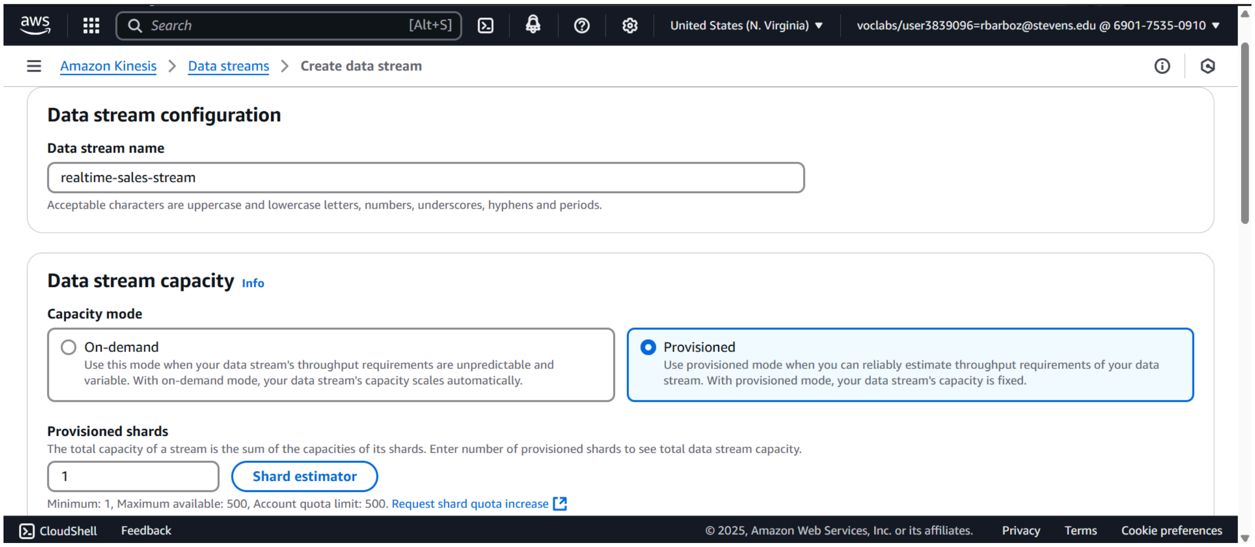
Lab 4: Real-Time Data Streaming with Amazon Kinesis & CloudWatch

Step 1: Create a Kinesis Data Stream


aws
Search [Alt+S]
United States (N. Virginia) ▼
voclabs/user3839096=rbarboz@stevens.edu @ 6901-7535-0910 ▼
Amazon Kinesis > Data streams > Create data stream

Data stream configuration

Data stream name

realtime-sales-stream

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.

Data stream capacity Info

Capacity mode

○ On-demand
Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.

● Provisioned
Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

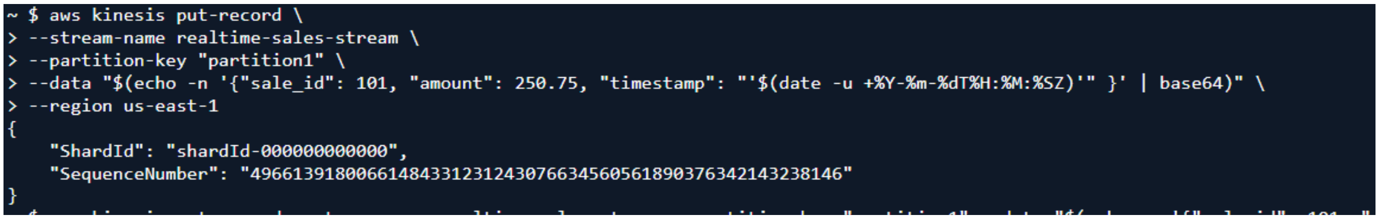
Provisioned shards

The total capacity of a stream is the sum of the capacities of its shards. Enter number of provisioned shards to see total data stream capacity.

