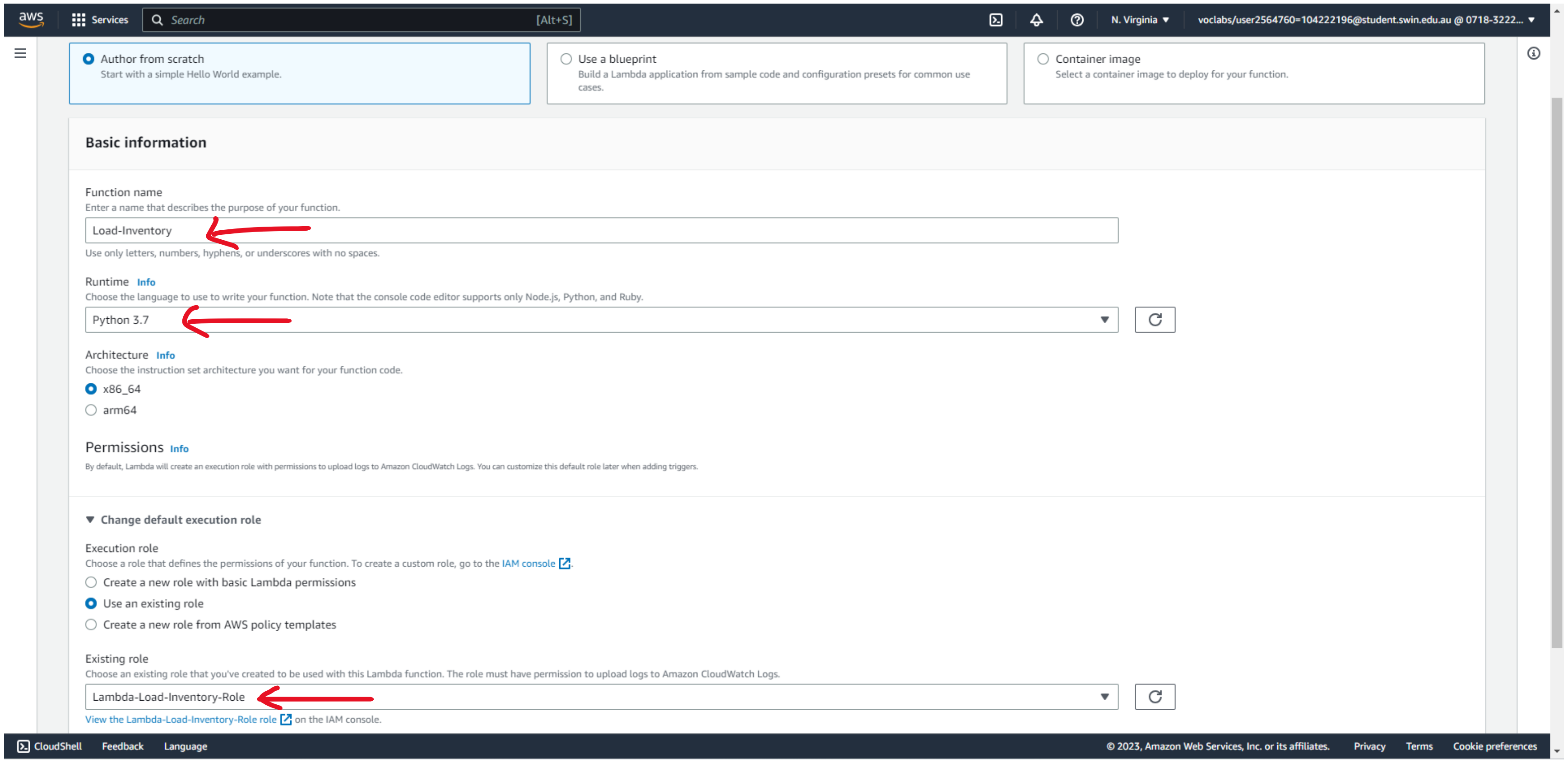
Ta Quang Tung – 104222196

COS20019 – Cloud Computing Architecture - Wk9: ACA Module 13 Guided Lab - Implementing a Serverless Architecture with AWS Lambda

Task 1 - Creating a Lambda function to load data


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...
CloudShell
Feedback
Language
© 2023, 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.
Load-Inventory
Use only letters, numbers, hyphens, or underscores with no spaces.

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 3.7

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64] arm64

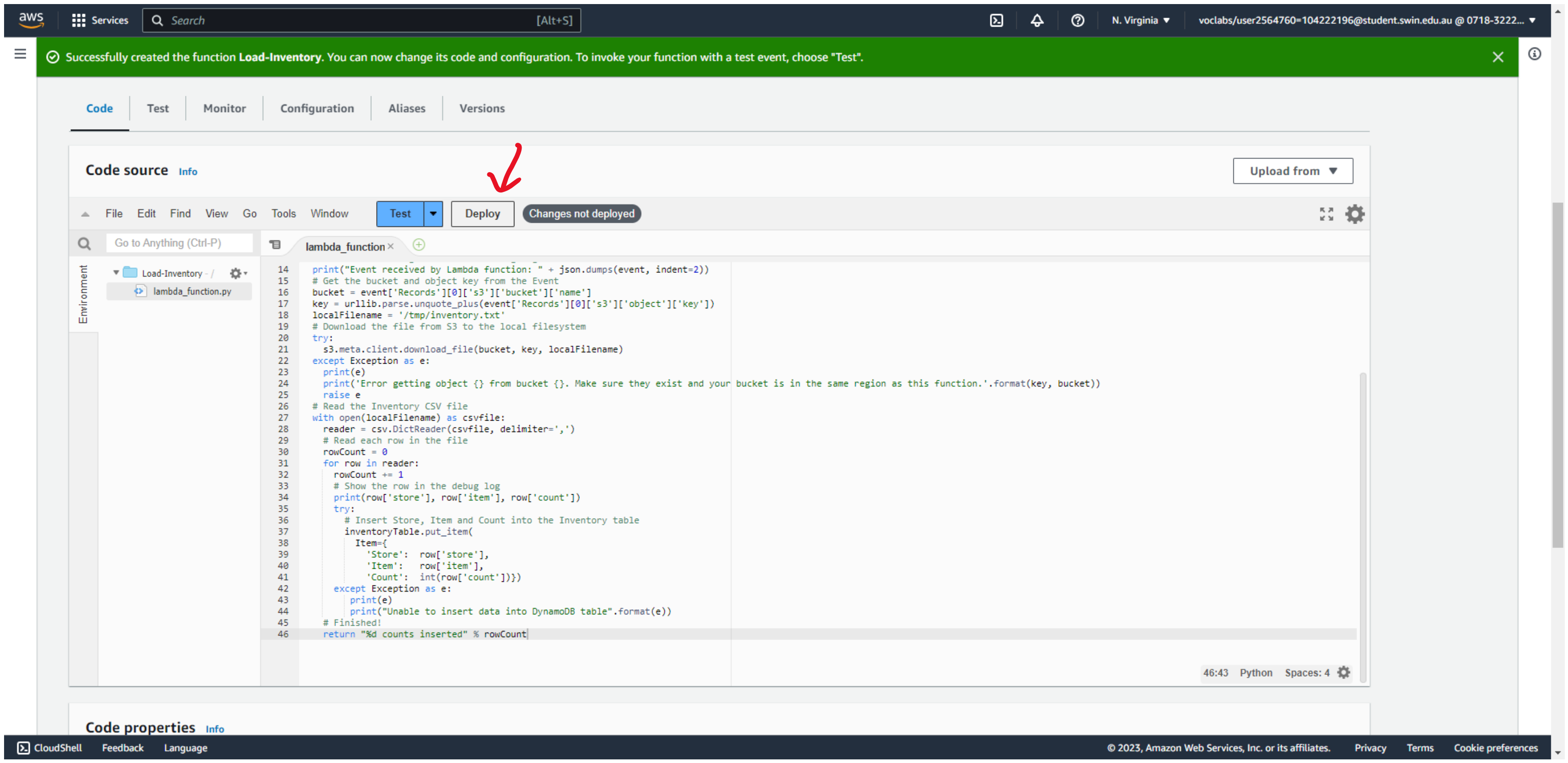
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ 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]
Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
Lambda-Load-Inventory-Role
View the Lambda-Load-Inventory-Role role on the IAM console.

Create a new Lambda function to load an inventory file into the database, specifying its name, runtime, and IAM role.



