Vivekanand Education Society's Institute of Technology, Chembur, Mumbai,

Department of Technology,

Year: 2024-2025(ODD SEM)

Advance DevOps Practical Examination

Name:Komal Vijay Singh **Div: D15B**

Roll No: 60 **Date of exam: 24/10/2024**

Case Study 20: Automated Notifications Using SNS

- **Concepts Used**: AWS Lambda, S3, SNS.
- **Problem Statement**: "Create a Lambda function that triggers when a new file is uploaded to an S3 bucket and sends an email notification using SNS with details of the uploaded file."
- **Tasks**:
  - Write a Python Lambda function that triggers on S3 upload events.
  - Extract the file name and size from the event and format a notification message.
  - Use SNS to send the notification to a configured email address.
  - Test by uploading a file to the S3 bucket and verifying that an email is received.

These case studies focus on AWS Lambda's event-driven capabilities, allowing students to explore how Lambda interacts with other AWS services like S3, DynamoDB, CloudWatch, and API Gateway. Each task provides practical experience with real-world scenarios and can be completed within a short timeframe.

Aim: The aim is to design a solution that automates the notification process when a file is uploaded to an Amazon S3 bucket. This is achieved by utilizing AWS Lambda to automatically trigger upon file uploads, extract the necessary file details (such as name and size), and send email notifications through Amazon SNS. The goal is to enhance real-time monitoring and automate alerting mechanisms for file uploads, thus improving system responsiveness and operational efficiency.

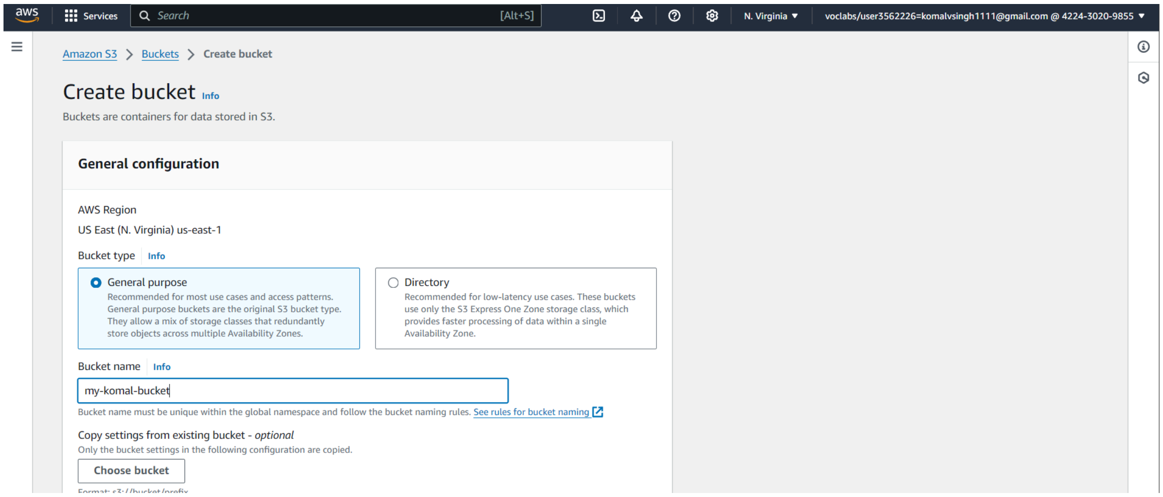
Implementation:

1. **Amazon S3 Setup:**
   - Create a new S3 bucket to store files that will trigger the notification process.
   - Configure S3 events to capture file upload actions (PUT requests).
2. **Amazon SNS Setup:**
   - Create an SNS topic to handle the distribution of notifications.
   - Subscribe an email address to this SNS topic, ensuring email notifications are received for each event.
3. **AWS Lambda Function Development:**
   - Write a Python-based Lambda function that will:
     - Trigger when a file is uploaded to the S3 bucket.
     - Extract the file's details (name and size) from the event payload.
     - Format a message that includes the file name and size.
     - Use the SNS service to publish this message to the subscribed email address.
4. **S3 Event Trigger:**
   - Associate the Lambda function with the S3 bucket, ensuring the function is invoked for every file upload event.
5. **Testing and Monitoring:**
   - Upload a test file to the S3 bucket to trigger the Lambda function.
   - Verify that the email notification is sent with the correct file details.
   - Monitor logs through AWS CloudWatch to confirm successful function execution.

Step 1: Create an S3 Bucket

1. **Login to AWS Management Console** in your Learner Lab environment.
2. **Go to S3** from the Services menu.
3. **Create a new bucket**:
   - Choose a bucket name (e.g., `my-file-upload-bucket`).
   - Select the appropriate region.

- Leave the rest of the settings as default unless specific customizations are needed.
- Click **Create Bucket**.


aws
Services
Search [Alt+S]
N. Virginia ▾
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▾

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

my-komal-bucket

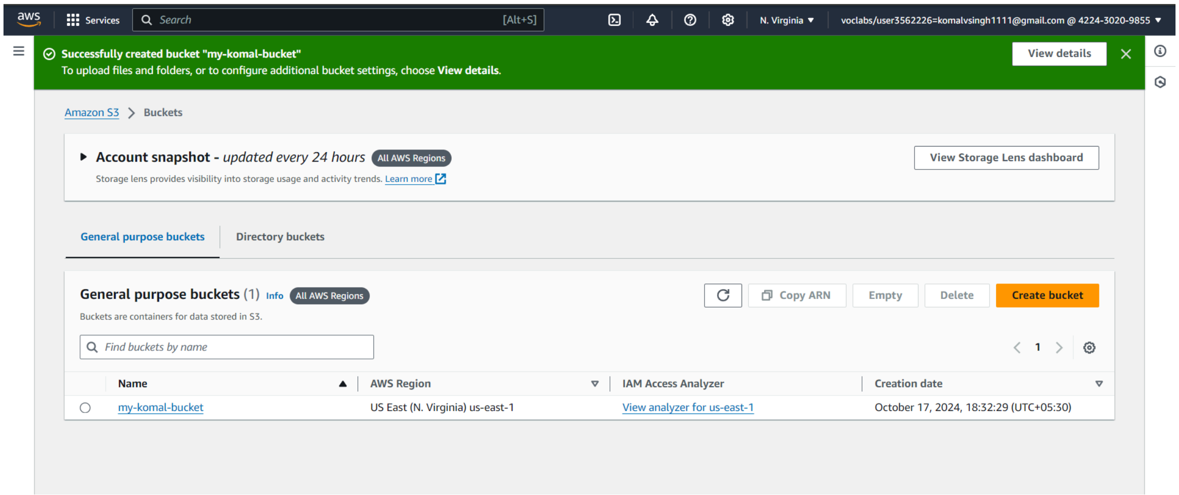
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix


aws
Services
Search [Alt+S]
N. Virginia ▾
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▾

Successfully created bucket "my-komal-bucket"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

View details

Amazon S3 > Buckets

▸ Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Learn more

View Storage Lens dashboard

General purpose buckets | Directory buckets

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

Copy ARN | Empty | Delete | Create bucket

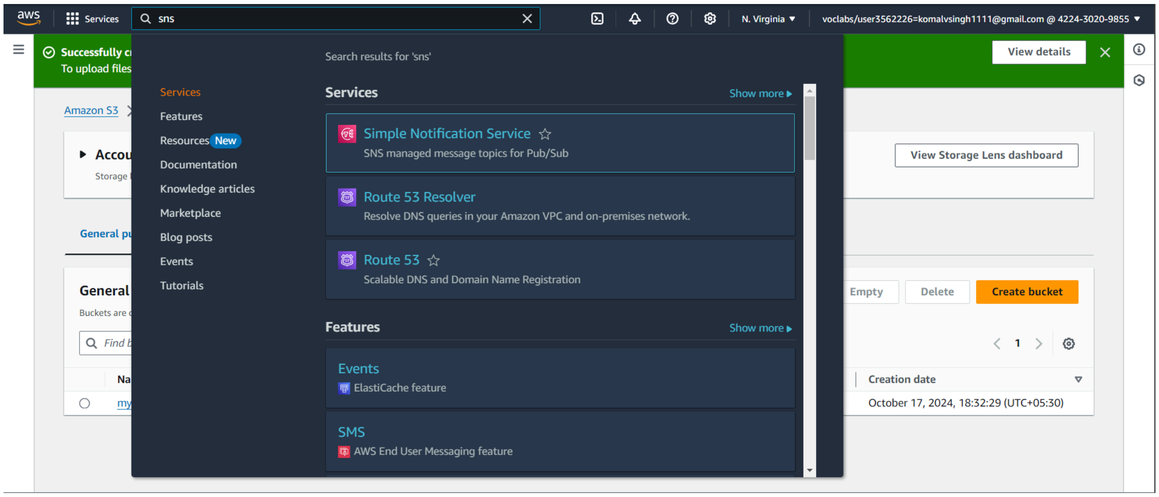
Find buckets by name

	Name ▲	AWS Region	IAM Access Analyzer	Creation date
○	my-komal-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 17, 2024, 18:32:29 (UTC+05:30)

Step 2: Create an SNS Topic

1. Go to **SNS** from the Services menu.
2. Click **Topics** on the left panel.
3. Click **Create Topic**.
   - Select **Standard** as the topic type.
   - Name the topic (e.g., `S3FileUploadNotification`).
   - Click **Create Topic**.
4. After creating the topic, click **Create Subscription**:
   - Choose the **Protocol** as `Email`.
   - Enter the **email address** where notifications should be sent.
   - Click **Create Subscription**.

