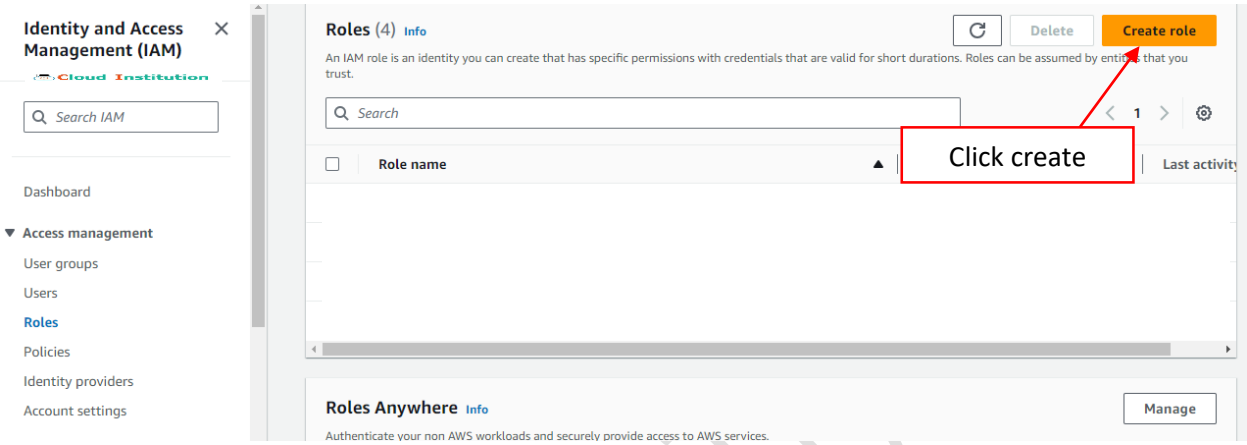
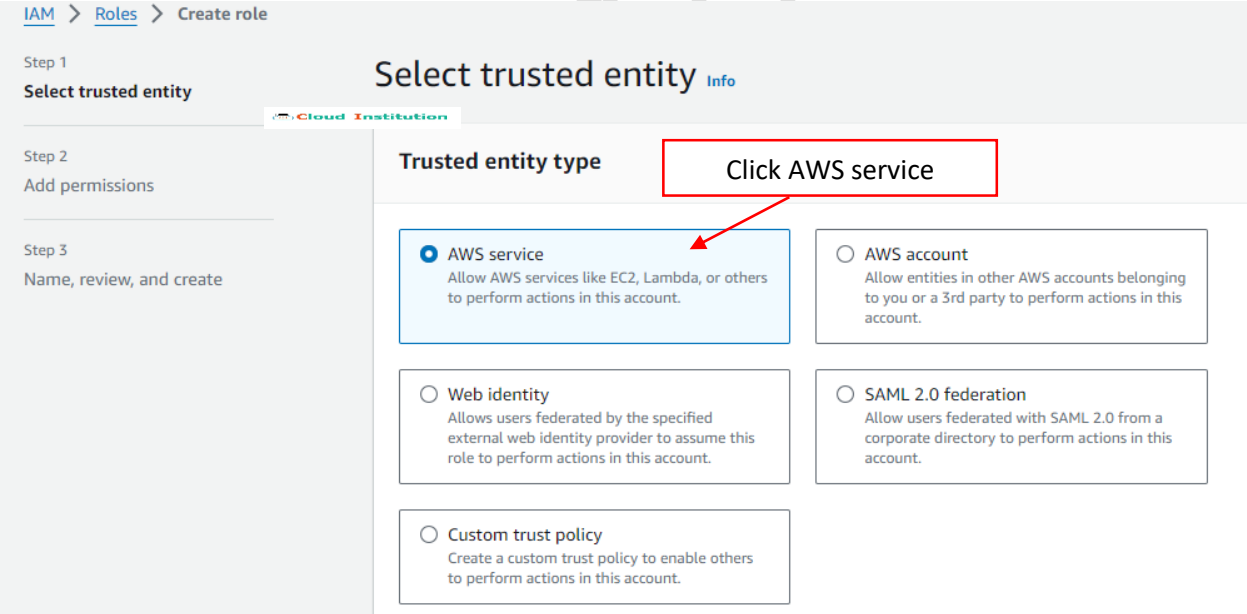
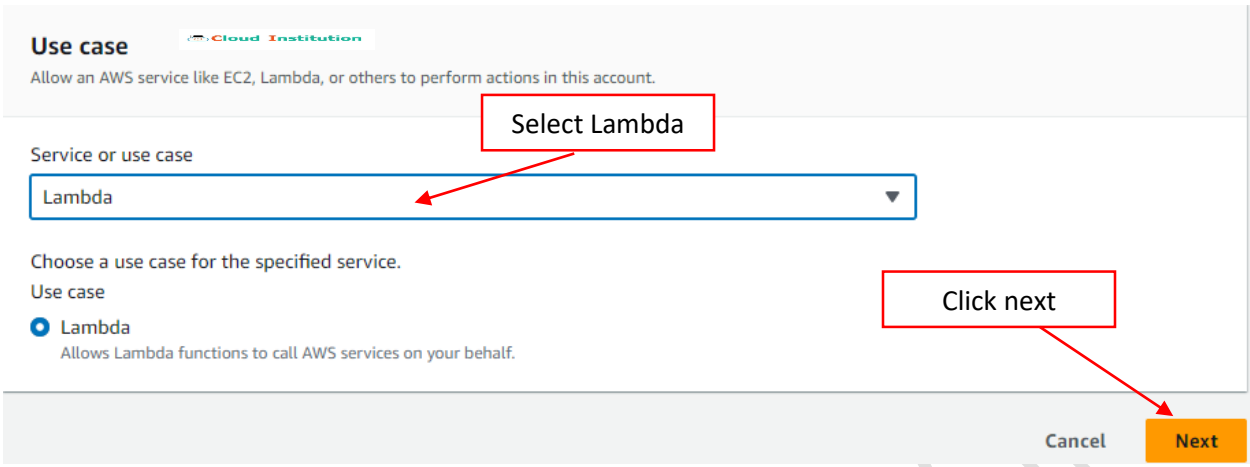
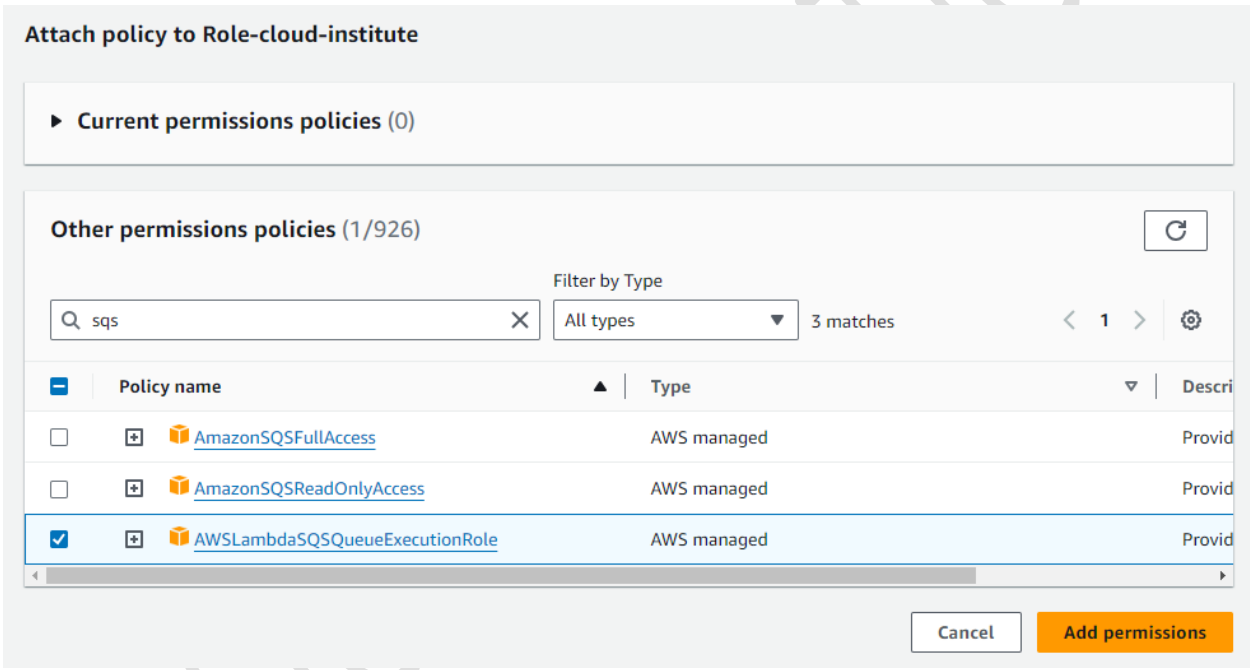
TRIGGERING SQS USING AWS LAMBDA

