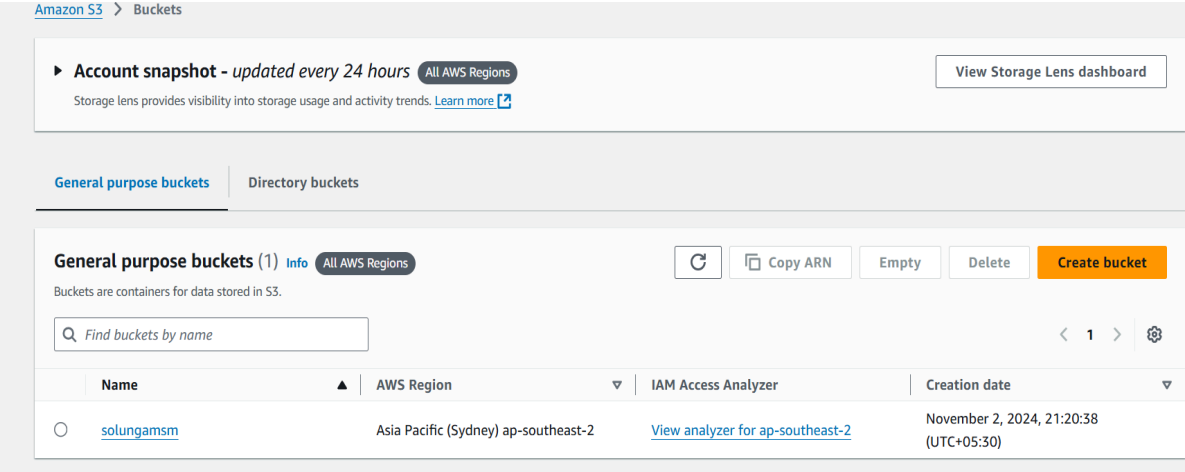


1.Create a S3 bucket, with no public access and upload files to the bucket & view the logs using cloudwatch for the uploaded files.

2.Launch two ec2-instances and connect it to a application load balancer, where the output traffic from the server must be an load balancer IP address

1.Create a S3 bucket, with no public access and upload files to the bucket & view the logs using cloudwatch for the uploaded files.

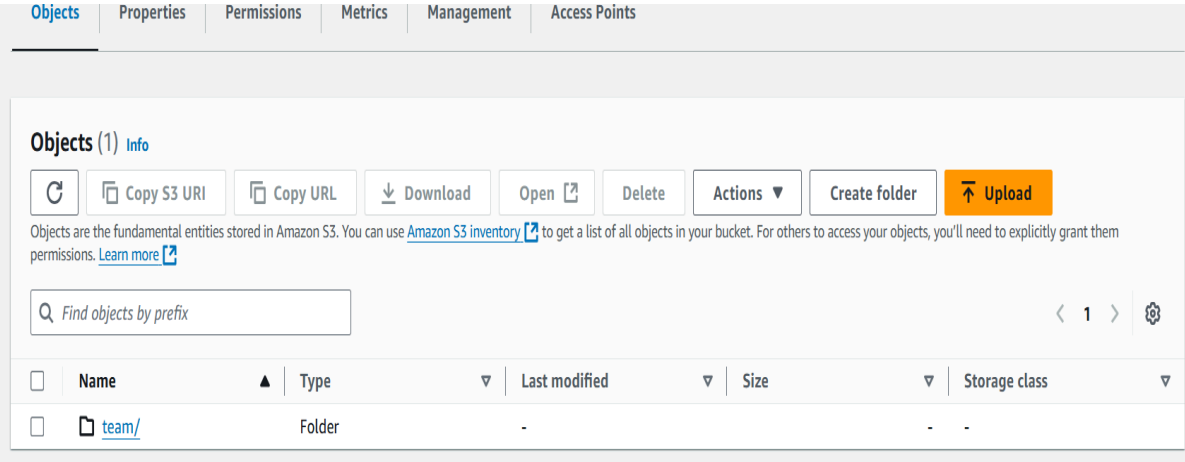
STEP 1- Create s3 bucket with no public access



