Cloud Native Application - Serverless II

Cloud Infrastructure Engineering

Nanyang Technological University & Skills Union - 2022/2023

Course Content

- Self Study Check In
- Introduction to Serverless
- Pros & Cons of Serverless
- Activity

Q1: What is the maximum execution time for an AWS Lambda function?

a. 5 minutes
b. 10 minutes
c. 15 minutes

Q2: What is an event source for an AWS Lambda function?

a. An external trigger that causes the function to run.
b. An input parameter passed to the function.
c. A configuration setting that determines the runtime environment for the function.

Q3: How to increase the CPU allocation of the Lambda Function?

a. Increase CPU
b. Increase Memory
c. There is no way to increase CPU

Q4: How can you grant permissions for your Lambda Function to access a AWS Service?

a. Using Lambda Resource-based Policy
b. Using Lambda Execution role
c. Using Lambda URL - AWS_IAM Authorizer

Activity

Instructor

- Ask to use AWS use single region for all learner for easier monitoring

AWS Lambda - Basics

Why AWS Lambda?

- No servers to manage
  - No patching required
- Limited by time - short executions
- Run on-demand
  - Not continuously running(eg:- ec2)
- Scaling is automated
  - How?
    - Concurrency options

Benefits of AWS Lambda

- Cheaper(Especially for inconsistent loads)
- Can integrate almost any AWS service(using SDK)
- Supports many programming languages

Lambda- Programming Language Support

- Node.js/JavaScript
- Python
- Java
- C#
- Golang
- Ruby

- For other languages:
  - Custom Runtime API
  - Lambda Container Image
    - Note:- Container image must implement Lambda Runtime API

AWS Lambda - Security

- What is your Lambda function permitted to do?

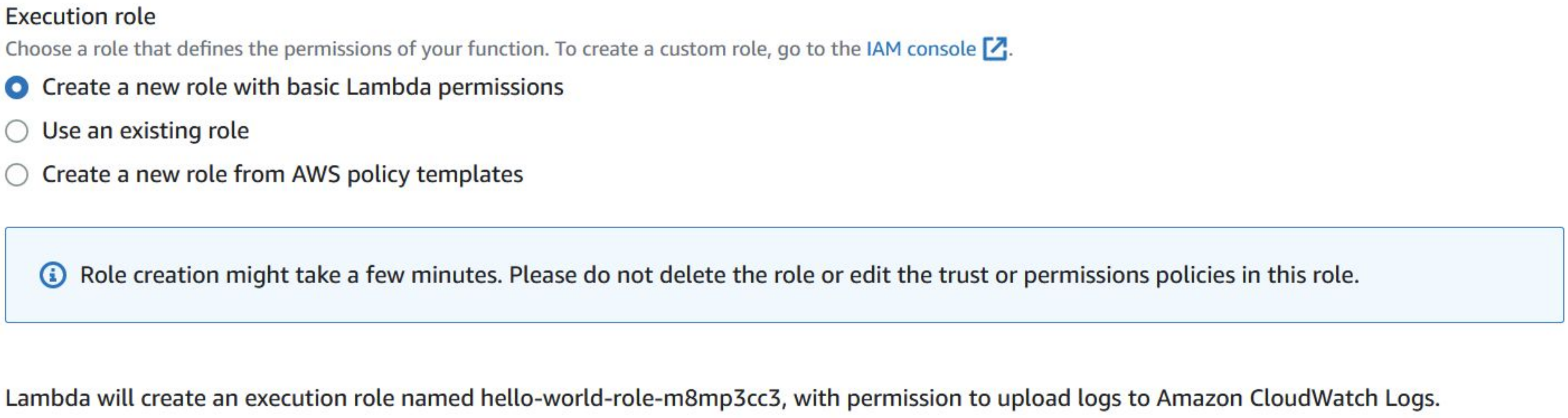
- How do you give granular access to your lambda function?

- Who can invoke your lambda function? Can we restrict?

Lambda - Execution Role

- What is your Lambda function permitted to do?

Lambda Execution Role : grants permissions to access AWS resources from your Lambda function



Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

● Create a new role with basic Lambda permissions

○ Use an existing role

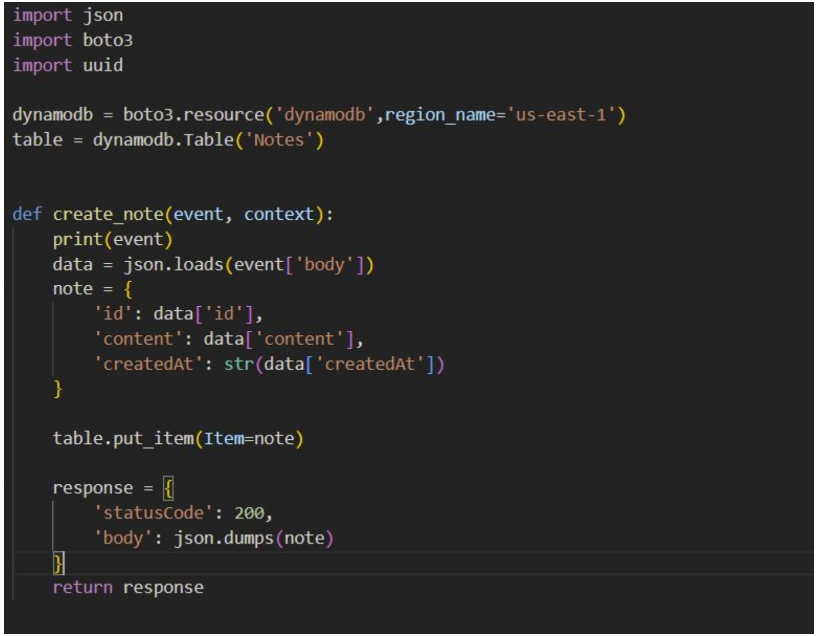
○ Create a new role from AWS policy templates

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named hello-world-role-m8mp3cc3, with permission to upload logs to Amazon CloudWatch Logs.

Lambda - Execution Role

- Q) What permissions you should add in your lambda execution role?

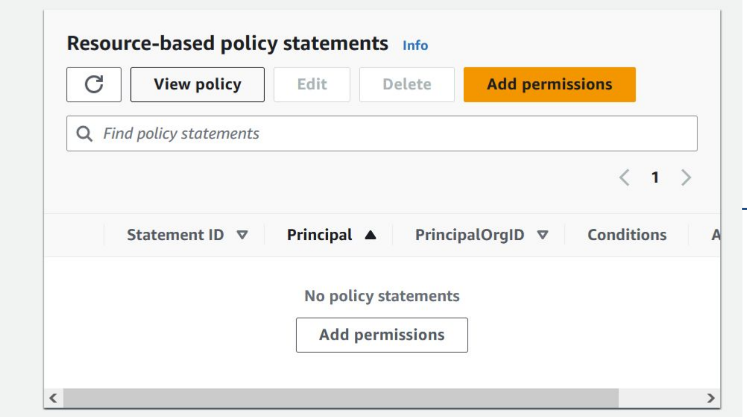


```python
import json
import boto3
import uuid

dynamodb = boto3.resource('dynamodb',region_name='us-east-1')
table = dynamodb.Table('Notes')


def create_note(event, context):
    print(event)
    data = json.loads(event['body'])
    note = {
        'id': data['id'],
        'content': data['content'],
        'createdAt': str(data['createdAt'])
    }

    table.put_item(Item=note)

    response = {
        'statusCode': 200,
        'body': json.dumps(note)
    }
    return response
```

Lambda - Resource Based Policies

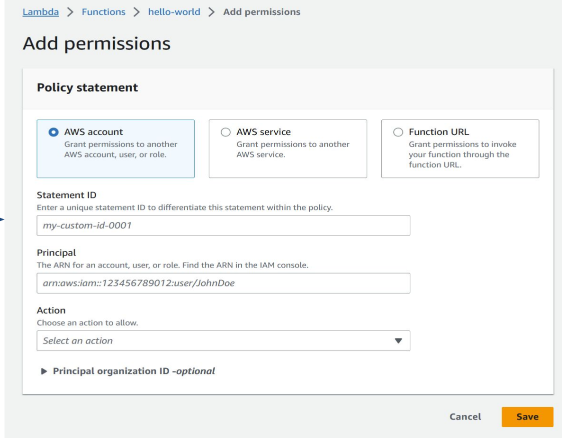
- Who can invoke your lambda function? Can we restrict?
- How do you give granular access to your lambda function?

Resource-based policy: AWS services and other AWS accounts **receive permissions to invoke** your function



Resource-based policy statements

No policy statements



Lambda > Functions > hello-world > Add permissions

Add permissions

Policy statement

● AWS account
Grant permissions to another AWS account, user, or role.

○ AWS service
Grant permissions to another AWS service.

○ Function URL
Grant permissions to invoke your function through the function URL.

