    volumes = ec2.describe_volumes(
        Filters=[{'Name': 'status', 'Values': ['in-use']}]
    )['Volumes']

    for volume in volumes:
        volume_id = volume['VolumeId']
        print(f"Creating snapshot for Volume: {volume_id}")

        # Create a snapshot
        try:
            snapshot = ec2.create_snapshot(
                VolumeId=volume_id,
                Description=f"Snapshot of {volume_id} from region {region}"
            )

            snapshots_created.append({
                "Region": region,
                "VolumeId": volume_id,
                "SnapshotId": snapshot['SnapshotId']
            })

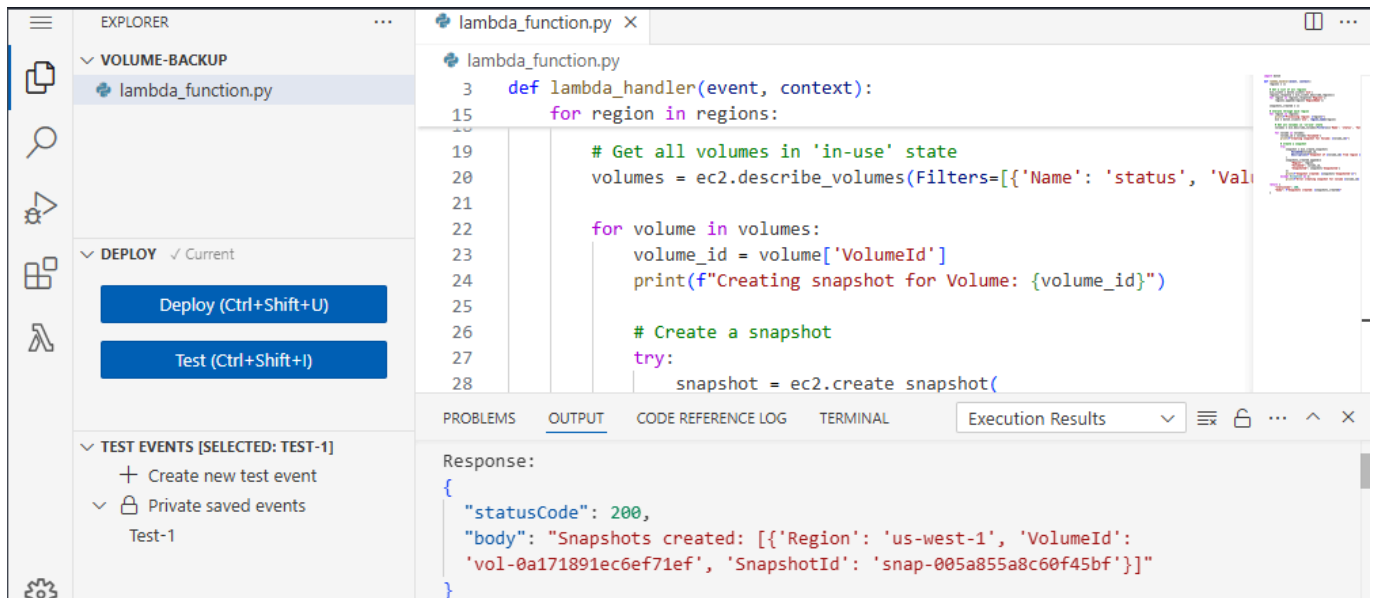
            print(f"Snapshot created: {snapshot['SnapshotId']}")

        except Exception as e:
            print(f"Error creating snapshot for volume {volume_id} in region {region}: {str(e)}")