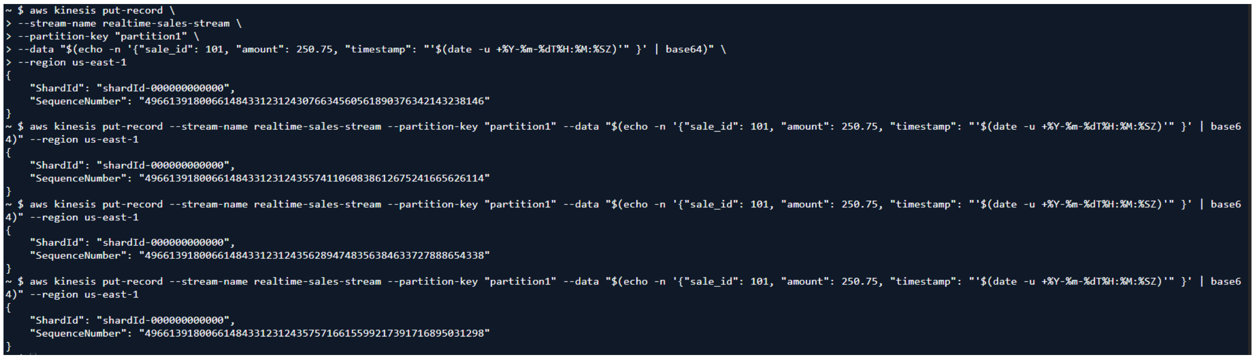
1

Minimum: 1, Maximum available: 500, Account quota limit: 500. Request shard quota increase

Step 2: Send Data to Kinesis Stream (AWS CLI)

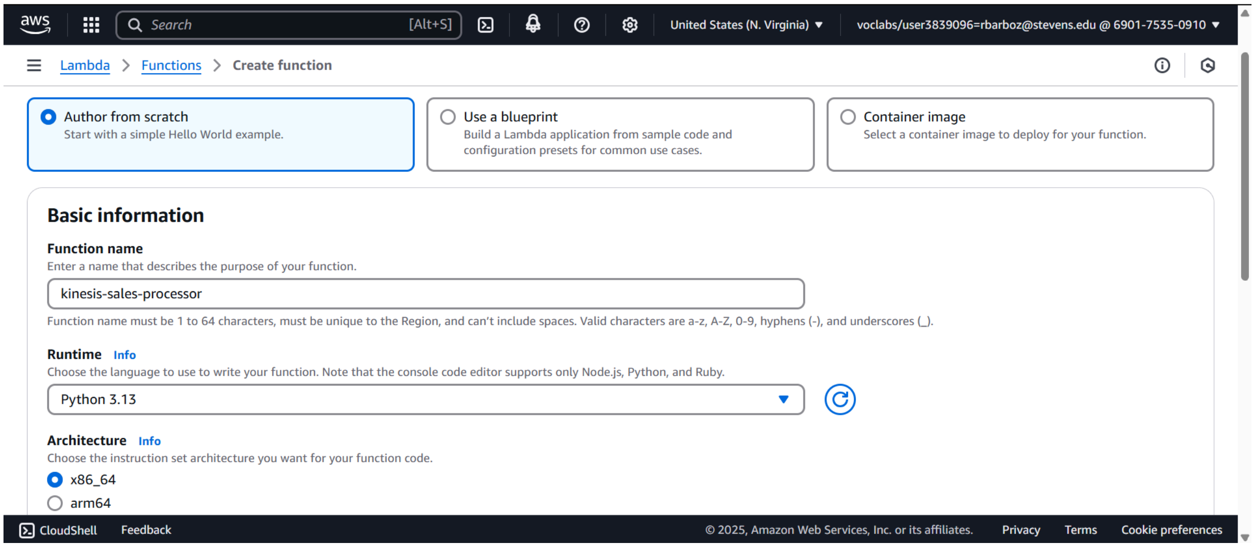


```
~ $ aws kinesis put-record \
> --stream-name realtime-sales-stream \
> --partition-key "partition1" \
> --data "$(echo -n '{"sale_id": 101, "amount": 250.75, "timestamp": "'$(date -u +%Y-%m-%dT%H:%M:%SZ)'" }' | base64)" \
> --region us-east-1
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49661391800661484331231243076634560561890376342143238146"
}
```



```
~ $ aws kinesis put-record \
> --stream-name realtime-sales-stream \
> --partition-key "partition1" \
> --data "$(echo -n '{"sale_id": 101, "amount": 250.75, "timestamp": "'$(date -u +%Y-%m-%dT%H:%M:%SZ)'" }' | base64)" \
> --region us-east-1
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49661391800661484331231243076634560561890376342143238146"
}
~ $ aws kinesis put-record --stream-name realtime-sales-stream --partition-key "partition1" --data "$(echo -n '{"sale_id": 101, "amount": 250.75, "timestamp": "'$(date -u +%Y-%m-%dT%H:%M:%SZ)'" }' | base6
4)" --region us-east-1
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49661391800661484331231243557411060838612675241665626114"
}
~ $ aws kinesis put-record --stream-name realtime-sales-stream --partition-key "partition1" --data "$(echo -n '{"sale_id": 101, "amount": 250.75, "timestamp": "'$(date -u +%Y-%m-%dT%H:%M:%SZ)'" }' | base6
4)" --region us-east-1
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49661391800661484331231243562894748356384633727888654338"
}
~ $ aws kinesis put-record --stream-name realtime-sales-stream --partition-key "partition1" --data "$(echo -n '{"sale_id": 101, "amount": 250.75, "timestamp": "'$(date -u +%Y-%m-%dT%H:%M:%SZ)'" }' | base6
4)" --region us-east-1
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49661391800661484331231243575716615599217391716895031298"
}
```