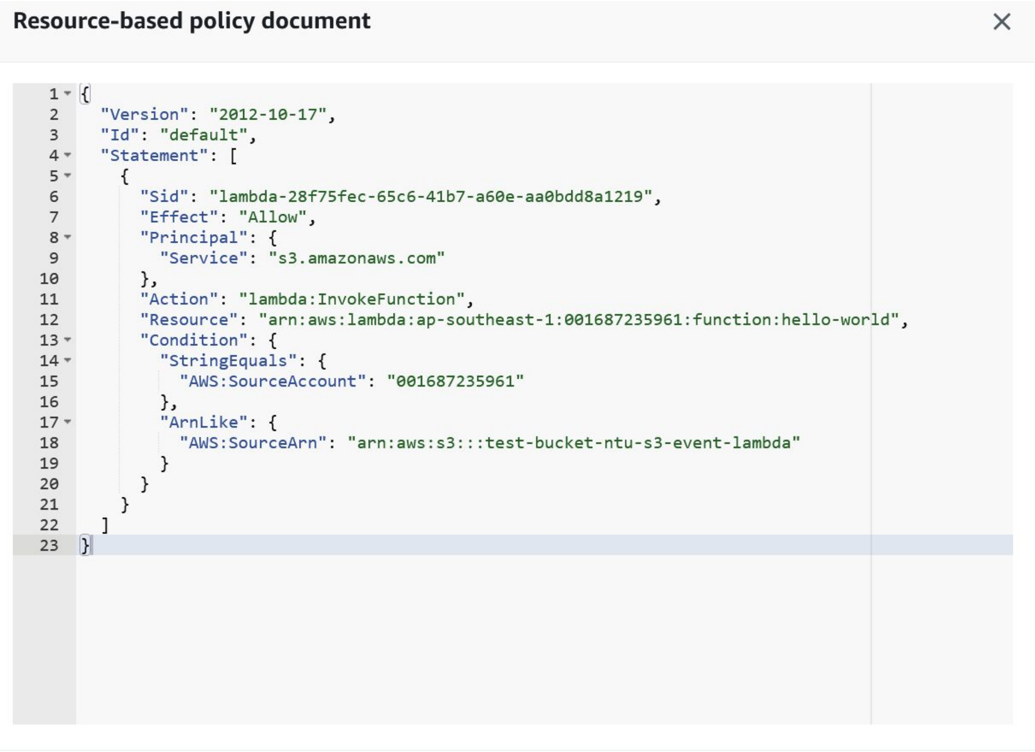
Statement ID
Enter a unique statement ID to differentiate this statement within the policy.
my-custom-id-0001

Principal
The ARN for an account, user, or role. Find the ARN in the IAM console.
arn:aws:iam::123456789012:user/JohnDoe

Action
Choose an action to allow.
Select an action

▶ Principal organization ID *-optional*

Lambda - Resource-based Policies



Resource-based policy document

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "lambda-28f75fec-65c6-41b7-a60e-aa0bdd8a1219",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:ap-southeast-1:001687235961:function:hello-world",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "001687235961"
        },
        "ArnLike": {
          "AWS:SourceArn": "arn:aws:s3:::test-bucket-ntu-s3-event-lambda"
        }
      }
    }
  ]
}
```

Activity:

1. Create a lambda function(AWS management console)
2. Create a S3 bucket
3. Add a S3 trigger

   (Choose your S3 bucket as the trigger)

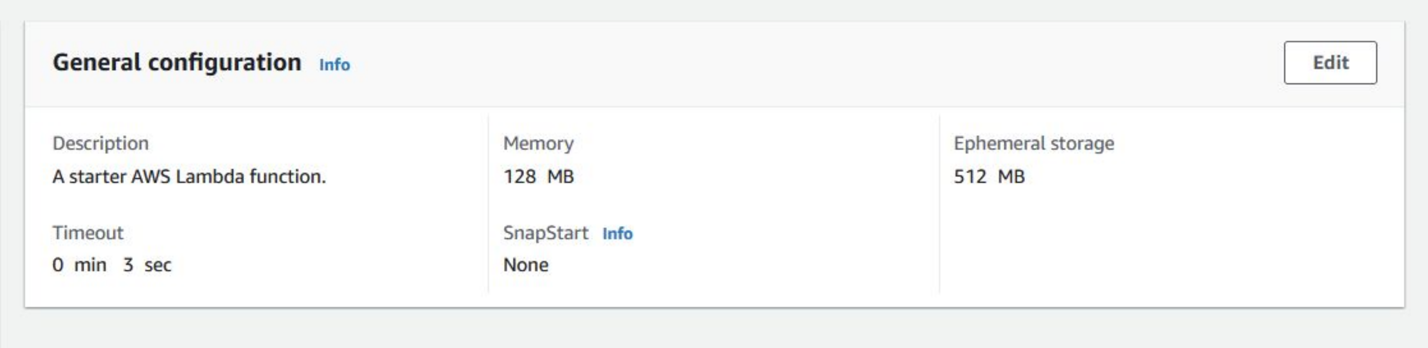
   - Note:- Resource-based policy automatically created
4. Check Resource-based policy in Lambda

Lambda - General configuration

RAM:

- We need to configure **RAM(Max Memory) for Lambda Functions**
  - Min: 128MB
  - Max: 10GB
- No need to configure CPU
  - More RAM you add, the more vCPU credits you get

Timeout: 3 seconds(default), Maximum 15 mins(900 seconds)



Description
A starter AWS Lambda function.

Memory
128 MB

Ephemeral storage
512 MB

Timeout
0 min 3 sec

SnapStart
None

Where to check the logs if my Lambda function fails?

Lambda - Logging,Monitoring & Tracing

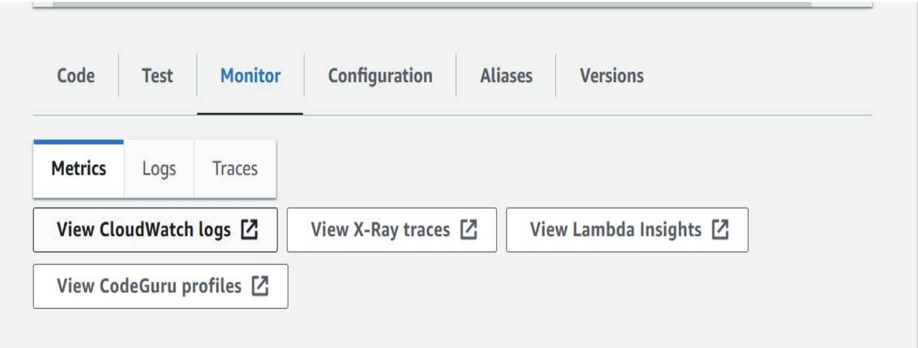
How to check the total number of requests, latency, Success/Error rates, and execution duration?

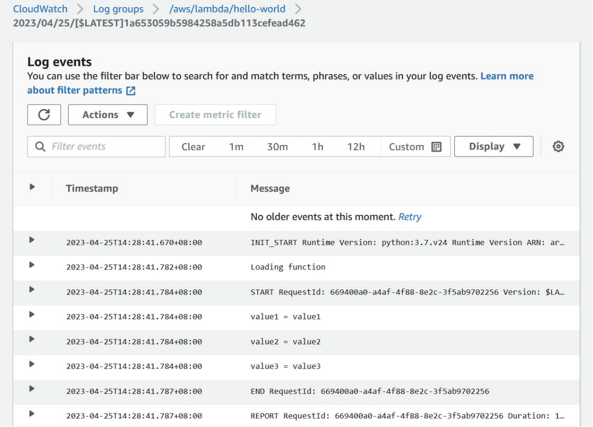
Which component of my application causing performance bottleneck?

Lambda - Logging

CloudWatch Logs

- By default, Cloudwatch logs are stored in AWS CloudWatch Logs.
- Make sure to include an execution role with an IAM policy that authorizes writes to CloudWatch Logs
- **/aws/lambda/<function name>**





2023/04/25/[$LATEST]1a653059b5984258a5db113cefead462

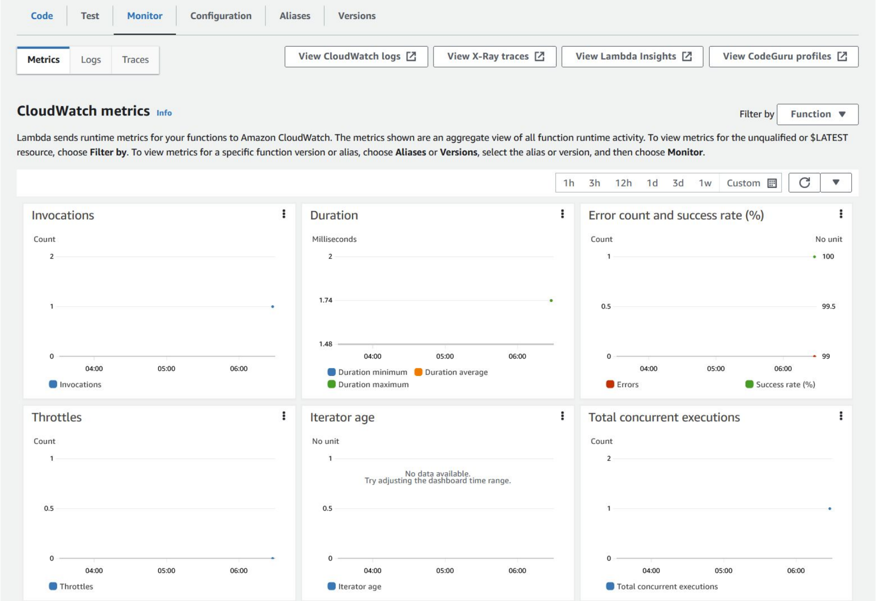
Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. **Learn more about filter patterns**

