CLOUD COMPUTING

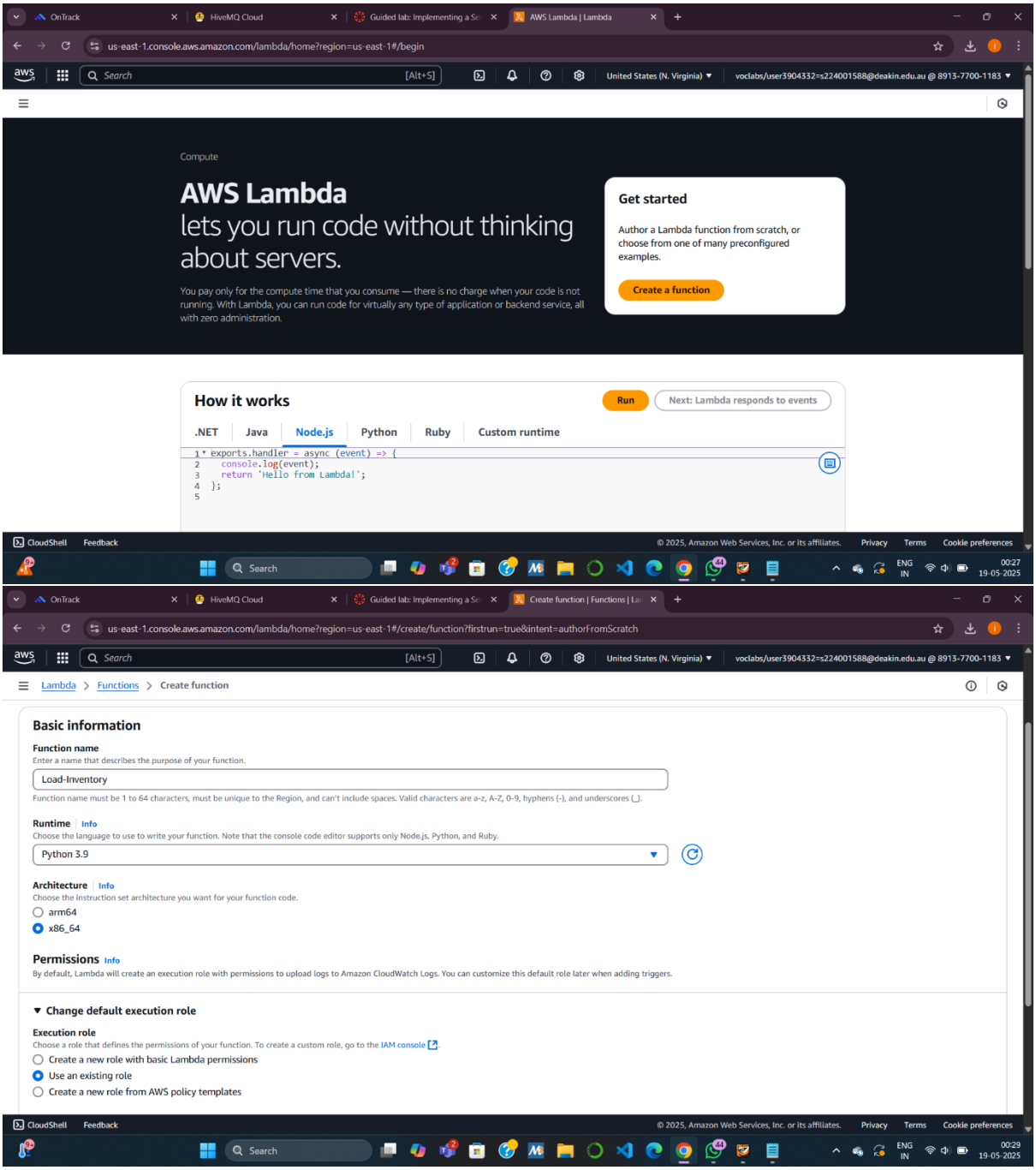
TASK 10.2D

Implementing a Serverless Architecture with AWS Lambda

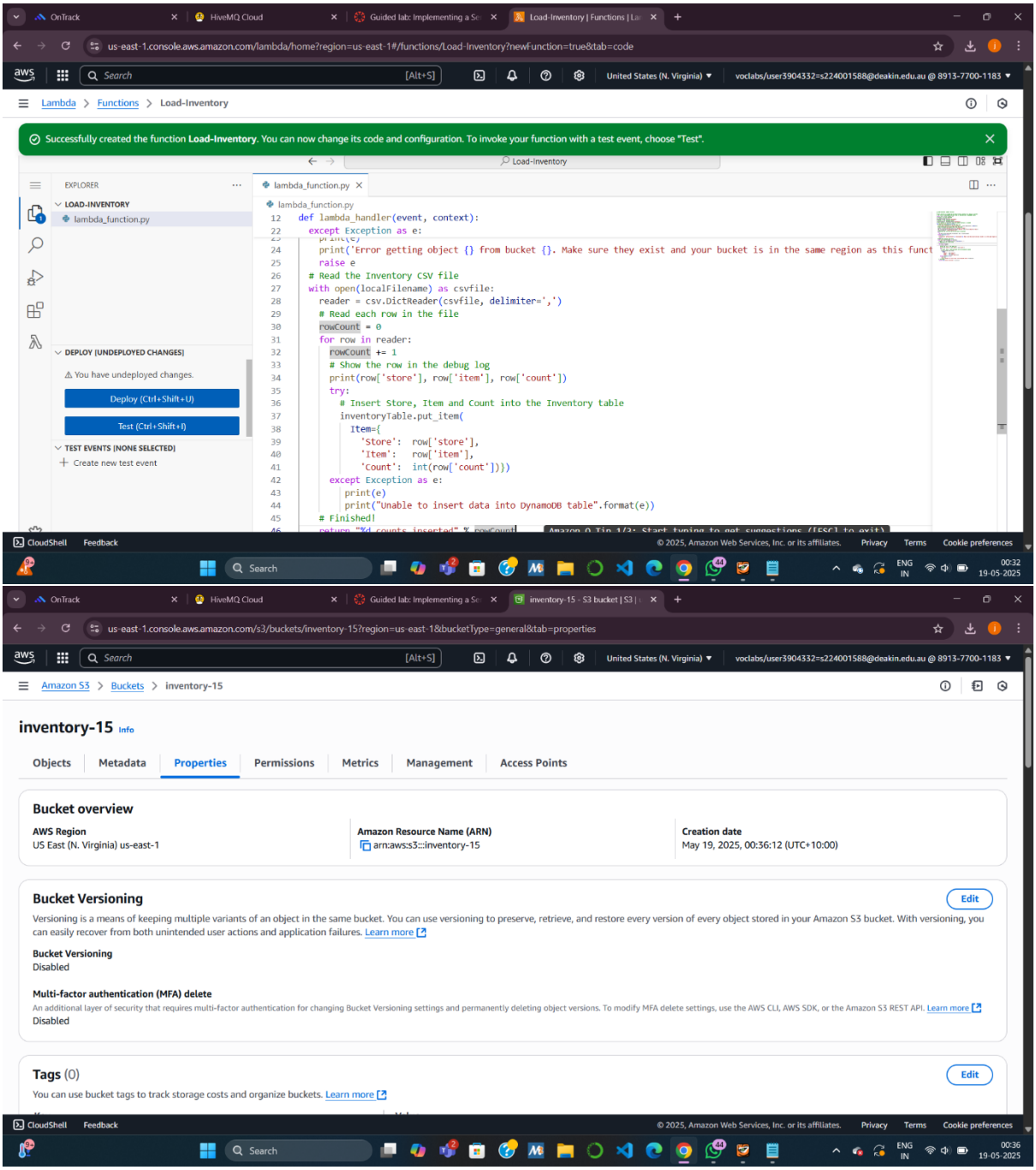
Task 1: Creating a Lambda Function to Load Data

Goal: Process a CSV inventory file uploaded to S3 and insert its contents into DynamoDB.