Step 3: Create a Lambda Function to Process Stream Data


aws
Search [Alt+S]
United States (N. Virginia) ▾
voclabs/user3839096=rbarboz@stevens.edu @ 6901-7535-0910 ▾
Lambda > Functions > Create function
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

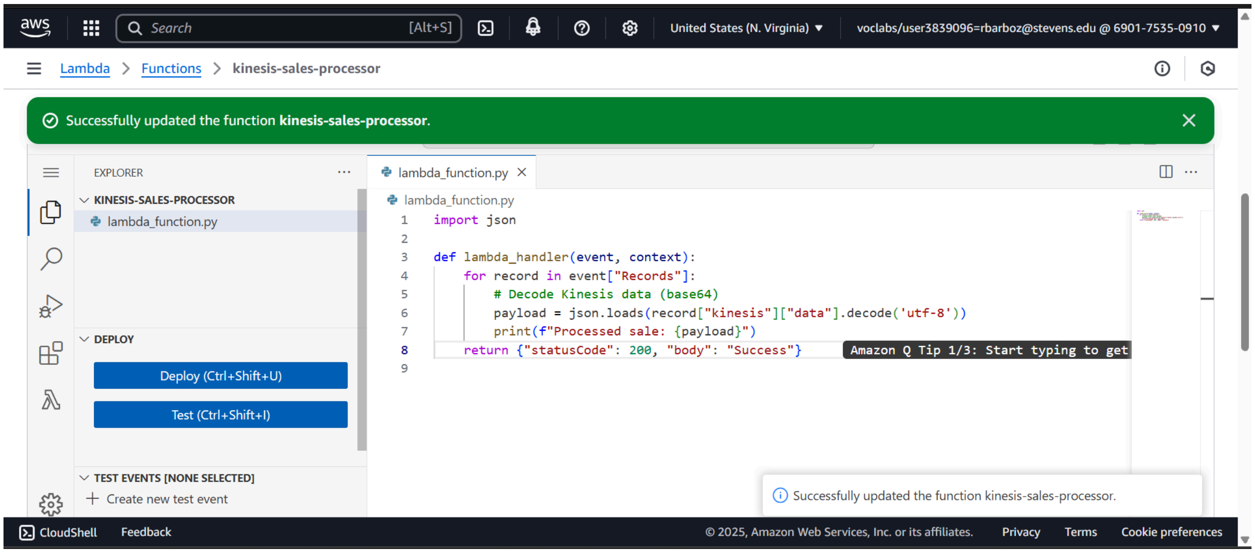
○ Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
kinesis-sales-processor
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64


aws
Search [Alt+S]
United States (N. Virginia) ▾
voclabs/user3839096=rbarboz@stevens.edu @ 6901-7535-0910 ▾
Lambda > Functions > kinesis-sales-processor
EXPLORER
KINESIS-SALES-PROCESSOR
lambda_function.py
DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)
TEST EVENTS [NONE SELECTED]
+ Create new test event
lambda_function.py ×
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Successfully updated the function **kinesis-sales-processor**.

lambda_function.py

```python
import json

def lambda_handler(event, context):
    for record in event["Records"]:
        # Decode Kinesis data (base64)
        payload = json.loads(record["kinesis"]["data"].decode('utf-8'))
        print(f"Processed sale: {payload}")
    return {"statusCode": 200, "body": "Success"}
```

