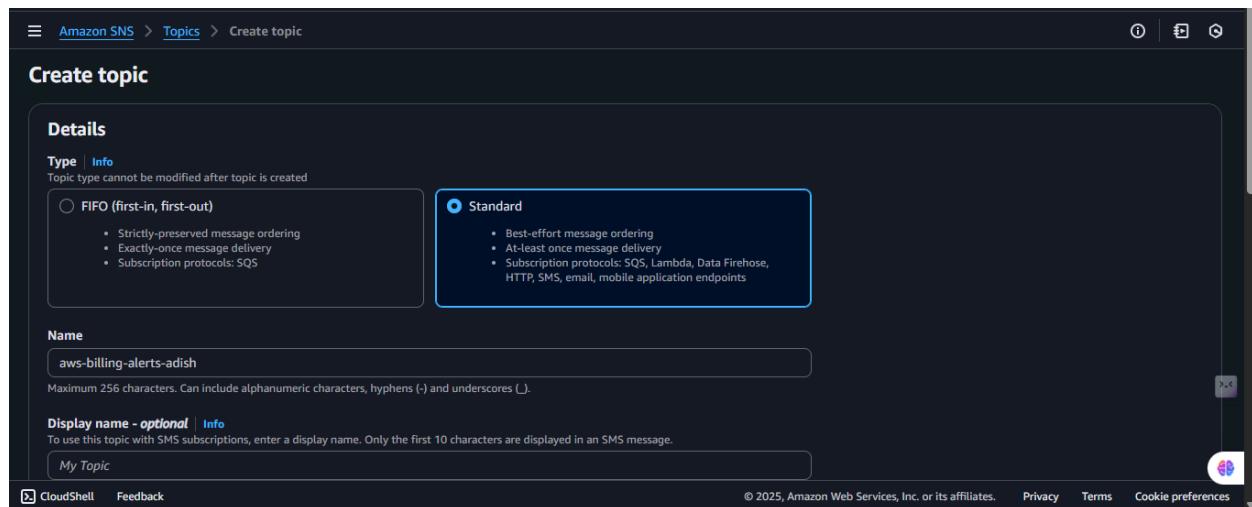


1. Create an SNS Topic

1. Go to **Amazon SNS → Topics → Create topic**
2. Type: **Standard**
3. Name: **aws-billing-alerts**
4. Click **Create topic**
5. Under **Subscriptions**, create a new one:
 - Protocol: **Email**
 - Endpoint: your email address
 - Confirm the subscription via the email link you receive.



Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN
arn:aws:sns:eu-west-2:975050024946:aws-billing-alerts-adish

Protocol
The type of endpoint to subscribe
Email

Endpoint
An email address that can receive notifications from Amazon SNS.
Adishansari786@gmail.com

ⓘ After your subscription is created, you must confirm it. [Info](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon SNS > Topics > aws-billing-alerts-adish

aws-billing-alerts-adish

Amazon SNS <

Topics

Subscriptions

Mobile

Push notifications

Text messaging (SMS)

Details

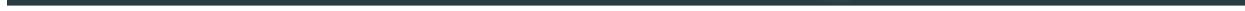
Name	aws-billing-alerts-adish	Display name	-
ARN	arn:aws:sns:eu-west-2:975050024946:aws-billing-alerts-adish	Topic owner	975050024946
Type	Standard		

Subscriptions | Access policy | Data protection policy | Delivery policy (HTTP/S) | Delivery status logging

Subscriptions (1)

Edit Delete Request confirmation Confirm subscription Create subscription

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



AWS Notification - Subscription Confirmation



AWS Notifications <no-reply@sns.amazonaws.com>
to me ▾

Sun, Oct 19, 11:10 AM (1 day ago)



You have chosen to subscribe to the topic:

arn:aws:sns:eu-west-2:975050024946:aws-billing-alerts-adish

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)