```
Timestamp                        Message
                                 No older events at this moment. Retry
2023-04-25T14:28:41.670+08:00    INIT_START Runtime Version: python:3.7.v24 Runtime Version ARN: ar…
2023-04-25T14:28:41.782+08:00    Loading function
2023-04-25T14:28:41.784+08:00    START RequestId: 669400a0-a4af-4f88-8e2c-3f5ab9702256 Version: $LA…
2023-04-25T14:28:41.784+08:00    value1 = value1
2023-04-25T14:28:41.784+08:00    value2 = value2
2023-04-25T14:28:41.784+08:00    value3 = value3
2023-04-25T14:28:41.787+08:00    END RequestId: 669400a0-a4af-4f88-8e2c-3f5ab9702256
2023-04-25T14:28:41.787+08:00    REPORT RequestId: 669400a0-a4af-4f88-8e2c-3f5ab9702256 Duration: 1…
```

Lambda - Monitoring

CloudWatch Metrics



CloudWatch metrics Info

Lambda sends runtime metrics for your functions to Amazon CloudWatch. The metrics shown are an aggregate view of all function runtime activity. To view metrics for the unqualified or $LATEST resource, choose **Filter by**. To view metrics for a specific function version or alias, choose **Aliases** or **Versions**, select the alias or version, and then choose **Monitor**.

No data available.
Try adjusting the dashboard time range.

Invocations – number of times your function is invoked

Duration – amount of time your function spends processing an event

Errors – number of invocations that result in a function error

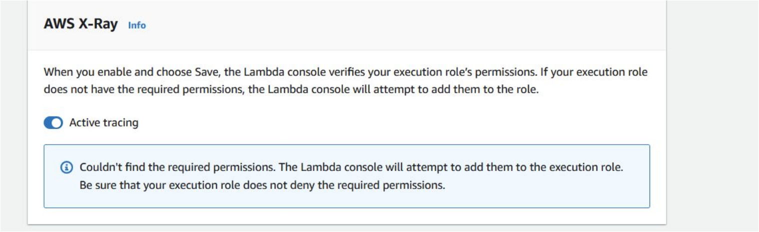
Throttles – number of invocation requests that are throttled (no concurrency available)

Total Concurrent Executions – number of function instances that are processing events

Lambda - Tracing

X-Ray

- You can enable in Lambda configuration
- Ideal tracing for microservices architecture



AWS X-Ray Info

When you enable and choose Save, the Lambda console verifies your execution role's permissions. If your execution role does not have the required permissions, the Lambda console will attempt to add them to the role.

● Active tracing

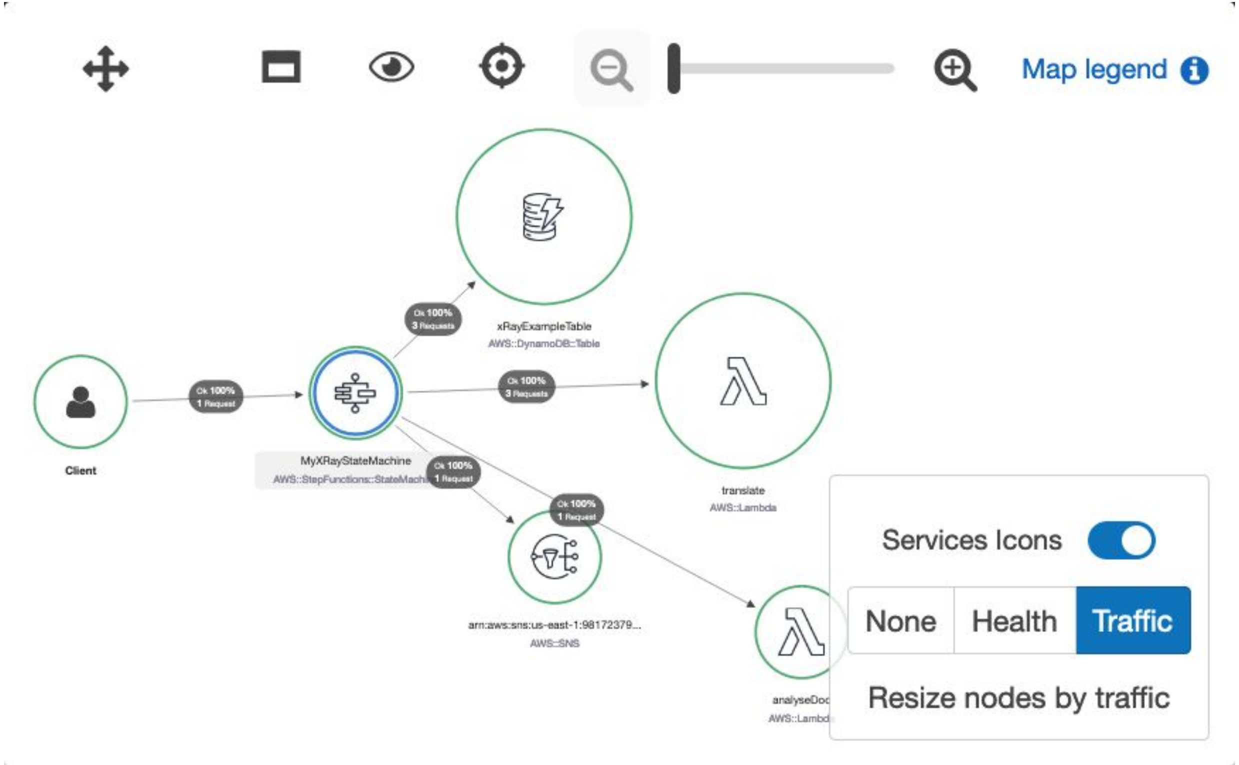
ⓘ Couldn't find the required permissions. The Lambda console will attempt to add them to the execution role. Be sure that your execution role does not deny the required permissions.

Why?
To provide an end-to-end view of your application to help you more efficiently **pinpoint performance bottlenecks** and **identify impacted users**

You can use the AWS X-Ray SDK to annotate the data sent to X-Ray, trace downstream calls, and record exceptions.

Example:
https://docs.aws.amazon.com/xray/latest/devguide/xray-scorekeep.html

- Permissions required in Execution role: AWSXRayDaemonWriteAccess

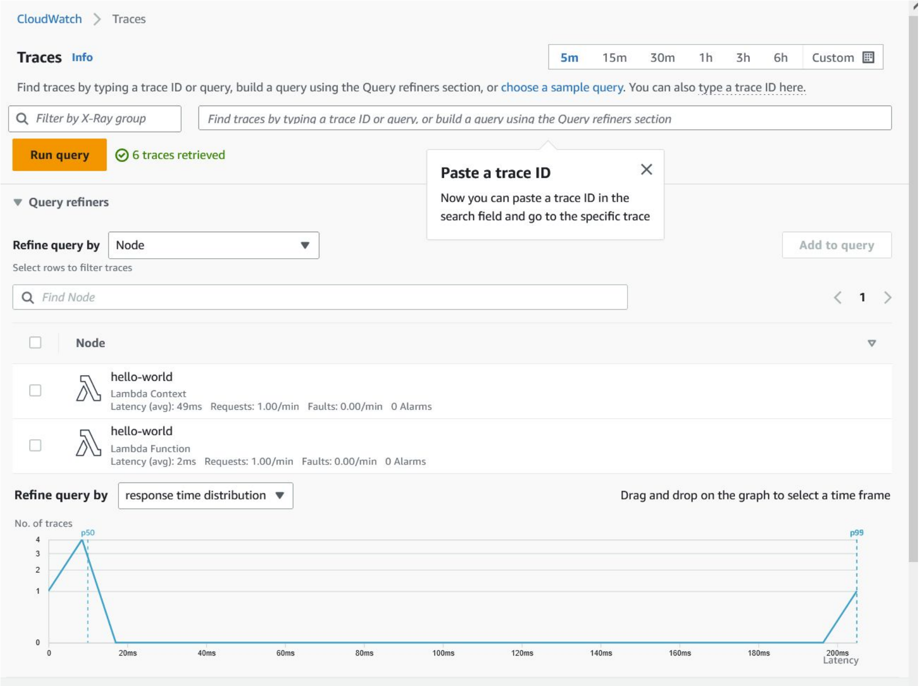


Services Icons ●

None | Health | **Traffic**

Resize nodes by traffic

Lambda - Tracing



Find traces by typing a trace ID or query, build a query using the Query refiners section, or choose a sample query. You can also type a trace ID here.

Find traces by typing a trace ID or query, or build a query using the Query refiners section

6 traces retrieved

Paste a trace ID

Now you can paste a trace ID in the search field and go to the specific trace

Refine query by: Node

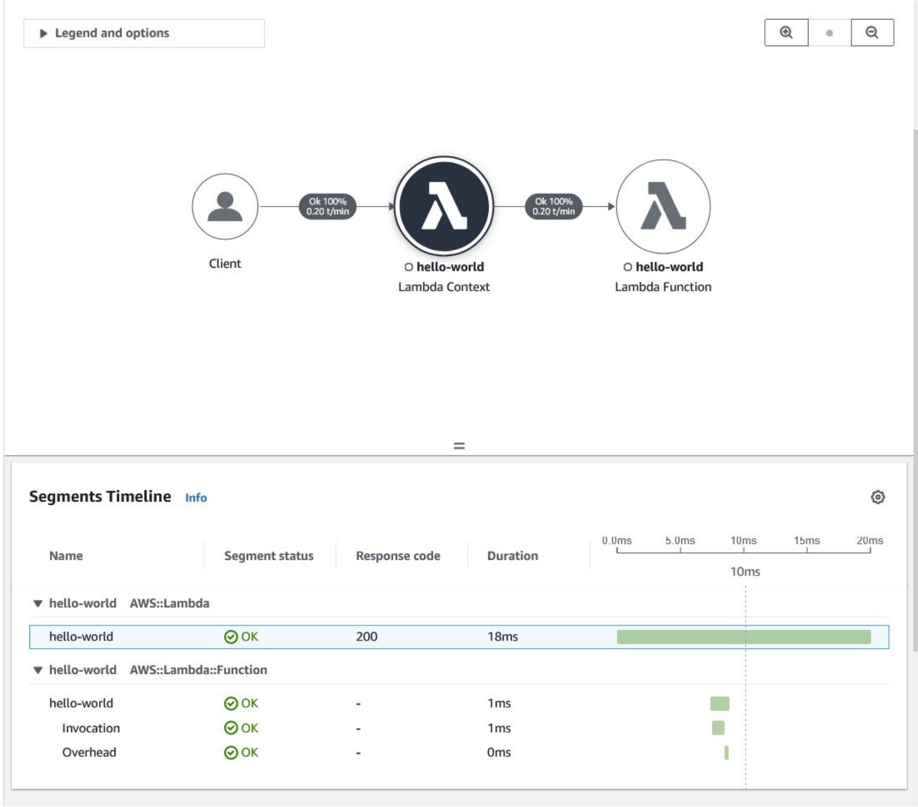
Select rows to filter traces

□	Node
□	hello-world Lambda Context Latency (avg): 49ms Requests: 1.00/min Faults: 0.00/min 0 Alarms
□	hello-world Lambda Function Latency (avg): 2ms Requests: 1.00/min Faults: 0.00/min 0 Alarms

Refine query by: response time distribution

Drag and drop on the graph to select a time frame

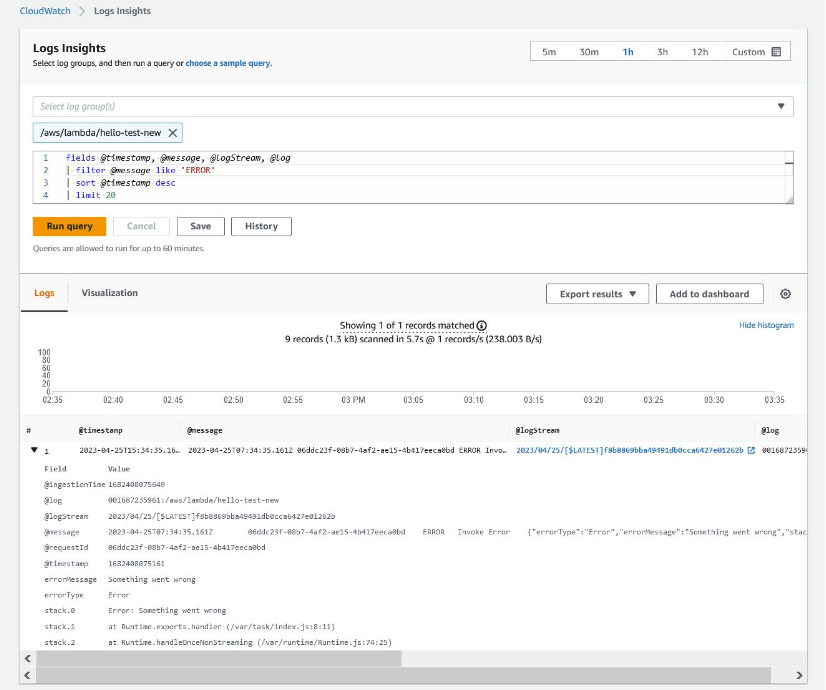
Lambda - Tracing



Segments Timeline Info

Name	Segment status	Response code	Duration
▼ hello-world AWS::Lambda			
hello-world	✓ OK	200	18ms
▼ hello-world AWS::Lambda::Function			
hello-world	✓ OK	-	1ms
Invocation	✓ OK	-	1ms
Overhead	✓ OK	-	0ms

Lambda - CloudWatch Logs Insights



Select log groups, and then run a query or choose a sample query.