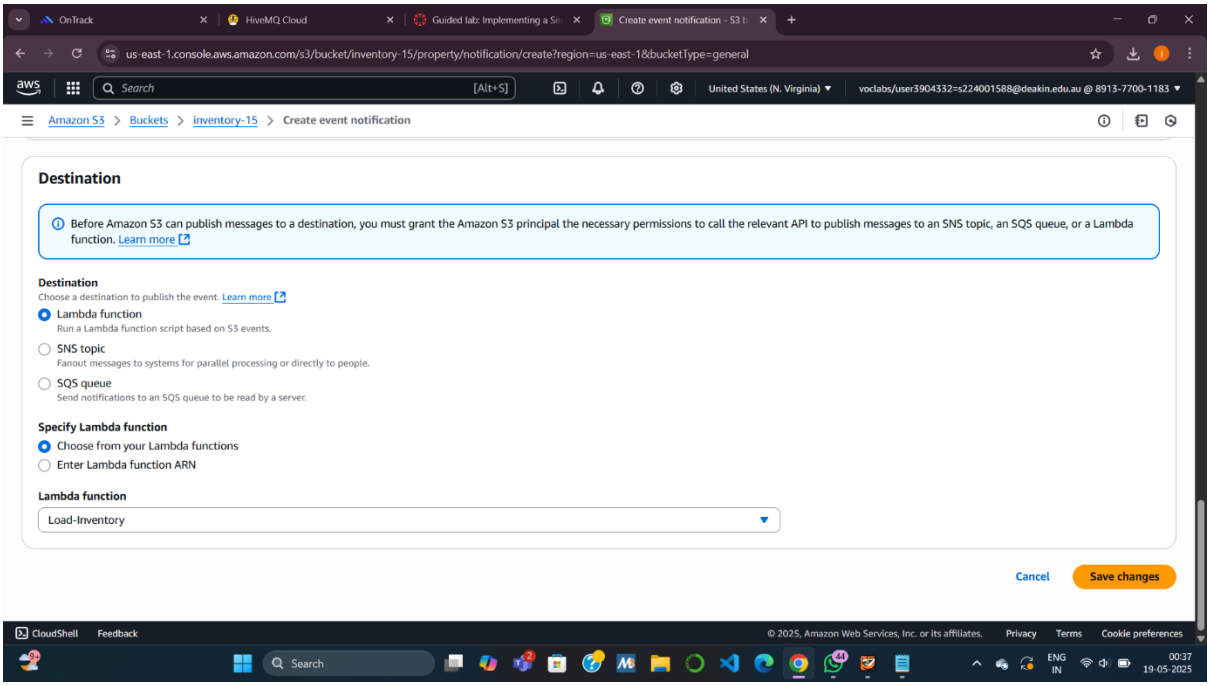
- **Create Lambda function (Load-Inventory)**:
  Uses Python 3.8, triggered by S3, inserts inventory rows into a DynamoDB table.
- **Attach existing role**:
  Lambda-Load-Inventory-Role gives permissions to access S3 and DynamoDB.
- **Paste provided code**:
  - Downloads file from S3.
  - Reads CSV line-by-line.
  - Adds each record to the Inventory DynamoDB table.
- **Deploy the function**:
  Save and deploy the code to make it active.


OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
AWS Lambda | Lambda
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/begin
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3904332=s224001588@deakin.edu.au @ 8913-7700-1183
Compute
AWS Lambda
lets you run code without thinking about servers.
You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.
Get started
Author a Lambda function from scratch, or choose from one of many preconfigured examples.
Create a function
How it works
Run
Next: Lambda responds to events
.NET
Java
Node.js
Python
Ruby
Custom runtime
exports.handler = async (event) => {
console.log(event);
return 'Hello from Lambda!';
};
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
00:27
19-05-2025

OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
Create function | Functions | La
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?firstrun=true&intent=authorFromScratch
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3904332=s224001588@deakin.edu.au @ 8913-7700-1183
Lambda
Functions
Create function
Basic information
Function name
Enter a name that describes the purpose of your function.
Load-Inventory
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 3.9
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
Create a new role from AWS policy templates
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
00:29
19-05-2025


Lambda > Functions > Load-Inventory
Successfully created the function Load-Inventory. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
LOAD-INVENTORY
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
def lambda_handler(event, context):
except Exception as e:
print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this funct
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!
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Amazon S3 > Buckets > inventory-15
inventory-15 Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Bucket overview
AWS Region
US East (N. Virginia) us-east-1
Amazon Resource Name (ARN)
arn:aws:s3:::inventory-15
Creation date
May 19, 2025, 00:36:12 (UTC+10:00)
Bucket Versioning
Edit
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disabled
Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled
Tags (0)
Edit
You can use bucket tags to track storage costs and organize buckets. Learn more
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

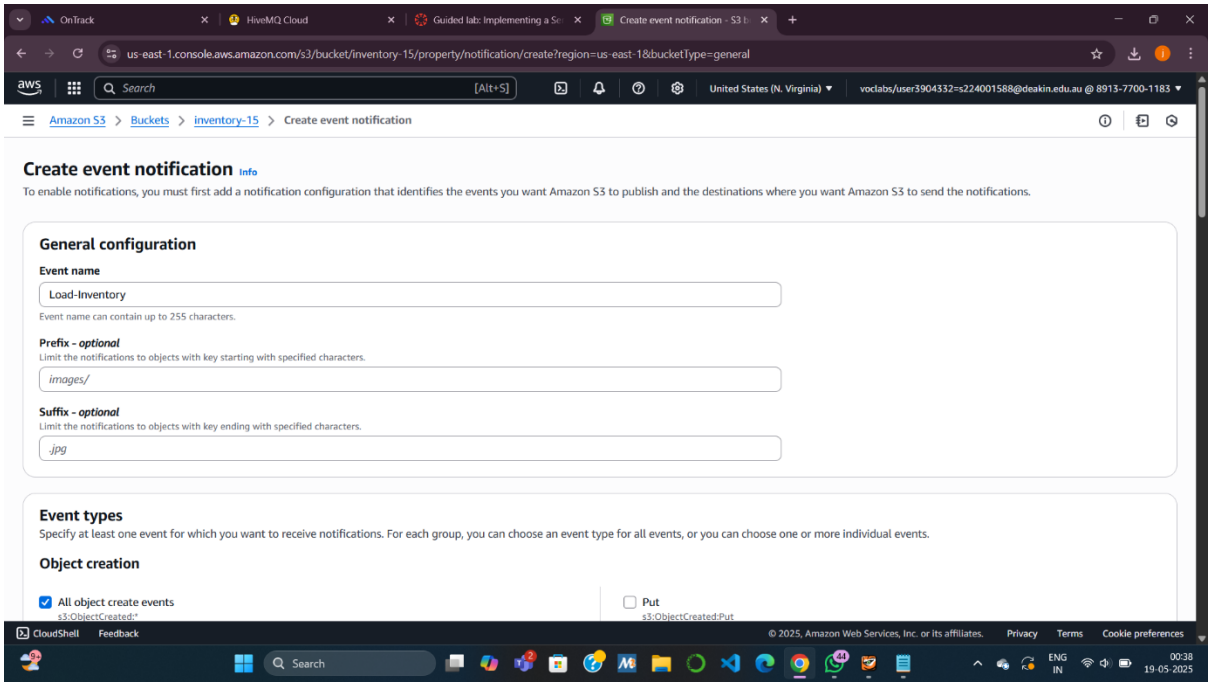


Task 2: Configuring an Amazon S3 Event

Goal: Automatically trigger the Lambda function when a CSV file is uploaded.

- **Create S3 bucket**:
  Use a unique name like inventory-123.
- **Add event notification**:
  - Trigger type: “All object create events”.
  - Destination: Lambda function (Load-Inventory).

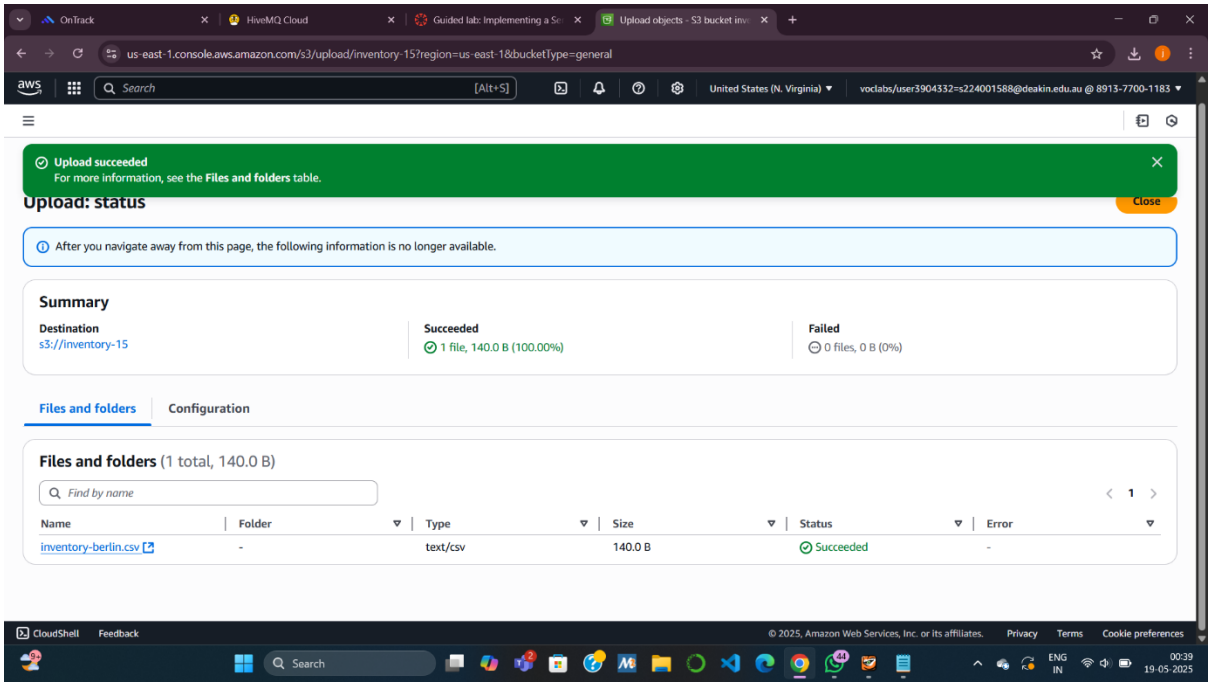
This links file uploads in the S3 bucket to your Lambda function.J

