Name: Abdurrahman Qureshi

Roll No: 242466

Practical No: 5

Date Of Performance: 06/09/2025

Aim: To create an automated serverless system that processes new files uploaded to an S3 bucket by triggering a Lambda function, which then has the necessary IAM permissions to interact with a DynamoDB table.

1. What is AWS Lambda?
2. What is serverless computing?
3. What languages does AWS Lambda support?
4. What are AWS DynamoDB Table?
5. Explain AWS IAM service.
6. To understand AWS Lambda, create your first Lambda functions using Python / Java / Nodejs. Create AWs Lambda function and configure a trigger for Amazon Simple Storage Service (Amazon S3). The trigger invokes your Lambda function every time that you add an object

to your Amazon S3 bucket. Allow AWS Lambda to access Amazon DynamoDB Table. Create IAM role that allows full access to DynamoDB Table

[Terminate the resources after performing the practical]

ANS.1

AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You simply upload your code, and Lambda automatically runs and scales it in response to triggers like HTTP requests or file uploads.

ANS.2

Serverless computing is a cloud-native development model where developers can build and run applications without managing servers, as the cloud provider handles the routine work of provisioning, maintaining, and scaling the server infrastructure. You only pay for the compute time your application consumes.

ANS.3

AWS Lambda natively supports several runtimes, including:

- Node.js (JavaScript)
- Python
- Java
- C# (.NET Core)
- Go
- Ruby

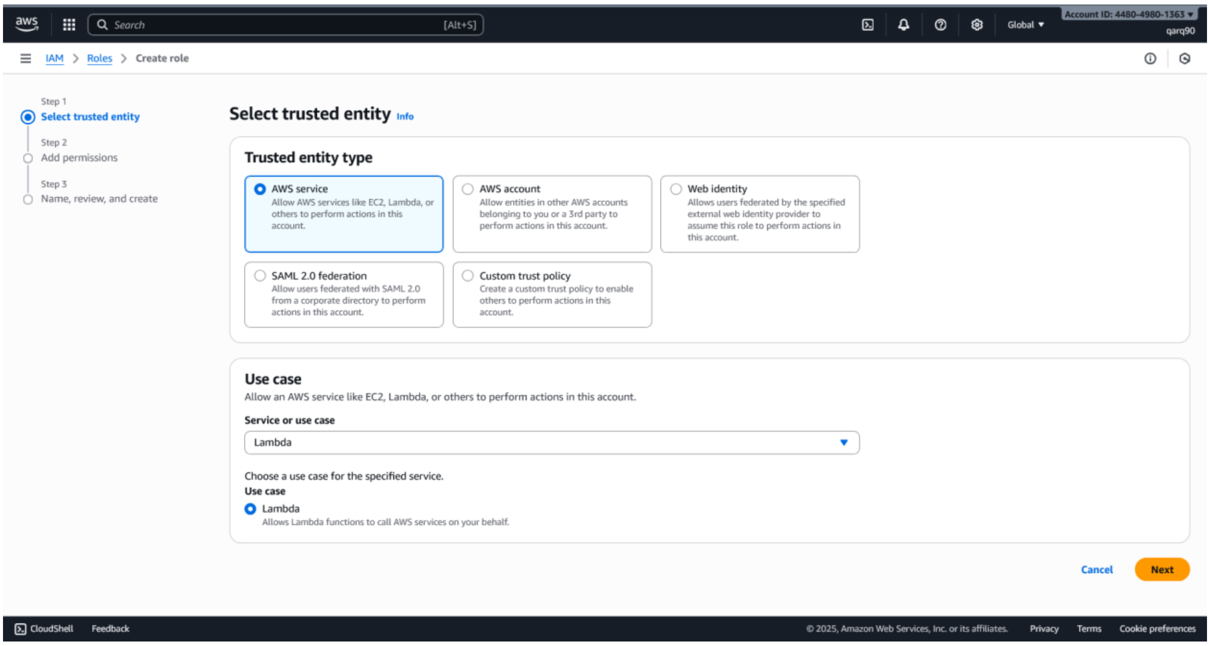
ANS.4

An AWS DynamoDB table is a fully managed, serverless NoSQL database provided by AWS that delivers reliable performance at any scale with seamless scalability and built-in security. Data is stored in items (rows) with attributes (columns), and each item is uniquely identified by a primary key.

ANS.5

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources by enabling you to manage users, groups, roles, and their corresponding permissions. Its core function is to ensure the principle of least privilege, meaning users and services are granted only the permissions they need to perform their specific tasks.

ANS.6



aws
Search
[Alt+S]
Global ▾
Account ID: 4480-4980-1363 ▾
qarq90
IAM > Roles > Create role
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step 1
● Select trusted entity

Step 2
○ Add permissions

Step 3
○ Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

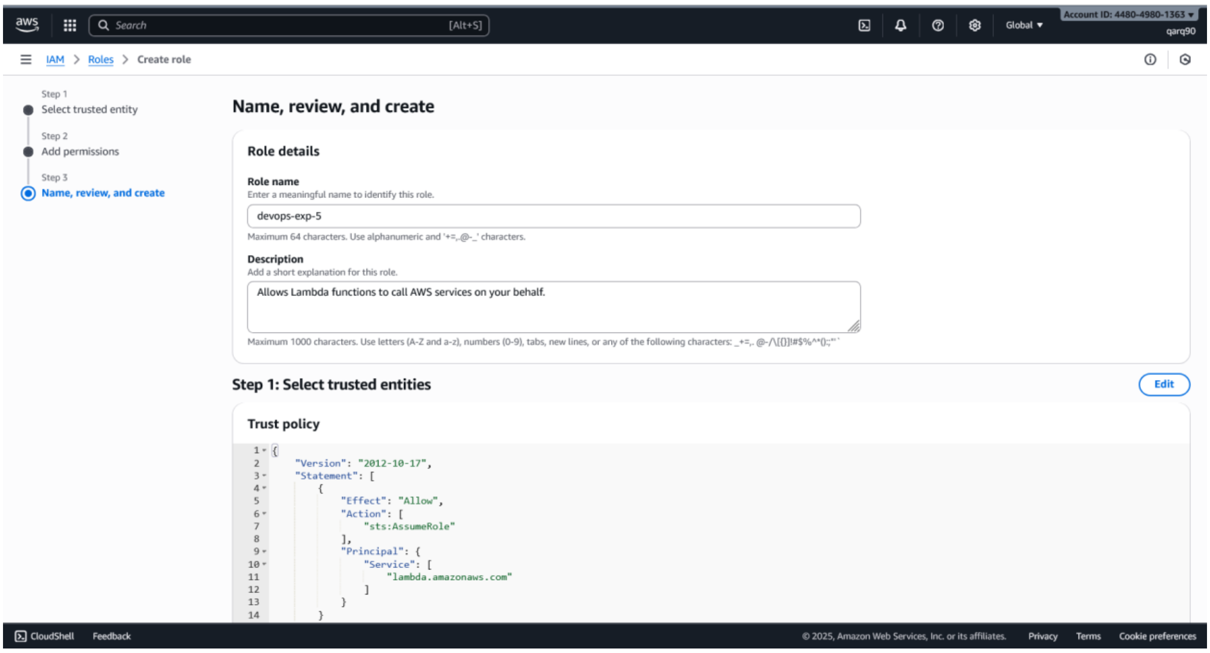
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

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.

Cancel   Next

Creating an IAM role



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

devops-exp-5

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.

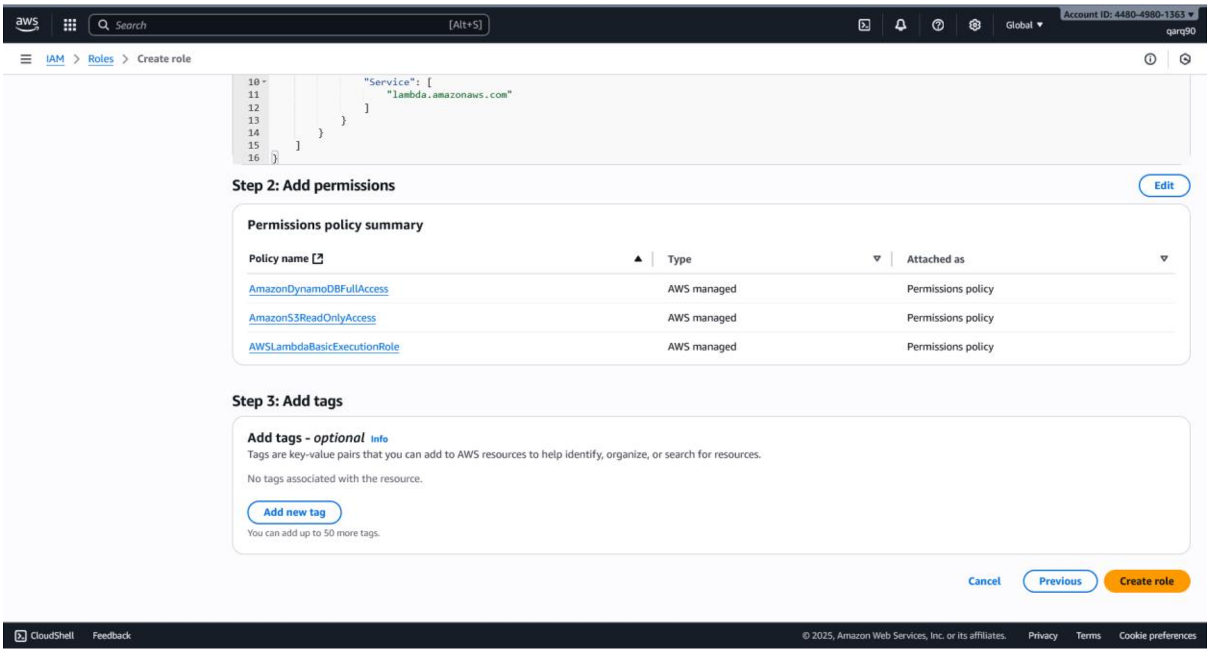
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();"'`