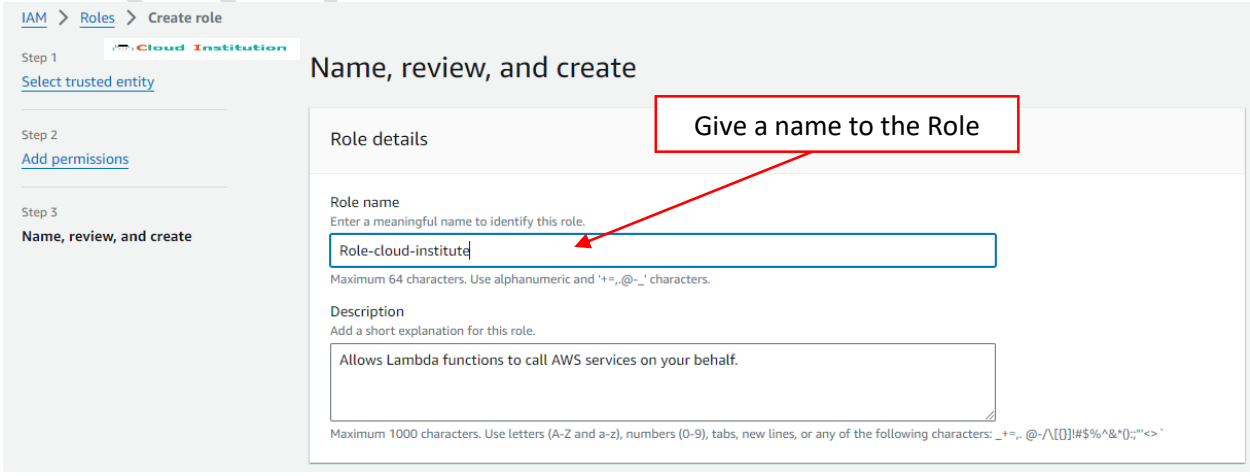
Step 1 : Create a IAM Role

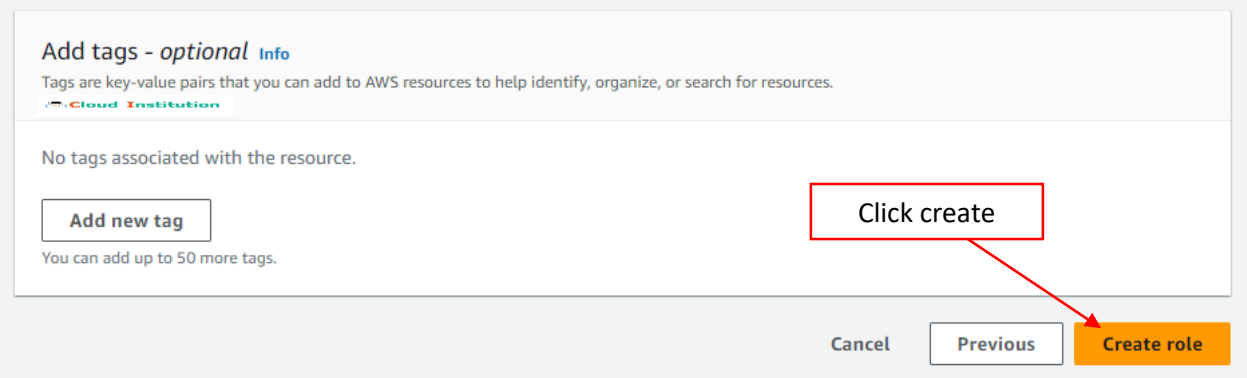




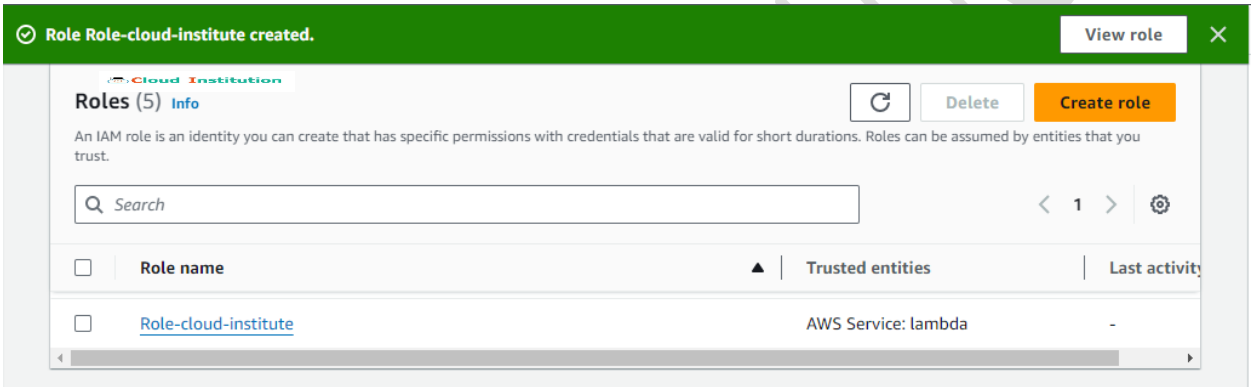

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Select Lambda
Service or use case
Lambda
Choose a use case for the specified service.
Use case
Lambda
Allows Lambda functions to call AWS services on your behalf.
Click next
Cancel
Next


Attach policy to Role-cloud-institute
Current permissions policies (0)
Other permissions policies (1/926)
Filter by Type
sqs
All types
3 matches
Policy name
Type
AmazonSQSFullAccess
AWS managed
AmazonSQSReadOnlyAccess
AWS managed
AWSLambdaSQSQueueExecutionRole
AWS managed
Cancel
Add permissions