- ○ Confirm the subscription by checking your email and clicking the confirmation link.
- ○ Replace `region` and `account-id` in `sns_topic_arn` with your values from the SNS topic.



Search: sns

Search results for 'sns'

Services

Simple Notification Service — SNS managed message topics for Pub/Sub

Route 53 Resolver — Resolve DNS queries in your Amazon VPC and on-premises network.

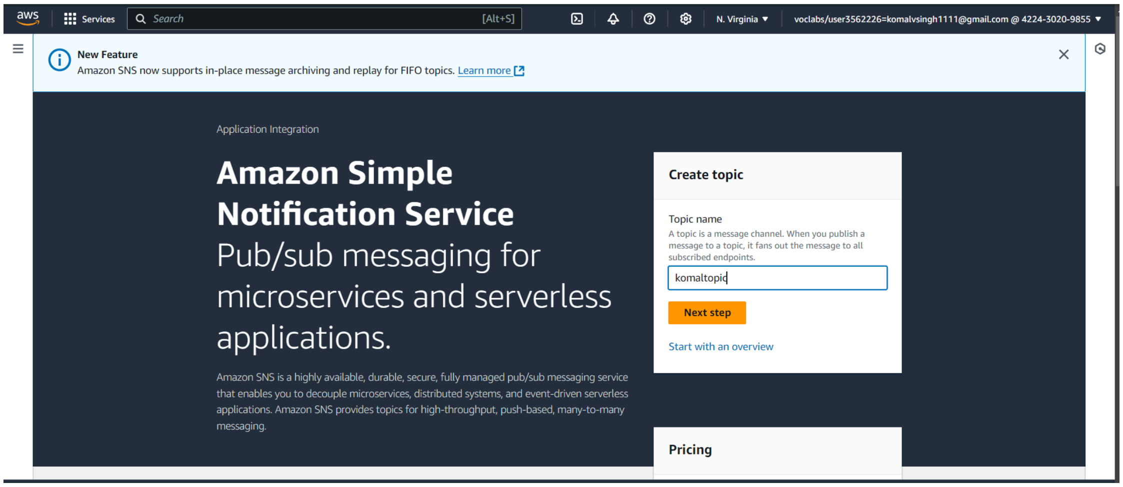
Route 53 — Scalable DNS and Domain Name Registration

Features

Events — ElastiCache feature

SMS — AWS End User Messaging feature

October 17, 2024, 18:32:29 (UTC+05:30)



New Feature

Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

Application Integration

Amazon Simple Notification Service

Pub/sub messaging for microservices and serverless applications.

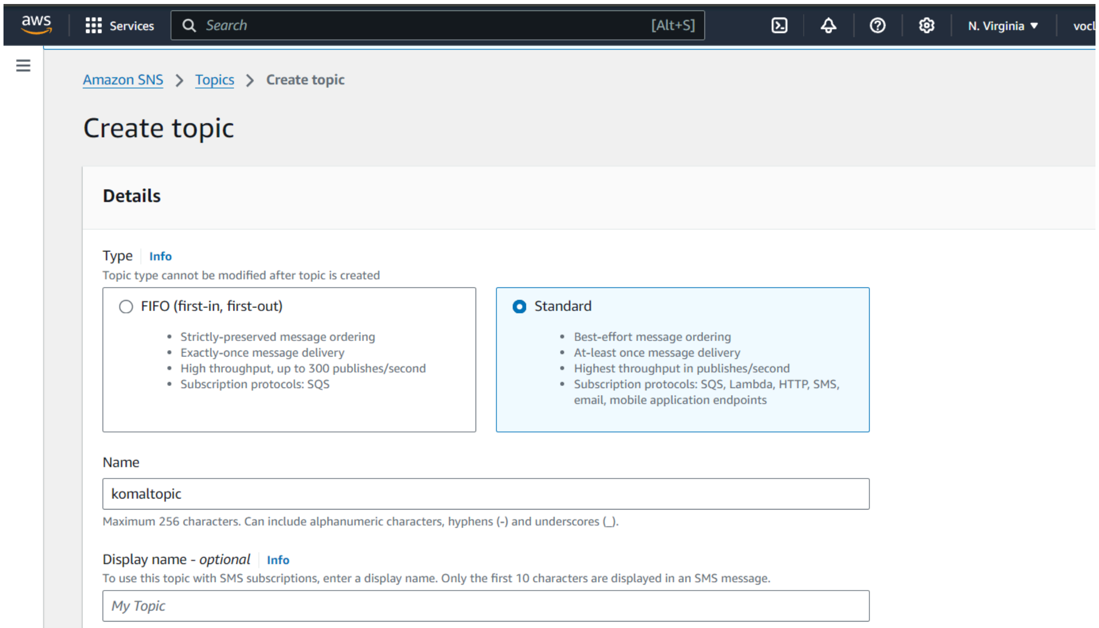
Amazon SNS is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and event-driven serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging.

Create topic

Topic name

A topic is a message channel. When you publish a message to a topic, it fans out the message to all subscribed endpoints.

komaltopic



Amazon SNS > Topics > Create topic

Create topic

Details

Type Info

Topic type cannot be modified after topic is created

○ FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

● Standard
- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

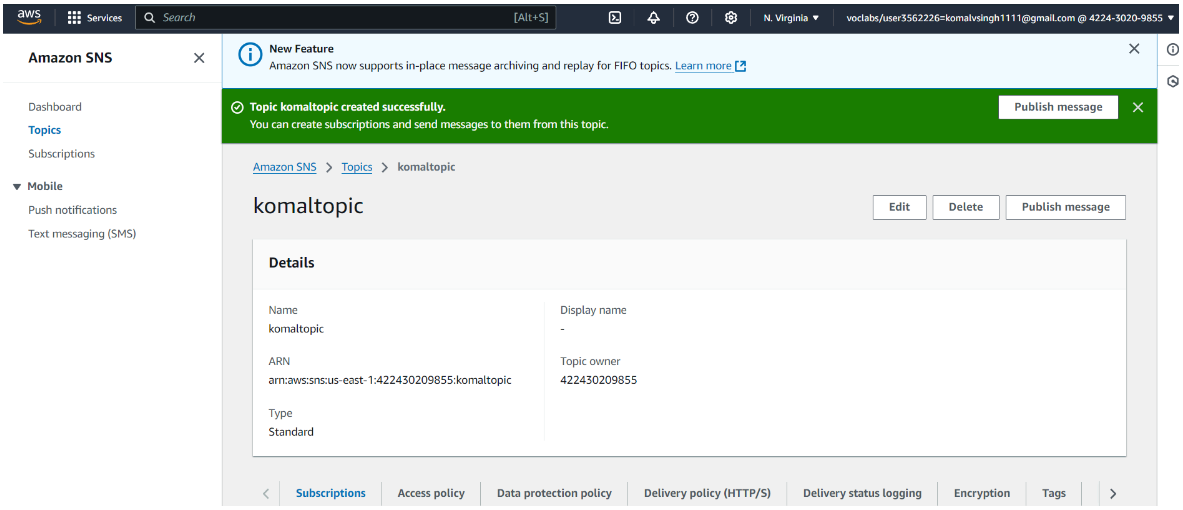
komaltopic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional* Info

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

My Topic



New Feature

Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

Topic komaltopic created successfully.

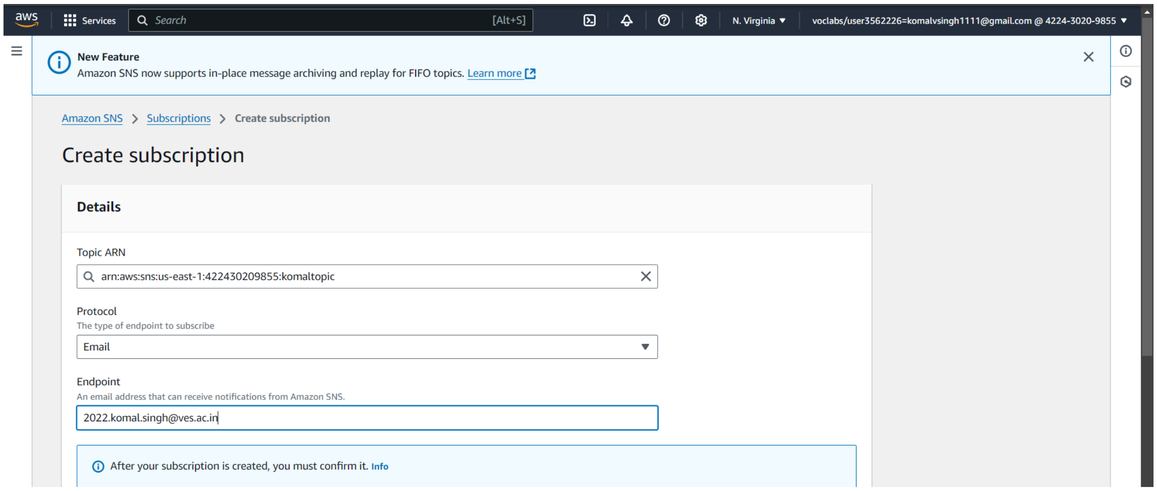
You can create subscriptions and send messages to them from this topic.

