

CLOUD SECURITY & MANAGEMENT
LAB

Name: Harsh Ranjan

SAP ID: 500097019

Roll no: R2142211262

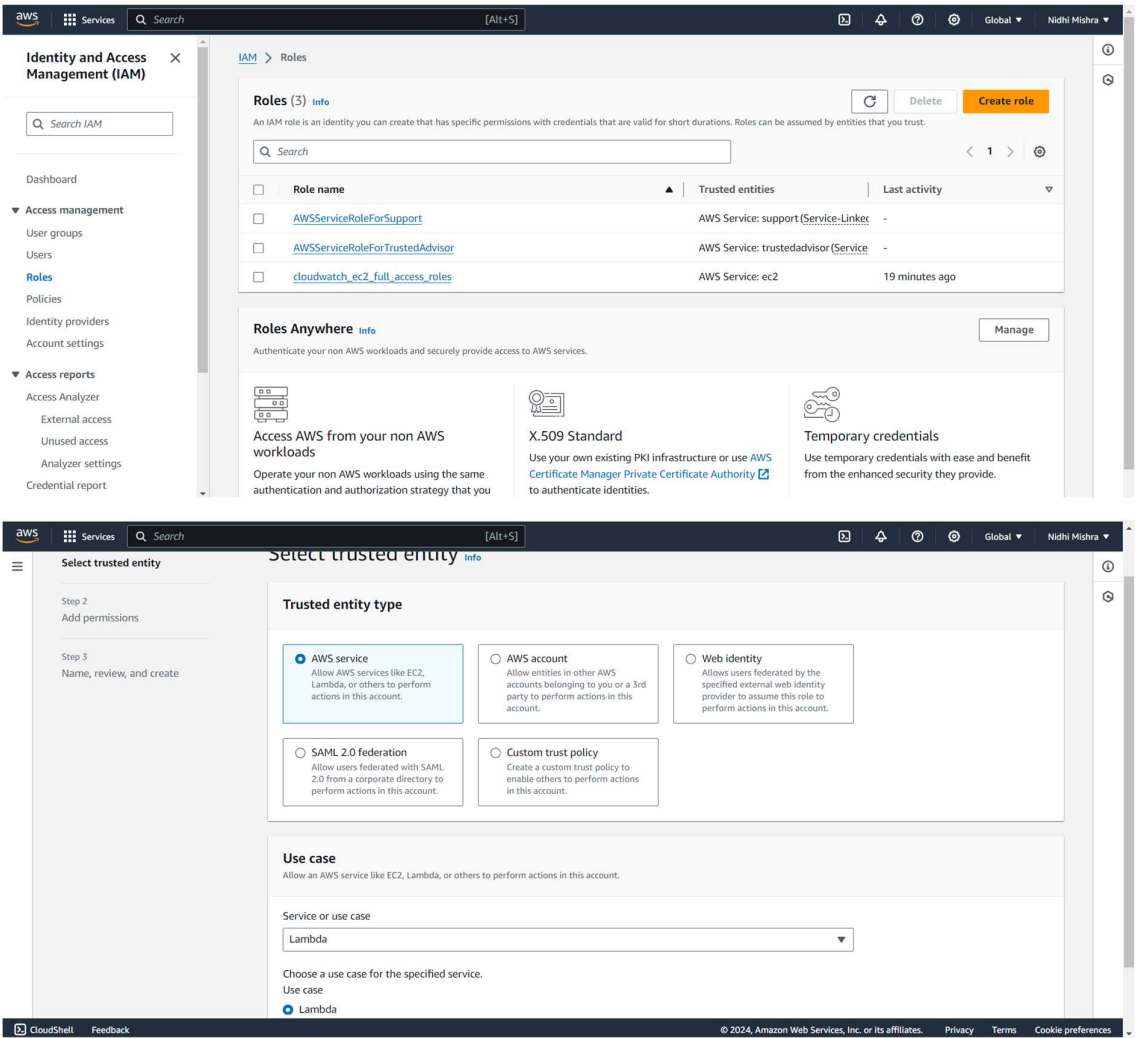
Batch :B7

SUBMISSION TO:

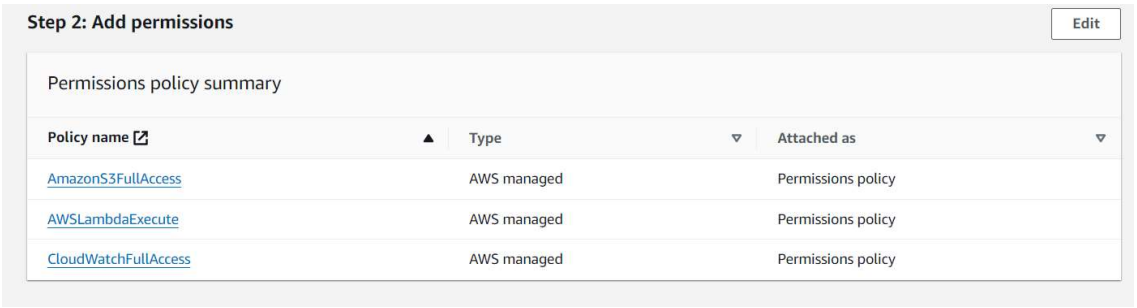
Ms. Avita Katal

Experiment 6 b)-Sending the cloudwatch logs to S3 with the help of AWS Lambda.

STEP 1: Firstly, go to IAM roles and create a role for lambda.



STEP 2: Attach these policies to that role and rename it.



aws Services Search [Alt+S] Global Nidhi Mishra

IAM > Roles > Create role

Step 1
[Select trusted entity](#)

Step 2
[Add permissions](#)

Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and "+,=,@,_" characters.

Description
Add a short explanation for this role.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: "+,=,@,/_[]!#\$%^&*()~<=>".

Step 1: Select trusted entities

Trust policy

```
1 {
2   "Version": "2012-10-17",
```

Edit

aws Services Search [Alt+S] Global Nidhi Mishra

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Role copy_logs_to_s3 created. View role

IAM > Roles

Roles (4) info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linker)	-
<input type="checkbox"/>	cloudwatch_ec2_full_access_roles	AWS Service: ec2	26 minutes ago
<input type="checkbox"/>	copy_logs_to_s3	AWS Service: lambda	-

STEP 3: Create a bucket in which we are going to store the log files.

aws Services Search [Alt+S] N. Virginia Nidhi Mishra

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 - New**
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

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.

Format: s3://bucket/prefix

Object Ownership info

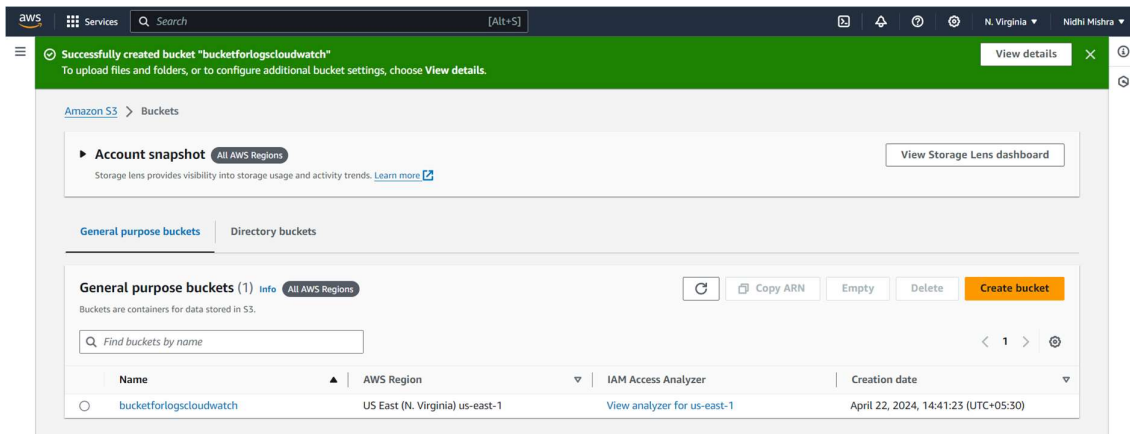
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**

☐ ACLs enabled

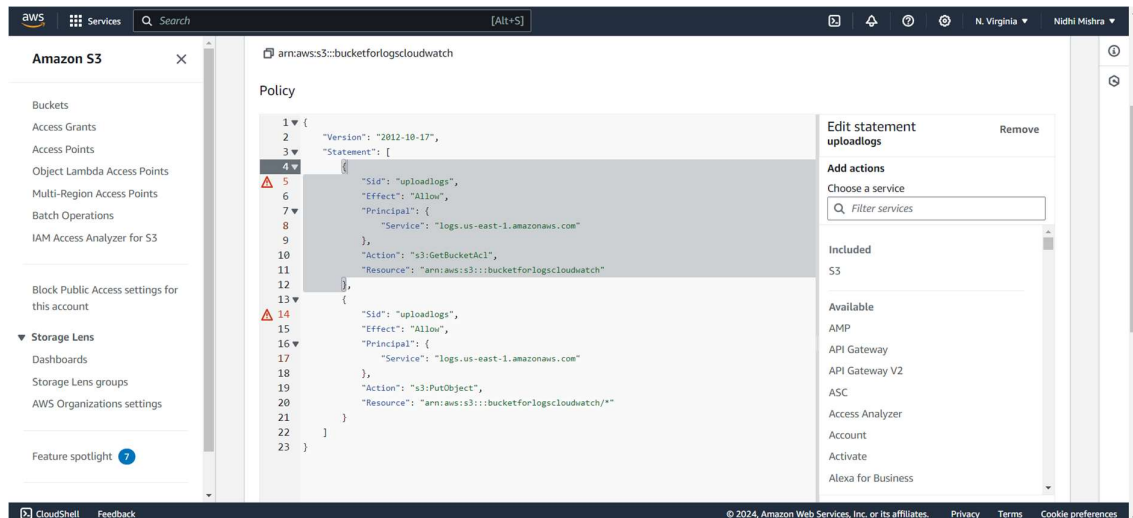
CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