Write your reply to generate with AI

Yes

No

Follow up

Reply

Forward



AI Reply

2. Create IAM Role for Lambda

Permissions required:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ce:GetCostAndUsage",  
                "sns:Publish"  
            ],  
            "Effect": "Allow",  
            "Resource": "*"  
        }  
    ]  
}
```

Attach this policy to a new **Lambda execution role** (e.g., [LambdaBillingMonitorRole](#)).

3. Create the Lambda Function

Runtime: Python 3.13

Handler: `lambda_function.lambda_handler`

Lambda > Functions > Create function

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.
BillingMonitorLambda

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture Info
Choose the instruction set architecture you want for your function code.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Lambda > Functions > BillingMonitorLambda

Code Test Monitor Configuration Aliases Versions

General configuration Info

Description
-

Memory
128 MB

Ephemeral storage
512 MB

Timeout
1 min 0 sec

SnapStart Info
None

Edit

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Lambda > Functions > BillingMonitorLambda

Code Test Monitor Configuration Aliases Versions

General configuration

Environment variables (2)

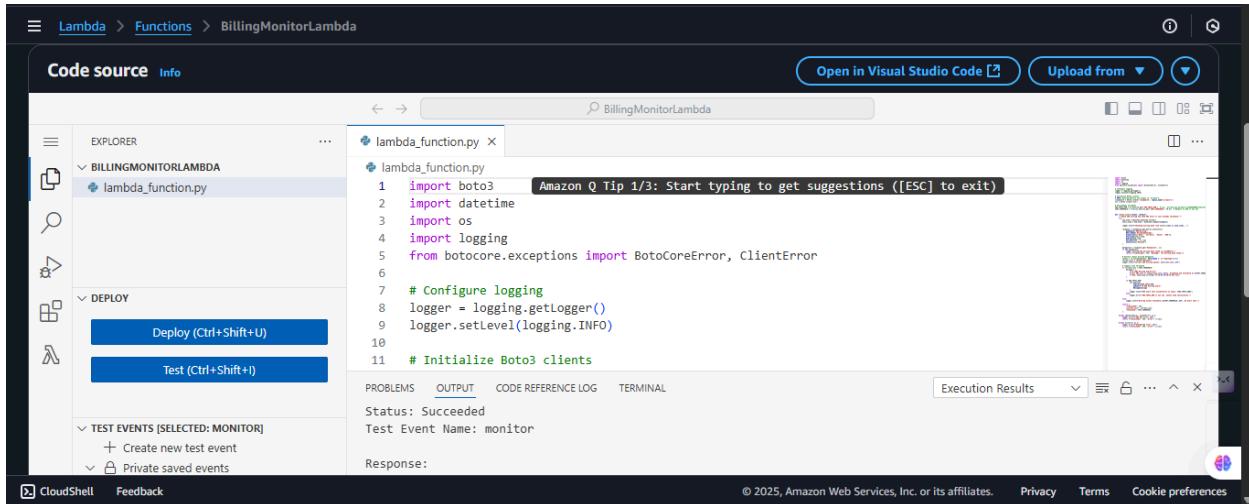
The environment variables below are encrypted at rest with the default Lambda service key.

Find environment variables

Key	Value
COST_THRESHOLD	50
SNS_TOPIC_ARN	arn:aws:sns:eu-west-2:975050024946:aws-billing-alerts-adish

Edit

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```
import boto3
import datetime
import os
import logging
from botocore.exceptions import BotoCoreError, ClientError

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)

# Initialize Boto3 clients
# NOTE: Billing metrics are always in 'us-east-1'
cloudwatch = boto3.client('cloudwatch', region_name='us-east-1')
sns = boto3.client('sns')

# Environment variables
SNS_TOPIC_ARN = os.environ.get('SNS_TOPIC_ARN') # e.g.,
arn:aws:sns:eu-west-2:123456789012:aws-billing-alerts
COST_THRESHOLD = float(os.environ.get('COST_THRESHOLD', 50.0)) # Default
to $50 if not set

def lambda_handler(event, context):
    """Check AWS billing and send SNS alert if cost exceeds threshold."""
```

```

try:
    end_time = datetime.datetime.utcnow()
    start_time = end_time - datetime.timedelta(days=1)

    logger.info(f"Checking billing data from {start_time} to
{end_time}...")

    response = cloudwatch.get_metric_statistics(
        Namespace='AWS/Billing',
        MetricName='EstimatedCharges',
        Dimensions=[{'Name': 'Currency', 'Value': 'USD'}],
        StartTime=start_time,
        EndTime=end_time,
        Period=86400, # 1 day
        Statistics=['Maximum']
    )

    datapoints = response.get('Datapoints', [])
    if not datapoints:
        logger.warning("No billing data found in CloudWatch.")
        return {"statusCode": 204, "message": "No billing data
found."}

    # Extract latest billing datapoint
    latest = sorted(datapoints, key=lambda x: x['Timestamp'])[-1]
    current_cost = latest['Maximum']
    logger.info(f"Current AWS billing amount: ${current_cost:.2f}")

    # Compare with threshold
    if current_cost > COST_THRESHOLD:
        message = (
            f"⚠ *AWS Billing Alert*\n\n"
            f"Your AWS cost is **${current_cost:.2f}**, exceeding the
threshold of ${COST_THRESHOLD:.2f}.\n"
            f"Time: {end_time.strftime('%Y-%m-%d %H:%M:%S UTC')}\n"
        )

        if SNS_TOPIC_ARN:
            sns.publish(
                TopicArn=SNS_TOPIC_ARN,

