**Spring Boot-AWS Lambda Documentation**

**To test spring cloud functions locally using curl Commands**

**Run Spring Boot application using the commands below**

* **./gradlew clean**
* **./gradlew build**
* **./gradlew bootRun**

1. **getAllOrder Method:** curl -H "Content-Type:application/json" <http://localhost:8080/getAllOrder>
2. **getOrderById :** curl -H "Content-Type:application/json" http://localhost:8080/getOrderById -d "idno"
3. **saveOrder :** curl -i -X POST -H "Content-Type: application/json" -d "{\"id\":\"T101\",\"item\":\"Test\",\"qty\":\"2\",\"price\":\"20\"}" <http://localhost:8080/saveOrder>

1. **deleteOrder :** curl -H "Content-Type:application/json" http://localhost:8080/deleteOrder -d "T101"

# **getAllOrder Method invocation in AWS LAMBDA**

**In Root Directory ->go to below path -->spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\src\main\resources\aws-credentials.properties**

**And uncomment getAllOrder method definition like this**

**spring.cloud.function.definition=getAllOrder**

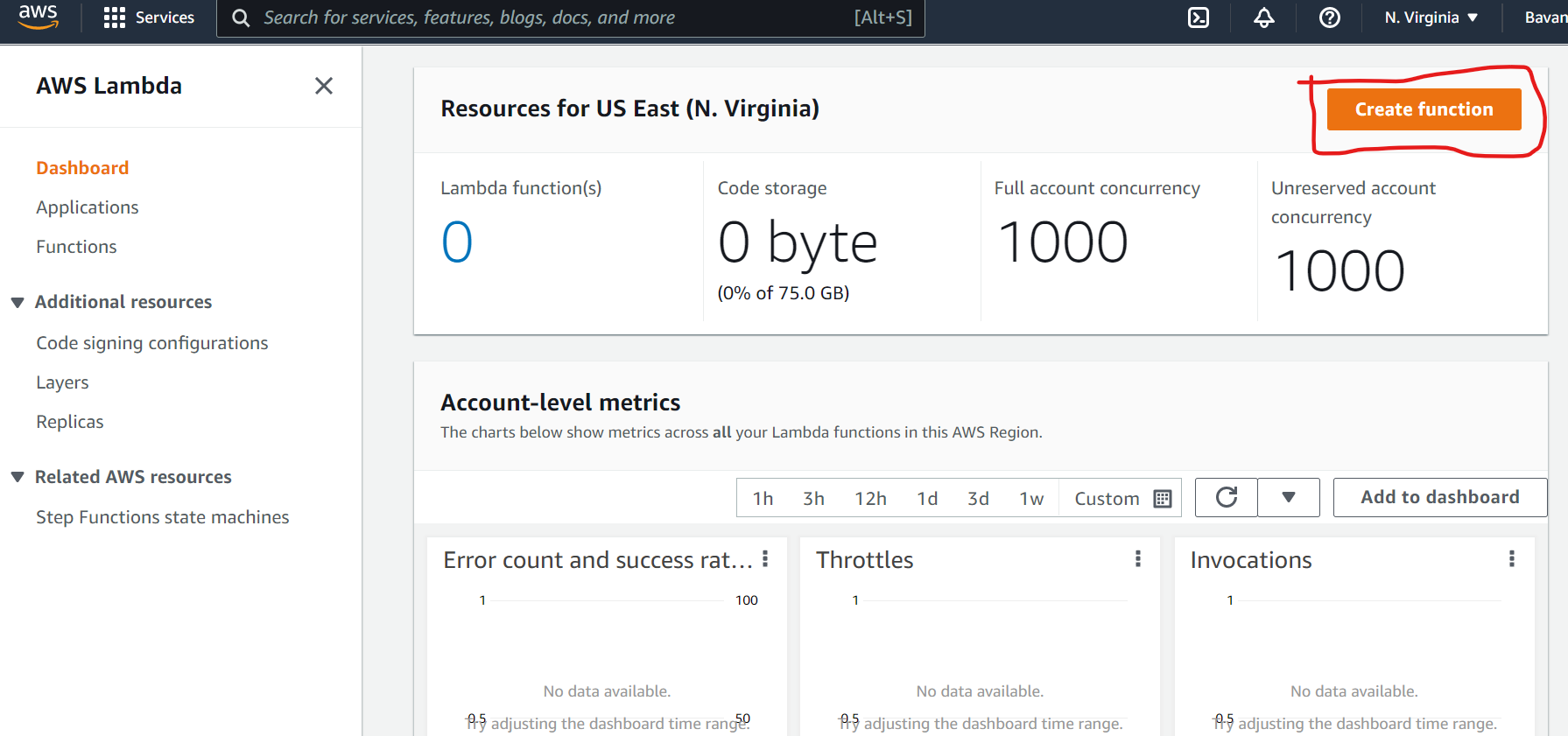
**#****spring.cloud.****function.definition=getOrderById**

**#****spring.cloud.****function.definition=saveOrder**

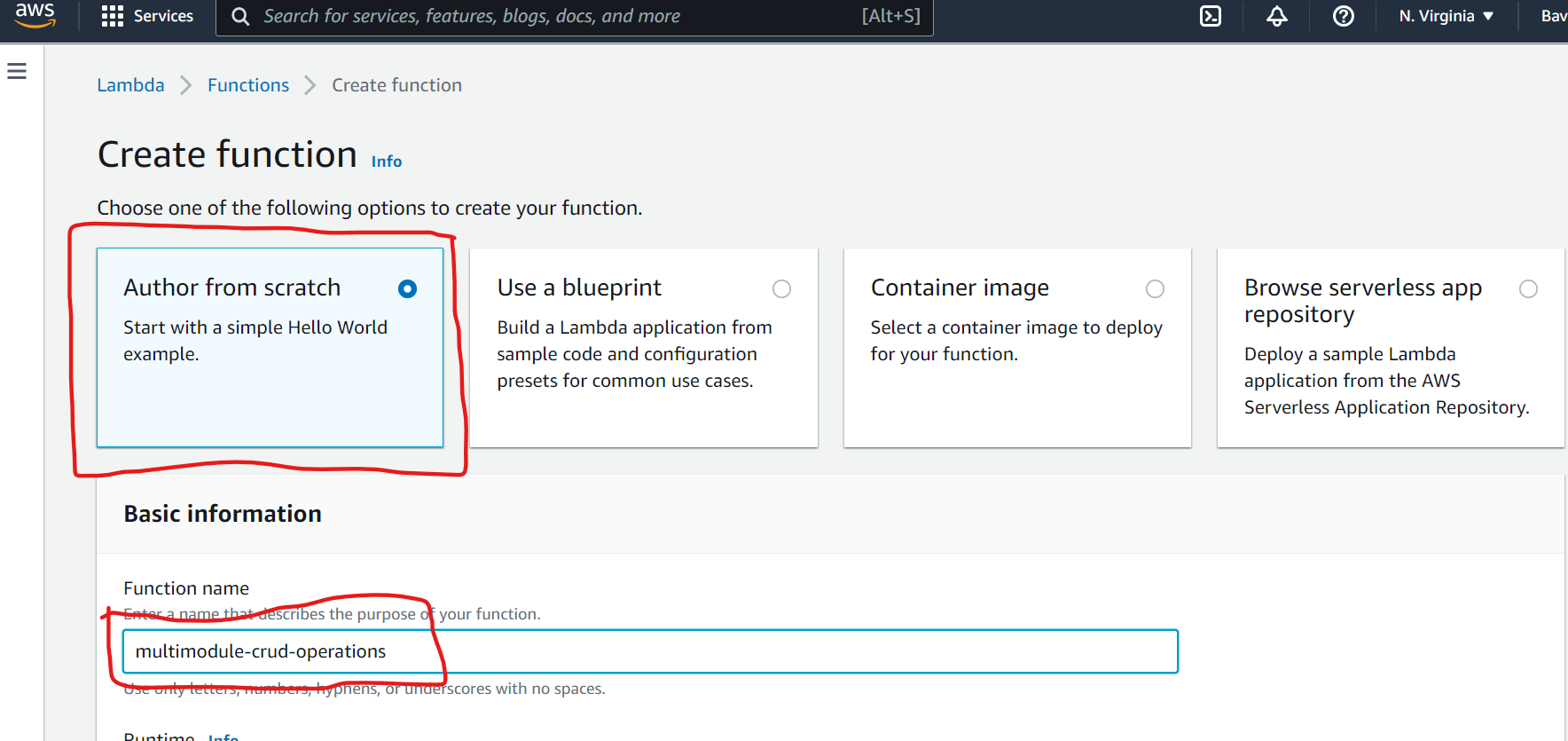
**#****spring.cloud.****function.definition=deleteOrder**