The screenshot shows the Amazon S3 Buckets console. At the top, there's a navigation bar with 'Amazon S3' and 'Buckets'. Below it, a banner for 'Account snapshot' is visible. The main section is titled 'General purpose buckets (1)' and includes a search bar 'Find buckets by name'. A table lists the bucket 'solungamsm' in the 'Asia Pacific (Sydney) ap-southeast-2' region, created on 'November 2, 2024, 21:20:38 (UTC+05:30)'. Action buttons like 'Copy ARN', 'Empty', 'Delete', and 'Create bucket' are present.

Name	AWS Region	IAM Access Analyzer	Creation date
solungamsm	Asia Pacific (Sydney) ap-southeast-2	View analyzer for ap-southeast-2	November 2, 2024, 21:20:38 (UTC+05:30)

STEP 2 - create a prefix inside the bucket name as team



The screenshot shows the Amazon S3 Objects console for the bucket 'solungamsm'. The 'Objects' tab is selected, showing a table with one entry: a folder named 'team/'. The table headers include 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. Action buttons like 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload' are visible.

Name	Type	Last modified	Size	Storage class
team/	Folder	-	-	-

STEP 3 - inside the folder add images

Copy S3 URI

Copy URL

Download

Open

Delete

Actions



Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 >

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 1.png	png	November 2, 2024, 21:22:00 (UTC+05:30)	235.7 KB	Standard
<input type="checkbox"/>	 4.png	png	November 2, 2024, 21:22:00 (UTC+05:30)	329.3 KB	Standard

STEP 4 - bucket properties > server access logging > edit > *enable*

Create configuration

Server access logging

Edit

Log requests for access to your bucket. Use [CloudWatch](#) to check the health of your server access logging. [Learn more](#)

Server access logging

Enabled

Destination bucket

[s3://solungamsm](#)

Log object key format

[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]

AWS CloudTrail data events

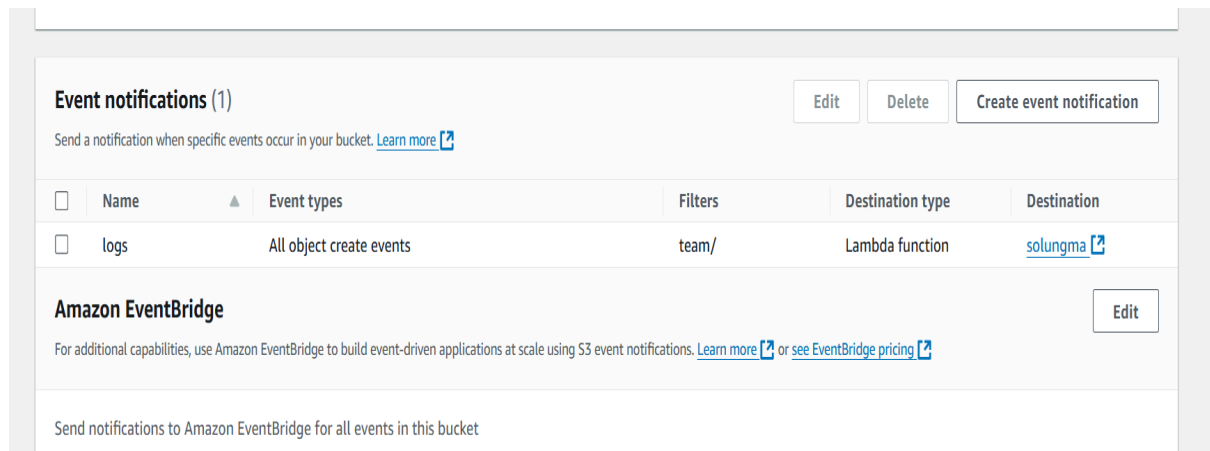
Info

Configure CloudTrail data events to log Amazon S3 object-level API operations in the CloudTrail console. [Learn more](#)

