

conienis

01

about us

02

history

03

services

04

team

05

portfolio

06

contact

Intro:

Serverless image processing refers to the practice of using serverless computing resources to handle tasks related to manipulating, transforming, and managing images. This approach leverages cloud computing services, such as AWS Lambda, Google Cloud Functions, or Azure Functions, where you can run code.



Benefits of Serverless Image Processing:

- **Cost-Effectiveness:** You only pay for the compute time used, without incurring costs during idle periods.
- **Scalability:** Serverless platforms automatically scale based on the incoming workload, ensuring efficient handling of image processing tasks, even during peak times.
- **Simplicity:** Developers can focus on writing code rather than managing infrastructure, as serverless platforms handle server provisioning, scaling, and maintenance.
- **Flexibility:** Serverless functions can be triggered by various events (HTTP requests, object storage events, etc.), allowing for flexible integration into different workflows.

Components:

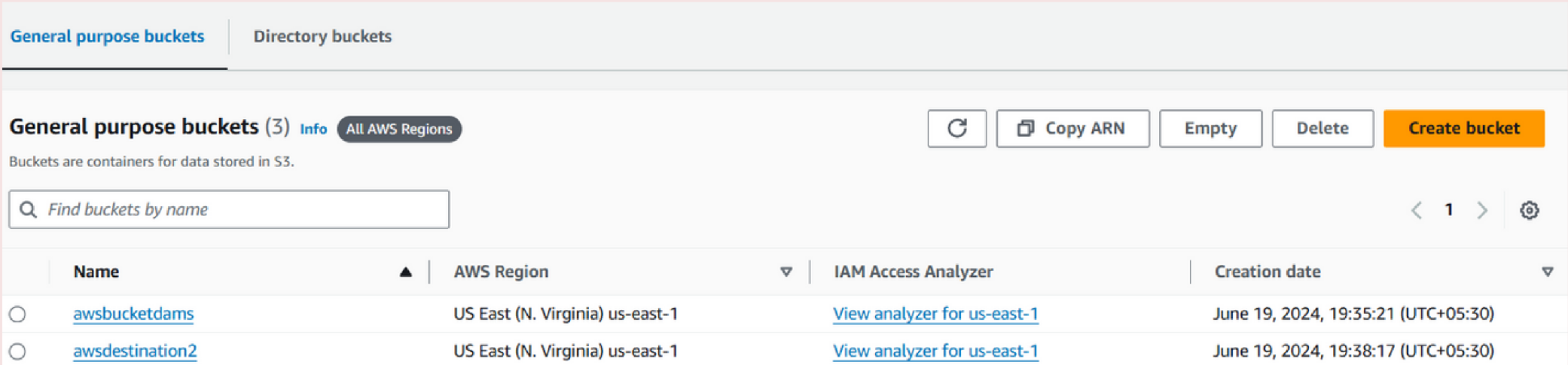
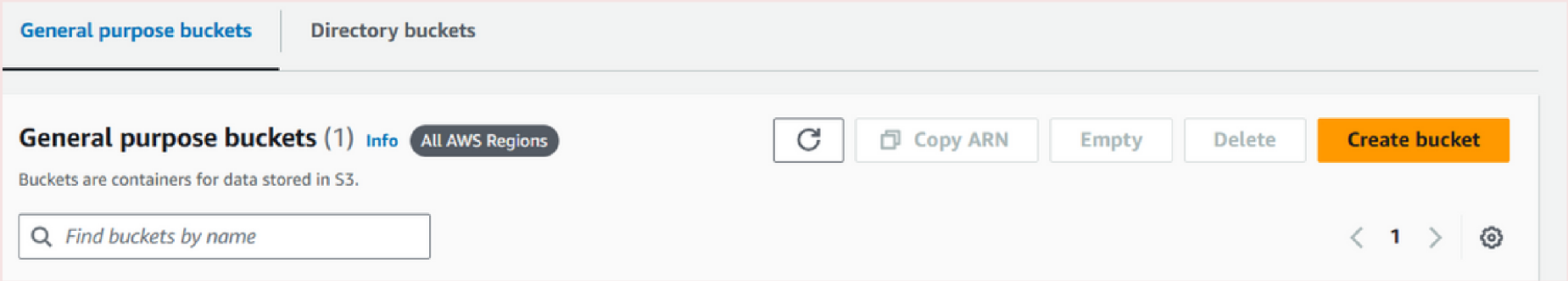
1. Amazon S3 Bucket
- 2: AWS lambda
- 3: Amazon API Gateway
- 4: Amazon S3 Event Notification

Step 1:

login to console aws cloud service.