## **Create a Method in Aws Lambda:**

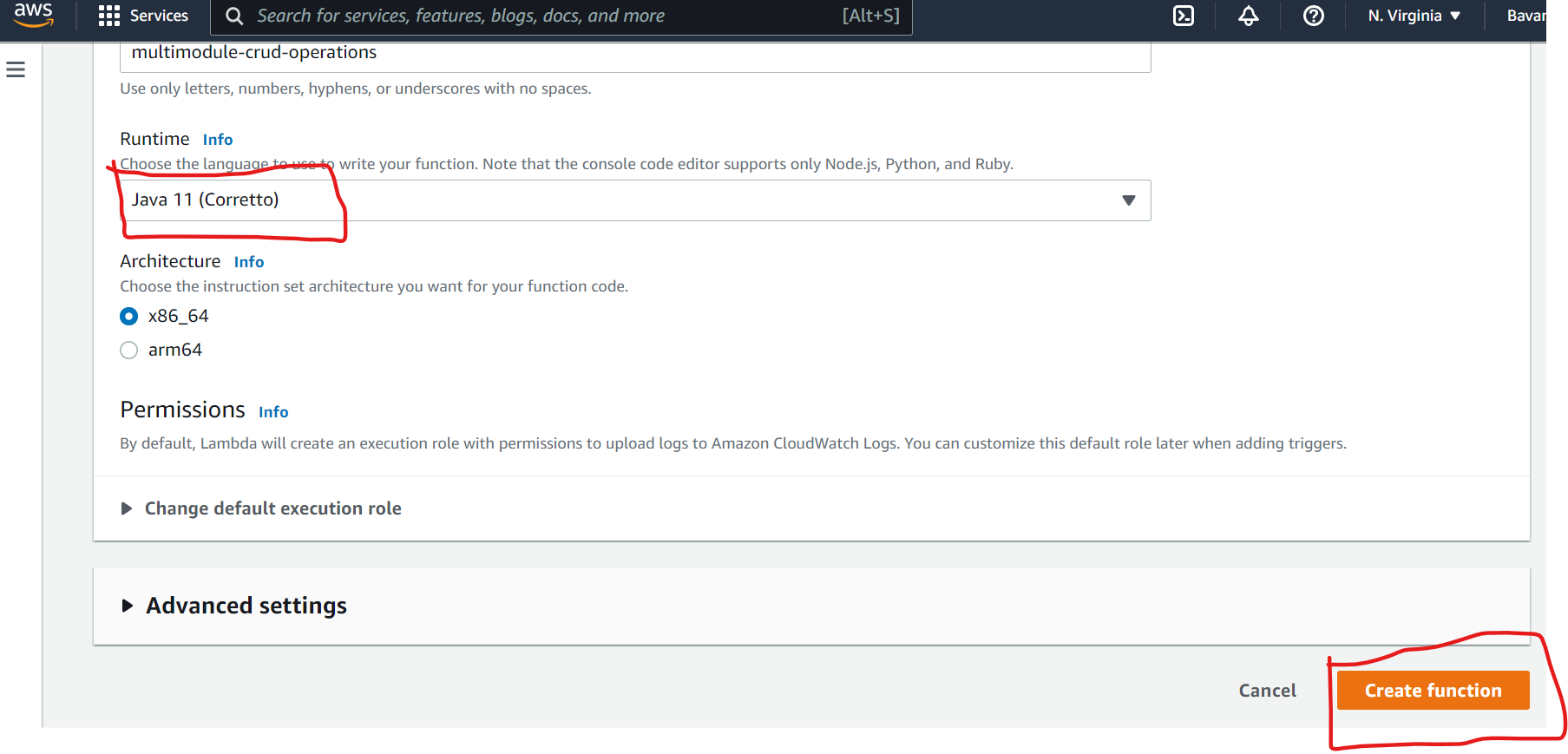
Go to Aws Services->Select Lambda. Then, the below screen appears.



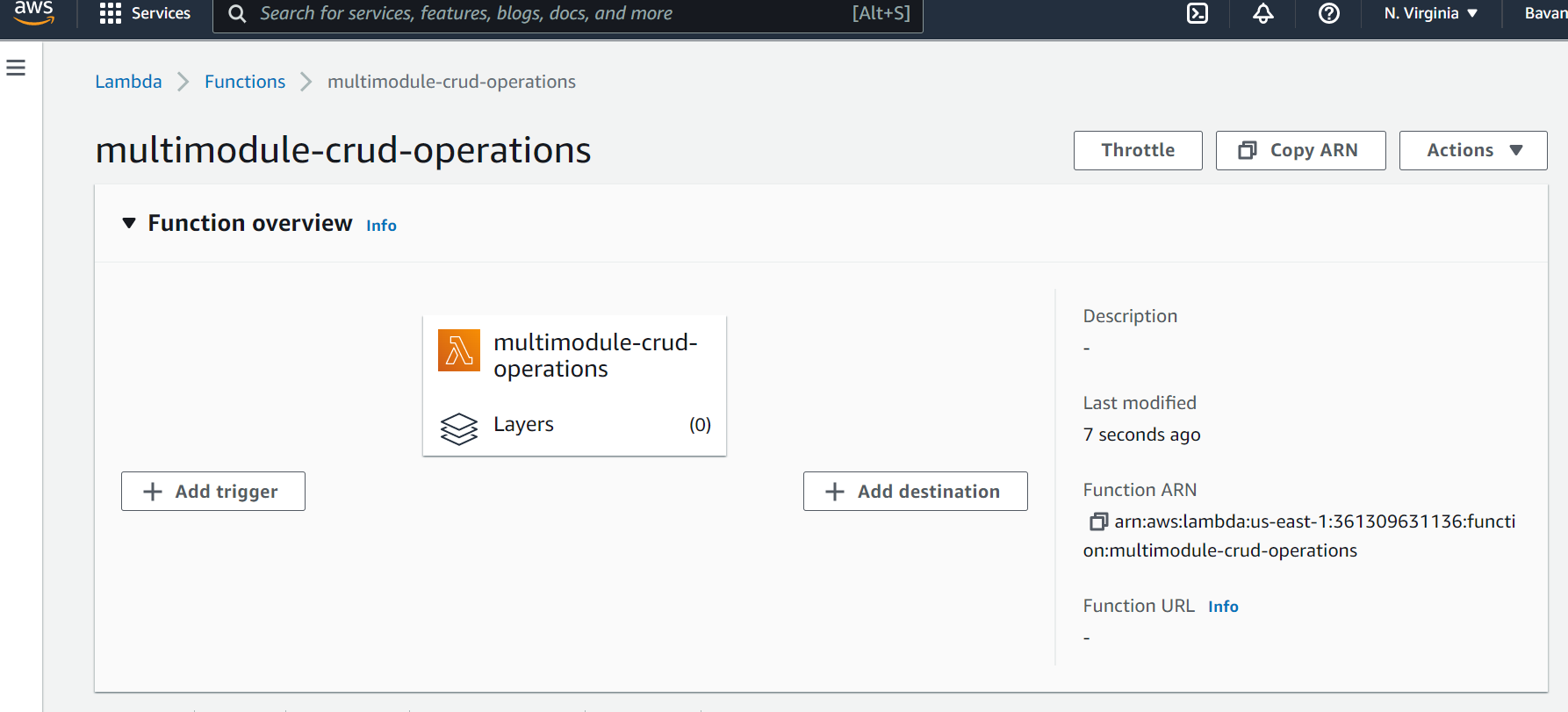
Click on **Create Function**. The Below screen appears.