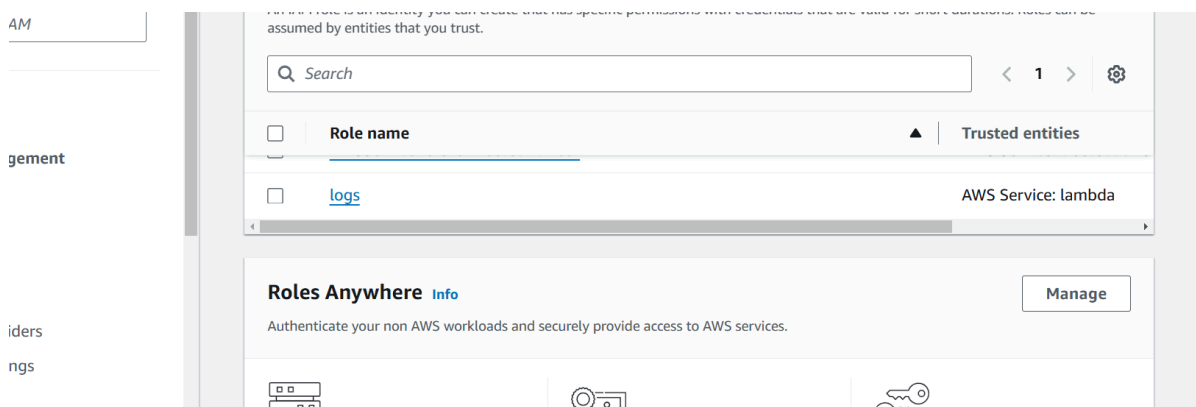
Configure in CloudTrail

Name	Access
------	--------

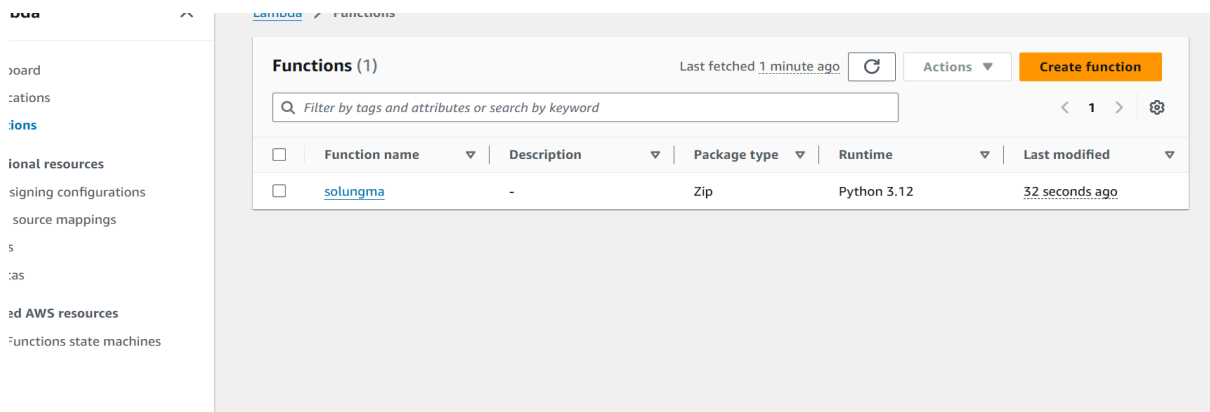
STEP 5 - bucket prop > event notifications > create event name *logs* > select event types as *all object create events*



