

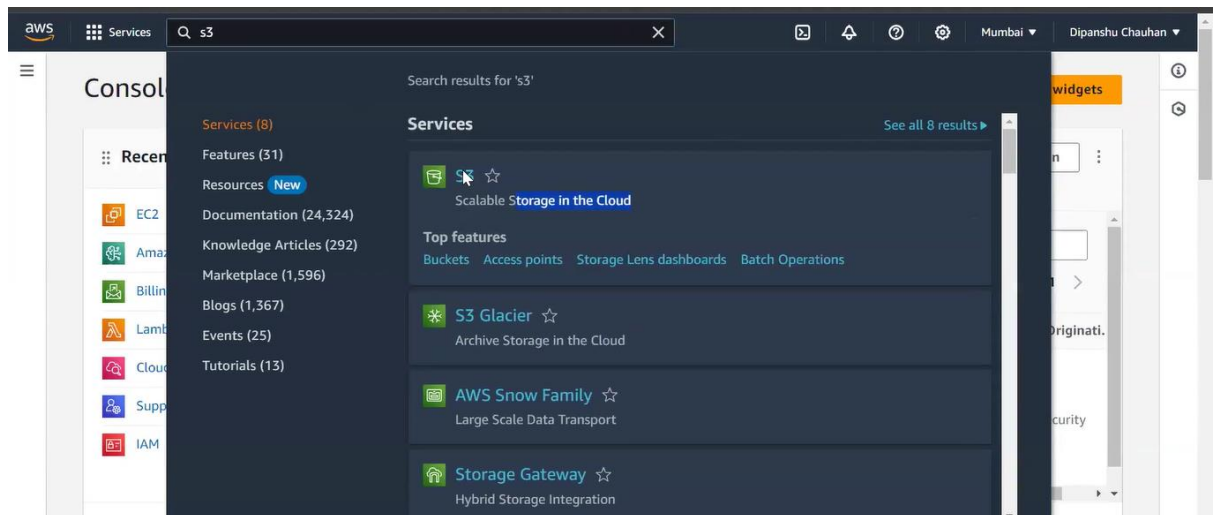


## AWS Session 08

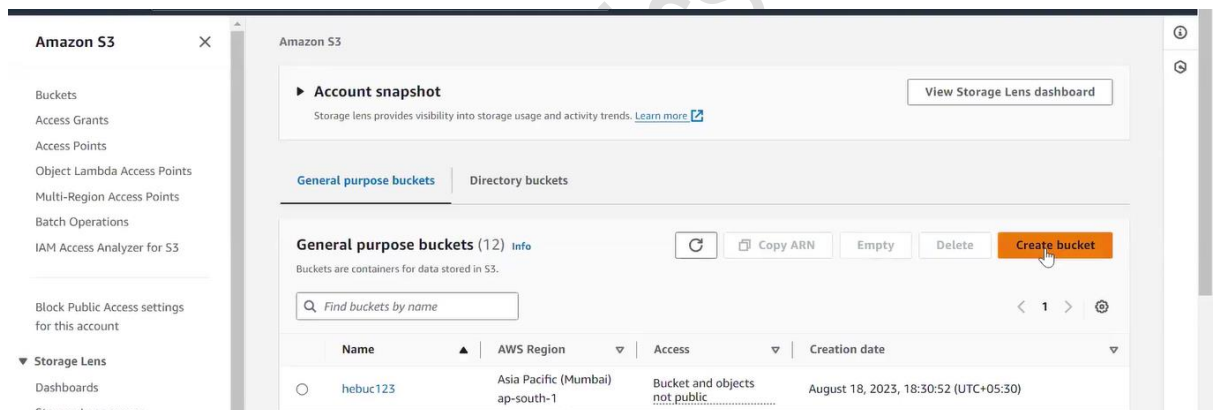
### Summary 22-03-2024

- To store any data permanently, we need a storage device for that. Storage devices are pen drive, hard disk etc.
- There is no direct way to store the data in the hard-disk, first we have to keep the data in a file and for creating a file we need to create a directory and again to keep the directory inside the hard-disk we first have to create partition in it.
- Before creating the partitions, we must first format the storage.
- Now the point here is for storing any data we must fulfil all the above requirements.
- If we want this storage management to be done by the cloud providers then this concept is known as the Storage as a Service.
- All the minimum requirements that are needed to store the data will be fulfilled by the cloud provider and all we need to do is just store our data.
- S3 is a serverless service provided by the AWS and it is storage service which we can use to storage data of any size.
- Usually, we have three types of use cases of storages.
  - For storing the objects, we have the object storage (S3 Service)
  - For block storage we have the EBS Service.
  - For file storage we have the EFS Service