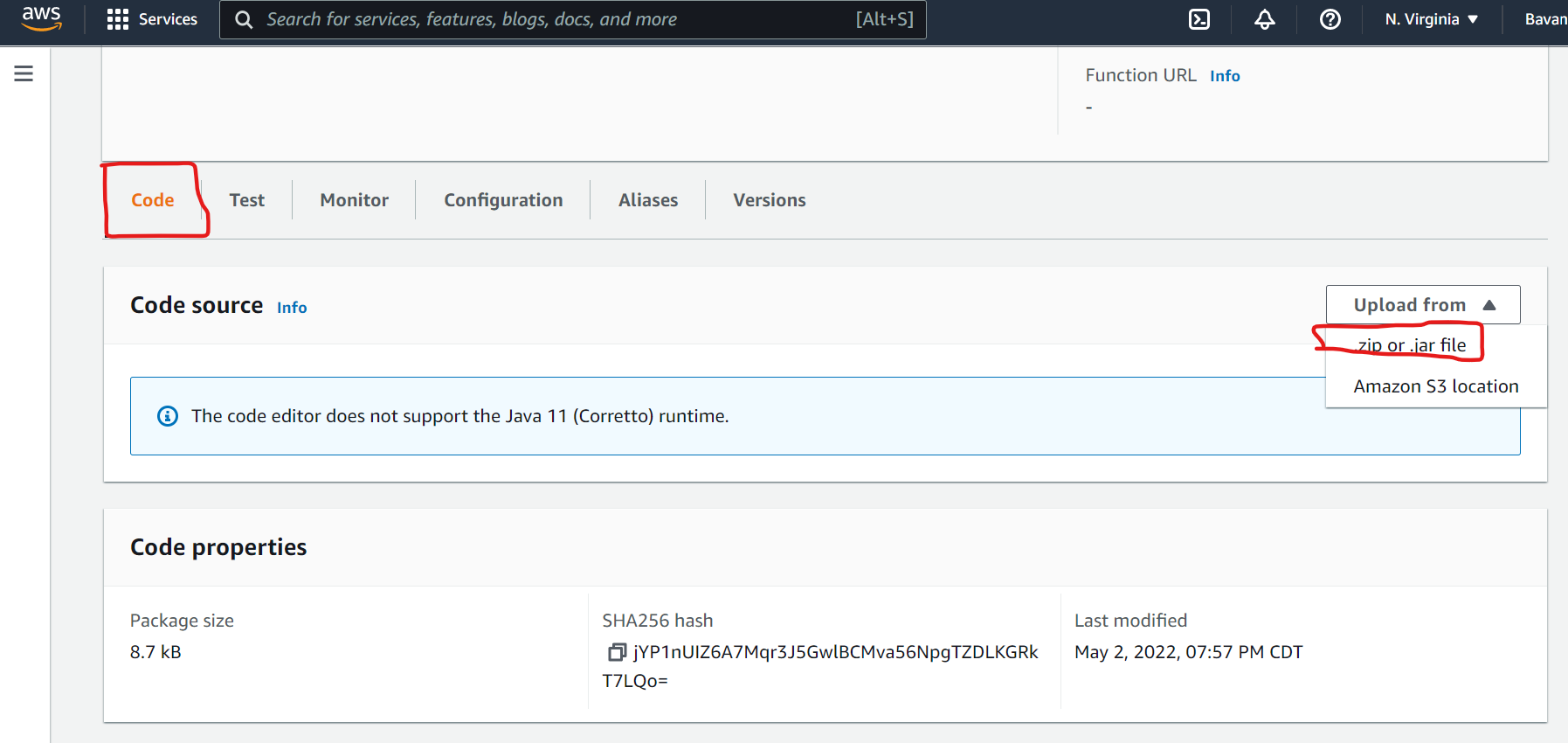
Select Author from Scratch and add Function name as shown in the above screen. Then Scroll down and select Runtime: Java 11(Corretto) . For now, Lambda supports up to Java 11. And Click on **Create function** Button.



Below Screen appears. Scroll Down

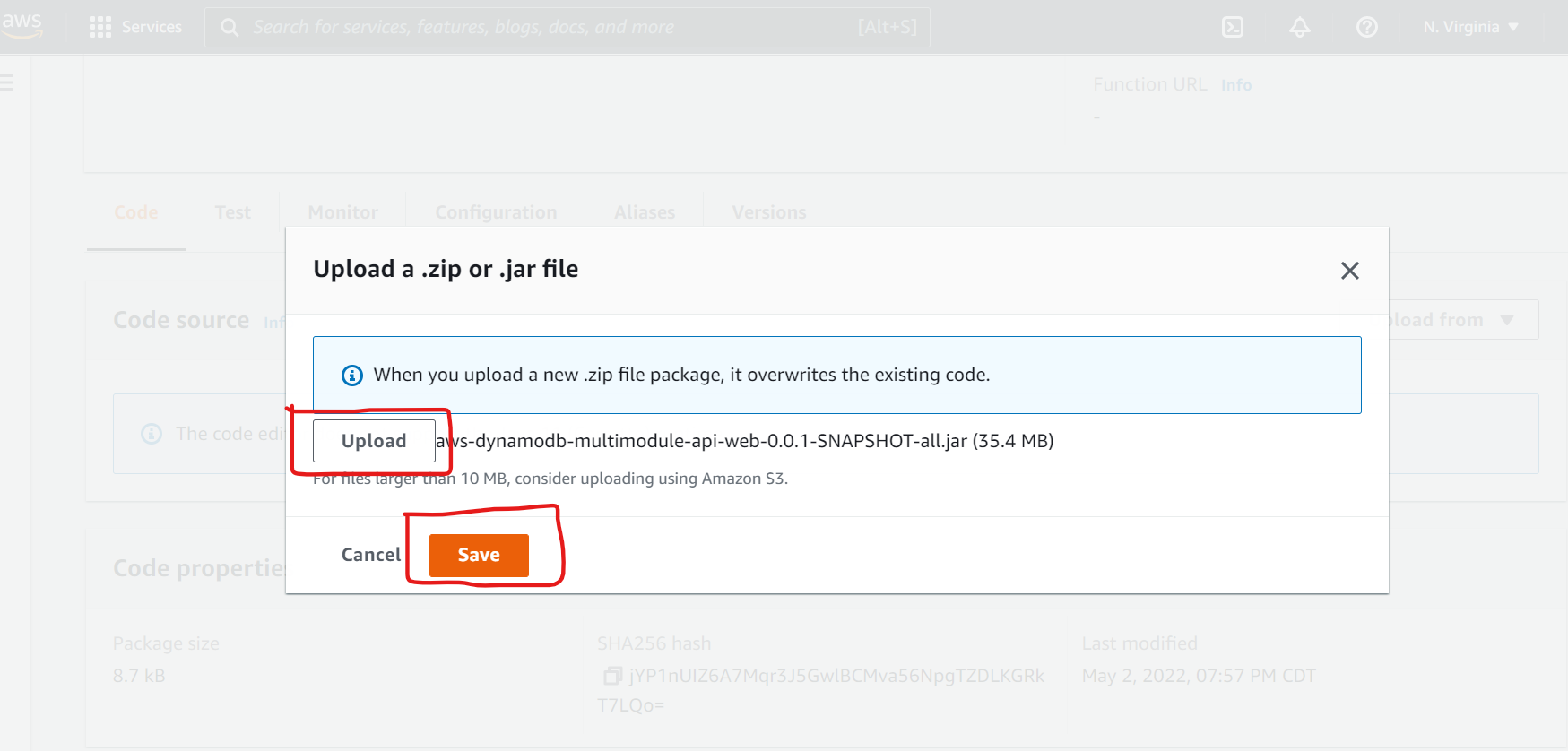


Click on **code** and click on upload from dropdown select **Zip or .jar file**.