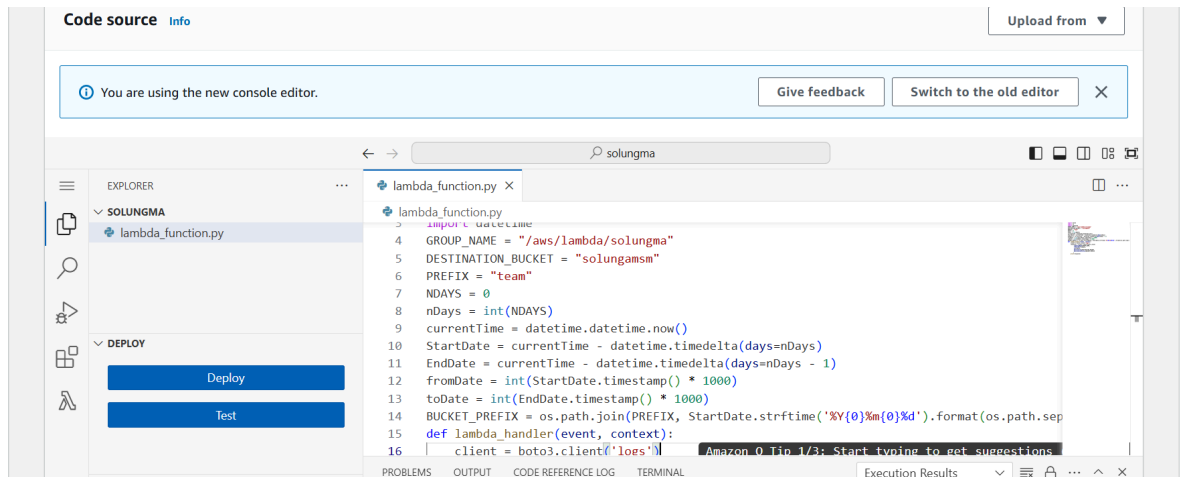
STEP 6 - create iam role name *logs* > attach policy *s3&cloudwatch full access* + *aws lambda execute*



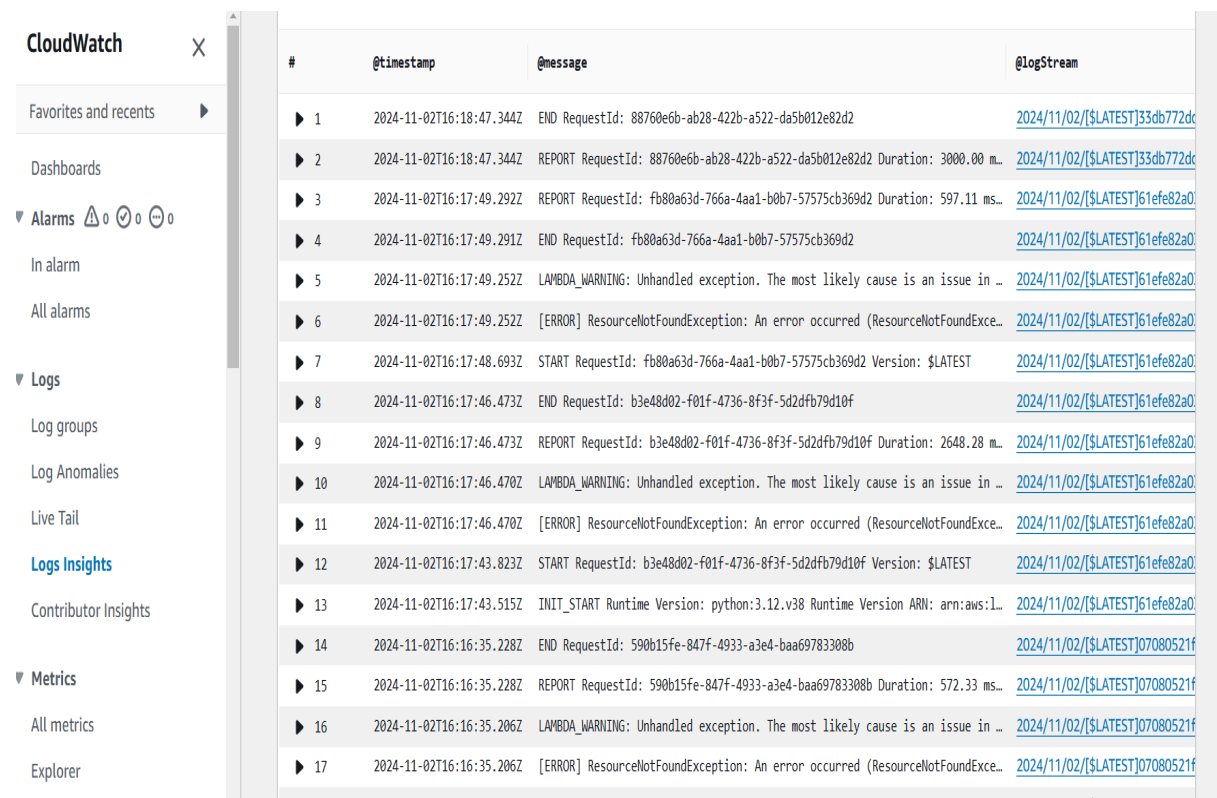
STEP 7 - create a lambda using python and name it as solungma also attach the created role