Amazon SNS > Topics > komaltopic

komaltopic

Details

Name komaltopic	Display name -
ARN arn:aws:sns:us-east-1:422430209855:komaltopic	Topic owner 422430209855
Type Standard	


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855

New Feature

Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN

arn:aws:sns:us-east-1:422430209855:komaltopic

Protocol

The type of endpoint to subscribe

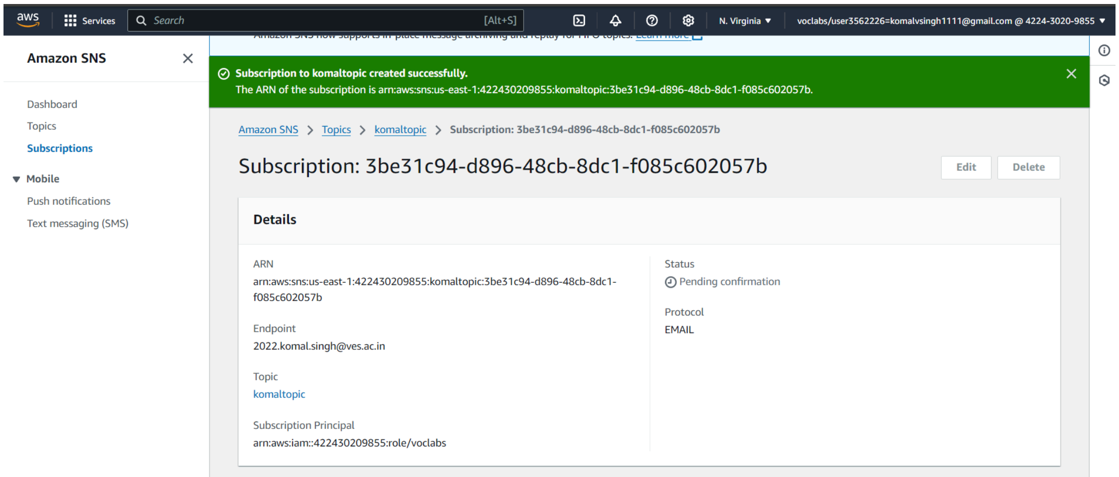
Email

Endpoint

An email address that can receive notifications from Amazon SNS.

2022.komal.singh@ves.ac.in

After your subscription is created, you must confirm it. Info


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855
Amazon SNS
Dashboard
Topics
Subscriptions
Mobile
Push notifications
Text messaging (SMS)
Amazon SNS > Topics > komaltopic > Subscription: 3be31c94-d896-48cb-8dc1-f085c602057b
Edit
Delete

Subscription to komaltopic created successfully.
The ARN of the subscription is arn:aws:sns:us-east-1:422430209855:komaltopic:3be31c94-d896-48cb-8dc1-f085c602057b.

Subscription: 3be31c94-d896-48cb-8dc1-f085c602057b

Details

ARN
arn:aws:sns:us-east-1:422430209855:komaltopic:3be31c94-d896-48cb-8dc1-f085c602057b

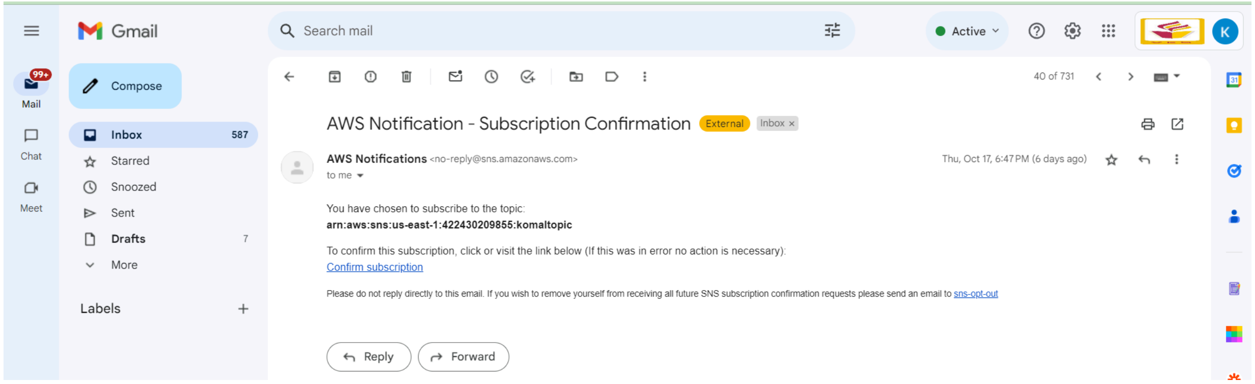
Endpoint
2022.komal.singh@ves.ac.in

Topic
komaltopic

Subscription Principal
arn:aws:iam::422430209855:role/voclabs

Status
Pending confirmation

Protocol
EMAIL



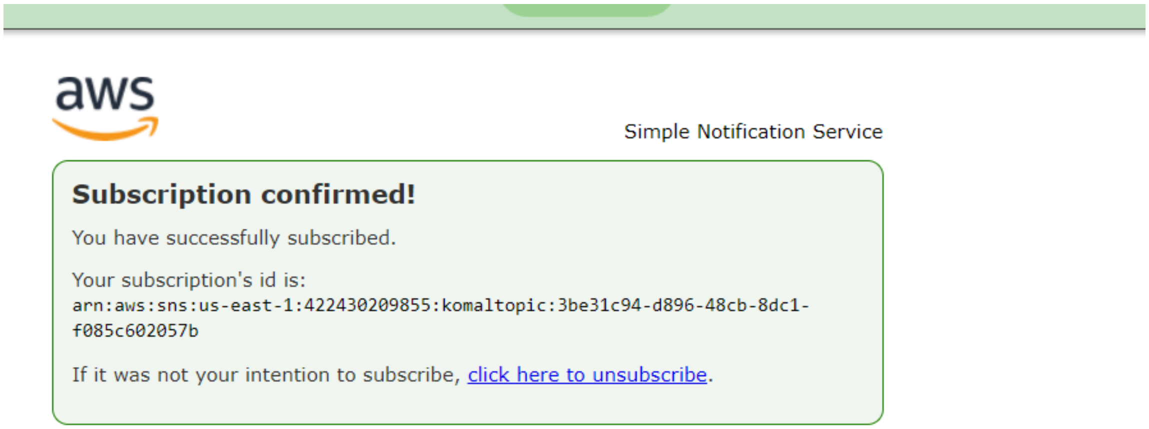
AWS Notification - Subscription Confirmation

AWS Notifications <no-reply@sns.amazonaws.com>
to me
Thu, Oct 17, 6:47 PM (6 days ago)

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:422430209855:komaltopic

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

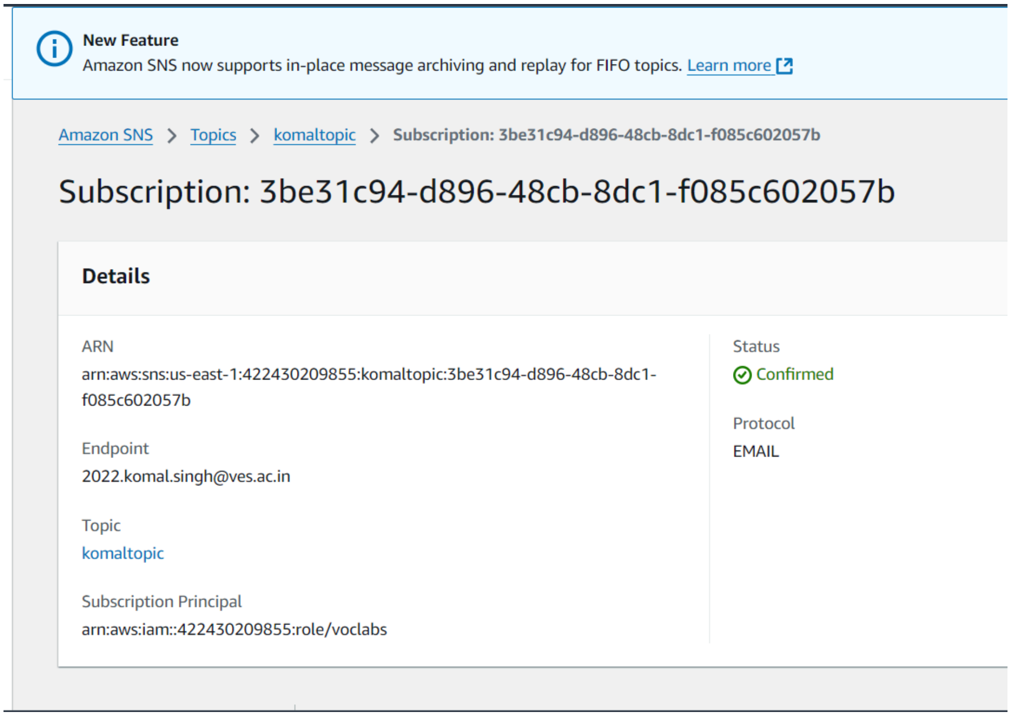


Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:us-east-1:422430209855:komaltopic:3be31c94-d896-48cb-8dc1-f085c602057b

If it was not your intention to subscribe, click here to unsubscribe.