Successfully created the function **Load-Inventory**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

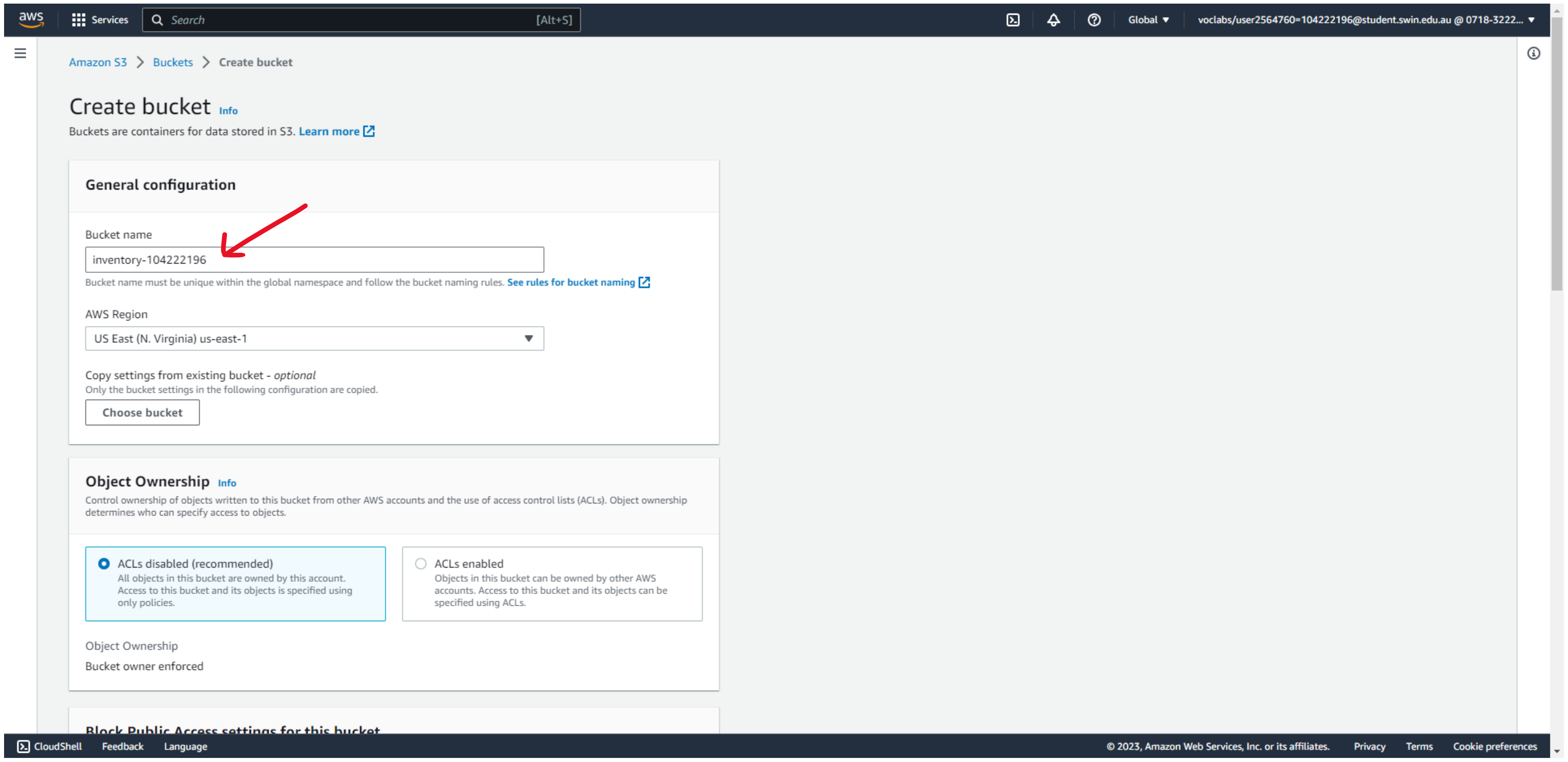
Code source Info

```
  print("Event received by Lambda function: " + json.dumps(event, indent=2))
  # Get the bucket and object key from the Event
  bucket = event['Records'][0]['s3']['bucket']['name']
  key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'])
  localFilename = '/tmp/inventory.txt'
  # Download the file from S3 to the local filesystem
  try:
    s3.meta.client.download_file(bucket, key, localFilename)
  except Exception as e:
    print(e)
    print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))
    raise e
  # Read the Inventory CSV file
  with open(localFilename) as csvfile:
    reader = csv.DictReader(csvfile, delimiter=',')
    # Read each row in the file
    rowCount = 0
    for row in reader:
      rowCount += 1
      # Show the row in the debug log
      print(row['store'], row['item'], row['count'])
      try:
        # Insert Store, Item and Count into the Inventory table
        inventoryTable.put_item(
          Item={
            'Store':  row['store'],
            'Item':   row['item'],
            'Count':  int(row['count'])})
      except Exception as e:
         print(e)
         print("Unable to insert data into DynamoDB table".format(e))
    # Finished!
    return "%d counts inserted" % rowCount
```

Code properties Info

After the function is created, copy and paste the code into the function, then deploy.

Task 2 - Configuring an Amazon S3 event


Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

inventory-104222196

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

AWS Region

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

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

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.

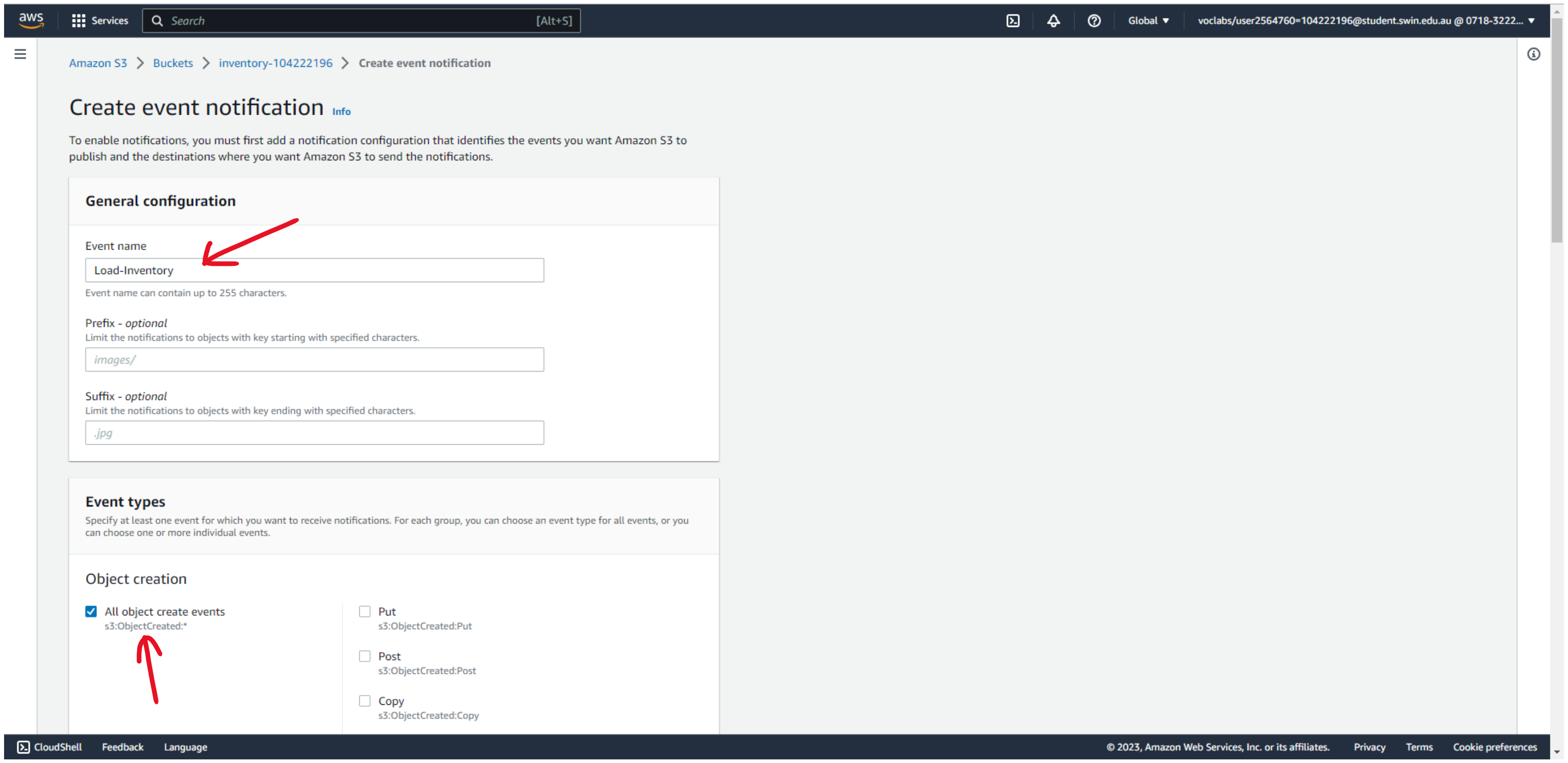
Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

Services
Search [Alt+S]
Global
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create an S3 bucket with a unique name.


aws
Services
Search
[Alt+S]
Global
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...
Amazon S3 > Buckets > inventory-104222196 > Create event notification

Create event notification Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

General configuration

Event name
Load-Inventory
Event name can contain up to 255 characters.