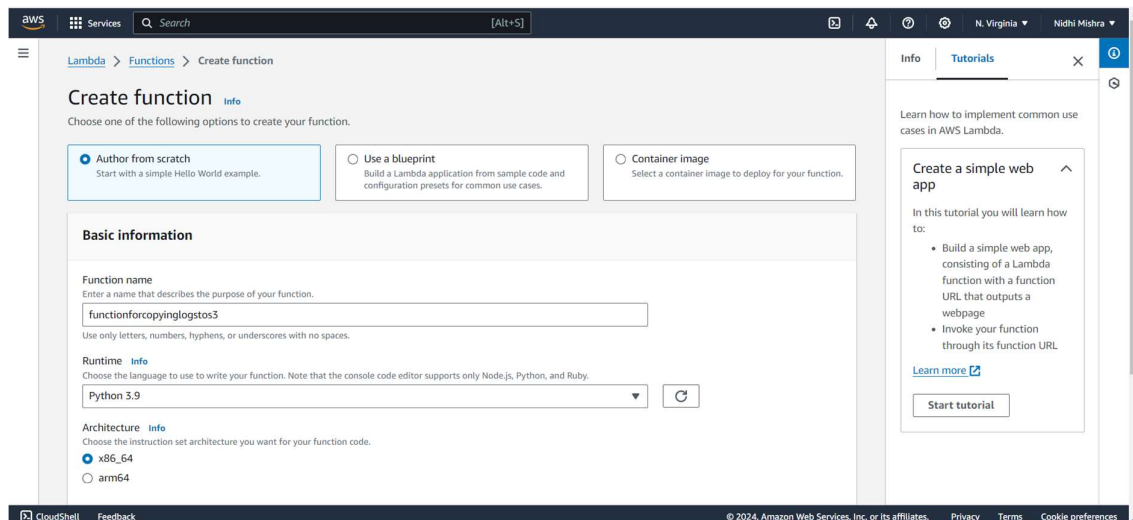
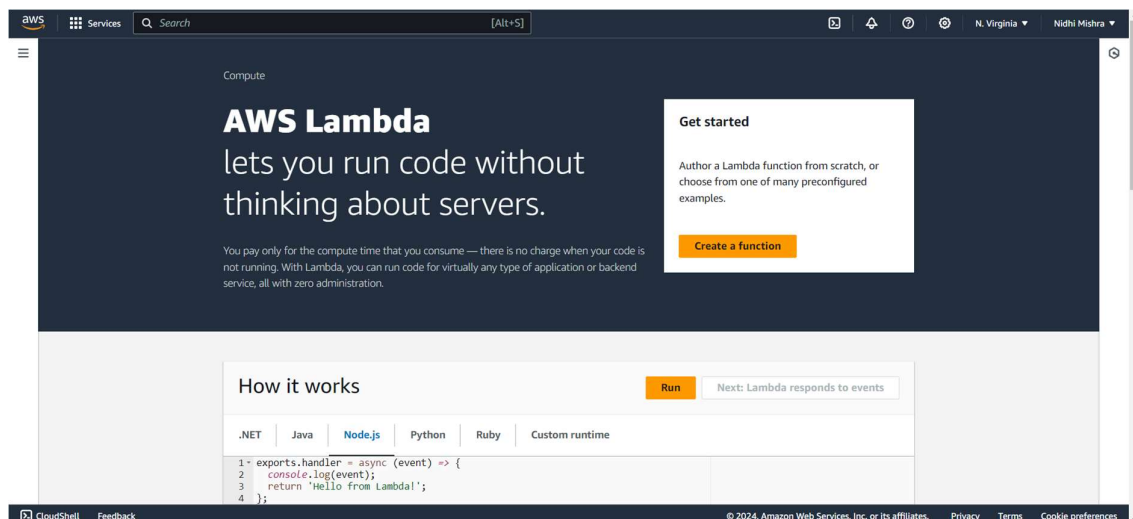