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "lambda.amazonaws.com"
                ]
            }
        }
```

Configuring the role



```
                "Service": [
                    "lambda.amazonaws.com"
                ]
            }
        }
    ]
}
```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonS3ReadOnlyAccess	AWS managed	Permissions policy
AWSLambdaBasicExecutionRole	AWS managed	Permissions policy

Step 3: Add tags

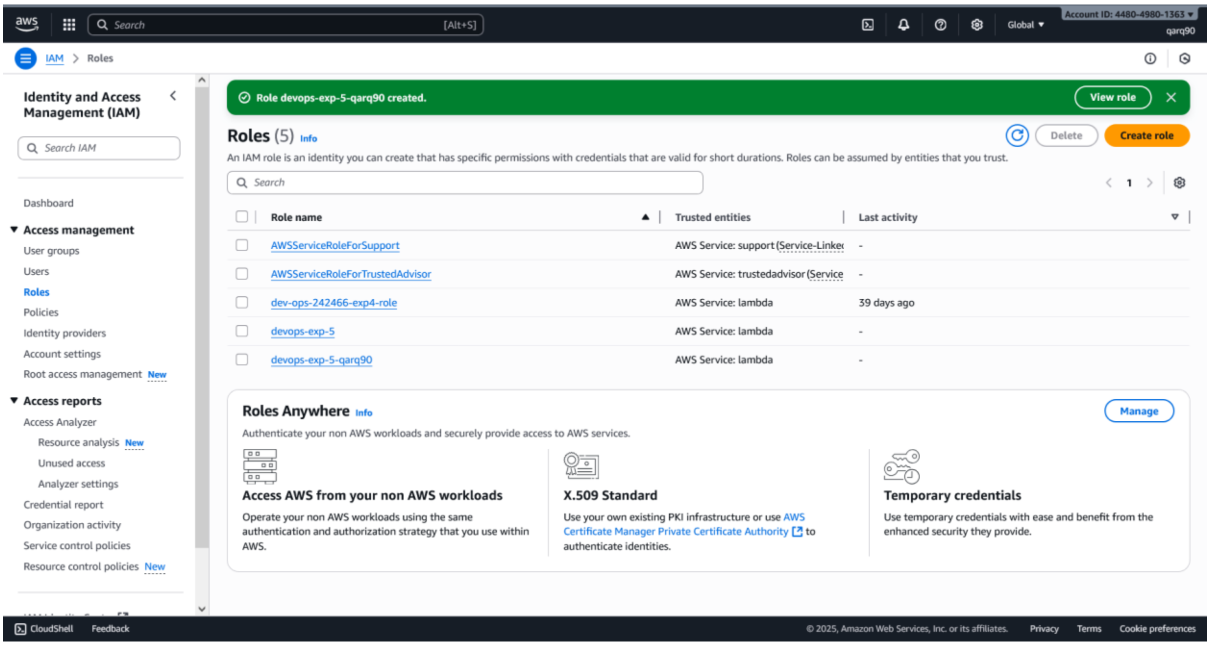
Add tags - *optional*

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

You can add up to 50 more tags.

Adding Permissions



Role devops-exp-5-qarq90 created.

Roles (5) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	dev-ops-242466-exp4-role	AWS Service: lambda	39 days ago
□	devops-exp-5	AWS Service: lambda	-
□	devops-exp-5-qarq90	AWS Service: lambda	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

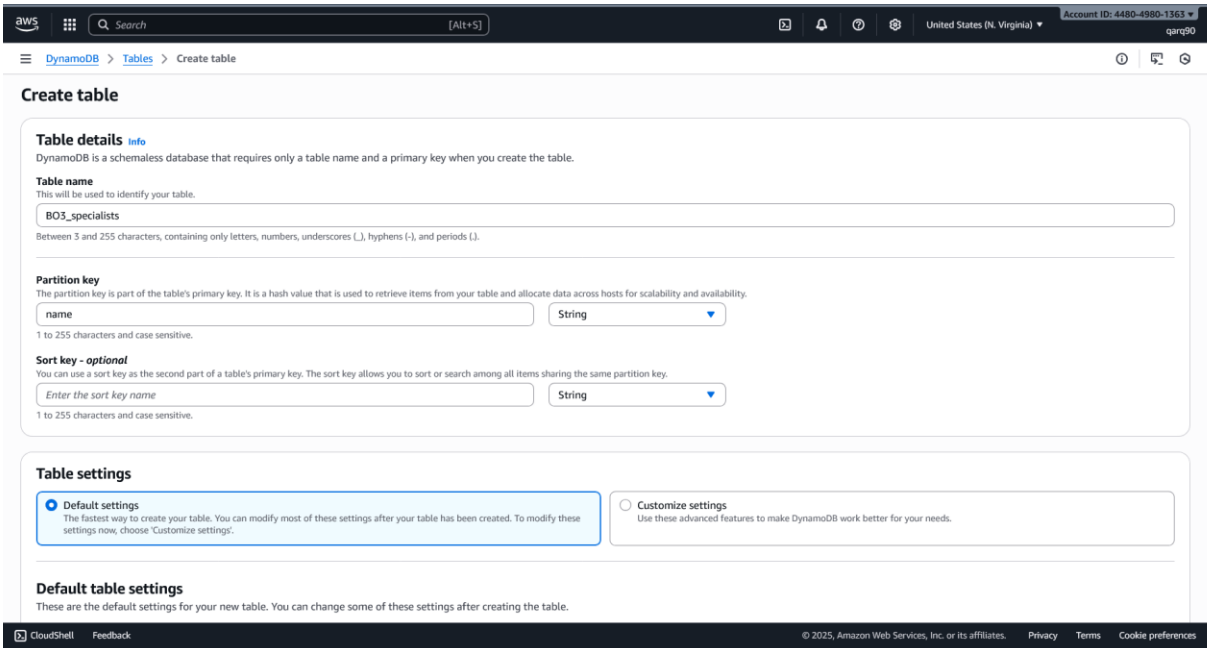
X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

Role Created Successfully


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 4480-4980-1363
qarq90
DynamoDB > Tables > Create table
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

BO3_specialists

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

name | String

1 to 255 characters and case sensitive.

Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name | String

