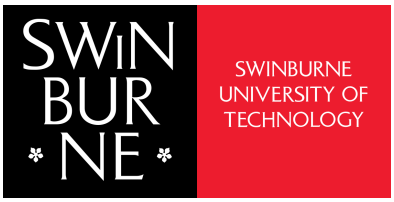
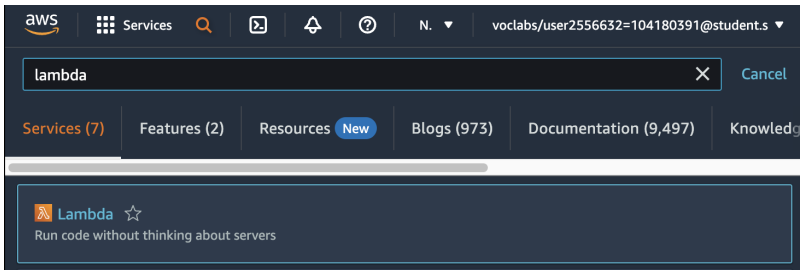
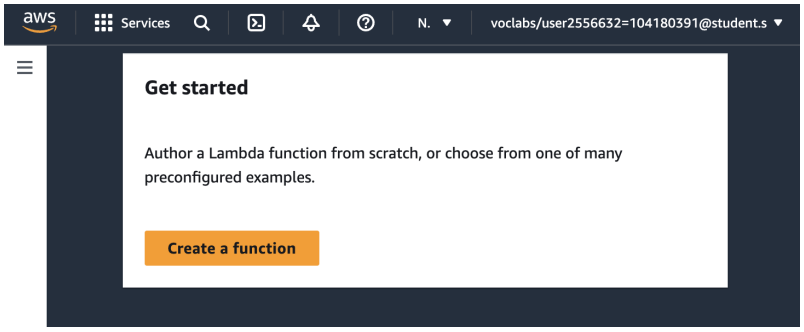
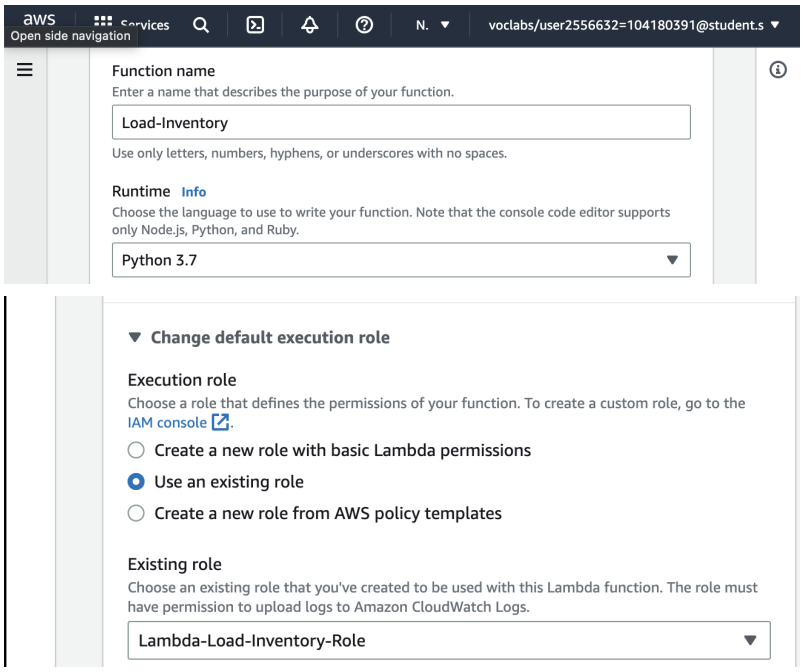
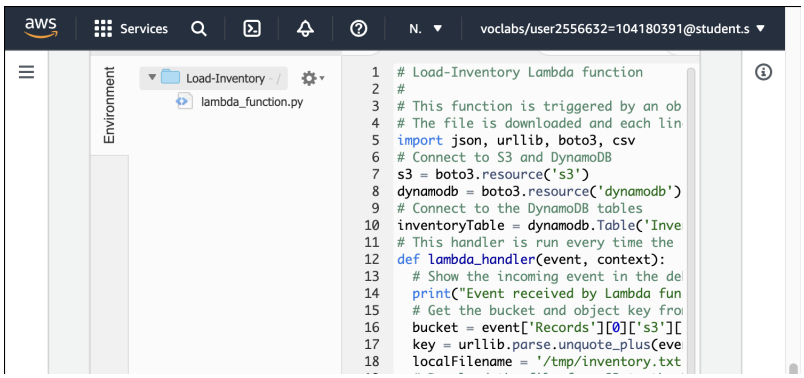