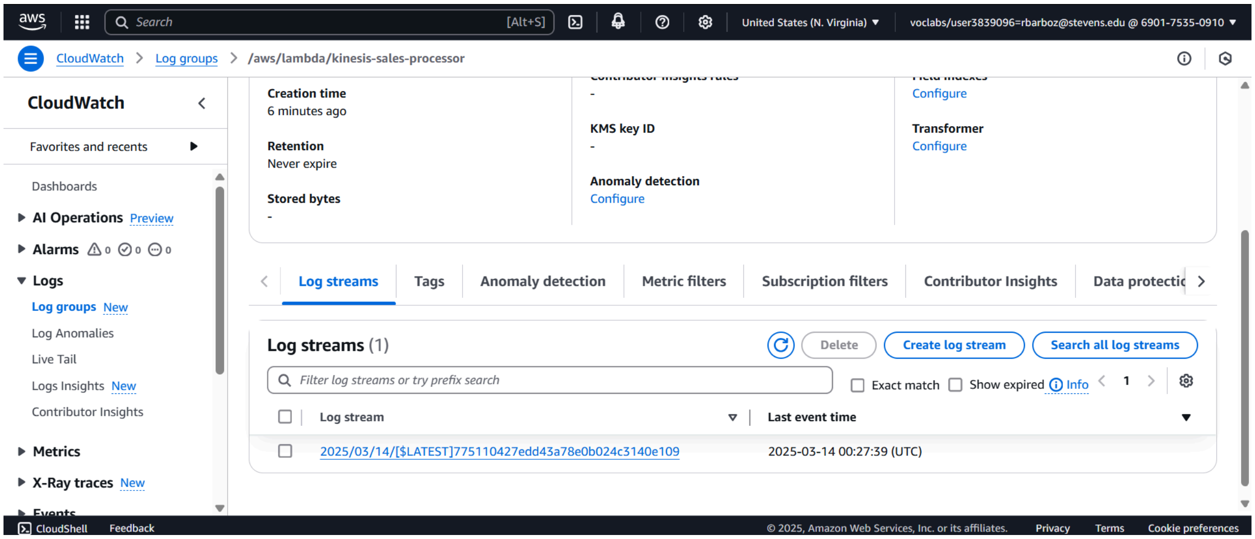
Amazon Q Tip 1/3: Start typing to get

Successfully updated the function kinesis-sales-processor.

Step 4:

Attached Lambda to Kinesis Stream and in trigger I Select realtime-sales-stream in Lambda settings

Step 5: Monitor Kinesis Stream Data in CloudWatch Logs Insights



Creation time
6 minutes ago

Retention
Never expire

Stored bytes
-

Contributor Insights rules
-

KMS key ID
-

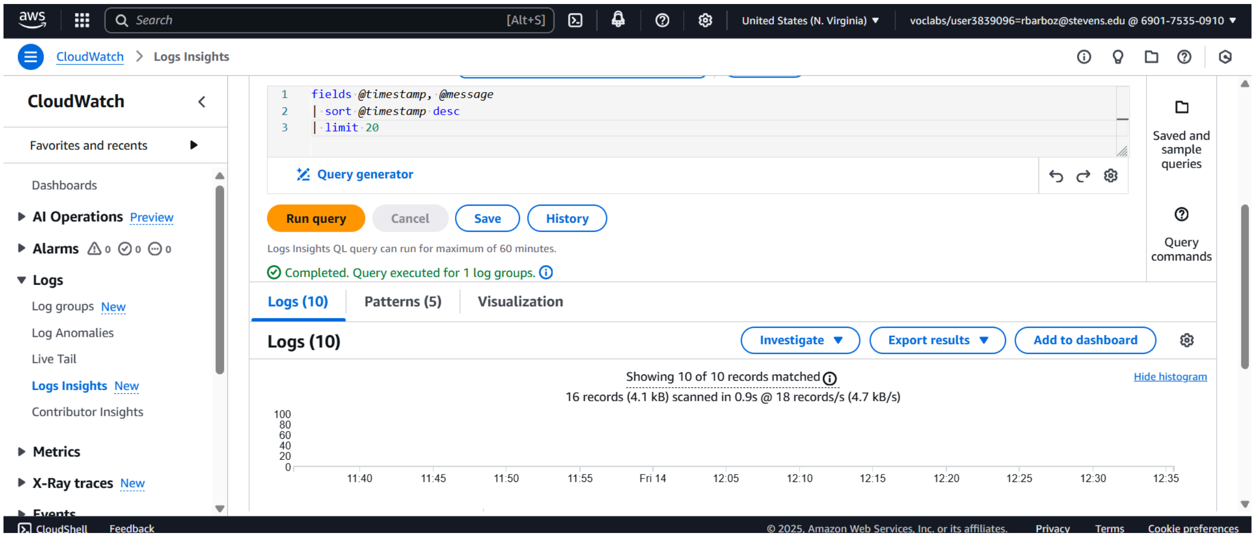
Anomaly detection
Configure

Field indexes
Configure

Transformer
Configure

Log streams (1)