```
fields @timestamp, @message, @logStream, @log
| filter @message like 'ERROR'
| sort @timestamp desc
| limit 20
```

Queries are allowed to run for up to 60 minutes.

Showing 1 of 1 records matched
9 records (1.3 kB) scanned in 5.7s @ 1 records/s (238.003 B/s)

#	@timestamp	@message	@logStream	@log
▼ 1	2023-04-25T15:34:35.16…	2023-04-25T07:34:35.161Z 06ddc23f-08b7-4af2-ae15-4b417eeca0bd ERROR Invo…	2023/04/25/[$LATEST]f8b8869bba49491db0cca6427e01262b	00168723596

```
Field          Value
@ingestionTime 1682408075649
@log           001687235961:/aws/lambda/hello-test-new
@logStream     2023/04/25/[$LATEST]f8b8869bba49491db0cca6427e01262b
@message       2023-04-25T07:34:35.161Z        06ddc23f-08b7-4af2-ae15-4b417eeca0bd    ERROR   Invoke Error    {"errorType":"Error","errorMessage":"Something went wrong","stac
@requestId     06ddc23f-08b7-4af2-ae15-4b417eeca0bd
@timestamp     1682408075161
errorMessage   Something went wrong
errorType      Error
stack.0        Error: Something went wrong
stack.1        at Runtime.exports.handler (/var/task/index.js:8:11)
stack.2        at Runtime.handleOnceNonStreaming (/var/runtime/Runtime.js:74:25)
```

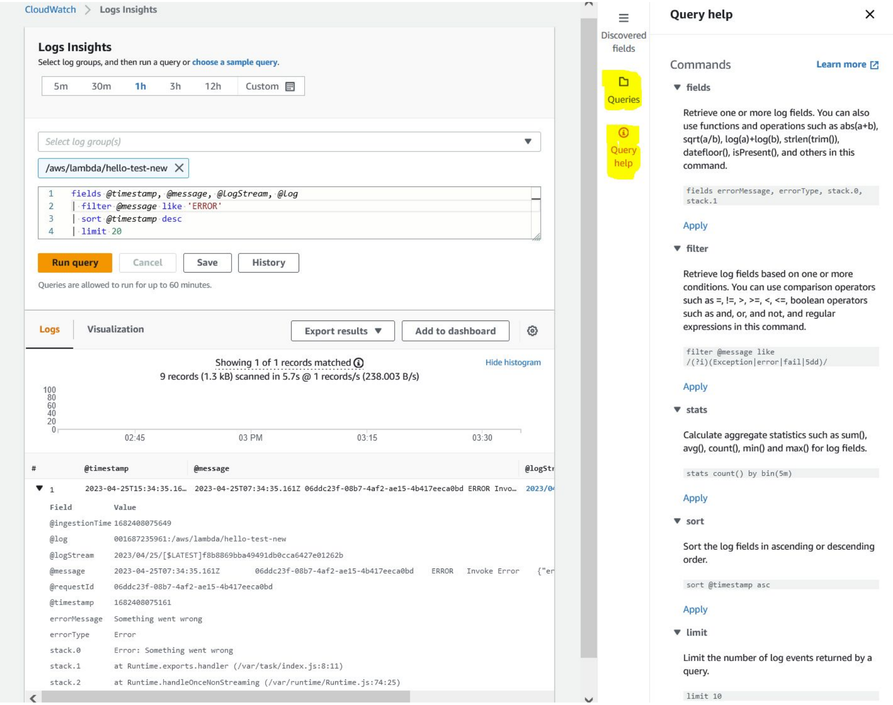
CloudWatch Logs Insights

- Which allows to query your Lambda Cloudwatch logs

Eg:- Get the latest 20 invocation logs having errors

Lambda - CloudWatch Logs Insights

How to create your query?



Logs Insights

Select log groups, and then run a query or choose a sample query.

/aws/lambda/hello-test-new

```
fields @timestamp, @message, @logStream, @log
| filter @message like 'ERROR'
| sort @timestamp desc
| limit 20
```

Queries are allowed to run for up to 60 minutes.

Showing 1 of 1 records matched

9 records (1.3 kB) scanned in 5.7s @ 1 records/s (238.003 B/s)

```
#   @timestamp                @message                                                                        @logStr
▼ 1 2023-04-25T15:34:35.16…   2023-04-25T07:34:35.161Z 06ddc23f-08b7-4af2-ae15-4b417eeca0bd ERROR Invo…   2023/04
    Field          Value
    @ingestionTime 1682408075649
    @log           001687235961:/aws/lambda/hello-test-new
    @logStream     2023/04/25/[$LATEST]f8b8869bba49491db0cca6427e01262b
    @message       2023-04-25T07:34:35.161Z      06ddc23f-08b7-4af2-ae15-4b417eeca0bd    ERROR   Invoke Error    {"er
    @requestId     06ddc23f-08b7-4af2-ae15-4b417eeca0bd
    @timestamp     1682408075161
    errorMessage   Something went wrong
    errorType      Error
    stack.0        Error: Something went wrong
    stack.1        at Runtime.exports.handler (/var/task/index.js:8:11)
    stack.2        at Runtime.handleOnceNonStreaming (/var/runtime/Runtime.js:74:25)
```

Query help

Commands

▼ **fields**

Retrieve one or more log fields. You can also use functions and operations such as abs(a+b), sqrt(a/b), log(a)+log(b), strlen(trim()), datefloor(), isPresent(), and others in this command.