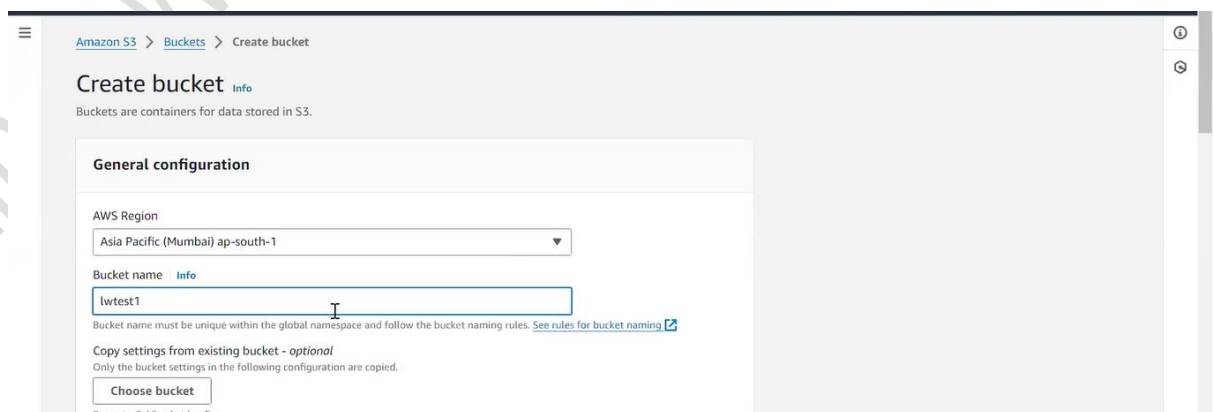
- Using the S3 service
  - Go to the S3 service in the AWS Console.



- To upload anything we have to first create the Bucket, bucket is like a folder.

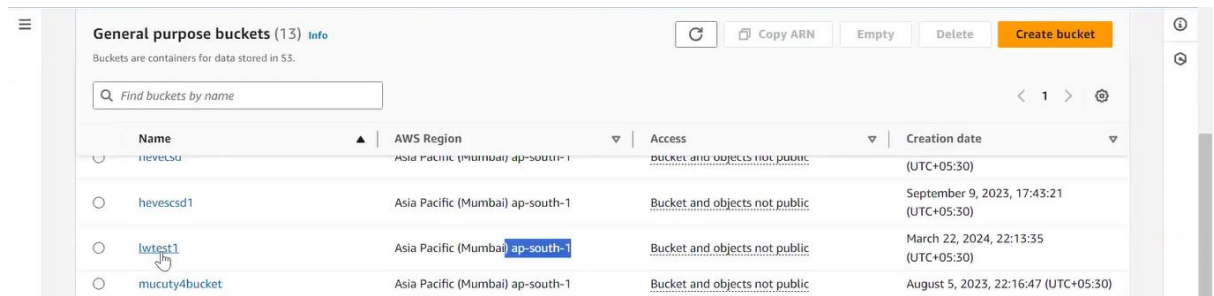


- Configure the bucket by selecting the region and the bucket name.

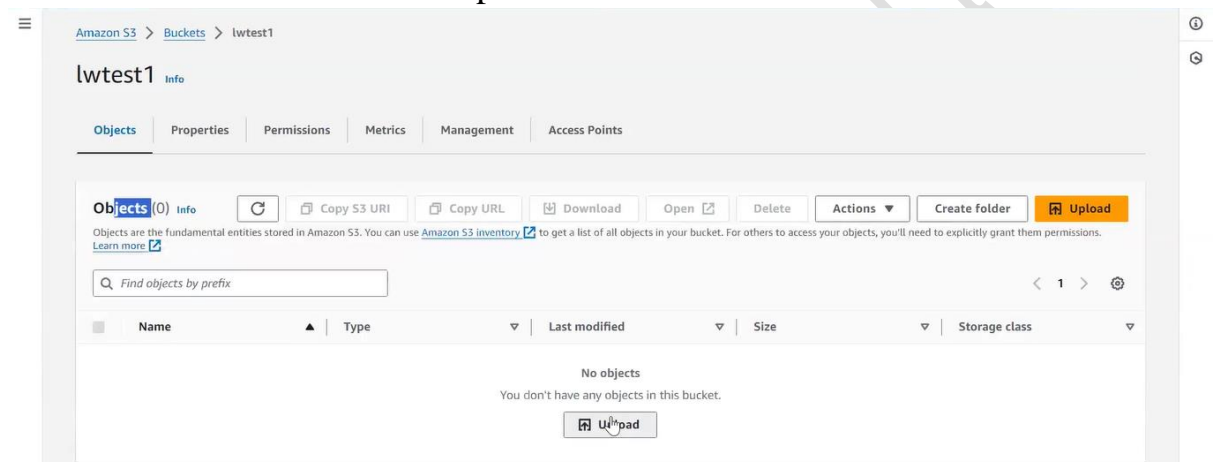


- There is one thing to keep in mind that the bucket name should be unique globally.

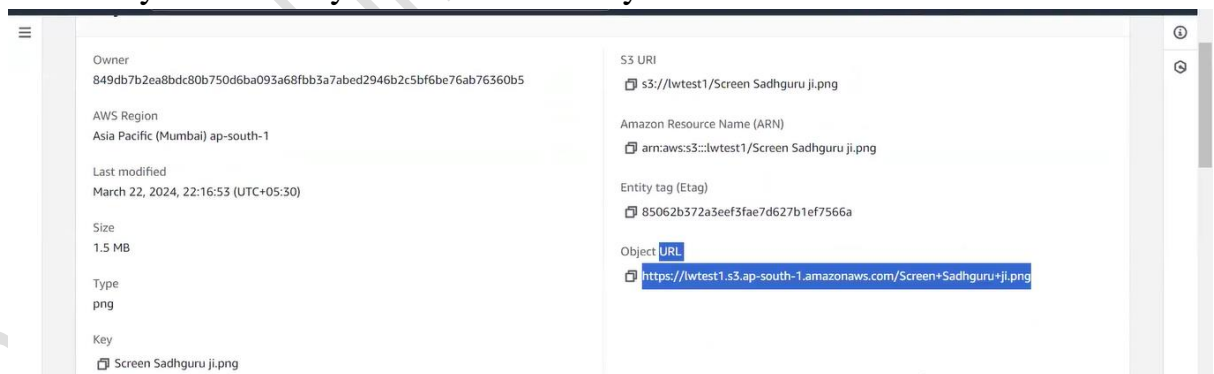
- Now go to the bucket we have created.



