

Aarya Arban

T11 - 05

Assignment No. 2

Aim: To build an application using S3 Bucket in AWS, and implementing lambda function.

Theory:

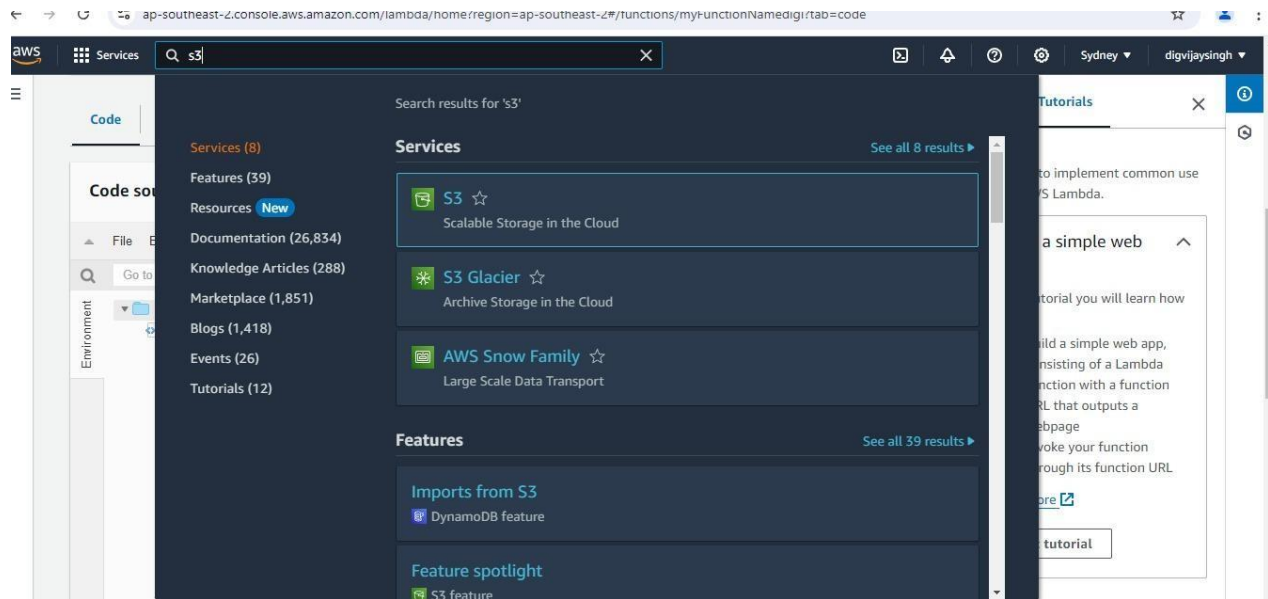
A bucket is a container for objects. To store your data in Amazon S3, you first create a bucket and specify a bucket name and AWS Region. Then, you upload your data to that bucket as objects in Amazon S3. Each object has a key (or key name), which is the unique identifier for the object within the bucket.

AWS Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying computer resources for you.

Steps:

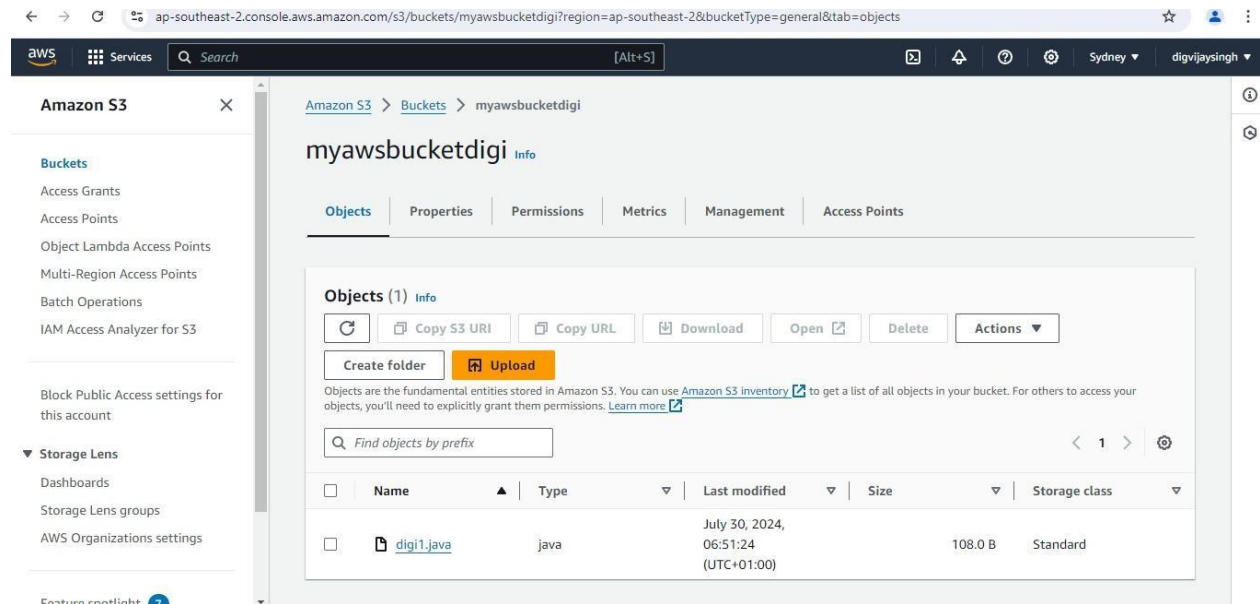
Login to AWS account.