```
fields errorMessage, errorType, stack.0,
stack.1
```

▼ **filter**

Retrieve log fields based on one or more conditions. You can use comparison operators such as =, !=, >, >=, <, <=, boolean operators such as and, or, and not, and regular expressions in this command.

```
filter @message like
/(?i)(Exception|error|fail|5dd)/
```

▼ **stats**

Calculate aggregate statistics such as sum(), avg(), count(), min() and max() for log fields.

```
stats count() by bin(5m)
```

▼ **sort**

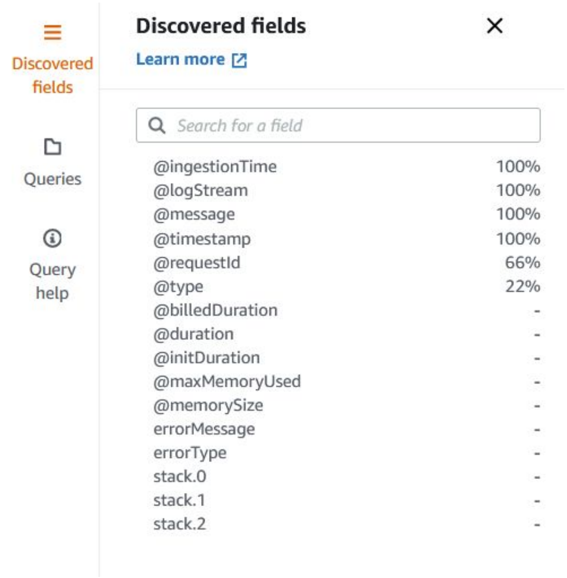
Sort the log fields in ascending or descending order.

```
sort @timestamp asc
```

▼ **limit**

Limit the number of log events returned by a query.

```
limit 10
```



Discovered fields

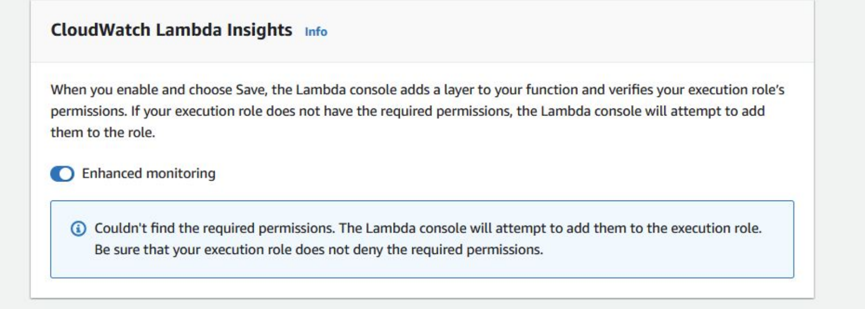
@ingestionTime	100%
@logStream	100%
@message	100%
@timestamp	100%
@requestId	66%
@type	22%
@billedDuration	-
@duration	-
@initDuration	-
@maxMemoryUsed	-
@memorySize	-
errorMessage	-
errorType	-
stack.0	-
stack.1	-
stack.2	-

Learn more: https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CWL_AnalyzeLogData-discoverable-fields.html

Lambda - Cloudwatch Lambda Insights

Why?

- Right Sizing: over- and under-utilized Lambda functions(over/under provisioned memory)
- Performance Monitoring:
  - Max Memory usage,Network Usage



CloudWatch Lambda Insights Info

When you enable and choose Save, the Lambda console adds a layer to your function and verifies your execution role's permissions. If your execution role does not have the required permissions, the Lambda console will attempt to add them to the role.

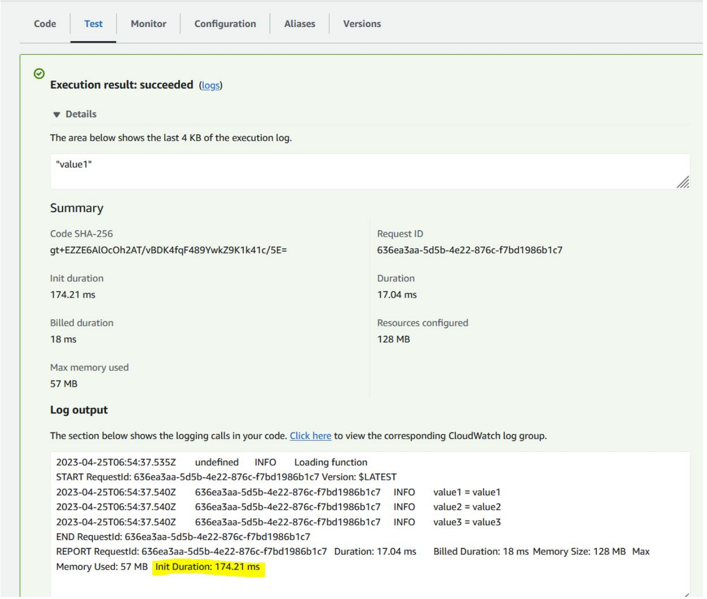
● Enhanced monitoring

ⓘ Couldn't find the required permissions. The Lambda console will attempt to add them to the execution role. Be sure that your execution role does not deny the required permissions.

Watch:
https://www.youtube.com/watch?v=m452gokXTME

Lambda - Cold Start Problem

Problem: First request has higher latency than the rest (init duration)



Execution result: succeeded (logs)

▼ Details

The area below shows the last 4 KB of the execution log.

```
"value1"
```

Summary

Code SHA-256 gt+EZZE6AlOcOh2AT/vBDK4fqF489YwkZ9K1k41c/5E=	Request ID 636ea3aa-5d5b-4e22-876c-f7bd1986b1c7
Init duration 174.21 ms	Duration 17.04 ms
Billed duration 18 ms	Resources configured 128 MB
Max memory used 57 MB	

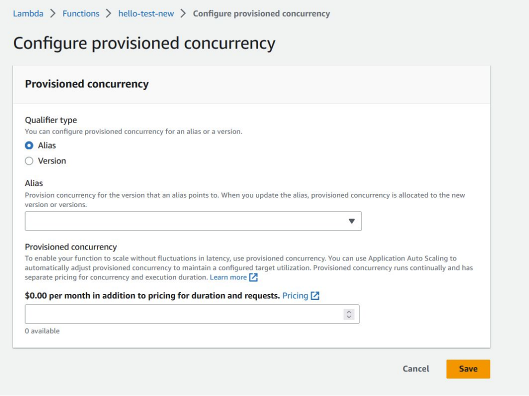
Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
2023-04-25T06:54:37.535Z	undefined	INFO	Loading function
START RequestId: 636ea3aa-5d5b-4e22-876c-f7bd1986b1c7 Version: $LATEST
2023-04-25T06:54:37.540Z	636ea3aa-5d5b-4e22-876c-f7bd1986b1c7	INFO	value1 = value1
2023-04-25T06:54:37.540Z	636ea3aa-5d5b-4e22-876c-f7bd1986b1c7	INFO	value2 = value2
2023-04-25T06:54:37.540Z	636ea3aa-5d5b-4e22-876c-f7bd1986b1c7	INFO	value3 = value3
END RequestId: 636ea3aa-5d5b-4e22-876c-f7bd1986b1c7
REPORT RequestId: 636ea3aa-5d5b-4e22-876c-f7bd1986b1c7	Duration: 17.04 ms	Billed Duration: 18 ms	Memory Size: 128 MB	Max Memory Used: 57 MB	Init Duration: 174.21 ms
```

Solution :

1) “Provisioned Concurrency”
   - Concurrency is allocated before the function is invoked



Lambda > Functions > hello-test-new > Configure provisioned concurrency

Configure provisioned concurrency

Provisioned concurrency

Qualifier type

You can configure provisioned concurrency for an alias or a version.

● Alias

○ Version

Alias

Provision concurrency for the version that an alias points to. When you update the alias, provisioned concurrency is allocated to the new version or versions.