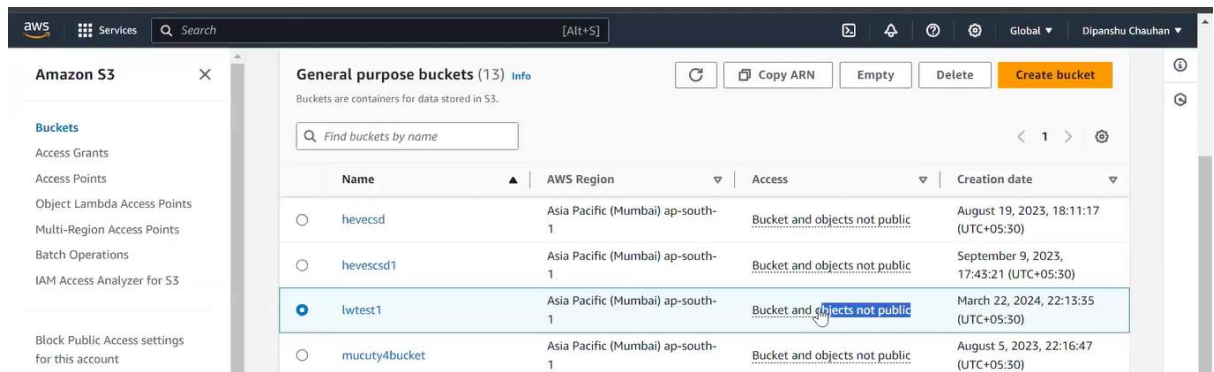
- To upload anything in the bucket, click on the upload button and select the file to be uploaded.



- This object is sharable which means we can share the url to anyone and they can access it easily.

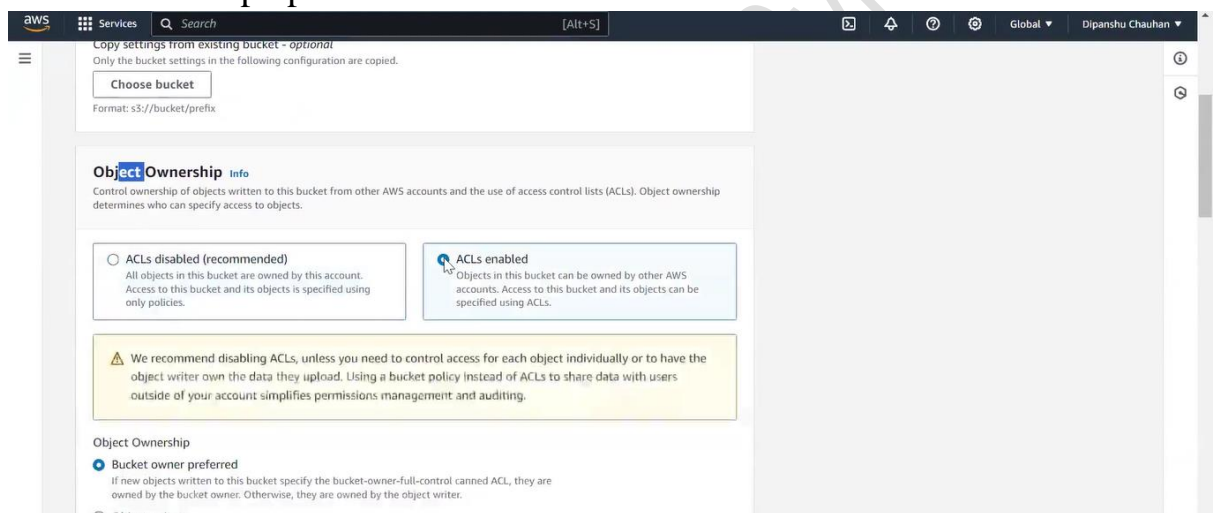


- By default, the bucket and the objects are not public due to some security reasons.