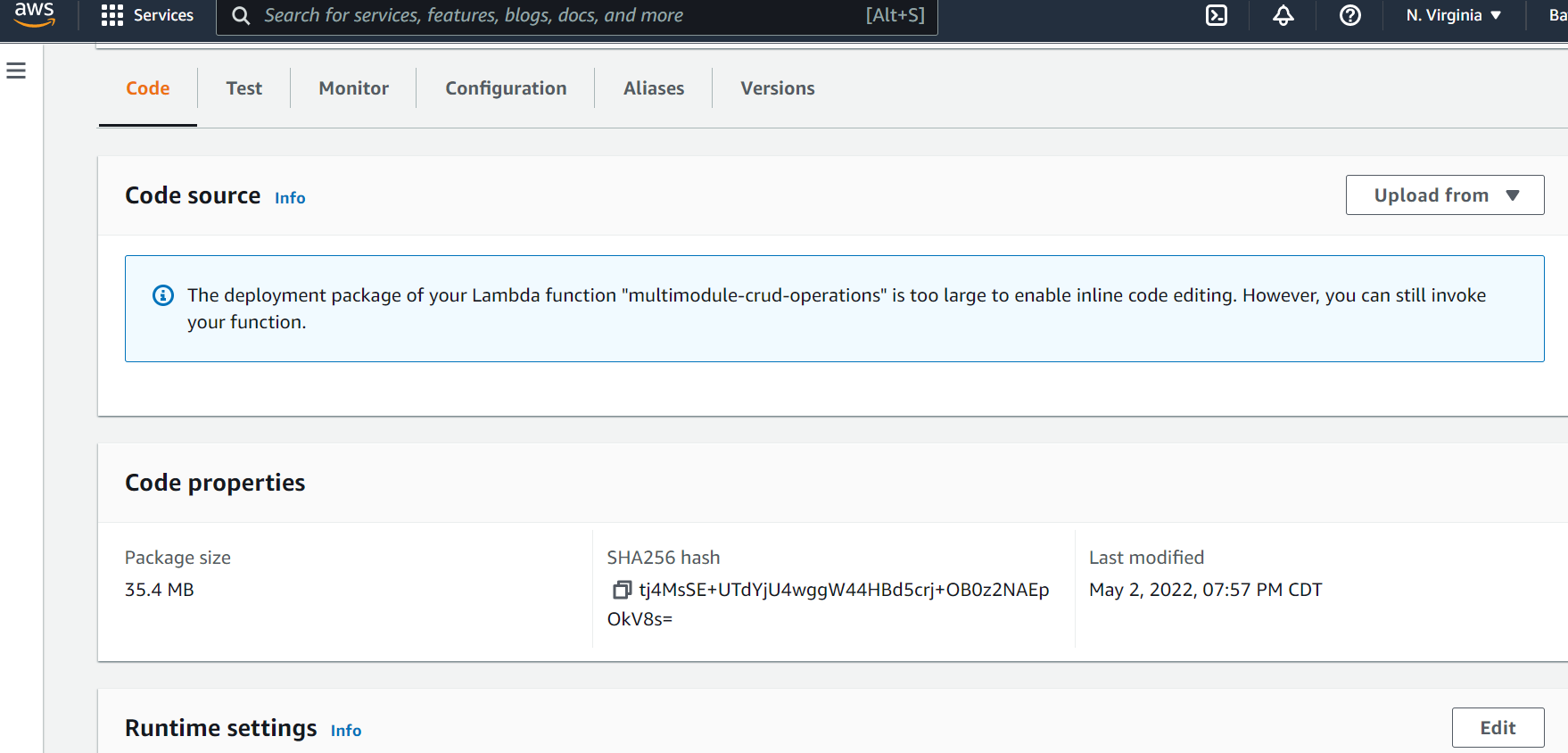


Then the screen below will appear. Click on **Upload from** Button and Select jar file from Your **Folder Path\spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\build\libs\aws-dynamodb-multimodule-api-web-0.0.1-SNAPSHOT-all.jar .**

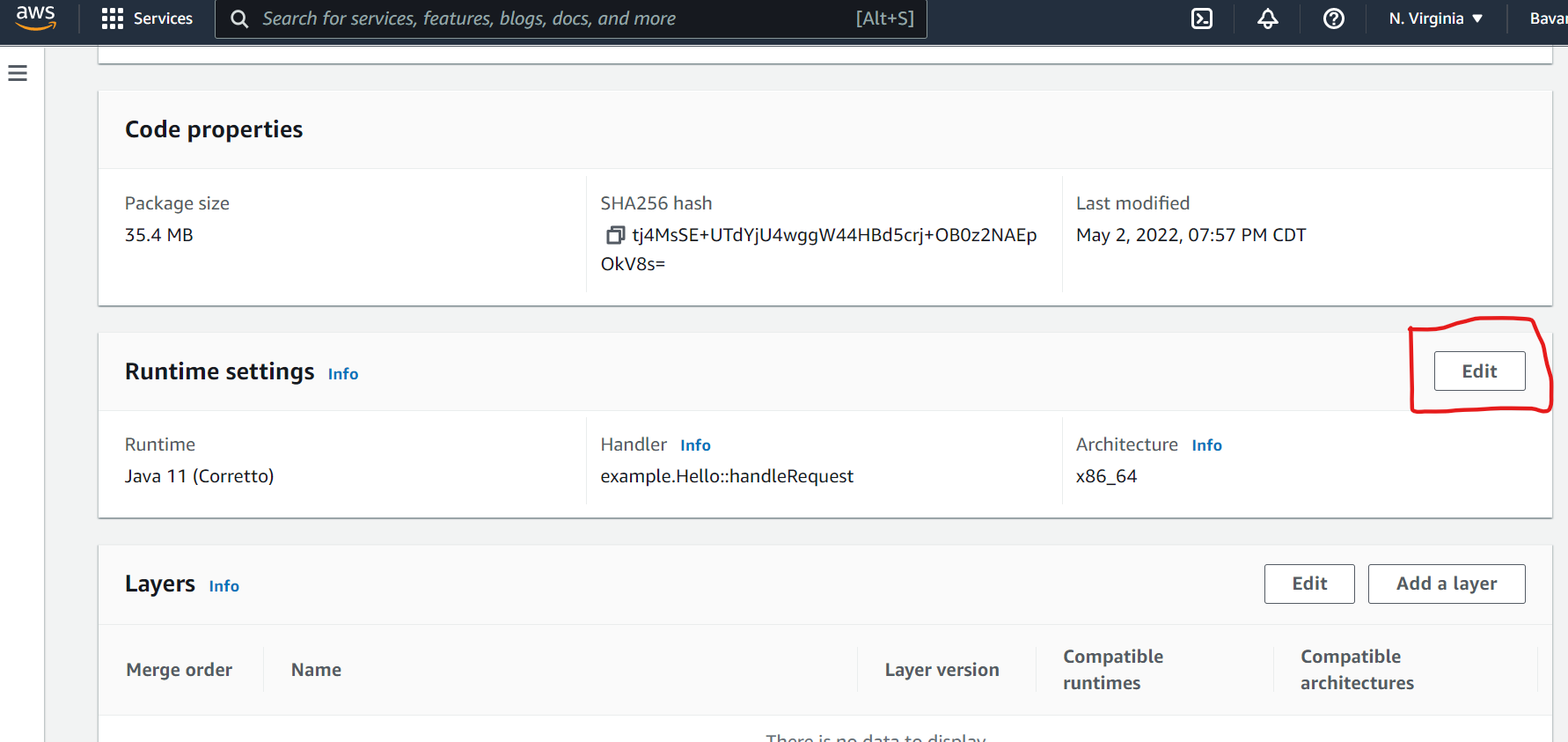
Click on the **Save** Button.



