Step-by-Step Guide: Serverless Image Resizing Application

1. Objective:

Create a serverless application to resize images using AWS Lambda and API Gateway.

2. Tools Required:

- AWS Lambda: To handle image resizing logic.
- AWS API Gateway: To expose the Lambda function via an HTTP API.
- AWS S3 (optional): To store uploaded and resized images.
- Python: For writing the Lambda function.
- Pillow: A Python library for image manipulation.

3. Steps to Implement:

Step 1: Define the Use Case

- The application resizes images to a fixed dimension (e.g., 300x300 pixels).

Step 2: Write Lambda Functions in Python

- Create a Lambda function in AWS Lambda with Python 3.x runtime.
- Use the Pillow library for image resizing
- Use AWS Lambda Layers to add the Pillow library. Create a ZIP file of the library and upload it.

Step 3: Set Up API Gateway

- Create an HTTP API in API Gateway.
- Add a POST route (e.g., /resize) and integrate it with the Lambda function.
- Enable CORS if required.

Step 4: Test the Application

- Use tools like Postman to send a POST request to the API Gateway endpoint.

- Request Body:

```
{
  "image": "<Base64-encoded image>",
  "width": 300,
  "height": 300
}
```

- Validate the output by decoding the Base64 string in the response.

4. Optional Enhancements:

- Store images in S3 using the boto3 library.

- Allow dynamic dimensions via query parameters.

- Enable CloudWatch Logs for monitoring.

- Optimize responses by sharing resized images via signed S3 URLs.