Prefix - *optional*
Limit the notifications to objects with key starting with specified characters.
images/

Suffix - *optional*
Limit the notifications to objects with key ending with specified characters.
.jpg

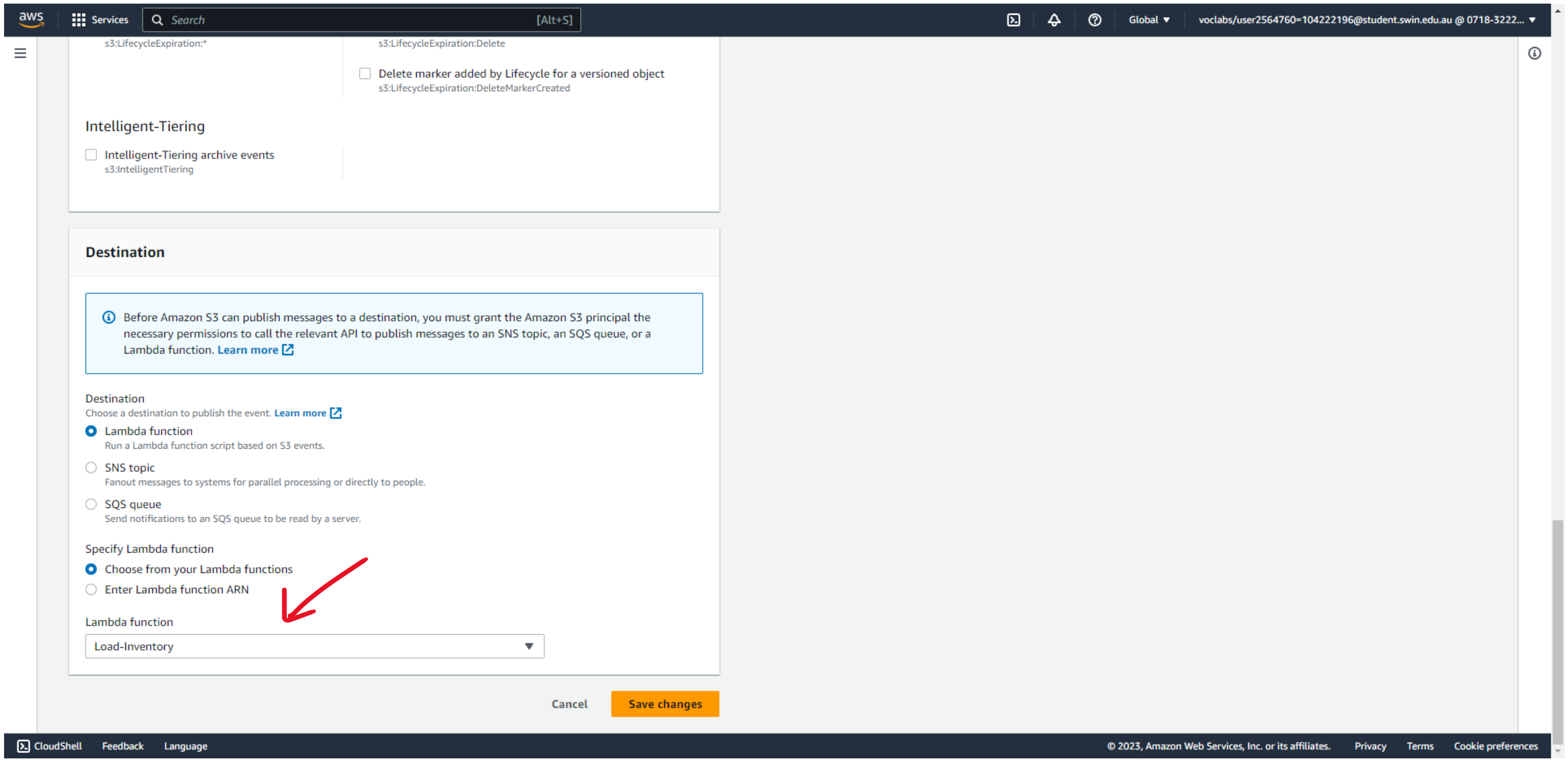
Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

- [✓] All object create events s3:ObjectCreated:*
- [ ] Put s3:ObjectCreated:Put
- [ ] Post s3:ObjectCreated:Post
- [ ] Copy s3:ObjectCreated:Copy

Create an event notification inside the S3 bucket (part 1/2), specifying its name and event type.


aws
Services
Search
[Alt+S]
Global
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...

s3:LifecycleExpiration:*

s3:LifecycleExpiration:Delete

☐ Delete marker added by Lifecycle for a versioned object
s3:LifecycleExpiration:DeleteMarkerCreated

Intelligent-Tiering

☐ Intelligent-Tiering archive events
s3:IntelligentTiering

Destination

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. Learn more

Destination
Choose a destination to publish the event. Learn more

● Lambda function
Run a Lambda function script based on S3 events.

○ SNS topic
Fanout messages to systems for parallel processing or directly to people.

○ SQS queue
Send notifications to an SQS queue to be read by a server.

Specify Lambda function

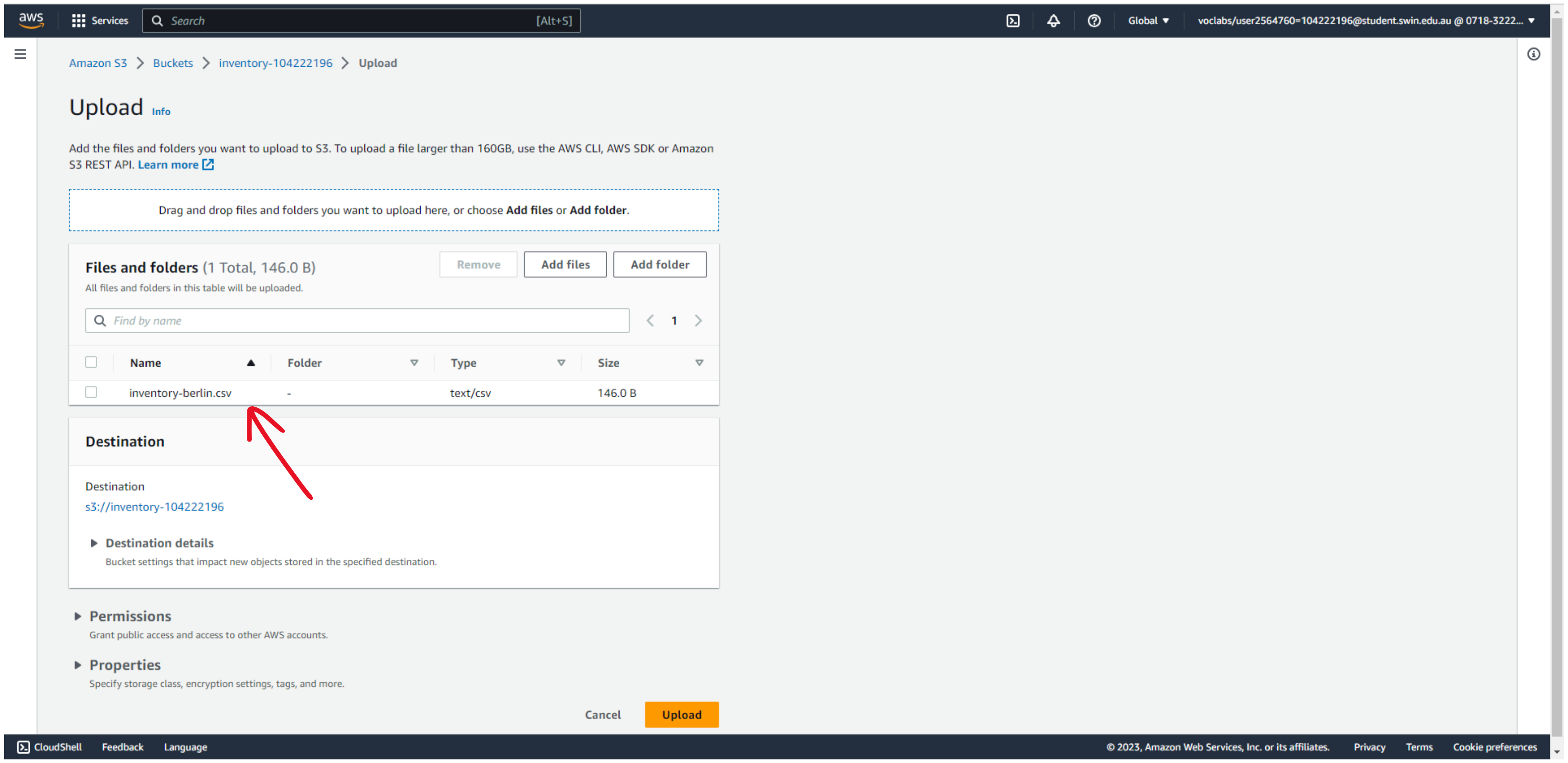
● Choose from your Lambda functions

○ Enter Lambda function ARN

Lambda function
Load-Inventory

Create an event notification inside the S3 bucket (part 2/2), specifying its destination Lambda function.

Task 3 - Testing the loading process



Amazon S3 > Buckets > inventory-104222196 > Upload

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 SDK 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 (1 Total, 146.0 B)

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
inventory-berlin.csv	-	text/csv	146.0 B

Destination

Destination

s3://inventory-104222196

Destination details

Bucket settings that impact new objects stored in the specified destination.