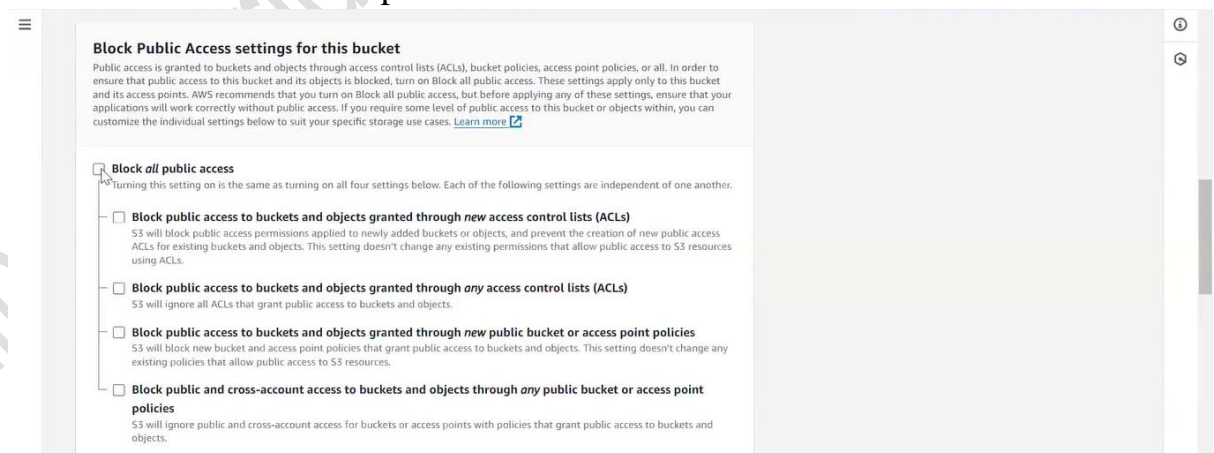


- Let's create a new bucket that have a public access.

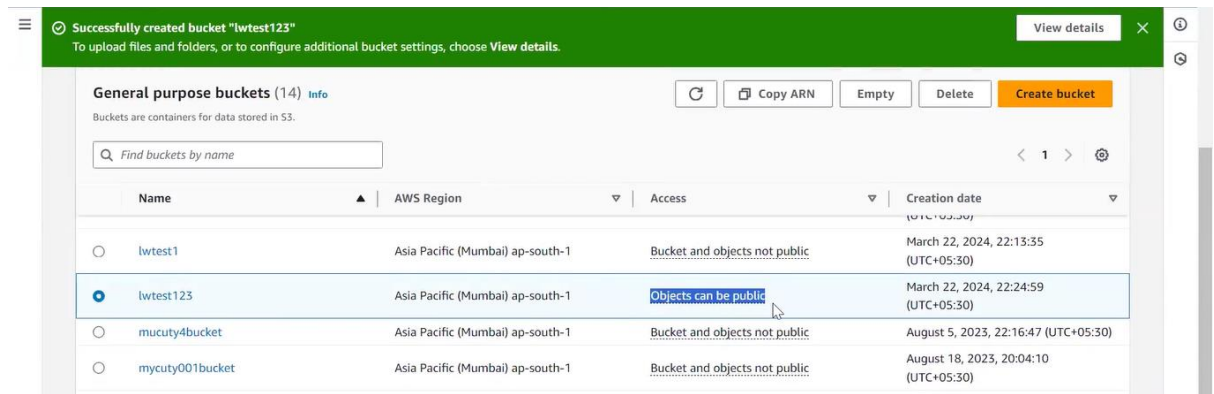
- While creating the bucket and configuring it, in the object ownership option enable the ACL.



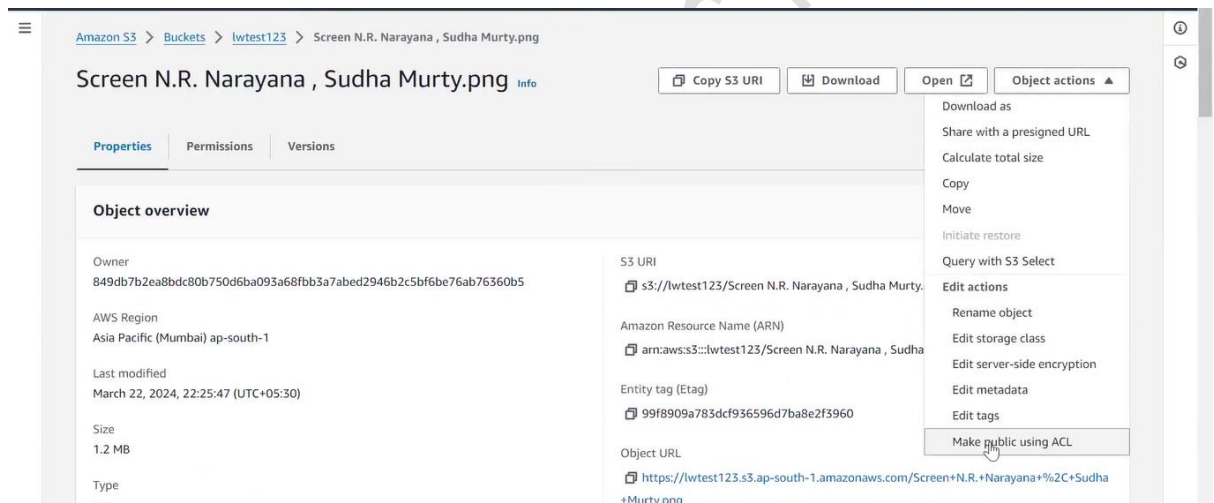
- Also disable this option.