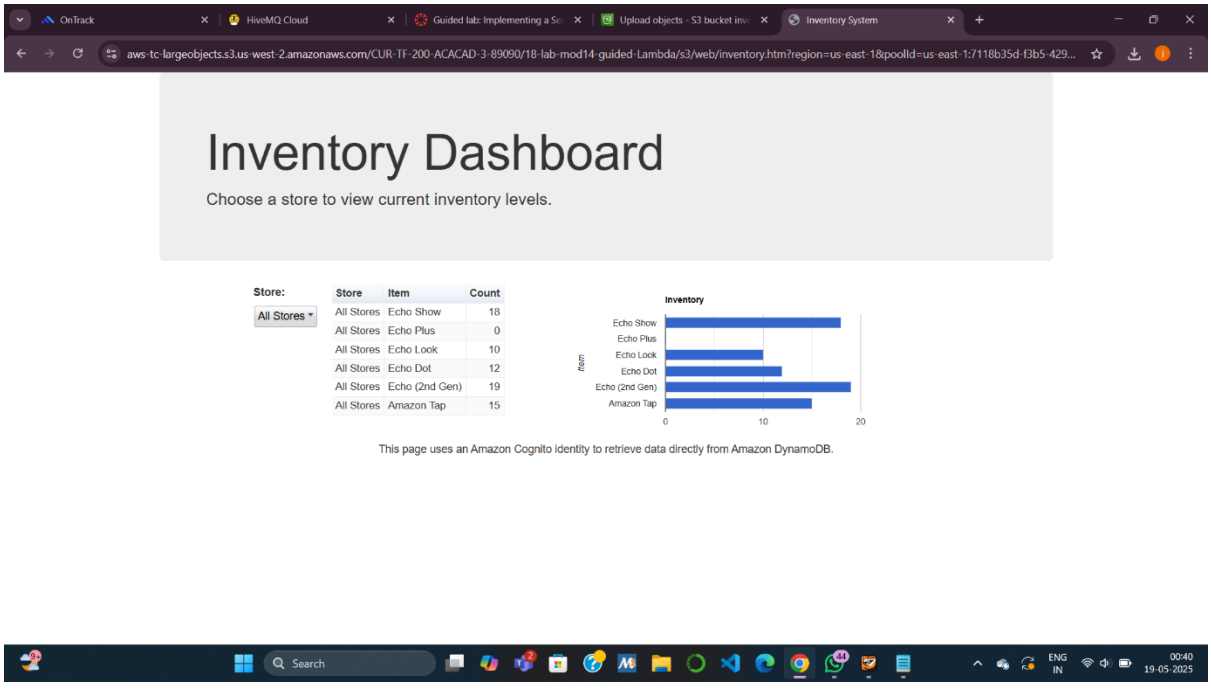


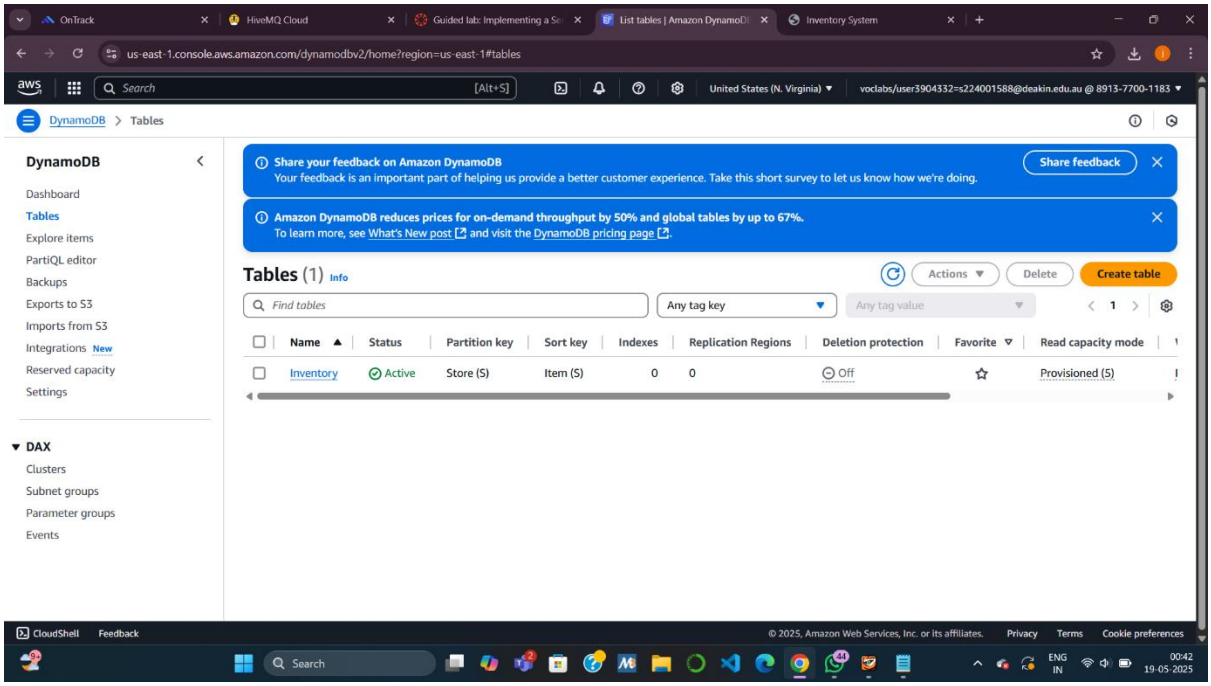
Task 3: Testing the Loading Process

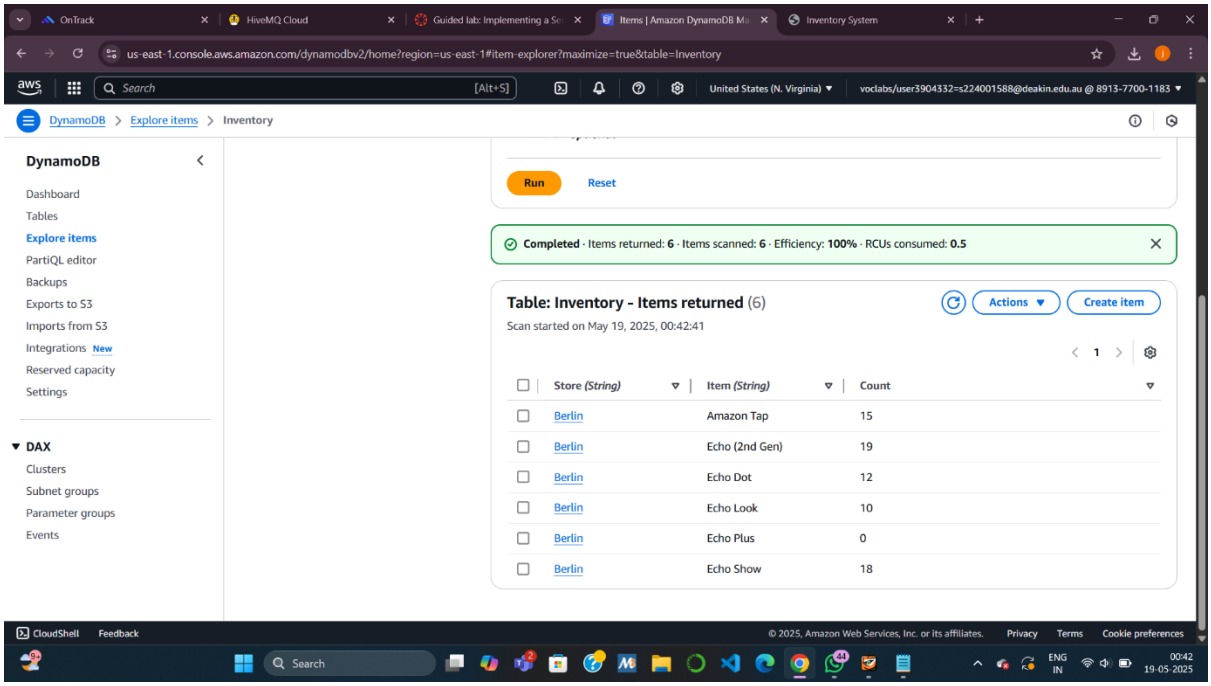
Goal: Test the setup by uploading an actual CSV file.

- **Download sample files**:
  Files like inventory-berlin.csv contain store, item, count.
- **Upload to S3 bucket**:
  Automatically triggers Load-Inventory.
- **View results**:
  - **Dashboard URL**: Opens a web app showing inventory (read from DynamoDB).
  - **DynamoDB Console**: Use *Explore table items* to confirm entries are inserted.


OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
Upload objects - S3 bucket inve
us-east-1.console.aws.amazon.com/s3/upload/inventory-15?region=us-east-1&bucketType=general
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3904332=s224001588@deakin.edu.au @ 8913-7700-1183
Upload succeeded
For more information, see the Files and folders table.
Upload: status
Close
After you navigate away from this page, the following information is no longer available.
Summary
Destination
s3://inventory-15
Succeeded
1 file, 140.0 B (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 total, 140.0 B)
Find by name
Name
Folder
Type
Size
Status
Error
inventory-berlin.csv
-
text/csv
140.0 B
Succeeded
-
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
00:39
19-05-2025


OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
Upload objects - S3 bucket inve
Inventory System
aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACAD-3-89090/18-lab-mod14-guided-Lambda/s3/web/inventory.htm?region=us-east-1&poolId=us-east-1:7118b35d-f3b5-429...
Inventory Dashboard
Choose a store to view current inventory levels.
Store:
All Stores
Store
Item
Count
All Stores
Echo Show
18
All Stores
Echo Plus
0
All Stores
Echo Look
10
All Stores
Echo Dot
12
All Stores
Echo (2nd Gen)
19
All Stores
Amazon Tap
15
Inventory
Item
Echo Show
Echo Plus
Echo Look
Echo Dot
Echo (2nd Gen)
Amazon Tap
0
10
20
This page uses an Amazon Cognito identity to retrieve data directly from Amazon DynamoDB.
Search
ENG
IN
00:40
19-05-2025



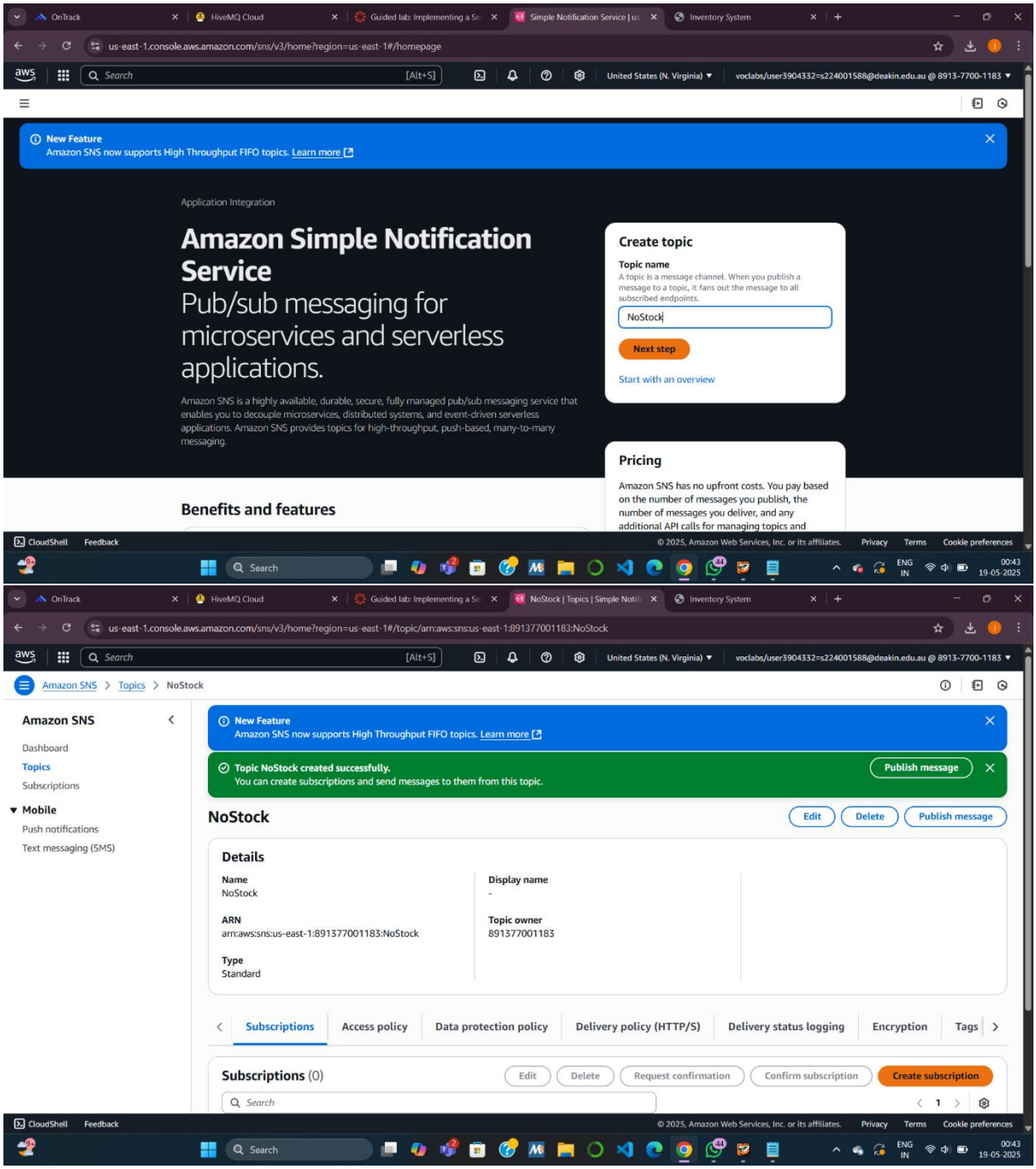


Task 4: Configuring Notifications with Amazon SNS

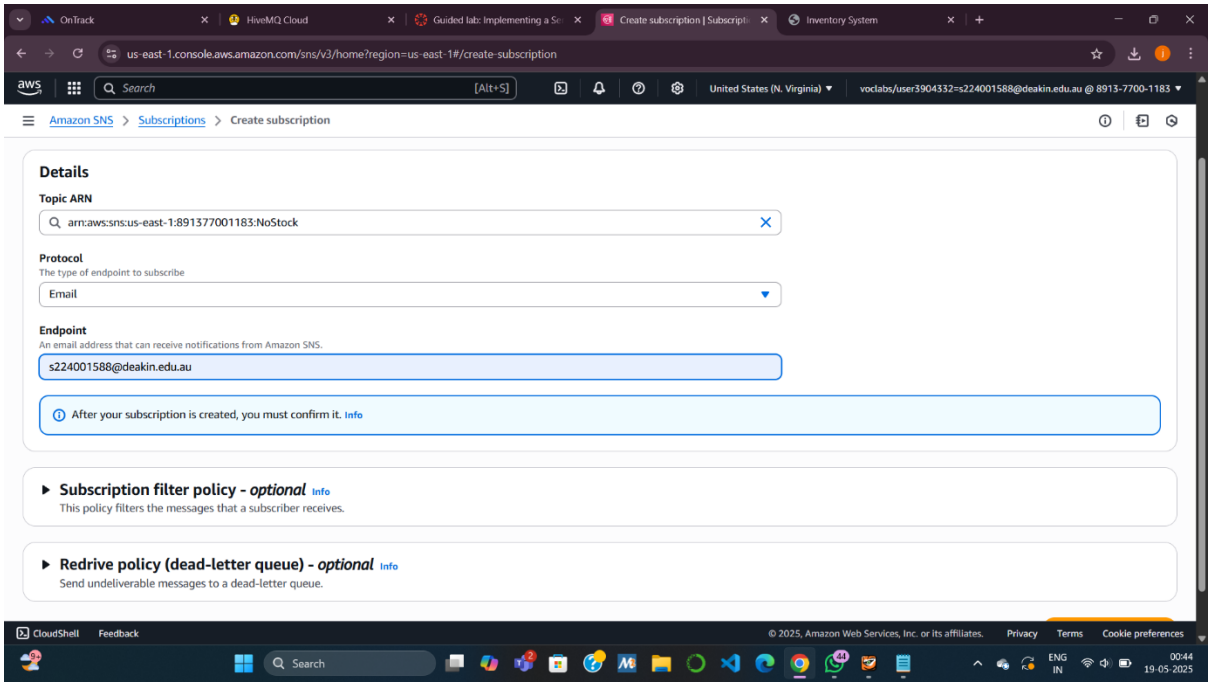
Goal: Set up an SNS topic to alert users when inventory is zero.

- **Create SNS Topic**:
  Name it NoStock.
- **Subscribe to topic**:
  - Choose **Email** as protocol.
  - Enter your email → Confirm via email link.

Now SNS is ready to send alerts when items are out of stock.