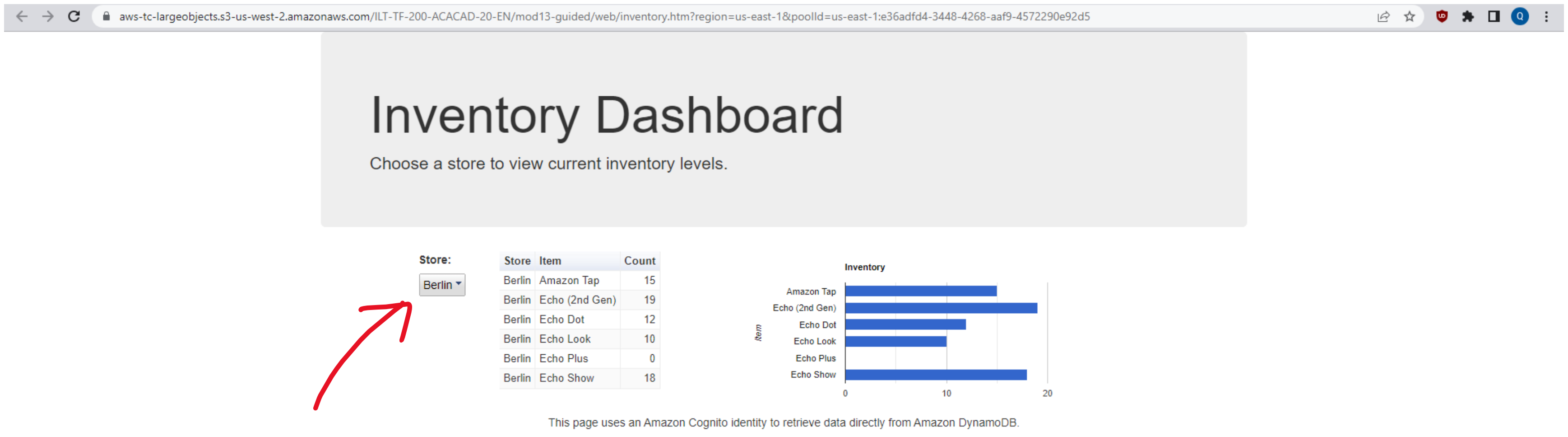
Permissions

Grant public access and access to other AWS accounts.

Properties

Specify storage class, encryption settings, tags, and more.

Upload an inventory csv file to S3.



aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod13-guided/web/inventory.htm?region=us-east-1&poolId=us-east-1:e36adfd4-3448-4268-aaf9-4572290e92d5

Inventory Dashboard

Choose a store to view current inventory levels.

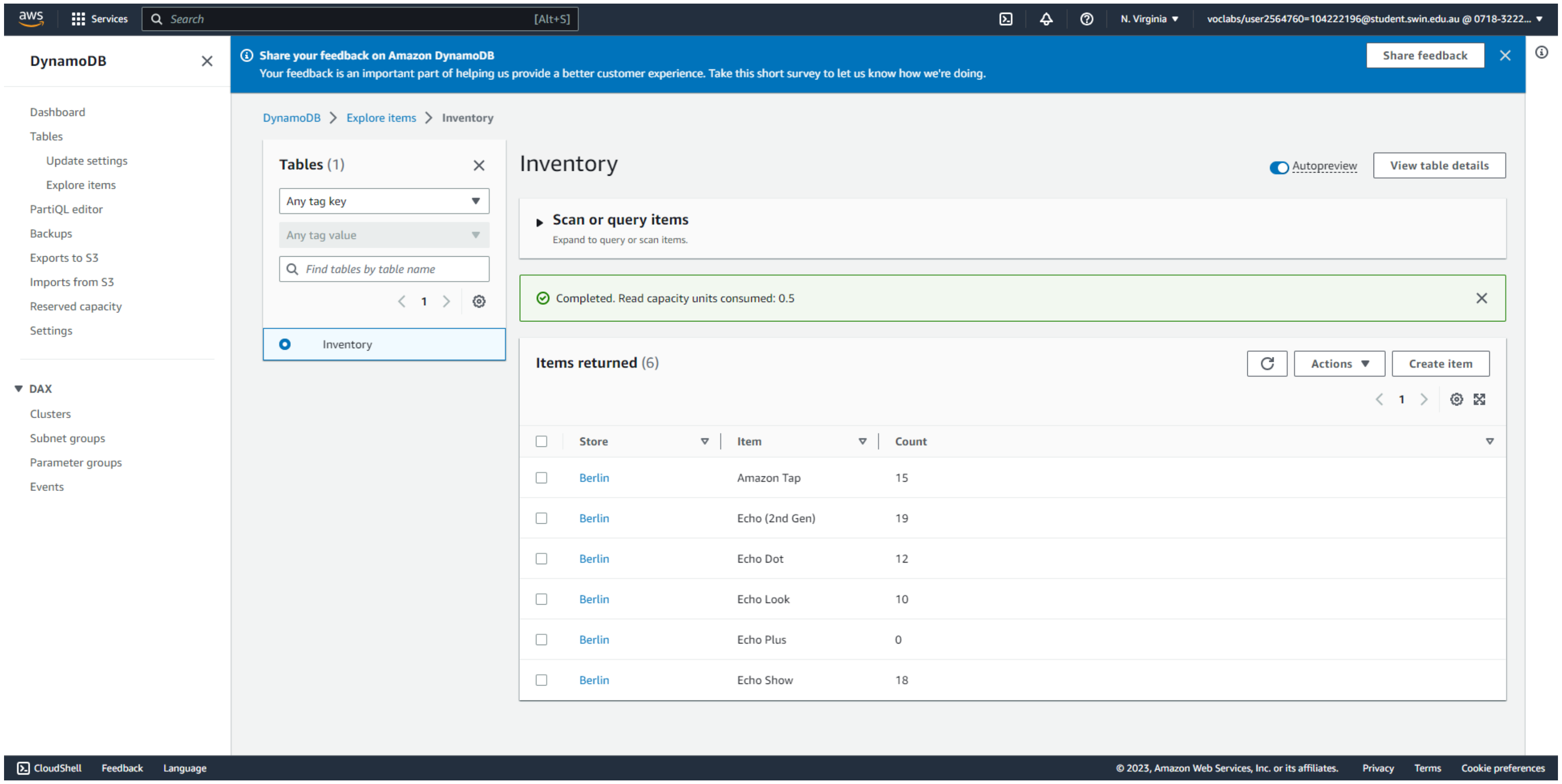
Store: Berlin

Store	Item	Count
Berlin	Amazon Tap	15
Berlin	Echo (2nd Gen)	19
Berlin	Echo Dot	12
Berlin	Echo Look	10
Berlin	Echo Plus	0
Berlin	Echo Show	18

Inventory
Item
Amazon Tap
Echo (2nd Gen)
Echo Dot
Echo Look
Echo Plus
Echo Show
0
10
20

This page uses an Amazon Cognito identity to retrieve data directly from Amazon DynamoDB.

The dashboard page correctly shows the data loaded from the inventory file, meaning that the Lambda function has been correctly executed.



Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Inventory

Scan or query items
Expand to query or scan items.

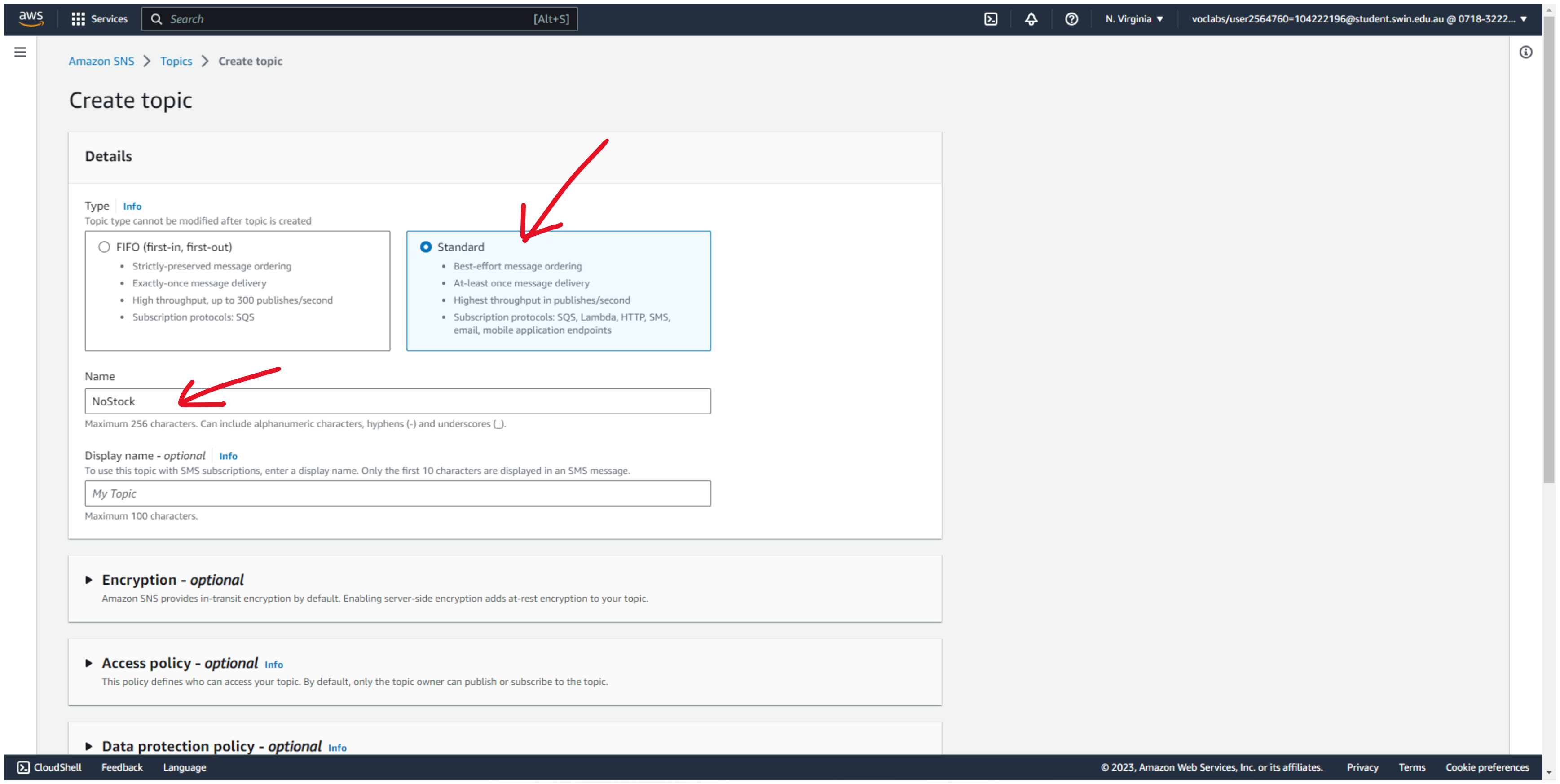
Completed. Read capacity units consumed: 0.5