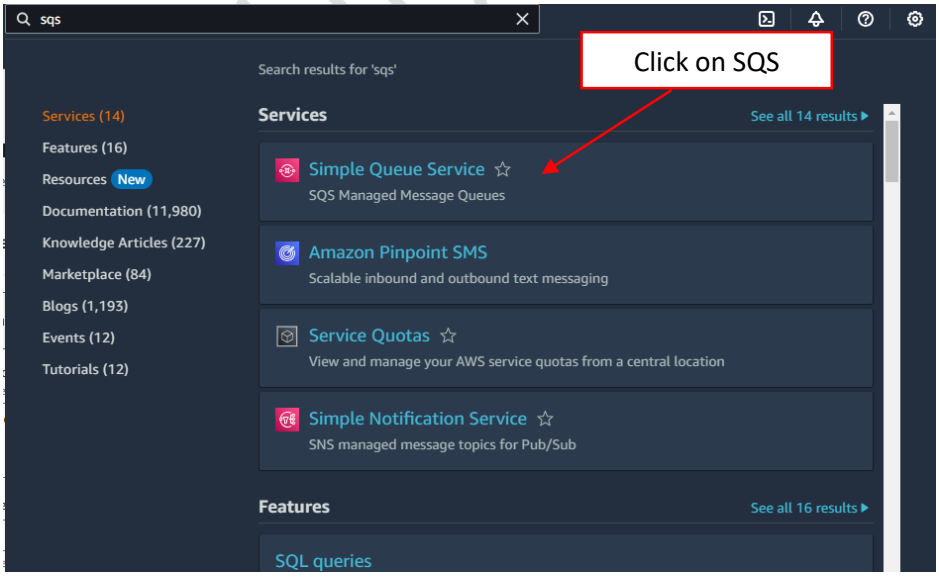

IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Give a name to the Role
Role name
Enter a meaningful name to identify this role.
Role-cloud-institute
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^&*();"'<> `