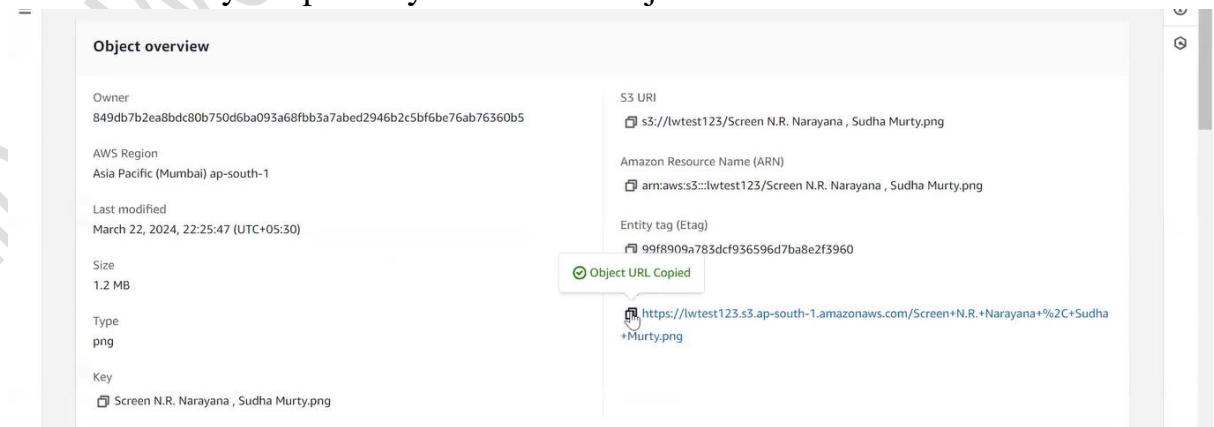
- Now we can see that the object and the bucket have the public access which means anyone in the world can access that object or the data.



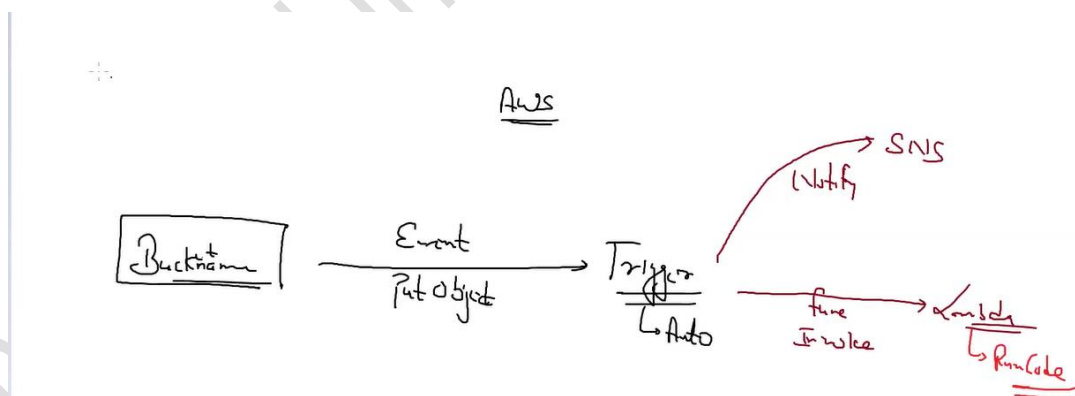
- Again there might be a chance that the object will deny the public access.
- For that we have to make some changes in the object permissions.
- Click on the object actions and select "make public using ACL".



- Now anyone publicly can use the object URL to access it.

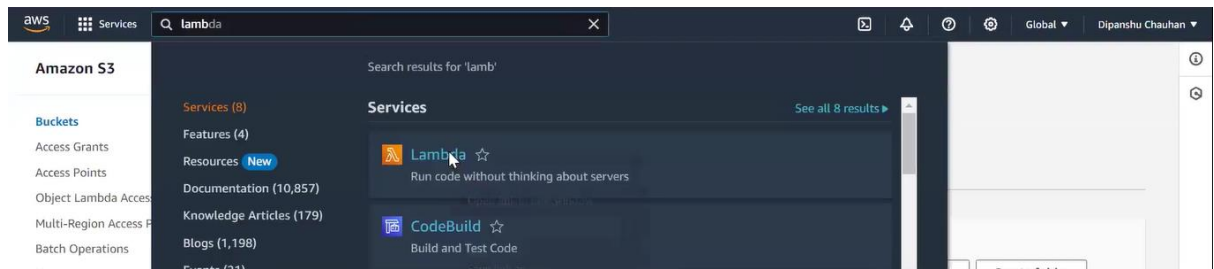


- S3 service is so popular because of its durability, it guarantees the durability of eleven 9's which is close to the 100 percent. This means that the data we are storing is very safe as compared to our physical hard disk.
- Whenever we do anything in any service, AWS keep on track of everything this is a known **event**.
- Here we are integrating S3, Lambda function, and SNS serverless services. Through all these services integration we are building a project that is based on **event-driven architecture**.
- AWS keeps the eye on all the events happening in any service. Example of the events may be put, delete of object in the S3.
- We can create an event driven architecture in which we will integrate the S3, Lambda, SNS services.
- Whenever any event occurs in the bucket it will trigger something, maybe it is sending the notification about the event.