New Feature
Amazon SNS now supports in-place message archiving and replay for FIFO topics. Learn more

Amazon SNS > Topics > komaltopic > Subscription: 3be31c94-d896-48cb-8dc1-f085c602057b

Subscription: 3be31c94-d896-48cb-8dc1-f085c602057b

Details

ARN
arn:aws:sns:us-east-1:422430209855:komaltopic:3be31c94-d896-48cb-8dc1-f085c602057b

Endpoint
2022.komal.singh@ves.ac.in

Topic
komaltopic

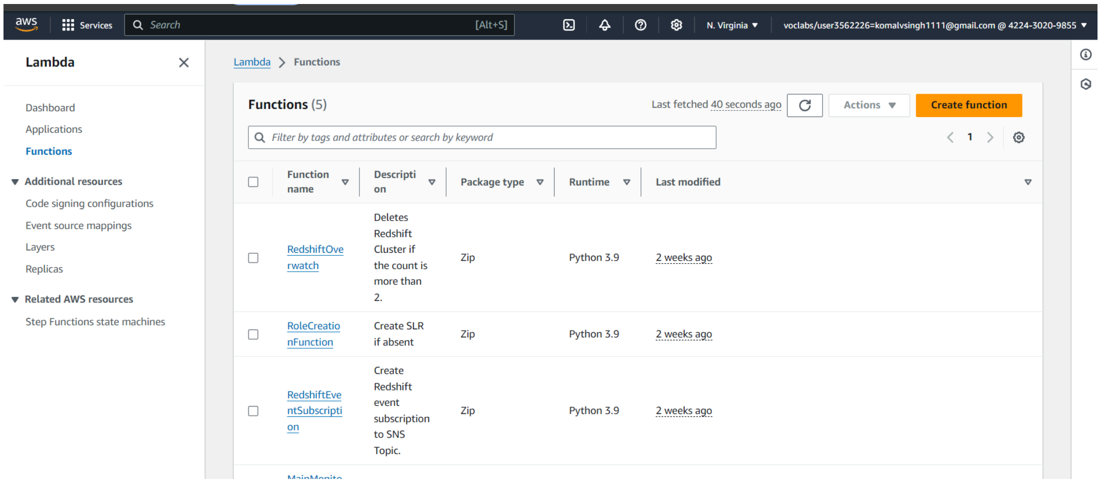
Subscription Principal
arn:aws:iam::422430209855:role/voclabs

Status
Confirmed

Protocol
EMAIL

Step 3: Create the Lambda Function

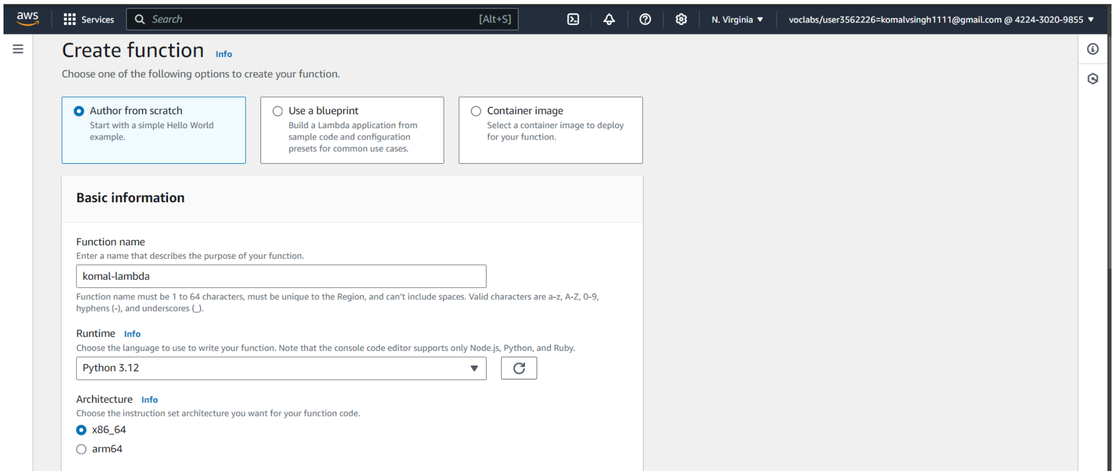
1. Go to **Lambda** from the Services menu.
2. Click **Create Function**.
   - Choose **Author from scratch**.
   - Name the function (e.g., `S3UploadNotificationLambda`).
   - Set the **runtime** to `Python 3.x`.
   - Choose or create an appropriate **execution role** that has access to S3 and SNS.
3. In the function editor, replace the default code with the following Python code:



Functions (5)

Last fetched 40 seconds ago

□	Function name	Description	Package type	Runtime	Last modified
□	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.9	2 weeks ago
□	RoleCreationFunction	Create SLR if absent	Zip	Python 3.9	2 weeks ago
□	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.9	2 weeks ago
	MainMonito				


aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▼

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.

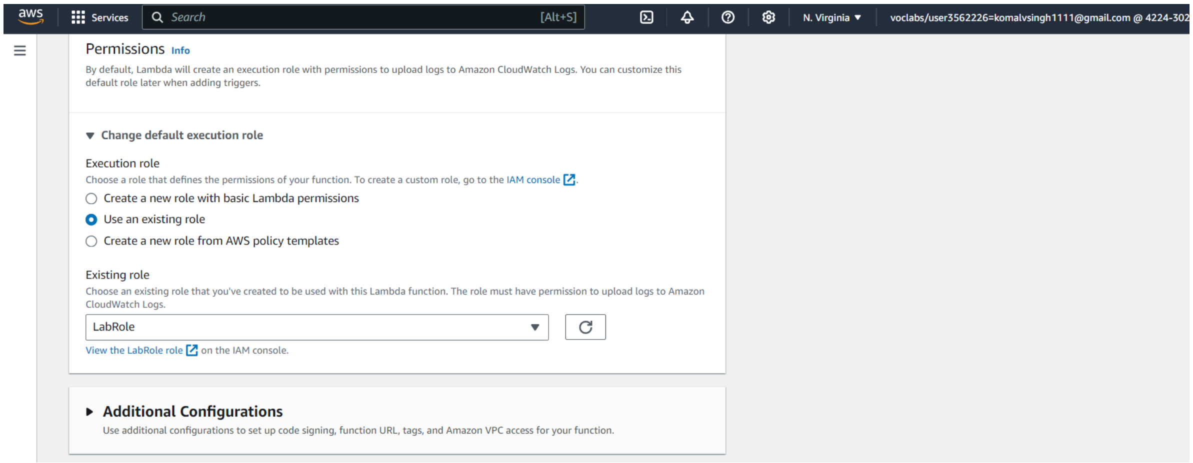
○ Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
komal-lambda
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.
Python 3.12

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64



Permissions Info

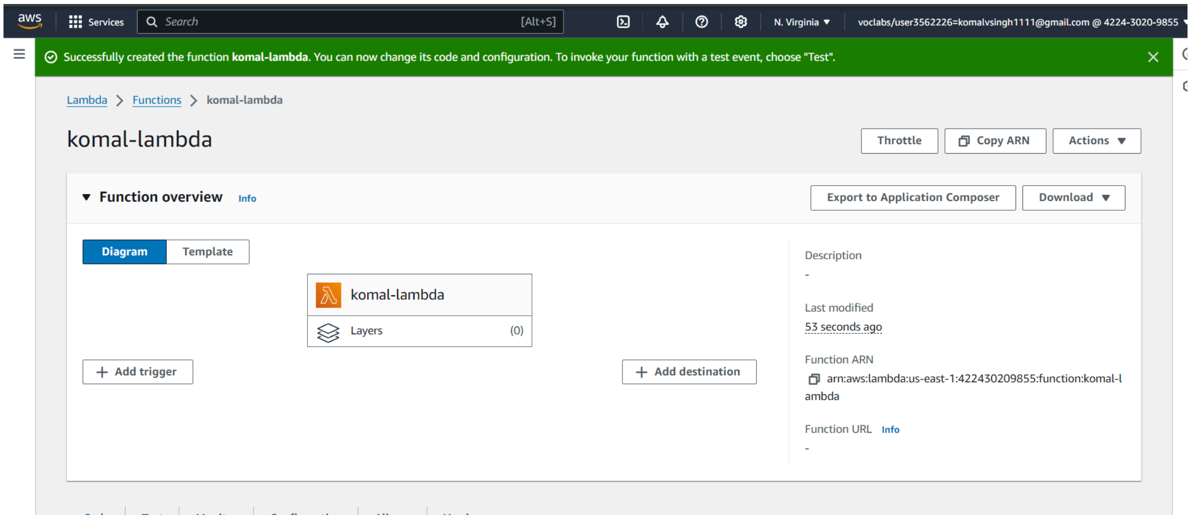
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
○ Create a new role with basic Lambda permissions
● 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.
LabRole
View the LabRole role on the IAM console.