Below screen will appear

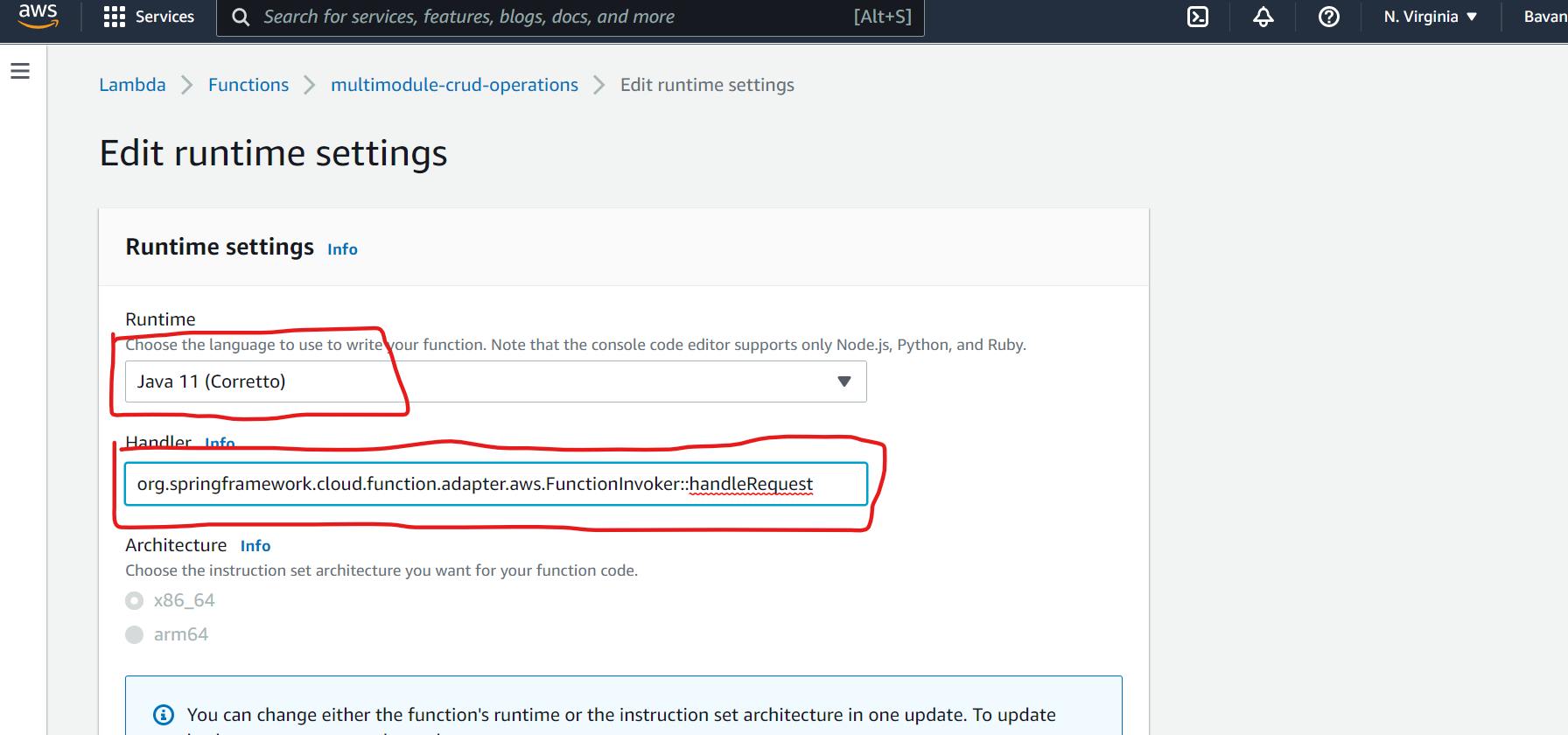


Click on Code and click on Edit in Runtime settings Section.

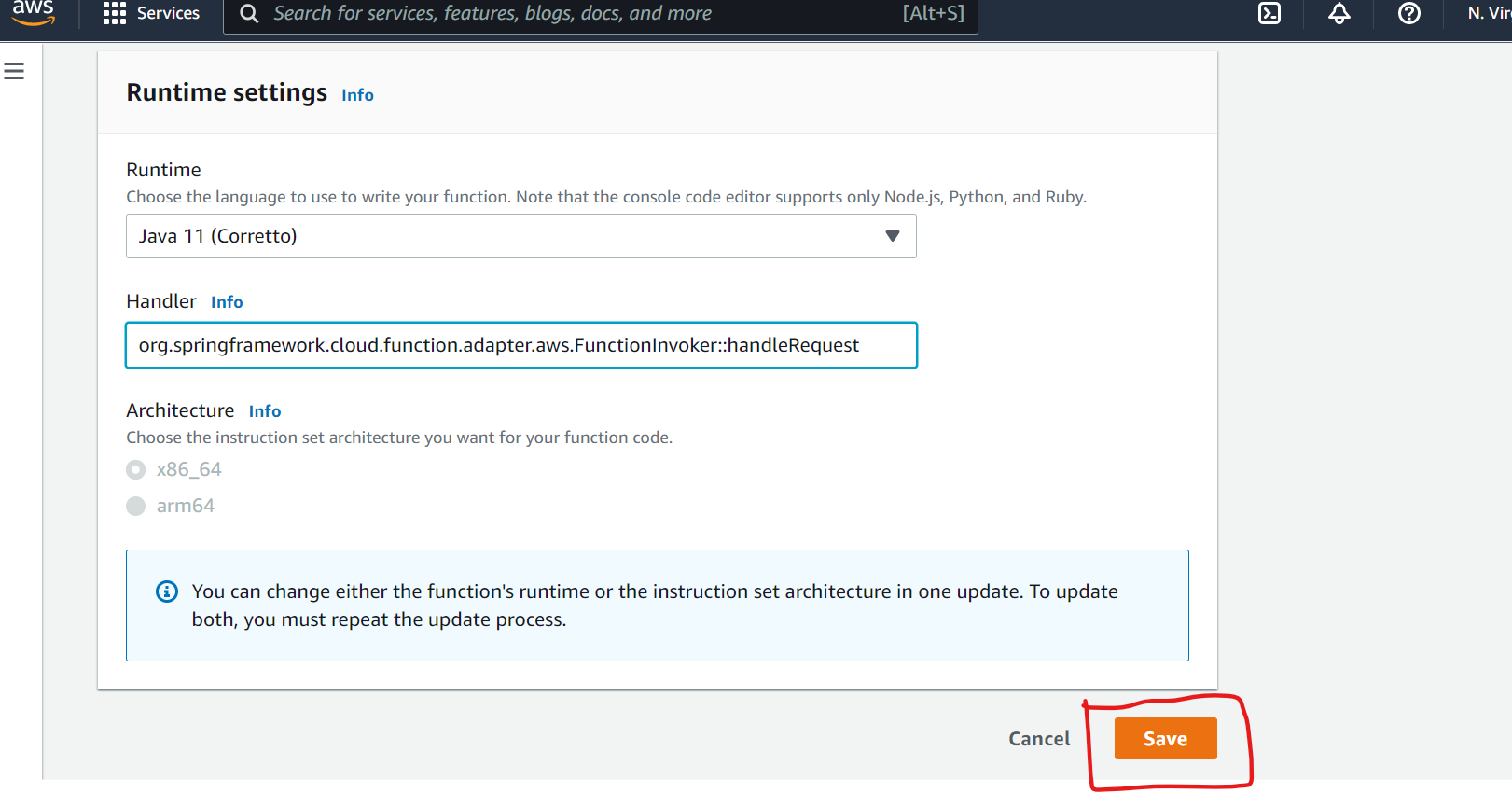


Select Java11(Coretto). If you didn’t select before and add handler in the text box with