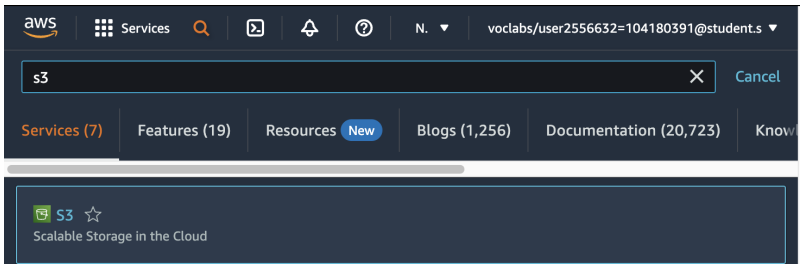
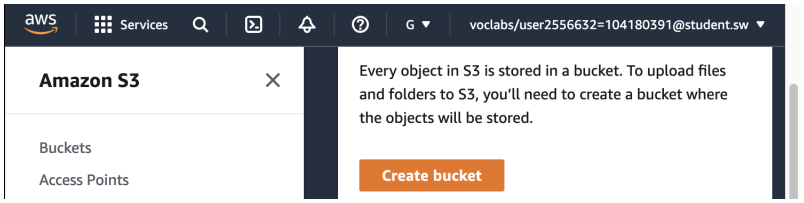
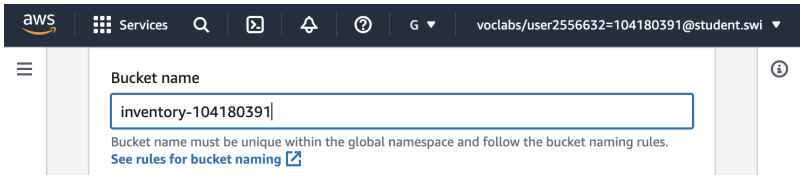
Module 13 Guided Lab - Implementing a Serverless Architecture with AWS Lambda

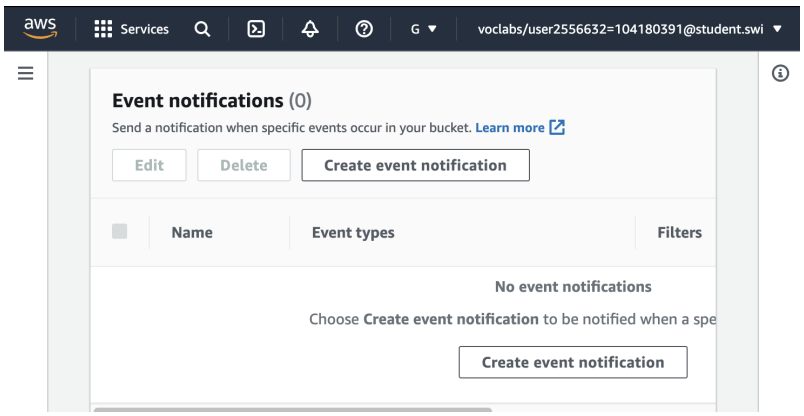
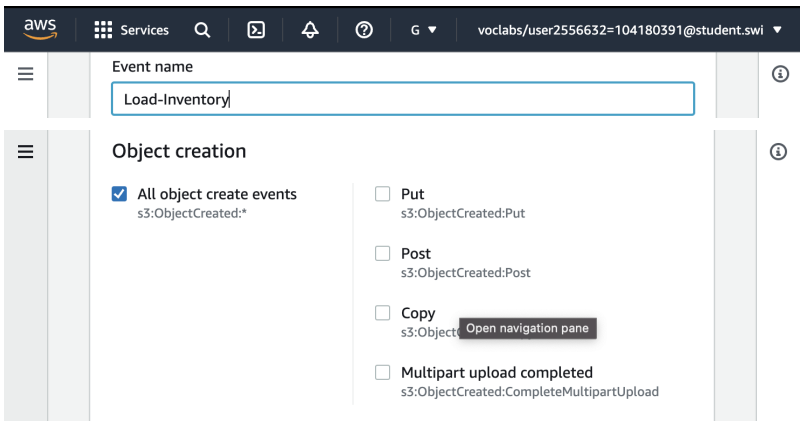
July 6, 2023