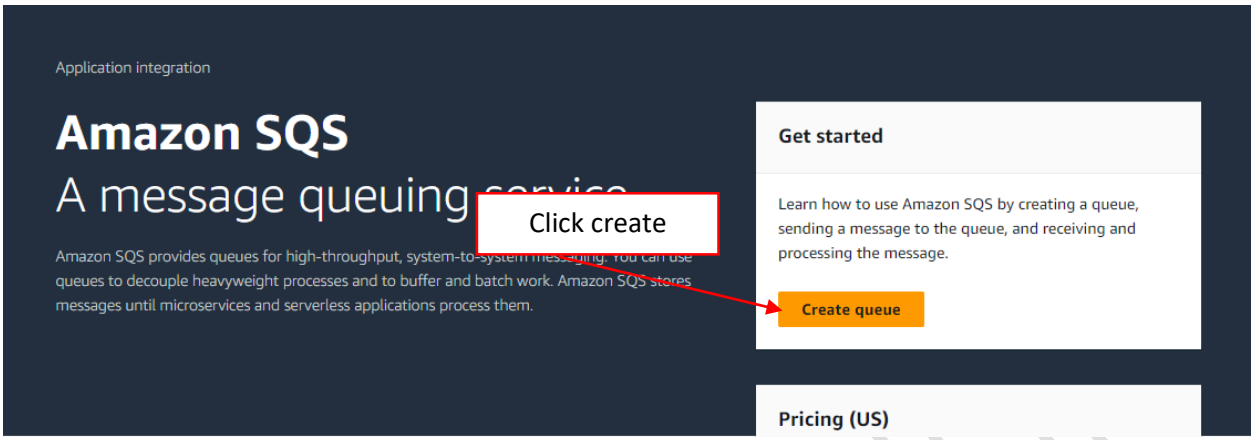


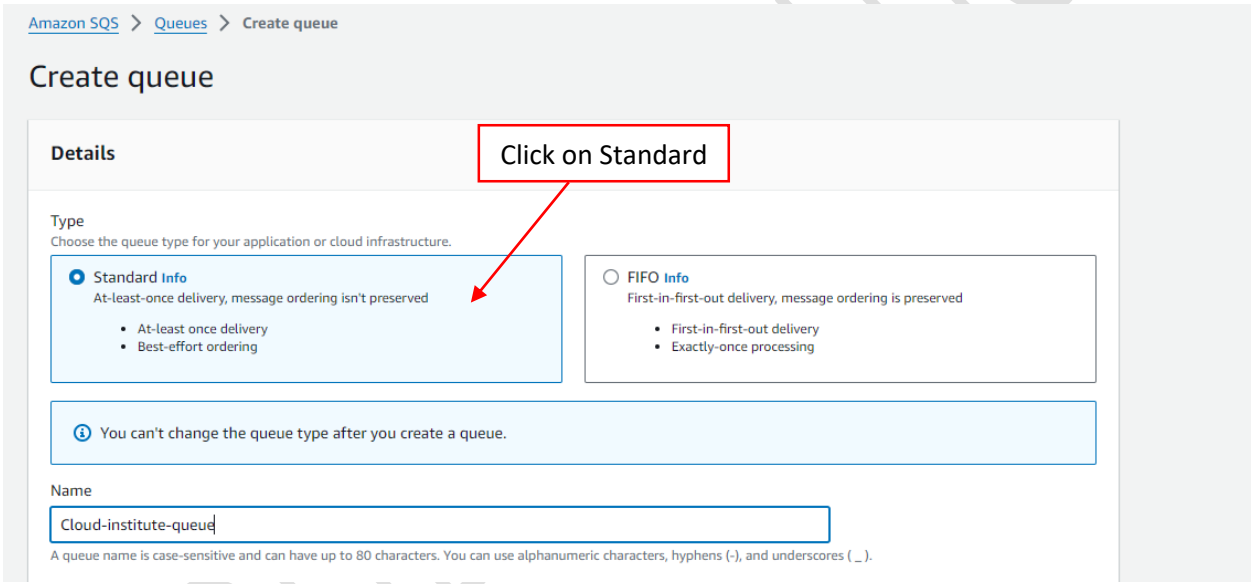
Role Created

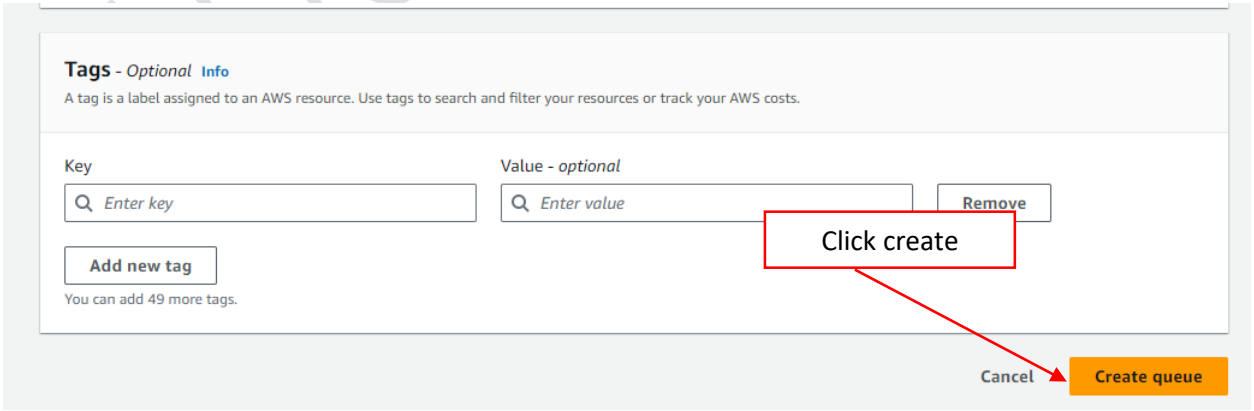


Step 2 : Create a SQS

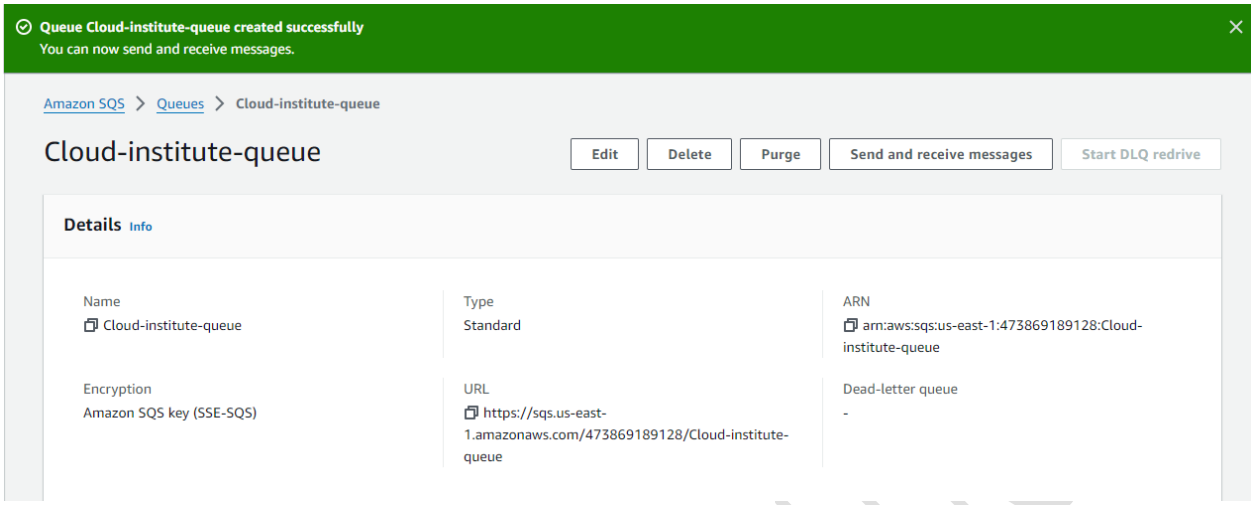



Application integration
Amazon SQS
A message queuing service
Amazon SQS provides queues for high-throughput, system-to-system messaging. You can use queues to decouple heavyweight processes and to buffer and batch work. Amazon SQS stores messages until microservices and serverless applications process them.
Click create
Get started
Learn how to use Amazon SQS by creating a queue, sending a message to the queue, and receiving and processing the message.
Create queue
Pricing (US)

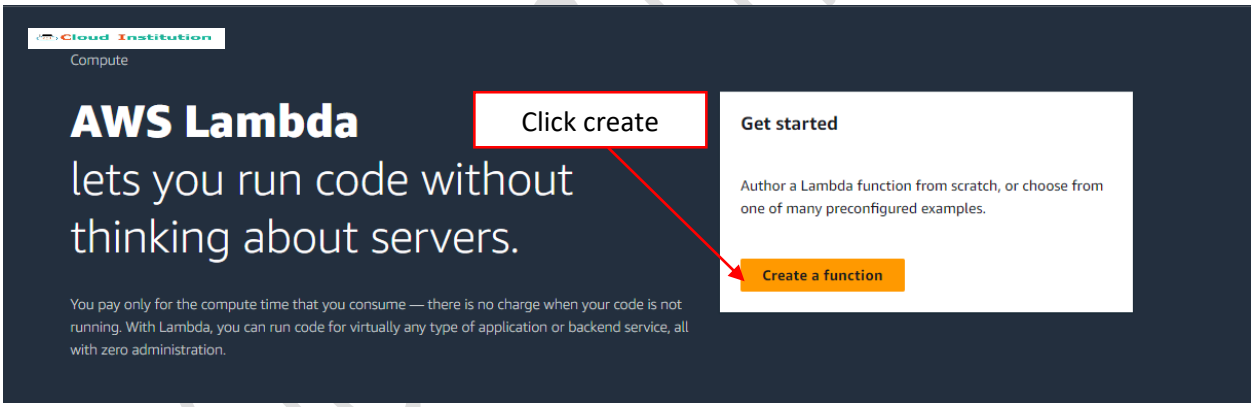