1 to 255 characters and case sensitive.

Table settings

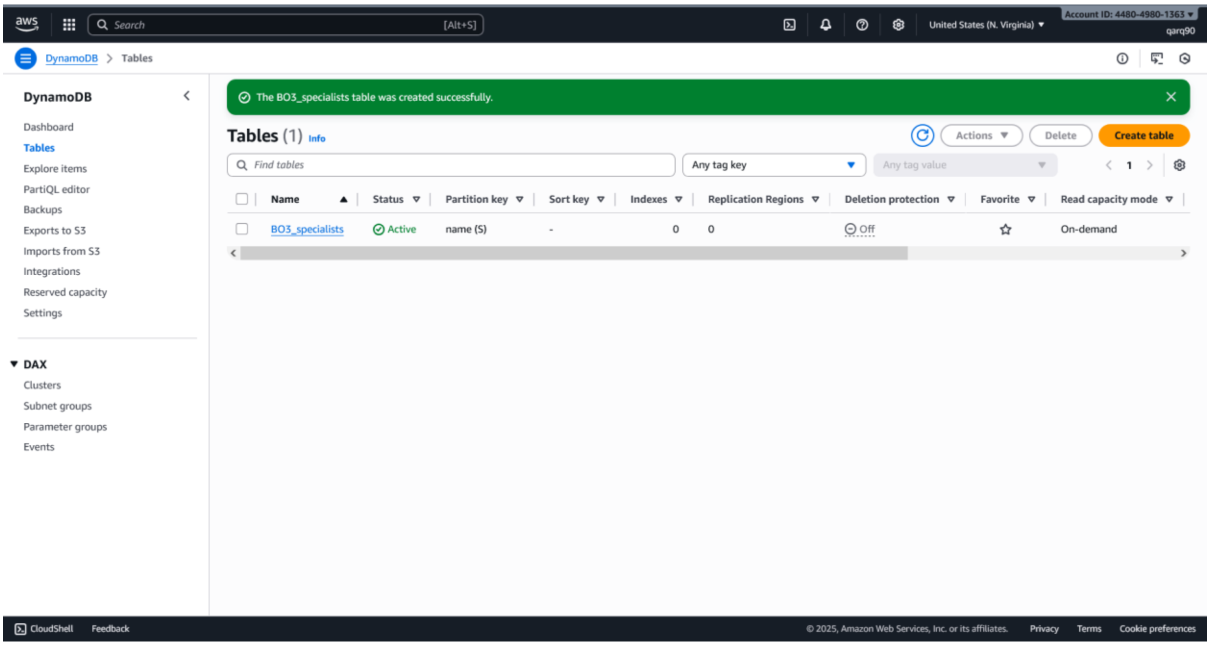
● Default settings
The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.

○ Customize settings
Use these advanced features to make DynamoDB work better for your needs.

Default table settings

These are the default settings for your new table. You can change some of these settings after creating the table.

Creating the DynamoDB table

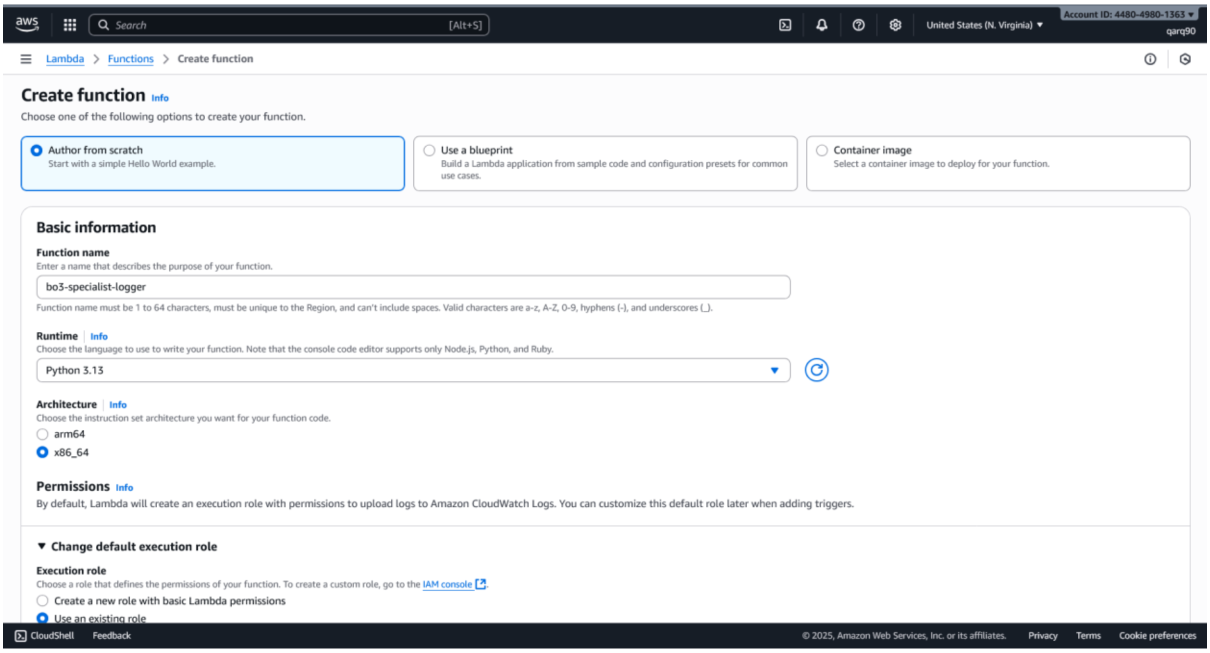


The BO3_specialists table was created successfully.

Tables (1) Info

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode
BO3_specialists	✓ Active	name (S)	-	0	0	Off	☆	On-demand

Table created Successfully



Create function Info

Choose one of the following options to create your function.

● 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.
bo3-specialist-logger
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

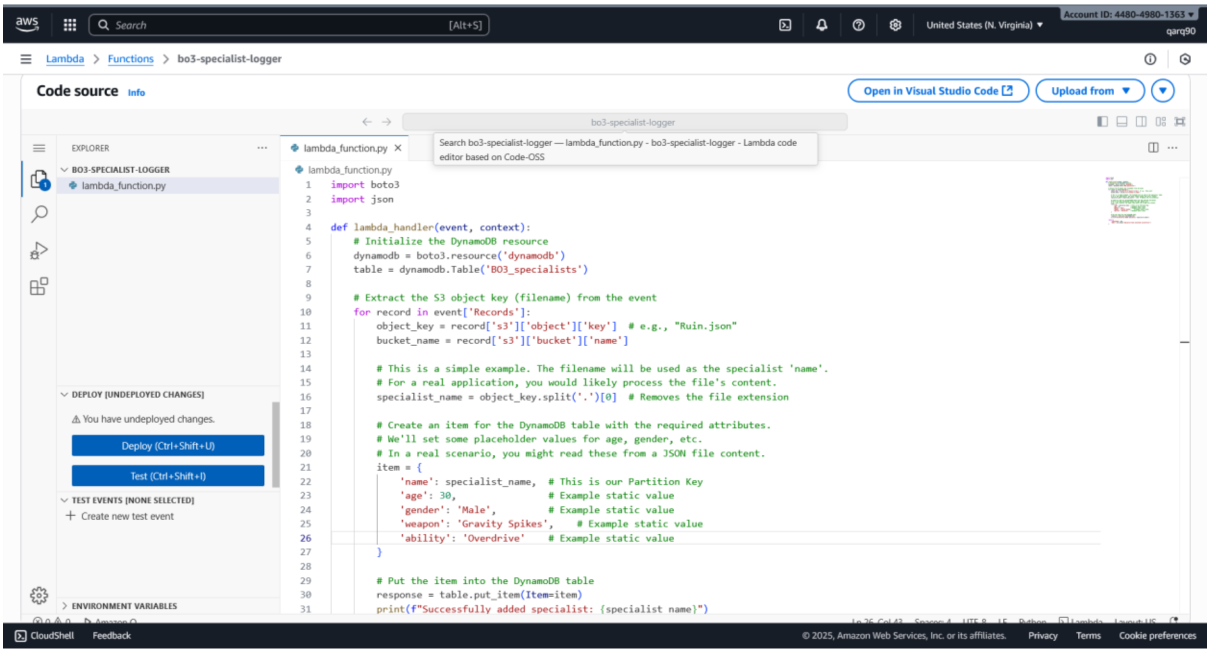
Architecture Info
Choose the instruction set architecture you want for your function code.
○ arm64
● x86_64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

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

Creating the lambda function



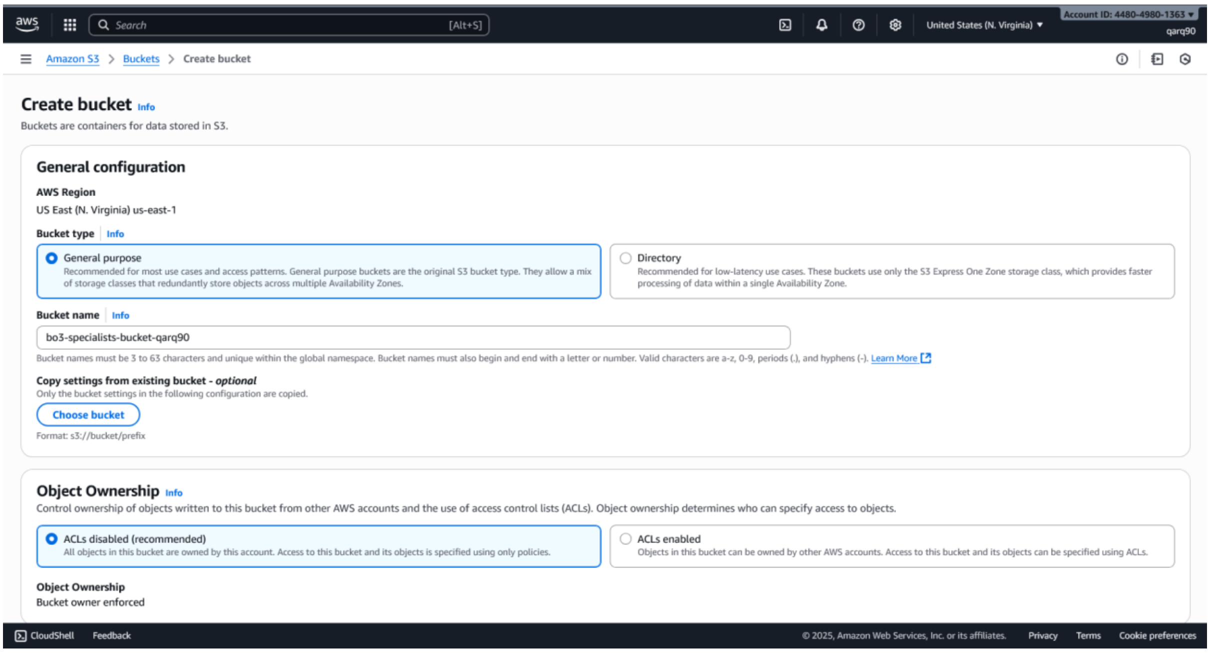
Search bo3-specialist-logger — lambda_function.py - bo3-specialist-logger - Lambda code editor based on Code-OSS

⚠ You have undeployed changes.

lambda_function.py