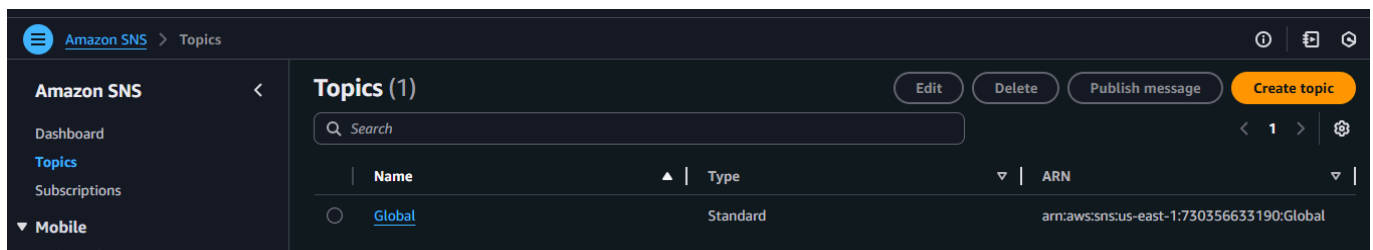
    return {
        "statusCode": 200,
        "body": f"Snapshots created: {snapshots_created}"
    }

```

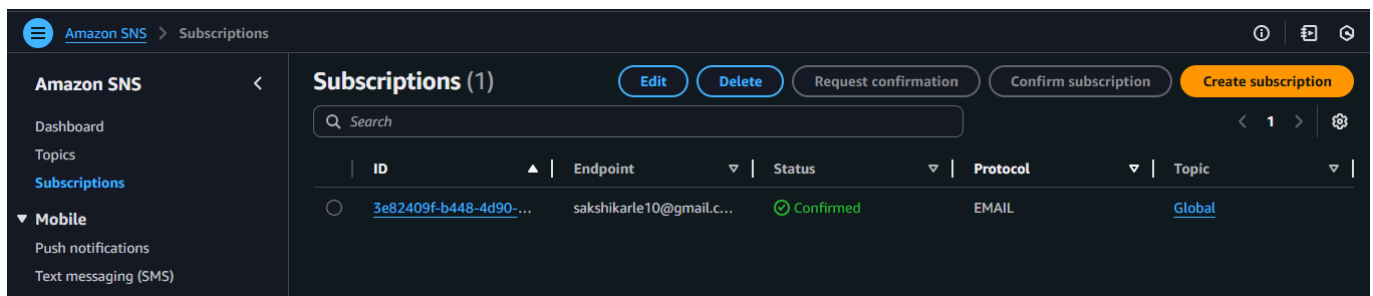
- After the Lambda function is created, scroll down the code section and remove default code and paste above code.
- Click Deploy



### Step 3: Create SNS Topic for Email Notification



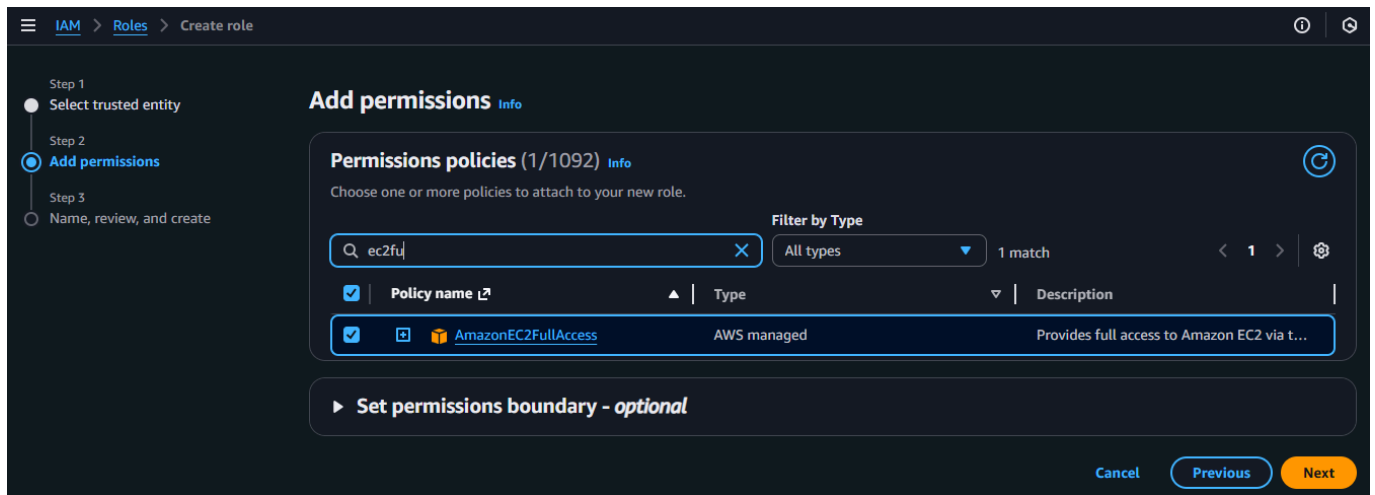
- Scroll down -> Create subscription in protocol enter your email address
- Go to your inbox and click confirm subscription



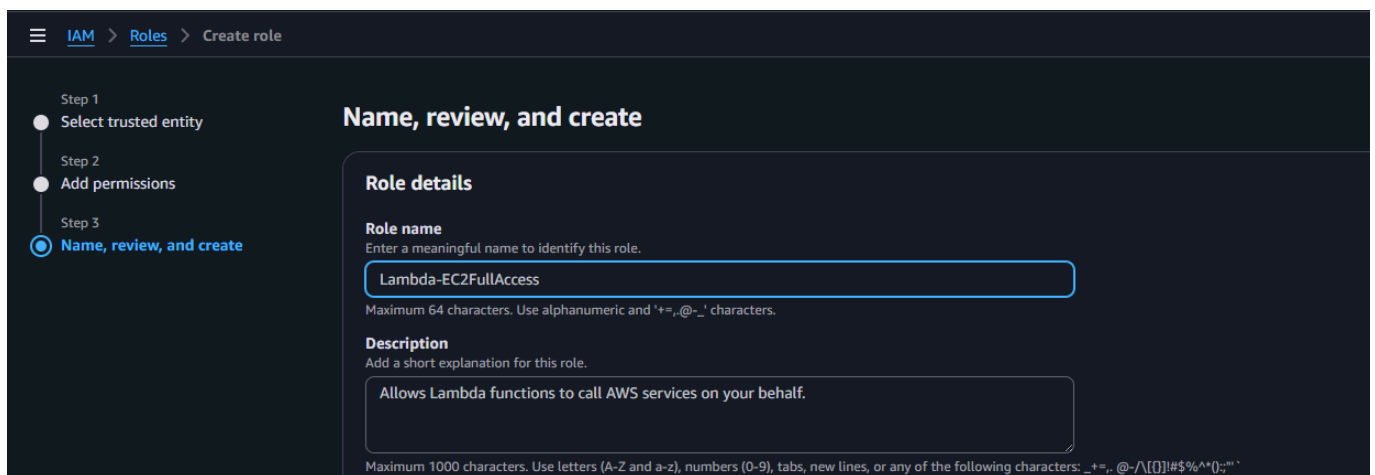
### Step 4: Attach Permissions to Lambda Role

- Open IAM Console → Roles.
- Search for the role created for your Lambda → example: demo-function-role.
- Click Attach policies.
- Add

AmazonEC2FullAccess



- Enter the role name



- Attach Role to your Lambda Function Open Lambda Console → Functions
  1. Select your function: Volume-backup
  2. Go to Configuration → Permissions
  3. Click Edit under Execution Role
  4. Select:
  5. Use an existing role

**Lambda > Functions > Volume-backup > Edit basic settings**

**SnapStart** **Info**  
 Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#). For Python and .NET runtimes, [view pricing](#).

None

Supported runtimes: .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Python 3.12, Python 3.13, Python 3.14.

**Timeout**  
 1 min 30 sec

**Execution role**  
 Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☒ Use an existing role  
☐ Create a new role from AWS policy templates

**Existing role**  
 Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

Lambda-EC2FullAccess

[View the Lambda-EC2FullAccess role](#) on the IAM console.

Cancel Save

## Step 5: Verify Snapshot in EC2 Console

- Go to AWS EC2 Console.