Provisioned concurrency

To enable your function to scale without fluctuations in latency, use provisioned concurrency. You can use Application Auto Scaling to automatically adjust provisioned concurrency to maintain a configured target utilization. Provisioned concurrency runs continually and has separate pricing for concurrency and execution duration. Learn more

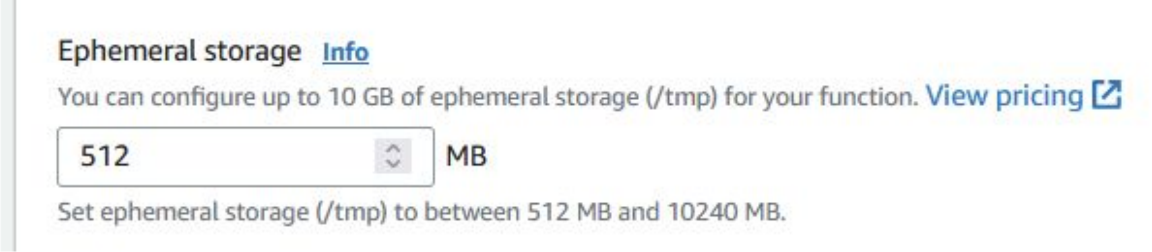
$0.00 per month in addition to pricing for duration and requests. Pricing

0 available

2) SnapStart (only for Java)

Lambda - Storage options

1) /tmp



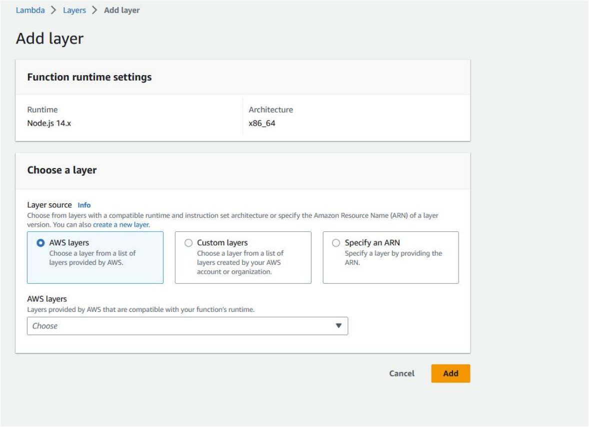
Ephemeral storage Info

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

2) Lambda Layers



Lambda > Layers > Add layer

Add layer

Function runtime settings

Runtime: Node.js 14.x

Architecture: x86_64

Choose a layer

Layer source Info

Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also create a new layer.

● AWS layers
Choose a layer from a list of layers provided by AWS.

○ Custom layers
Choose a layer from a list of layers created by your AWS account or organization.

○ Specify an ARN
Specify a layer by providing the ARN.

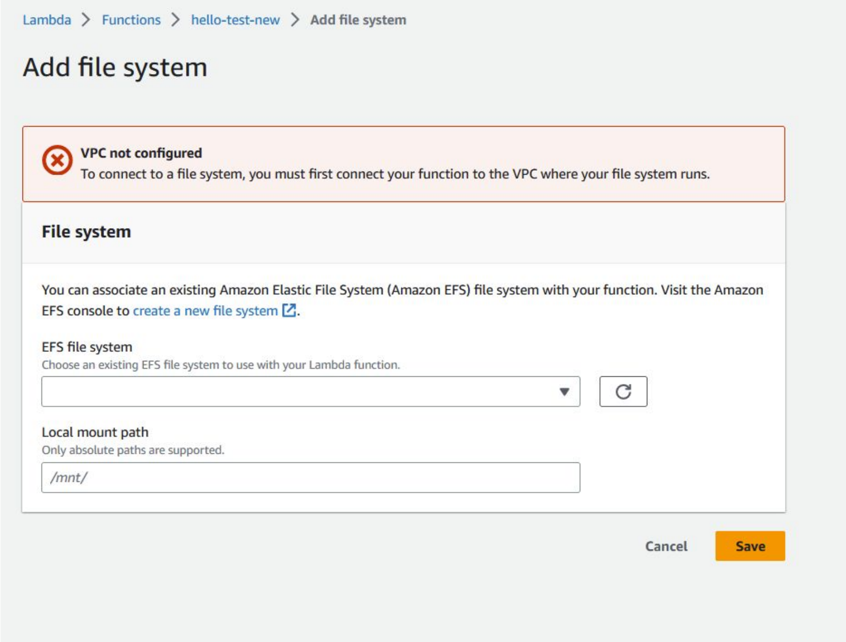
AWS layers
Layers provided by AWS that are compatible with your function's runtime.

Choose ▼

3) S3

Lambda - Storage options

4) EFS



Add file system

VPC not configured
To connect to a file system, you must first connect your function to the VPC where your file system runs.

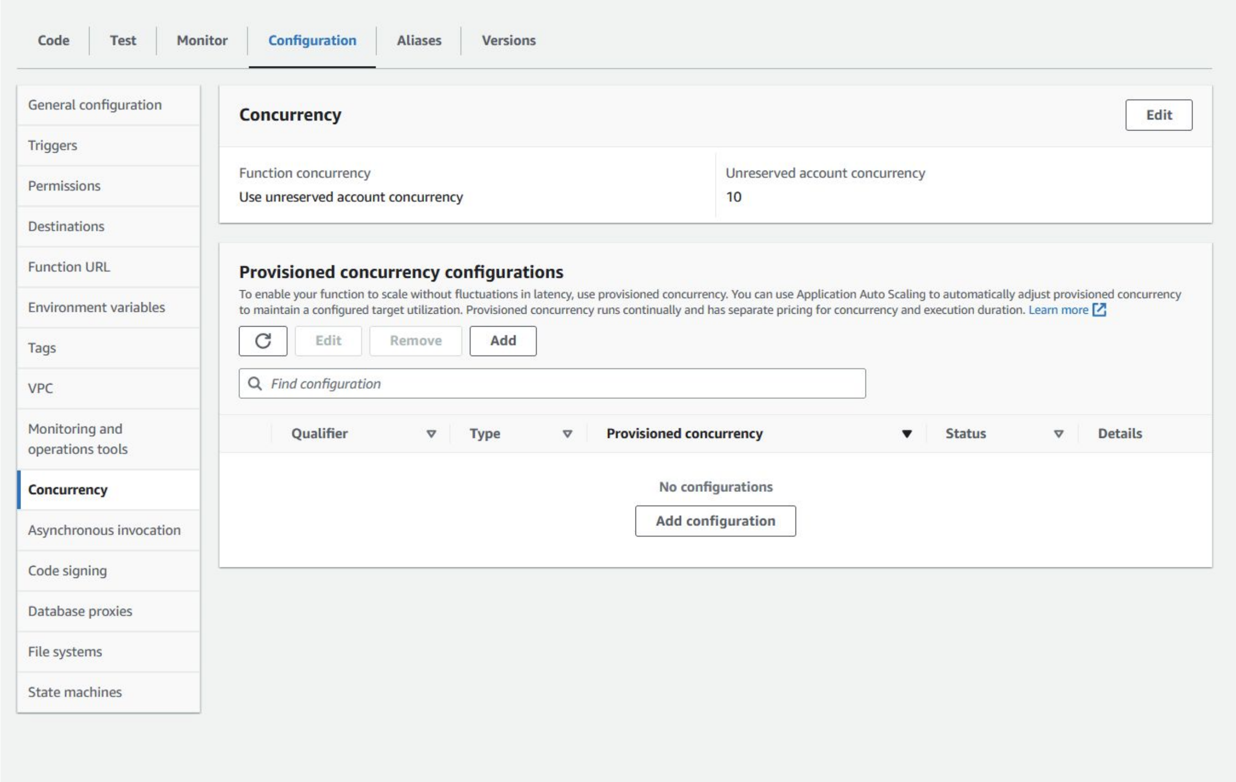
File system

You can associate an existing Amazon Elastic File System (Amazon EFS) file system with your function. Visit the Amazon EFS console to create a new file system.

EFS file system
Choose an existing EFS file system to use with your Lambda function.

Local mount path
Only absolute paths are supported.
/mnt/

Lambda - Concurrency



Concurrency

Function concurrency: Use unreserved account concurrency

Unreserved account concurrency: 10

Provisioned concurrency configurations

To enable your function to scale without fluctuations in latency, use provisioned concurrency. You can use Application Auto Scaling to automatically adjust provisioned concurrency to maintain a configured target utilization. Provisioned concurrency runs continually and has separate pricing for concurrency and execution duration. Learn more

Qualifier	Type	Provisioned concurrency	Status	Details

No configurations

Lambda - Pricing

Lambda Cost Calculator: https://calculator.aws/#/addService/Lambda10

Try to change the number of requests, memory and see the cost.

Lambda - Invocation options

1. Synchronous Invokes

Synchronous invocations are the most straightforward way to invoke your Lambda functions. In this model, your functions execute immediately when you perform the Lambda Invoke API call.

The Invocation-type flag specifies a value of “**RequestResponse**”. This instructs AWS to execute your Lambda function and wait for the function to complete. When you perform a synchronous invoke, you are responsible for checking the response and determining if there was an error and if you should retry the invoke.

2. Asynchronous Invokes

Asynchronous invokes **place your invoke request in Lambda service queue** and we process the requests as they arrive.
You should use AWS X-Ray to review how long your request spent in the service queue by checking the “dwell time” segment.