Items returned (6)

Store	Item	Count
Berlin	Amazon Tap	15
Berlin	Echo (2nd Gen)	19
Berlin	Echo Dot	12
Berlin	Echo Look	10
Berlin	Echo Plus	0
Berlin	Echo Show	18

The data correctly appears in the DynamoDB database.

Task 4 - Configuring notifications


aws
Services
Search [Alt+S]
N. Virginia
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...
Amazon SNS > Topics > Create topic

Create topic

Details

Type Info
Topic type cannot be modified after topic is created

FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

[✓ Standard]
- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name
NoStock
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

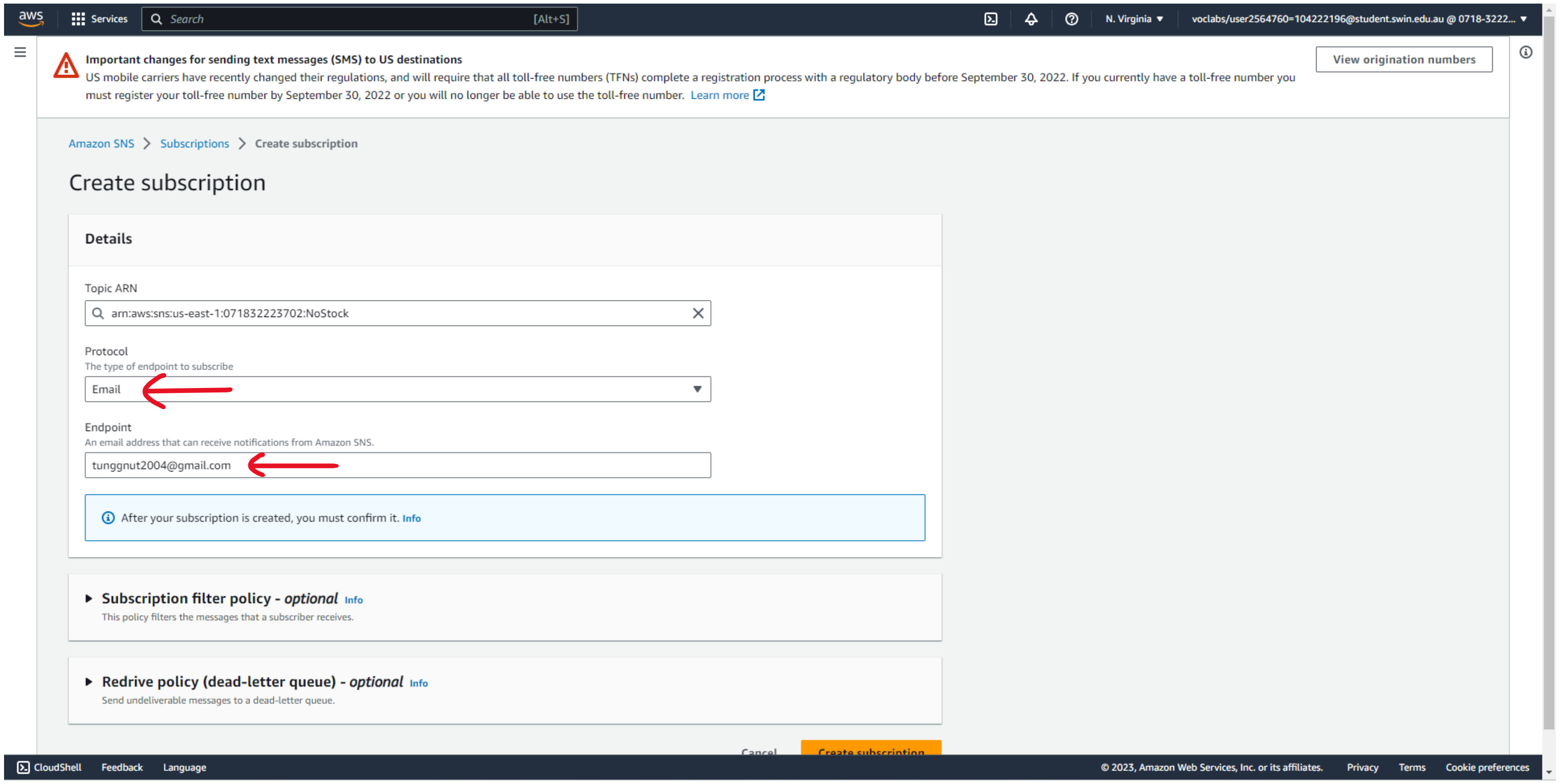
Display name - *optional* Info
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.
My Topic
Maximum 100 characters.

Encryption - *optional*
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

Access policy - *optional* Info
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

Data protection policy - *optional* Info

Create a standard SNS topic with the name of NoStock.


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...
View origination numbers
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Important changes for sending text messages (SMS) to US destinations
US mobile carriers have recently changed their regulations, and will require that all toll-free numbers (TFNs) complete a registration process with a regulatory body before September 30, 2022. If you currently have a toll-free number you must register your toll-free number by September 30, 2022 or you will no longer be able to use the toll-free number. Learn more

Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN: arn:aws:sns:us-east-1:071832223702:NoStock

Protocol
The type of endpoint to subscribe
Email

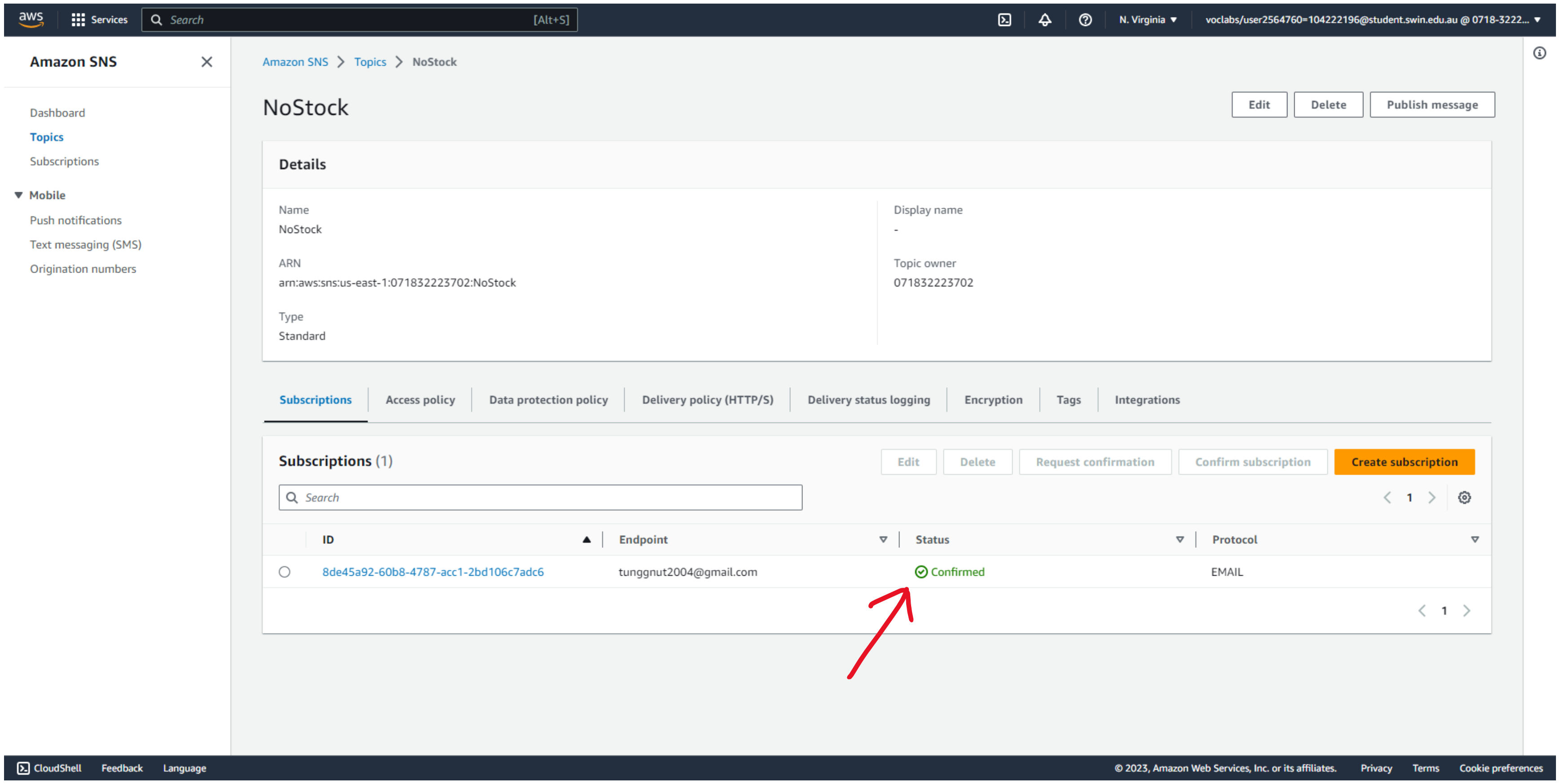
Endpoint
An email address that can receive notifications from Amazon SNS.
tunggnut2004@gmail.com

After your subscription is created, you must confirm it. Info

Subscription filter policy - *optional* Info
This policy filters the messages that a subscriber receives.