New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more
Application Integration
Amazon Simple Notification Service
Pub/sub messaging for microservices and serverless applications.
Amazon SNS is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and event-driven serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging.
Create topic
Topic name
A topic is a message channel. When you publish a message to a topic, it fans out the message to all subscribed endpoints.
NoStock
Next step
Start with an overview
Pricing
Amazon SNS has no upfront costs. You pay based on the number of messages you publish, the number of messages you deliver, and any additional API calls for managing topics and
Benefits and features
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

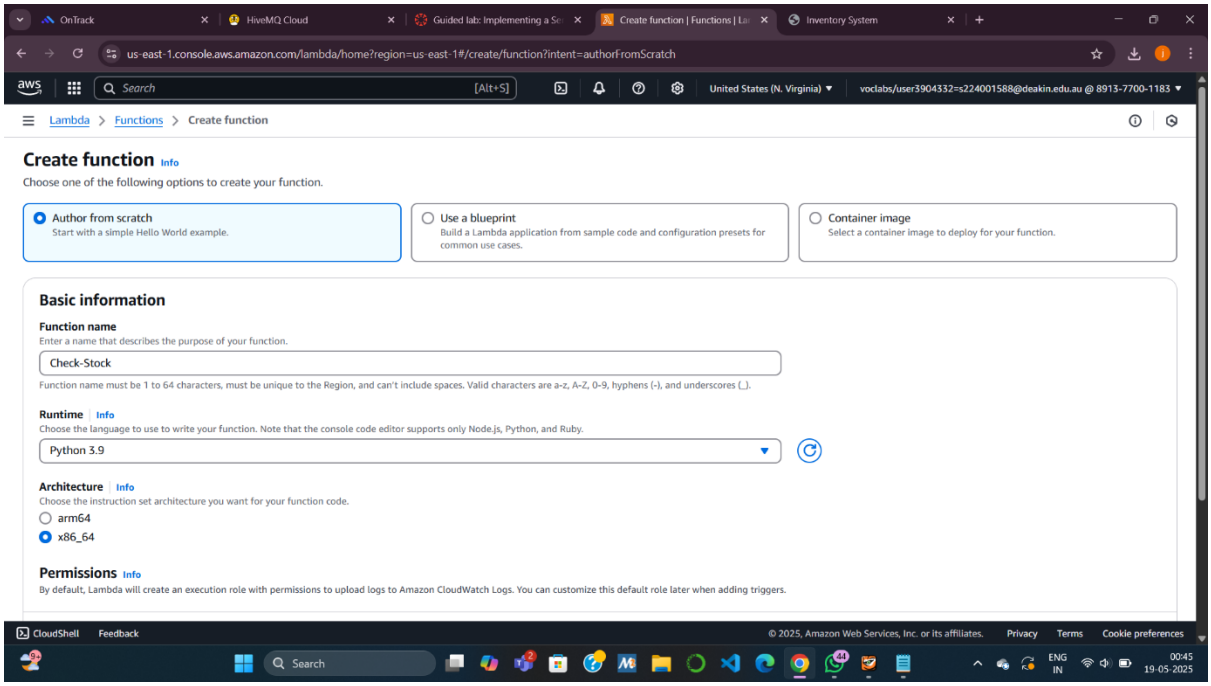
Amazon SNS > Topics > NoStock
Amazon SNS
Dashboard
Topics
Subscriptions
Mobile
Push notifications
Text messaging (SMS)
New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more
Topic NoStock created successfully.
You can create subscriptions and send messages to them from this topic.
Publish message
NoStock
Edit
Delete
Publish message
Details
Name
NoStock
Display name
-
ARN
arn:aws:sns:us-east-1:891377001183:NoStock
Topic owner
891377001183
Type
Standard
Subscriptions
Access policy
Data protection policy
Delivery policy (HTTP/S)
Delivery status logging
Encryption
Tags
Subscriptions (0)
Edit
Delete
Request confirmation
Confirm subscription
Create subscription
Search
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

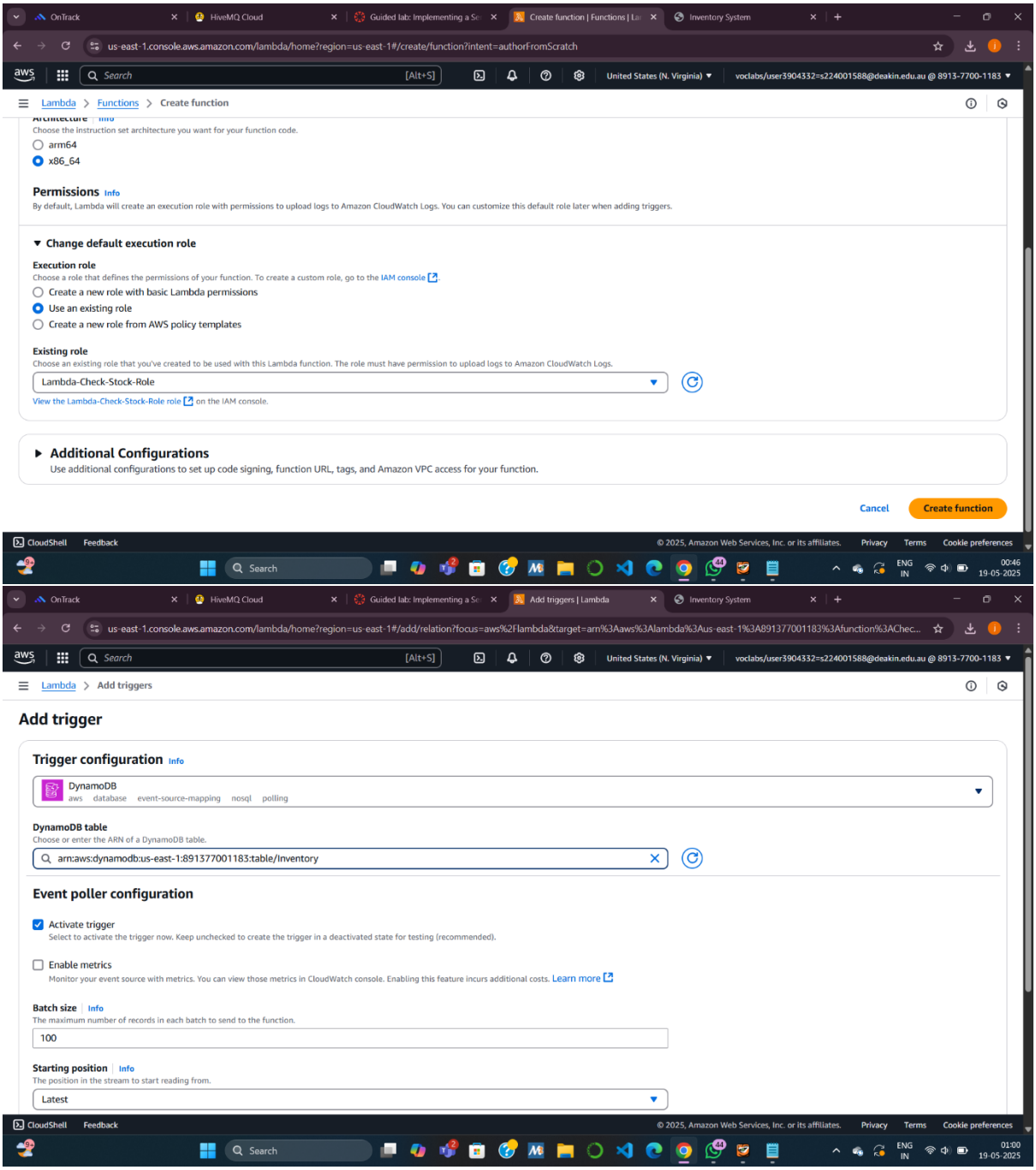


Task 5: Creating a Lambda Function to Send Notifications

Goal: Notify via SNS when item count is 0 in the database.

- **Create Lambda function (Check-Stock)**:
    - Triggered by **DynamoDB Streams** when new item is added.
    - If count == 0, send alert to NoStock SNS topic.
- **Attach execution role**:
  Lambda-Check-Stock-Role allows access to SNS.
- **Deploy and Add Trigger**:
    - Add DynamoDB Stream trigger on Inventory table.
    - Enables real-time stock checking and alerting.

