Graded Assignment on Serverless Architecture

Table of Contents

[1. S3 Setup: 1](#_Toc148623228)

[2. Lambda IAM Role: 3](#_Toc148623229)

[3. Lambda Function: 5](#_Toc148623230)

[4. Manual Invocation: 7](#_Toc148623231)

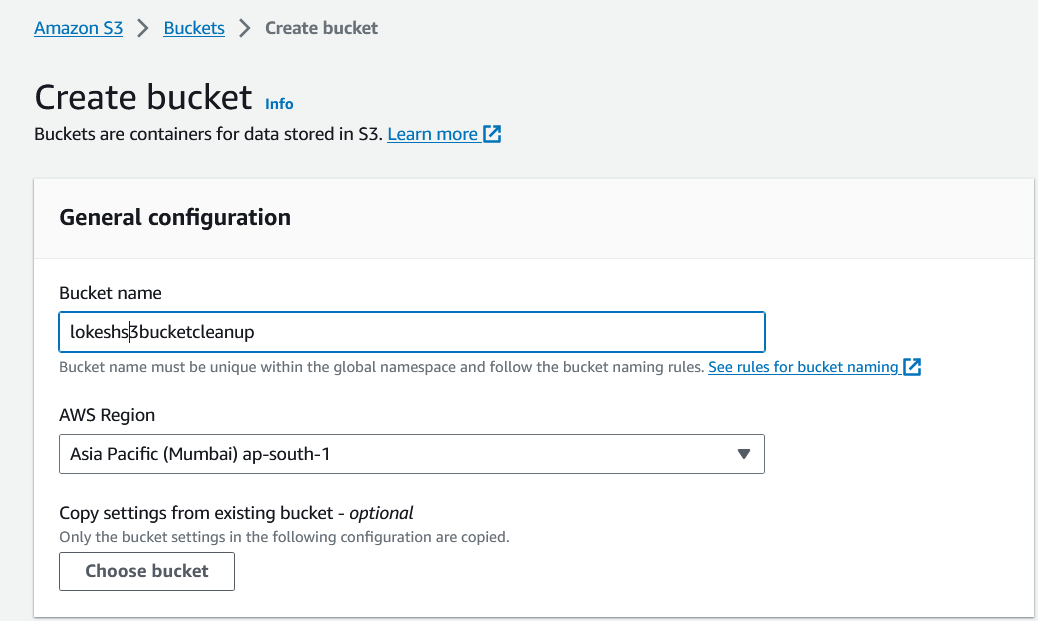
Assignment 2: Automated S3 Bucket Cleanup Using AWS Lambda and Boto3

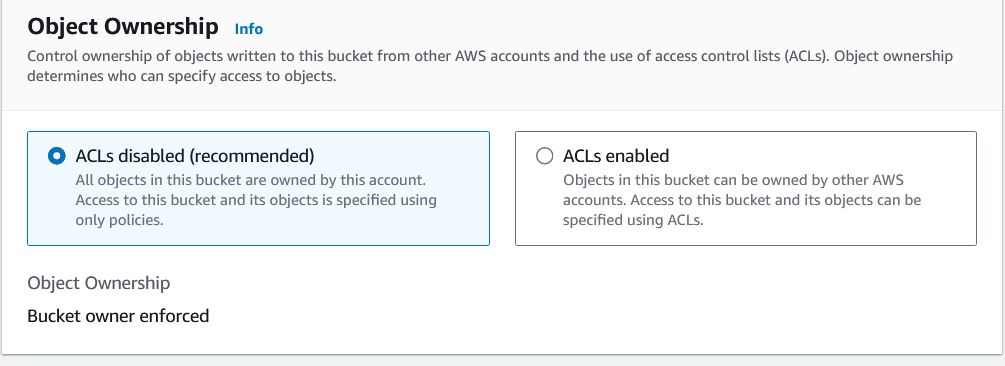
**Objective**: To gain experience with AWS Lambda and Boto3 by creating a Lambda function that will automatically clean up old files in an S3 bucket.

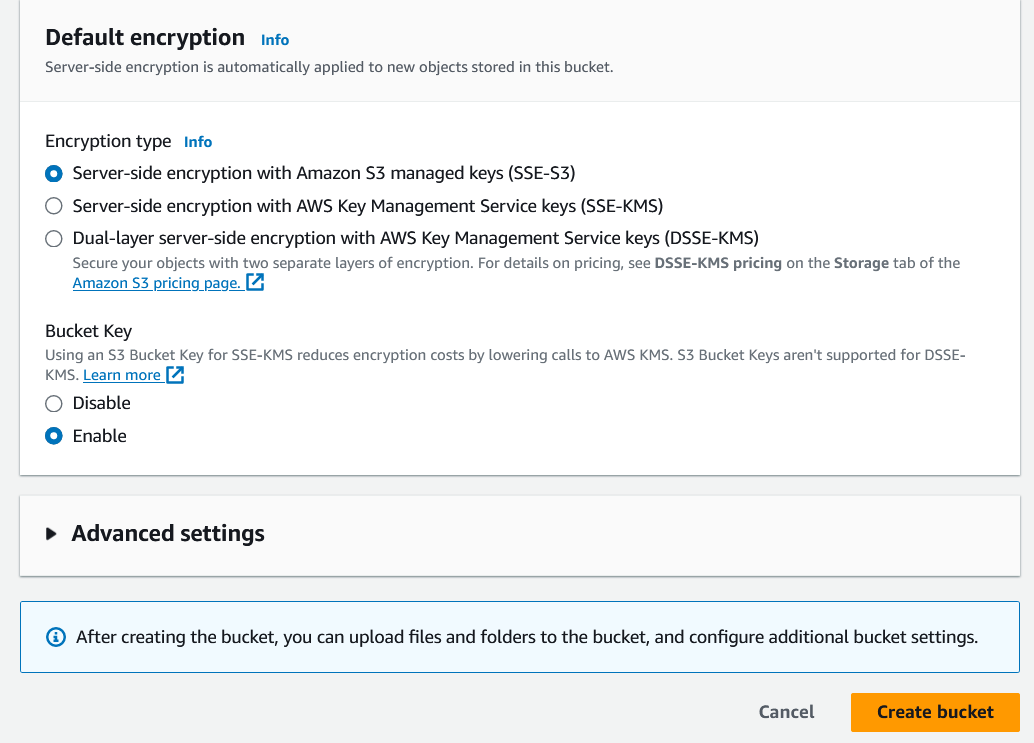
**Task:** Automate the deletion of files older than 30 days in a specific S3 bucket.

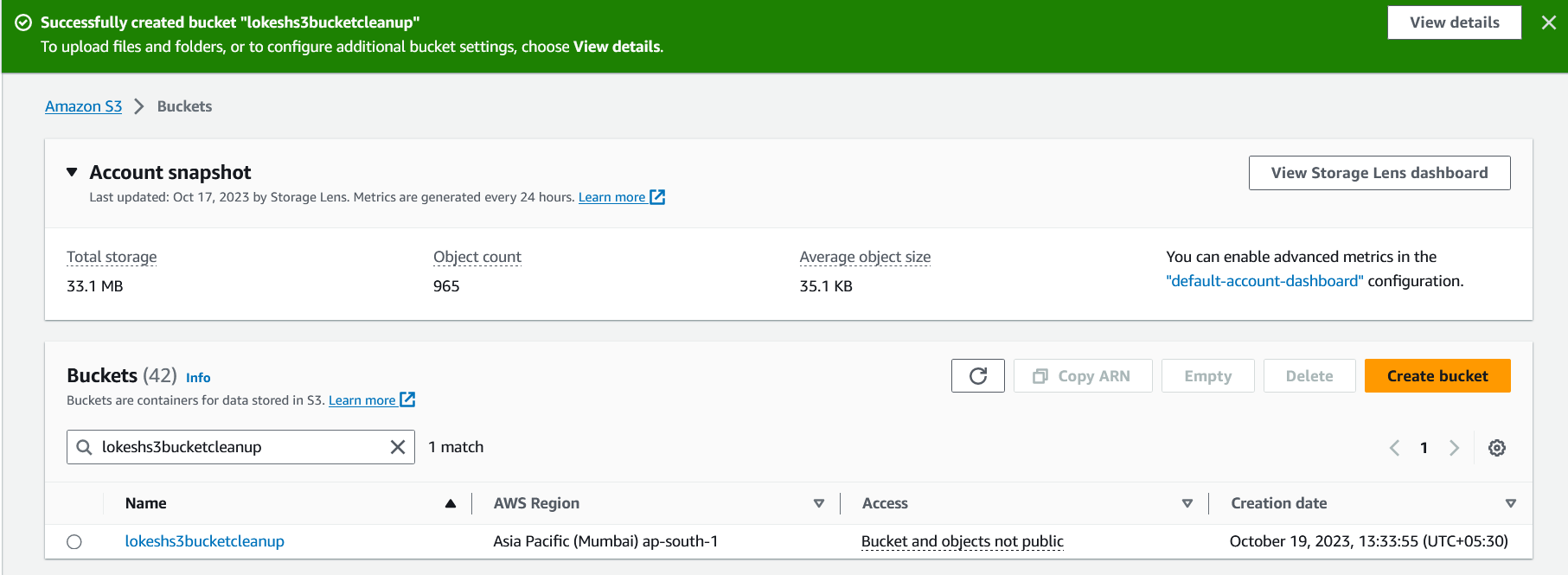
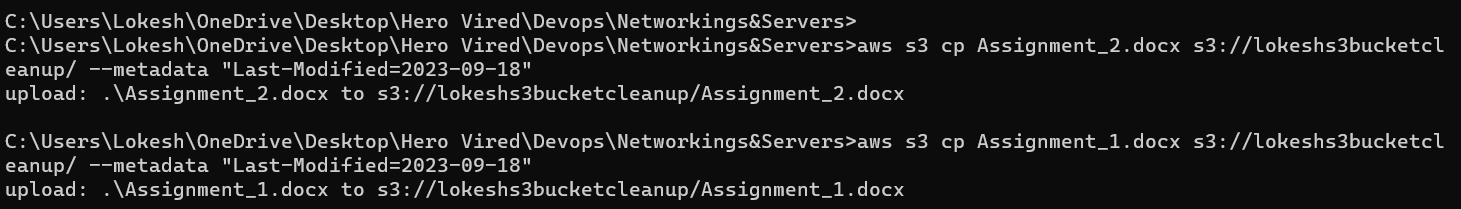
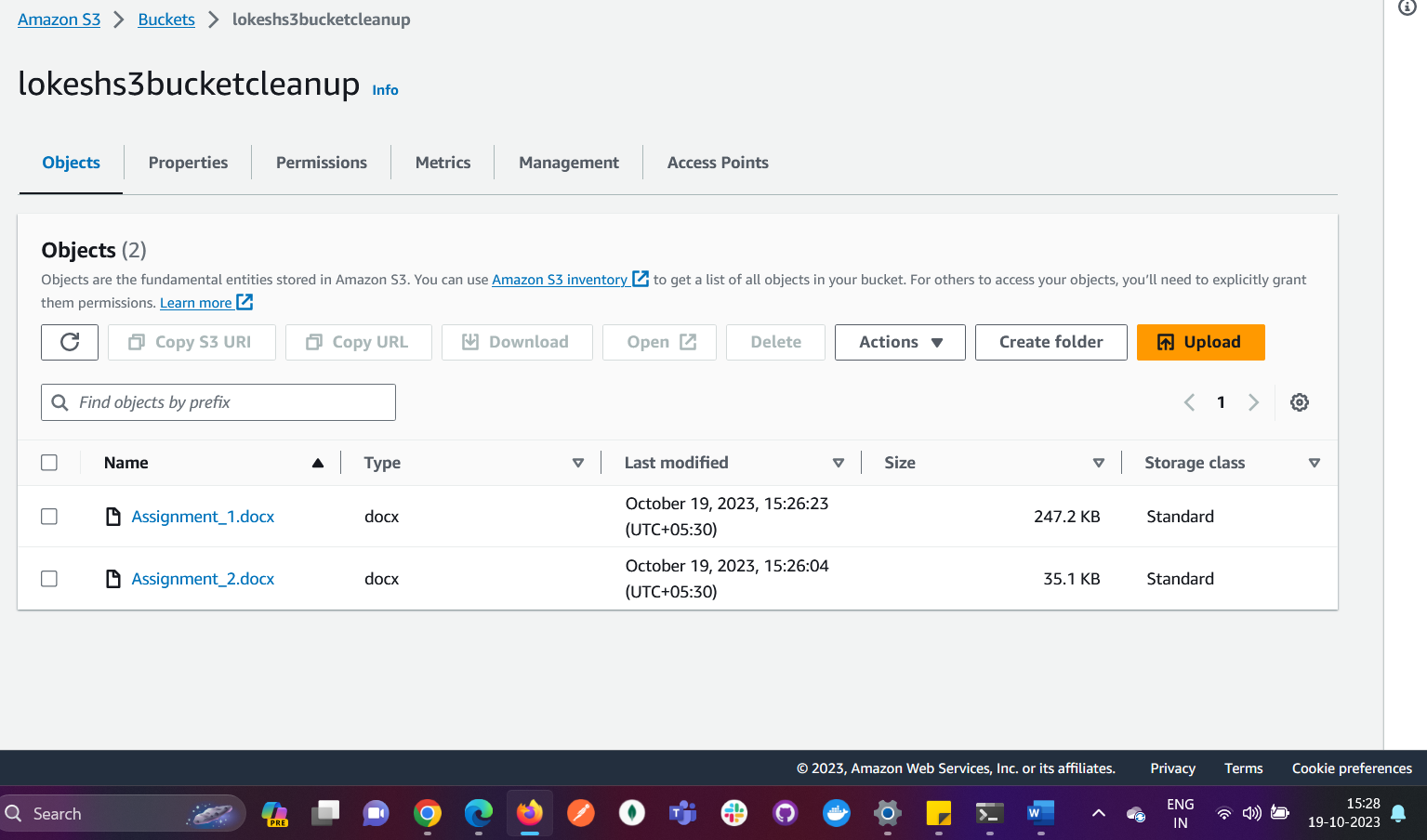
# S3 Setup:

* Navigate to the S3 dashboard and create a new bucket.

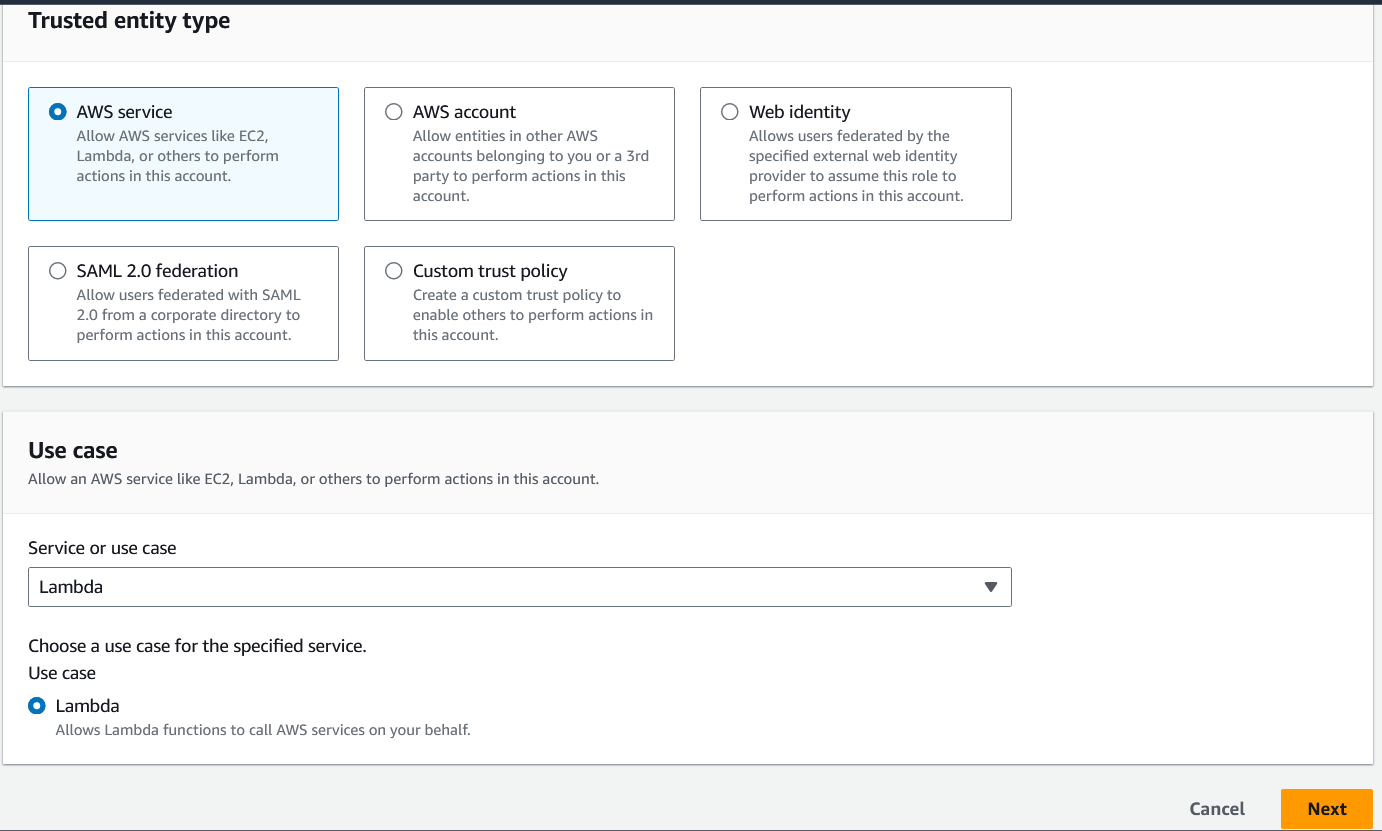
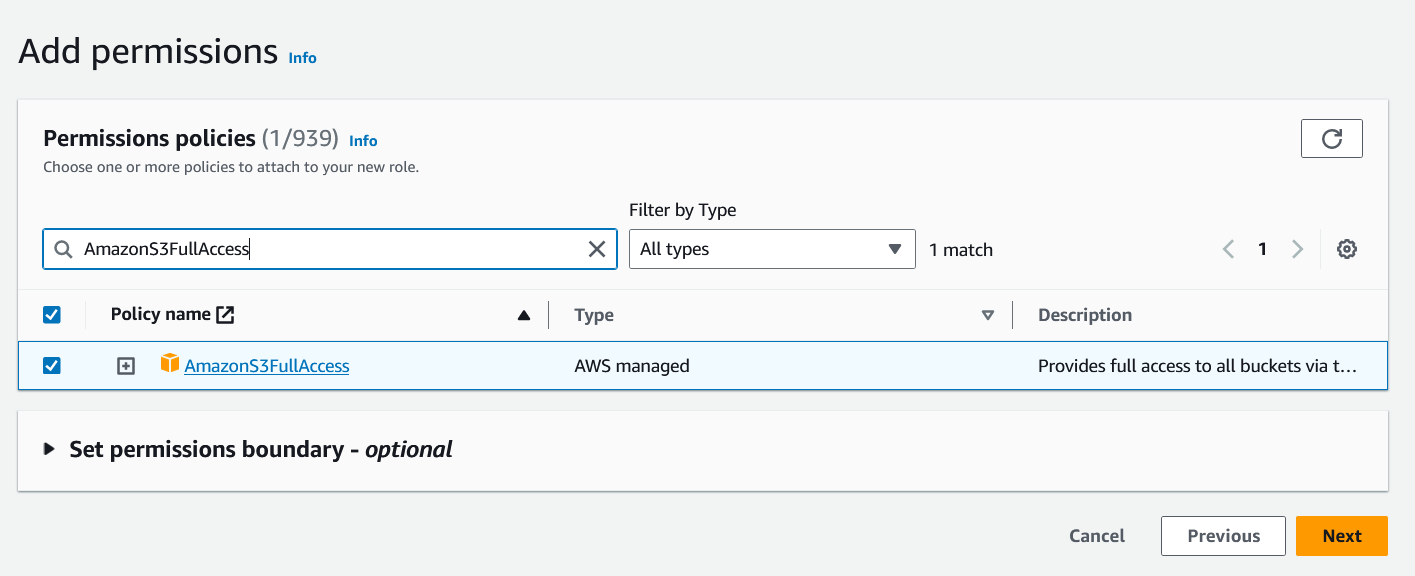
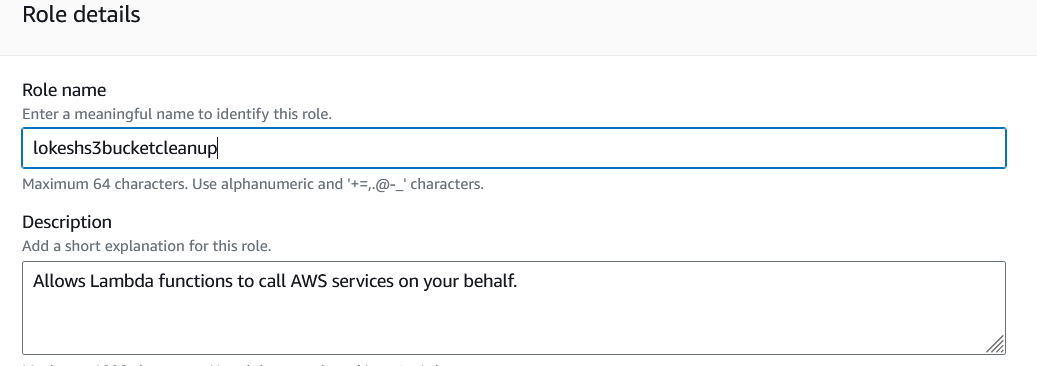
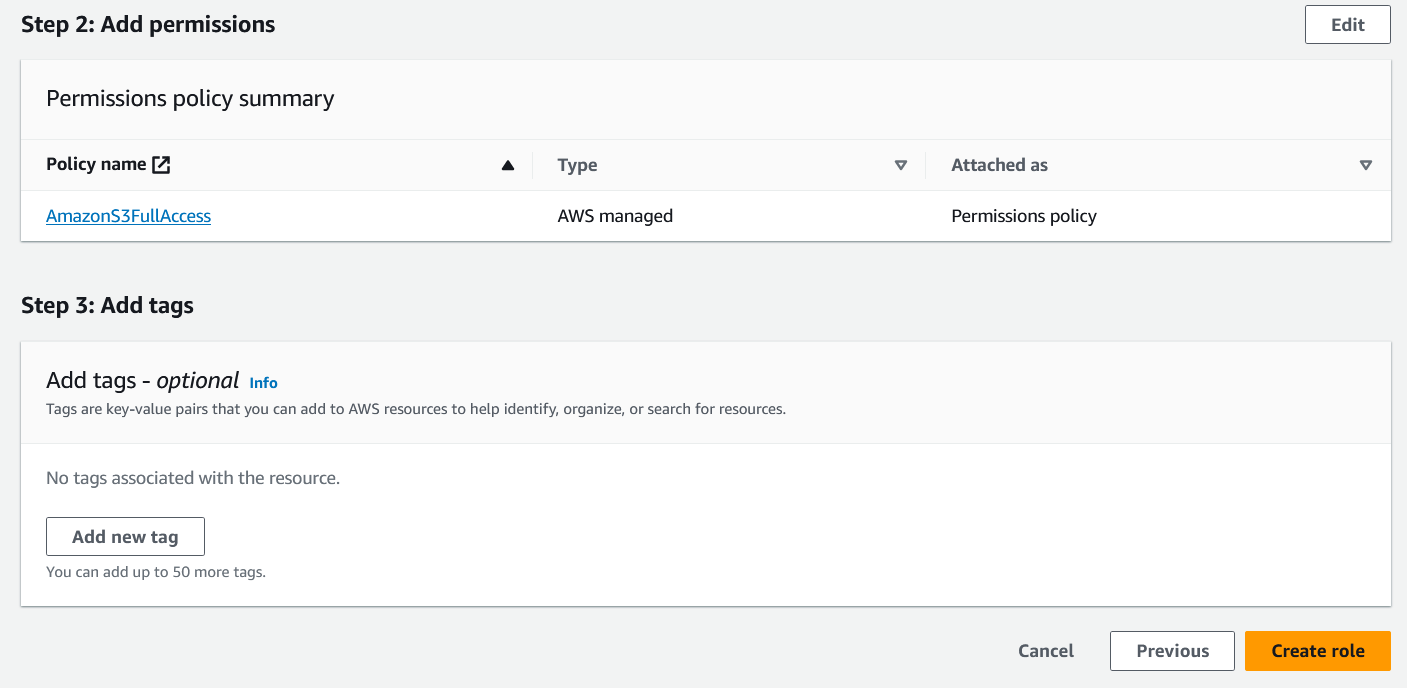
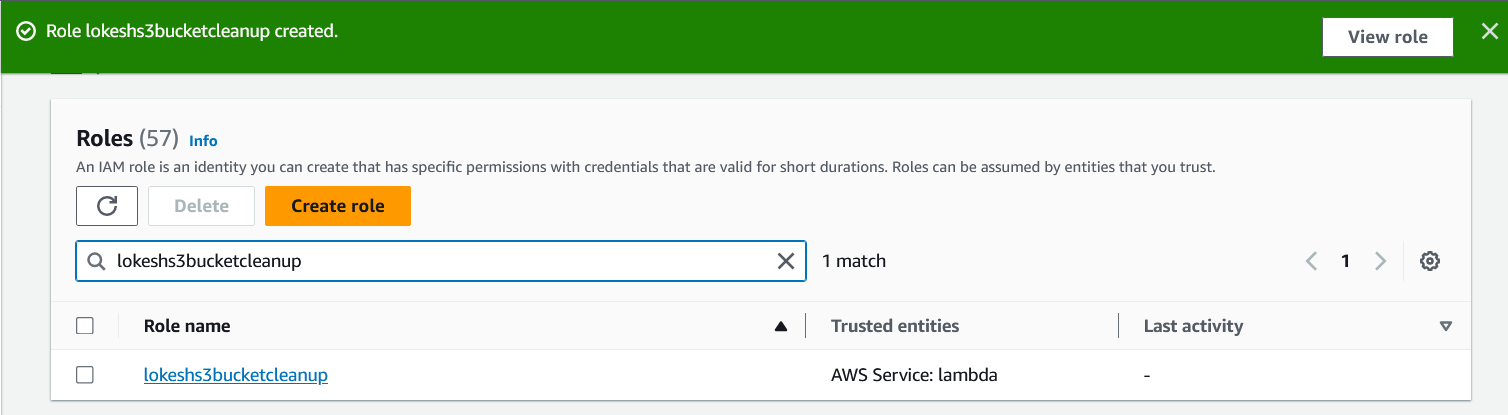




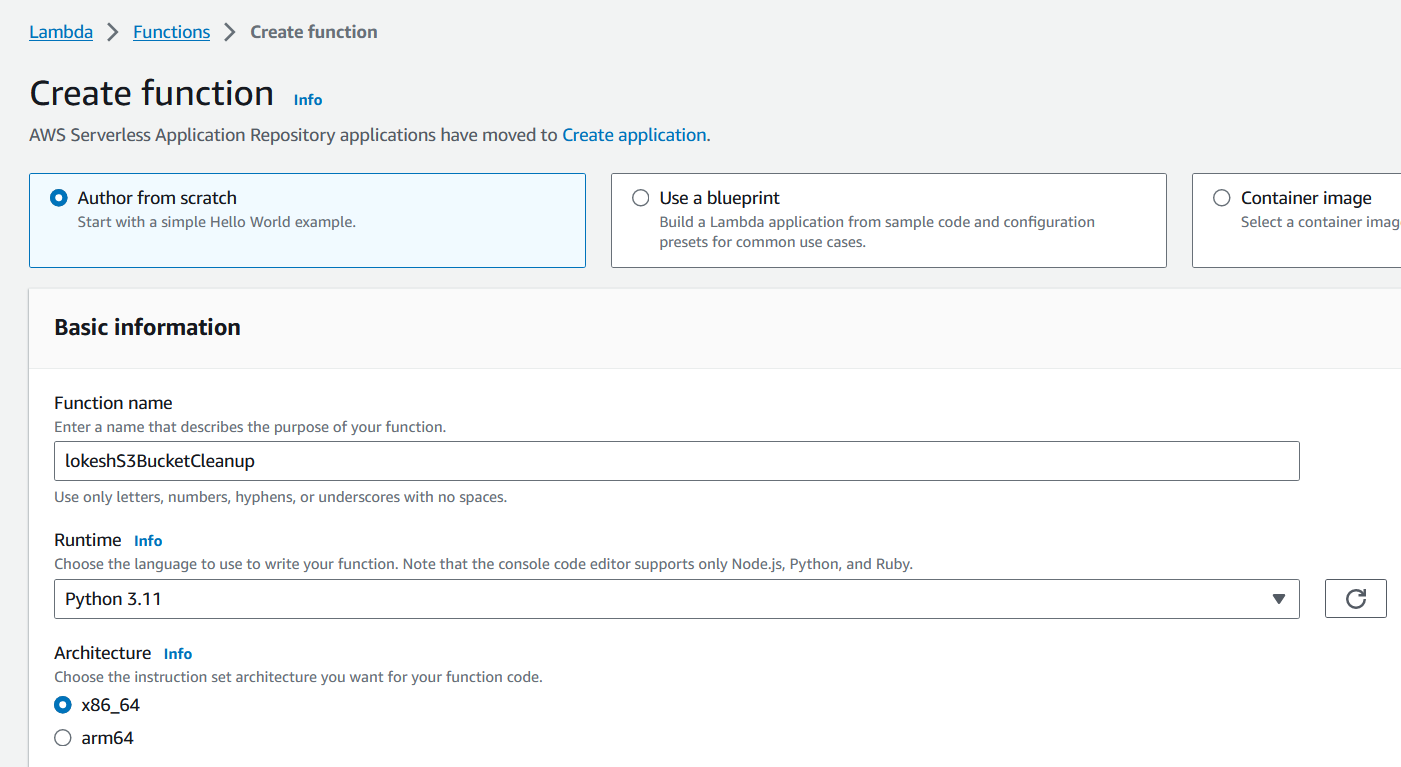
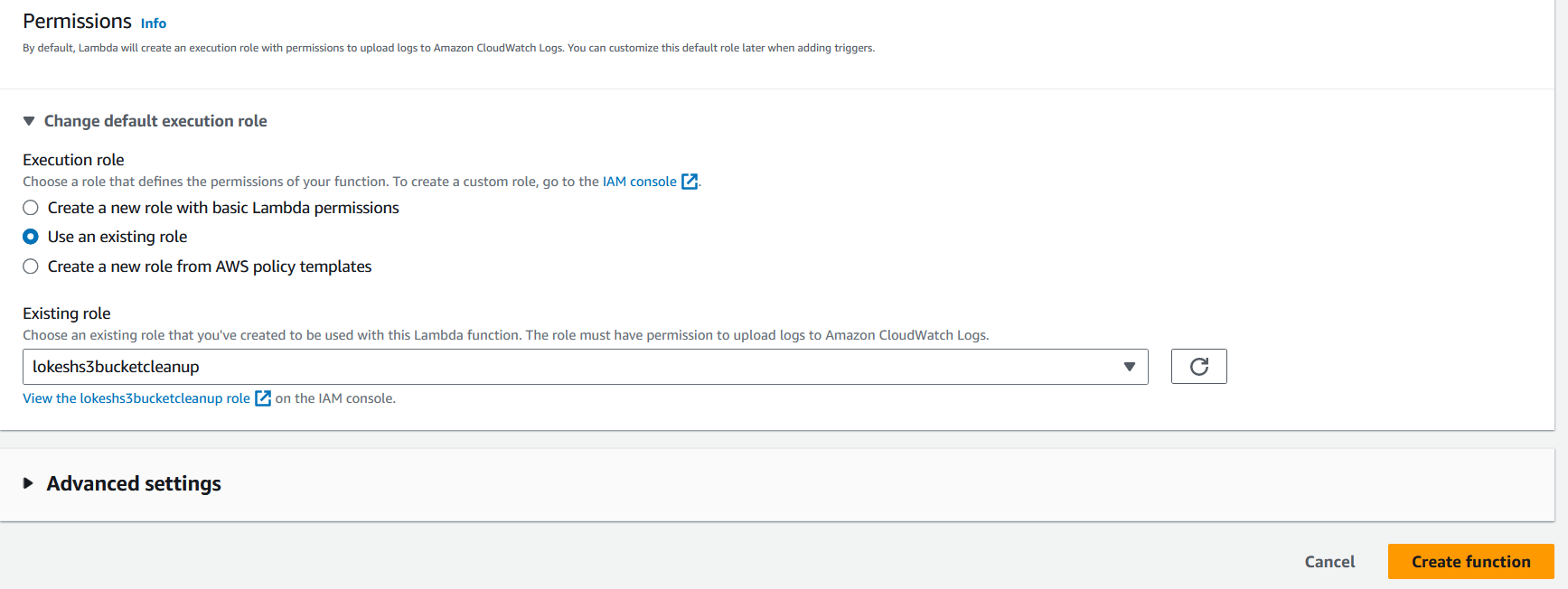
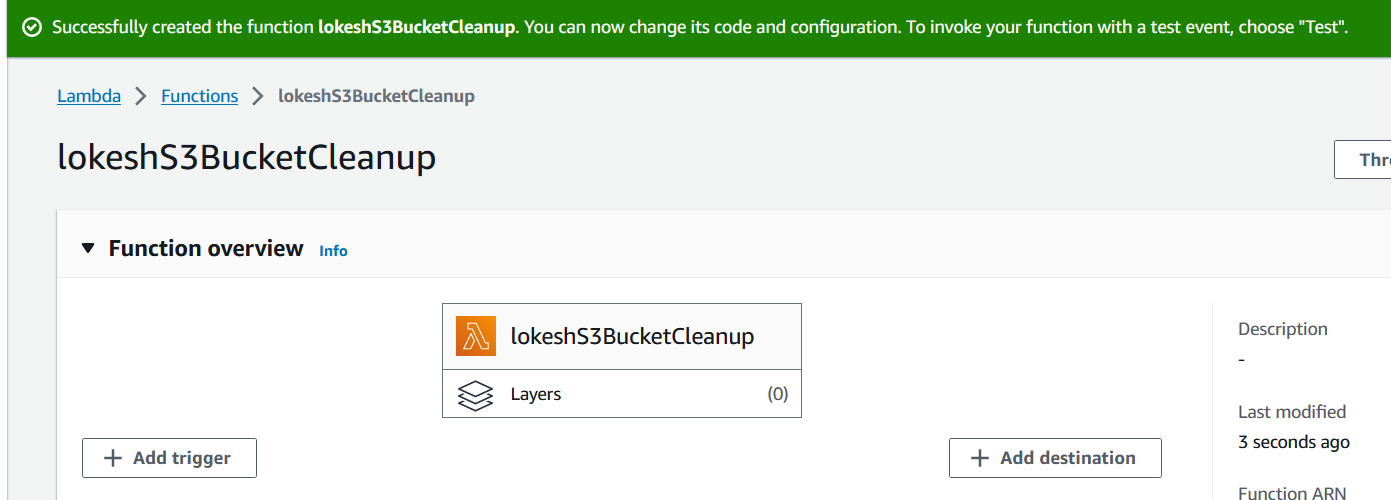
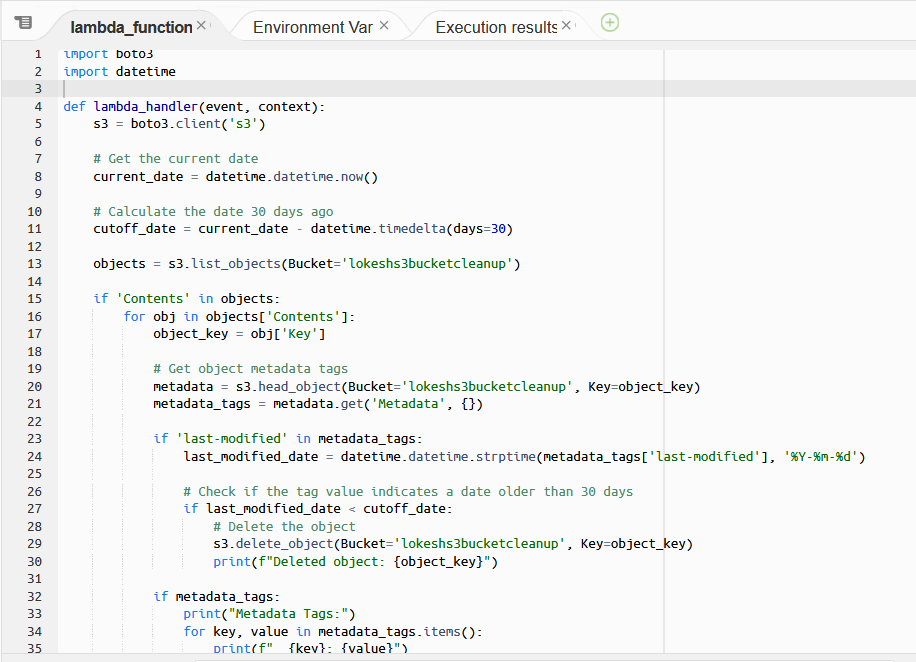


* We have successfully created S3 bucket with the name **lokeshs3bucketcleanup**.
* We need to upload some random files to test. The below is the command to upload the file without changing the system time.
* 
* 

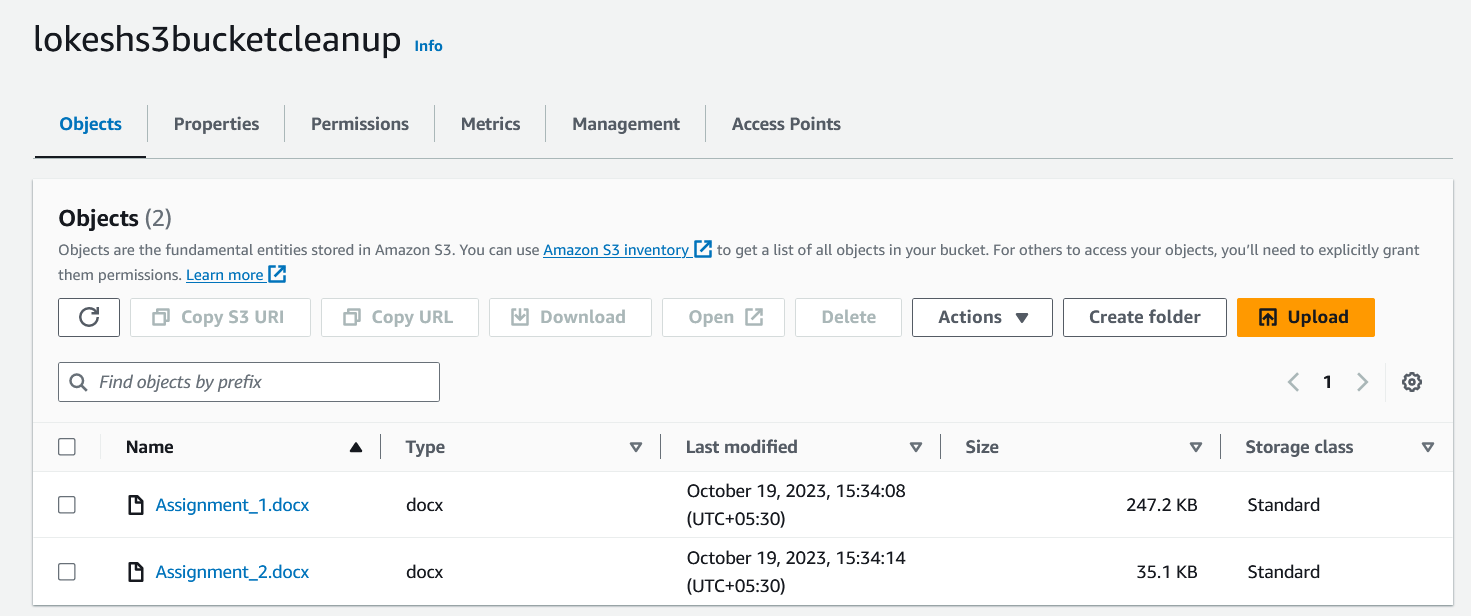
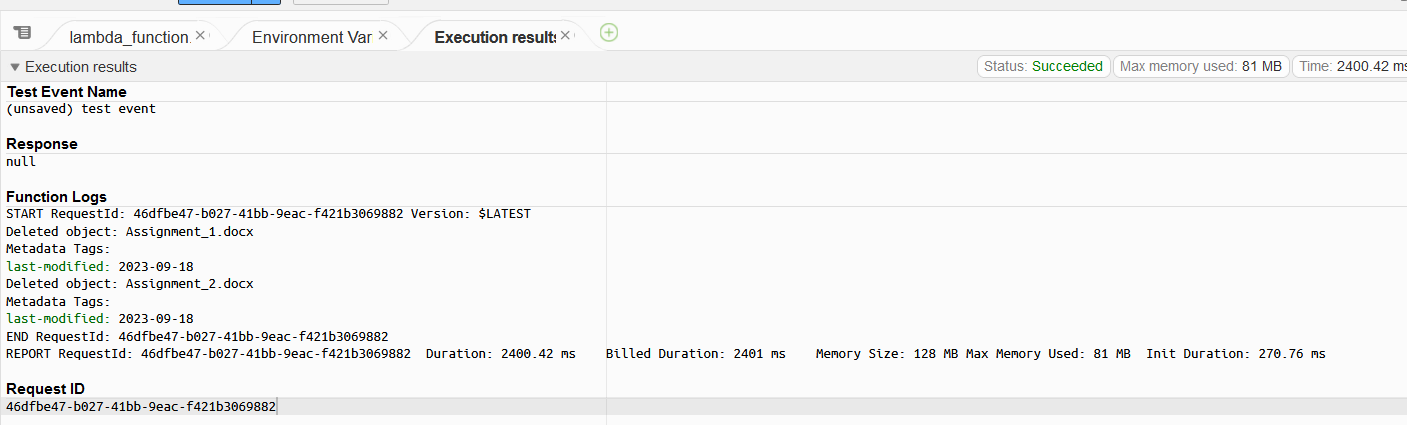
# Lambda IAM Role:

* In the IAM dashboard, create a new role for Lambda and select AmazonS3FullAccess for the permissions. Then in the next page give the name and click on create **Role**.
* 
* 
* 
* 
* A new Role with the name lokeshs3bucketcleanup has been created.

# Lambda Function:

* Navigate to the Lambda dashboard and create a new function with the name **lokeshS3BucketCleanup** .
* 
* In the permissions we need to select IAM role we have created above and click on **Create Function**. A new function will be created as shown below.
* 
* 
* Since we have created a function, now it’s to write a code to execute the task. The below is the code.
* import boto3
* import datetime
* def lambda\_handler(event, context):
* s3 = boto3.client('s3')
* # Get the current date
* current\_date = datetime.datetime.now()
* # Calculate the date 30 days ago
* cutoff\_date = current\_date - datetime.timedelta(days=30)
* objects = s3.list\_objects(Bucket='lokeshs3bucketcleanup')
* if 'Contents' in objects:
* for obj in objects['Contents']:
* object\_key = obj['Key']
* # Get object metadata tags
* metadata = s3.head\_object(Bucket='lokeshs3bucketcleanup', Key=object\_key)
* metadata\_tags = metadata.get('Metadata', {})
* if 'last-modified' in metadata\_tags:
* last\_modified\_date = datetime.datetime.strptime(metadata\_tags['last-modified'], '%Y-%m-%d')
* # Check if the tag value indicates a date older than 30 days
* if last\_modified\_date < cutoff\_date:
* # Delete the object
* s3.delete\_object(Bucket='lokeshs3bucketcleanup', Key=object\_key)
* print(f"Deleted object: {object\_key}")
* if metadata\_tags:
* print("Metadata Tags:")
* for key, value in metadata\_tags.items():
* print(f" {key}: {value}")
* 

# Manual Invocation:

* We have 2 files in the S3 bucket now we will test the code so that the files in the S3 bucket will be deleted.
* 
* The below is the result screenshot and the files in the S3 bucket has been deleted successfully.
* 
* 