▶ Additional Configurations
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.


aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▼

Successfully created the function **komal-lambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > komal-lambda

komal-lambda

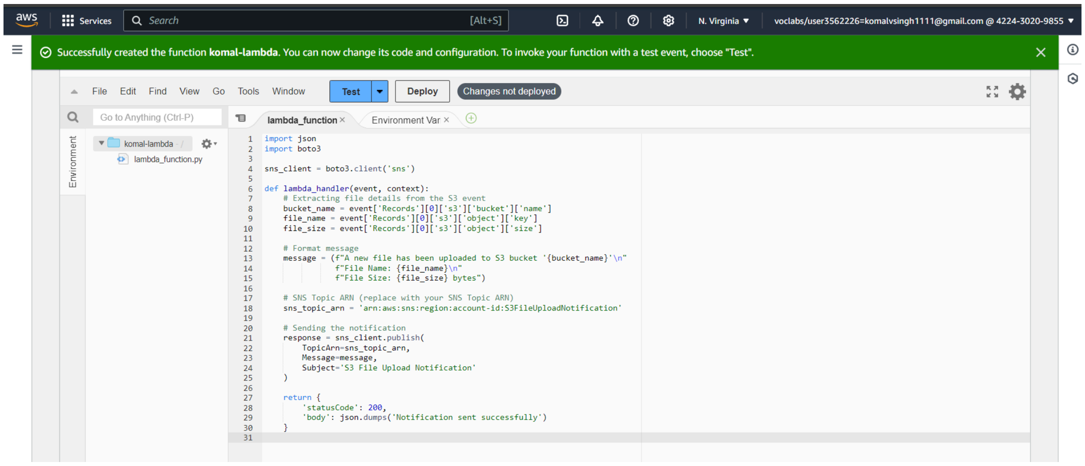
▼ Function overview Info

Description
-

Last modified
53 seconds ago

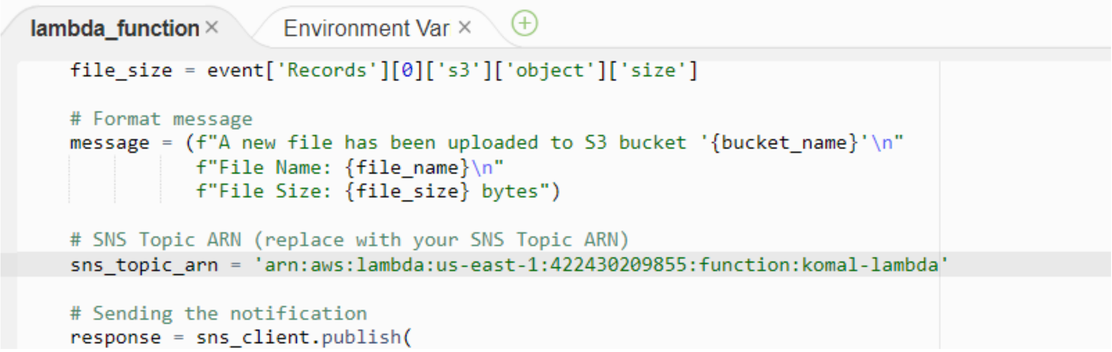
Function ARN
arn:aws:lambda:us-east-1:422430209855:function:komal-lambda

Function URL Info
-

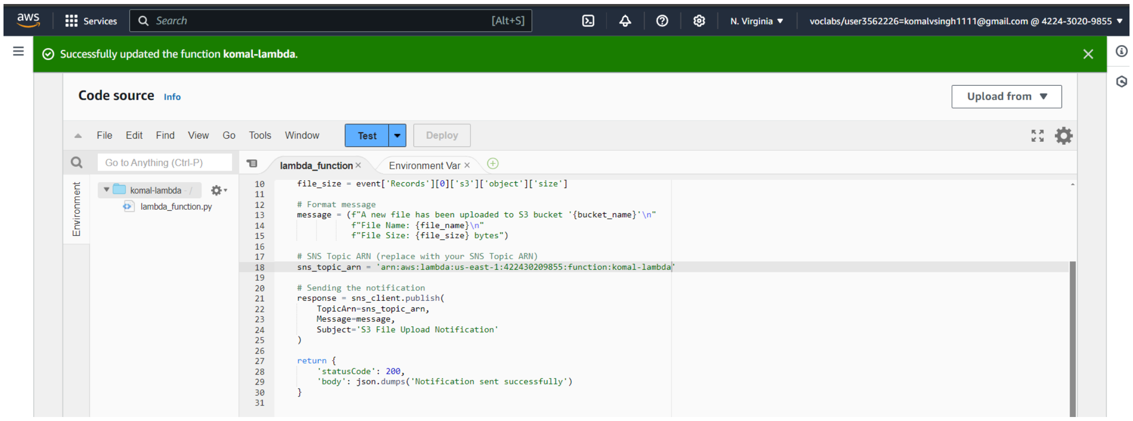


Successfully created the function **komal-lambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

```
import json
import boto3

sns_client = boto3.client('sns')

def lambda_handler(event, context):
    # Extracting file details from the S3 event
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    file_name = event['Records'][0]['s3']['object']['key']
    file_size = event['Records'][0]['s3']['object']['size']

    # Format message
    message = (f"A new file has been uploaded to S3 bucket '{bucket_name}'\n"
               f"File Name: {file_name}\n"
               f"File Size: {file_size} bytes")

    # SNS Topic ARN (replace with your SNS Topic ARN)
    sns_topic_arn = 'arn:aws:sns:region:account-id:S3FileUploadNotification'

    # Sending the notification
    response = sns_client.publish(
        TopicArn=sns_topic_arn,
        Message=message,
        Subject='S3 File Upload Notification'
    )

    return {
        'statusCode': 200,
        'body': json.dumps('Notification sent successfully')
    }

```



```
file_size = event['Records'][0]['s3']['object']['size']

# Format message
message = (f"A new file has been uploaded to S3 bucket '{bucket_name}'\n"
           f"File Name: {file_name}\n"
           f"File Size: {file_size} bytes")

# SNS Topic ARN (replace with your SNS Topic ARN)
sns_topic_arn = 'arn:aws:lambda:us-east-1:422430209855:function:komal-lambda'

# Sending the notification
response = sns_client.publish(
```



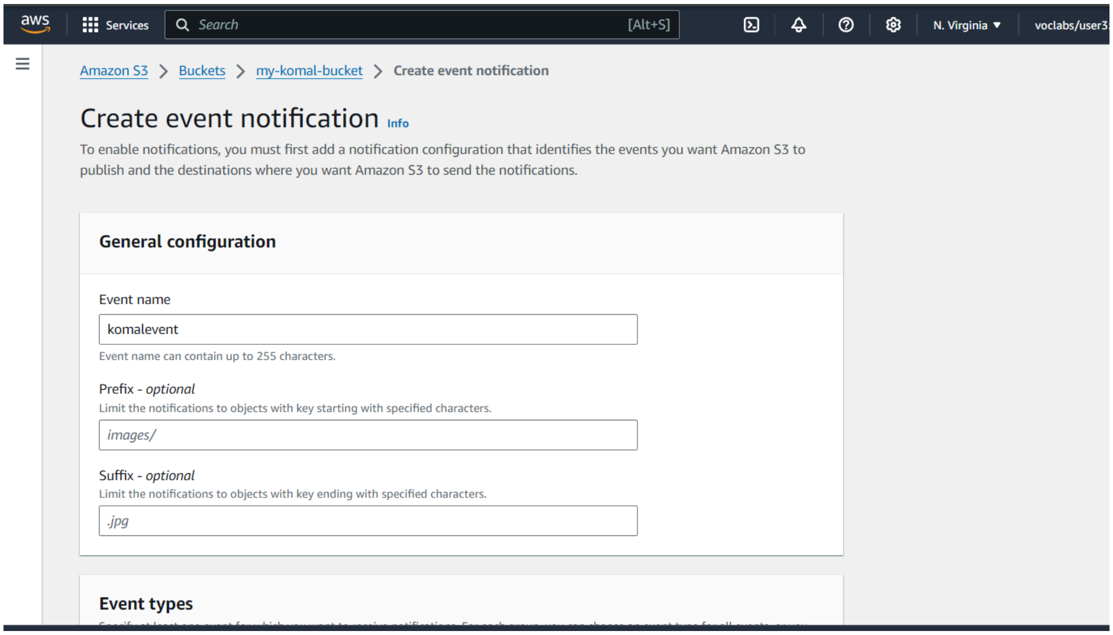
Successfully updated the function **komal-lambda**.

```
    file_size = event['Records'][0]['s3']['object']['size']

    # Format message
    message = (f"A new file has been uploaded to S3 bucket '{bucket_name}'\n"
               f"File Name: {file_name}\n"
               f"File Size: {file_size} bytes")

    # SNS Topic ARN (replace with your SNS Topic ARN)
    sns_topic_arn = 'arn:aws:lambda:us-east-1:422430209855:function:komal-lambda'

    # Sending the notification
    response = sns_client.publish(
        TopicArn=sns_topic_arn,
        Message=message,
        Subject='S3 File Upload Notification'
    )

    return {
        'statusCode': 200,
        'body': json.dumps('Notification sent successfully')
    }

```

Step 4: Configure S3 Trigger for Lambda

1. Go to the **S3 bucket** you created earlier.
2. Click on **Properties**.
3. Scroll down to **Event notifications** and click **Create event notification**.
   - Name the event (e.g., `S3UploadEvent`).
   - Under **Event types**, select `All object create events`.

- Under **Destination**, select **Lambda Function** and choose your Lambda function (`S3UploadNotificationLambda`).
- Click **Save**.



Amazon S3 > Buckets > my-komal-bucket > Create event notification

Create event notification Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

General configuration

Event name

komalevent

Event name can contain up to 255 characters.

Prefix - *optional*

Limit the notifications to objects with key starting with specified characters.