- 
OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
Create function | Functions | La
Inventory System
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?intent=authorFromScratch
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3904332=s224001588@deakin.edu.au @ 8913-7700-1183
Lambda > Functions > Create function
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.
Check-Stock
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 3.9
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.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
00:45
19-05-2025

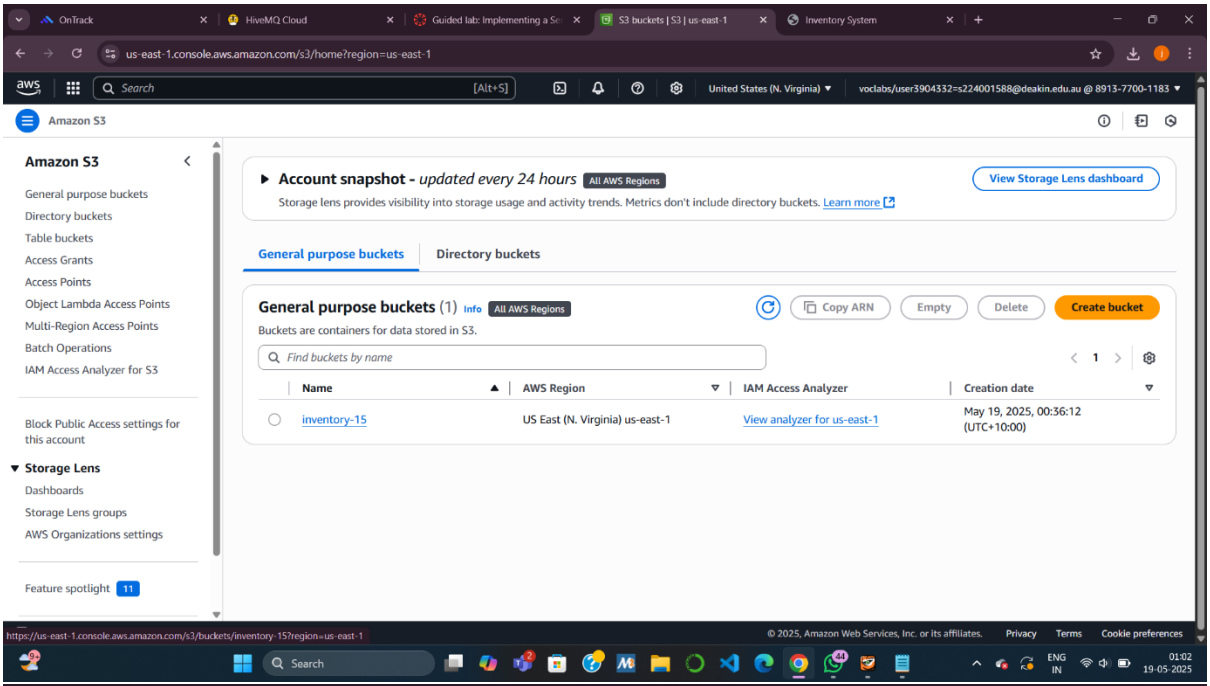
- 

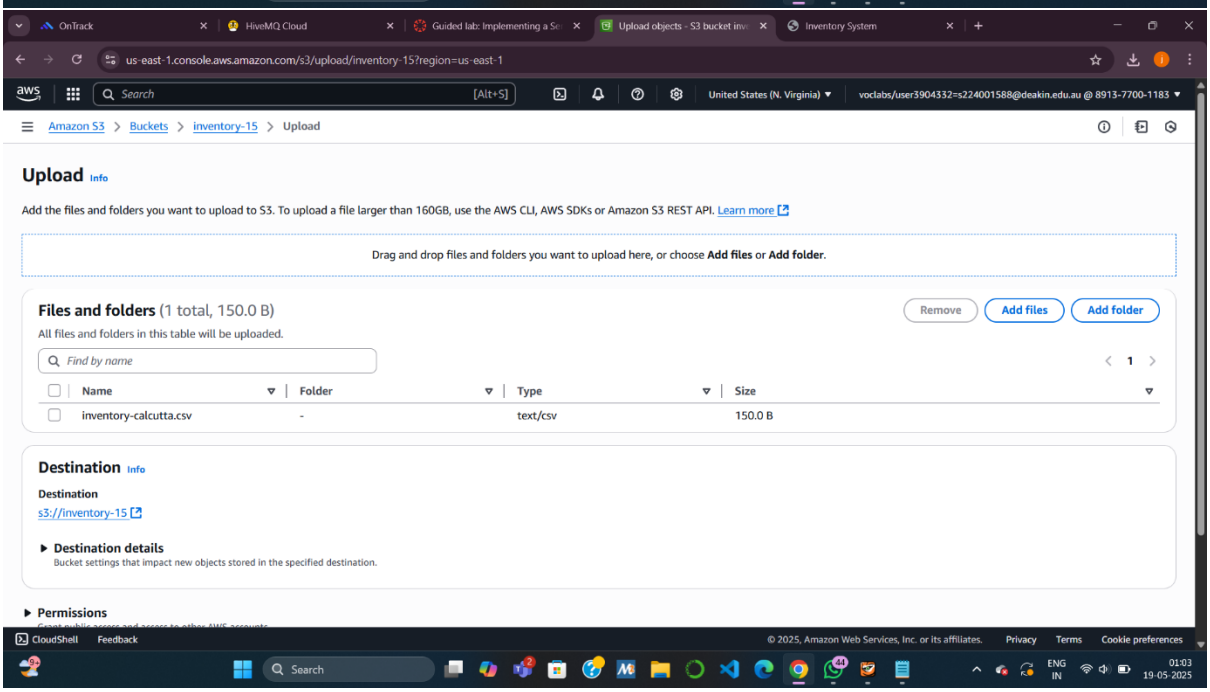
Task 6: Testing the System

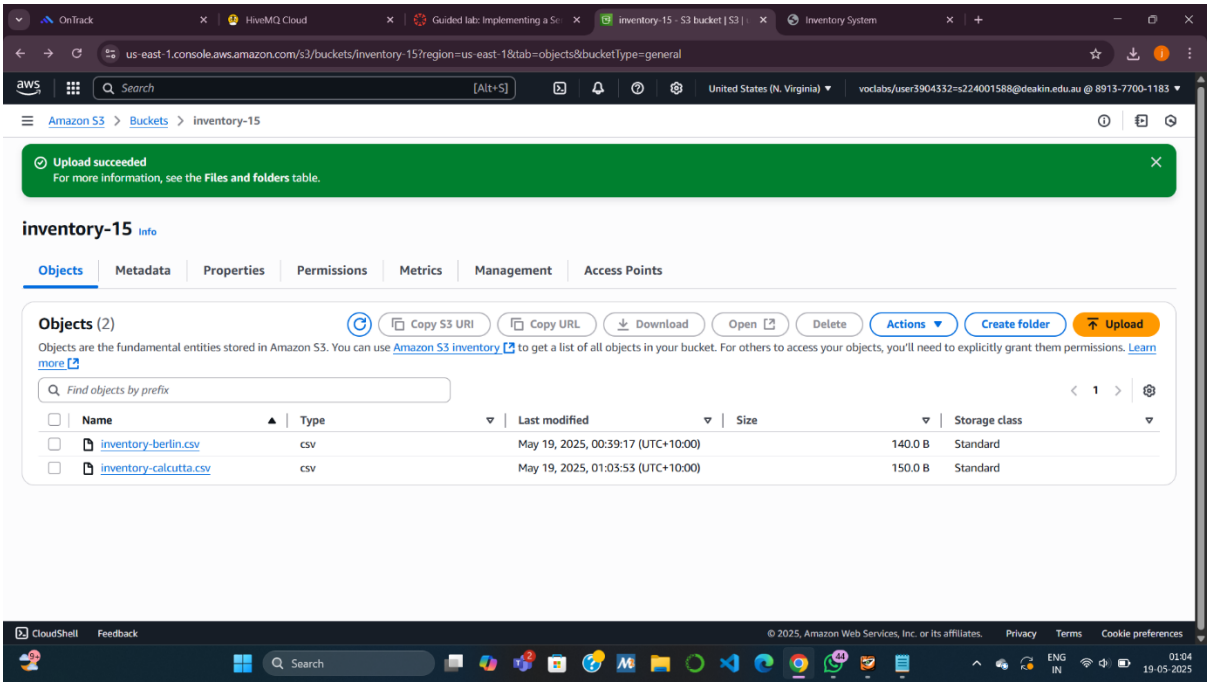
Goal: Validate the full flow (S3 → Lambda → DynamoDB → SNS).