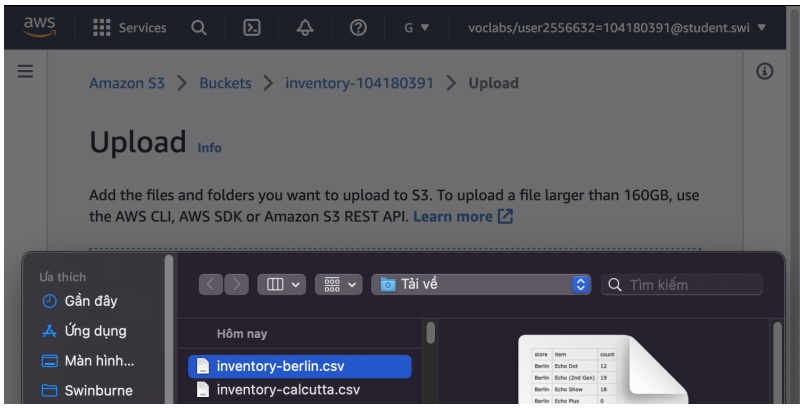
Luu Tuan Hoang
Student ID: 104180391

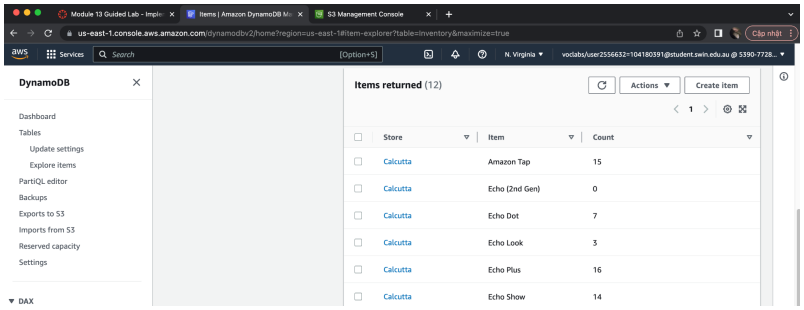
Task 1: Creating a Lambda function to load data		
Step	**Description**	**Screenshot**
1	In the AWS Management Console, on the **Services** menu, choose **Lambda**.	
2	Choose **Create function**	
3	Use the Author from scratch option, configure the following settings: - **Function name**: Load-Inventory - **Runtime**: Python 3.7 Expand Choose or create an execution role. - **Execution role**: Use an existing role - **Existing role**: Lambda-Load-Inventory-Role Choose **Create function**	