Ref:

- https://docs.aws.amazon.com/lambda/latest/operatorguide/invocation-modes.html
- https://docs.aws.amazon.com/lambda/latest/dg/lambda-services.html

Synchronous Invokes

Here is a list of services that invoke Lambda functions synchronously:

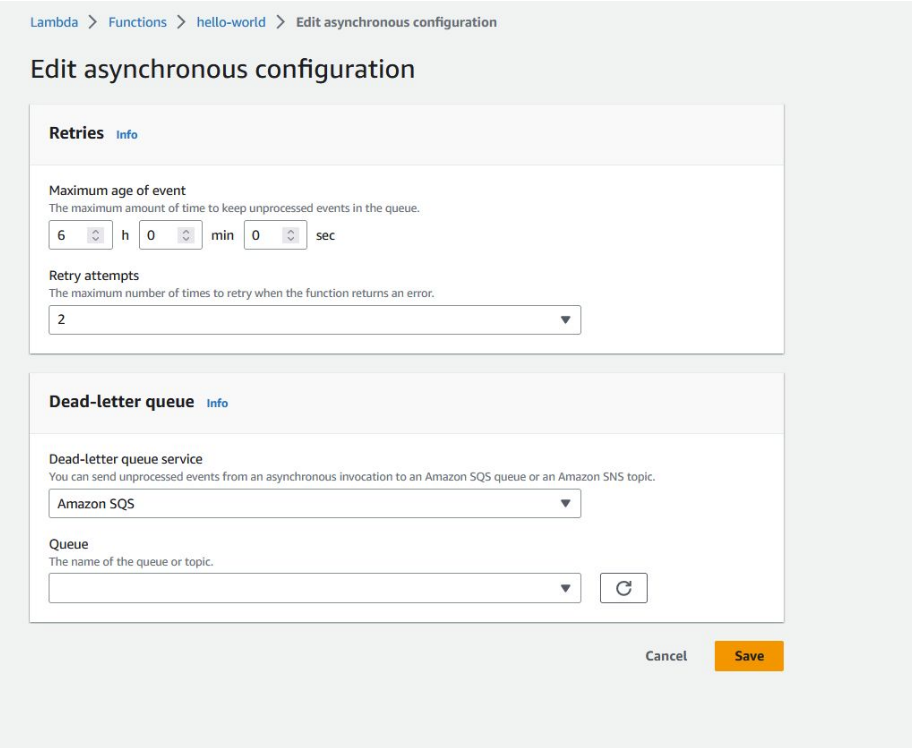
- Amazon API Gateway
- Elastic Load Balancing (Application Load Balancer)
- Amazon Cognito
- Amazon Lex
- Amazon Alexa
- Amazon Kinesis Data Firehose

Asynchronous Invokes

Here is a list of services that invoke Lambda functions asynchronously:

- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon Simple Email Service
- CloudFormation
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CodeCommit

Asynchronous Invokes



Lambda > Functions > hello-world > Edit asynchronous configuration

Edit asynchronous configuration

Retries

Maximum age of event
The maximum amount of time to keep unprocessed events in the queue.
6 h 0 min 0 sec

Retry attempts
The maximum number of times to retry when the function returns an error.
2

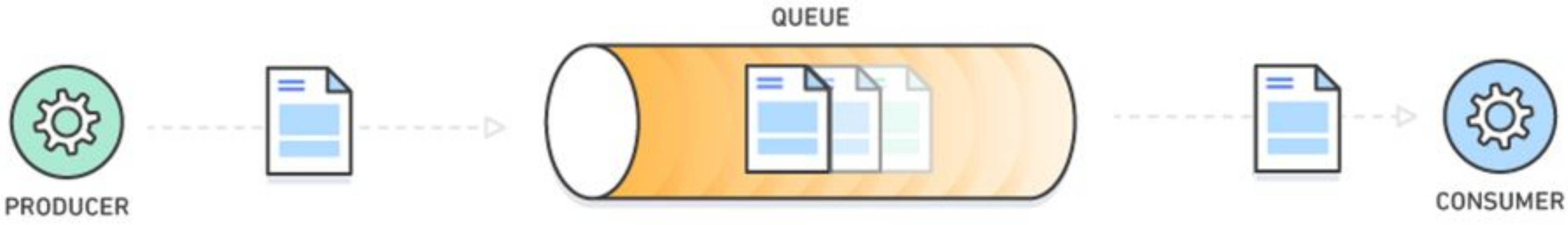
Dead-letter queue

Dead-letter queue service
You can send unprocessed events from an asynchronous invocation to an Amazon SQS queue or an Amazon SNS topic.
Amazon SQS

Queue
The name of the queue or topic.

Message Queues:
https://aws.amazon.com/message-queue/

Message Queue Architecture



AWS SQS

Amazon Simple Queue Service (Amazon SQS) offers a secure, durable, and available hosted queue that lets you integrate and decouple distributed software systems and components.

Benefits of AWS SQS

Security – You control who can send messages to and receive messages from an Amazon SQS queue. You can choose to transmit sensitive data by protecting the contents of messages in queues by using default Amazon SQS managed server-side encryption (SSE), or by using custom SSE keys managed in AWS Key Management Service (AWS KMS).

Durability – For the safety of your messages, Amazon SQS stores them on multiple servers. Standard queues support at-least-once message delivery, and FIFO queues support exactly-once message processing and high-throughput mode.

Availability – Amazon SQS uses redundant infrastructure to provide highly-concurrent access to messages and high availability for producing and consuming messages.

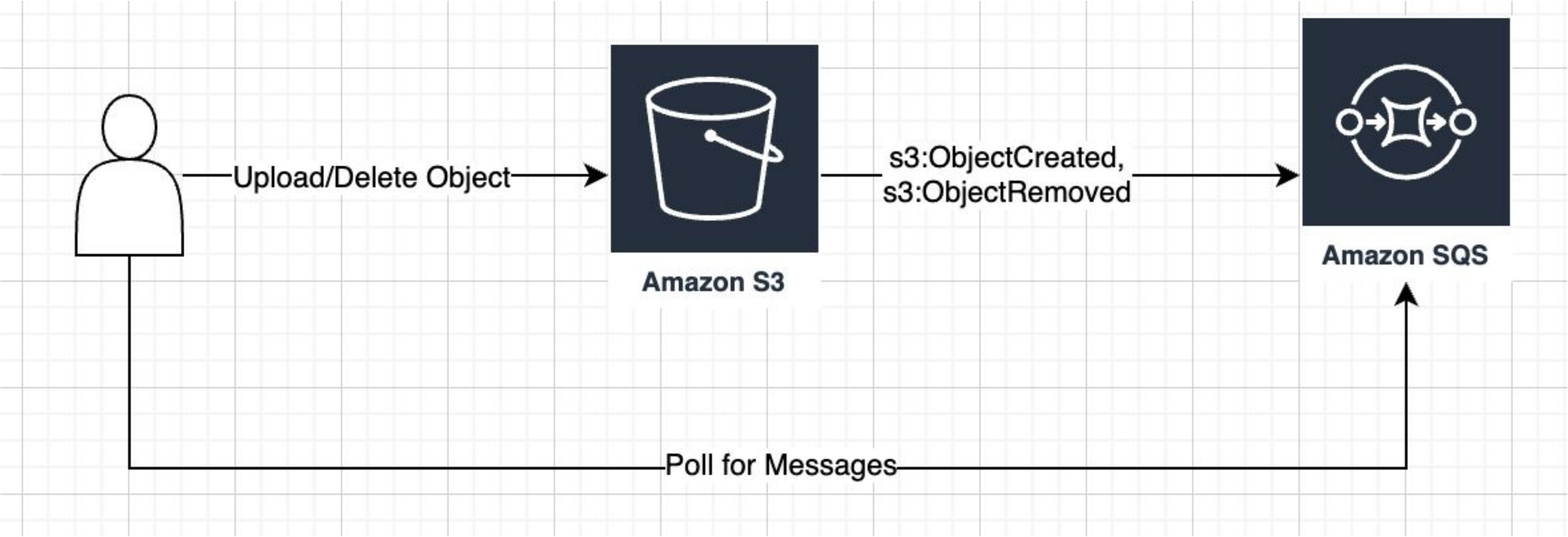
Scalability – Amazon SQS can process each buffered request independently, scaling transparently to handle any load increases or spikes without any provisioning instructions.

Reliability – Amazon SQS locks your messages during processing, so that multiple producers can send and multiple consumers can receive messages at the same time.

Customization – Your queues don't have to be exactly alike—for example, you can set a default delay on a queue. You can store the contents of messages larger than 256 KB using Amazon Simple Storage Service (Amazon S3) or Amazon DynamoDB, with Amazon SQS holding a pointer to the Amazon S3 object, or you can split a large message into smaller messages.