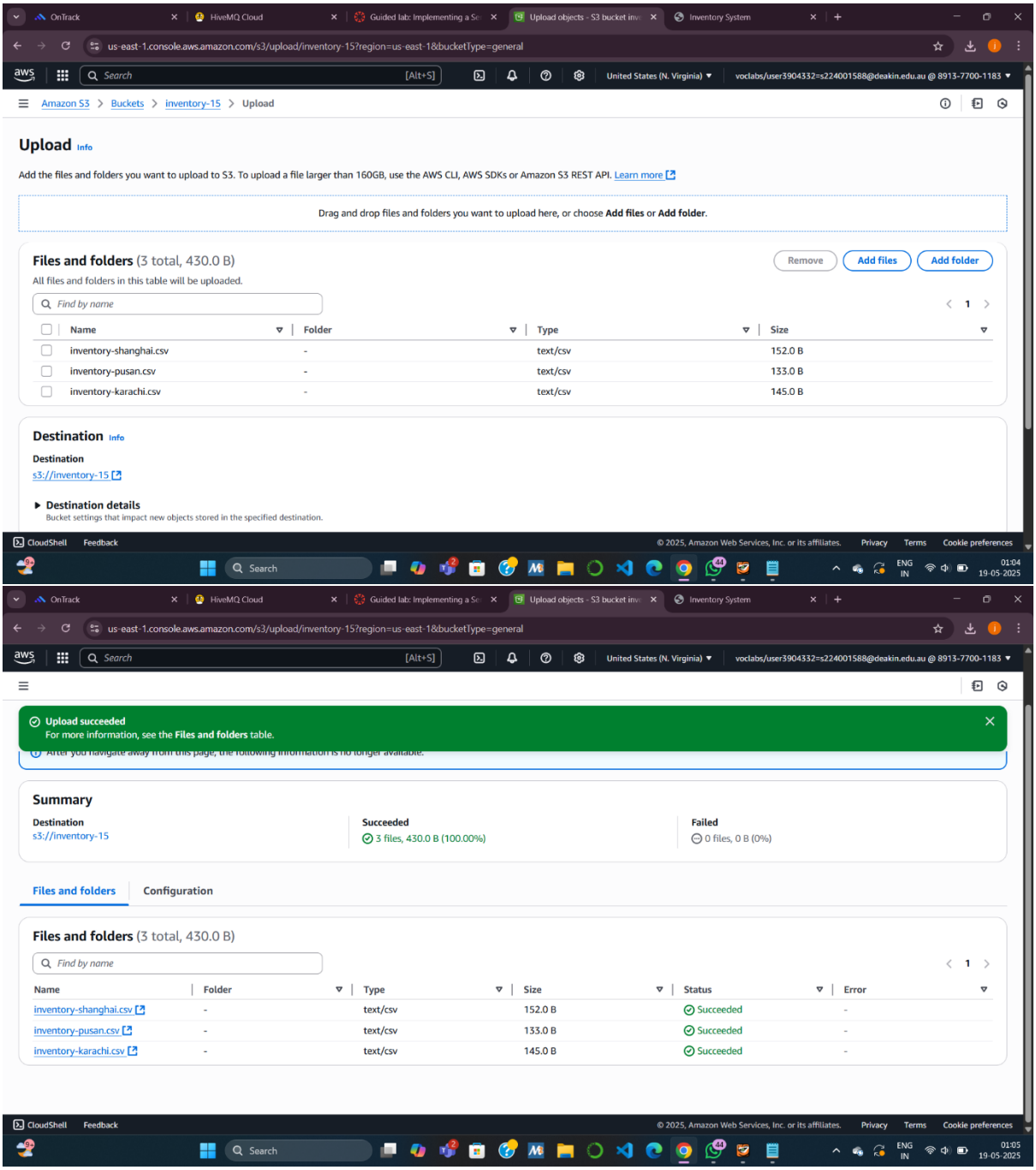
- **Upload another CSV**:
  Triggers full pipeline:
    - Data added to DynamoDB.
    - Check-Stock function invoked.
    - Sends email if count is 0.
- **Dashboard update**:
  Refresh to see new store's inventory.

- **Check email**:
  You should receive an "Out of Stock" alert.
- **Bonus:**
  Try uploading **multiple files** together — all will trigger the process in parallel due to scalability of serverless architecture.

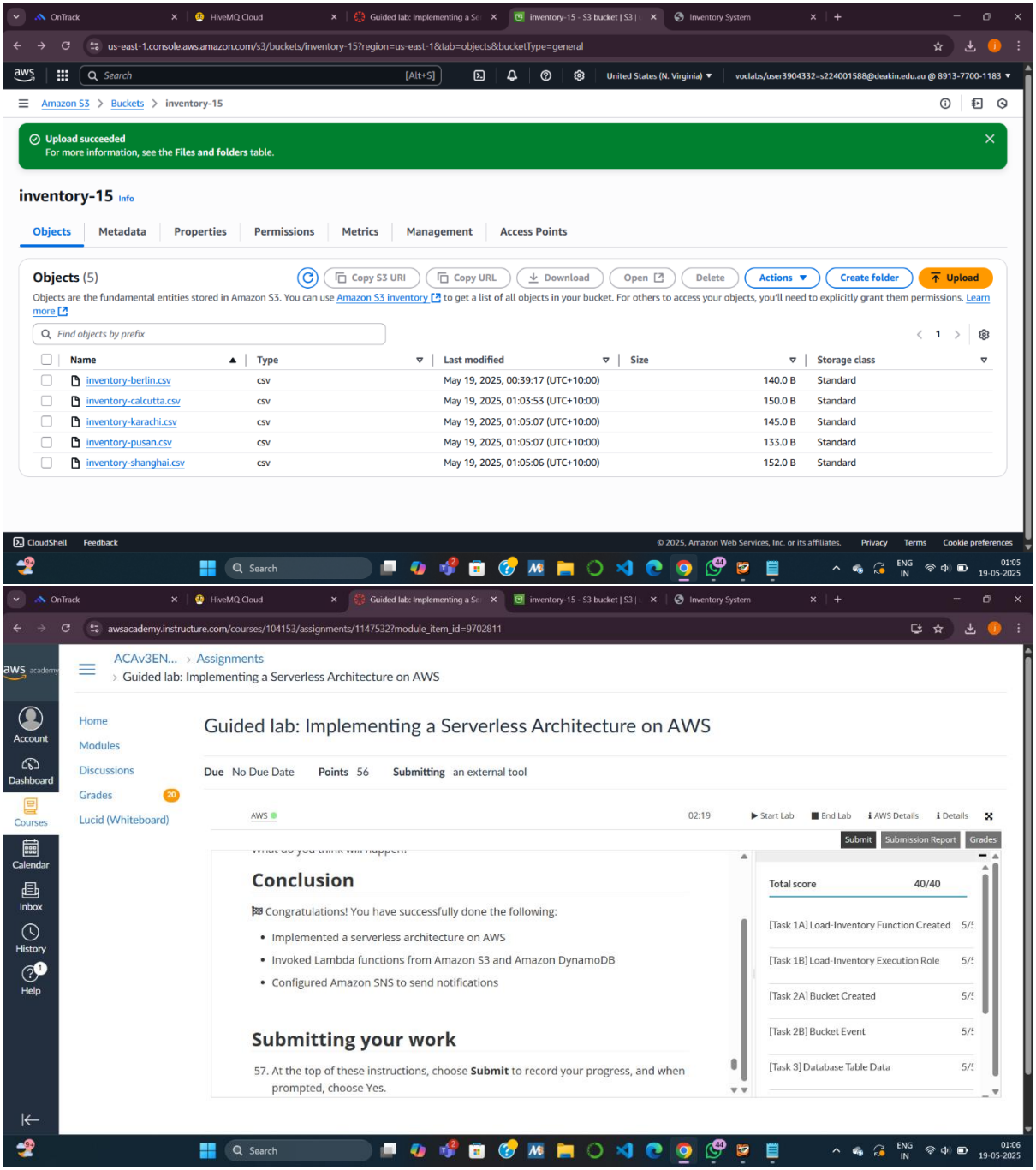

us-east-1.console.aws.amazon.com/s3/home?region=us-east-1
Amazon S3
Account snapshot - updated every 24 hours
All AWS Regions
View Storage Lens dashboard
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more
General purpose buckets
Directory buckets
General purpose buckets (1) Info
All AWS Regions
Copy ARN
Empty
Delete
Create bucket
Buckets are containers for data stored in S3.
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
inventory-15
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 19, 2025, 00:36:12 (UTC+10:00)
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight
11
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/s3/upload/inventory-15?region=us-east-1
Amazon S3 > Buckets > inventory-15 > 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 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 (1 total, 150.0 B)
All files and folders in this table will be uploaded.
Remove
Add files
Add folder
Find by name
Name
Folder
Type
Size
inventory-calcutta.csv
-
text/csv
150.0 B
Destination Info
Destination
s3://inventory-15
Destination details
Bucket settings that impact new objects stored in the specified destination.
Permissions
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