Redrive policy (dead-letter queue) - *optional* Info
Send undeliverable messages to a dead-letter queue.

Create an email subscription to the SNS topic.



Amazon SNS > Topics > NoStock

NoStock

Details

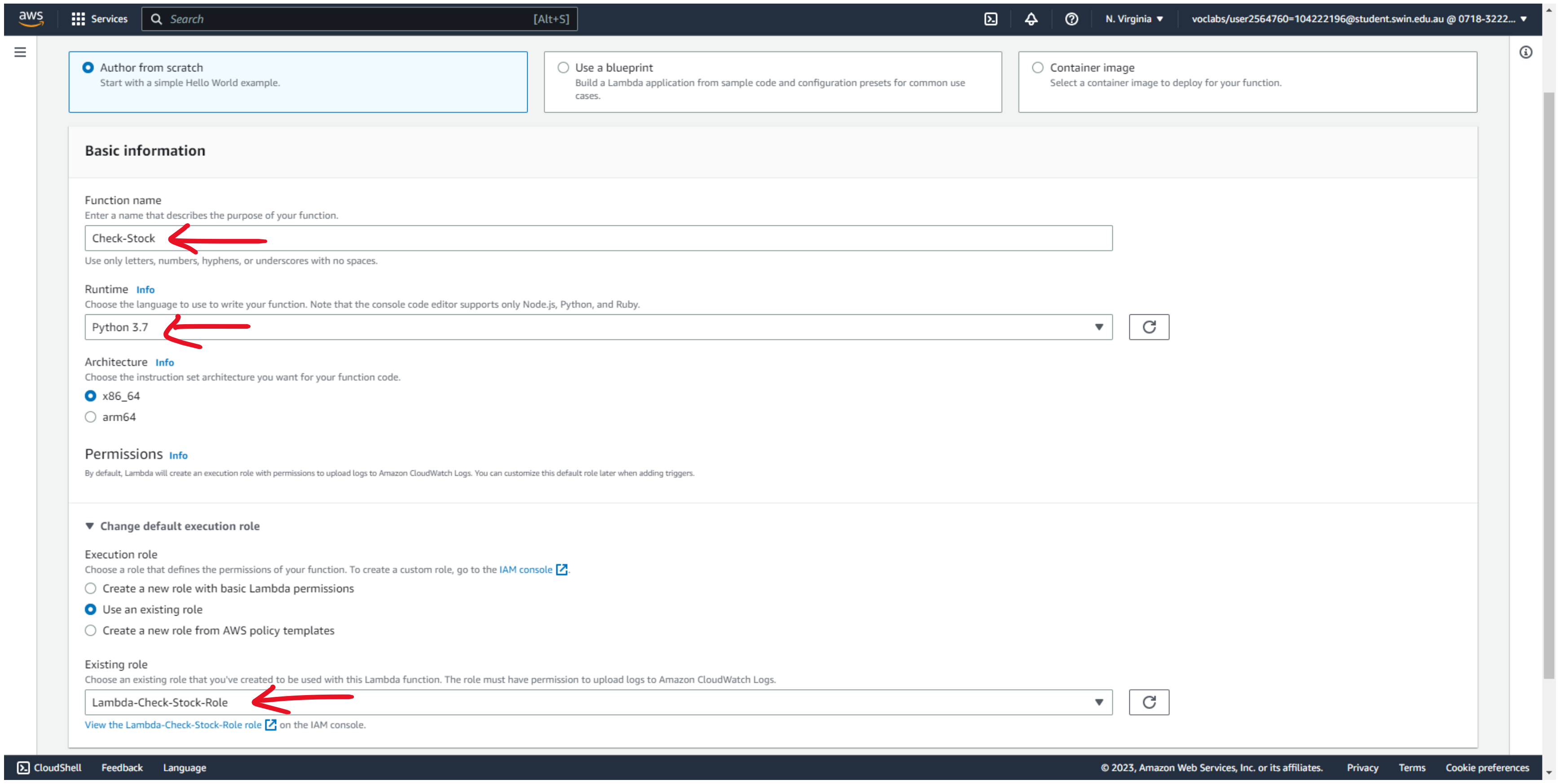
Name	NoStock
ARN	arn:aws:sns:us-east-1:071832223702:NoStock
Type	Standard
Display name	-
Topic owner	071832223702

Subscriptions (1)

ID	Endpoint	Status	Protocol
8de45a92-60b8-4787-acc1-2bd106c7adc6	tunggnut2004@gmail.com	Confirmed	EMAIL

Confirm the email subscription to the SNS topic.

Task 5 - Creating a Lambda function to send notifications



aws
Services
Search [Alt+S]
N. Virginia
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...

● 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.
Check-Stock
Use only letters, numbers, hyphens, or underscores with no spaces.

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 3.7

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

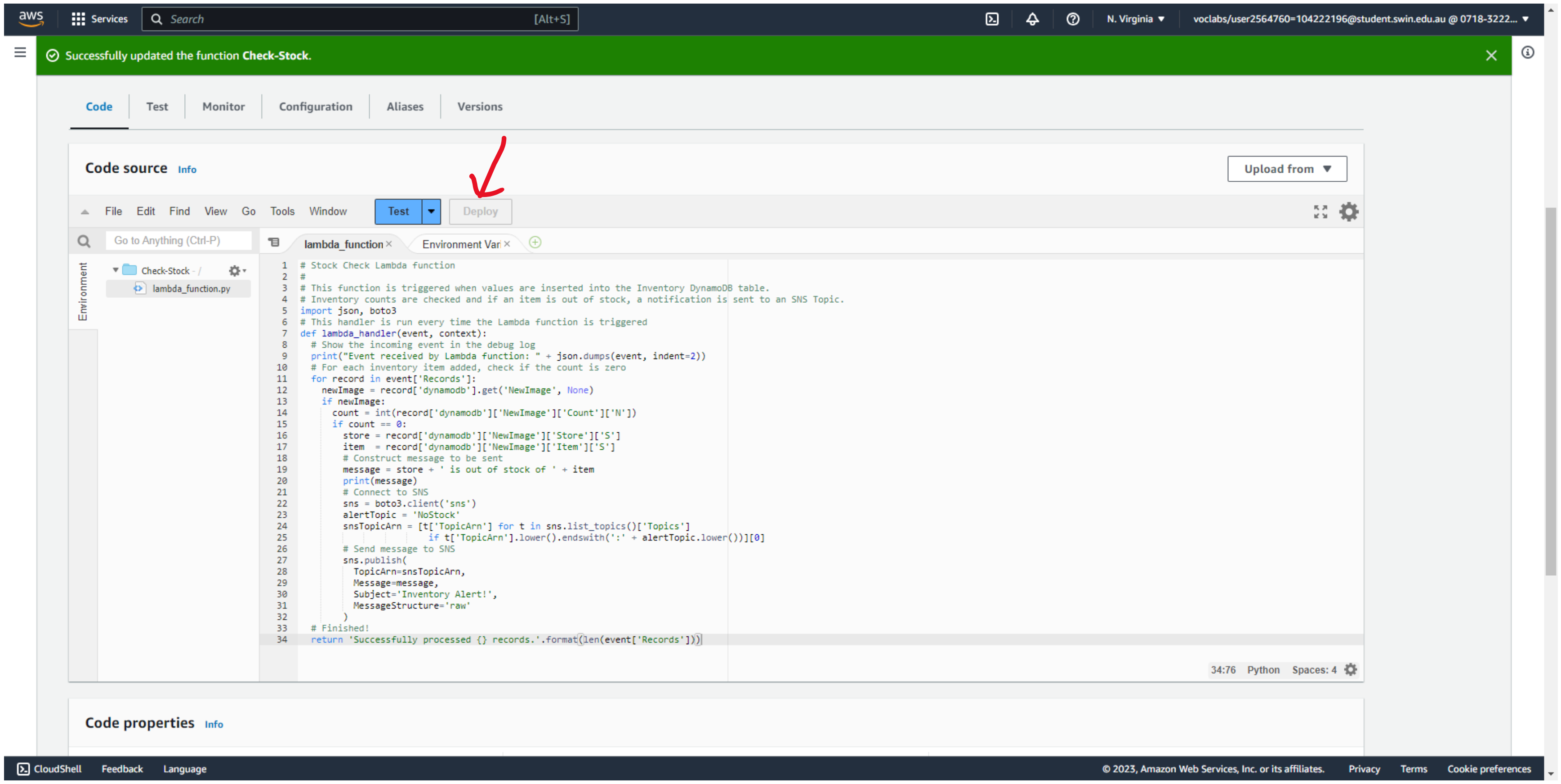
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ 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
○ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
Lambda-Check-Stock-Role
View the Lambda-Check-Stock-Role role on the IAM console.