Try S3 event with SQS

Activity


Upload/Delete Object
Amazon S3
s3:ObjectCreated,
s3:ObjectRemoved
Amazon SQS
Poll for Messages

Activity

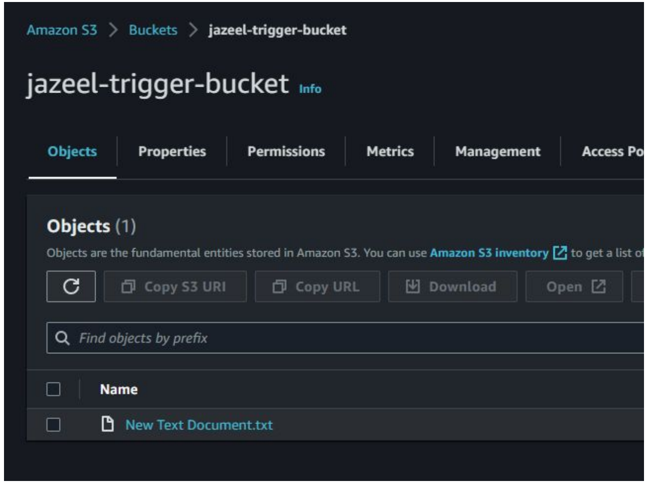
Demo: Creating above architecture with S3 and SQS using AWS Console

Objective:

1. Create an S3 bucket using terraform
2. Create an SQS queue using terraform with any IAM policies needed
3. Link an upload activity in S3 to SQS such that it will trigger an alert notification.
   a. Hint: Think of *aws_s3_bucket_notification*
4. Test your resources by going to your S3 bucket and uploading a non-sensitive file from your local machine. Open another browser tab that contains your SQS created too.
5. If successful, you will see alerts from polled messages coming into SQS from your S3

Activity

Once the terraform apply has completed, Go to AWS Console and go to the S3 bucket created and upload any file into it.



Amazon S3 > Buckets > jazeel-trigger-bucket

jazeel-trigger-bucket Info

Objects (1)

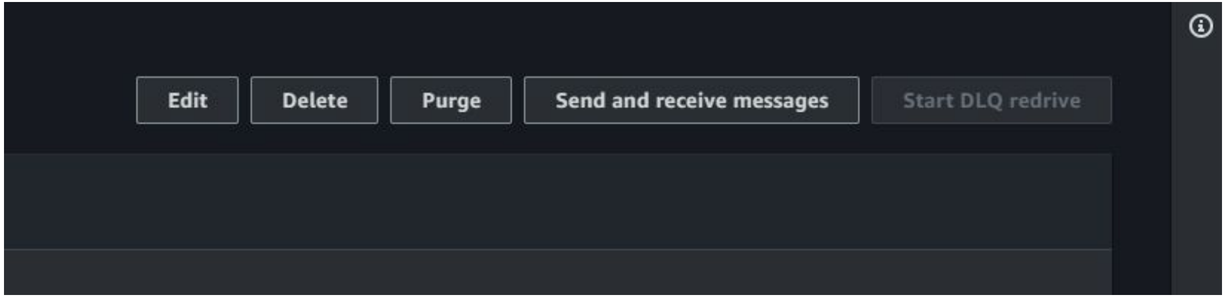
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of

Name

□ New Text Document.txt

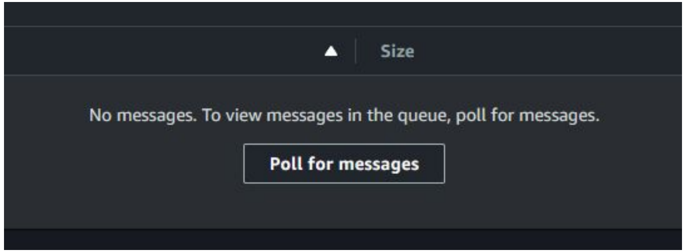
Activity

Once the upload has completed, Go to AWS SQS service and go to the SQS Queue that you created. And then click on “Send and Receive messages” on the top right

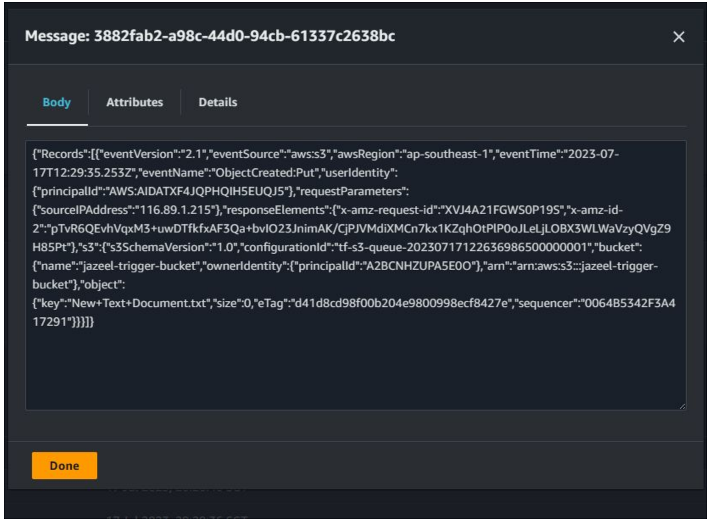


Activity

Click on “Poll for messages”



No messages. To view messages in the queue, poll for messages.



Message: 3882fab2-a98c-44d0-94cb-61337c2638bc

```
{"Records":[{"eventVersion":"2.1","eventSource":"aws:s3","awsRegion":"ap-southeast-1","eventTime":"2023-07-
17T12:29:35.253Z","eventName":"ObjectCreated:Put","userIdentity":
{"principalId":"AWS:AIDATXF4JQPHQIH5EUQJ5"},"requestParameters":
{"sourceIPAddress":"116.89.1.215"},"responseElements":{"x-amz-request-id":"XVJ4A21FGWS0P19S","x-amz-id-
2":"pTvR6QEvhVqxM3+uwDTfkfxAF3Qa+bvIO23JnimAK/CjPJVMdiXMCn7kx1KZqhOtPlP0oJLeLjLOBX3WLWaVzyQVgZ9
H85Pt"},"s3":{"s3SchemaVersion":"1.0","configurationId":"tf-s3-queue-20230717122636986500000001","bucket":
{"name":"jazeel-trigger-bucket","ownerIdentity":{"principalId":"A2BCNHZUPA5E0O"},"arn":"arn:aws:s3:::jazeel-trigger-
bucket"},"object":
{"key":"New+Text+Document.txt","size":0,"eTag":"d41d8cd98f00b204e9800998ecf8427e","sequencer":"0064B5342F3A4
17291"}}}]}
```

Trying it in Terraform

Activity

Create a new repo in Github and clone it to your local computer

Create your required files and file structure:

1) .gitignore -> terraform template
2) README.md
3) backend.tf
4) main.tf
5) provider.tf

Activity

Create a provider.tf file with region set to “us-east-1” <*Change as needed*>

Create a backend.tf file(**optional but recommended**) with bucket set to “sctp-ce3-tfstate-bucket-1” and region set to “us-east-1” .

Activity

Create a main.tf file using the guide below(NOTE: You will need to make some changes based on the guide below. Mostly the same):

https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/s3_bucket_notification#add-notification-configuration-to-sqs-queue

Activity

Once all files above have been created, Run the following commands:

```
terraform init

terraform plan

terraform apply
```

Questions?

Activity

Learner:

- Clean up AWS.
- Remove/delete/terminate all service/ resources that you created.

Instructor

- Clean up AWS.
- Remove/delete/terminate all service/ resources that you created.
- Check the AWS account after learner clean up.

END

Try S3 event for Lambda

Activity

Let spend 2-3 mins to

- Learner create new repository on github

Activity

Instructor demo how to create lambda function with S3 event on the new repository.

Activity

Learners to create lambda function with S3 event on the new repository

Break for 10-15 mins

Learners check new lambda function that has been created on AWS.

Activity

Instructor demo how to access Lambda Function log on AWS Cloudwatch

AWS X-Ray & AWS Cloudwatch

If still have more time. If not, there will be a session to cover this lesson in more detail (3.15 Application Logging - CloudWatch)

AWS Cloudwatch

Amazon CloudWatch collects and visualizes real-time logs, metrics, and event data in automated dashboards to streamline your infrastructure and application maintenance.

AWS Lambda automatically monitors Lambda functions on your behalf, pushing logs to Amazon CloudWatch. To help you troubleshoot failures in a function, after you set up permissions, Lambda logs all requests handled by your function and also automatically stores logs generated by your code through Amazon CloudWatch Logs.

You can insert logging statements into your code to help you validate that your code is working as expected. Lambda automatically integrates with CloudWatch Logs and pushes all logs from your code to a CloudWatch Logs group associated with a Lambda function, which is named /aws/lambda/<function name>.

There will be separated topic that will focus only on AWS Cloudwatch

AWS X-Ray

AWS X-Ray makes it easy for developers to **analyze** the behavior of their production, **distributed applications with end-to-end tracing capabilities**. You can use X-Ray to identify performance bottlenecks, edge case errors, and other hard to detect issues. This enables developers to quickly **find and address problems** in their applications and improve the experience for end users of their applications.

AWS X-Ray creates a **map of services** used by your application with **trace data** that you can use to drill into specific services or issues. This provides a view of connections between services in your application and aggregated data for each service, including average latency and failure rates.

Activity

Instructor demo how to access Lambda Function log on AWS Cloudwatch

To-do

- update service name on serverless-yml
-