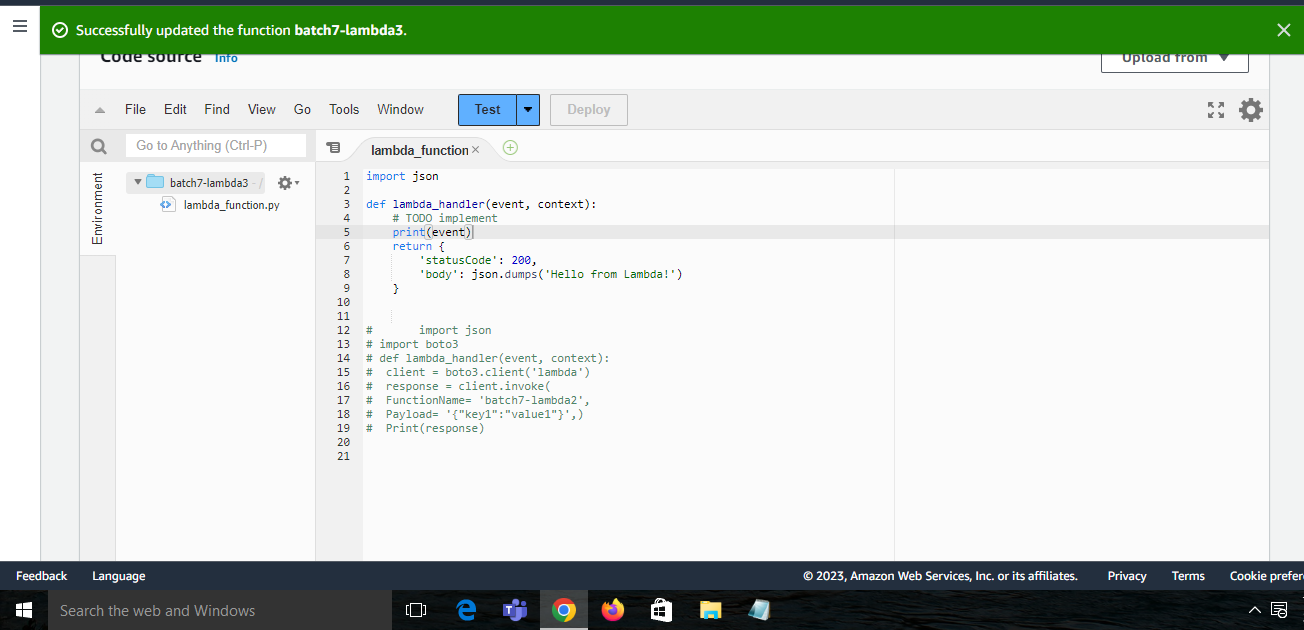
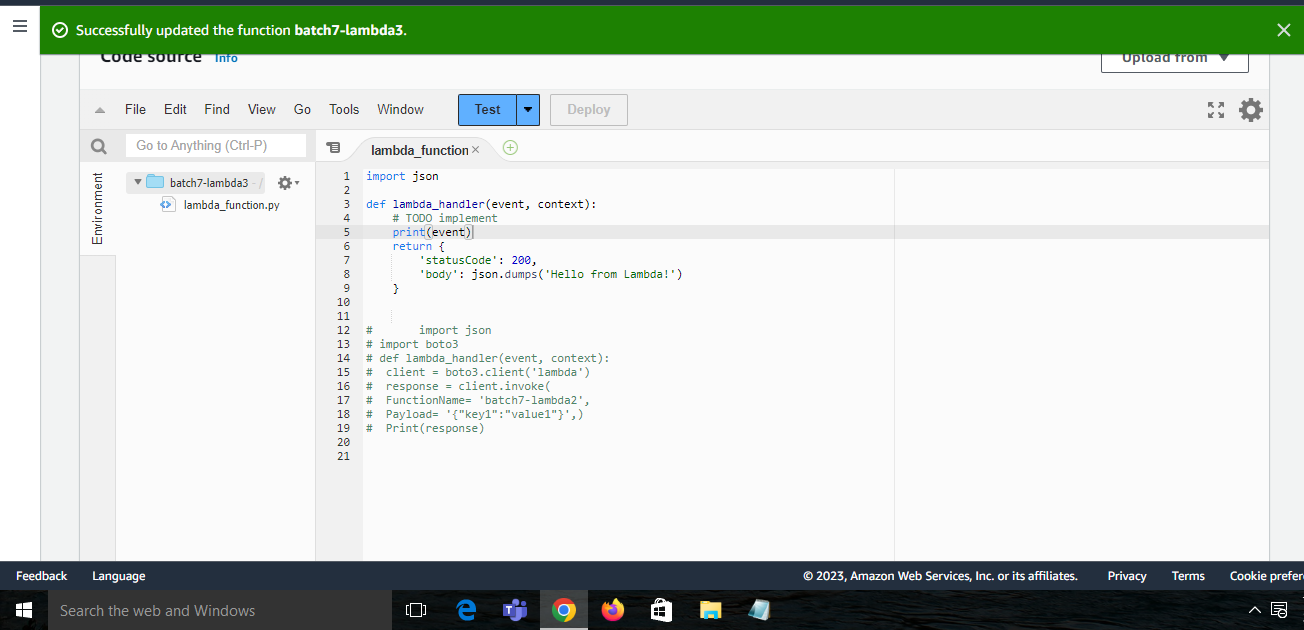
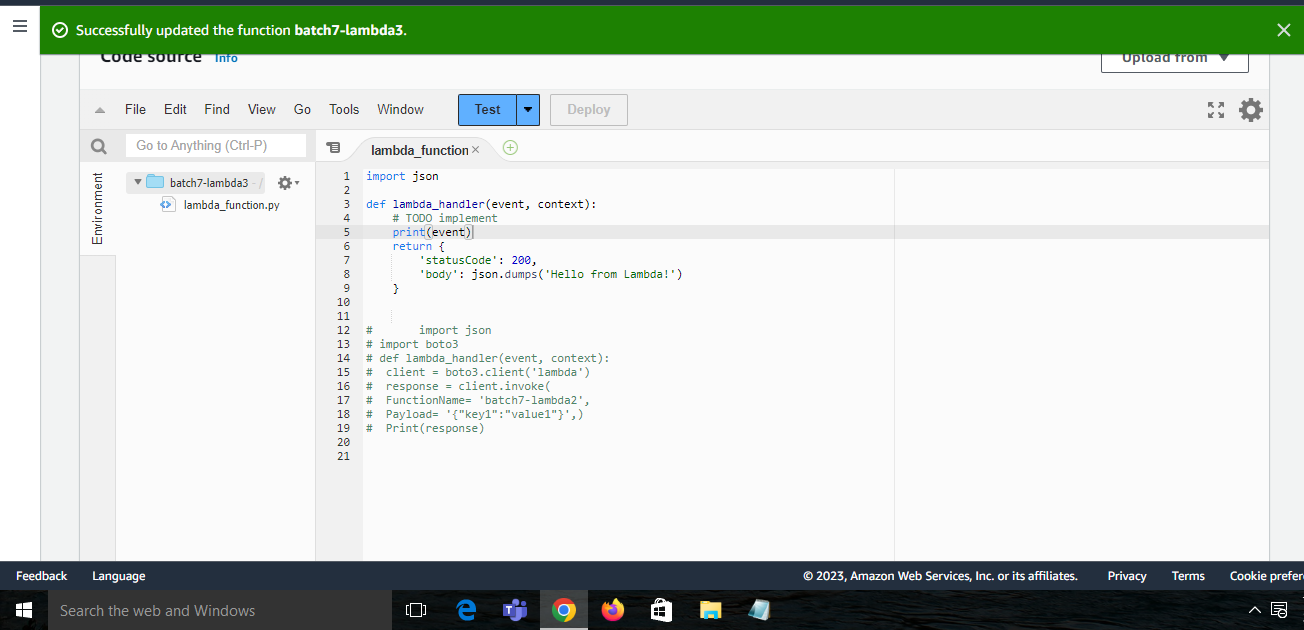
***Lamda Function Creation***

1. Log in to the AWS Management Console and navigate to the Lambda service.
2. Click on the "Create function" button.
3. Choose the "Author from scratch" option and give your function a name.
4. Select the runtime environment you want to use for your function. This will depend on the type of documentation you want to work with.