Amazon SQS > Queues > Create queue
Create queue
Details
Click on Standard
Type
Choose the queue type for your application or cloud infrastructure.
Standard Info
At-least-once delivery, message ordering isn't preserved
At-least once delivery
Best-effort ordering
FIFO Info
First-in-first-out delivery, message ordering is preserved
First-in-first-out delivery
Exactly-once processing
You can't change the queue type after you create a queue.
Name
Cloud-institute-queue
A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

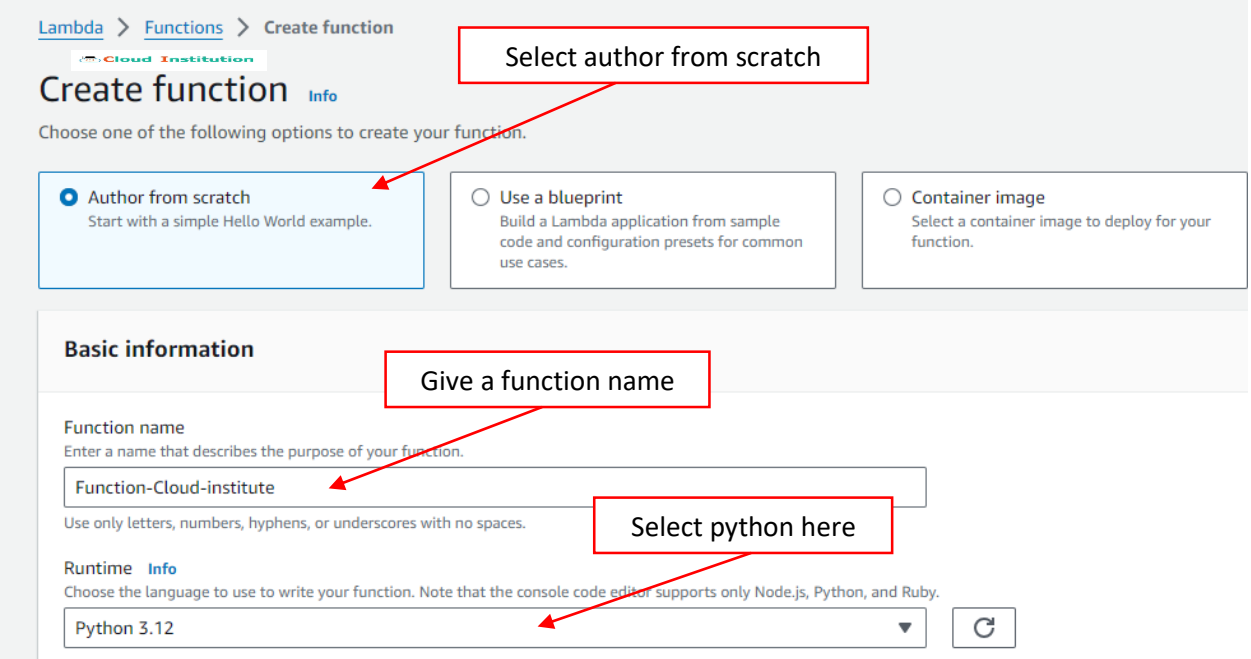

Tags - Optional Info
A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs.
Key
Enter key
Value - optional
Enter value
Remove
Add new tag
You can add 49 more tags.
Click create
Cancel
Create queue