□	Log stream	Last event time
□	2025/03/14/[$LATEST]775110427edd43a78e0b024c3140e109	2025-03-14 00:27:39 (UTC)



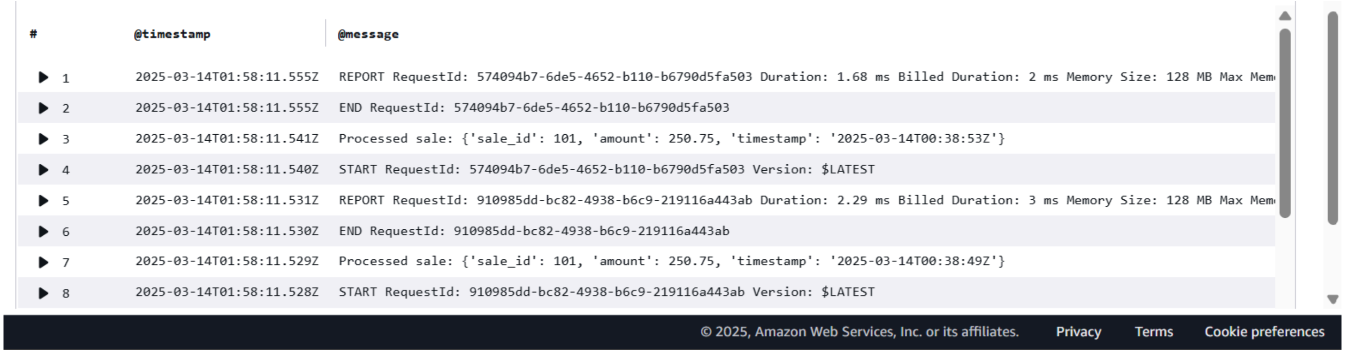
```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
```

Logs Insights QL query can run for maximum of 60 minutes.

✓ Completed. Query executed for 1 log groups.

Logs (10)

Showing 10 of 10 records matched
16 records (4.1 kB) scanned in 0.9s @ 18 records/s (4.7 kB/s)

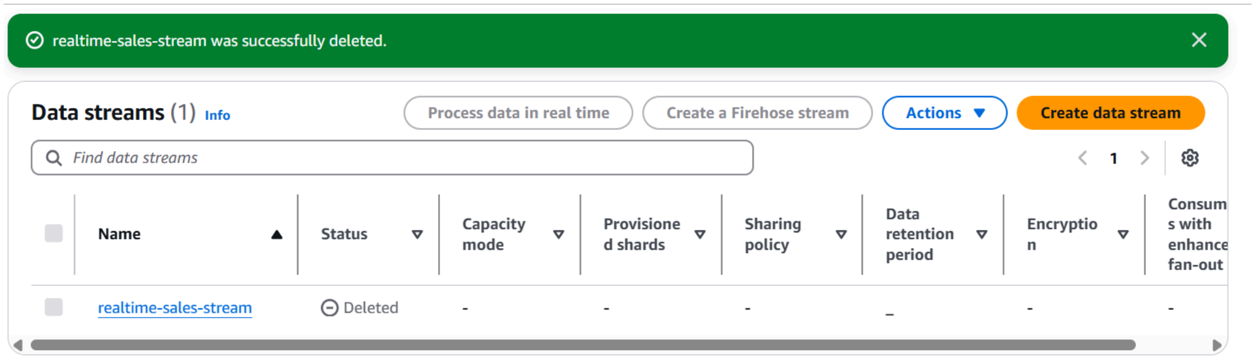


```
#        @timestamp                @message

▶ 1      2025-03-14T01:58:11.555Z  REPORT RequestId: 574094b7-6de5-4652-b110-b6790d5fa503 Duration: 1.68 ms Billed Duration: 2 ms Memory Size: 128 MB Max Mem
▶ 2      2025-03-14T01:58:11.555Z  END RequestId: 574094b7-6de5-4652-b110-b6790d5fa503
▶ 3      2025-03-14T01:58:11.541Z  Processed sale: {'sale_id': 101, 'amount': 250.75, 'timestamp': '2025-03-14T00:38:53Z'}
▶ 4      2025-03-14T01:58:11.540Z  START RequestId: 574094b7-6de5-4652-b110-b6790d5fa503 Version: $LATEST
▶ 5      2025-03-14T01:58:11.531Z  REPORT RequestId: 910985dd-bc82-4938-b6c9-219116a443ab Duration: 2.29 ms Billed Duration: 3 ms Memory Size: 128 MB Max Mem
▶ 6      2025-03-14T01:58:11.530Z  END RequestId: 910985dd-bc82-4938-b6c9-219116a443ab
▶ 7      2025-03-14T01:58:11.529Z  Processed sale: {'sale_id': 101, 'amount': 250.75, 'timestamp': '2025-03-14T00:38:49Z'}
▶ 8      2025-03-14T01:58:11.528Z  START RequestId: 910985dd-bc82-4938-b6c9-219116a443ab Version: $LATEST
```