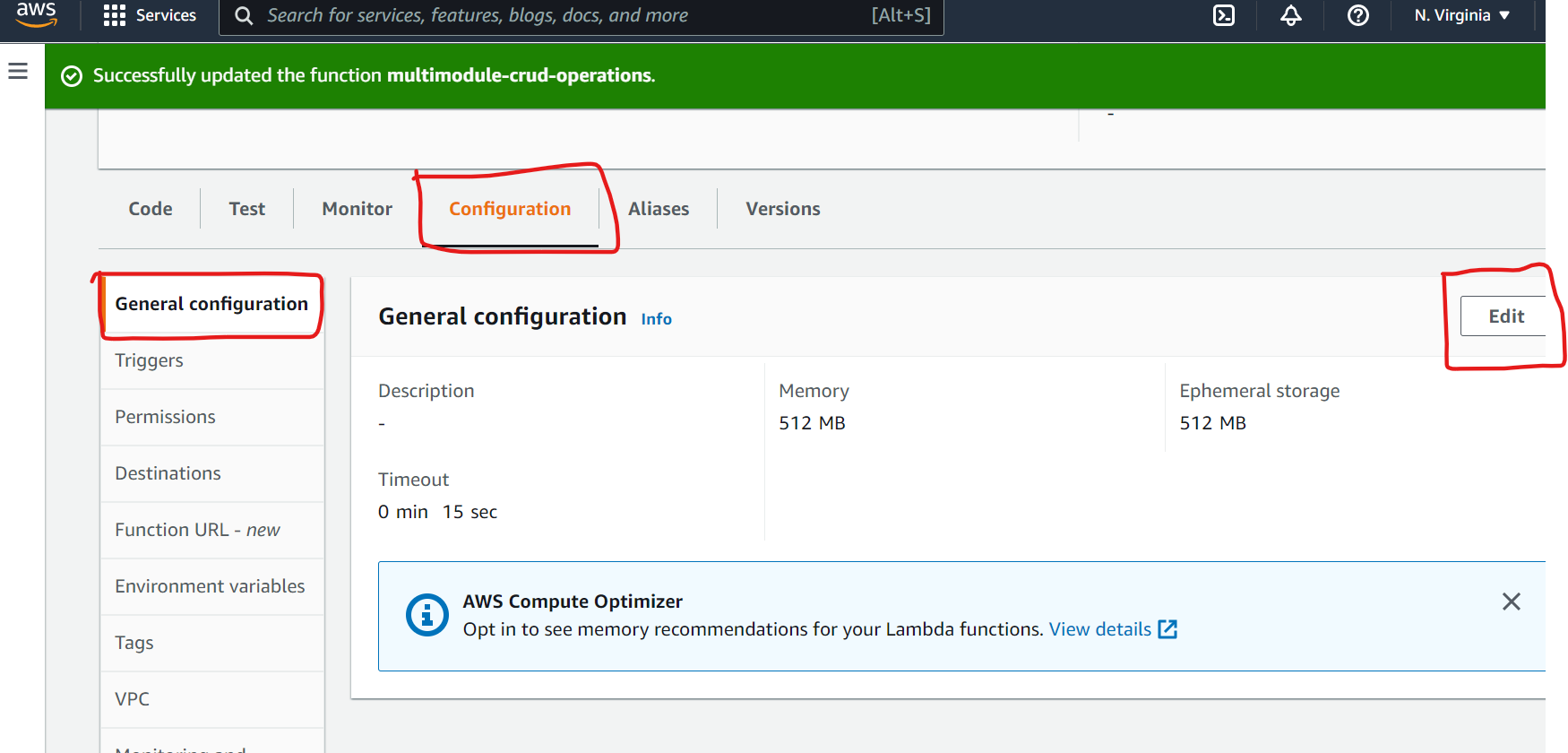
**org.springframework.cloud.function.adapter.aws.FunctionInvoker::handleRequest**



Then Scroll Down and Click on **Save** Button.

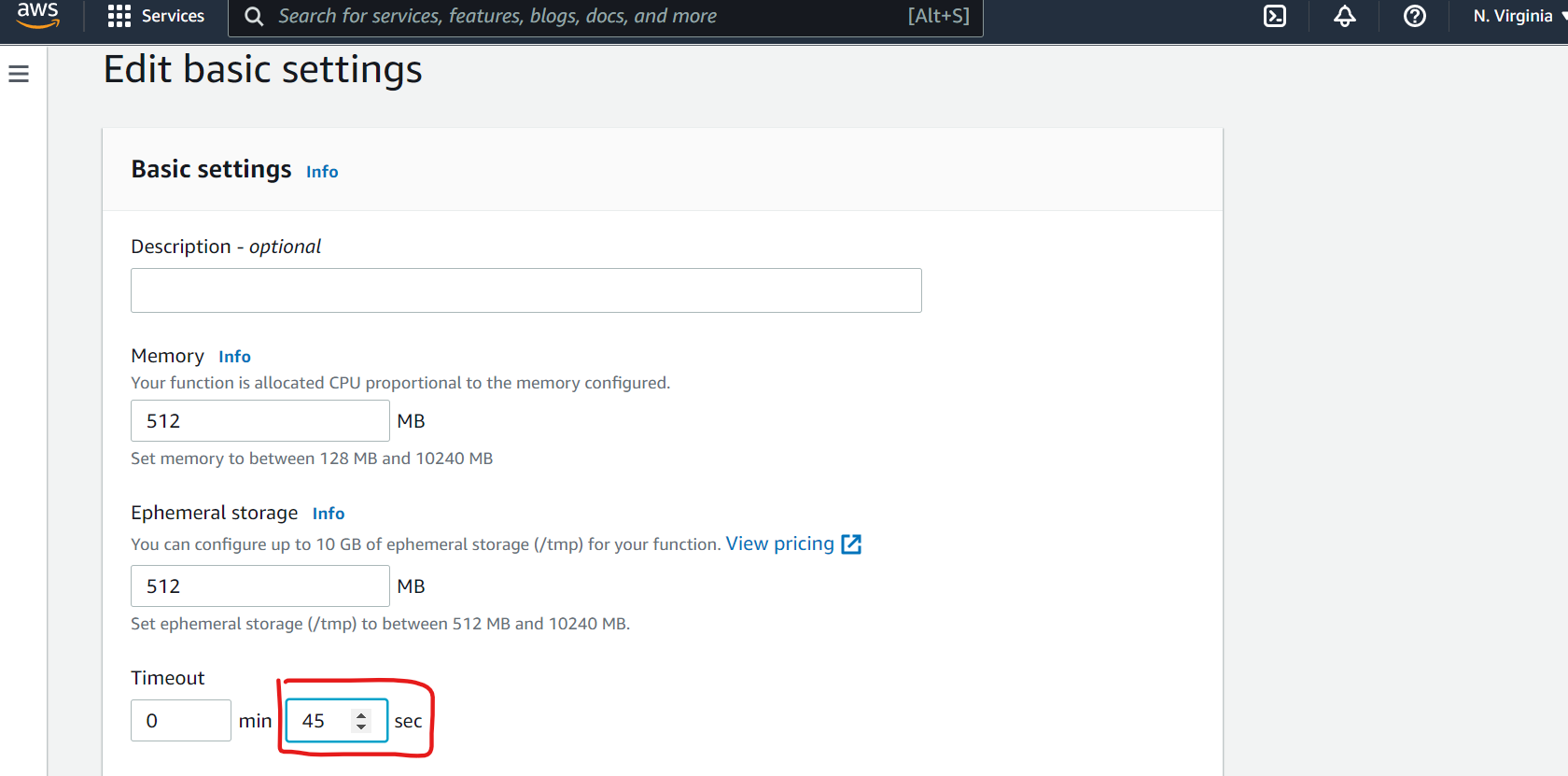


Then Click on “**Configuration**” and click on “**General Configuration**” and click on “**Edit**” Button.