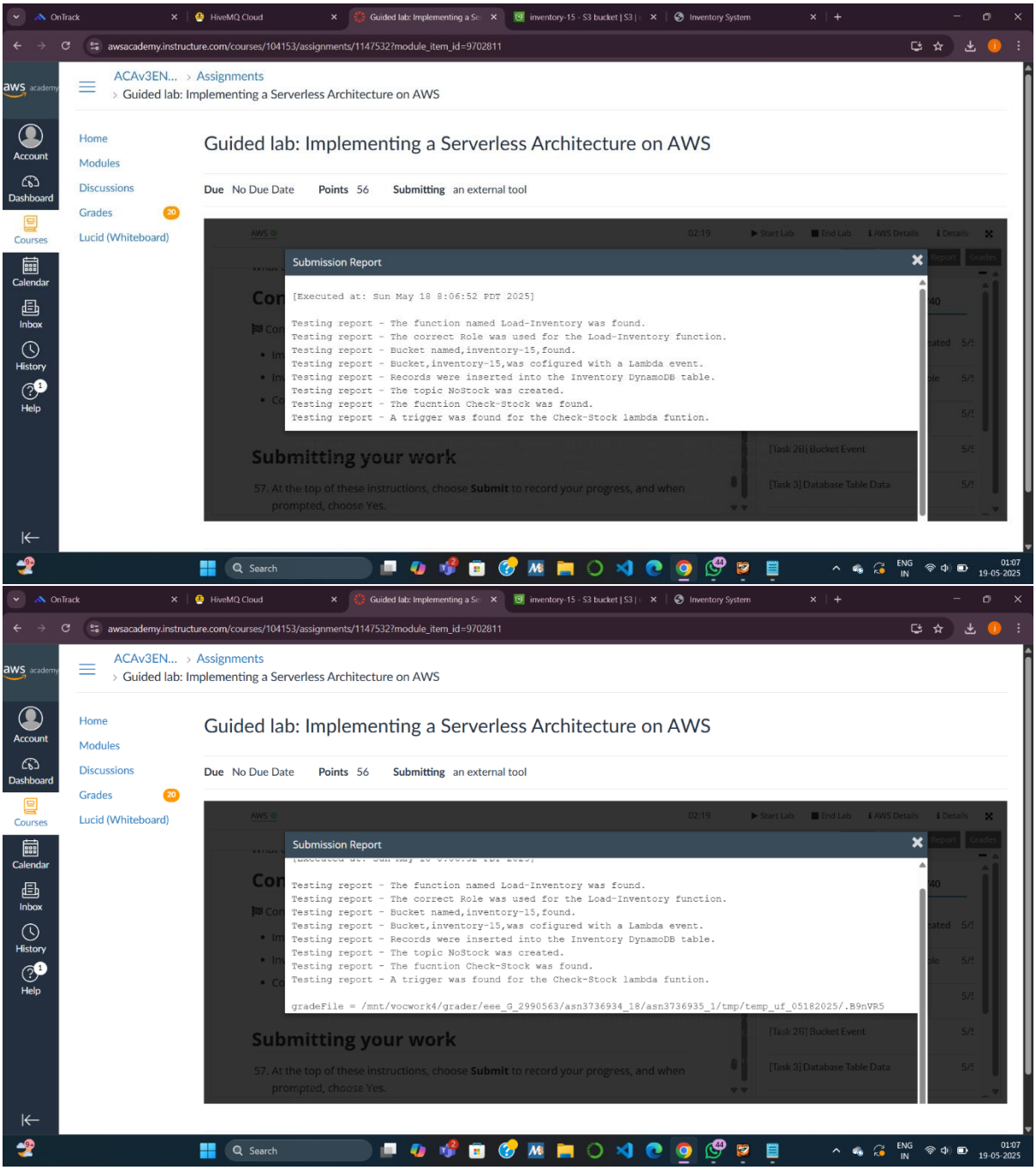

OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
inventory-15 - S3 bucket | S3 |
Inventory System
us-east-1.console.aws.amazon.com/s3/buckets/inventory-15?region=us-east-1&tab=objects&bucketType=general
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3904332=s224001588@deakin.edu.au @ 8913-7700-1183
Amazon S3 > Buckets > inventory-15
Upload succeeded
For more information, see the Files and folders table.
inventory-15 Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Objects (2)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
inventory-berlin.csv
csv
May 19, 2025, 00:39:17 (UTC+10:00)
140.0 B
Standard
inventory-calcutta.csv
csv
May 19, 2025, 01:03:53 (UTC+10:00)
150.0 B
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
01:04
19-05-2025


OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
Upload objects - S3 bucket inve
Inventory System
us-east-1.console.aws.amazon.com/s3/upload/inventory-15?region=us-east-1&bucketType=general
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3904332=s224001588@deakin.edu.au @ 8913-7700-1183
Amazon S3 > Buckets > inventory-15 > 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 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 (3 total, 430.0 B)
All files and folders in this table will be uploaded.
Remove
Add files
Add folder
Find by name
Name
Folder
Type
Size
inventory-shanghai.csv - text/csv 152.0 B
inventory-pusan.csv - text/csv 133.0 B
inventory-karachi.csv - text/csv 145.0 B
Destination Info
Destination
s3://inventory-15
Destination details
Bucket settings that impact new objects stored in the specified destination.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
01:04
19-05-2025

OnTrack
HiveMQ Cloud
Guided lab: Implementing a Se
Upload objects - S3 bucket inve
Inventory System
us-east-1.console.aws.amazon.com/s3/upload/inventory-15?region=us-east-1&bucketType=general
Search
[Alt+S]
United States (N. Virginia)
voclabs/user3904332=s224001588@deakin.edu.au @ 8913-7700-1183
Upload succeeded
For more information, see the Files and folders table.
Summary
Destination
s3://inventory-15
Succeeded
3 files, 430.0 B (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (3 total, 430.0 B)
Find by name
Name
Folder
Type
Size
Status
Error
inventory-shanghai.csv - text/csv 152.0 B Succeeded -
inventory-pusan.csv - text/csv 133.0 B Succeeded -
inventory-karachi.csv - text/csv 145.0 B Succeeded -
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
01:05
19-05-2025


us-east-1.console.aws.amazon.com/s3/buckets/inventory-15?region=us-east-1&tab=objects&bucketType=general
Amazon S3 > Buckets > inventory-15
Upload succeeded
For more information, see the Files and folders table.
inventory-15 Info
Objects Metadata Properties Permissions Metrics Management Access Points
Objects (5)
Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name Type Last modified Size Storage class
inventory-berlin.csv csv May 19, 2025, 00:39:17 (UTC+10:00) 140.0 B Standard
inventory-calcutta.csv csv May 19, 2025, 01:03:53 (UTC+10:00) 150.0 B Standard
inventory-karachi.csv csv May 19, 2025, 01:05:07 (UTC+10:00) 145.0 B Standard
inventory-pusan.csv csv May 19, 2025, 01:05:07 (UTC+10:00) 133.0 B Standard
inventory-shanghai.csv csv May 19, 2025, 01:05:06 (UTC+10:00) 152.0 B Standard
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

awsacademy.instructure.com/courses/104153/assignments/1147532?module_item_id=9702811
ACAv3EN... > Assignments
Guided lab: Implementing a Serverless Architecture on AWS
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Guided lab: Implementing a Serverless Architecture on AWS
Due No Due Date Points 56 Submitting an external tool
02:19 Start Lab End Lab AWS Details Details
Submit Submission Report Grades
Conclusion
Congratulations! You have successfully done the following:
Implemented a serverless architecture on AWS
Invoked Lambda functions from Amazon S3 and Amazon DynamoDB
Configured Amazon SNS to send notifications
Submitting your work
57. At the top of these instructions, choose Submit to record your progress, and when prompted, choose Yes.
Total score 40/40
[Task 1A] Load-Inventory Function Created
[Task 1B] Load-Inventory Execution Role
[Task 2A] Bucket Created
[Task 2B] Bucket Event
[Task 3] Database Table Data



Answering questions about the S3 bucket

1. Was an S3 bucket created, even if you did not specify a name for the bucket? If so, what name was it given?

Yes, an S3 bucket was created **even without explicitly specifying a name** in the CloudFormation template.

- When no name is provided, AWS automatically generates a **globally unique name** for the bucket in the format like:
  update-cafe-app-bucket-1a2b3c4d5e6f
- This name includes the stack name and a unique string to ensure it doesn't conflict with existing bucket names globally.

2. What Region was the bucket created in, and why was it created in this Region?

The bucket was created in the **same Region where the CloudFormation stack was launched**.

- For example: us-east-1 (N. Virginia), ap-southeast-2 (Sydney), etc.
- It is created **in that Region by default** to ensure **low latency and compliance** with resource dependencies and regional policies.

3. To define an S3 bucket, how many lines of code did you need to enter in the Resources: section of the template file?

Typically, an S3 bucket definition in a YAML template looks like this:

```
Resources:
  CafeS3Bucket:
    Type: AWS::S3::Bucket
```

- This is **3 lines of code** under the Resources section.
- If you include additional properties (like versioning or lifecycle rules), more lines would be required, but a basic bucket needs just **3 lines**.

Answering questions about the results of creating an application layer

4. Go to the *Parameters* tab of the update-cafe-app stack. What value do you see for the LatestAmiId?

ami-0abcdef1234567890

- This AMI ID corresponds to the **latest Amazon Linux 2** or another base image being used for the EC2 instance in the application layer.
- This ID is **dynamically referenced** using an SSM parameter like:

```
LatestAmiId:
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
```

5. Go to the *Stack info* tab of the update-cafe-app stack. What is the Amazon Resource Name (ARN) of the IAM role that grants the permissions to create and update the stack?

we will see an ARN similar to:

arn:aws:iam::123456789012:role/cafe-app-CloudFormationExecutionRole-ABC123XYZ

- This **IAM role** allows CloudFormation to assume necessary permissions (like creating EC2 instances, S3 buckets, etc.).
- It's specified in the stack's execution role during creation.

6. In the AWS Management Console, navigate to the CodeCommit repository where your AWS CloudFormation templates are stored. Choose *Commits*, and in the Commits list, open one of the commits by choosing its commit ID. What do you observe?

Upon opening a commit, you will see:

- **Commit metadata**:
  - Commit ID
  - Commit message (e.g., "Updated EC2 instance type to t3.micro")
  - Author and timestamp
- **Changed files**:
  - You can view which files were modified, such as template.yaml.
  - A **side-by-side diff** shows the exact line changes (additions, deletions).

Observation: This helps track template version history, document changes, and collaborate with others.