STEP 8 - write a lambda handler code inside ***lambda_function.py** file* followed by **create a test file** then add **bucket name, prefix name, log group name** inside lambda handler then **deploy a code** and test it.



STEP 9 - check the logs in cloudwatch for s3 bucket under log group or log insights



2.Launch two ec2-instances and connect it to a application load balancer, where the output traffic from the server must be an load balancer IP address

STEP 1 - launch 2 ec2 instances

EC2 > Instances > i-0369a7099558780f4

Instance summary for i-0369a7099558780f4 (class1) Info

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Instance ID i-0369a7099558780f4 (class1)	Public IPv4 address 54.79.141.162 open address	Private IPv4 addresses 12.0.1.37
IPv6 address -	Instance state Running	Public IPv4 DNS -
Hostname type IP name: ip-12-0-1-37.ap-southeast-2.compute.internal	Private IP DNS name (IPv4 only) ip-12-0-1-37.ap-southeast-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 54.79.141.162 [Public IP]	VPC ID vpc-0f148b889a3c9a21a (class) ↗	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-026562e8464500516 (class1) ↗	

EC2 > Instances > i-012b6fb93ccb35e8

Instance summary for i-012b6fb93ccb35e8 (class2) Info

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Instance ID i-012b6fb93ccb35e8 (class2)	Public IPv4 address 52.64.6.194 open address	Private IPv4 addresses 12.0.0.57
IPv6 address -	Instance state Running	Public IPv4 DNS -
Hostname type IP name: ip-12-0-0-57.ap-southeast-2.compute.internal	Private IP DNS name (IPv4 only) ip-12-0-0-57.ap-southeast-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 52.64.6.194 [Public IP]	VPC ID vpc-0f148b889a3c9a21a (class) ↗	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-00f87d718a61b8435 (class2) ↗	

- 2 ec2 instance launched and connect to vpc,sunet,security grp inbound with os and aml

STEP 2 - Connect it to the Application load balancer

class

Actions ▾

▼ Details

Load balancer type Application	Status ✔ Active	VPC vpc-0f148b889a3c9a21a	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z1GM3OXH4ZPM65	Availability Zones subnet-00f87d718a61b8435 ap-southeast-2b (apse2-az1) subnet-026562e8464500516 ap-southeast-2a (apse2-az3)	Date created October 21, 2024, 13:16 (UTC+05:30)
Load balancer ARN <code>arn:aws:elasticloadbalancing:ap-southeast-2:762233750461:loadbalancer/app/class/6008ca406d3c9988</code>		DNS name Info <code>class-1761534347.ap-southeast-2.elb.amazonaws.com</code> (A Record)	

- Vpc created
- Internet gateway created
- Attached vpc and internet gate way
- 2 subnet created
- Each instance connect to each subnet
- 2 route tables created and each subnet and internet gate way
- Target group created and connected to subnet
- Load balancer created and connected to target group
- 2 ec2 instance launched and connect to vpc,subnet,security grp inbound with os and aml
- 2 instances same webserver installed
- Instance and loadbalncer should be same inbound principles

NOTE : The above steps are explained detail in last task

STEP 3

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Load balancer ip

: <http://source-836852145.ap-southeast-2.elb.amazonaws.com/>