```python
import boto3
import json

def lambda_handler(event, context):
    # Initialize the DynamoDB resource
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('BO3_specialists')

    # Extract the S3 object key (filename) from the event
    for record in event['Records']:
        object_key = record['s3']['object']['key']  # e.g., "Ruin.json"
        bucket_name = record['s3']['bucket']['name']

        # This is a simple example. The filename will be used as the specialist 'name'.
        # For a real application, you would likely process the file's content.
        specialist_name = object_key.split('.')[0]  # Removes the file extension

        # Create an item for the DynamoDB table with the required attributes.
        # We'll set some placeholder values for age, gender, etc.
        # In a real scenario, you might read these from a JSON file content.
        item = {
            'name': specialist_name,  # This is our Partition Key
            'age': 30,                # Example static value
            'gender': 'Male',         # Example static value
            'weapon': 'Gravity Spikes',    # Example static value
            'ability': 'Overdrive'    # Example static value
        }

        # Put the item into the DynamoDB table
        response = table.put_item(Item=item)
        print(f"Successfully added specialist: {specialist name}")
```

Setting up the lambda function logic


Search [Alt+S]
United States (N. Virginia)
Account ID: 4480-4980-1363
qarq90
Amazon S3 > Buckets > Create bucket
Info
Choose bucket
Learn More
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create bucket

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type
● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
○ Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name
bo3-specialists-bucket-qarq90
Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-).

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.
Format: s3://bucket/prefix

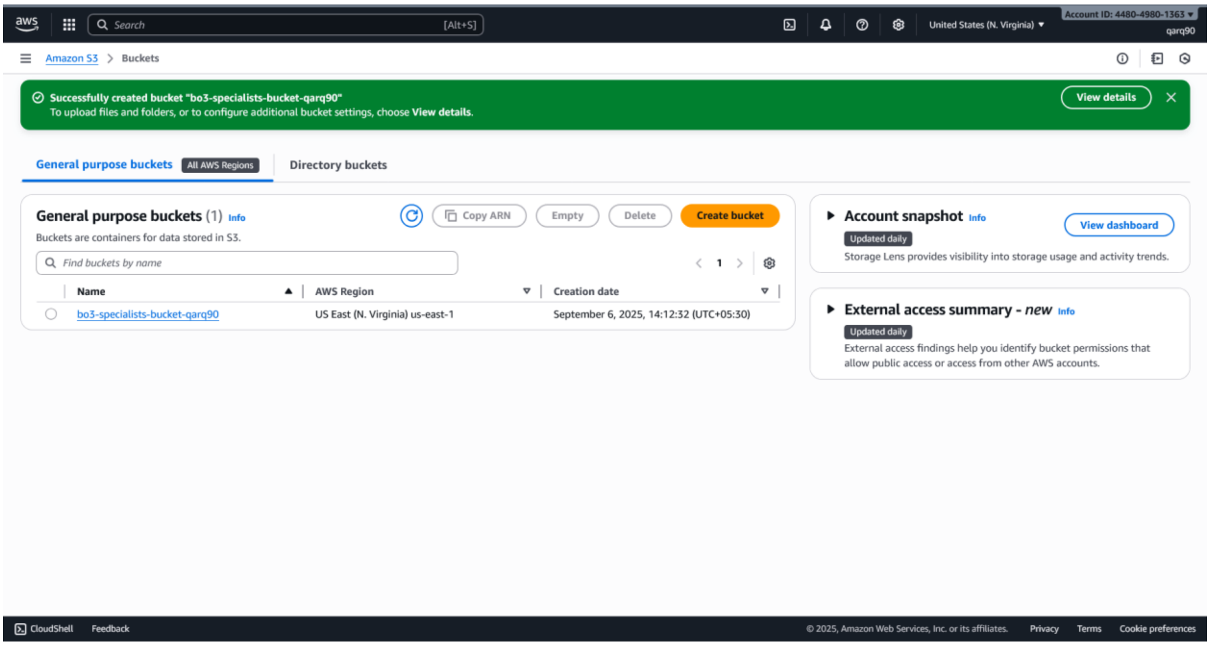
Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Creating the S3 bucket



Successfully created bucket "bo3-specialists-bucket-qarq90"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

General purpose buckets (1) Info

Buckets are containers for data stored in S3.

	Name	AWS Region	Creation date
○	bo3-specialists-bucket-qarq90	US East (N. Virginia) us-east-1	September 6, 2025, 14:12:32 (UTC+05:30)

Account snapshot Info

Updated daily

Storage Lens provides visibility into storage usage and activity trends.

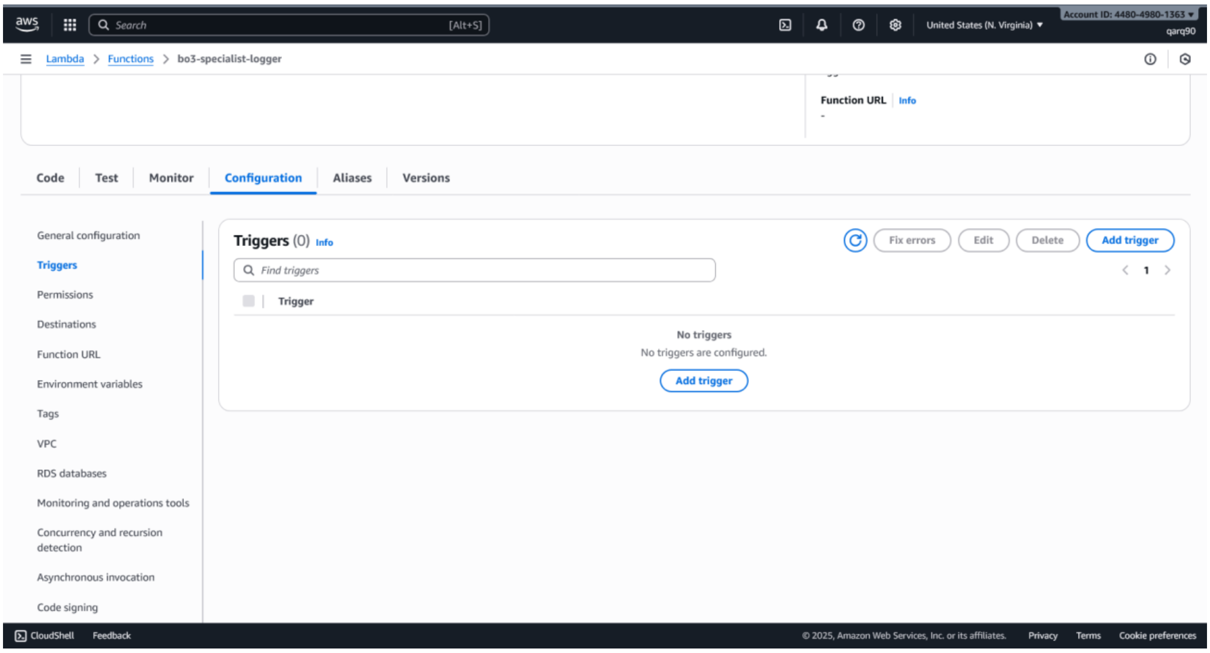
External access summary - *new* Info

Updated daily

External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

© 2025, Amazon Web Services, Inc. or its affiliates.

Bucket Created Successfully



Triggers (0) Info

□	Trigger

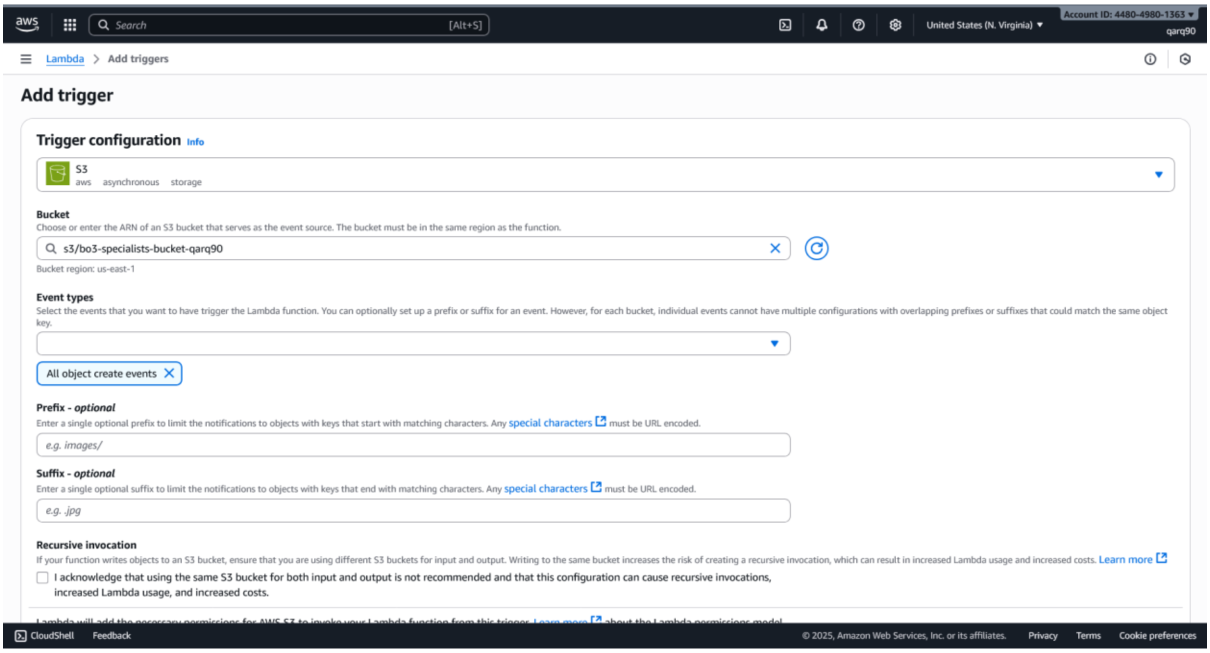
No triggers

No triggers are configured.

Add trigger

© 2025, Amazon Web Services, Inc. or its affiliates.

Creating the trigger



Add trigger

Trigger configuration Info

S3
aws asynchronous storage ▾

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/bo3-specialists-bucket-qarq90

Bucket region: us-east-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events ✕

Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.

e.g. images/

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.

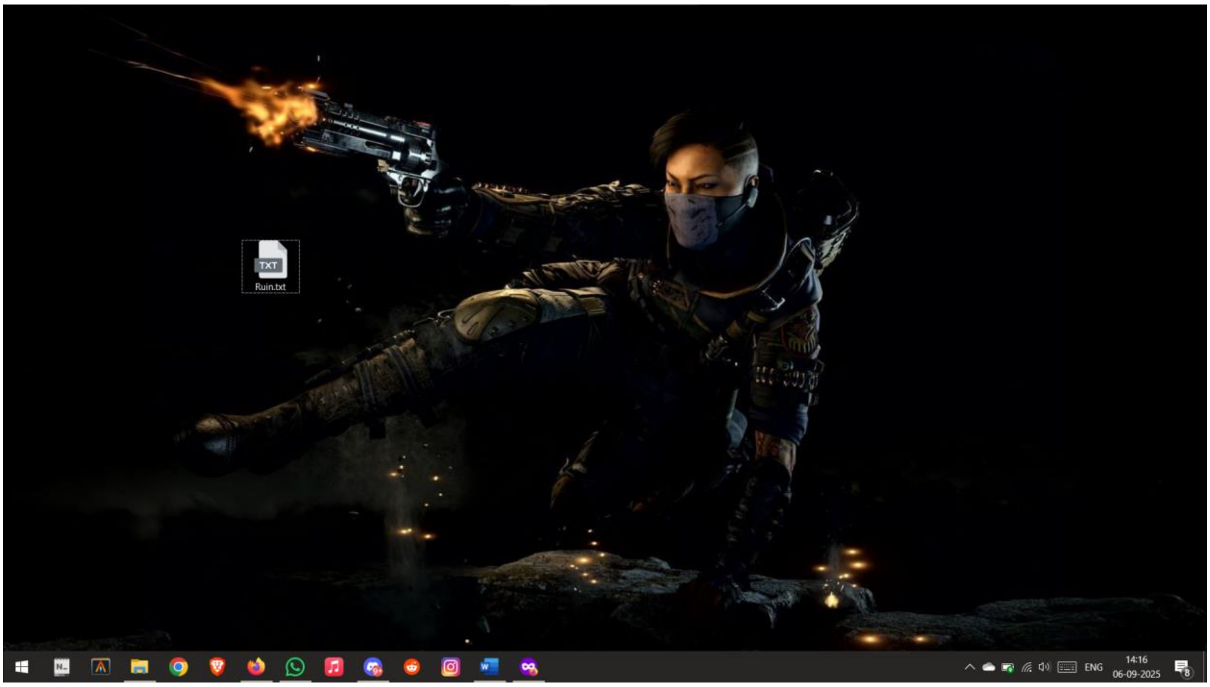
e.g. .jpg

Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more

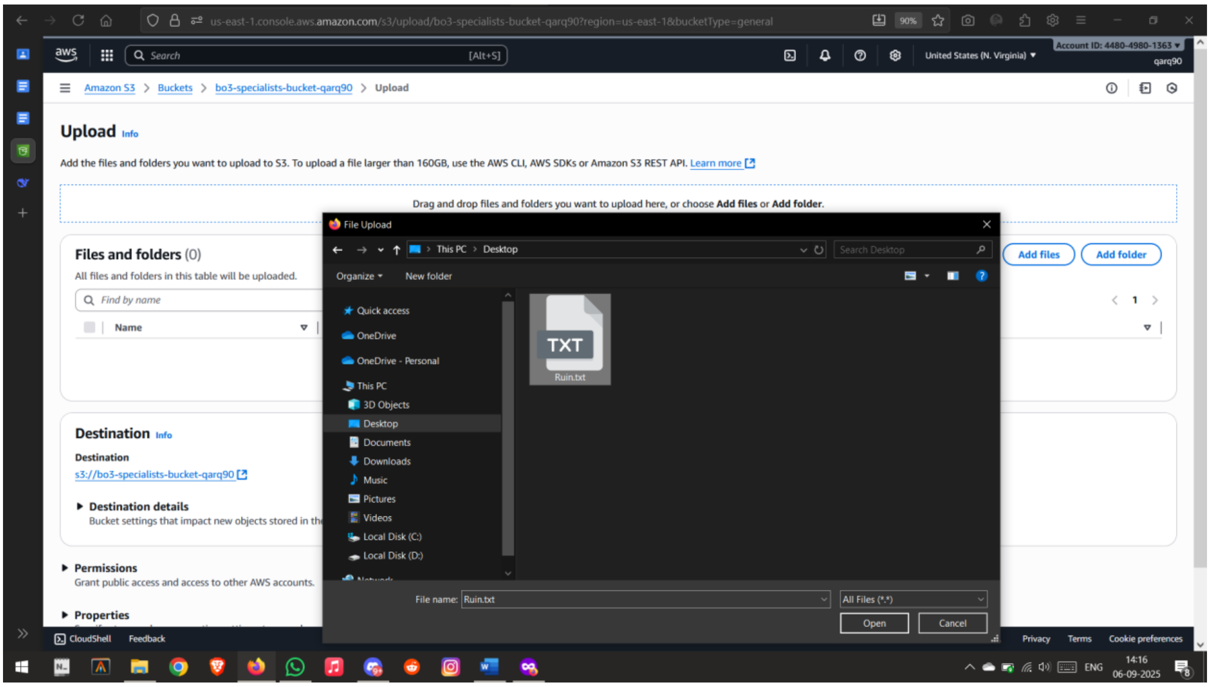
□ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

Adding the trigger to the bucket



Created a file to trigger the trigger on the bucket



Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (0)

All files and folders in this table will be uploaded.

Destination Info

Destination

s3://bo3-specialists-bucket-qarq90

▶ **Destination details**

Bucket settings that impact new objects stored in th

▶ **Permissions**

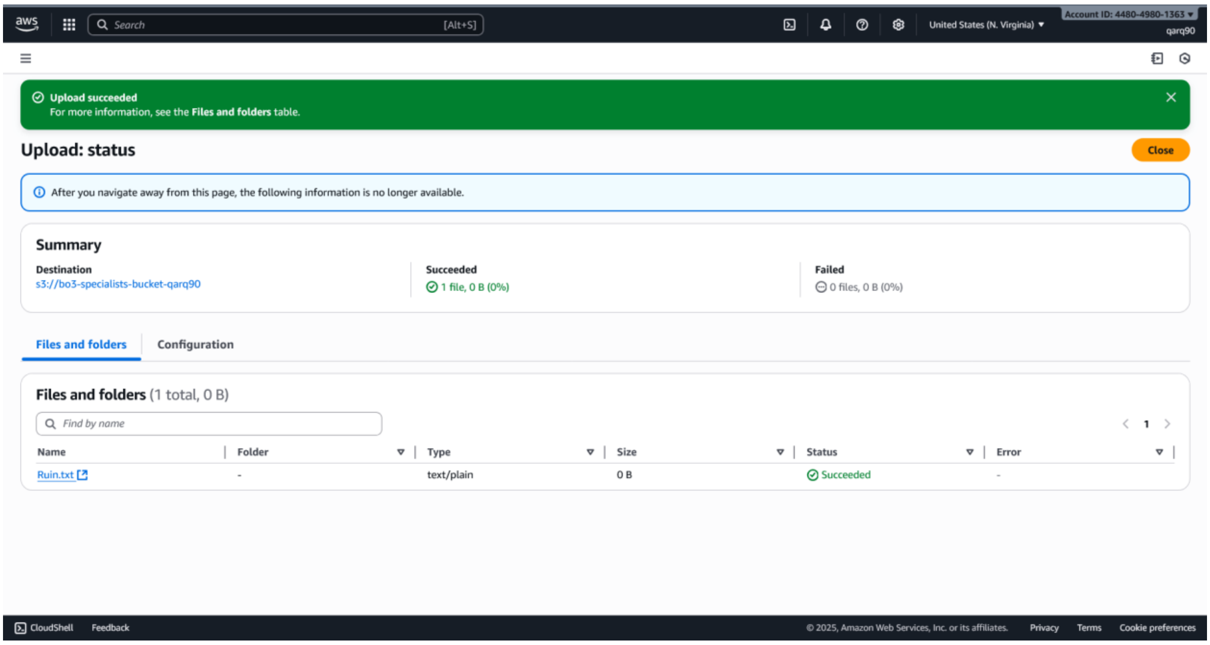
Grant public access and access to other AWS accounts.

▶ **Properties**

File name: Ruin.txt

All Files (*.*)