Search S3 and click on the option.



S3 dashboard is opened.

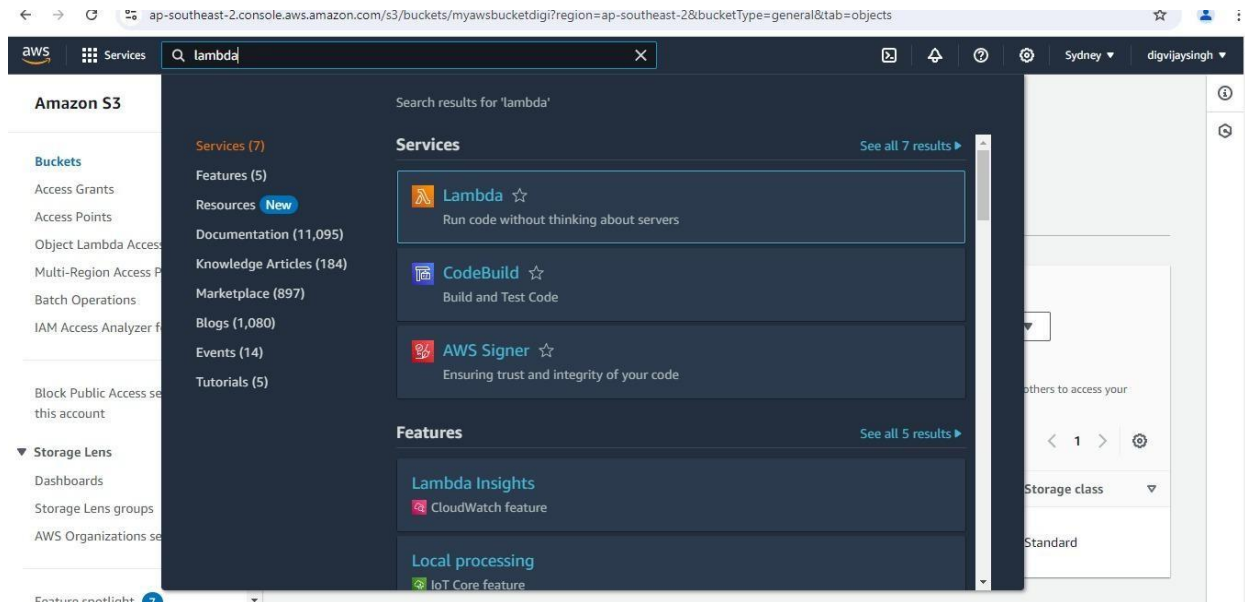
Create a bucket and give it a name.



Click on the 'upload' button.

