cloud

computing

PROJECT



Project-1

SERVERLESS IMAGE PROCESSING

Here’s a step-by-step guide to create a serverless image processing application using AWS Lambda and Amazon S3:

Create an S3 Bucket:

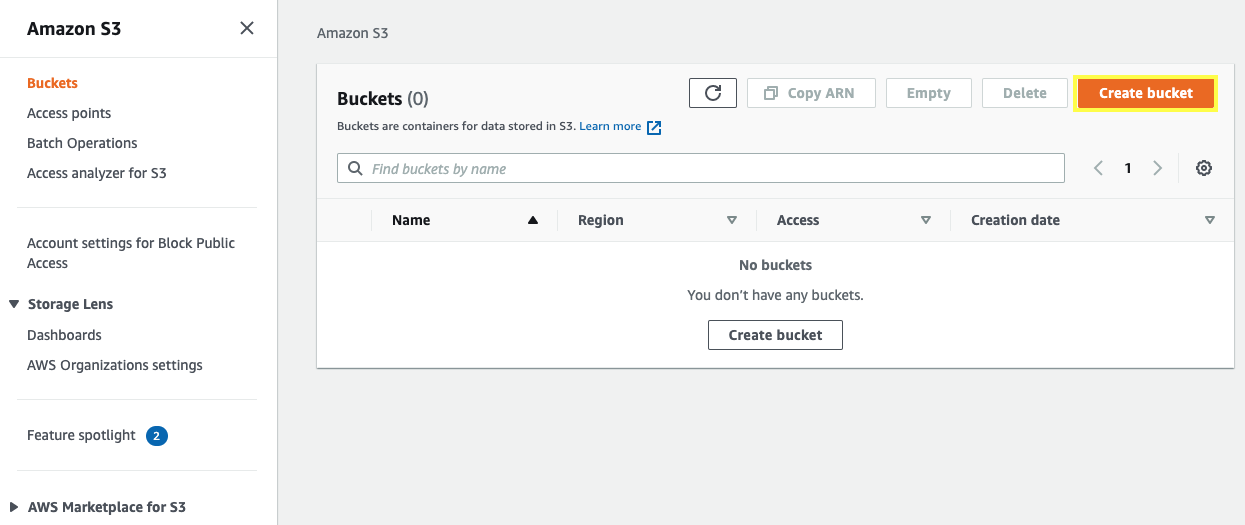
->Sign in to the AWS Management Console and open the Amazon S3 console.

->Choose ‘Create bucket’.

->In the ‘Bucket name’ field, type a unique DNS-compliant name for your new bucket.

->In the ‘Region’ field, choose the AWS Region where you want the bucket to reside.

->Choose ‘Create’.



Create an IAM Role for Lambda:

->Open the IAM console.

->In the navigation pane, choose ‘Roles’, then ‘Create role’.

->In the ‘Choose the service that will use this role’ section, choose ‘Lambda’.