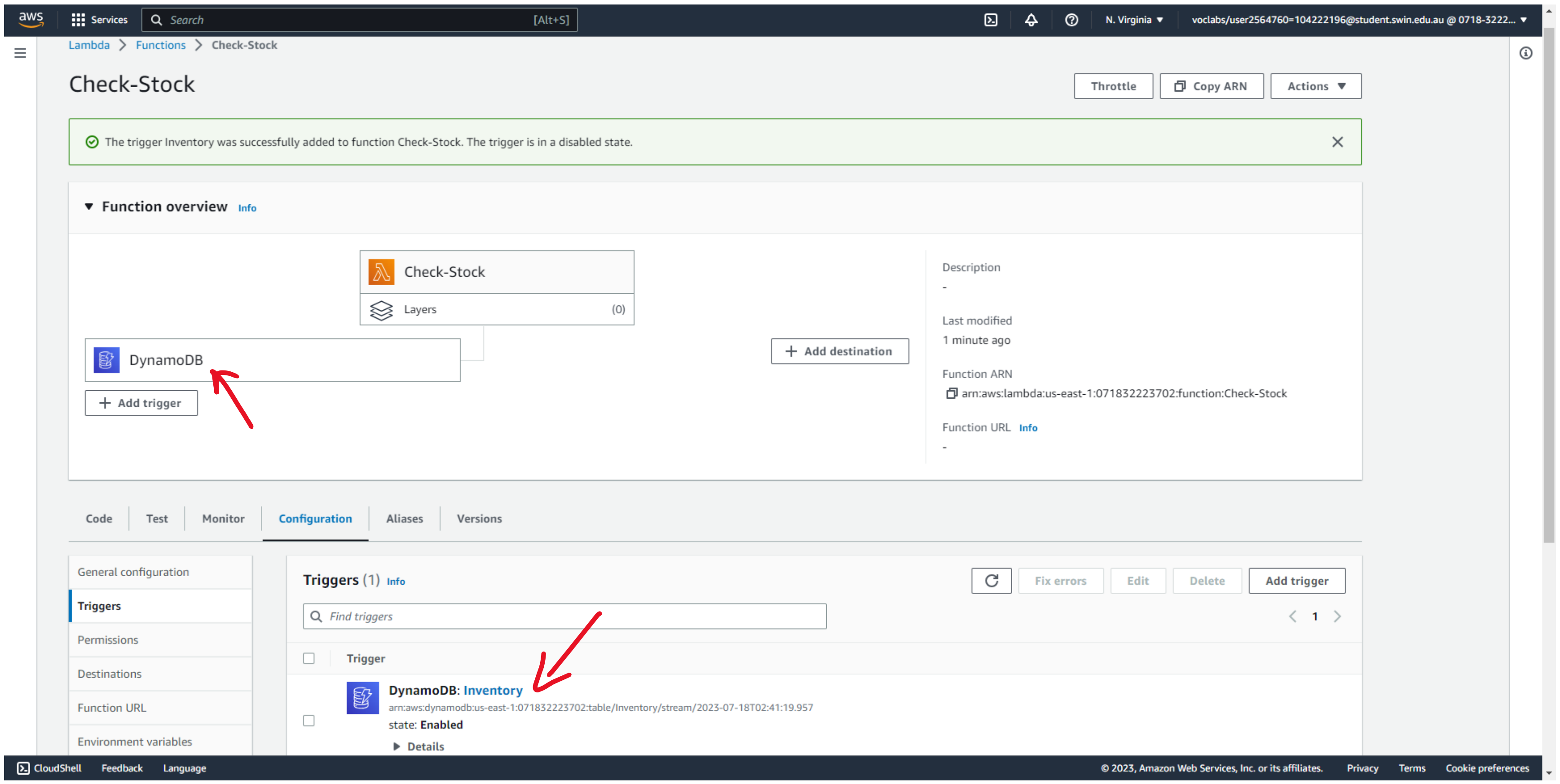
Create a new Lambda function to automatically sends email notifications, specifying its name, runtime, and IAM role.



Successfully updated the function **Check-Stock**.

```python
# Stock Check Lambda function
#
# This function is triggered when values are inserted into the Inventory DynamoDB table.
# Inventory counts are checked and if an item is out of stock, a notification is sent to an SNS Topic.
import json, boto3
# This handler is run every time the Lambda function is triggered
def lambda_handler(event, context):
  # Show the incoming event in the debug log
  print("Event received by Lambda function: " + json.dumps(event, indent=2))
  # For each inventory item added, check if the count is zero
  for record in event['Records']:
    newImage = record['dynamodb'].get('NewImage', None)
    if newImage:
      count = int(record['dynamodb']['NewImage']['Count']['N'])
      if count == 0:
        store = record['dynamodb']['NewImage']['Store']['S']
        item  = record['dynamodb']['NewImage']['Item']['S']
        # Construct message to be sent
        message = store + ' is out of stock of ' + item
        print(message)
        # Connect to SNS
        sns = boto3.client('sns')
        alertTopic = 'NoStock'
        snsTopicArn = [t['TopicArn'] for t in sns.list_topics()['Topics']
                if t['TopicArn'].lower().endswith(':' + alertTopic.lower())][0]
        # Send message to SNS
        sns.publish(
          TopicArn=snsTopicArn,
          Message=message,
          Subject='Inventory Alert!',
          MessageStructure='raw'
        )
  # Finished!
  return 'Successfully processed {} records.'.format(len(event['Records']))
```

After the function has been created, copy and paste the code, then deploy.


Services
Search [Alt+S]
N. Virginia
voclabs/user2564760=104222196@student.swin.edu.au @ 0718-3222...
Lambda > Functions > Check-Stock
Throttle
Copy ARN
Actions

Check-Stock

The trigger Inventory was successfully added to function Check-Stock. The trigger is in a disabled state.