Uploading the file to the bucket



Upload succeeded

For more information, see the **Files and folders** table.

Upload: status

After you navigate away from this page, the following information is no longer available.

Summary

Destination	Succeeded	Failed
s3://bo3-specialists-bucket-qarq90	1 file, 0 B (0%)	0 files, 0 B (0%)

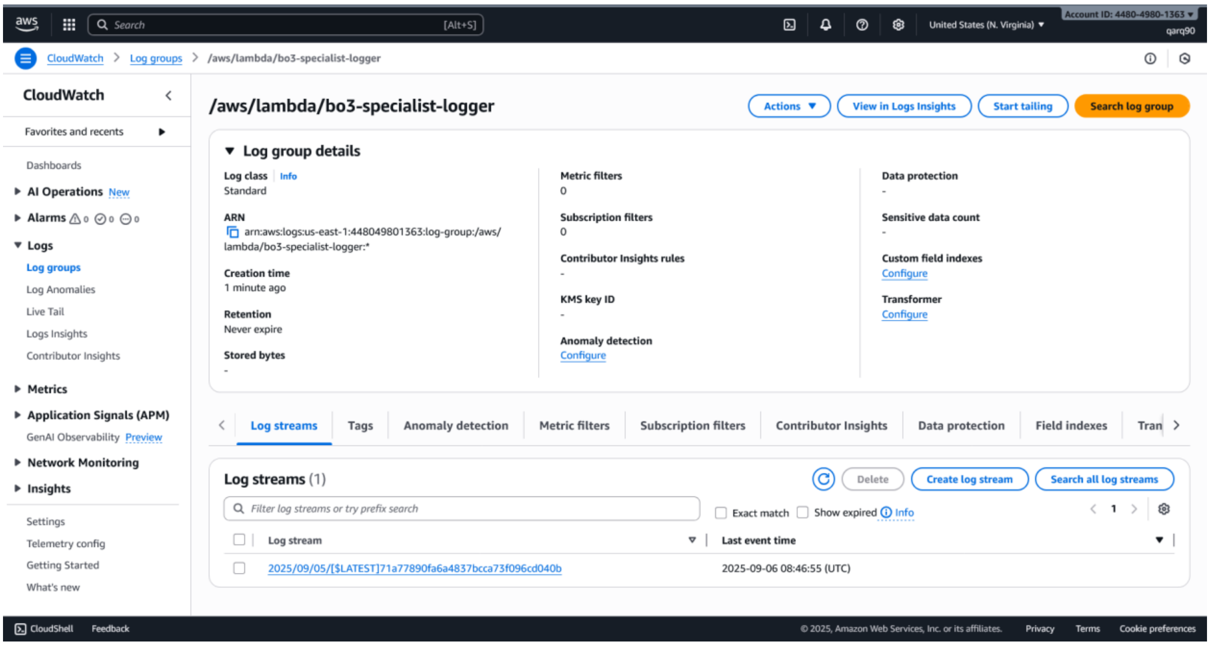
Files and folders | Configuration

Files and folders (1 total, 0 B)

Name	Folder	Type	Size	Status	Error
Ruin.txt	-	text/plain	0 B	Succeeded	-

© 2025, Amazon Web Services, Inc. or its affiliates.

File uploaded successfully



/aws/lambda/bo3-specialist-logger

Log group details

Log class Info
Standard

ARN
arn:aws:logs:us-east-1:448049801363:log-group:/aws/lambda/bo3-specialist-logger:*

Creation time
1 minute ago

Retention
Never expire

Stored bytes
-

Metric filters
0

Subscription filters
0

Contributor Insights rules
-

KMS key ID
-

Anomaly detection
Configure

Data protection
-

Sensitive data count
-

Custom field indexes
Configure

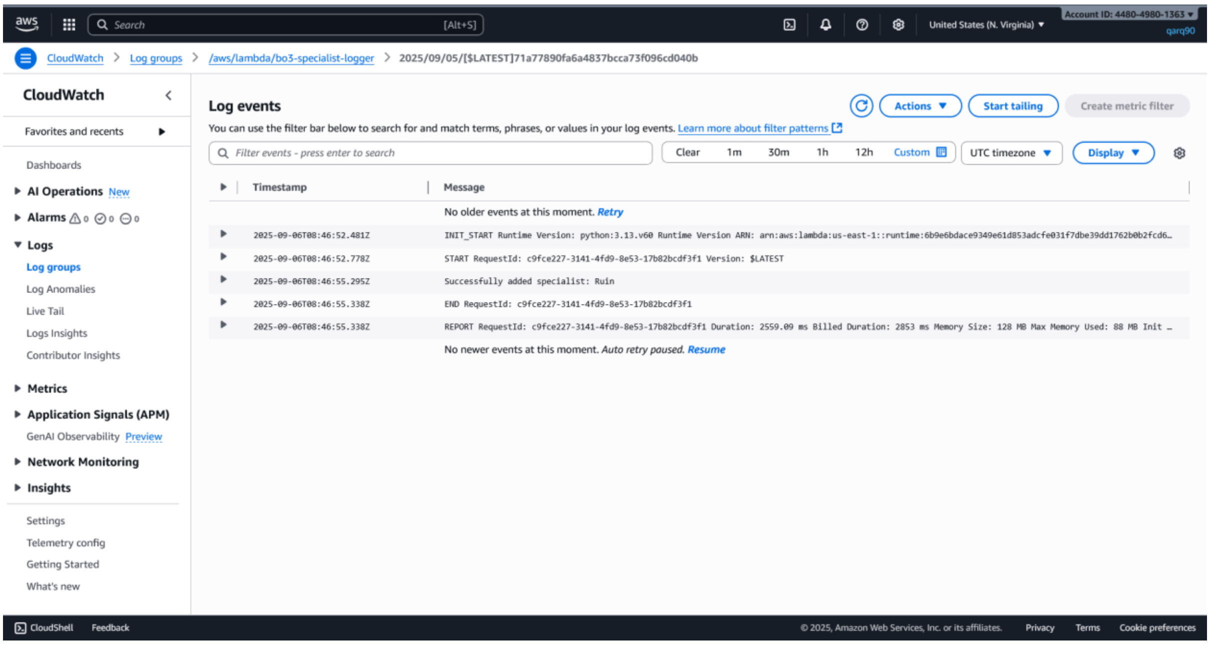
Transformer
Configure

Log streams (1)

□ Exact match □ Show expired Info

□	Log stream	Last event time
□	2025/09/05/[$LATEST]71a77890fa6a4837bcca73f096cd040b	2025-09-06 08:46:55 (UTC)

Checking the logs

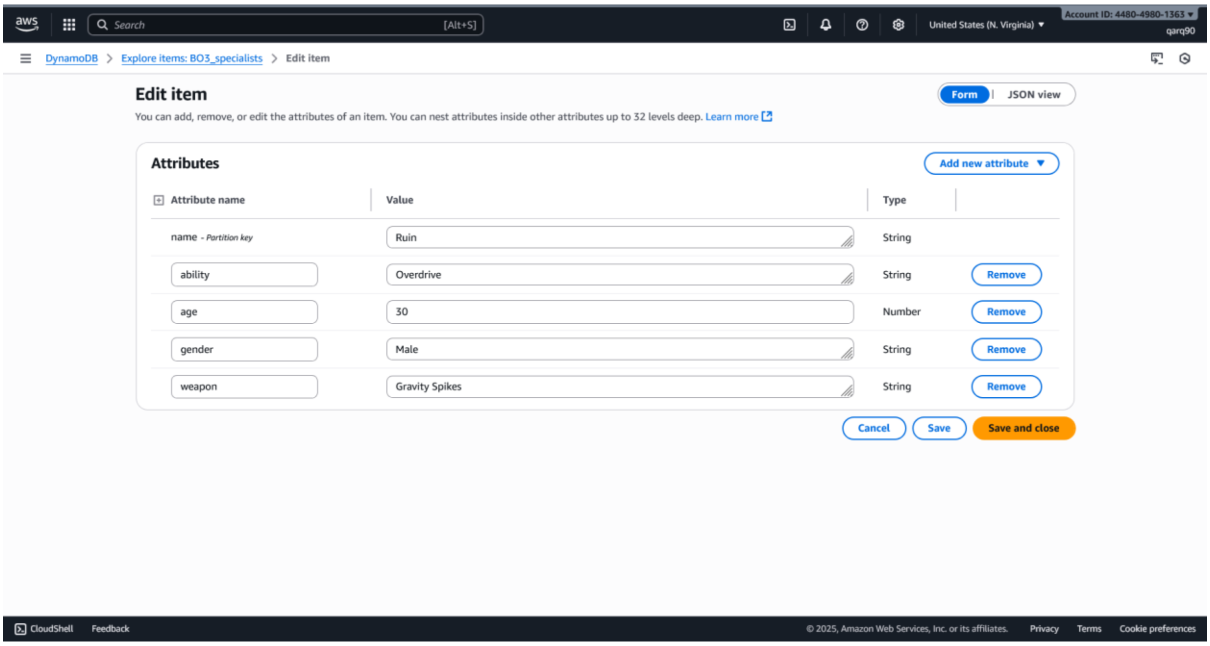


Log events

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
2025-09-06T08:46:52.481Z   INIT_START Runtime Version: python:3.13.v60 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:6b9e6bdace9349e61d853adcfe031f7dbe39dd1762b0b2fcd6…
2025-09-06T08:46:52.778Z   START RequestId: c9fce227-3141-4fd9-8e53-17b82bcdf3f1 Version: $LATEST
2025-09-06T08:46:55.295Z   Successfully added specialist: Ruin
2025-09-06T08:46:55.338Z   END RequestId: c9fce227-3141-4fd9-8e53-17b82bcdf3f1
2025-09-06T08:46:55.338Z   REPORT RequestId: c9fce227-3141-4fd9-8e53-17b82bcdf3f1 Duration: 2559.09 ms Billed Duration: 2853 ms Memory Size: 128 MB Max Memory Used: 88 MB Init …
                           No newer events at this moment. Auto retry paused. Resume
```