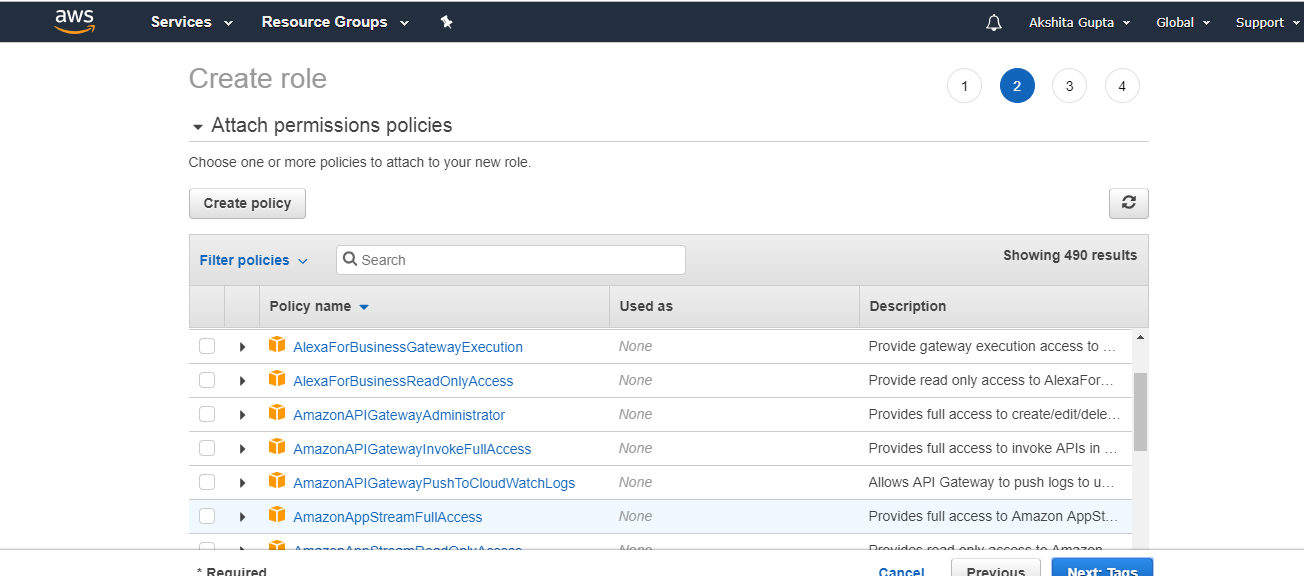
->Choose ‘Next: Permissions’.

->In the ‘Attach permissions policies’ section, choose ‘AWSLambdaExecute’. This managed policy has permissions for Amazon S3 and CloudWatch Logs.

->Choose ‘Next: Tags’.

->Choose ‘Next: Review’.

->For ‘Role name’, type a name for your role, such as ‘lambda-s3-role’, then choose ‘Create role’.



Create a Lambda Function:

->Open the AWS Lambda console.

->Choose ‘Author from scratch’.

->In the ‘Basic information’ section, do the following:

->For ‘Function name’, type a name.

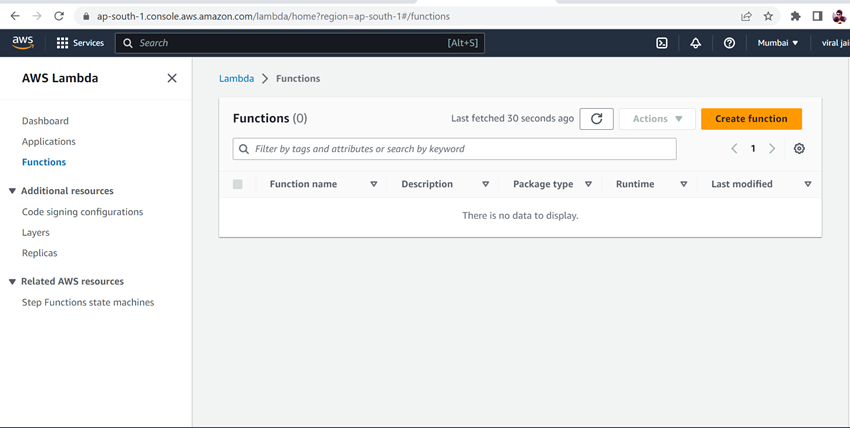
->For ‘Runtime’, choose ‘Node.js’.

->Choose ‘Change default execution role’ to specify the IAM role you created.

->Choose ‘Create function’.

->Add the Image Processing Code:

->In the AWS Lambda console, in the ‘Function code’ section, you can add your image processing code. You can use libraries like sharp for resizing and optimizing images in Node.js.



Set Up the Trigger:

->In the AWS Lambda console, in the designer section, choose ‘Add trigger’.

->Choose ‘S3’ from the dropdown.

->Configure the trigger with the bucket name and event type (like ‘PUT’). This means the Lambda function will be triggered whenever a new image is uploaded to the S3 bucket.

Test the Setup:

Upload an image to your S3 bucket and check if the Lambda function is triggered and the image is processed as expected.