```

```

        Subject="🚨 AWS Billing Alert",
        Message=message
    )
    logger.info(f"SNS alert sent successfully to topic:
{SNS_TOPIC_ARN}")
else:
    logger.error("SNS_TOPIC_ARN is not set. Cannot send
notification.")

else:
    logger.info(f"Billing within threshold
(${COST_THRESHOLD:.2f}). No alert sent.")

return {
    "statusCode": 200,
    "current_cost": current_cost,
    "threshold": COST_THRESHOLD
}

except (BotoCoreError, ClientError) as e:
    logger.error(f"AWS API error: {e}")
    return {"statusCode": 500, "error": str(e)}

except Exception as e:
    logger.error(f"Unexpected error: {e}")
    return {"statusCode": 500, "error": str(e)}

```

Step 4: Set Up Event Source (Daily Trigger)

1. Go to **Amazon EventBridge** → **Rules** → **Create rule**.
2. **Name:**
`daily-aws-billing-check`
3. **Description:**
`Triggers Lambda function daily to monitor AWS billing cost`
4. **Rule type:**
`Schedule`

5. Schedule pattern:

Choose “A schedule pattern that runs at a regular rate or cron expression.”

Expression:

```
cron(0 0 * * ? *) # Runs every day at 12:00 AM UTC
```

6. Adjust if you want a different time:

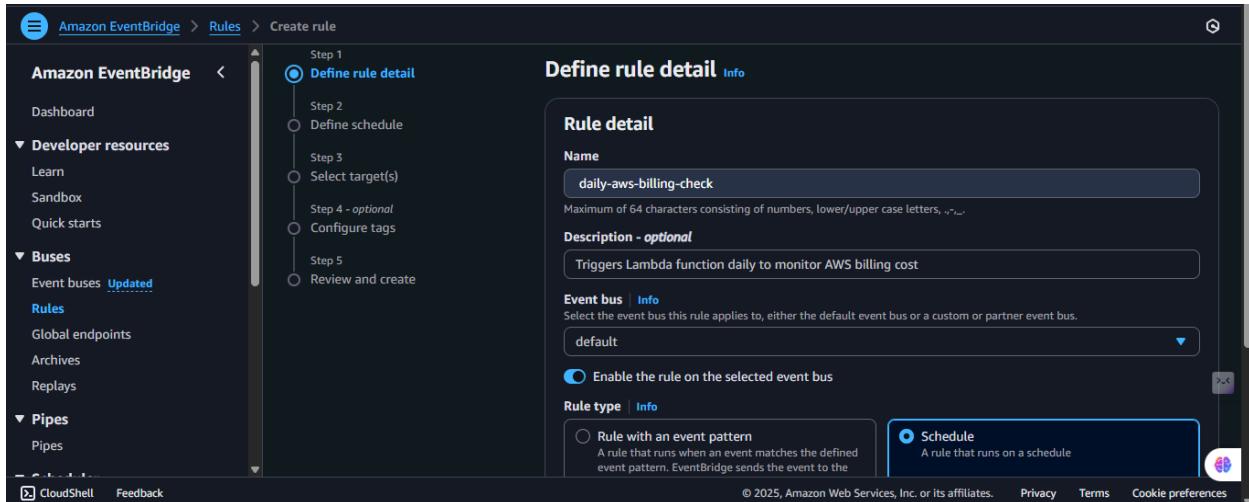
- `cron(0 6 * * ? *)` → runs at 6 AM UTC daily
- `rate(1 day)` → simpler option, runs every 24 hours.

7. Target:

- Choose **Lambda function**
- Select your function: `aws-billing-alert-lambda`

8. Permissions:

- Allow EventBridge to invoke the Lambda function.



Amazon EventBridge > Schedules > Create schedule

Specify schedule detail

Schedule name and description

Schedule name: daily-aws-billing-check

Description - optional: Triggers Lambda function daily to monitor AWS billing cost

Schedule group: default

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EventBridge > Schedules > Create schedule

Schedule pattern

Occurrence | Info: You can define an one-time or recurrent schedule.