Step 2:

Create S3 bucket .

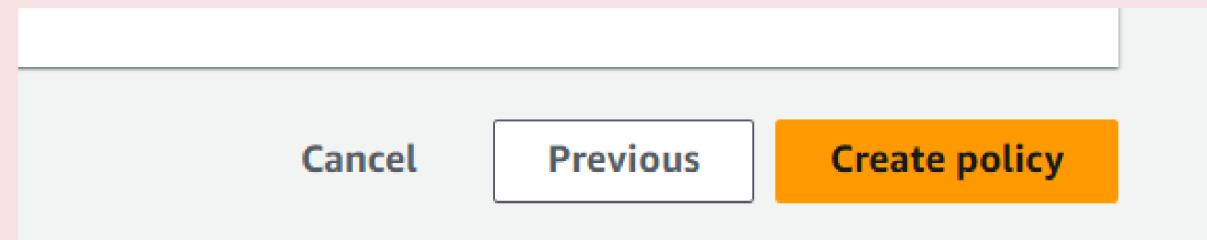


Step 3:

For uploading photo , click on upload .

Step 4:

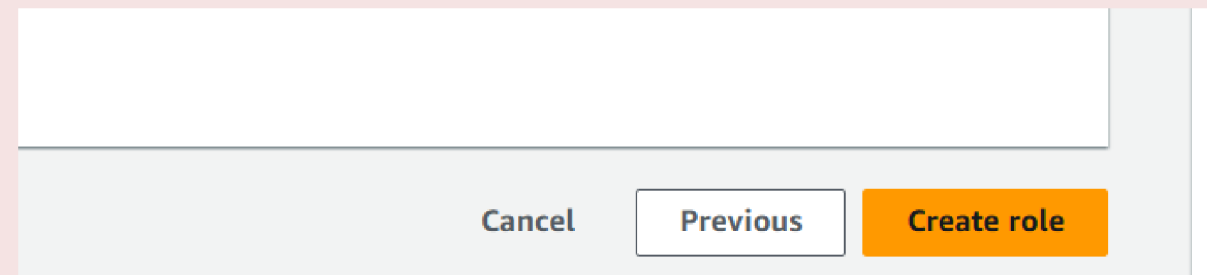
create policy



Go to IAM , click policies given on left hand side and create policied.

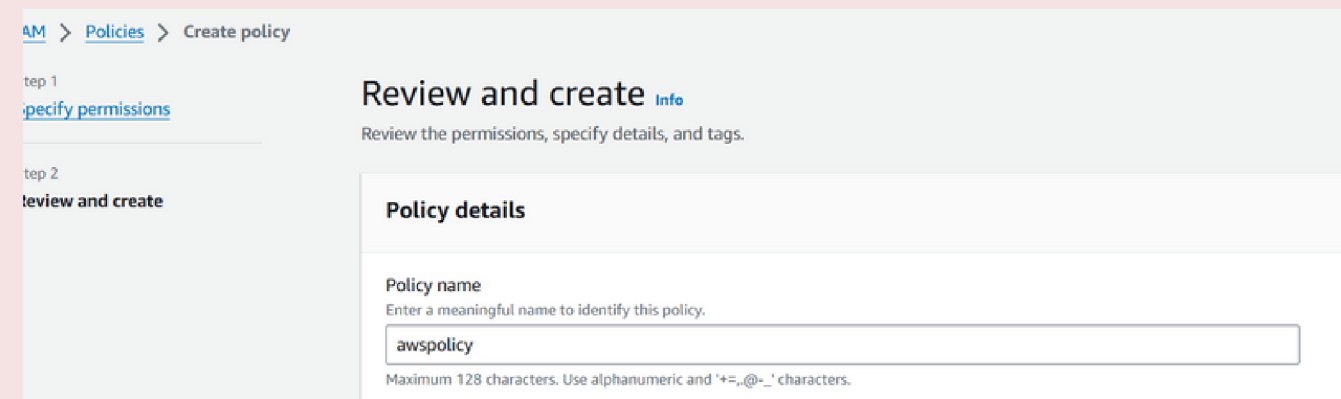
Step 5:

create role

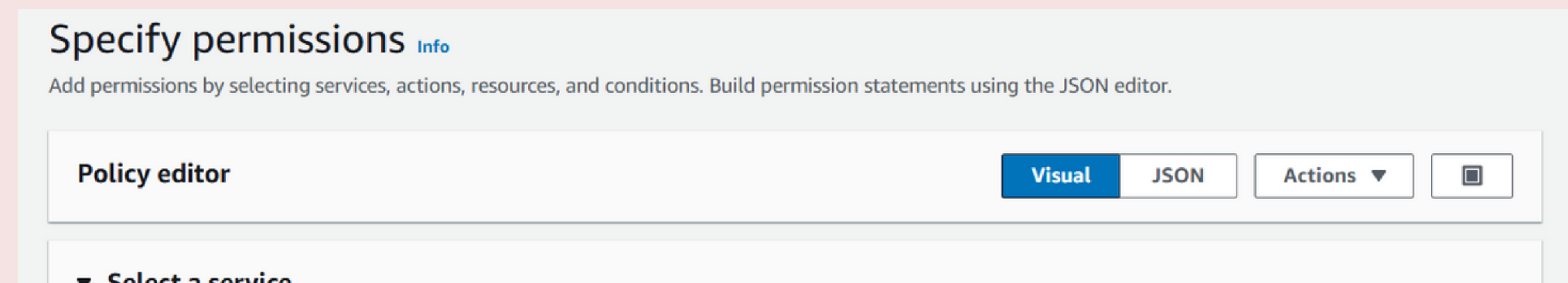


Step 6:

click on the
JSON
button and
write the
accurate code



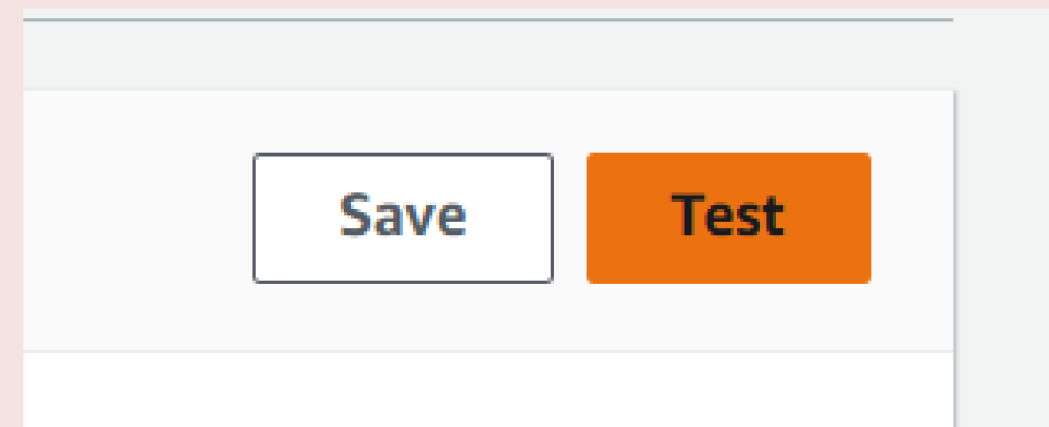
The screenshot shows the 'Review and create' step of the 'Create policy' process in the AWS IAM console. The breadcrumb trail is 'IAM > Policies > Create policy'. The left sidebar shows 'Step 1: Specify permissions' and 'Step 2: Review and create'. The main heading is 'Review and create' with an 'Info' link. Below the heading is the instruction 'Review the permissions, specify details, and tags.' The 'Policy details' section contains a 'Policy name' field with the value 'awspolicy'. A note below the field states: 'Enter a meaningful name to identify this policy. Maximum 128 characters. Use alphanumeric and '+=, @-_' characters.'



The screenshot shows the 'Specify permissions' step of the 'Create policy' process in the AWS IAM console. The breadcrumb trail is 'IAM > Policies > Create policy'. The left sidebar shows 'Step 1: Specify permissions' and 'Step 2: Review and create'. The main heading is 'Specify permissions' with an 'Info' link. Below the heading is the instruction 'Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.' The 'Policy editor' section has tabs for 'Visual' (selected), 'JSON', and 'Actions'. There is also a 'Select a service' dropdown menu.

Step 7:

After writting the code ,
test the code



The screenshot shows the 'Save' and 'Test' buttons in the AWS IAM console. The 'Save' button is a white button with a black border, and the 'Test' button is an orange button with white text.