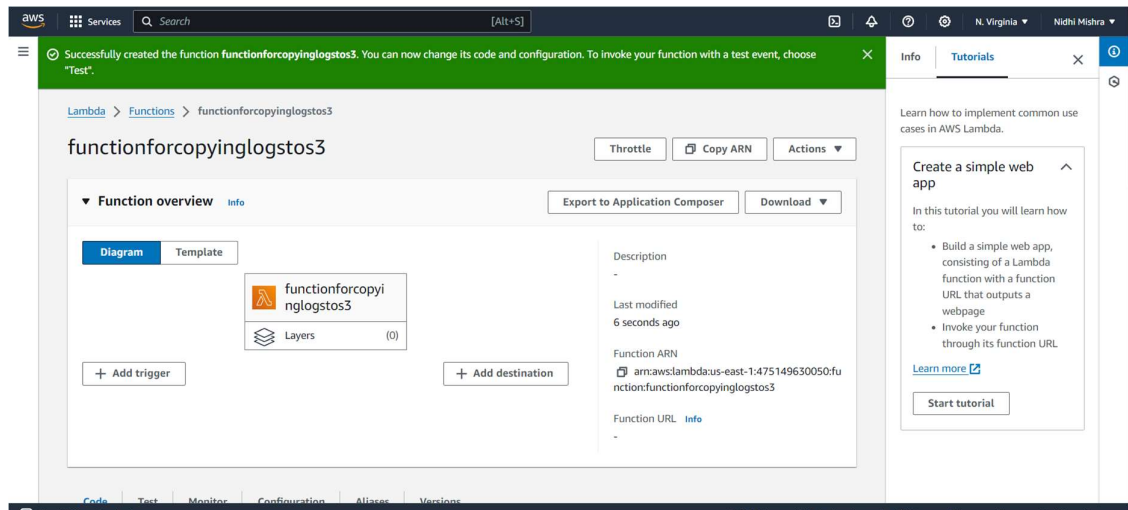
STEP 4: Edit the bucket policy as given:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "uploadlogs",
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.us-east-1.amazonaws.com"
      },
      "Action": "s3:GetBucketAcl",
      "Resource": "arn:aws:s3:::bucketforlogscloudwatch"
    },
    {
      "Sid": "uploadlogs",
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.us-east-1.amazonaws.com"
```

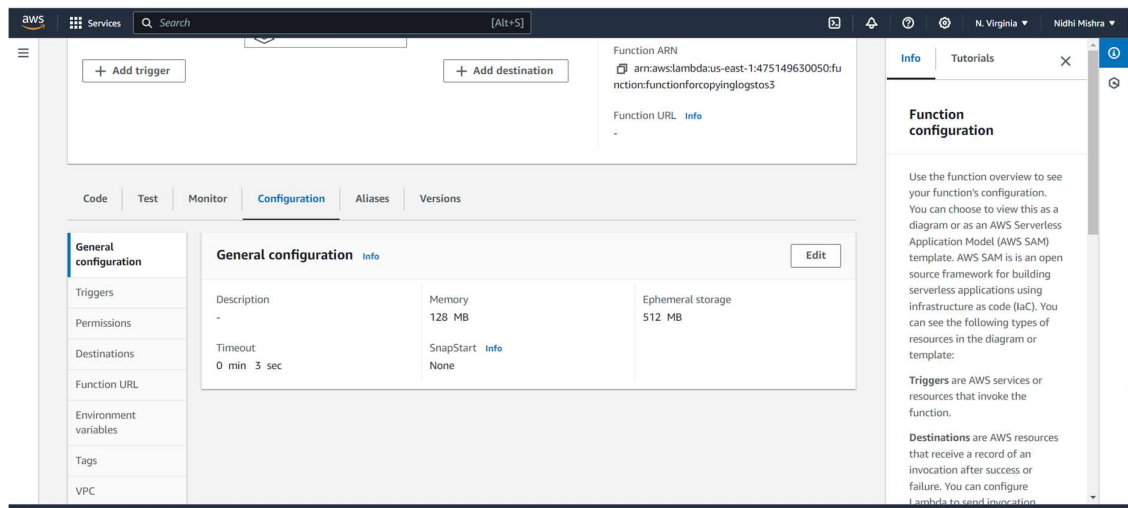


STEP 5: Now, go to AWS Lambda tab and create a lambda function.





STEP 6: Now go to configuration tab in Lambda function that we have created and attach the IAM role to it.



STEP 7: Now go to code tab and run the script to copy log files to s3 as shown below.