- On the left panel → Click Snapshots under Elastic Block Store (EBS).

You will see the automatically created snapshot from your Lambda function.

**Snapshots (1)** **Info** Last updated less than a minute ago [Recycle Bin](#) [Actions](#) [Create snapshot](#)

Owned by me

<input type="checkbox"/>	Name	Snapshot ID	Full snapshot size	Volume size	Description	Storage tier	S
<input type="checkbox"/>		snap-005a855a8c60f45bf	1.65 GiB	8 GiB	Snapshot of vol-0a171891...	Standard	

To create an EBS volume, you must have at least one EC2 instance running, because EBS volumes attach to EC2 instances.

## Step 6: Create EventBridge Rule (Trigger Lambda Automatically)

- Click Create Rule.
- Rule name → rule 1

The screenshot shows the 'Define rule detail' step in the Amazon EventBridge console. The left sidebar contains navigation links for Dashboard, Developer resources, Buses, Pipes, and Scheduler. The main content area is titled 'Define rule detail' and includes a progress bar with five steps: Define rule detail (selected), Define schedule, Select target(s), optional Configure tags, and Review and create. The 'Rule detail' section contains a 'Name' field with 'rule-1', a 'Description' field, and an 'Event bus' dropdown set to 'default'. The 'Rule type' section has two options: 'Rule with an event pattern' and 'Schedule' (selected). The 'Schedule' option is highlighted with a blue border and contains the text 'Activate Windows'.

- Choose Schedule → rate

Example (10 minutes):

The screenshot shows the 'Define schedule' step in the Amazon EventBridge console. The left sidebar is the same as the previous screenshot. The main content area is titled 'Define schedule' and includes a progress bar with five steps: Define rule detail, Define schedule (selected), Select target(s), optional Configure tags, and Review and create. The 'Schedule pattern' section has two options: 'A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.' and 'A schedule that runs at a regular rate, such as every 10 minutes.' (selected). The 'Rate expression' section shows a 'rate' button, a text input field with '10', and a dropdown menu set to 'Minutes'. The 'Previous' and 'Next' buttons are visible at the bottom right.

- Choose Target → sns topic

**Amazon EventBridge** > **Rules** > Create rule

**Amazon EventBridge**

- Dashboard
- Developer resources
  - Learn
  - Sandbox
  - Quick starts
- Buses
  - Event buses **Updated**
  - Rules**
  - Global endpoints
  - Archives
  - Replays
- Pipes
  - Pipes
- Scheduler

**Step 1** Define rule detail  
**Step 2** Define schedule  
**Step 3** **Select target(s)**  
Step 4 - optional Configure tags  
Step 5 Review and create

### Select target(s)

**Permissions**  
Note: When using the EventBridge console, EventBridge will automatically configure the proper permissions for the selected targets. If you're using the AWS CLI, SDK, or CloudFormation, you'll need to configure the proper permissions.

**Target 1**

**Target types**  
Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

- ☐ EventBridge event bus
- ☐ EventBridge API destination
- ☒ **AWS service**

**Select a target** **Info**  
Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)

SNS topic

☒ Target in this account ☐ Target in another AWS account

**Topic**  
Global

**Permissions**  
☒ Use execution role (recommended)

**Execution role**  
EventBridge needs permission to send events to the target specified above. By continuing, you are allowing us to do so.  
[EventBridge and AWS Identity and Access Management](#)

☒ Create a new role for this specific resource ☐ Use existing role

**Role name**  
Amazon\_EventBridge\_Invoke\_Sns\_1322804103

**Additional settings**

**Buttons:** Add another target, Cancel, Skip to Review and create, Previous, Next

- Click Create Rule. Here, Rule is Created

**Amazon EventBridge** > **Rules**

with one or more targets.