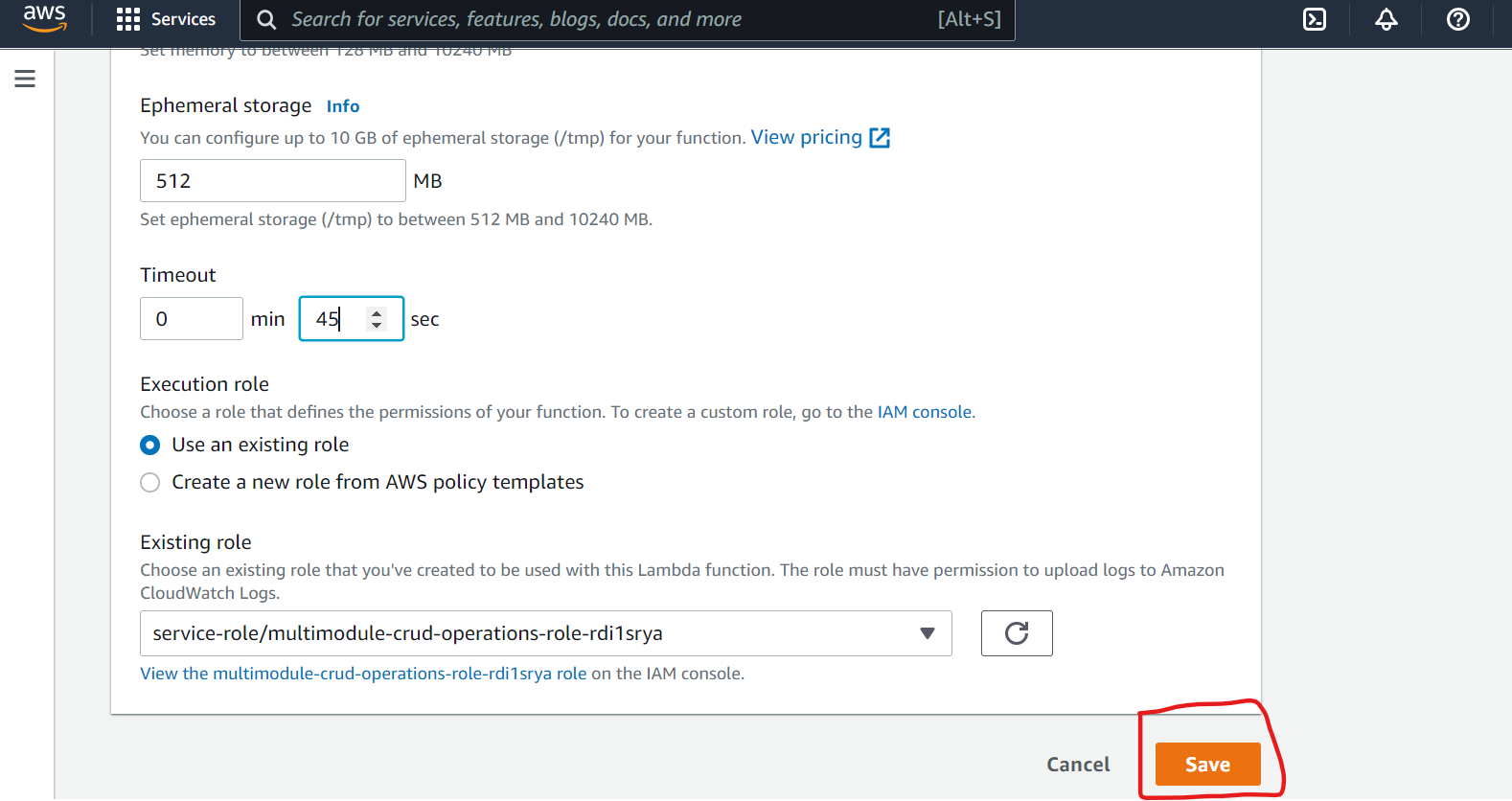


Then Below Screen appears. Change “**Timeout”** 15 to 45 otherwise, Timeout issue will come when try to execute a method.

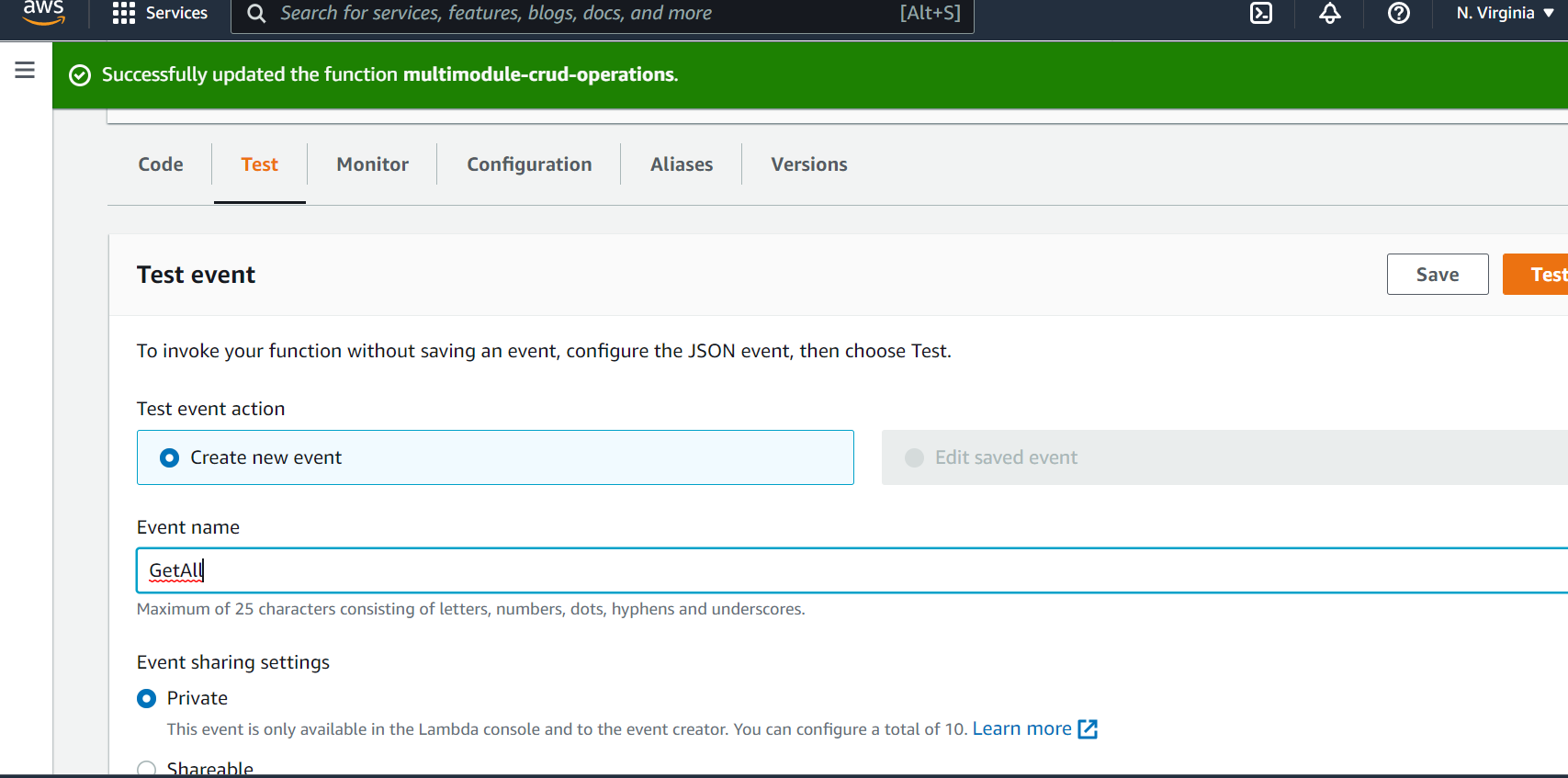
**Note:** Timeout issue will appear, while executing methods.



