

Task - 2

Name: Jenin Krishna K P

Location: Chennai

Real-time data processing pipeline for an e-commerce application :

The screenshot shows the AWS DynamoDB Stream Details page for the 'Orders' table. The stream is currently active ('On'). It has a 'Resource-based policy' attached. The 'Latest stream ARN' is listed as `arn:aws:dynamodb:ap-south-1:297186792789:table/Orders/stream/2025-11-04T10:00:10.863`. There are no triggers defined for this stream.

The screenshot shows the AWS DynamoDB Explore Items page for the 'Orders' table. A successful scan operation named 'UserSubmissions' is displayed, returning 2 items. The scan started at 15:54:44 on November 6, 2025. The results table shows two entries:

orderId (String)	amount	createdAt	customerEmail
ORDER301	2000	2025-11-0...	vip@example.com
ORDER201	250	2025-11-0...	user@example.c...

S3 buckets | S3 | ap-south-1 Items | Amazon DynamoDB Manager StandardOrderProcessor | Function

<https://ap-south-1.console.aws.amazon.com/lambda/home/?region=ap-south-1#/functions/StandardOrderProcessor?tab=code>

aws Search [Alt+S] Account ID: 2971-8679-2789 Jenin%20Krishna Asia Pacific (Mumbai)

Lambda > Functions > StandardOrderProcessor

Code source Info

EXPLORER STANDARDORDERPROCESSOR lambda_function.py

DEPLOY [UNDEPLOYED CHANGES]

- You have undeployed changes.
- Deploy (Ctrl+Shift+U)
- Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]

- Create new test event

Open in Visual Studio Code Upload from

```

lambda_function.py
1 import json
2 def lambda_handler(event, context):
3     # EventBridge Pipes sends a list of records
4     if isinstance(event, list):
5         records = event
6     else:
7         records = [event]
8     for record in records:
9         new_image = record.get('dynamodb', {}).get('NewImage', {})
10
11         order_id = new_image.get('orderId', {}).get('S')
12         status = new_image.get('status', {}).get('S')
13         amount_str = new_image.get('amount', {}).get('N')
14         email = new_image.get('customerEmail', {}).get('S')
15
16         try:
17             amount = float(amount_str)
18         except (TypeError, ValueError):

```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 4:04 PM 11/6/2025

S3 buckets | S3 | ap-south-1 Items | Amazon DynamoDB Manager PremiumOrderProcessor | Function

<https://ap-south-1.console.aws.amazon.com/lambda/home/?region=ap-south-1#/functions/PremiumOrderProcessor?tab=code>

aws Search [Alt+S] Account ID: 2971-8679-2789 Jenin%20Krishna Asia Pacific (Mumbai)

Lambda > Functions > PremiumOrderProcessor

Code Test Monitor Configuration Aliases Versions

Code source Info

EXPLORER PREMIUMORDERPROCESSOR lambda_function.py

DEPLOY [UNDEPLOYED CHANGES]

- You have undeployed changes.
- Deploy (Ctrl+Shift+U)
- Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]

Open in Visual Studio Code Upload from

```

lambda_function.py
1 import json
2 def lambda_handler(event, context):
3     premium_orders = []
4     if isinstance(event, list):
5         records = event
6     else:
7         records = [event]
8     for record in records:
9         dynamodb = record.get('dynamodb', {})
10        new = dynamodb.get('NewImage', {})
11        old = dynamodb.get('OldImage', {})
12
13        order_id = new.get('orderId', {}).get('S')
14        new_status = new.get('status', {}).get('S')
15        old_status = old.get('status', {}).get('S')

```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 4:05 PM 11/6/2025

S3 buckets | S3 | ap-south-1 | Items | Amazon DynamoDB Man | Amazon EventBridge | ap-south-1 | +

https://ap-south-1.console.aws.amazon.com/events/home?region=ap-south-1#/pipe/OrdersStandardPipe

aws Search [Alt+S] Account ID: 2971-8679-2789 Asia Pacific (Mumbai) Jenin%20Krishna

Amazon EventBridge > Pipes > OrdersStandardPipe

OrdersStandardPipe

Details Info

Name: OrdersStandardPipe	Status: Running	Created on: Nov 4, 2025, 05:29 PM GMT+5:30
Description:	Status reason: No records processed	Last modified on: Nov 4, 2025, 05:29 PM GMT+5:30
ARN: arn:aws:pipes:ap-south-1:297186792789:pipe/OrdersStandardPipe		

Pipe Components Info

```
graph LR; Source[DynamoDB Stream] --> Filtering[Filtering]; Filtering --> Target[Lambda function]
```