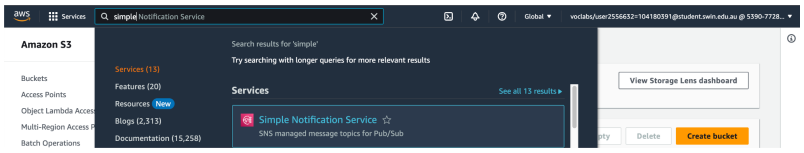
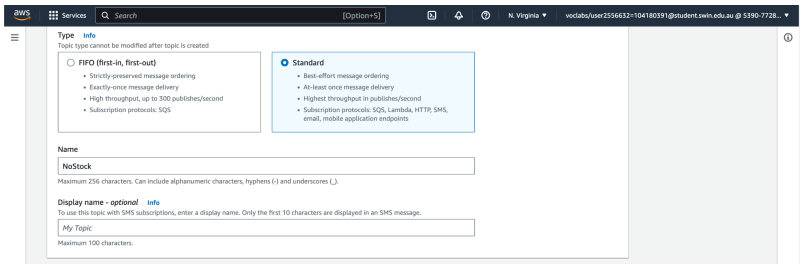
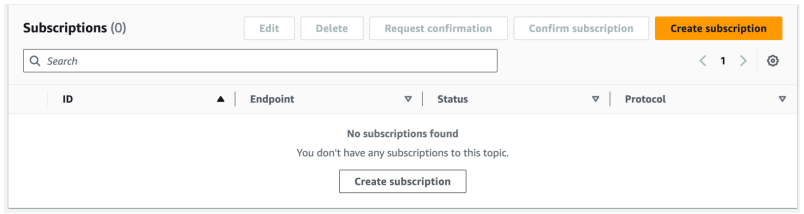
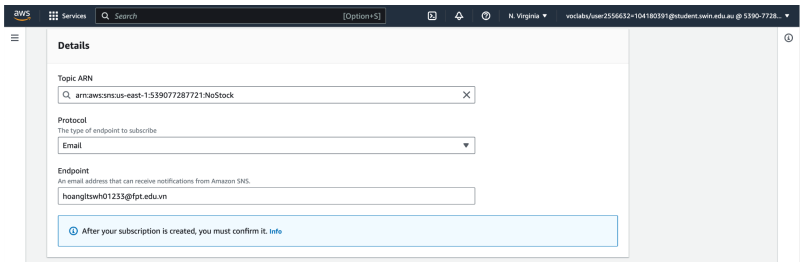
4	In the **Code source** editor, copy and paste the code. Choose **Deploy** to save your changes.	

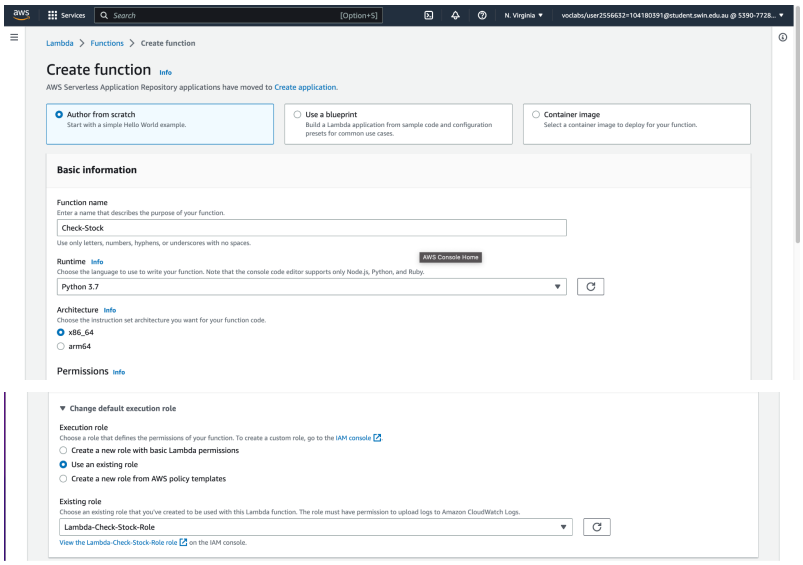
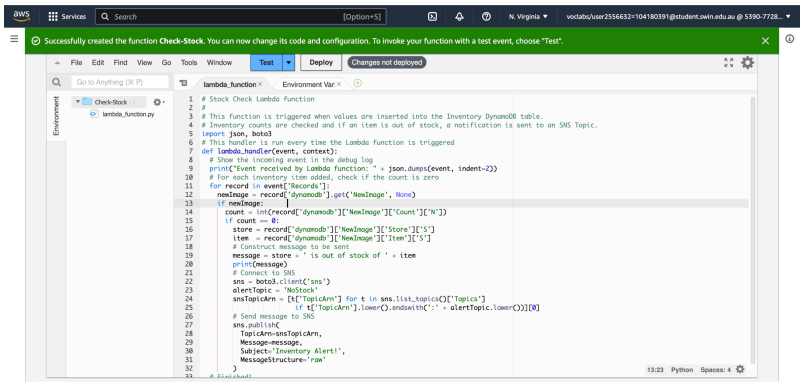
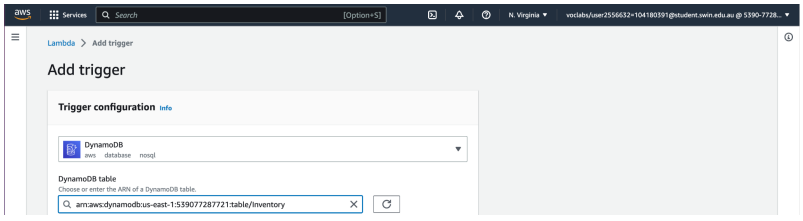
Task 2: Configuring an Amazon S3 event		
Step	**Description**	**Screenshot**
1	In the AWS Management Console, on the **Services** menu, choose **S3**.	
2	Choose **Create bucket**	
3	For Bucket name enter: **inventory-<number>** (Replace with a random number) Choose **Create bucket**	