```
import boto3

import os

import datetime

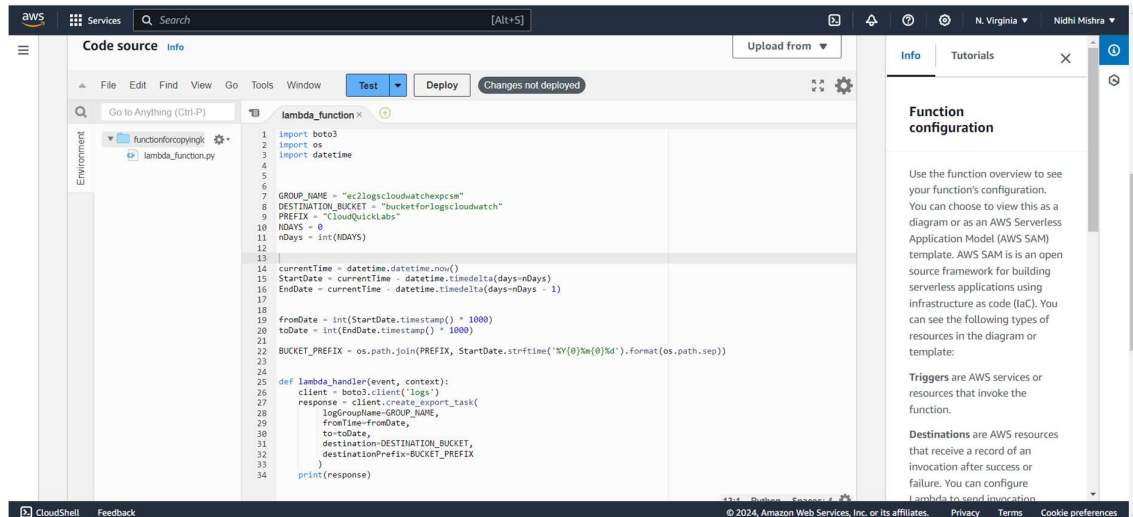

GROUP_NAME = "ec2logscloudwatchexpcsm"
DESTINATION_BUCKET = "bucketforlogscloudwatch"
PREFIX = "CloudQuickLabs"
NDAYS = 0
nDays = int(NDAYS)


currentTime = datetime.datetime.now()
StartDate = currentTime - datetime.timedelta(days=nDays)
EndDate = currentTime - datetime.timedelta(days=nDays - 1)

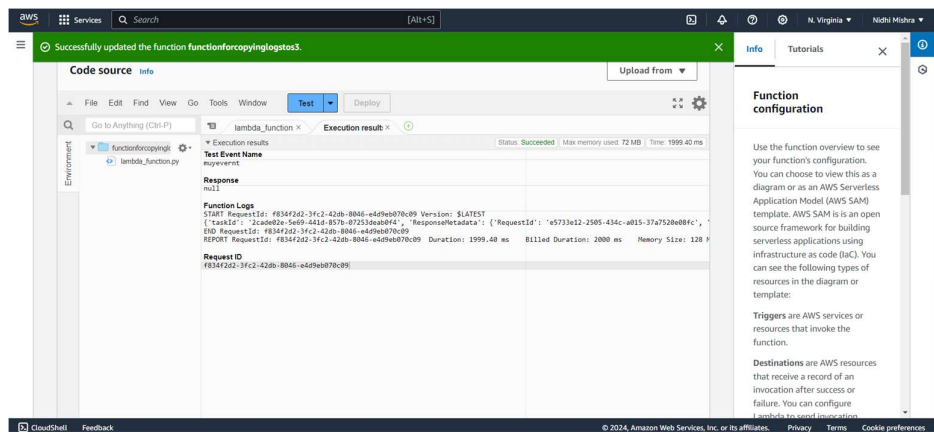

fromDate = int(StartDate.timestamp() * 1000)
toDate = int(EndDate.timestamp() * 1000)


BUCKET_PREFIX = os.path.join(PREFIX, StartDate.strftime('%Y{0}%m{0}%d').format(os.path.sep))


def lambda_handler(event, context):
    client = boto3.client('logs')
    response = client.create_export_task(
        logGroupName=GROUP_NAME,
        fromTime=fromDate,
        to=toDate,
        destination=DESTINATION_BUCKET,
        destinationPrefix=BUCKET_PREFIX
    )
    print(response)
```



STEP 8: Deploy and test the code.



STEP 9: Now check that if the logs are copied inside the bucket or not.