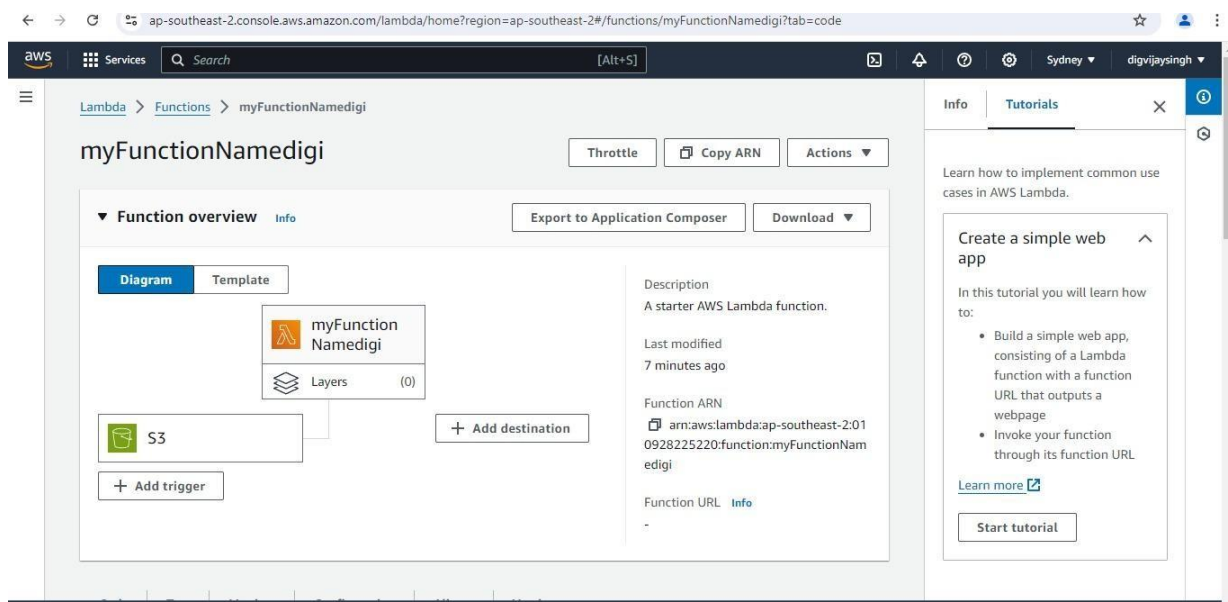
Add any .py or .java extension file and click on upload.

Search Lambda.

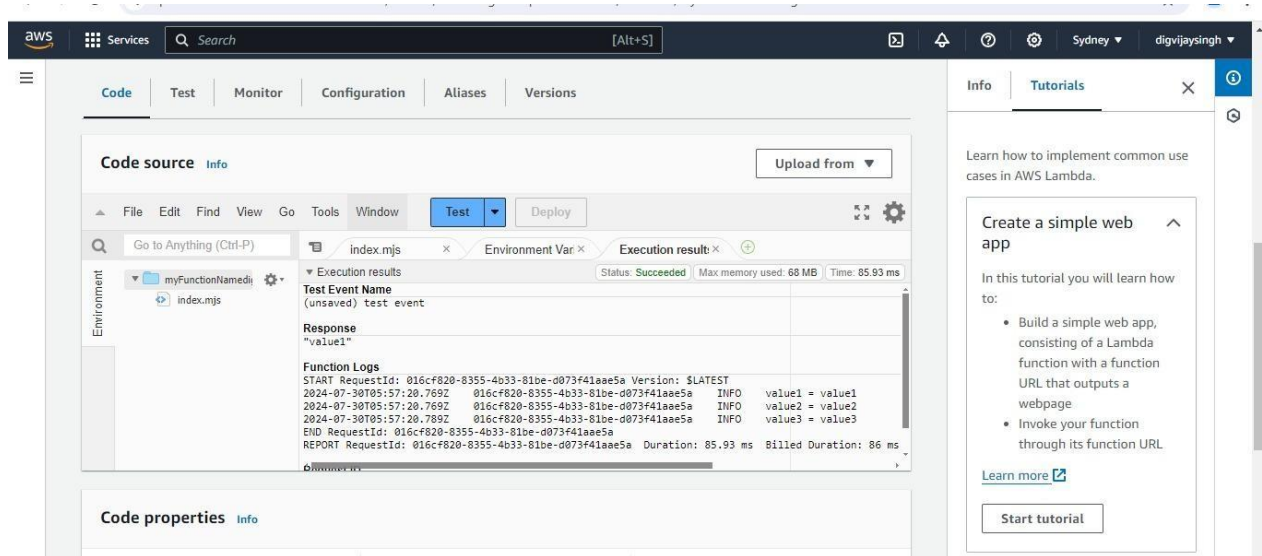


Select the created bucket and create trigger.

Create a S3 trigger.