Source (Required): DynamoDB Stream

Filtering (Optional): Added

Target (Required): Lambda function

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 3:56 PM 11/6/2025

S3 buckets | S3 | ap-south-1 | Items | Amazon DynamoDB Man | Amazon EventBridge | ap-south-1 | +

https://ap-south-1.console.aws.amazon.com/events/home?region=ap-south-1#/pipe/OrdersStandardPipe

aws Search [Alt+S] Account ID: 2971-8679-2789 Asia Pacific (Mumbai) Jenin%20Krishna

Amazon EventBridge > Pipes > OrdersStandardPipe

Execution Role

service-role/Amazon_EventBridge_Pipe_OrdersStandardPipe_5f1bc60f

Retry policy and Dead-letter queue (DLQ)

Dead-Letter Queue: orders-standard-dlq

SQS Queue ARN: arn:aws:sqs:ap-south-1:297186792789:orders-standard-dlq

Retry Policy:

- Retry attempts: -1 attempts
- Maximum Age of Event: 59m

CloudWatch logs

Log destination: CloudWatch logs

Log level: ERROR

Log execution data: OFF

CloudWatch log group: /aws/vendedlogs/pipes/OrdersStandardPipe

ARN: arn:aws:logs:ap-south-1:297186792789:log-group:/aws/vendedlogs/pipes/OrdersStandardPipe

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 3:56 PM 11/6/2025

S3 buckets | S3 | ap-south-1 | Items | Amazon DynamoDB Man | Amazon EventBridge | ap-south-1 | +

https://ap-south-1.console.aws.amazon.com/events/home?region=ap-south-1#/pipe/OrdersPremiumPipe

aws Search [Alt+S] Account ID: 2971-8679-2789 Jenin%20Krishna Asia Pacific (Mumbai)

Amazon EventBridge > Pipes > OrdersPremiumPipe

OrdersPremiumPipe

Details Info

Name	Status	Created on
OrdersPremiumPipe	Running	Nov 4, 2025, 05:38 PM GMT+5:30
Description	Status reason	Last modified on
No records processed	ARN	Nov 4, 2025, 05:38 PM GMT+5:30
	arn:aws:pipes:ap-south-1:297186792789:pipe/OrdersPremiumPipe	

Pipe Components Info

```
graph LR; Source[DynamoDB Stream] --> Filtering[Filtering  
Optional  
Added]; Filtering --> Target[Lambda function]
```

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback 3:55 PM 11/6/2025

S3 buckets | S3 | ap-south-1 | Items | Amazon DynamoDB Man | Amazon EventBridge | ap-south-1 | +

https://ap-south-1.console.aws.amazon.com/events/home?region=ap-south-1#/pipe/OrdersPremiumPipe

aws Search [Alt+S] Account ID: 2971-8679-2789 Jenin%20Krishna Asia Pacific (Mumbai)

Amazon EventBridge > Pipes > OrdersPremiumPipe

Execution Role

service-role/Amazon_EventBridge_Pipe_OrdersPremiumPipe_ba6c90f5

Retry policy and Dead-letter queue (DLQ)

Dead-Letter Queue: orders-premium-dlq
SQS Queue: arn:aws:sqs:ap-south-1:297186792789:orders-premium-dlq

Retry Policy:

- Retry attempts: -1 attempts
- Maximum Age of Event: 59m

CloudWatch logs

Log destination: CloudWatch logs
Log level: ERROR
Log execution data: OFF

CloudWatch log group: /aws/vendedlogs/pipes/OrdersPremiumPipe
ARN: arn:aws:logs:ap-south-1:297186792789:log-group:/aws/vendedlogs/pipes/OrdersPremiumPipe

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback 3:55 PM 11/6/2025

S3 buckets | S3 | ap-south-1 | Items | Amazon DynamoDB Manager | Queue details | Simple Queue Service | +

https://ap-south-1.console.aws.amazon.com/sqs/v3/home?region=ap-south-1#queues/https%3A%2F%2Fsqs.ap-south-1.amazonaws.com%2F297186792789... JK ...

aws Search [Alt+S] Account ID: 2971-8679-2789 Asia Pacific (Mumbai) Jenin%20Krishna

Amazon SQS > Queues > orders-standard-dlq

orders-standard-dlq

Edit Delete Purge Send and receive messages Start DLQ redrive

Details Info

Name orders-standard-dlq	Type Standard	ARN arn:aws:sqs:ap-south-1:297186792789:orders-standard-dlq
Encryption Amazon SQS key (SSE-SQS)	URL https://sqs.ap-south-1.amazonaws.com/297186792789/orders-standard-dlq	Dead-letter queue

More

Queue policies Monitoring SNS subscriptions Lambda triggers EventBridge Pipes Dead-letter queue Tagging Encryption

Access policy Info Define who can access your queue Edit

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3:56 PM 11/6/2025

S3 buckets | S3 | ap-south-1 | Items | Amazon DynamoDB Manager | Log Groups | CloudWatch | ap-south-1 | +

https://ap-south-1.console.aws.amazon.com/cloudwatch/home?region=ap-south-1#logsV2:log-groups/log-group/\$252Faws\$252Flambda\$252FStandardOrderProcessor

aws Search [Alt+S] Account ID: 2971-8679-2789 Asia Pacific (Mumbai) Jenin%20Krishna

CloudWatch > Log groups > /aws/lambda/StandardOrderProcessor

CloudWatch

- Favorites and recents
- Dashboards
- AI Operations
- Alarms
- Logs
- Metrics
- Application Signals (APM)

Log groups

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

Log streams (2)

By default, we only load the most recent log streams.