One-time schedule (selected) Recurring schedule

Time zone: (UTC+05:30) Asia/Calcutta

Schedule type: Cron-based schedule (selected) Rate-based schedule

Cron expression | Info: Define the cron expression for the schedule

cron (2 Minutes * Hours * Day of month * Month ? Day of the week * Year)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EventBridge > Schedules > Create schedule

Step 1 Specify schedule detail

Step 2 Select target (selected)

Step 3 - optional Settings

Step 4 Review and create schedule

Select target

Target detail

Target API | Info: Select an API that will be invoked as a target for your schedule.

Templated targets (selected) All APIs

CodeBuild StartBuild

CodePipeline StartPipelineExecu...

Amazon ECS RunTask

Amazon EventBridge PutEvents

Amazon Inspector V1 StartAssessmentRun

Kinesis Data Firehose PutRecord

Kinesis Data Streams PutRecord

AWS Lambda Invoke (selected)

Amazon SNS Publish

Amazon SQS SendMessage

SageMaker StartPipelineExecu...

AWS Step Functions StartExecution

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EventBridge > Schedules > Create schedule

Invoke

AWS Lambda

Lambda function

BillingMonitorLambda

Configure version/alias

Payload

The JSON that you want to provide to your Lambda function as input. For example, '{ key: value }'. [Learn more](#)

1

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EventBridge > Schedules > Create schedule

Retry policy and dead-letter queue (DLQ)

Retry policy [Info](#)
Retry policy allows EventBridge Scheduler to re-run a schedule if it fails to invoke its target. You can specify the maximum age of the event and the maximum number of times to retry.
 Retry

Dead-letter queue (DLQ)
Standard Amazon SQS queues that EventBridge Scheduler uses to store events that couldn't be delivered successfully to a target.
 None
 Select an Amazon SQS queue in my AWS account as a DLQ
 Specify an Amazon SQS queue in other AWS accounts as a DLQ

Encryption [Info](#)
By default, EventBridge Scheduler encrypts event metadata and message data that it stores under an AWS owned key (encryption at rest). EventBridge Scheduler also encrypts data that passes between EventBridge Scheduler and other services using Transport layer Security (TLS) (encryption in transit).
Your data is encrypted by default with a key that AWS owns and manages for you. To choose a different key, customize your encryption settings.
 Customize encryption settings (advanced)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EventBridge > Schedules > Create schedule

Encryption [Info](#)
By default, EventBridge Scheduler encrypts event metadata and message data that it stores under an AWS owned key (encryption at rest). EventBridge Scheduler also encrypts data that passes between EventBridge Scheduler and other services using Transport layer Security (TLS) (encryption in transit).
Your data is encrypted by default with a key that AWS owns and manages for you. To choose a different key, customize your encryption settings.
 Customize encryption settings (advanced)

Permissions [Info](#)

Permissions
EventBridge Scheduler requires permission to send events to the target, and based on the preferences you select, integrate with other AWS services such as AWS KMS and Amazon SQS.

Execution role
 Create new role for this schedule Use existing role

Role name
This is the role name we will be creating on your behalf. You can change the name.
Amazon_EventBridge_Scheduler_LAMBDA_235ca9e3c3

Go to IAM console [\[\]](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EventBridge > Schedules > daily_aws_billing_check

Your schedule daily_aws_billing_check is being created.

daily_aws_billing_check

Schedule detail

Schedule name daily_aws_billing_check	Status Enabled	Schedule start time -	Flexible time window -
Description Triggers Lambda function daily to monitor AWS billing cost	Schedule ARN arn:aws:scheduler:eu-west-2:133562846627:schedule/default/daily_aws_billing_check	Schedule end time -	Created date Oct 20, 2025, 14:17:57 (UTC+05:30)
Schedule group name default	Action after completion NONE	Execution time zone Asia/Calcutta	Last modified date Oct 20, 2025, 14:17:57 (UTC+05:30)

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

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon EventBridge > Schedules > daily_aws_billing_check

Your schedule daily_aws_billing_check is being created.

Schedule

Cron expression [Info](#)

2	*	*	*	?	*
Minutes	Hours	Day of month	Month	Day of week	Year

Copy cron expression

Next 10 trigger dates
Date and time are displayed in the selected time zone for which this schedule is set in UTC format, e.g. "Wed, Nov 9, 2022 09:00 (UTC -08:00)"

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences