Since 1962

Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra
Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology

A.Y. 2024-25

Advance DevOps Lab

Experiment 12

Aim: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Roll No.	44
Name	GANESH SANJAY PANDHRE
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements. LO6: To engineer a composition of nano services using AWS Lambda and Step Functions with the Serverless Framework
Grade:	

AIM : To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

THEORY :

AWS Lambda and S3 Integration

Delving into the powerful integration between AWS Lambda and Amazon S3, a popular use case in serverless architecture. AWS Lambda enables automatic execution of code in response to events generated by other AWS services, such as the addition of an object to an S3 bucket.

Event-Driven Architecture

AWS Lambda operates on an event-driven model, where specific events trigger the execution of predefined functions. In this case, when an object like an image is uploaded to a specific S3 bucket, S3 generates an event notification. This event is sent to AWS Lambda, which then triggers the function you have created. The function can be designed to perform various tasks, such as processing the object, extracting metadata, or, as in this experiment, simply logging a message—“An Image has been added.”

Automation and Real-Time Processing

The integration of AWS Lambda with S3 is a prime example of automation in the cloud. It allows you to set up a real-time response system where actions are automatically triggered without manual intervention. This is particularly useful in scenarios where immediate processing or reaction is required, such as in applications that handle large volumes of incoming data or media files.

Scalability and Efficiency

One of the key advantages of using AWS Lambda with S3 is its scalability. Lambda functions can automatically scale to handle multiple events concurrently, ensuring that no matter how many objects are added to the S3 bucket, each will trigger the necessary function without delay. This setup is highly efficient, reducing operational overhead and ensuring that processes remain consistent and reliable, even under heavy workloads.

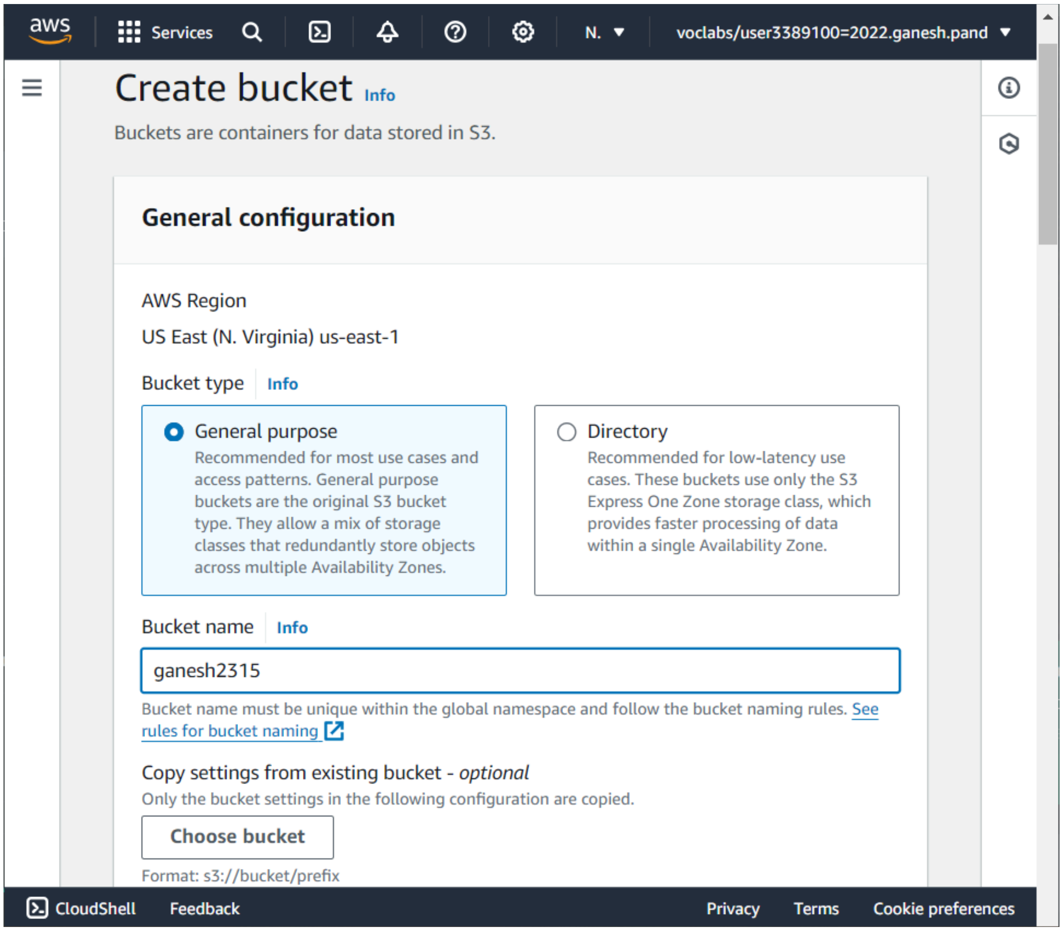
Use Cases and Practical Applications

This type of setup is widely used in various domains, including media processing, where images or videos uploaded to S3 can trigger Lambda functions to generate thumbnails or transcode files. It is also commonly used in data analytics, where data uploaded to S3 triggers processing functions that prepare the data for further analysis or storage. Another application is in backup systems, where new files added to S3 are immediately backed up or replicated in other locations.

Security Considerations

When working with AWS Lambda and S3, security is paramount. Each Lambda function interacts with S3 via permissions defined by AWS IAM roles and policies. These permissions must be carefully configured to ensure that the function has the necessary access rights—such as read and write permissions for the specific S3 bucket—without exposing unnecessary access to other resources. This ensures that the system remains secure while performing the required tasks efficiently.

Create an S3 Bucket. Enter a unique bucket name.



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

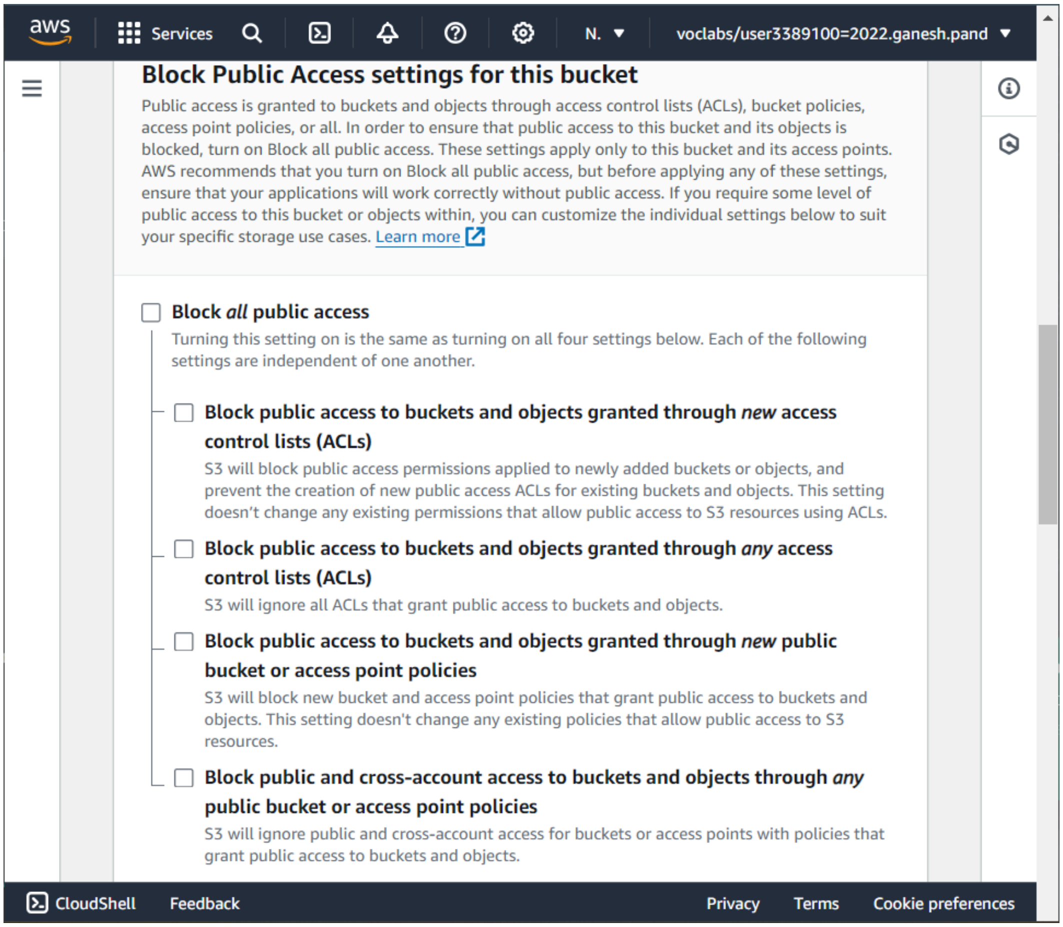
ganesh2315

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

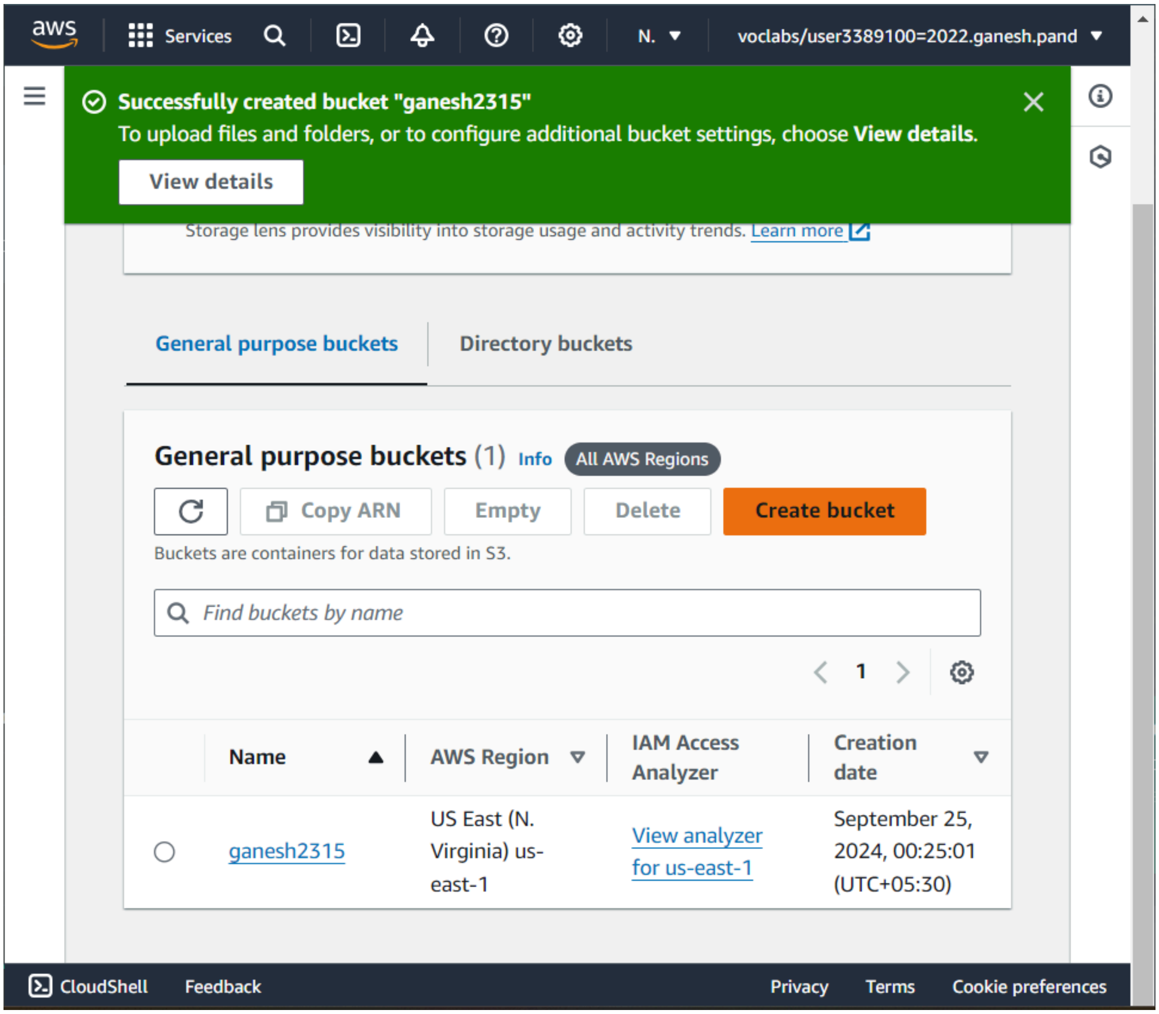


Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

□ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- □ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Successfully created bucket "ganesh2315"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

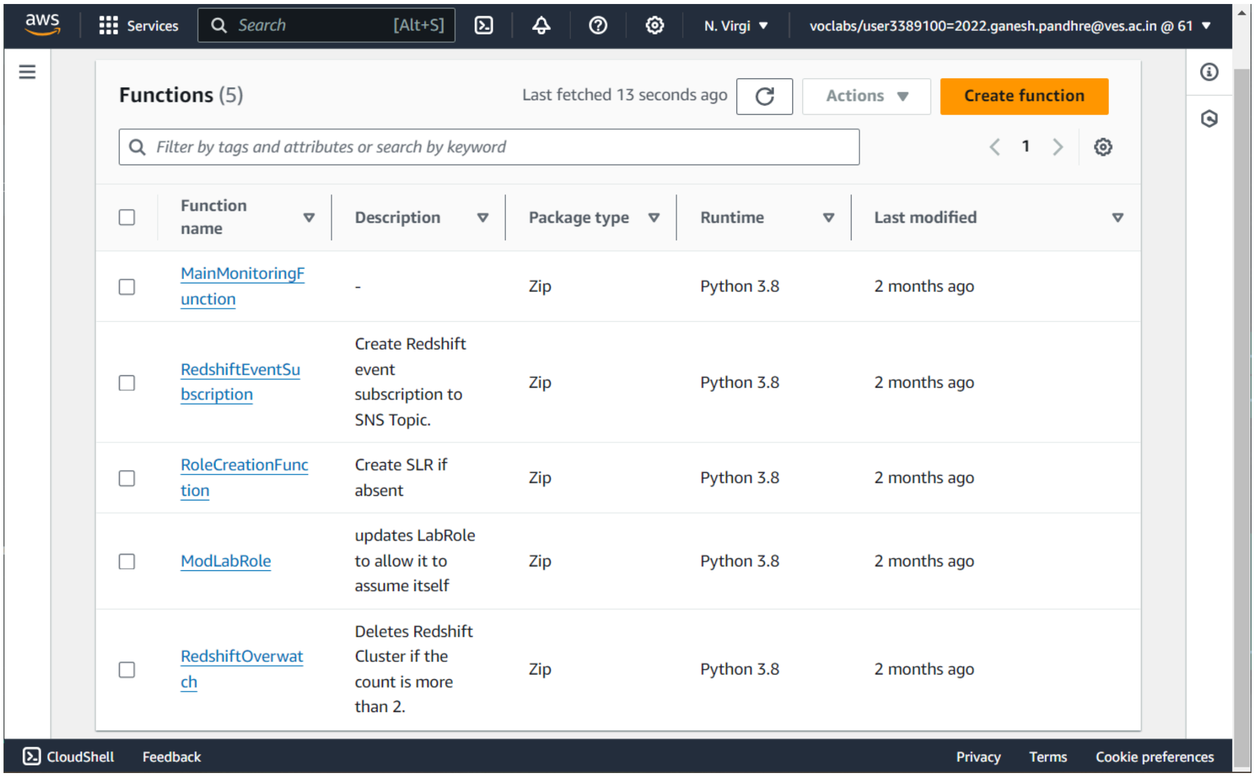
Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1)

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	ganesh2315	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 25, 2024, 00:25:01 (UTC+05:30)

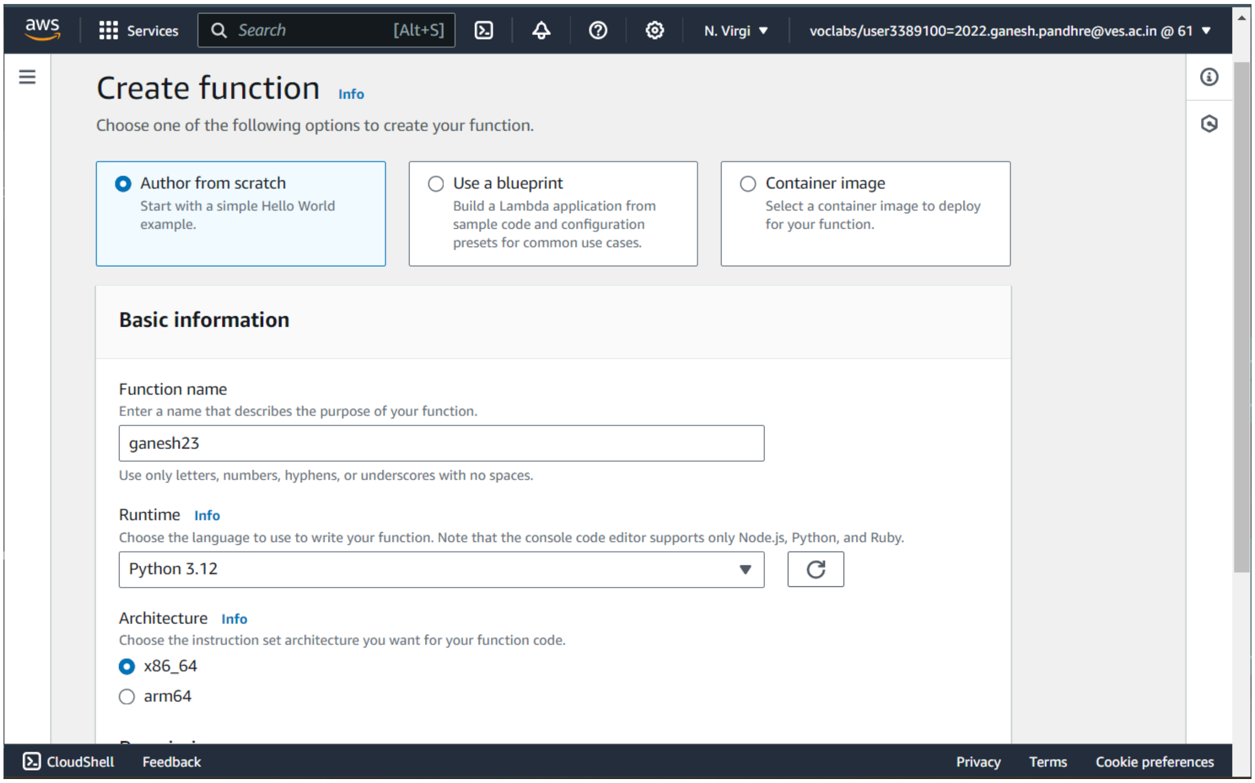
Navigate to the Lambda service. Click on "Create function."


aws
Services
Search
[Alt+S]
N. Virgi
voclabs/user3389100=2022.ganesh.pandhre@ves.ac.in @ 61
Actions
Create function
Filter by tags and attributes or search by keyword
1
CloudShell
Feedback
Privacy
Terms
Cookie preferences

Functions (5) Last fetched 13 seconds ago

□	Function name	Description	Package type	Runtime	Last modified
□	MainMonitoringFunction	-	Zip	Python 3.8	2 months ago
□	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	2 months ago
□	RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	2 months ago
□	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	2 months ago
□	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	2 months ago

Choose "Author from scratch." Enter a name for your function. Select a runtime (Python 3.x).



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.

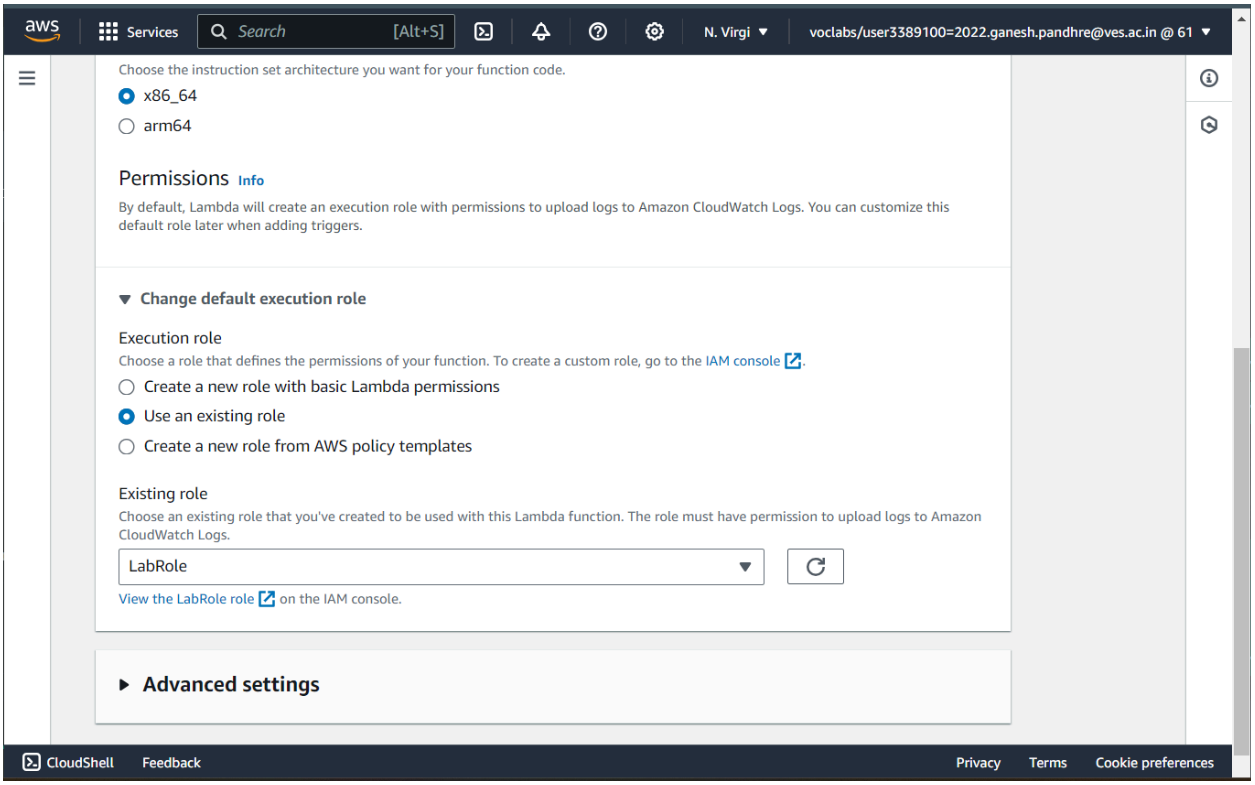
○ Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
ganesh23
Use only letters, numbers, hyphens, or underscores with no spaces.

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



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

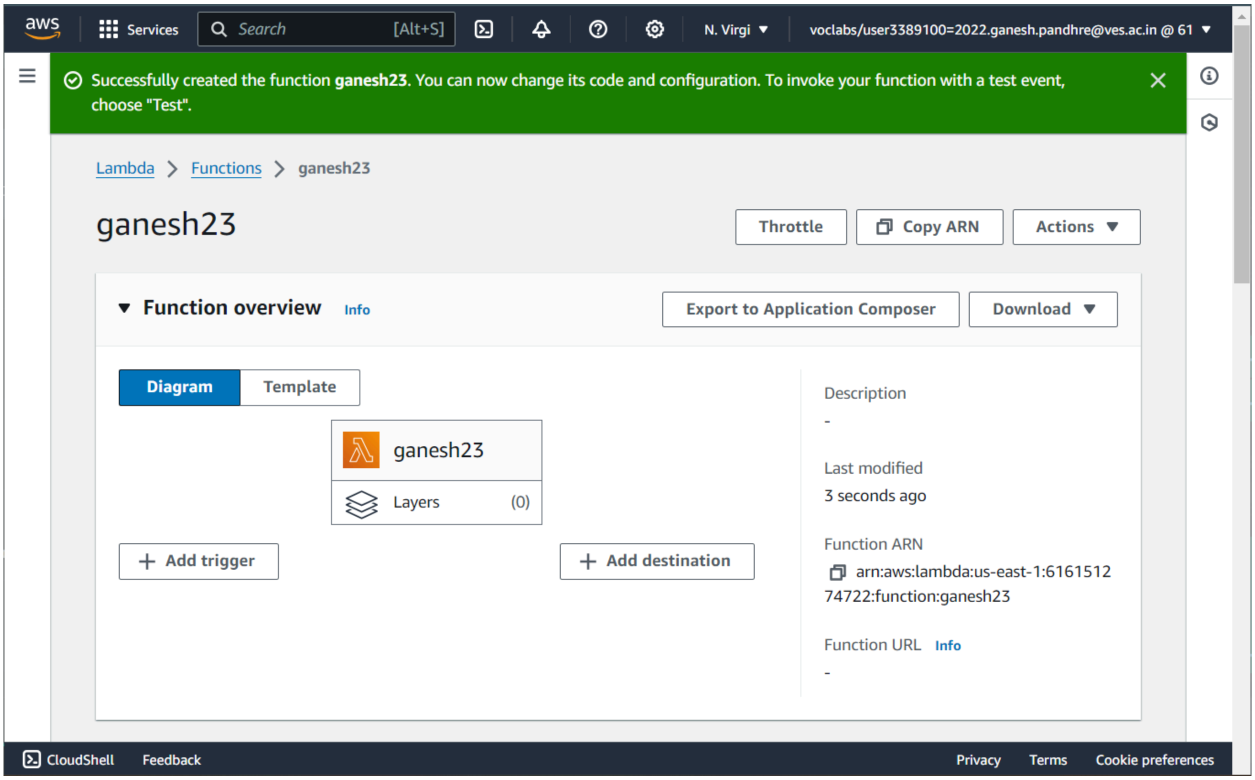
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ 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.

▶ Advanced settings



Successfully created the function **ganesh23**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > ganesh23

ganesh23

Function overview Info

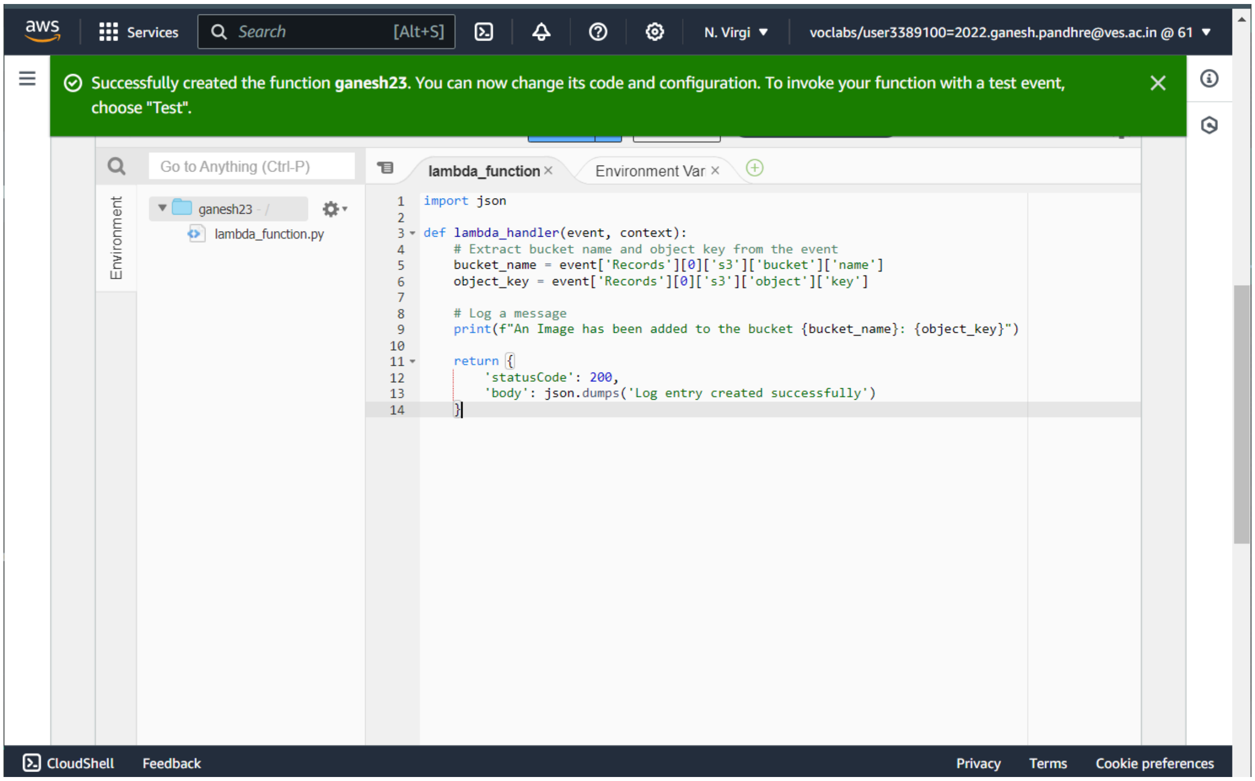
Description
-

Last modified
3 seconds ago

Function ARN
arn:aws:lambda:us-east-1:616151274722:function:ganesh23

Function URL Info
-

Write the Lambda Function Code. Click "Deploy" to save your changes.



Successfully created the function **ganesh23**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

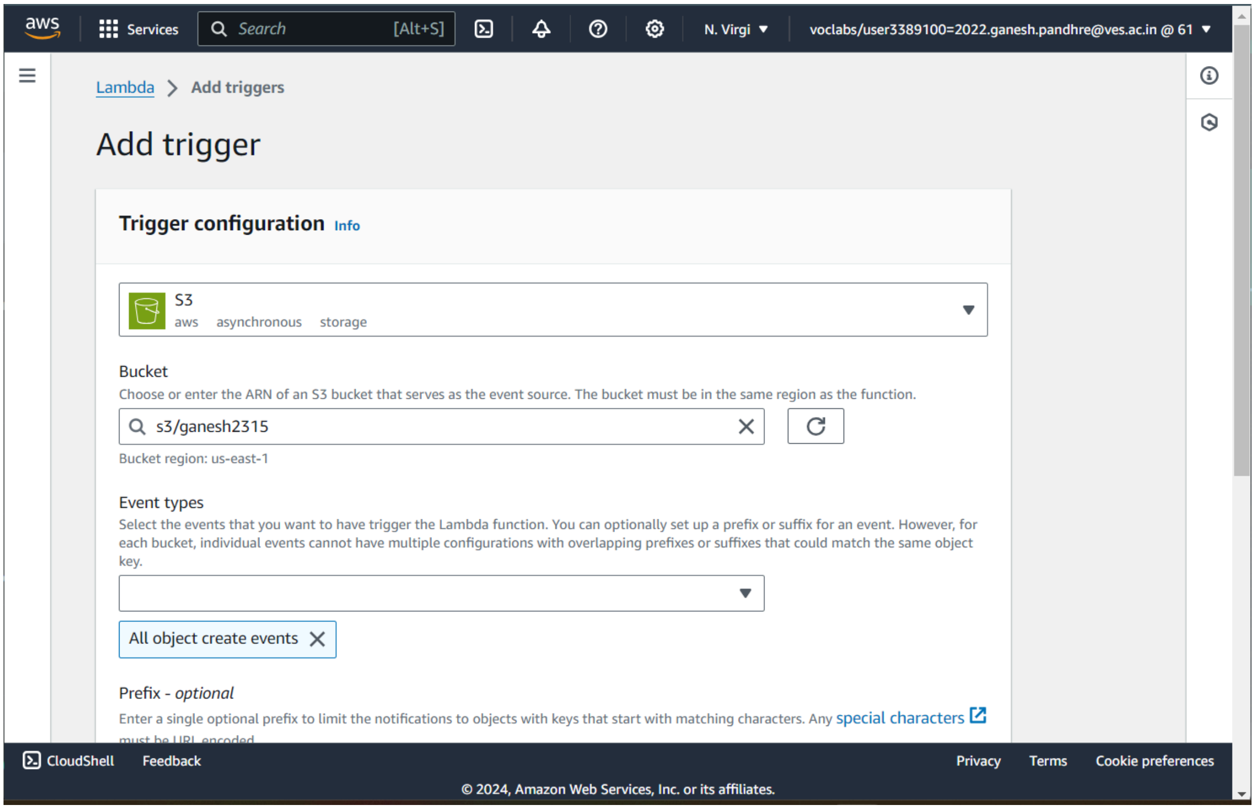
```python
import json

def lambda_handler(event, context):
    # Extract bucket name and object key from the event
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    object_key = event['Records'][0]['s3']['object']['key']

    # Log a message
    print(f"An Image has been added to the bucket {bucket_name}: {object_key}")

    return {
        'statusCode': 200,
        'body': json.dumps('Log entry created successfully')
    }
```

Select "S3" from the list of triggers. Choose the S3 bucket you created earlier. In the "Event type" dropdown, select "All object create events." Click "Add."



Add trigger

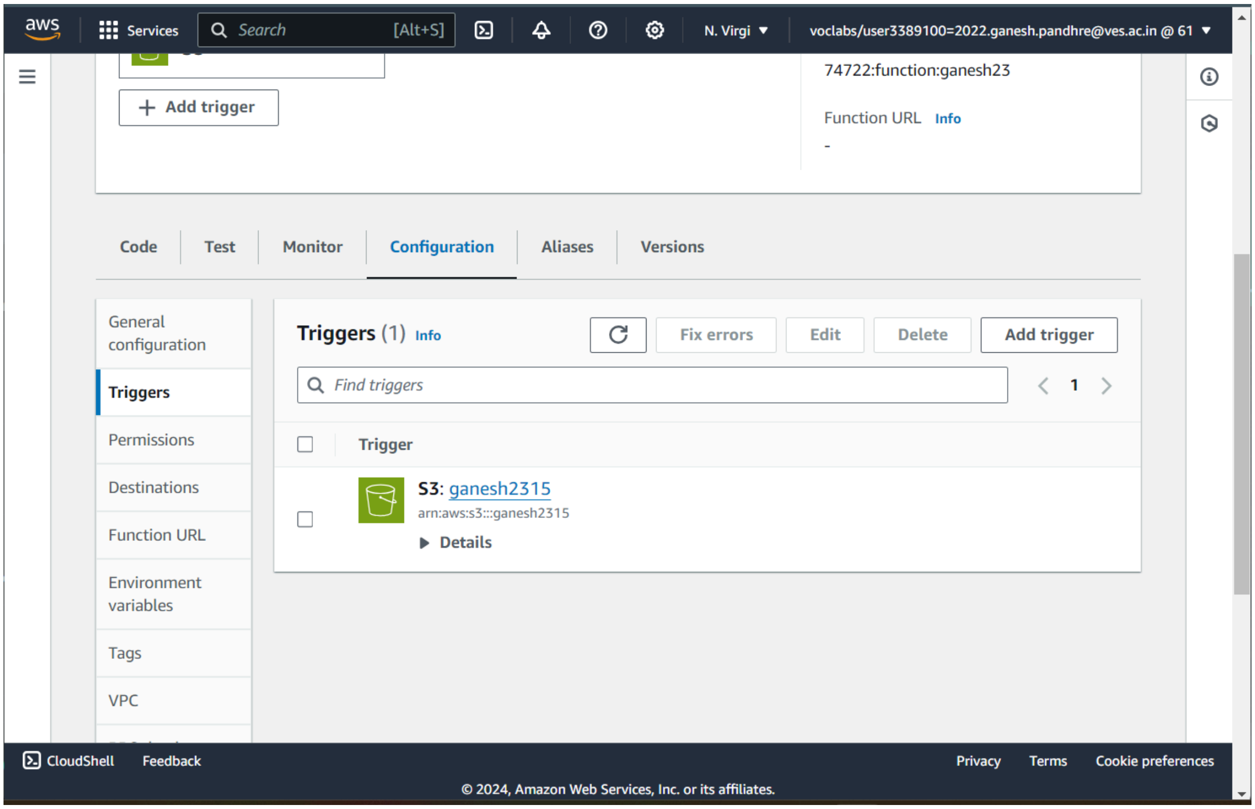
Trigger configuration Info

S3
aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/ganesh2315
Bucket region: us-east-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object create events

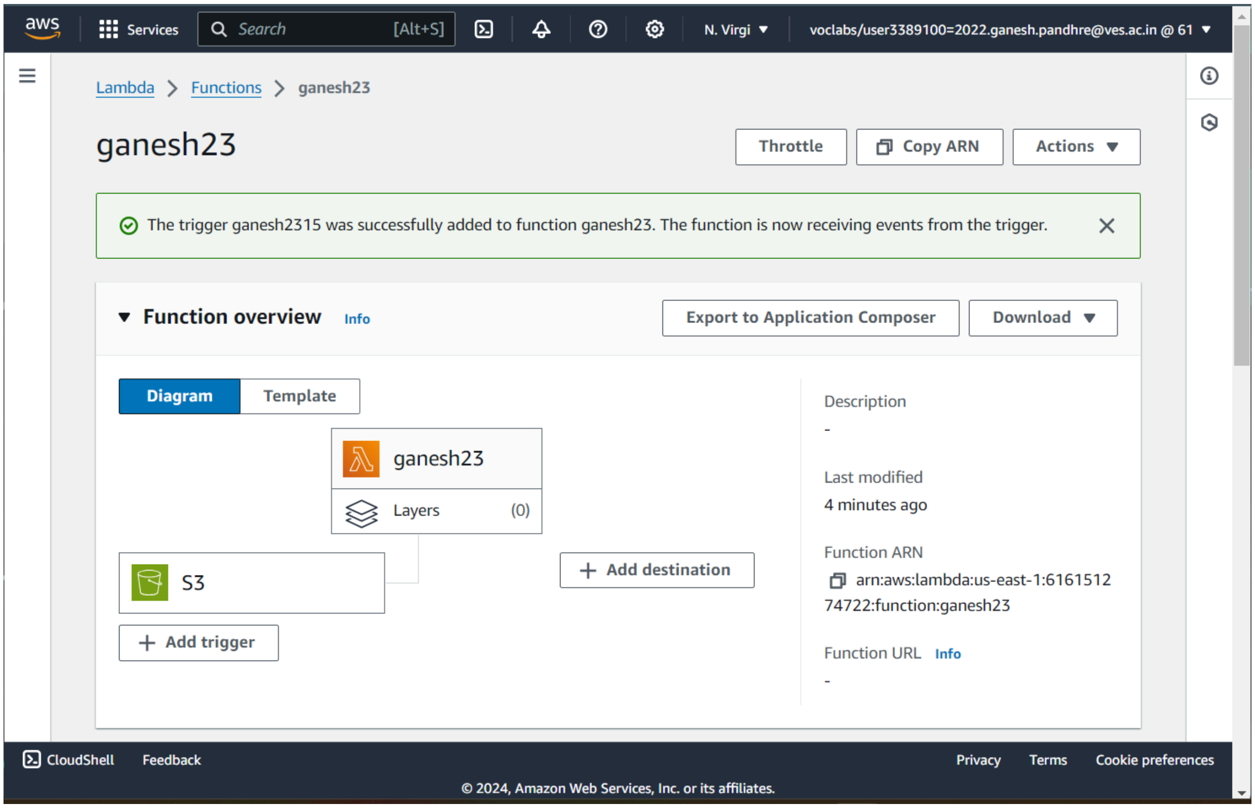
Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.



74722:function:ganesh23
Function URL Info
-

Triggers (1) Info

□	Trigger
□	S3: ganesh2315 arn:aws:s3:::ganesh2315 ▶ Details



ganesh23

The trigger ganesh2315 was successfully added to function ganesh23. The function is now receiving events from the trigger.

Function overview Info

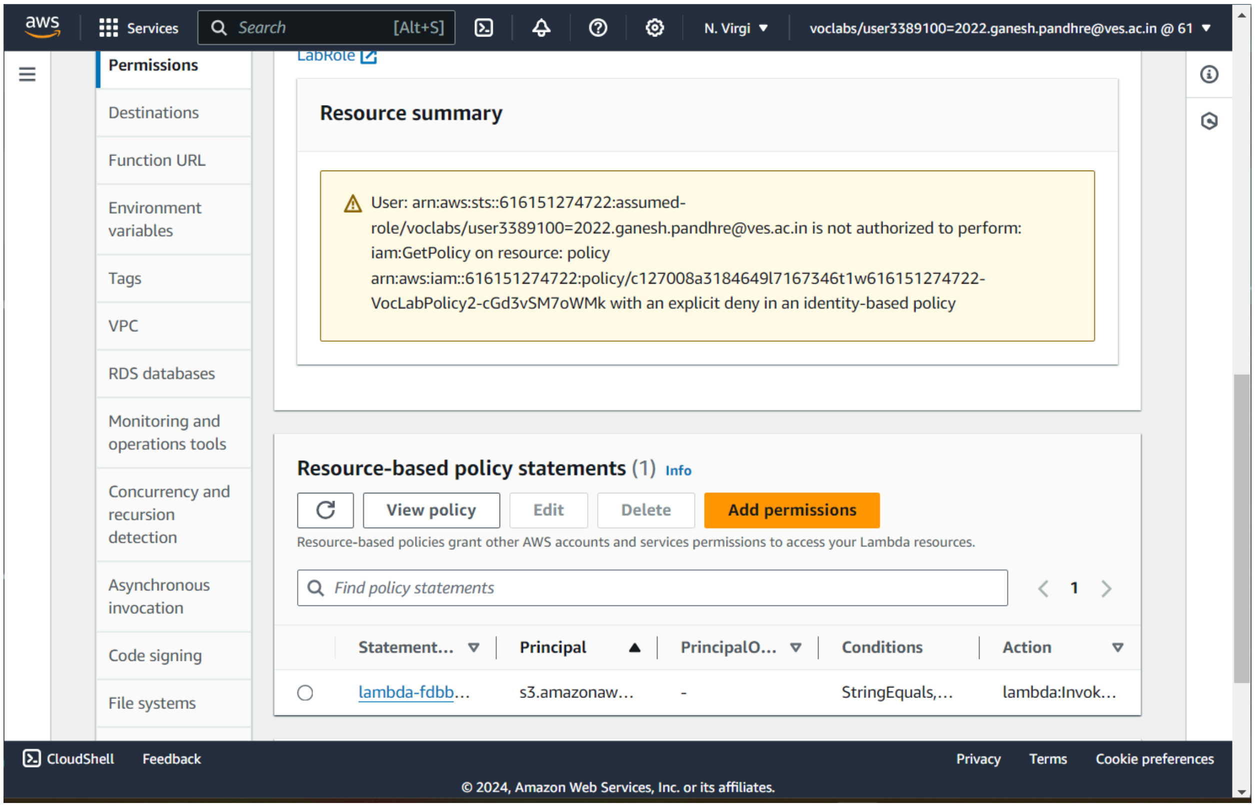
Description
-

Last modified
4 minutes ago

Function ARN
arn:aws:lambda:us-east-1:616151274722:function:ganesh23

Function URL Info
-

Navigate to the "Permissions" tab of your Lambda function. Ensure the Lambda function's execution role has the necessary permissions to access the S3 bucket.



Resource summary

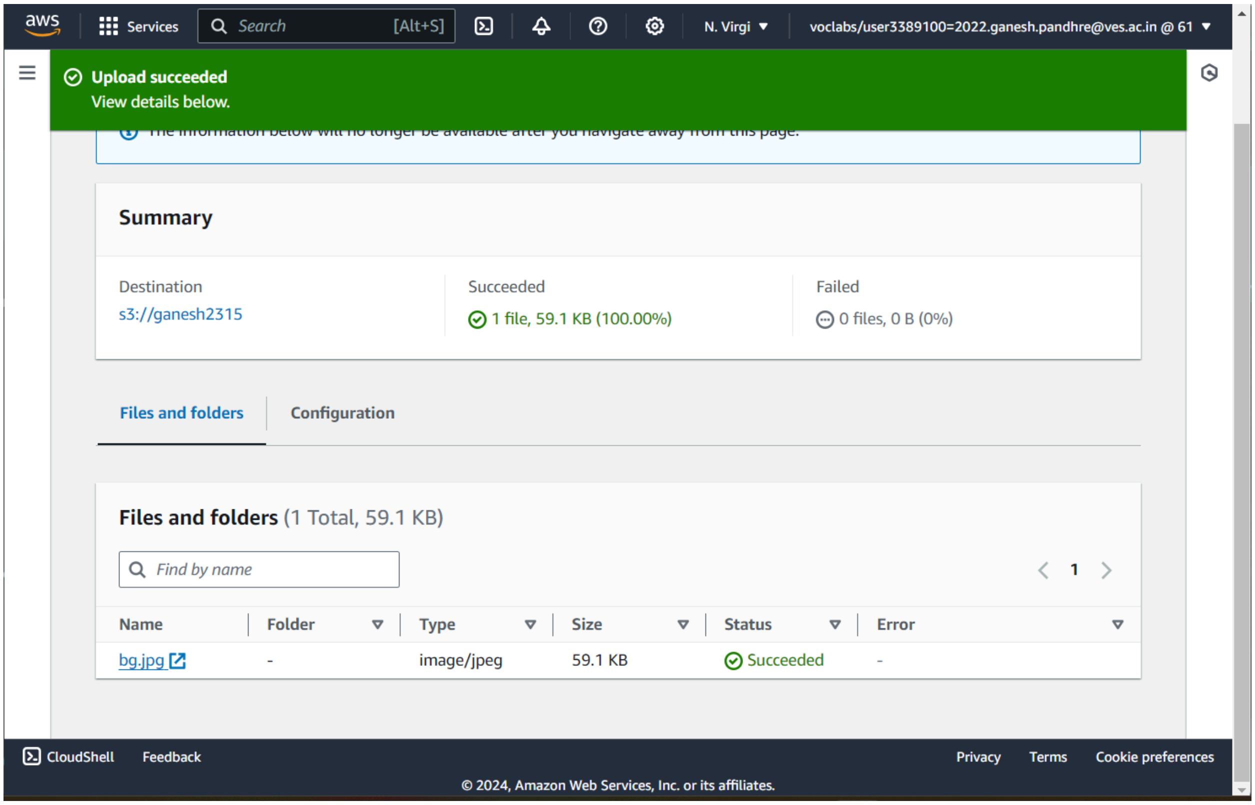
User: arn:aws:sts::616151274722:assumed-role/voclabs/user3389100=2022.ganesh.pandhre@ves.ac.in is not authorized to perform: iam:GetPolicy on resource: policy arn:aws:iam::616151274722:policy/c127008a3184649l7167346t1w616151274722-VocLabPolicy2-cGd3vSM7oWMk with an explicit deny in an identity-based policy

Resource-based policy statements (1) Info

Resource-based policies grant other AWS accounts and services permissions to access your Lambda resources.

	Statement...	Principal	PrincipalO...	Conditions	Action
○	lambda-fdbb...	s3.amazonaw...	-	StringEquals,...	lambda:Invok...

Upload an image file to your S3 bucket.



Upload succeeded
View details below.

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

Summary

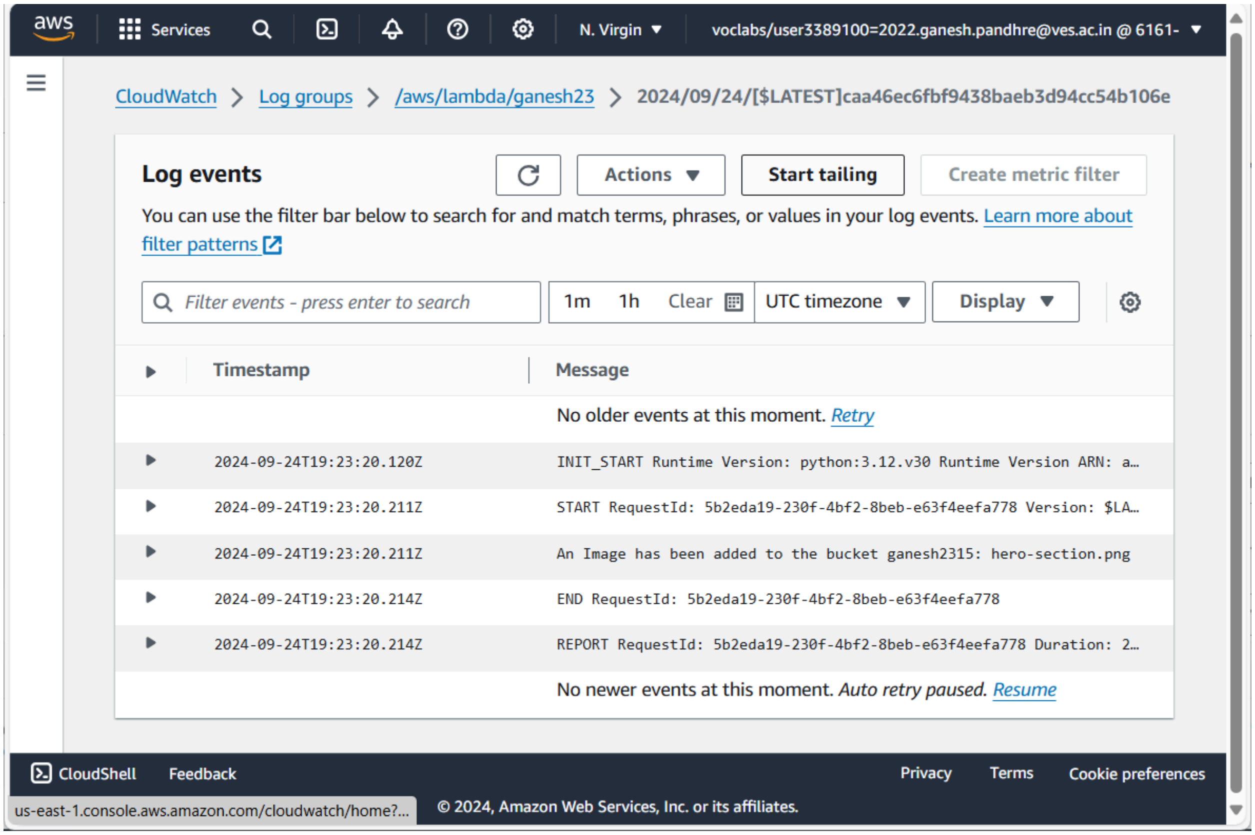
Destination	Succeeded	Failed
s3://ganesh2315	✓ 1 file, 59.1 KB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 59.1 KB)

Name	Folder	Type	Size	Status	Error
bg.jpg	-	image/jpeg	59.1 KB	✓ Succeeded	-

© 2024, Amazon Web Services, Inc. or its affiliates.

Go to the "Monitoring" tab in your Lambda function to check the logs. Use CloudWatch Logs to view the output and confirm that the message "An Image has been added" has been logged.


aws
Services
N. Virgin
voclabs/user3389100=2022.ganesh.pandhre@ves.ac.in @ 6161-

CloudWatch > Log groups > /aws/lambda/ganesh23 > 2024/09/24/[$LATEST]caa46ec6fbf9438baeb3d94cc54b106e

Log events

Actions
Start tailing
Create metric filter

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

Filter events - press enter to search
1m
1h
Clear
UTC timezone
Display

Timestamp	Message
	No older events at this moment. Retry
2024-09-24T19:23:20.120Z	INIT_START Runtime Version: python:3.12.v30 Runtime Version ARN: a…
2024-09-24T19:23:20.211Z	START RequestId: 5b2eda19-230f-4bf2-8beb-e63f4eefa778 Version: $LA…
2024-09-24T19:23:20.211Z	An Image has been added to the bucket ganesh2315: hero-section.png
2024-09-24T19:23:20.214Z	END RequestId: 5b2eda19-230f-4bf2-8beb-e63f4eefa778
2024-09-24T19:23:20.214Z	REPORT RequestId: 5b2eda19-230f-4bf2-8beb-e63f4eefa778 Duration: 2…
	No newer events at this moment. *Auto retry paused.* Resume

© 2024, Amazon Web Services, Inc. or its affiliates.

CONCLUSION : By integrating AWS Lambda with S3, you can create a seamless, automated process that responds instantly to changes in your S3 bucket. This not only improves efficiency but also allows for greater flexibility in handling data and media in real-time, making it a powerful tool in modern cloud-based applications.