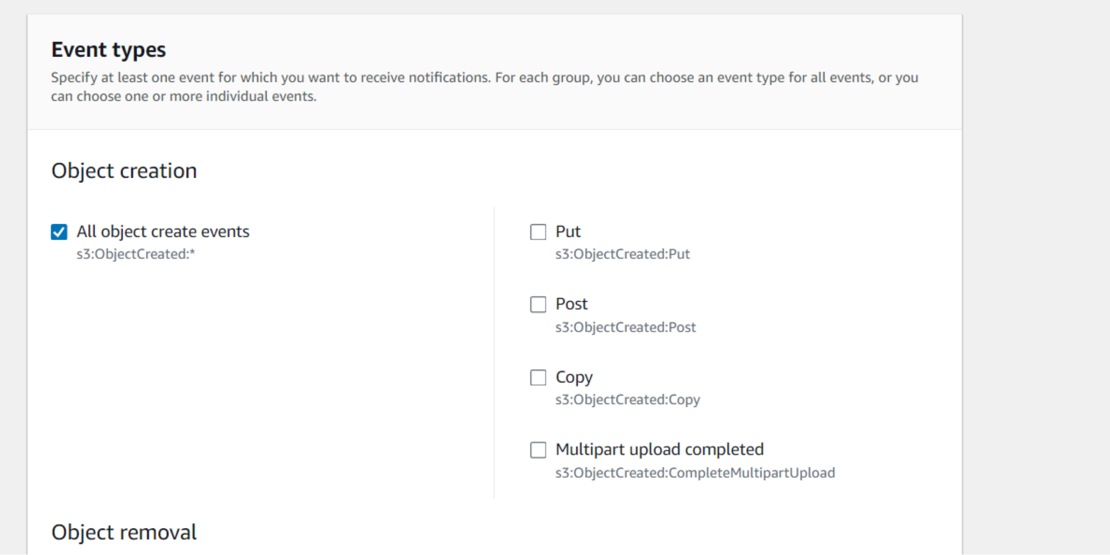
images/

Suffix - *optional*

Limit the notifications to objects with key ending with specified characters.

.jpg

Event types



Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

☑ All object create events
s3:ObjectCreated:*

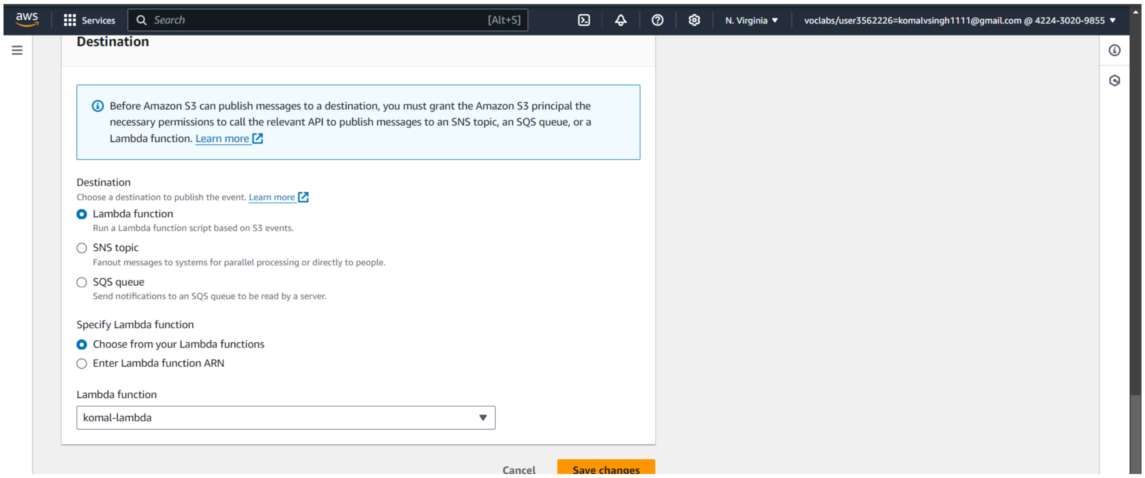
□ Put
s3:ObjectCreated:Put

□ Post
s3:ObjectCreated:Post

□ Copy
s3:ObjectCreated:Copy

□ Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload

Object removal


aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▼
Cancel
Save changes

Destination

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. Learn more

Destination

Choose a destination to publish the event. Learn more

● Lambda function
Run a Lambda function script based on S3 events.

○ SNS topic
Fanout messages to systems for parallel processing or directly to people.

○ SQS queue
Send notifications to an SQS queue to be read by a server.

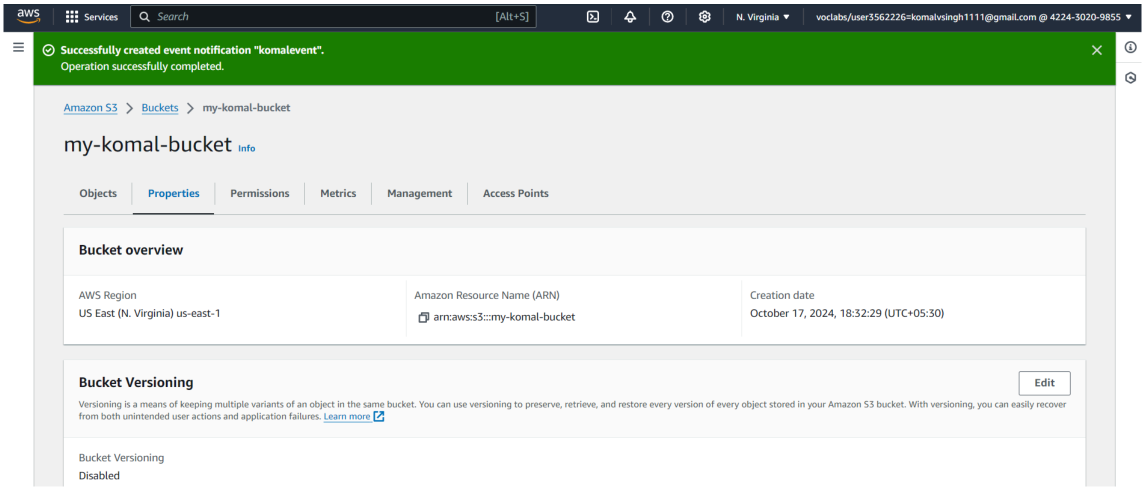
Specify Lambda function

● Choose from your Lambda functions

○ Enter Lambda function ARN

Lambda function

komal-lambda


aws
Services
Search
[Alt+S]
N. Virginia ▼
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▼
Amazon S3 > Buckets > my-komal-bucket
Objects
Properties
Permissions
Metrics
Management
Access Points
Edit

Successfully created event notification "komalevent".
Operation successfully completed.

my-komal-bucket Info

Bucket overview

AWS Region
US East (N. Virginia) us-east-1

Amazon Resource Name (ARN)
arn:aws:s3:::my-komal-bucket

Creation date
October 17, 2024, 18:32:29 (UTC+05:30)

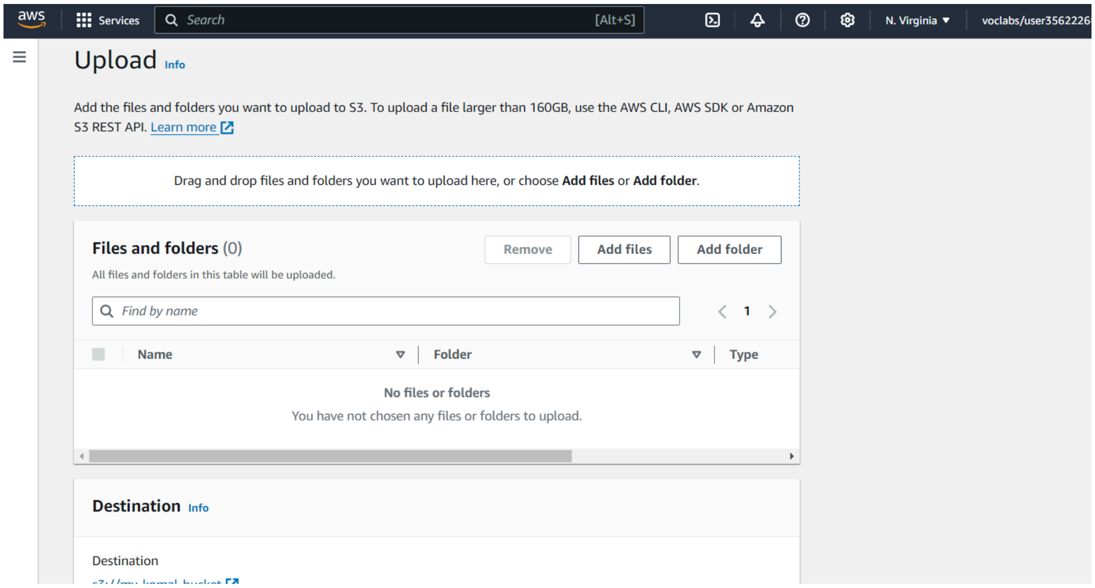
Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Disabled

Step 5: Test the Setup

1. Upload a file to your S3 bucket.
   - Go to the **S3 bucket**.
   - Click **Upload** and choose a file.
2. Wait for the Lambda function to trigger.
3. Check the configured email for the notification with the uploaded file's details.



Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (0)

All files and folders in this table will be uploaded.

□	Name	Folder	Type

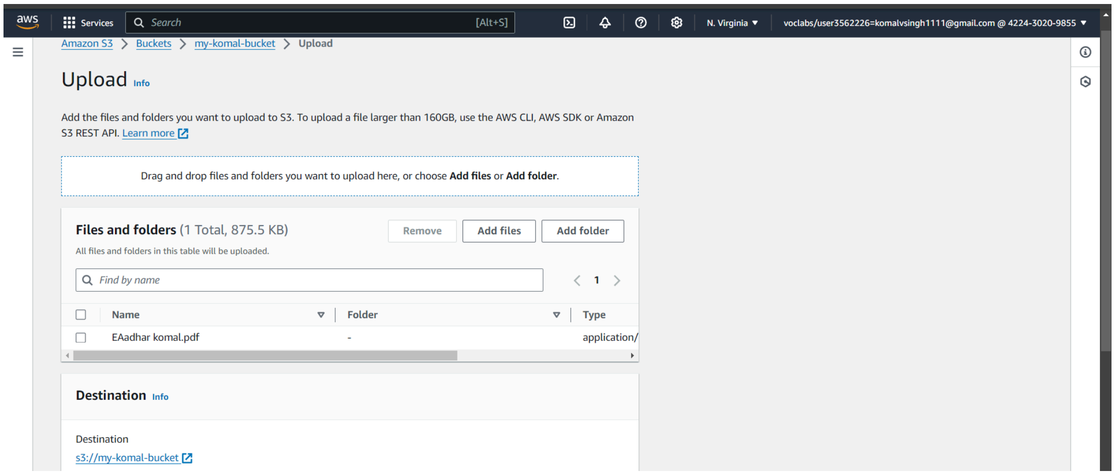
No files or folders

You have not chosen any files or folders to upload.