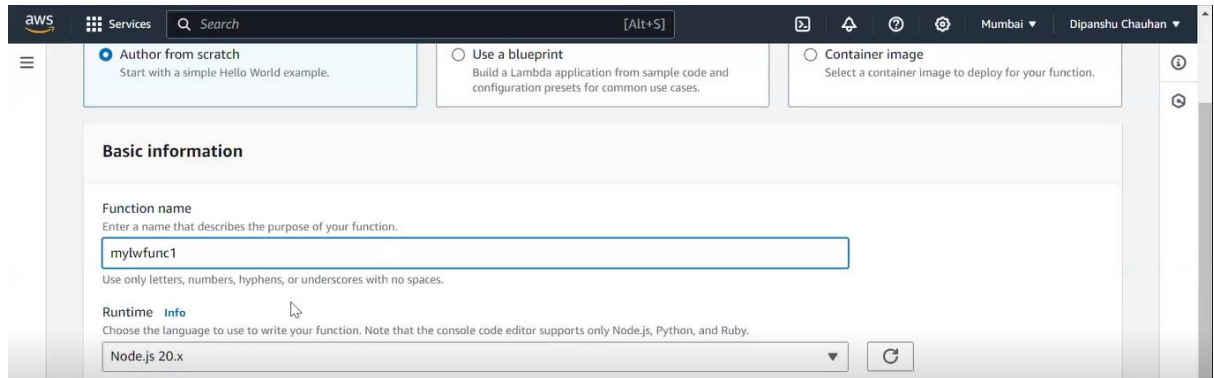


- Creating the Event driven architecture.
  - We have to tell what we want whenever any event occurs.
  - Our requirement is whenever any event occur in the bucket, we must be notified about that.
  - First setup the lambda so that we can trigger anything.

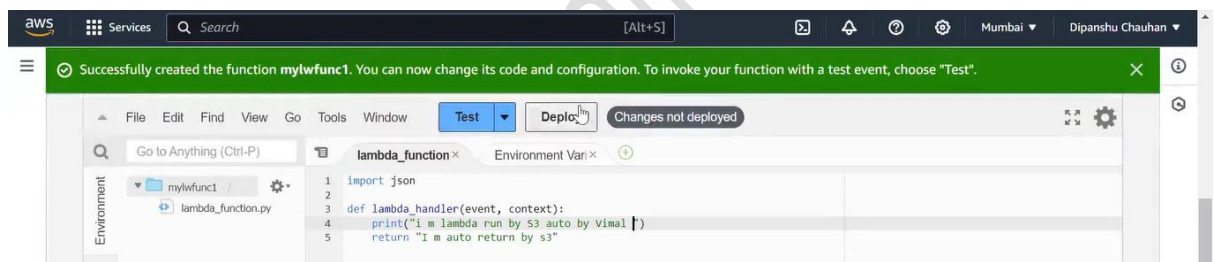




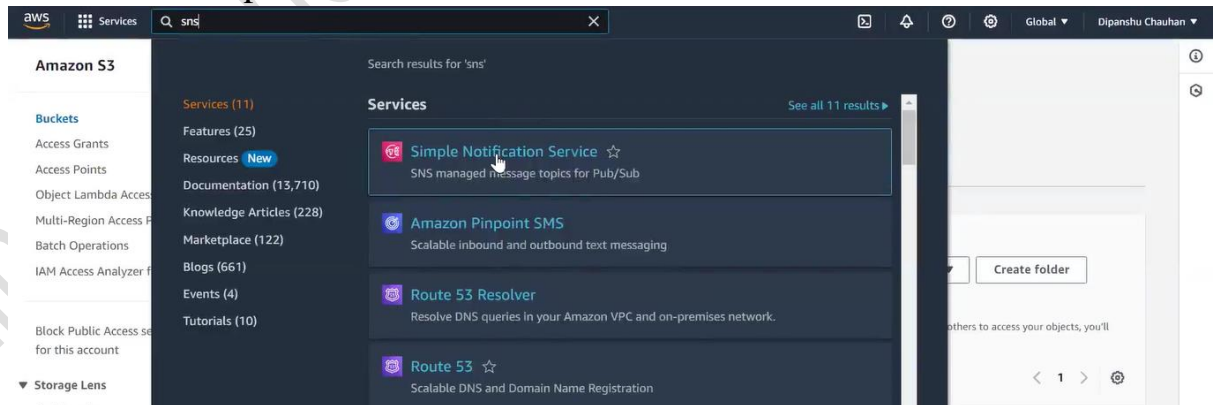
➤ Create the lambda function.



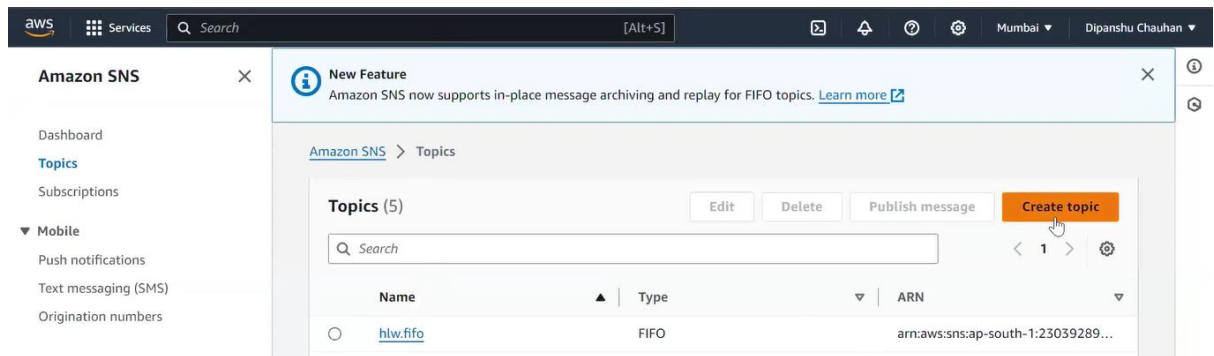
➤ After creating the lambda function, write this code and deploy it.