Function overview Info

Description
-

Last modified
1 minute ago

Function ARN
arn:aws:lambda:us-east-1:071832223702:function:Check-Stock

Function URL Info
-

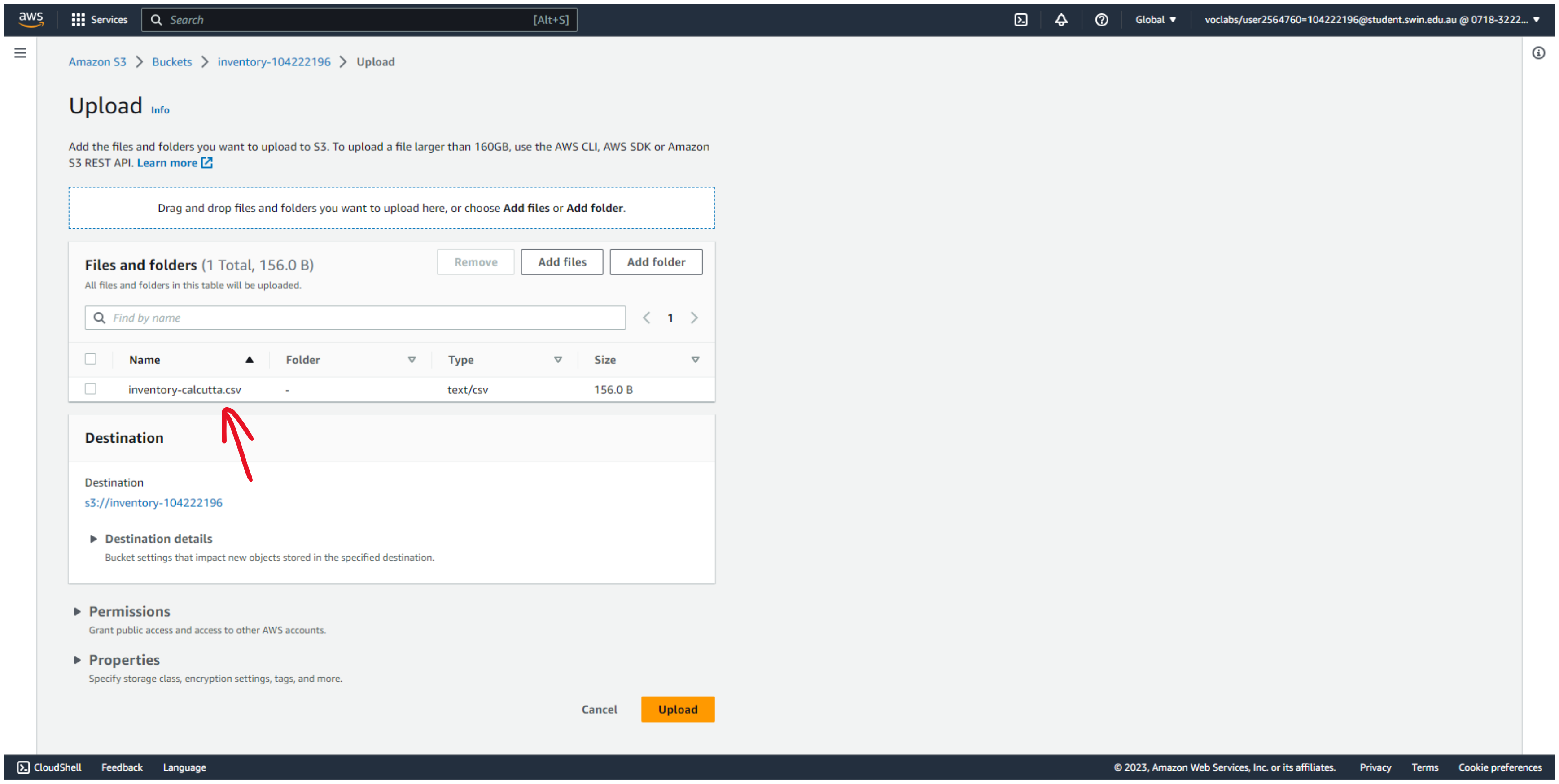
Triggers (1) Info

Trigger

DynamoDB: Inventory
arn:aws:dynamodb:us-east-1:071832223702:table/Inventory/stream/2023-07-18T02:41:19.957
state: Enabled
Details

Add a trigger to the Lambda function, specifying the Inventory table in DynamoDB.

Task 6 - Testing the System


Amazon S3 > Buckets > inventory-104222196 > Upload

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 SDK 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 (1 Total, 156.0 B)

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
inventory-calcutta.csv	-	text/csv	156.0 B

Destination

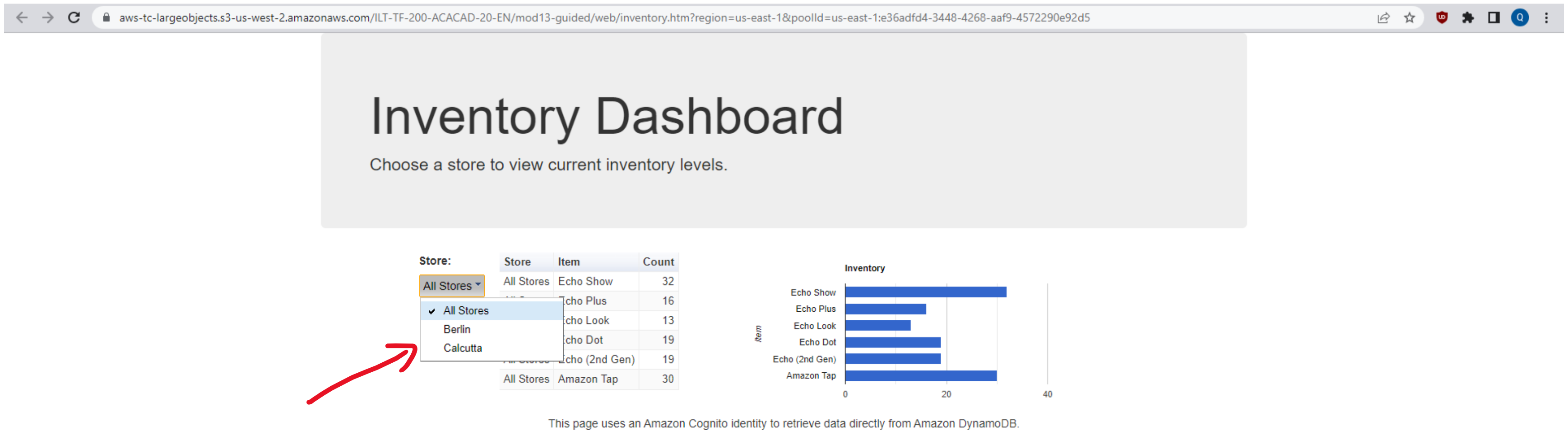
Destination
s3://inventory-104222196

Destination details
Bucket settings that impact new objects stored in the specified destination.

Permissions
Grant public access and access to other AWS accounts.

Properties
Specify storage class, encryption settings, tags, and more.

Upload another inventory file to S3.


aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod13-guided/web/inventory.htm?region=us-east-1&poolId=us-east-1:e36adfd4-3448-4268-aaf9-4572290e92d5
Inventory
Echo Show
Echo Plus
Echo Look
Echo Dot
Echo (2nd Gen)
Amazon Tap
Item
0
20
40

Inventory Dashboard

Choose a store to view current inventory levels.

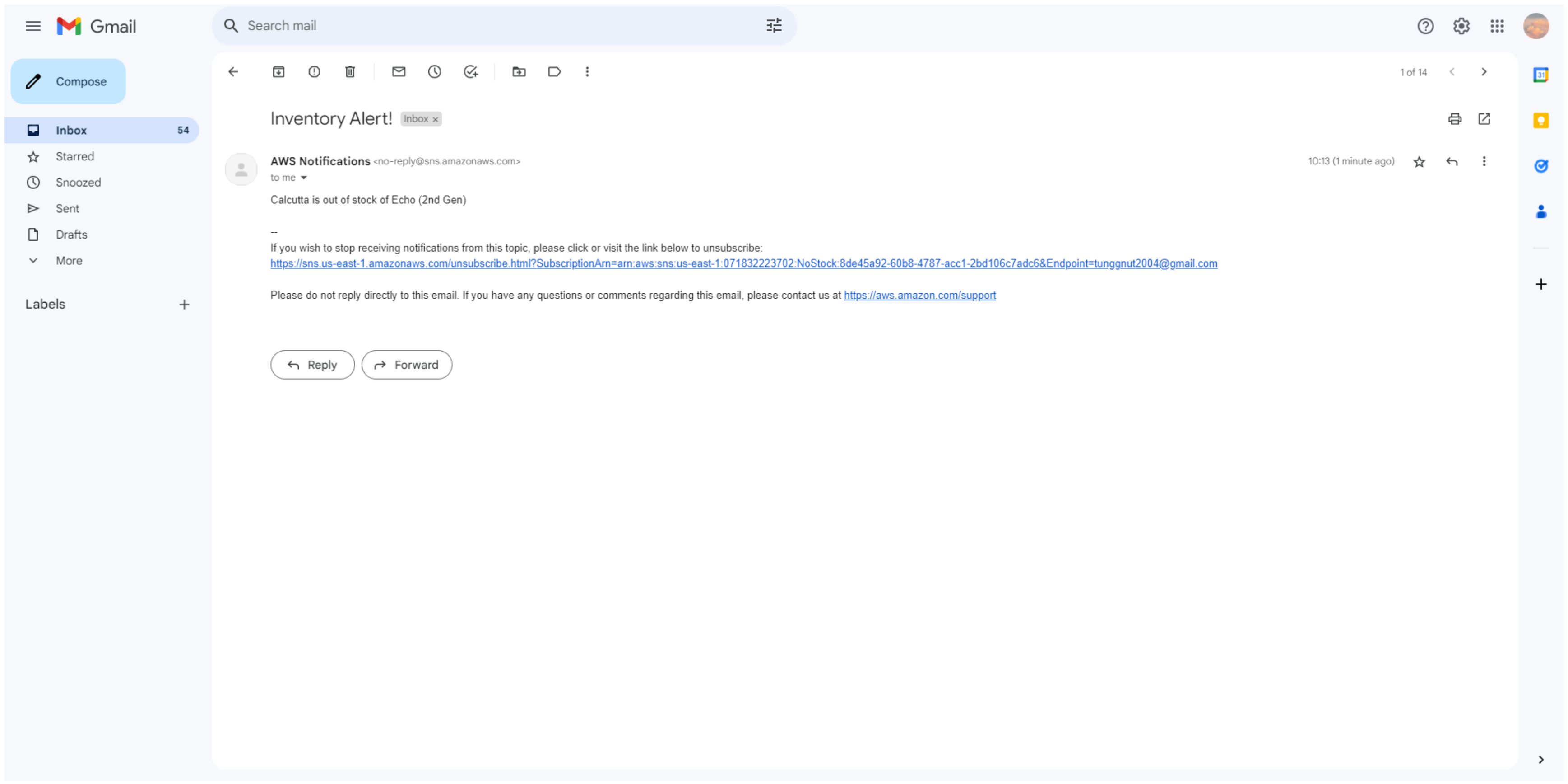
Store: All Stores

[✓ All Stores]
Berlin
Calcutta

Store	Item	Count
All Stores	Echo Show	32
	cho Plus	16
	cho Look	13
	cho Dot	19
	Echo (2nd Gen)	19
All Stores	Amazon Tap	30

This page uses an Amazon Cognito identity to retrieve data directly from Amazon DynamoDB.

The new inventory file is correctly read and loaded into the database, as evidenced by the presence of a new store.



Inventory Alert! Inbox

AWS Notifications <no-reply@sns.amazonaws.com>
to me

10:13 (1 minute ago)

Calcutta is out of stock of Echo (2nd Gen)

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:071832223702:NoStock:8de45a92-60b8-4787-acc1-2bd106c7adc6&Endpoint=tunggnut2004@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

An email notification is sent to the subscribed email to inform that an item is out of stock.