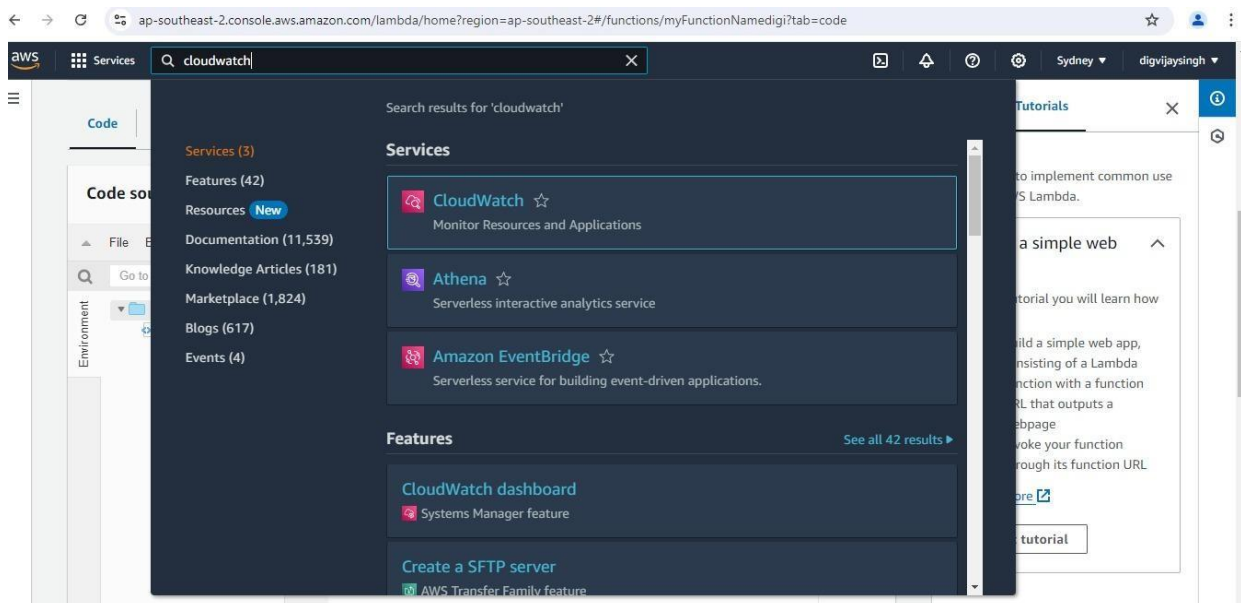


Click on the test button to test the code.



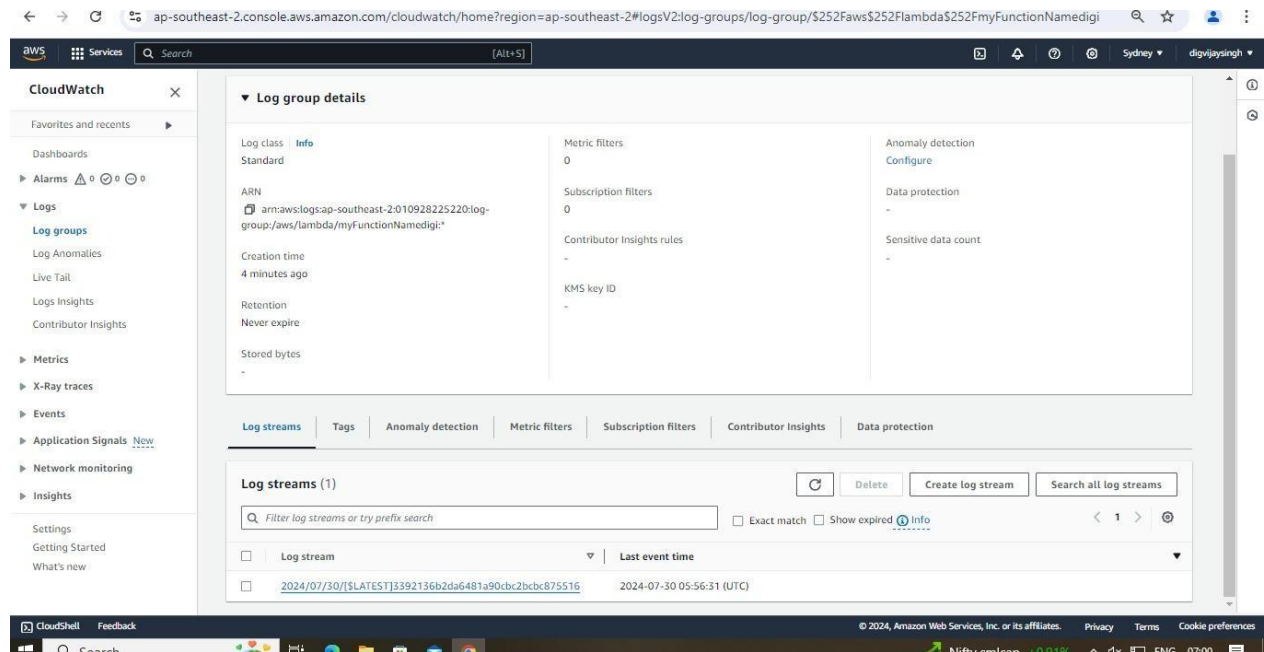
Execution status: **successful**

Now, search CloudWatch.