Destination Info

Destination

s3://my-komal-bucket


aws
Services
Search
[Alt+S]
N. Virginia ▾
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▾
Amazon S3 > Buckets > my-komal-bucket > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 875.5 KB)

Remove
Add files
Add folder

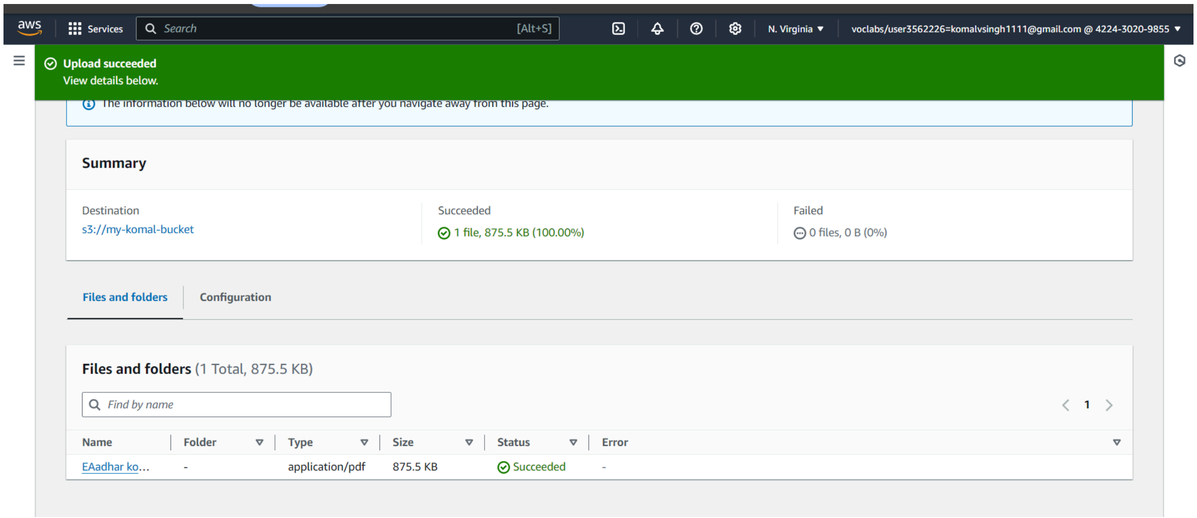
All files and folders in this table will be uploaded.

□	Name	Folder	Type
□	EAadhar komal.pdf	-	application/

Destination Info

Destination

s3://my-komal-bucket


aws
Services
Search
[Alt+S]
N. Virginia ▾
voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855 ▾

Upload succeeded
View details below.

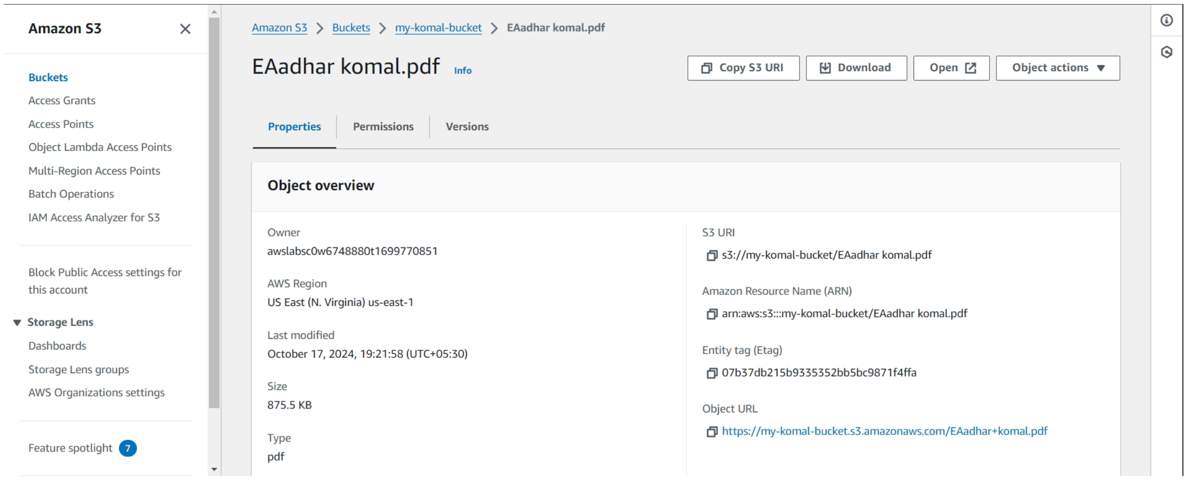
The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://my-komal-bucket	1 file, 875.5 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 875.5 KB)

Name	Folder	Type	Size	Status	Error
EAadhar ko...	-	application/pdf	875.5 KB	Succeeded	-



Amazon S3

Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings

Feature spotlight 7

Amazon S3 > Buckets > my-komal-bucket > EAadhar komal.pdf

EAadhar komal.pdf Info

Copy S3 URI | Download | Open | Object actions

Properties | Permissions | Versions

Object overview

Owner
awslabsc0w6748880t1699770851

AWS Region
US East (N. Virginia) us-east-1

Last modified
October 17, 2024, 19:21:58 (UTC+05:30)

Size
875.5 KB

Type
pdf

S3 URI
s3://my-komal-bucket/EAadhar komal.pdf

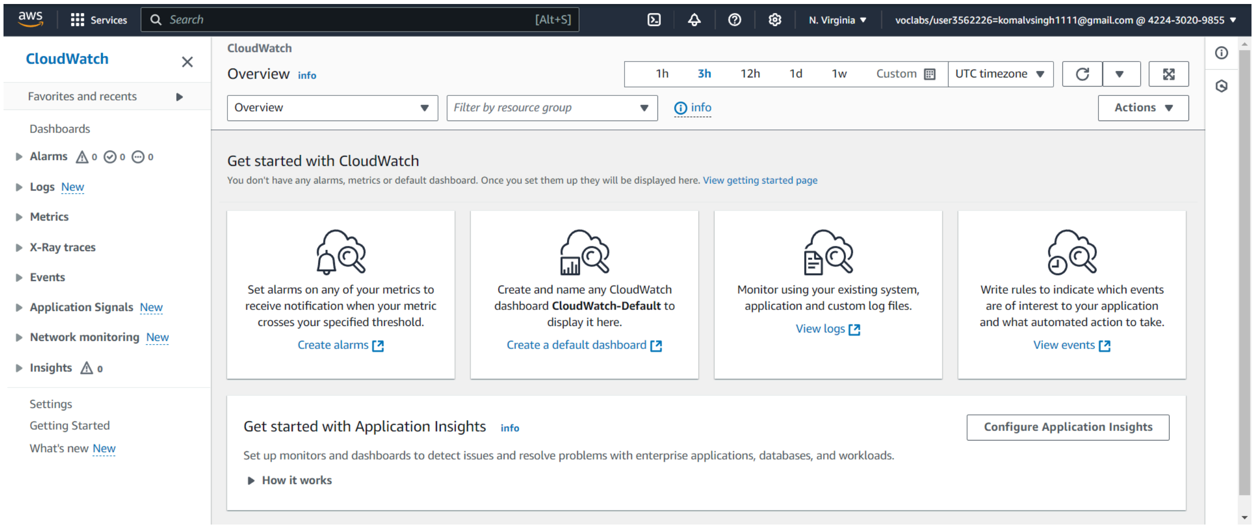
Amazon Resource Name (ARN)
arn:aws:s3:::my-komal-bucket/EAadhar komal.pdf

Entity tag (Etag)
07b37db215b9335352bb5bc9871f4ffa

Object URL
https://my-komal-bucket.s3.amazonaws.com/EAadhar+komal.pdf

Step 6: Monitor Logs

1. Go to **CloudWatch Logs** to verify Lambda execution:
   - Navigate to **Log groups**.
   - Find the log group for your Lambda function.
   - Check for any issues or confirm that the Lambda executed successfully.