5. Under "Permissions", choose the execution role for your function. You can either create a new role or use an existing one.
6. Click on the "Create function" button to create your function.
7. On the function configuration page, you can set environment variables, add layers, and configure other settings for your function.
8. Under the "Function code" section, you can write the code for your Lambda function
9. Once you've finished writing your function code, click on the "Deploy" button to deploy your function.
10. You can now test your function by clicking on the "Test" button and providing sample input data.

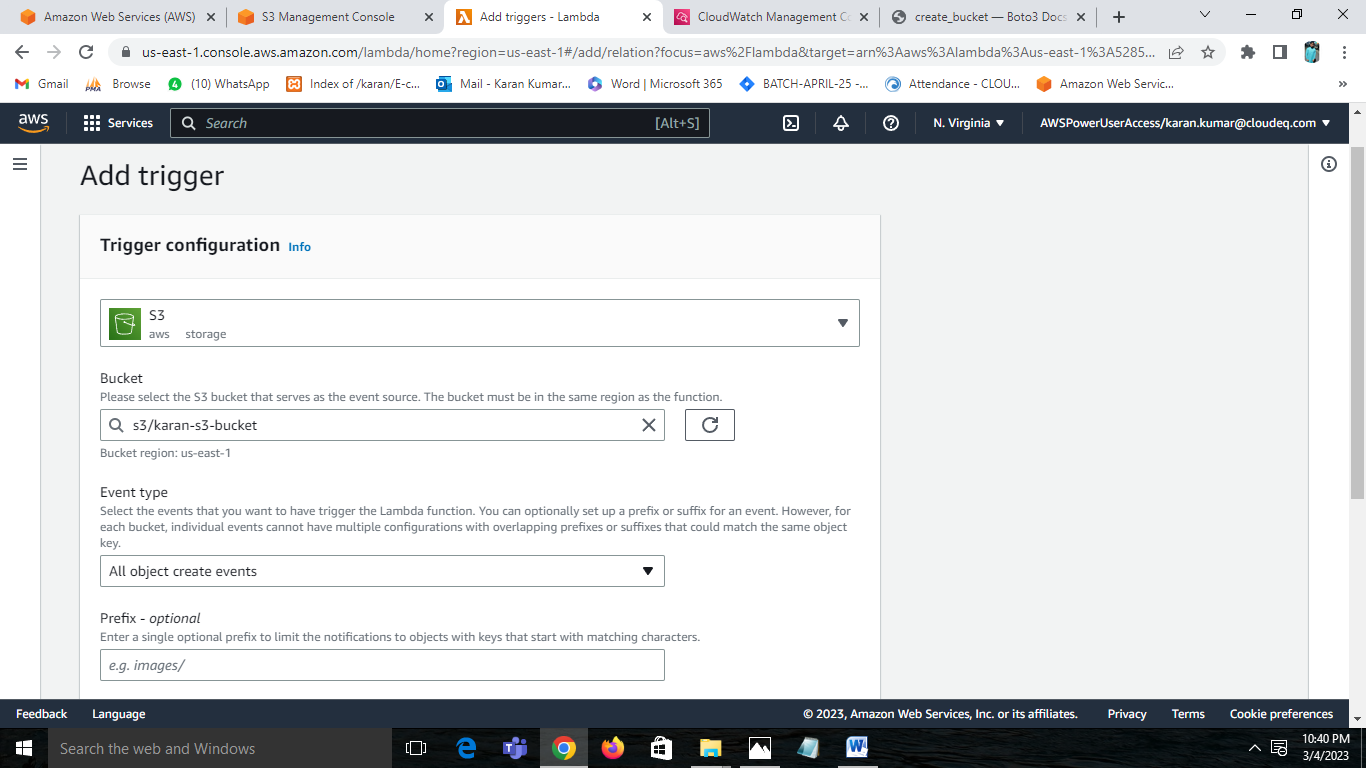


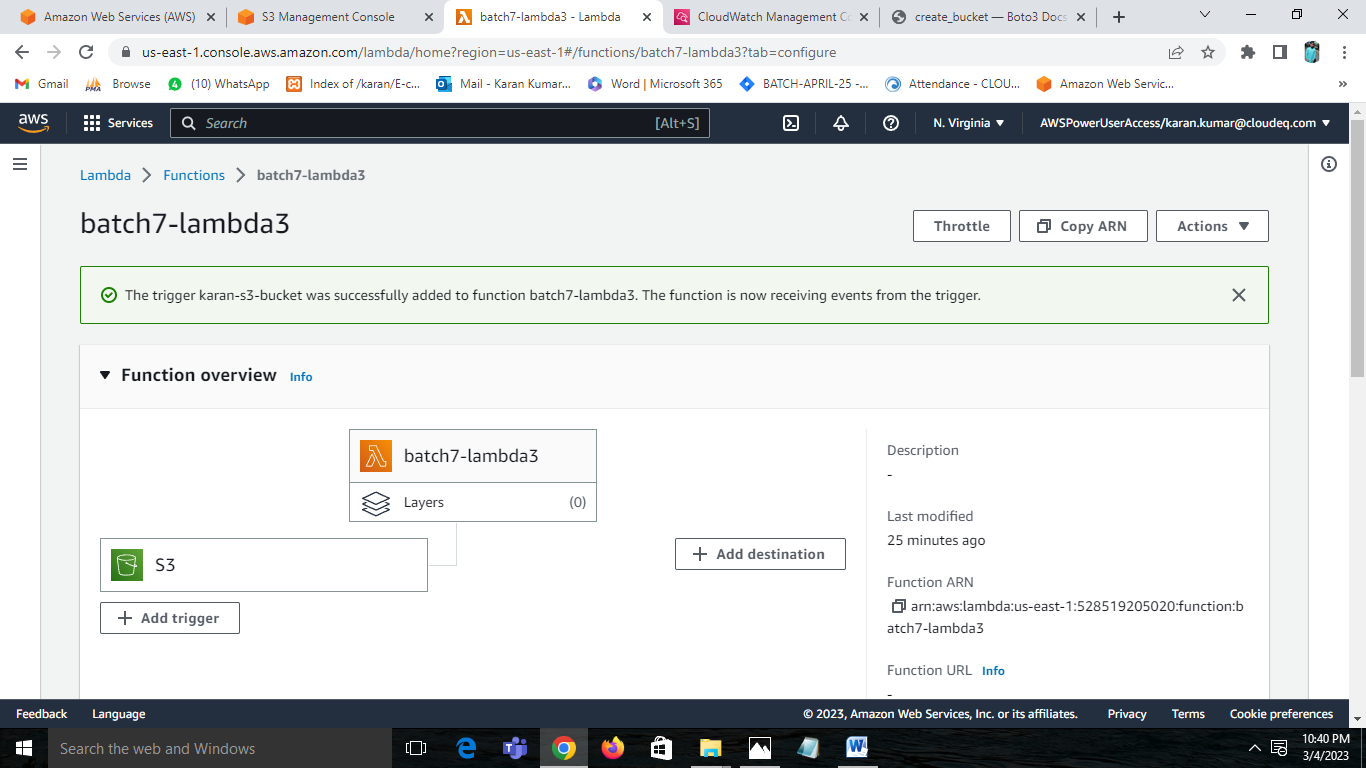


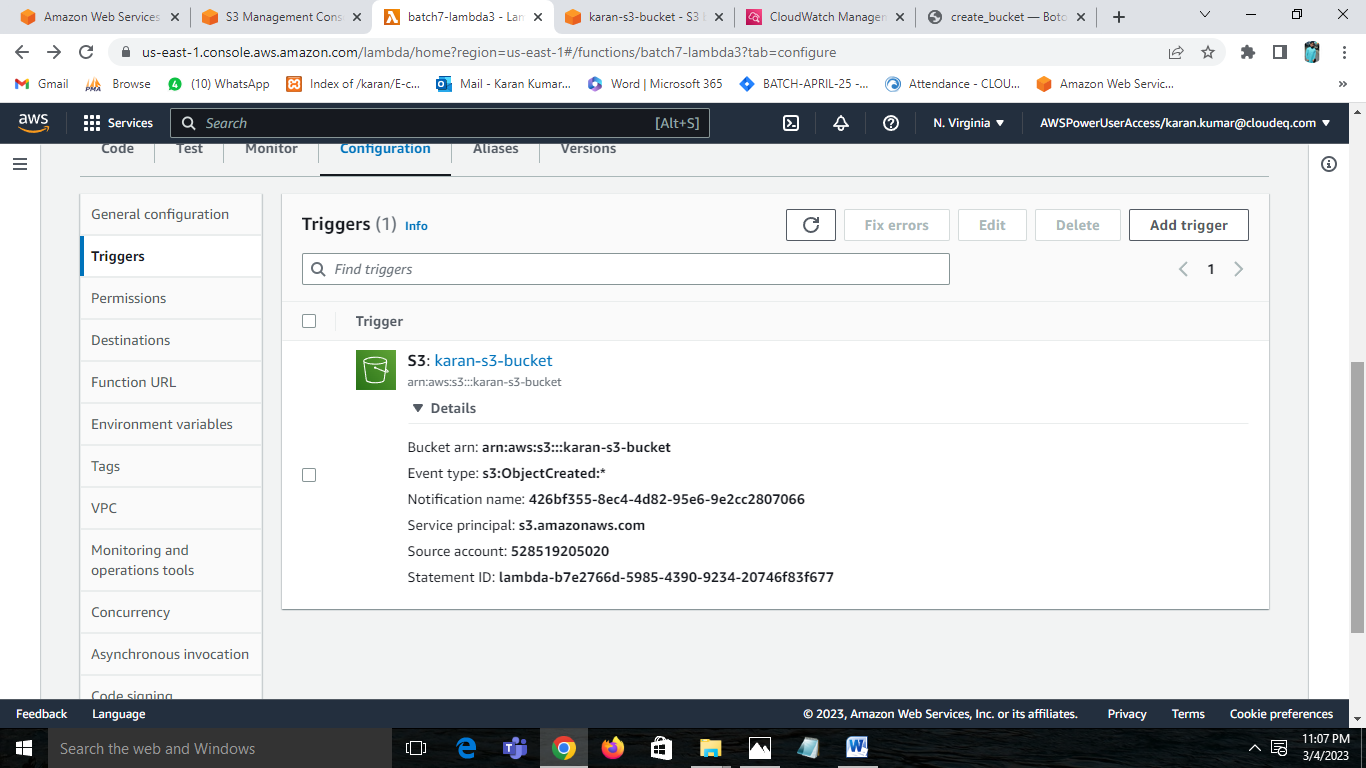


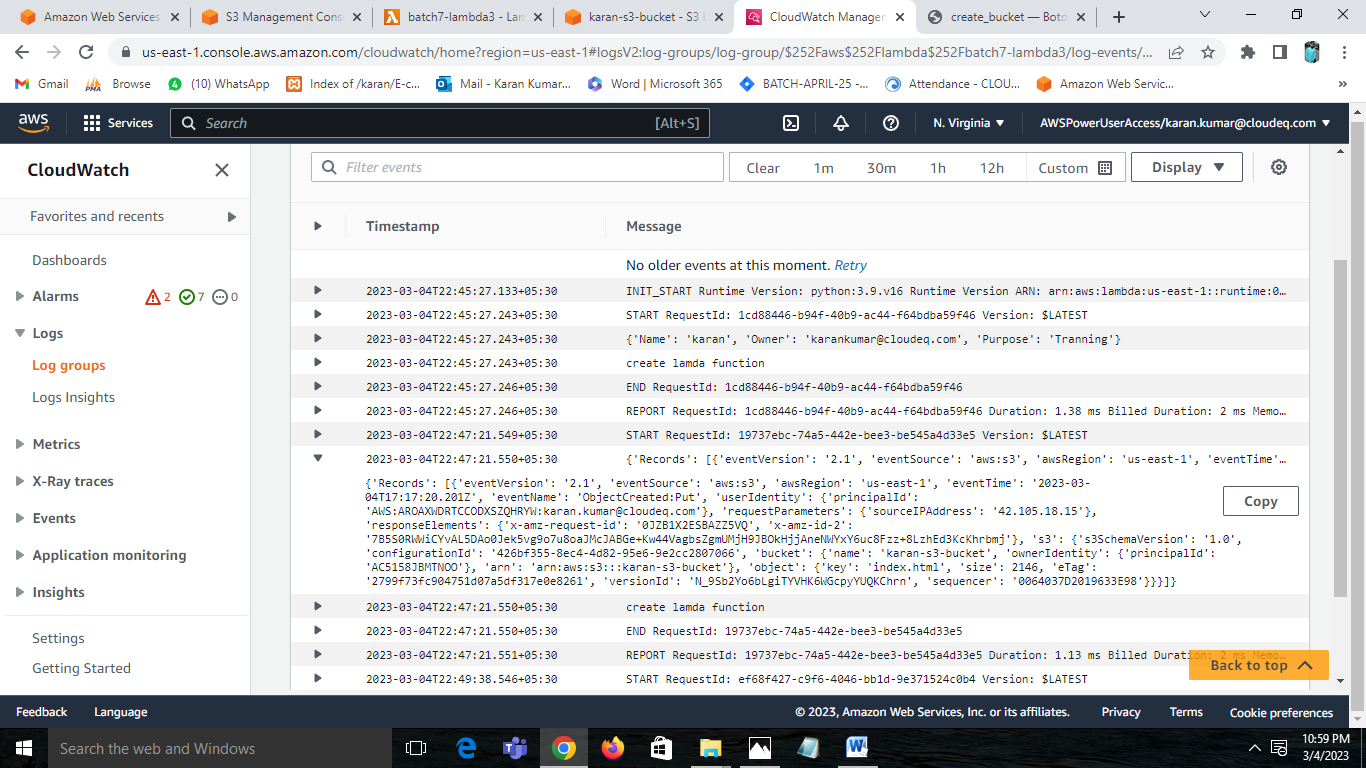
***Add Trigger:***

1. Create an S3 bucket where the trigger event will occur.
2. Create or select an existing Lambda function in your AWS account.
3. Add the S3 trigger to the Lambda function by selecting the S3 bucket and event type.
4. Configure permissions for the function to access the S3 bucket.
5. Test the trigger by uploading a file to the S3 bucket.
6. Monitor the trigger for any issues and troubleshoot as needed.









***create a lambda function which will invoke another lambda function:***

1. Boto3 is the Python SDK for AWS.
2. To invoke a Lambda function using Boto3, you can create a new instance of the lambda client object using boto3.client('lambda').
3. Then, you can use the invoke() method of the lambda client to make an API request to invoke the other Lambda function.
4. The invoke() method takes a number of parameters, including the name of the function to invoke, the input data to pass to the function, and the type of invocation (synchronous or asynchronous).
5. If the invocation is synchronous, the invoke() method will return the response from the function.
6. If the invocation is asynchronous, the invoke() method will return immediately and you will need to check for the function's response later.
7. Note that the invoker Lambda function must have permission to invoke the target function, which can be granted using AWS IAM policies.