Log created successfully via the trigger


Search [Alt+S]
United States (N. Virginia) ▾
Account ID: 4480-4980-1363 ▾
qarq90
DynamoDB > Explore items: BO3_specialists > Edit item
Form | JSON view
Add new attribute ▼
Remove
Cancel
Save
Save and close
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

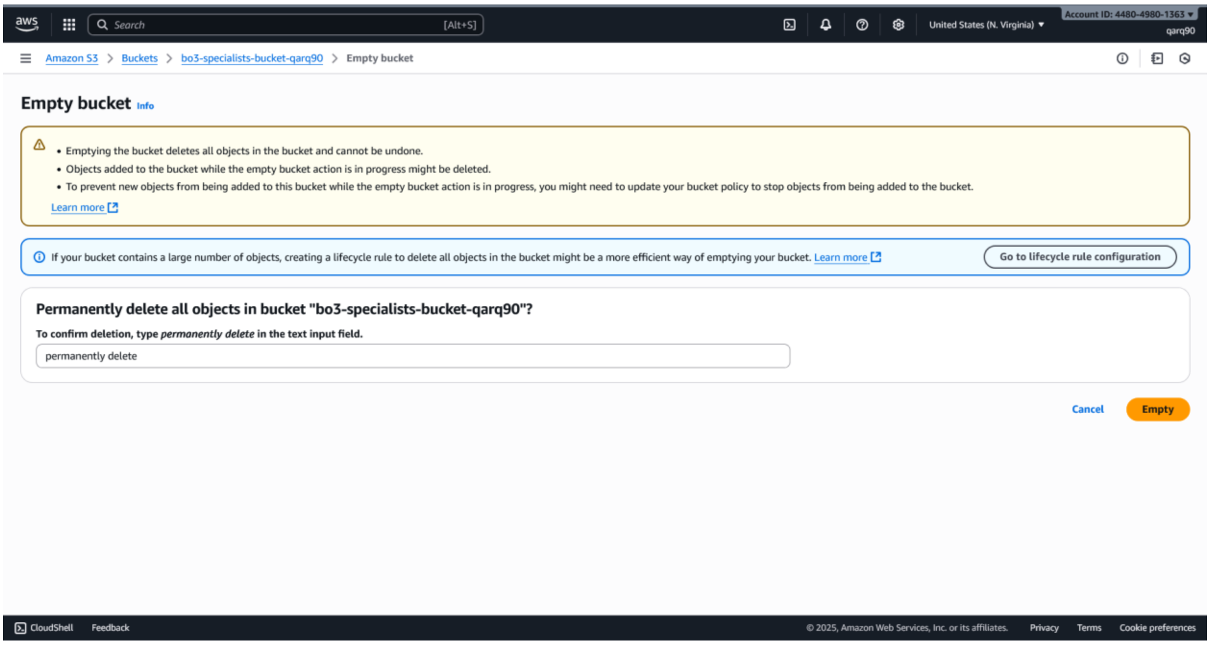
Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more

Attributes

Attribute name	Value	Type
name - *Partition key*	Ruin	String
ability	Overdrive	String
age	30	Number
gender	Male	String
weapon	Gravity Spikes	String

Verified the insertion in the Table


Search [Alt+S]
United States (N. Virginia) ▾
Account ID: 4480-4980-1363 ▾
qarq90
Amazon S3 > Buckets > bo3-specialists-bucket-qarq90 > Empty bucket
Info
Learn more
Go to lifecycle rule configuration
Cancel
Empty
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Empty bucket

- Emptying the bucket deletes all objects in the bucket and cannot be undone.
- Objects added to the bucket while the empty bucket action is in progress might be deleted.
- To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

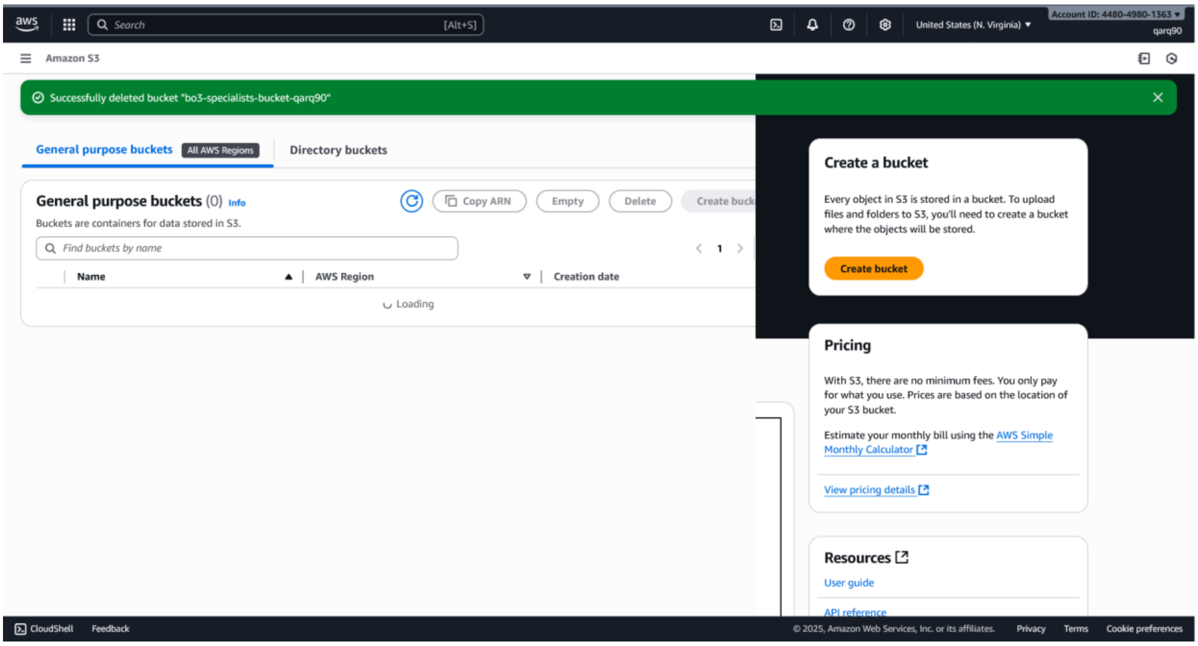
If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket. Learn more

Permanently delete all objects in bucket "bo3-specialists-bucket-qarq90"?