Log stream	Last event time
2025/11/04/[\$LATEST]256c4d1877d94b5380db3bbaa5e5abc6	2025-11-04 12:24:00 (UTC)
2025/11/04/[\$LATEST]54635694ca4a03b3244ccf11a3dbd6	2025-11-04 12:15:36 (UTC)

Log streams (2) Delete Create log stream Search all log streams

Filter log streams or try prefix search Exact match Show expired

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3:57 PM 11/6/2025

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar navigation includes 'CloudWatch' (selected), 'Favorites and recents', 'Dashboards', 'AI Operations', 'Alarms', 'Logs' (selected), 'Log groups' (selected), 'Log Anomalies', 'Live Tail', 'Logs Insights', 'Contributor Insights', 'Metrics', and 'Application Signals'. The main content area is titled 'Log events' and displays log entries for a Lambda function. The first entry is an INIT event:

```
INIT_START Runtime Version: python:3.12.v89 Runtime Version ARN: arn:aws:lambda:ap-south-1::runtime:644f999c44280c3dd580a0b580d80576fe347cc4d4089106e2c52ed35b626e0fb3c
```

The second entry is a START event:

```
START RequestId: 95e4491e-4306-431b-84e8-f6fff6cfa1576 Version: $LATEST
```

The third entry is a PROCESSING event:

```
Processing order ORDER301 | Status=shipped | Amount=2000.0 | Email=vip@example.com
```

The fourth entry is an END event:

```
END RequestId: 95e4491e-4306-431b-84e8-f6fff6cfa1576
```

The fifth entry is a REPORT event:

```
REPORT RequestId: 95e4491e-4306-431b-84e8-f6fff6cfa1576 Duration: 1.98 ms Billed Duration: 92 ms Memory Size: 128 MB Max Memory Used: 37 MB Init Duration: 89.35 ms
```

No newer events are present at the moment.

The screenshot shows the AWS Simple Queue Service (SQS) Queue details interface for the 'orders-premium-dlq' queue. The top navigation bar includes 'CloudWatch' (selected), 'Items | Amazon DynamoDB Manager', and 'Queue details | Simple Queue Service'. The main content area shows the queue's details:

Name	orders-premium-dlq	Type	Standard		
Encryption	Amazon SQS key (SSE-SQS)	URL	https://sqs.ap-south-1.amazonaws.com/297186792789/orders-premium-dlq	Dead-letter queue	-

Below the details, there are tabs for 'Queue policies' (selected), 'Monitoring', 'SNS subscriptions', 'Lambda triggers', 'EventBridge Pipes', 'Dead-letter queue', 'Tagging', and 'Encryption'. The 'Access policy' section allows defining who can access the queue. The bottom navigation bar includes 'CloudShell' and 'Feedback'.

The screenshot shows the AWS CloudWatch Log streams page. On the left, a sidebar menu includes CloudWatch, Favorites and recents, Dashboards, AI Operations, Alarms, Logs (selected), Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights, Metrics, and Application Signals (APM). The main content area displays a log group for '/aws/lambda/PremiumOrderProcessor'. It shows a single log stream named '2025/11/04/[\$LATEST]ec3cba4e38a24a13b70834796be35b6'. The stream was created 2 days ago and has never expired. It contains 539 bytes of stored data. The 'Log streams' tab is selected. Other tabs include Tags, Anomaly detection, Metric filters, Subscription filters, Contributor Insights, and Data.

The screenshot shows the AWS CloudWatch Log events page. The sidebar menu is identical to the previous screenshot. The main content area displays log events for the same log group. The events show the initialization of the Lambda function with runtime information (python:3.12.v89, ARN: arn:aws:lambda:ap-south-1:runtime:644f999c4428c3dd580a0b58d0576fe347c4d088916e2c52ed35b626e0fb3c) and a successful execution of a premium order (RequestId: 4d7023ca-7411-4cb4-8eee-ff96e16af025, Version: \$LATEST). The event details include the order ID (ORDER301), amount (\$200.0), and the fact that it was shipped. The log also shows the end of the request (RequestId: 4d7023ca-7411-4cb4-8eee-ff96e16af025) and the report of the execution duration (1.97 ms), billed duration (95 ms), memory size (128 MB), and maximum memory used (36 MB). A message at the bottom indicates no newer events are present.