Step 8:

create lambda function .
Lambda function is used to
compute service to execute
image processing tasks.

Lambda > Functions > awsfunction > Edit environment variables

Edit environment variables

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

| Key | Value | |
|------------|-----------------|--------|
| awsdestkey | awsdestination2 | Remove |

[Add environment variable](#)

► Encryption configuration

Cancel [Save](#)

Function name
Enter a name that describes the purpose of your function.

awsfunction

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.

Node.js 18.x [↕](#) [↻](#)

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64

☐ arm64

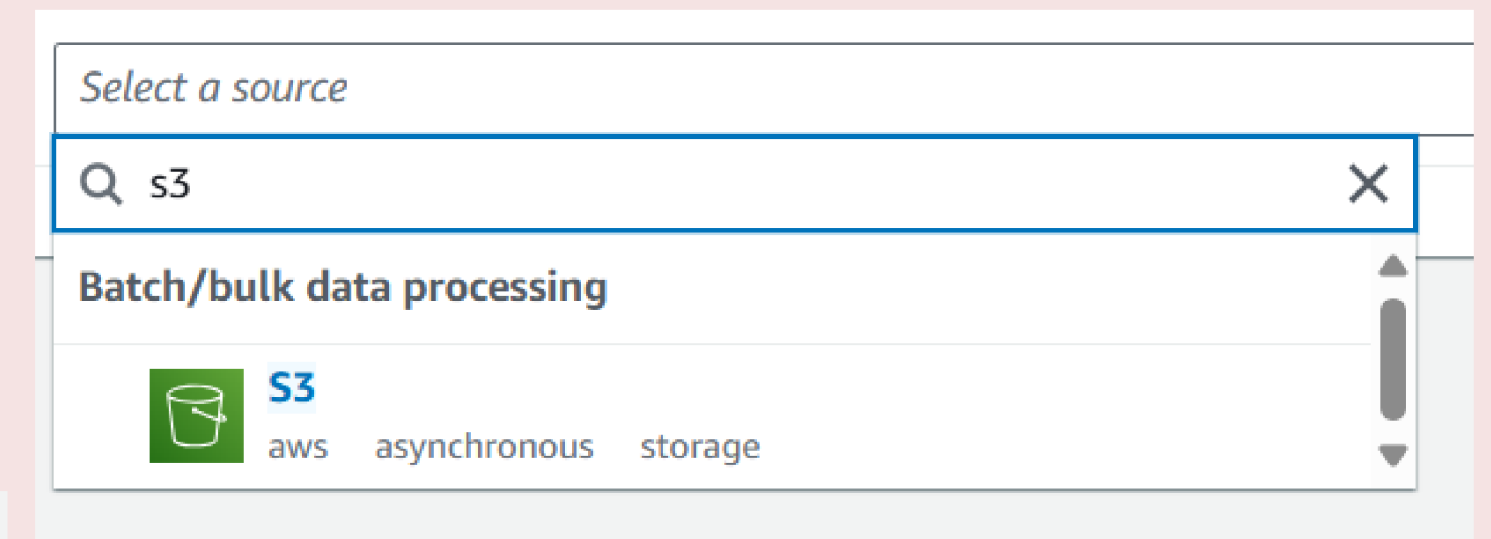
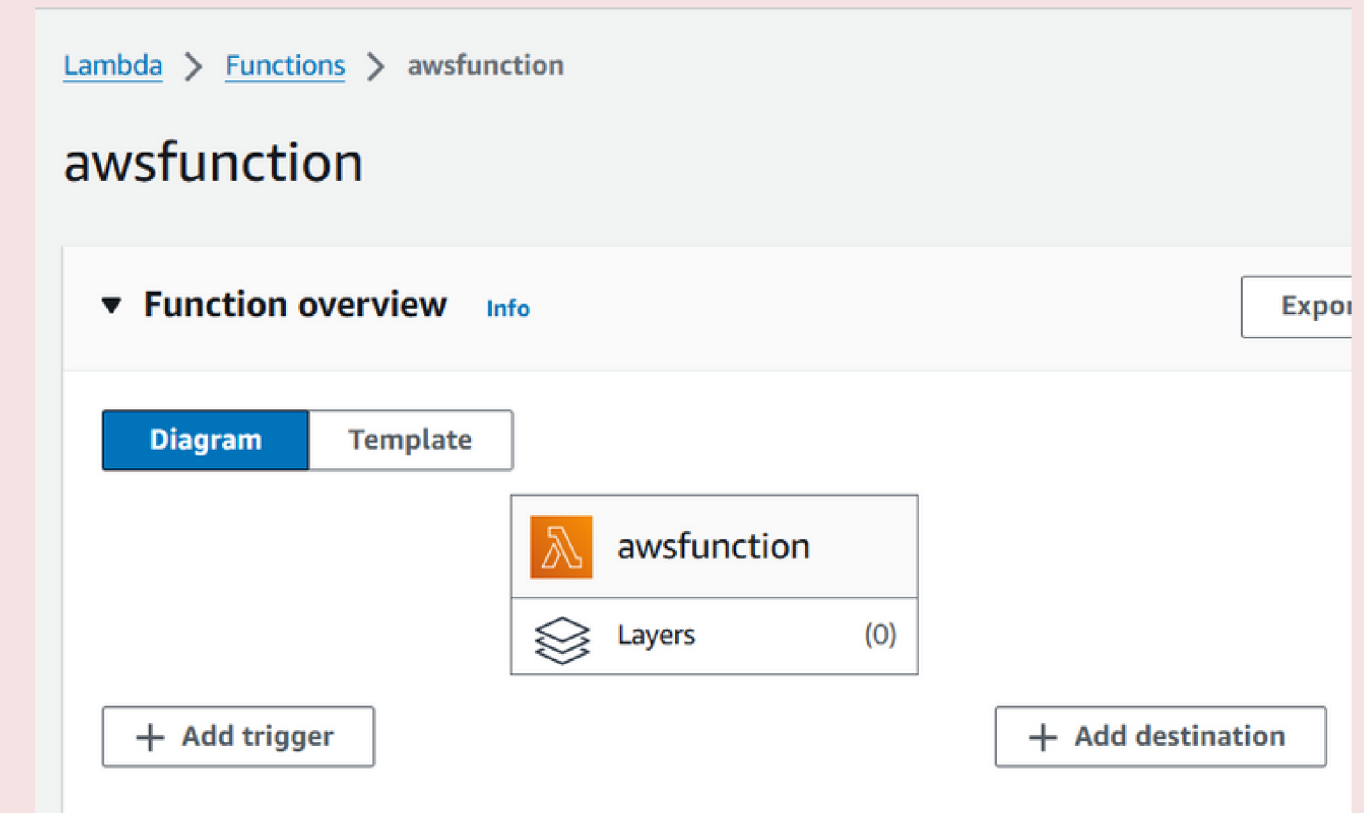
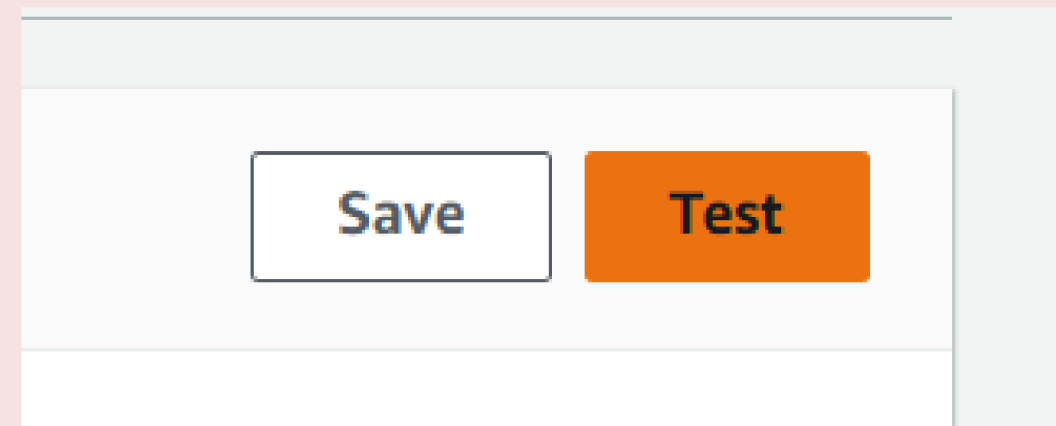
After that click on create function

Step 9:

click on Add trigger then select
S3 bucket then select your source bucket

Step 10:

Go to code and upload zip file
and test the code



The background features a light beige color with several abstract shapes in muted orange and terracotta tones. In the bottom right corner, there is a stylized illustration of a plant with three leaves in a dark brown color.

Thank you

Presented by

LAXMI PANDEY