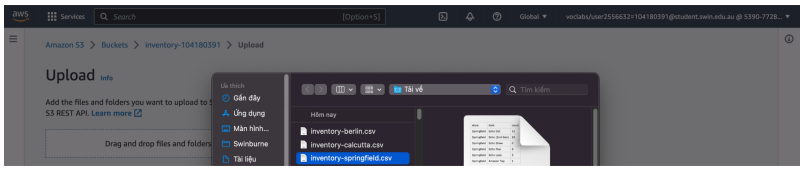
4	Choose the name of your **inventory-bucket**. Choose the **Properties** tab. Scroll down to **Event notifications**.	
5	Click **Create event notification** then configure these settings: - **Name**: Load-Inventory - **Event types**: All object create events - **Destination**: Lambda Function - **Lambda function**: Load-Inventory - Choose **Save changes**	

Task 3: Testing the loading process		
Step	**Description**	**Screenshot**
1	Download the inventory files In the console, return to your S3 bucket by choosing the **Objects** tab. Choose **Upload** Choose **Add files**, and select one of the inventory CSV files. (You can choose any inventory file.) Choose **Upload**	

2	Since I cannot open the dashboard, this is the data from the inventory file will be displayed. It shows the store, item and inventory count in DynamoDB.	

Task 4: Configuring notifications		
Step	**Description**	**Screenshot**
1	On the **Services** menu, choose **Simple Notification Service**.	
2	In the **Create topic** box, for **Topic name**, enter: **NoStock**. Keep **Standard** selected. Choose **Create topic**	
3	In the lower half of the page, choose **Create subscription**	
4	Configure these settings: - **Protocol**: Email - **Endpoint**: Enter your email address - Choose **Create subscription**	

Task 5: Creating a Lambda function to send notifications		
Step	**Description**	**Screenshot**
1	On the **Services** menu, choose **Lambda**. Choose **Create function** and configure these settings: - **Function name**: Check-Stock - **Runtime**: Python 3.7 Expand **Choose or create an execution role**. - **Execution role**: Use an existing role - **Existing role**: Lambda-Check-Stock-Role Choose **Create function**	
2	Copy the code to the **Code Source editor** Choose **Deploy** to save your code changes	
3	Scroll to the **Designer** section (which is at the top of the page). Choose **Add trigger** and then configure these settings: - **Select a trigger**: DynamoDB - **DynamoDB Table**: Inventory - Choose **Add**	

Task 5: Creating a Lambda function to send notifications		
Step	**Description**	**Screenshot**
1	On the **Services** menu, choose **S3**. Choose the name of your **inventory-bucket**. Choose Upload and upload a different **inventory** file	
2	Lab completed with all tasks finished.