Services | Search [Alt+S] | N. Virginia | voclabs/user3562226=komalvsingh1111@gmail.com @ 4224-3020-9855

CloudWatch

Favorites and recents
Dashboards
Alarms ⚠ 0 ✓ 0 ⊖ 0
Logs New
Metrics
X-Ray traces
Events
Application Signals New
Network monitoring New
Insights ⚠ 0
Settings
Getting Started
What's new New

CloudWatch

Overview info

1h | 3h | 12h | 1d | 1w | Custom | UTC timezone

Overview | Filter by resource group | info | Actions

Get started with CloudWatch

You don't have any alarms, metrics or default dashboard. Once you set them up they will be displayed here. View getting started page

Set alarms on any of your metrics to receive notification when your metric crosses your specified threshold.
Create alarms

Create and name any CloudWatch dashboard **CloudWatch-Default** to display it here.
Create a default dashboard

Monitor using your existing system, application and custom log files.
View logs

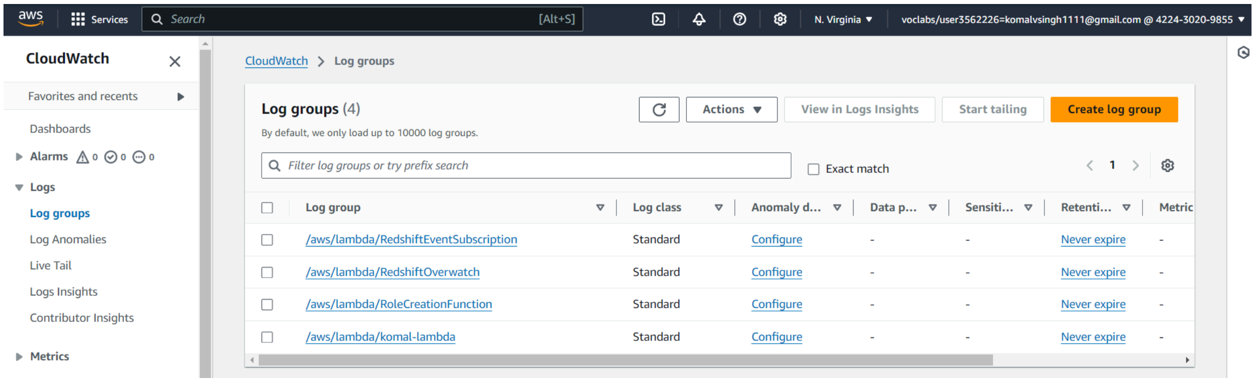
Write rules to indicate which events are of interest to your application and what automated action to take.
View events

Get started with Application Insights info

Configure Application Insights

Set up monitors and dashboards to detect issues and resolve problems with enterprise applications, databases, and workloads.

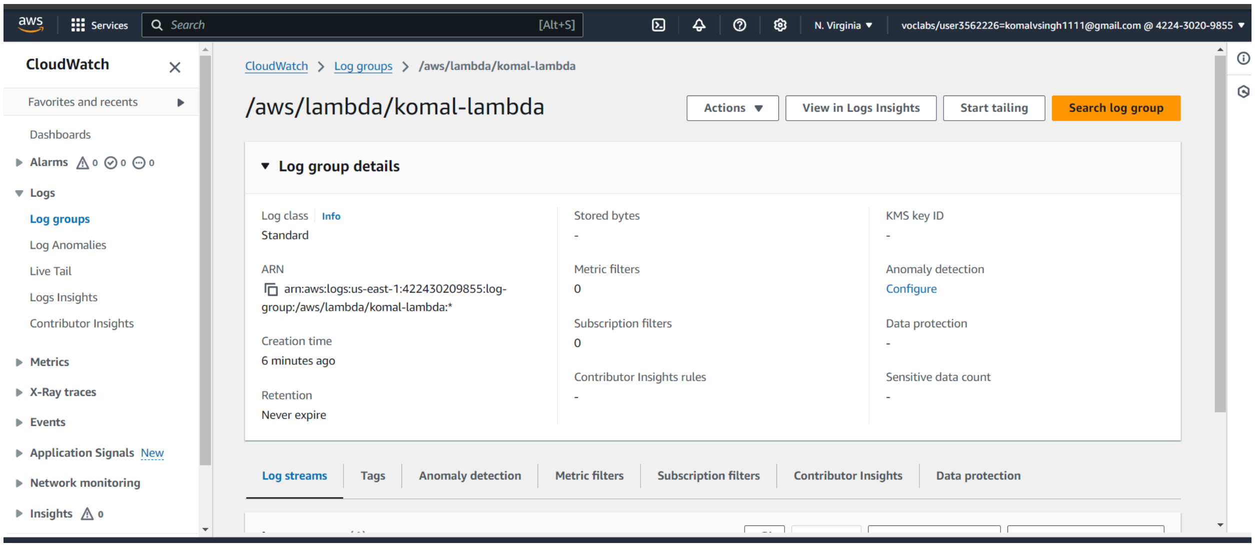
How it works



Log groups (4)

By default, we only load up to 10000 log groups.

	Log group	Log class	Anomaly d...	Data p...	Sensiti...	Retenti...	Metric
□	/aws/lambda/RedshiftEventSubscription	Standard	Configure	-	-	Never expire	-
□	/aws/lambda/RedshiftOverwatch	Standard	Configure	-	-	Never expire	-
□	/aws/lambda/RoleCreationFunction	Standard	Configure	-	-	Never expire	-
□	/aws/lambda/komal-lambda	Standard	Configure	-	-	Never expire	-



/aws/lambda/komal-lambda

Log group details

Log class Info
Standard

ARN
arn:aws:logs:us-east-1:422430209855:log-group:/aws/lambda/komal-lambda:*

Creation time
6 minutes ago

Retention
Never expire

Stored bytes
-

Metric filters
0

Subscription filters
0

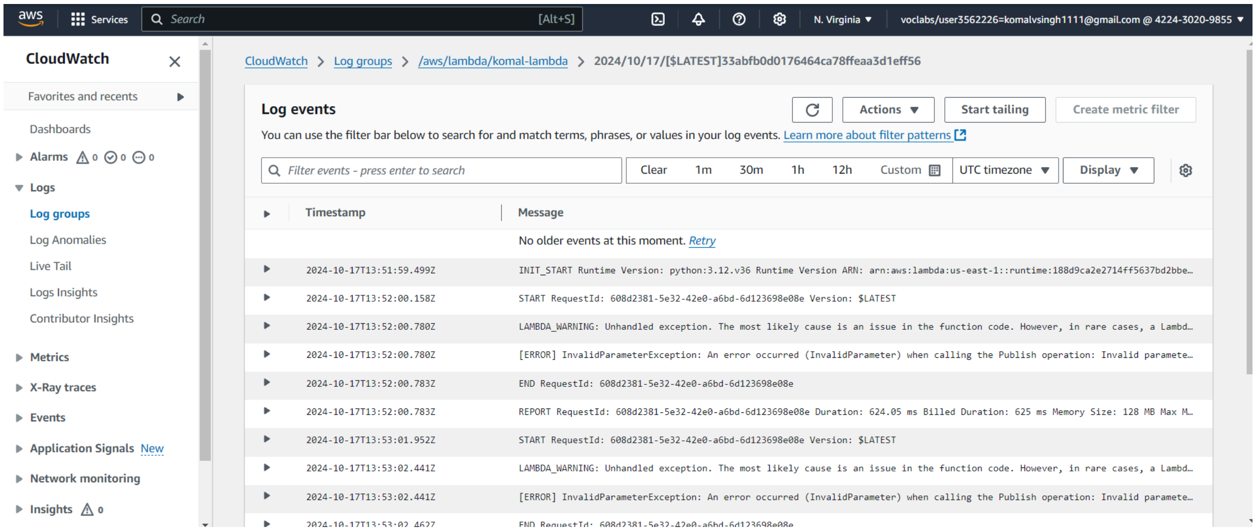
Contributor Insights rules
-

KMS key ID
-

Anomaly detection
Configure

Data protection
-

Sensitive data count
-



Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                 Message
                          No older events at this moment. Retry
2024-10-17T13:51:59.499Z  INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:188d9ca2e2714ff5637bd2bbe…
2024-10-17T13:52:00.158Z  START RequestId: 608d2381-5e32-42e0-a6bd-6d123698e08e Version: $LATEST
2024-10-17T13:52:00.780Z  LAMBDA_WARNING: Unhandled exception. The most likely cause is an issue in the function code. However, in rare cases, a Lambd…
2024-10-17T13:52:00.780Z  [ERROR] InvalidParameterException: An error occurred (InvalidParameter) when calling the Publish operation: Invalid paramete…
2024-10-17T13:52:00.783Z  END RequestId: 608d2381-5e32-42e0-a6bd-6d123698e08e
2024-10-17T13:52:00.783Z  REPORT RequestId: 608d2381-5e32-42e0-a6bd-6d123698e08e Duration: 624.05 ms Billed Duration: 625 ms Memory Size: 128 MB Max M…
2024-10-17T13:53:01.952Z  START RequestId: 608d2381-5e32-42e0-a6bd-6d123698e08e Version: $LATEST
2024-10-17T13:53:02.441Z  LAMBDA_WARNING: Unhandled exception. The most likely cause is an issue in the function code. However, in rare cases, a Lambd…
2024-10-17T13:53:02.441Z  [ERROR] InvalidParameterException: An error occurred (InvalidParameter) when calling the Publish operation: Invalid paramete…
2024-10-17T13:53:02.462Z  END RequestId: 608d2381-5e32-42e0-a6bd-6d123698e08e
```

Conclusion:

The integration of AWS Lambda, S3, and SNS provides an efficient, scalable solution for automated file upload notifications. By setting up this system, users receive real-time alerts whenever a file is uploaded to the S3 bucket. This process improves operational awareness, facilitates prompt responses to data uploads, and reduces the need for manual monitoring. The solution is highly customizable, allowing easy adaptation for various use cases such as alerting for critical file uploads, workflow automation, or data pipeline monitoring. Moreover, the serverless nature of Lambda and the robustness of SNS ensure cost-effectiveness, reliability, and scalability without the need for ongoing management of infrastructure.