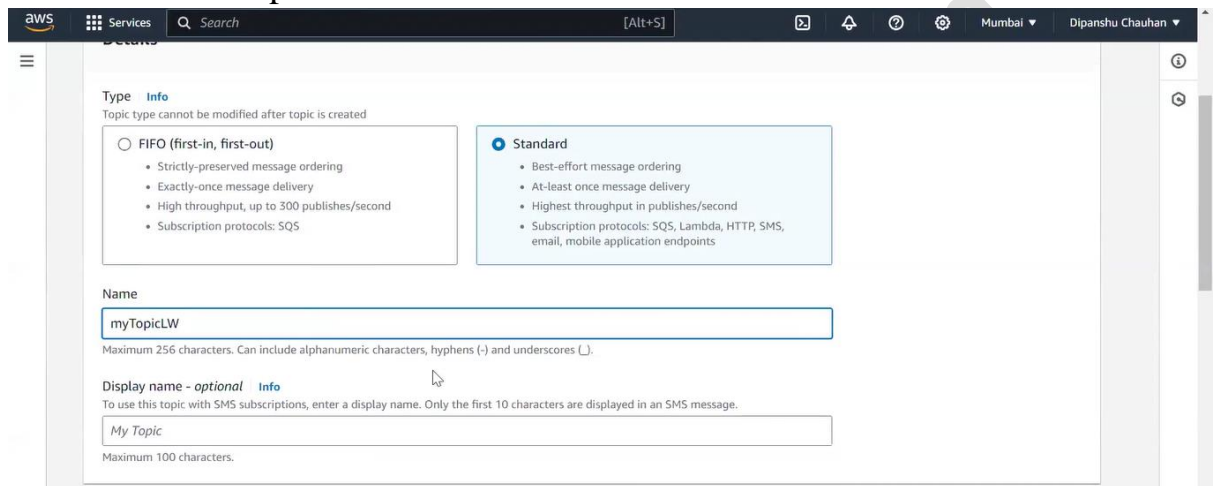
➤ Also setup the SNS service.



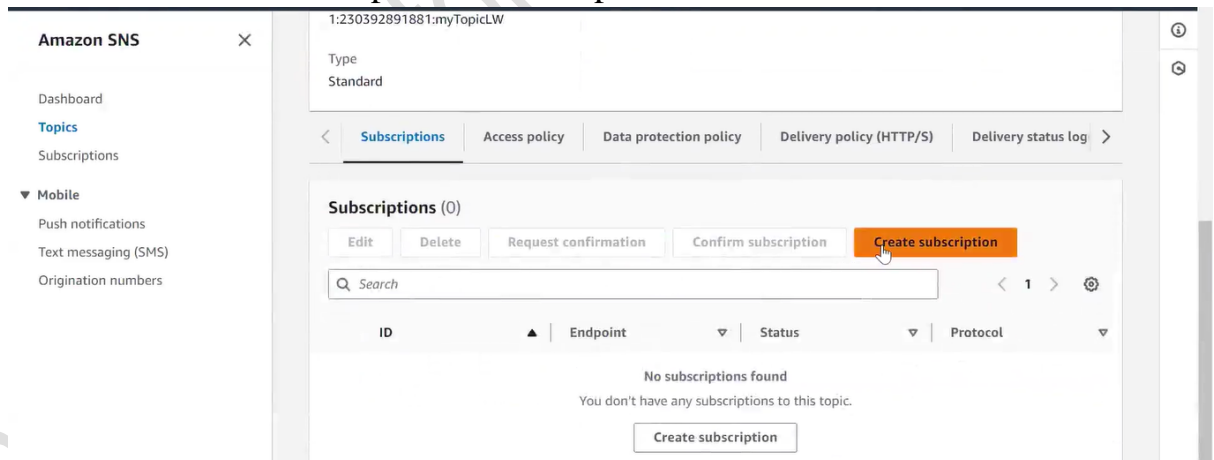
➤ Create the new SNS topic.



➤ Give the topic a name.



➤ Create the subscription for the topic.



➤ Here we have to give the email address on which we want to receive the notification.



The type of endpoint to subscribe  
Email

Endpoint  
An email address that can receive notifications from Amazon SNS.  
vimal.linuxworld@gmail.com

After your subscription is created, you must confirm it. [Info](#)

► **Subscription filter policy - optional** [Info](#)  
This policy filters the messages that a subscriber receives.

► **Redrive policy (dead-letter queue) - optional** [Info](#)  
Send undeliverable messages to a dead-letter queue.

Cancel **Create subscription**

- Now we want the S3 to invoke the lambda function and the SNS service.
- Go to the properties of the bucket.

Amazon S3 > Buckets > lwtest123

lwtest123 [Info](#)

Objects Properties Permissions Metrics Management Access Points

Objects (1) [Info](#)

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](#)

- In the properties we have the option for Event notification.