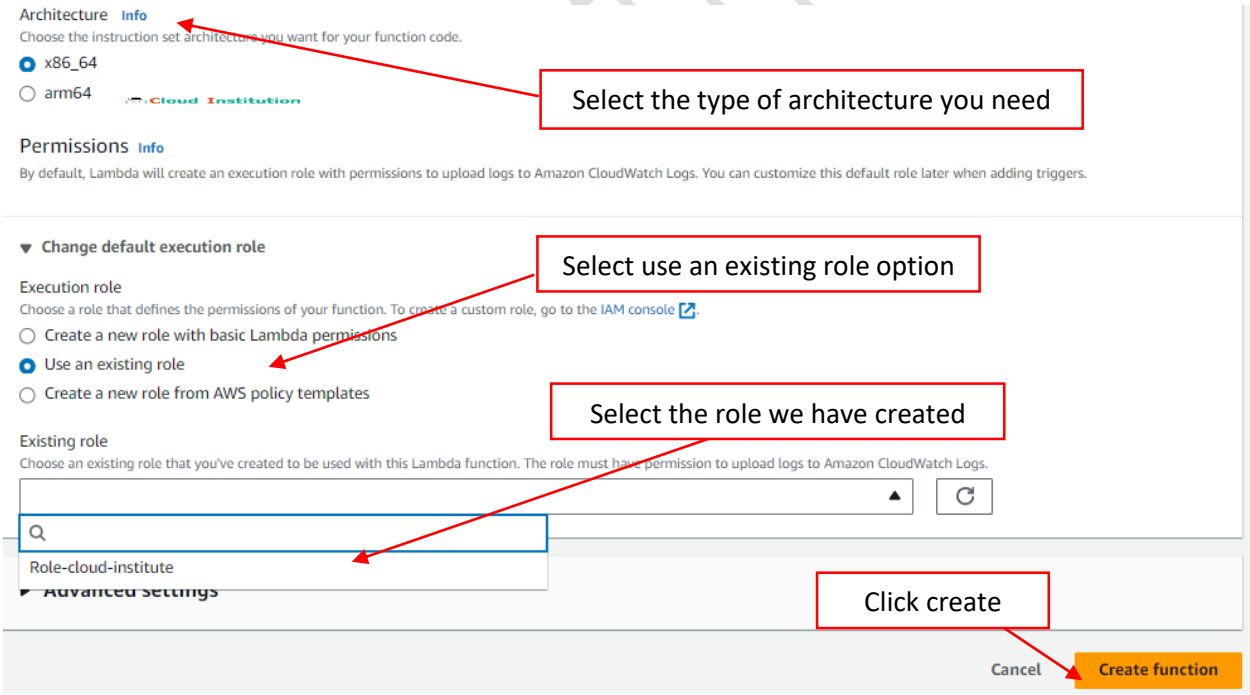
Queue created



Step 3: Create a Lambda function

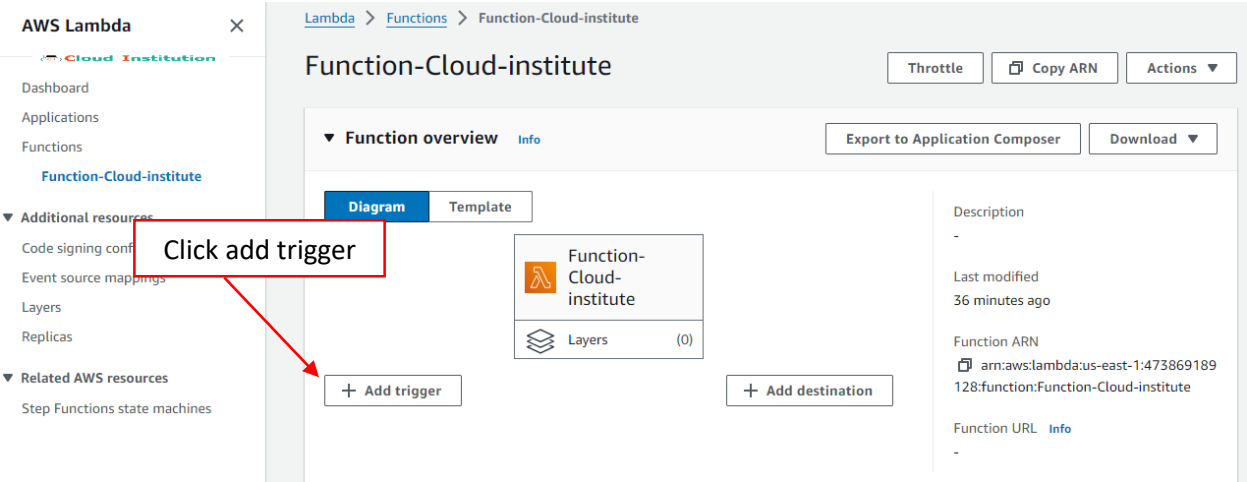


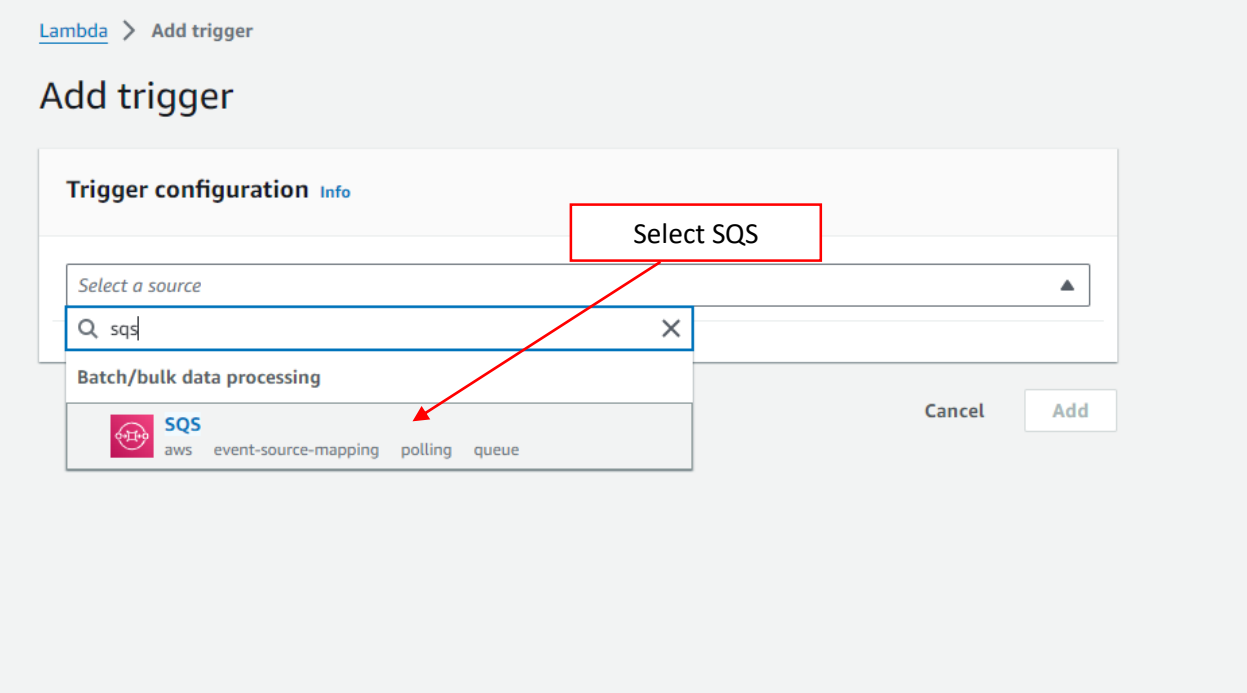


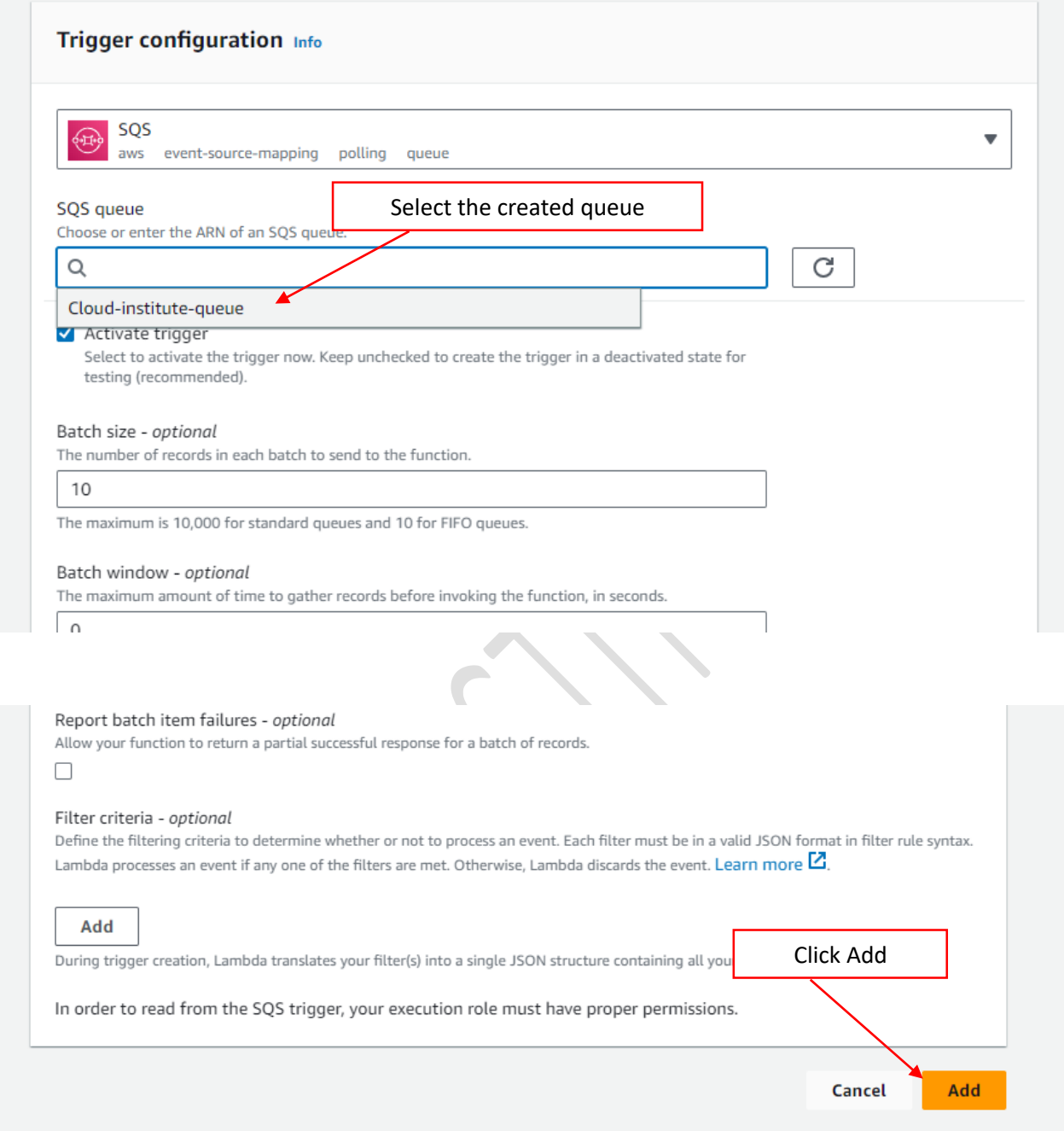


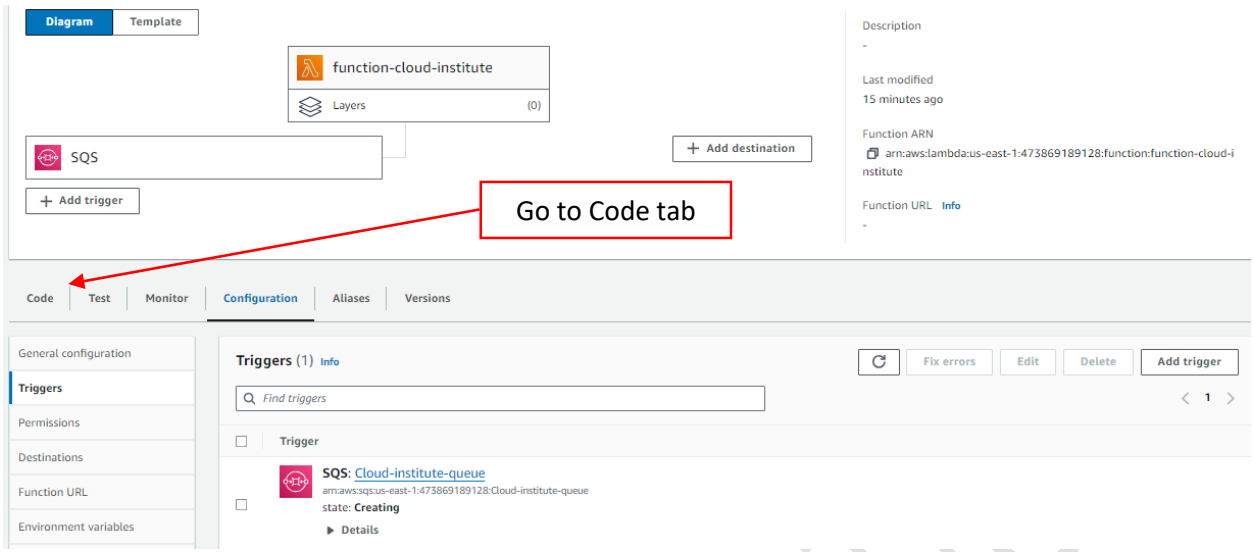
Function created

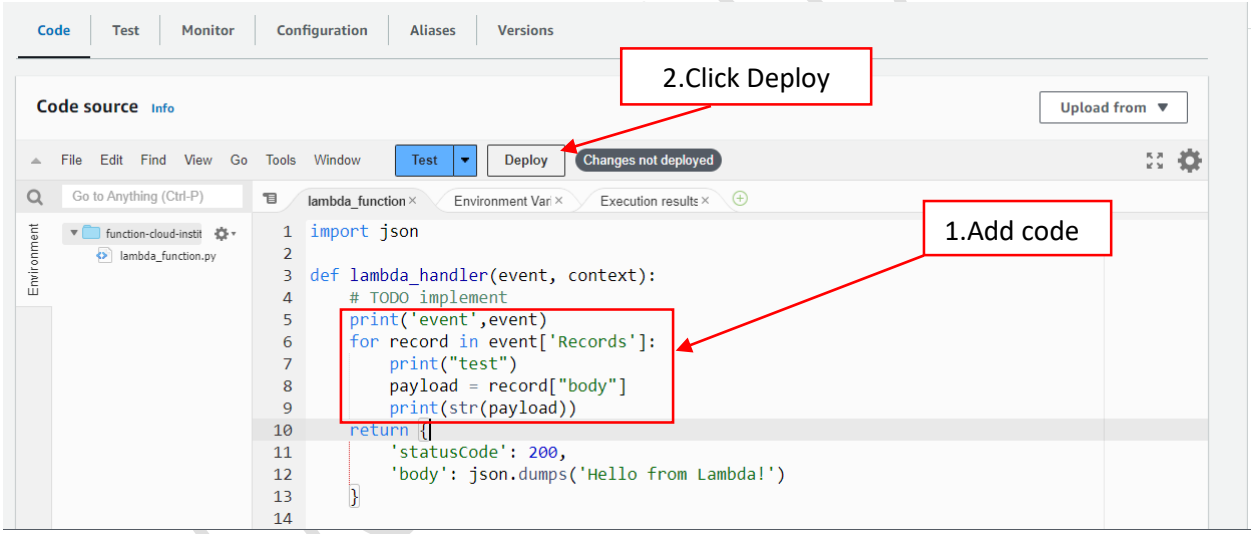
Step : 4 Triggering SNS





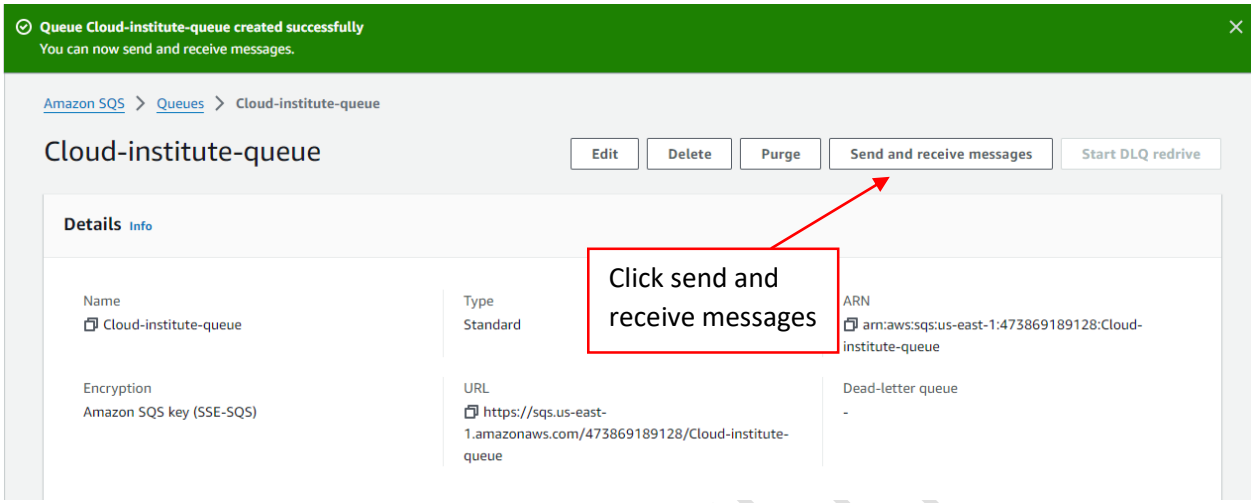

Trigger configuration Info
SQS
aws event-source-mapping polling queue
SQS queue
Choose or enter the ARN of an SQS queue.
Select the created queue
Cloud-institute-queue
Activate trigger
Select to activate the trigger now. Keep unchecked to create the trigger in a deactivated state for testing (recommended).
Batch size - optional
The number of records in each batch to send to the function.
10
The maximum is 10,000 for standard queues and 10 for FIFO queues.
Batch window - optional
The maximum amount of time to gather records before invoking the function, in seconds.
Report batch item failures - optional
Allow your function to return a partial successful response for a batch of records.
Filter criteria - optional
Define the filtering criteria to determine whether or not to process an event. Each filter must be in a valid JSON format in filter rule syntax.
Lambda processes an event if any one of the filters are met. Otherwise, Lambda discards the event. Learn more
Add
During trigger creation, Lambda translates your filter(s) into a single JSON structure containing all you
Click Add
In order to read from the SQS trigger, your execution role must have proper permissions.
Cancel
Add