To confirm deletion, type *permanently delete* in the text input field.

permanently delete

Emptying the bucket



Successfully deleted bucket "bo3-specialists-bucket-qarq90"

General purpose buckets (0) Info

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date

Loading

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the AWS Simple Monthly Calculator

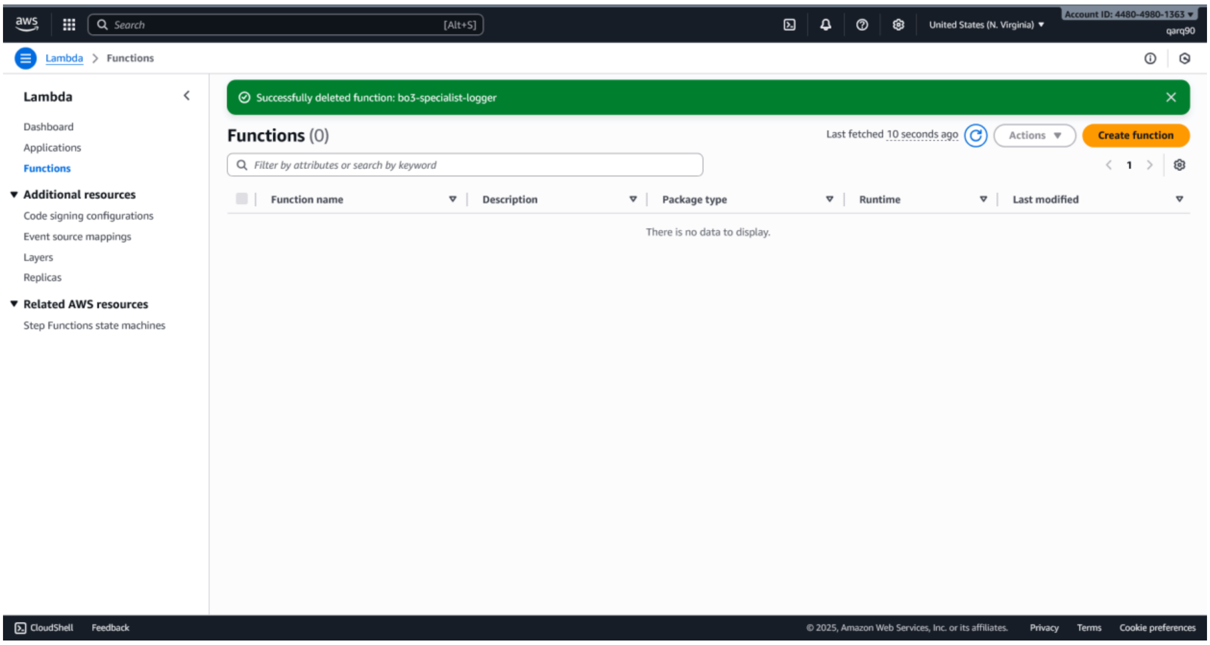
View pricing details

Resources

User guide

API reference

Deleting the bucket


aws
Search [Alt+S]
United States (N. Virginia) ▾
Account ID: 4480-4980-1363 ▾
qarq90
Lambda > Functions
Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Event source mappings
Layers
Replicas
Related AWS resources
Step Functions state machines

Successfully deleted function: bo3-specialist-logger

Functions (0)

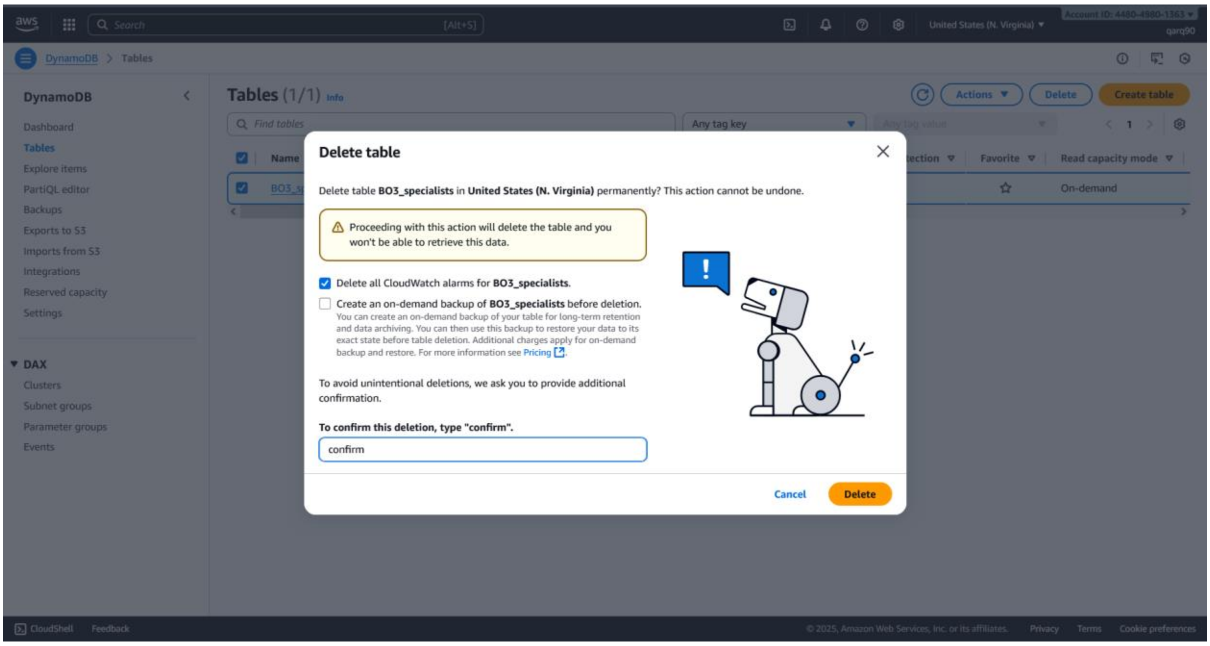
Last fetched 10 seconds ago
Actions ▾
Create function
Filter by attributes or search by keyword
1

Function name	Description	Package type	Runtime	Last modified

There is no data to display.

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

Deleting the lambda function



Delete table

Delete table **BO3_specialists** in **United States (N. Virginia)** permanently? This action cannot be undone.

⚠ Proceeding with this action will delete the table and you won't be able to retrieve this data.

☑ Delete all CloudWatch alarms for **BO3_specialists**.

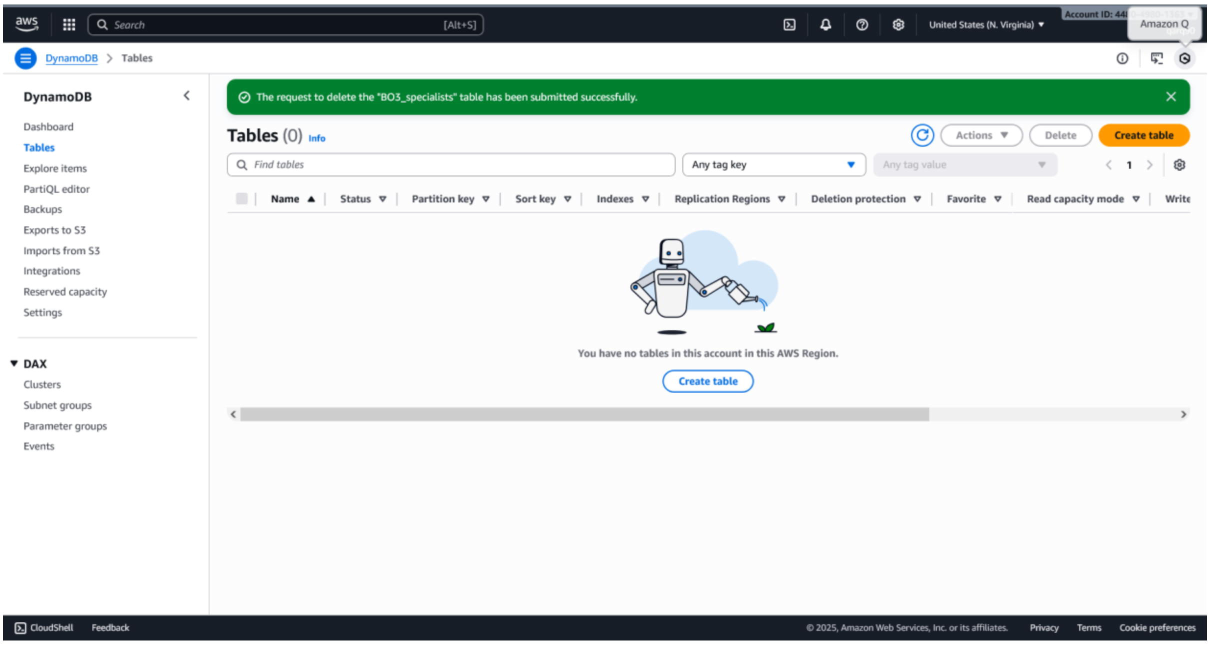
□ Create an on-demand backup of **BO3_specialists** before deletion.
You can create an on-demand backup of your table for long-term retention and data archiving. You can then use this backup to restore your data to its exact state before table deletion. Additional charges apply for on-demand backup and restore. For more information see Pricing.

To avoid unintentional deletions, we ask you to provide additional confirmation.

To confirm this deletion, type "confirm".

confirm

Emptying the table



The request to delete the "BO3_specialists" table has been submitted successfully.

Tables (0)

You have no tables in this account in this AWS Region.

Deleted the table