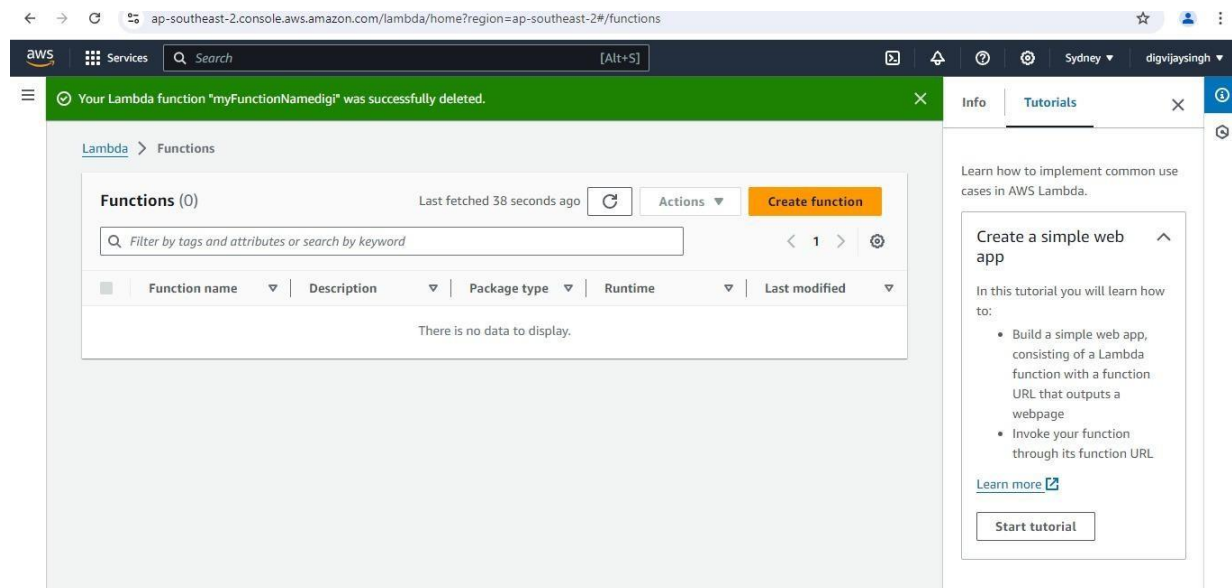


Go to Logs > Log groups.

Check the log group details and the log streams.



Click on delete function.



Empty the bucket.

The screenshot shows the AWS console interface for the 'Empty bucket' action. The header includes the AWS logo, 'Services' menu, a search bar, and navigation icons. The main content area is titled 'Empty bucket' with an 'Info' link. A yellow warning box contains the following text: 'Emptying the bucket deletes all objects in the bucket and cannot be undone.', 'Objects added to the bucket while the empty bucket action is in progress might be deleted.', and 'To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.' Below this is a 'Learn more' link. A blue information box states: 'If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket.' with a 'Go to lifecycle rule configuration' button and a 'Learn more' link. The main section is titled 'Permanently delete all objects in bucket "myawsbucketdigi"?'. It includes a text input field with the placeholder 'To confirm deletion, type *permanently delete* in the text input field.' and the text 'permanently delete' entered. At the bottom are 'Cancel' and 'Empty' buttons.

Finally, delete the bucket.

The screenshot shows the AWS console interface for the 'Delete bucket' action. The header is identical to the previous screenshot. The breadcrumb trail is 'Amazon S3 > Buckets > myawsbucketdigi > Delete bucket'. The main content area is titled 'Delete bucket' with an 'Info' link. A yellow warning box contains the following text: 'Deleting a bucket cannot be undone.', 'Bucket names are unique. If you delete a bucket, another AWS user can use the name.', 'If this bucket is used with a Multi-Region Access Point in an external account, initiate failover before deleting the bucket.', and 'If this bucket is used with an access point in an external account, the requests made through those access points will fail after you delete this bucket.' Below this is a 'Learn more' link. The main section is titled 'Delete bucket "myawsbucketdigi"?'. It includes a text input field with the placeholder 'To confirm deletion, enter the name of the bucket in the text input field.' and the text 'myawsbucketdigi' entered. At the bottom are 'Cancel' and 'Delete bucket' buttons.

Conclusion: Hence, created an application using s3 bucket and implemented it using a lambda function.

LO MAP: LO1, LO2