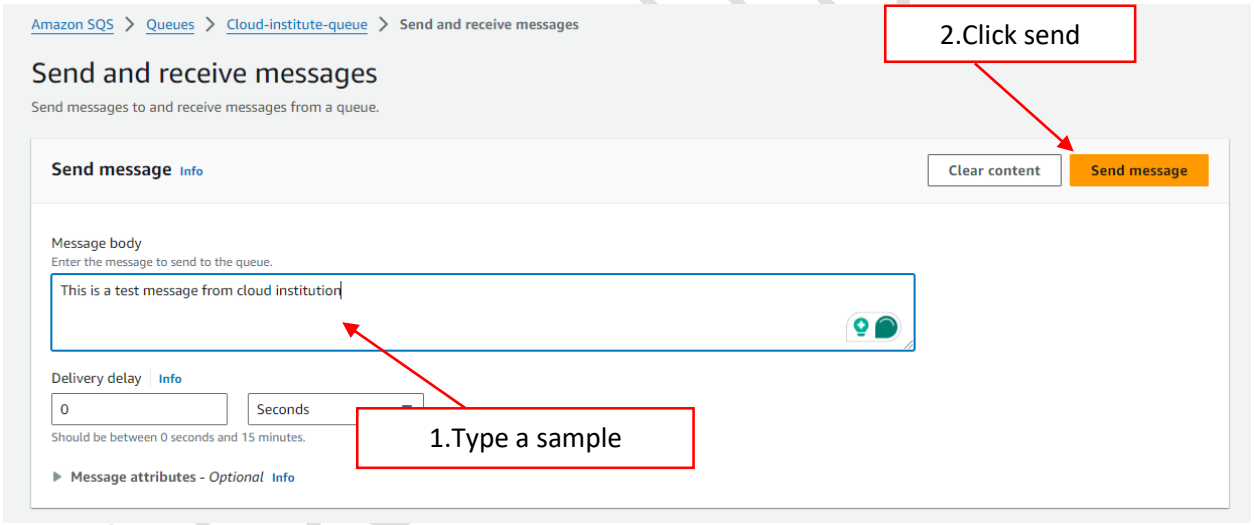


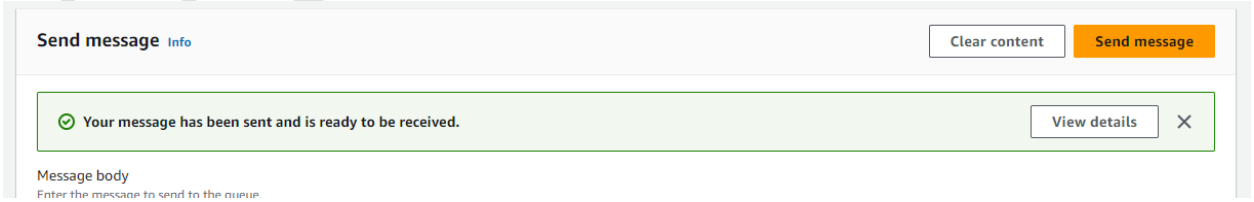



```
print('event',event)
    for record in event['Records']:
        print("test")
        payload = record["body"]
        print(str(payload))
```

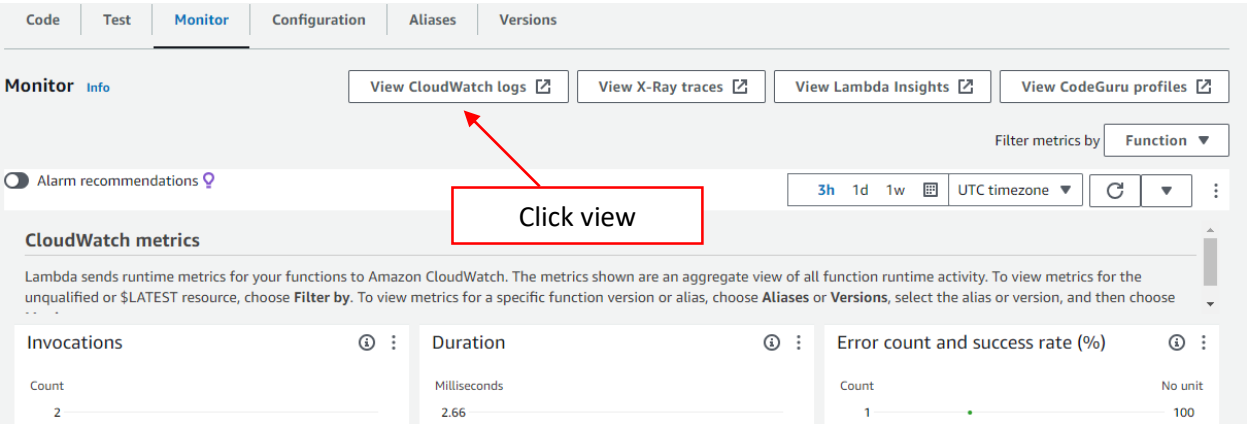
Navigate to the SQS

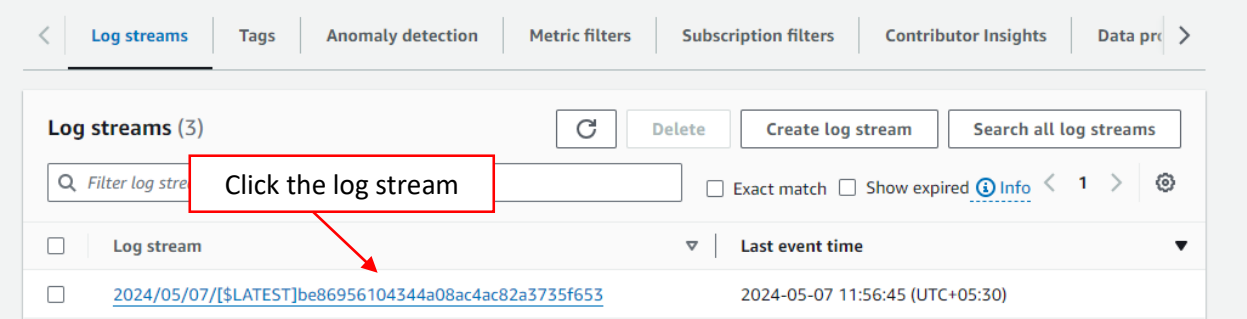


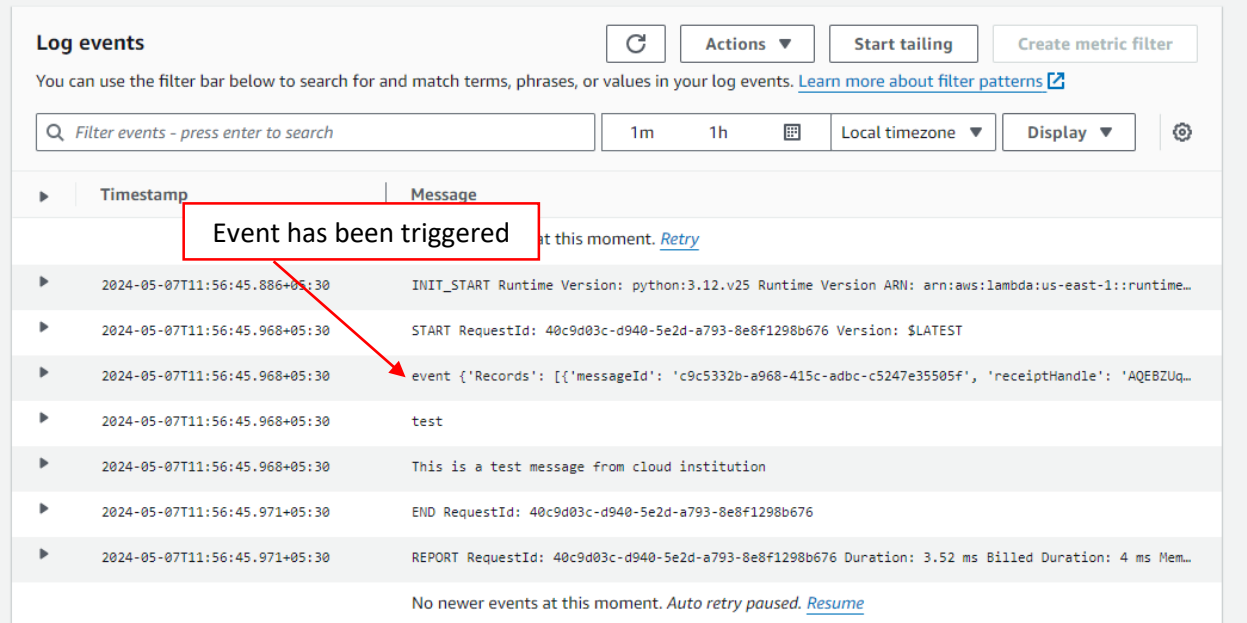




Now go back to lambda functions







CLOUD INSTITUTE