### Select event bus

**Event bus**  
Select or enter event bus name

default

**Rules on default event bus (1)** **Refresh** **Delete** **Enable** **Edit** **CloudFormation Template** **Create rule**

**Find rules** **Any status** **< 1 >** **Settings**

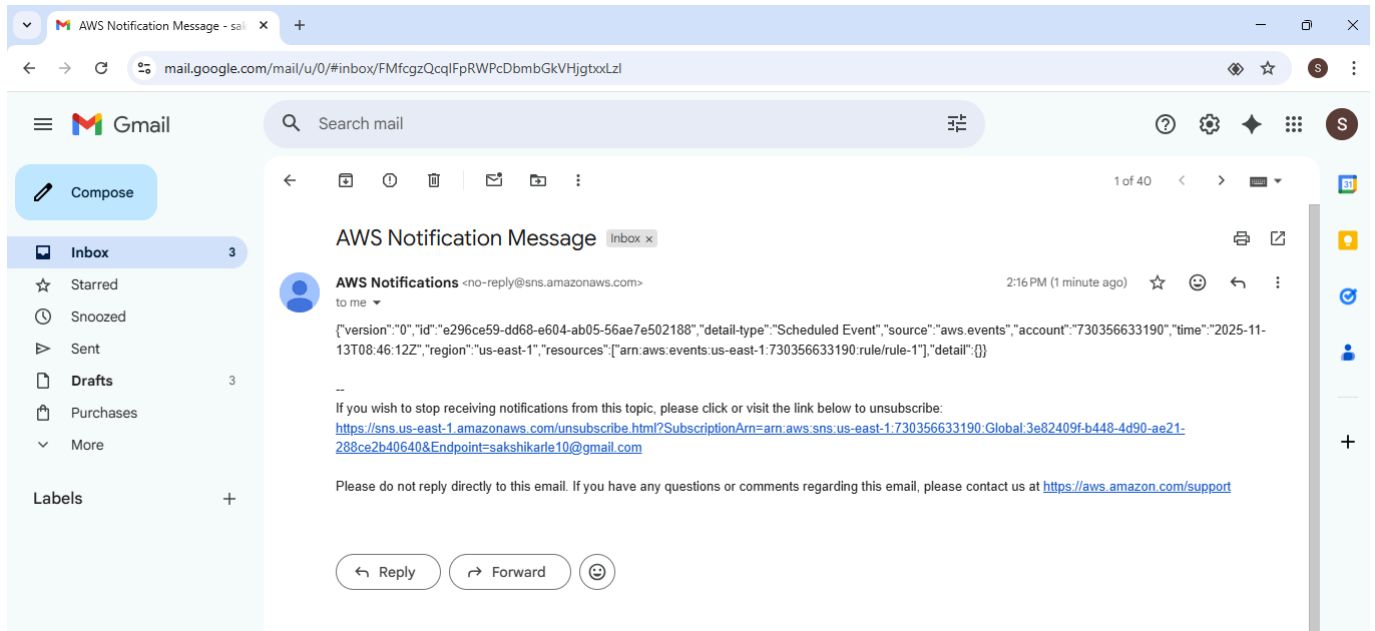
<input type="checkbox"/>	Name	Status	Type	Event bus	ARN	Description
<input type="checkbox"/>	<a href="#">rule-1</a>	Enabled	Scheduled Standard	default	arn:aws:events:us-east-1:730356633190:rule/rule-1	-

**Buttons:** Activate Windows, Go to Settings to activate Windows

## Step 7: Check SNS Email Notification

1. Open your email inbox.
2. You should receive a message from SNS showing: Snapshot ID, Date & Time

Confirmation that backup was created successfully



## Conclusion:

In this project, we automated the process of creating EBS snapshots across all AWS regions using AWS Lambda and EventBridge. The Lambda function identifies all in-use EBS volumes and automatically creates snapshots for backup and recovery. EventBridge triggers the Lambda function on a scheduled basis, ensuring regular and consistent backups without manual effort. This automation improves data protection, reduces operational overhead, and ensures reliable disaster recovery for critical EC2 instances.