Running a serverless application on AWS involves several key steps. Below is a high-level overview of the procedures to deploy and run a serverless application using AWS services such as AWS Lambda, API Gateway, and AWS SAM (Serverless Application Model).

1. \*\*Set Up Your AWS Environment\*\*

- \*\*Create an AWS Account\*\*: If you don't have an AWS account, create one at [aws.amazon.com](https://aws.amazon.com).

- \*\*Install AWS CLI\*\*: Download and install the AWS Command Line Interface (CLI) on your local machine.

- \*\*Configure AWS CLI\*\*: Run `aws configure` in your terminal and provide your AWS Access Key, Secret Key, region, and output format.

2. \*\*Design Your Serverless Architecture\*\*

- \*\*Define Your Application Logic\*\*: Determine what parts of your application will run as serverless functions (e.g., using AWS Lambda).

- \*\*Identify Event Sources\*\*: Choose the AWS services that will trigger your Lambda functions (e.g., API Gateway, S3, DynamoDB, etc.).

- \*\*Choose Data Storage\*\*: Decide on the storage services you will use, such as Amazon S3 for object storage or DynamoDB for NoSQL databases.

3. \*\*Develop Your Lambda Functions\*\*

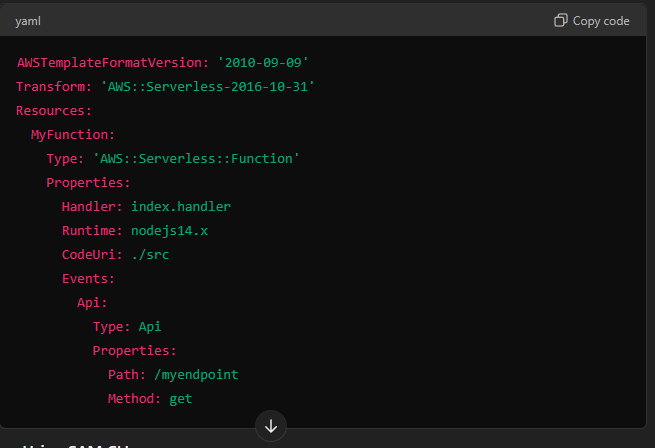
- \*\*Write Your Function Code\*\*: Create the code that will run in your Lambda functions. AWS Lambda supports various languages like Python, Node.js, Java, and more.

- \*\*Package Your Code\*\*: Bundle your application code and dependencies. You can do this manually or use tools like AWS SAM or the Serverless Framework to automate packaging.

4. \*\*Deploy Your Serverless Application\*\*

- \*\*Use AWS SAM or Serverless Framework\*\*: AWS SAM simplifies deploying serverless applications by allowing you to define your application in a template file (YAML).

- Example of a SAM template



Source Code:

yaml

AWSTemplateFormatVersion: '2010-09-09'

Transform: 'AWS::Serverless-2016-10-31'

Resources:

MyFunction:

Type: 'AWS::Serverless::Function'

Properties:

Handler: index.handler

Runtime: nodejs14.x

CodeUri: ./src

Events:

Api:

Type: Api

Properties:

Path: /myendpoint

Method: get

```

- \*\*Deploy Using SAM CLI\*\*:

- Build the application: `sam build`

- Deploy the application: `sam deploy --guided`

- \*\*Manual Deployment\*\*: Alternatively, you can upload the Lambda function code manually via the AWS Management Console.

5. \*\*Configure Event Sources\*\*

- \*\*Set Up API Gateway\*\*: Create a new API in Amazon API Gateway and link it to your Lambda function. This will allow your function to be triggered via HTTP requests.

- \*\*Set Up S3 Events\*\*: If using S3 as an event source, configure bucket notifications to trigger your Lambda function when certain actions (e.g., file uploads) occur.

6. \*\*Monitor and Scale Your Application\*\*

- \*\*Use CloudWatch\*\*: Monitor your application’s performance using AWS CloudWatch. Set up logs and metrics to track usage and troubleshoot issues.

- \*\*Auto Scaling\*\*: AWS Lambda automatically scales your function in response to incoming events, so you don’t need to manage servers.

7. \*\*Secure Your Application\*\*

- \*\*IAM Roles and Policies\*\*: Assign appropriate AWS Identity and Access Management (IAM) roles and policies to your Lambda functions to control permissions.

- \*\*API Gateway Security\*\*: Use API Gateway to enforce security mechanisms like API keys, throttling, and request validation.

8. \*\*Optimize and Maintain\*\*

- \*\*Cost Management\*\*: Monitor your usage and optimize to ensure cost efficiency.

- \*\*Version Control\*\*: Utilize Lambda versioning and aliases to manage updates to your functions without disrupting the production environment.

- \*\*CI/CD Integration\*\*: Integrate with CI/CD pipelines for automated deployment and testing.

9. \*\*Test Your Application\*\*

- \*\*Unit and Integration Testing\*\*: Test your functions locally using tools like SAM CLI (`sam local invoke`) or the Serverless Framework.

- \*\*End-to-End Testing\*\*: Deploy to a staging environment and test the full workflow of your application.

10. \*\*Go Live and Maintain\*\*

- \*\*Deploy to Production\*\*: Once your application is tested, deploy it to the production environment.

- \*\*Monitor\*\*: Continuously monitor the application and make improvements as needed.

This process will help you effectively deploy and manage a serverless application on AWS.

For further Reference …

https://chatgpt.com/c/68b45cf9-f8a9-487d-a954-2c21d7890261