Configure in CloudTrail

Event notifications [Edit](#) [Delete](#) [Create event notification](#)

Send a notification when specific events occur in your bucket. [Learn more](#)

Name	Event types	Filters	Destination type	Destination
No event notifications				
Choose <a href="#">Create event notification</a> to be notified when a <b>specific event</b> occurs.				

[Create event notification](#)

- Give this event a name.

Amazon S3 > Buckets > lwtest123 > Create event notification

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  
myEventNameS3  
Event name can contain up to 255 characters.

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

➤ Select the event type here.

The screenshot shows the AWS SNS console with the search bar set to 'sns'. The 'Event types' section is active, showing a list of event types under 'Object creation'. The 'Put' event type is selected, with the event name 's3:ObjectCreated:Put'. Other event types listed include 'Post', 'Copy', and 'Multipart upload completed'. The 'Object removal' section is also visible but not selected.

➤ Select the Lambda function in the destination.

The screenshot shows the AWS SNS console with the search bar set to 'sns'. The 'Destination' section is active, showing the 'Lambda function' option selected. Below it, the 'Specify Lambda function' section shows 'Choose from your Lambda functions' selected. A dropdown menu shows 'mylwfnc1'. A red error box is displayed with the message 'Unknown Error: An unexpected error occurred.' and a link to 'API response'. The 'Save changes' button is highlighted in orange.

➤ Now as soon as we upload or put anything in the S3 bucket it will trigger the lambda function.

The screenshot shows the AWS S3 console with the search bar set to 'sns'. The 'Amazon S3' > 'Buckets' > 'lwtest123' path is shown. The bucket 'lwtest123' is selected, and the 'Objects' tab is active. The 'Objects (1)' section shows a list of objects. The 'Upload' button is highlighted in orange.

➤ We can see the details in the cloud watch logs of the lambda function.

The screenshot displays the AWS CloudWatch console. The top navigation bar includes the AWS logo, a search bar, and user information (Mumbai, Dipanshu Chauhan). The main interface is divided into tabs: Code, Test, Monitor (selected), Configuration, Aliases, and Versions. The Monitor tab shows a 'Monitor' section with buttons for 'View CloudWatch logs', 'View X-Ray traces', and 'View Lambda Insights'. Below this, there's a section for 'CloudWatch metrics' with a filter dropdown set to 'Function'. The left sidebar contains a 'CloudWatch' section with links to 'Log groups', 'Log Anomalies', 'Live Tail', 'Logs Insights', 'Metrics', 'X-Ray traces', 'Events', 'Application Signals', and 'Network monitoring'. The main content area shows the 'Log events' for a specific log group: '2024/03/22/[\$LATEST]fa27b079103f418cbbd7823c59d79c1e'. It includes a search bar, filters for '1m', '1h', and 'Local timezone', and a 'Display' dropdown. The log events are listed in a table with columns for 'Timestamp' and 'Message'. The messages include 'INIT\_START Runtime Version: python:3.12.v21 Runtime Version ARN: arn:aws:lambda:ap-sou...', 'START RequestId: 8b53e034-5ca9-46c4-bfd1-2f4c0cbc53b3 Version: \$LATEST', 'i m lambda run by S3 auto by Vimal', 'END RequestId: 8b53e034-5ca9-46c4-bfd1-2f4c0cbc53b3', and 'REPORT RequestId: 8b53e034-5ca9-46c4-bfd1-2f4c0cbc53b3 Duration: 1.96 ms Billed Durati...'. A 'Copy' button is visible next to the log entry.

aws Services Search [Alt+S] Mumbai Dipanshu Chauhan

Code Test **Monitor** Configuration Aliases Versions

**Monitor** Info View CloudWatch logs View X-Ray traces View Lambda Insights

Filter metrics by Function

Alarm recommendations 3h 1d 1w UTC timezone

**CloudWatch metrics**

**CloudWatch** X Favorites and recents Dashboards Alarms 0 0 0 Logs Log groups Log Anomalies Live Tail Logs Insights Metrics X-Ray traces Events Application Signals Network monitoring

CloudWatch > Log groups > /aws/lambda/mylwfunc1 > 2024/03/22/[\$LATEST]fa27b079103f418cbbd7823c59d79c1e

**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 1m 1h Local timezone Display

Timestamp	Message
	No older events at this moment. <a href="#">Retry</a>
2024-03-22T23:07:39.128+05:30	INIT_START Runtime Version: python:3.12.v21 Runtime Version ARN: arn:aws:lambda:ap-sou...
2024-03-22T23:07:39.217+05:30	START RequestId: 8b53e034-5ca9-46c4-bfd1-2f4c0cbc53b3 Version: \$LATEST
2024-03-22T23:07:39.217+05:30	i m lambda run by S3 auto by Vimal
	i m lambda run by S3 auto by Vimal
2024-03-22T23:07:39.219+05:30	END RequestId: 8b53e034-5ca9-46c4-bfd1-2f4c0cbc53b3
2024-03-22T23:07:39.219+05:30	REPORT RequestId: 8b53e034-5ca9-46c4-bfd1-2f4c0cbc53b3 Duration: 1.96 ms Billed Durati...

Copy