Step 6: Cleanup

1. Delete Kinesis Stream:

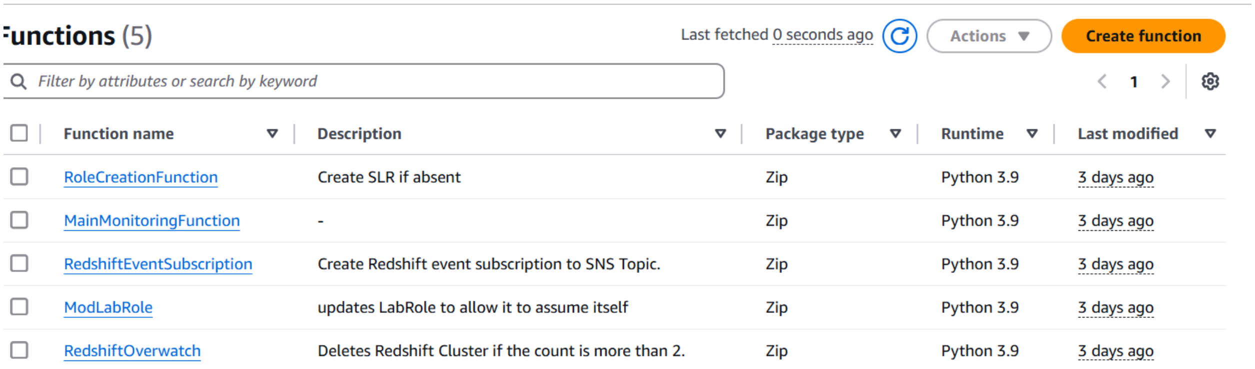


realtime-sales-stream was successfully deleted.

Data streams (1) Info

□	Name ▲	Status	Capacity mode	Provisioned shards	Sharing policy	Data retention period	Encryption	Consumers with enhanced fan-out
□	realtime-sales-stream	⊖ Deleted	-	-	-	_	-	-

2. Delete Lambda Function:

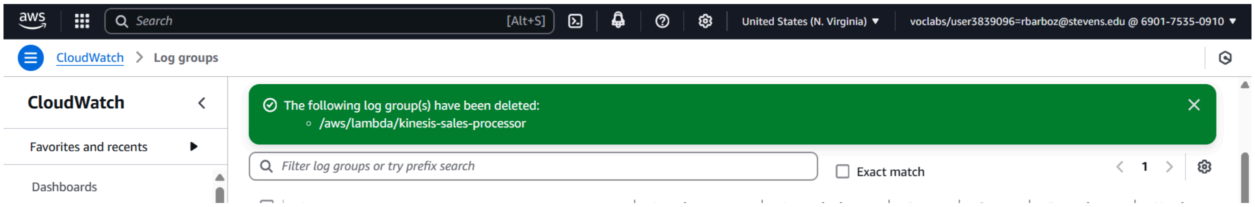


Functions (5)

Last fetched 0 seconds ago

□	Function name	Description	Package type	Runtime	Last modified
□	RoleCreationFunction	Create SLR if absent	Zip	Python 3.9	3 days ago
□	MainMonitoringFunction	-	Zip	Python 3.9	3 days ago
□	RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.9	3 days ago
□	ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.9	3 days ago
□	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.9	3 days ago

3. Delete CloudWatch Log Group:


aws
Search [Alt+S]
United States (N. Virginia) ▼
voclabs/user3839096=rbarboz@stevens.edu @ 6901-7535-0910 ▼
CloudWatch > Log groups
CloudWatch
Favorites and recents ▶
Dashboards
Filter log groups or try prefix search
□ Exact match
< 1 >

The following log group(s) have been deleted:
- /aws/lambda/kinesis-sales-processor