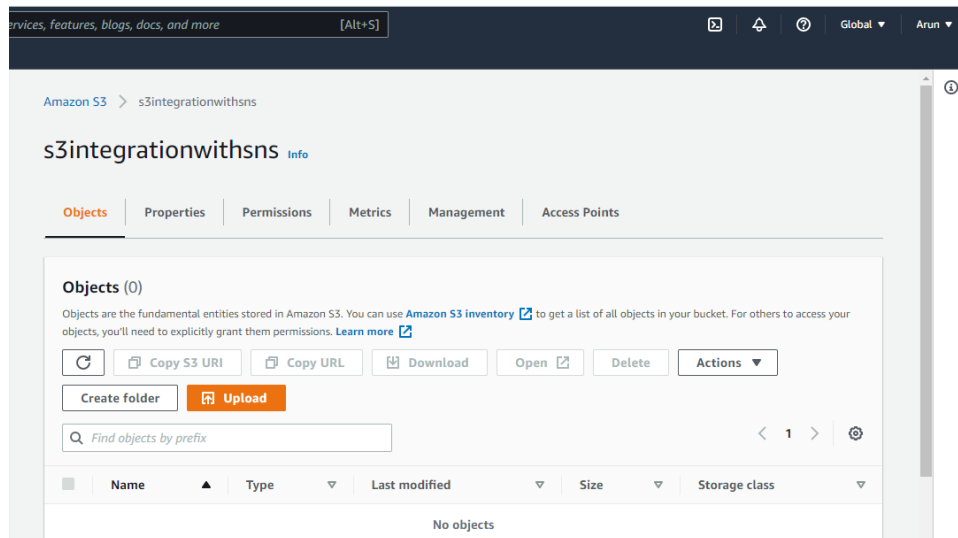


S3 integration with SNS

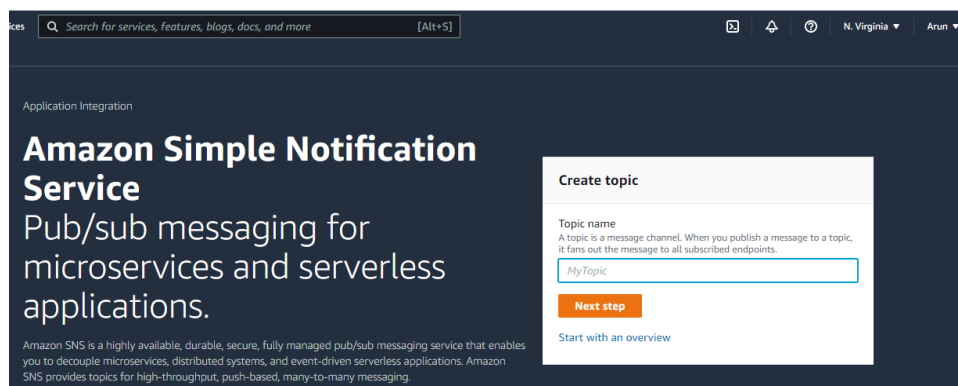
Overview

1. Create s3 bucket
2. Create an Event
3. Create Subscription
4. Update the topic Policy
5. Add event notification rule in s3

Step 1 : Create a s3 bucket



Step 2 : Goto SNS dashboard in management console >> Create topic



Step 3 : Select type as standard , click create topic

Services [Alt+S] N. Virginia Arun

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

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

Display name - optional
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)

Maximum 100 characters.

Step 4 : create subscription in the topic

Search for services, features, blogs, docs, and more [Alt+S] N. Virginia Arun

arn:aws:sns:us-east-1:986041421187:s3integrationwithsns

Type
Standard

Subscriptions | Access policy | Delivery retry policy (HTTP/S) | Delivery status logging | Encryption | Tags

Subscriptions (0) [Edit](#) [Delete](#) [Request confirmation](#) [Confirm subscription](#) [Create subscription](#)

< 1 > [Settings](#)

ID	Endpoint	Status	Protocol
No subscriptions found You don't have any subscriptions to this topic.			

[Create subscription](#)

Step 5 : Select the protocol based on the requirement and provide emailid as endpoint click create subscription

Services [Alt+S] N. Virginia Arun

Topic ARN

Protocol
The type of endpoint to subscribe

Select protocol

- Amazon Kinesis Data Firehose
- Amazon SQS
- AWS Lambda
- Email**
- Email-JSON
- HTTP
- HTTPS
- Platform application endpoint
- SMS

Send undeliverable messages to a dead-letter queue. [Info](#)

[Cancel](#) [Create subscription](#)

Step 6 : confirm email id



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-east-1:986041421187:s3integrationwithsns:bc80347c-572c-4989-89d3-fadcb3d7aa40

If it was not your intention to subscribe, [click here to unsubscribe](#).

Step 7 : goto Topic Access Policy update policy with below json code

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-1:986041421187:s3integrationwithsns",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:::s3integrationwithsns"
        }
      }
    }
  ]
}
```

Step 8 : Go to s3 bucket >> Properties >> event notification >> create event notification >> Event name >> select desired event types >> select destination as SNS and select topic >> click save changes

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

Event name can contain up to 255 characters.

Prefix - *optional*
Limit the notifications to objects with key starting with specified characters.

Suffix - *optional*
Limit the notifications to objects with key ending with specified characters.

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

☐ All object removal events
s3:ObjectRemoved:*

☐ Permanently deleted
s3:ObjectRemoved:Delete

☐ Delete marker created
s3:ObjectRemoved:DeleteMarkerCreated

Object restore

☐ All restore object events
s3:ObjectRestore:*

☐ Restore initiated
s3:ObjectRestore:Post

services, features, blogs, docs, and more [\[Alt+S\]](#) Global Arun

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
Send notifications to email, SMS, or an HTTP endpoint.

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

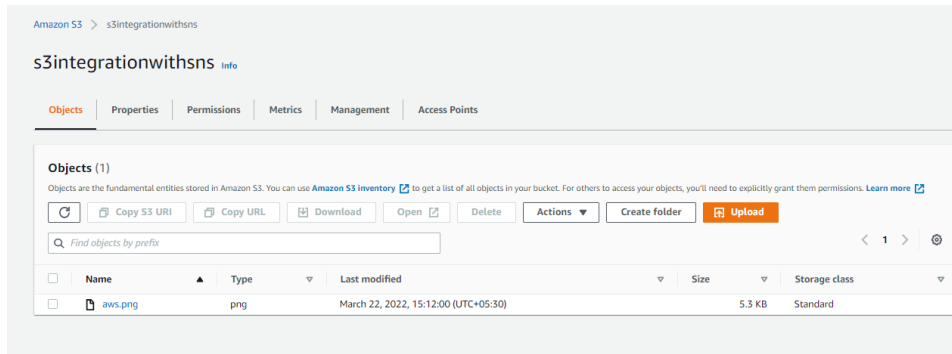
Specify SNS topic

☒ Choose from your SNS topics

☐ Enter SNS topic ARN

Cancel [Save changes](#)

Step 9 : Upload a object in bucket



Step 10 : check the mail inbox >> there is notification about object addition