Scroll Down and Click on **Save** Button**.**



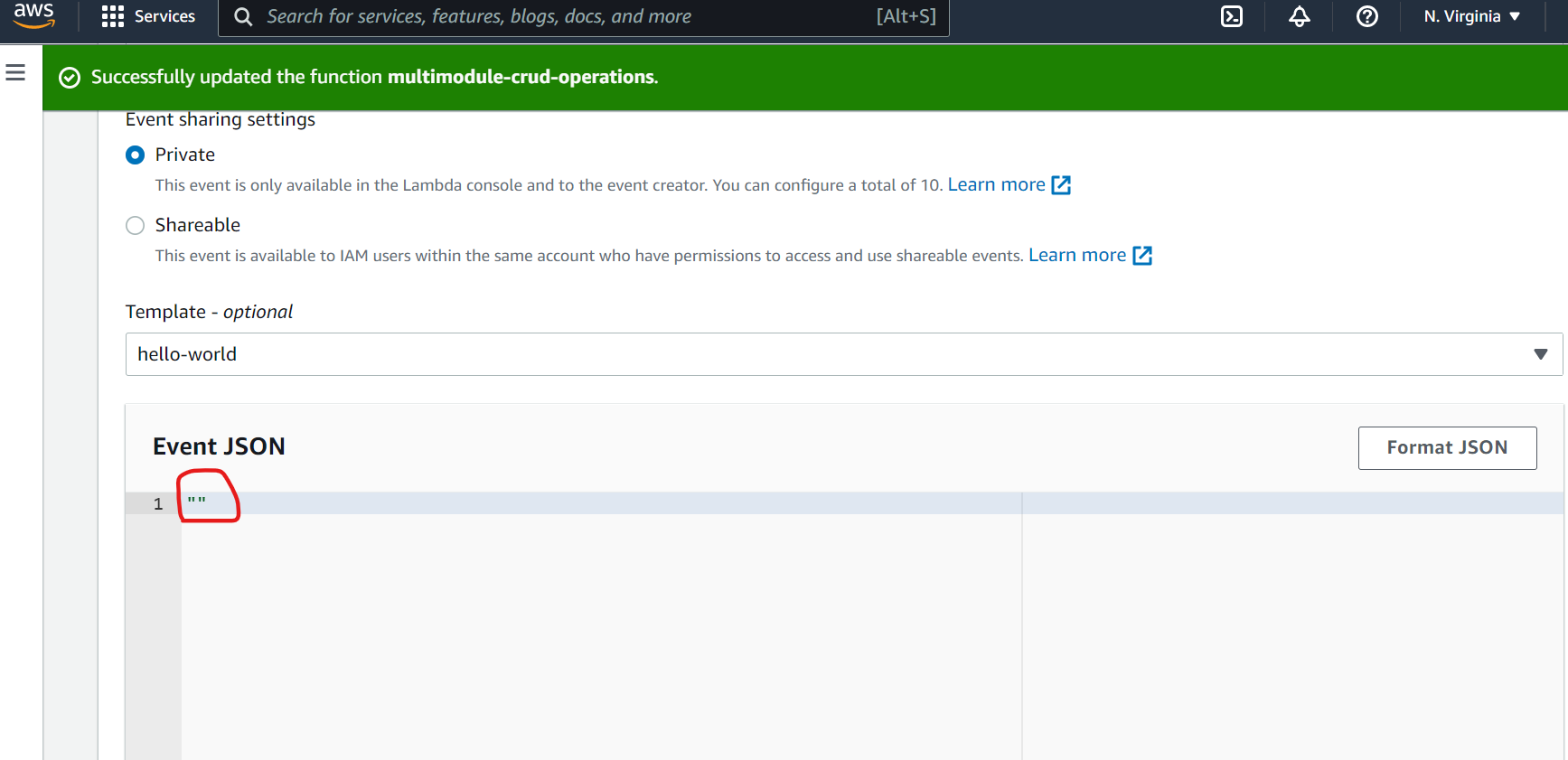
Then Click on **Test** tab. Add **Event Name.**



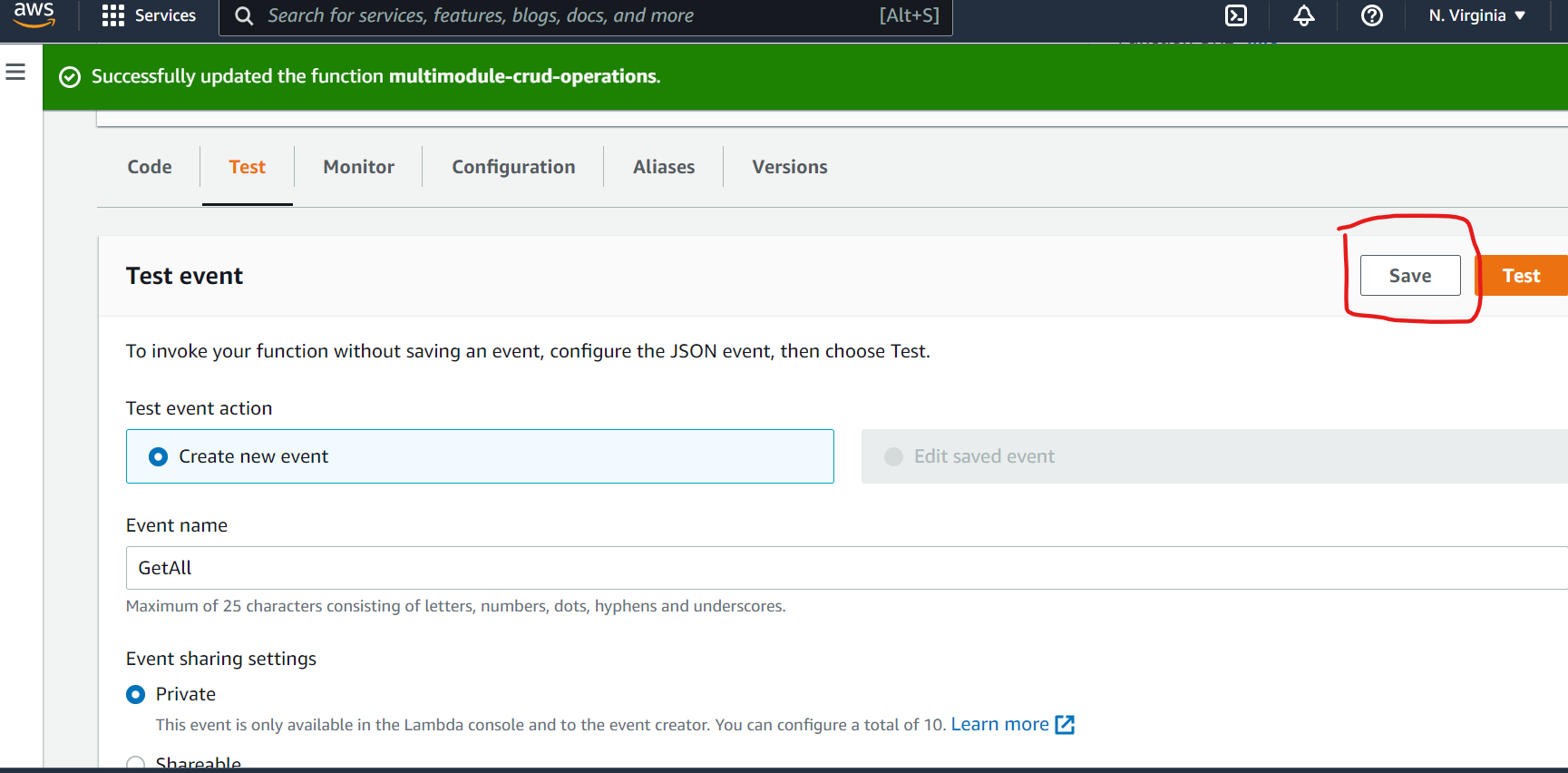
Scroll Down and add Input inside **Event Json**.

For Example:

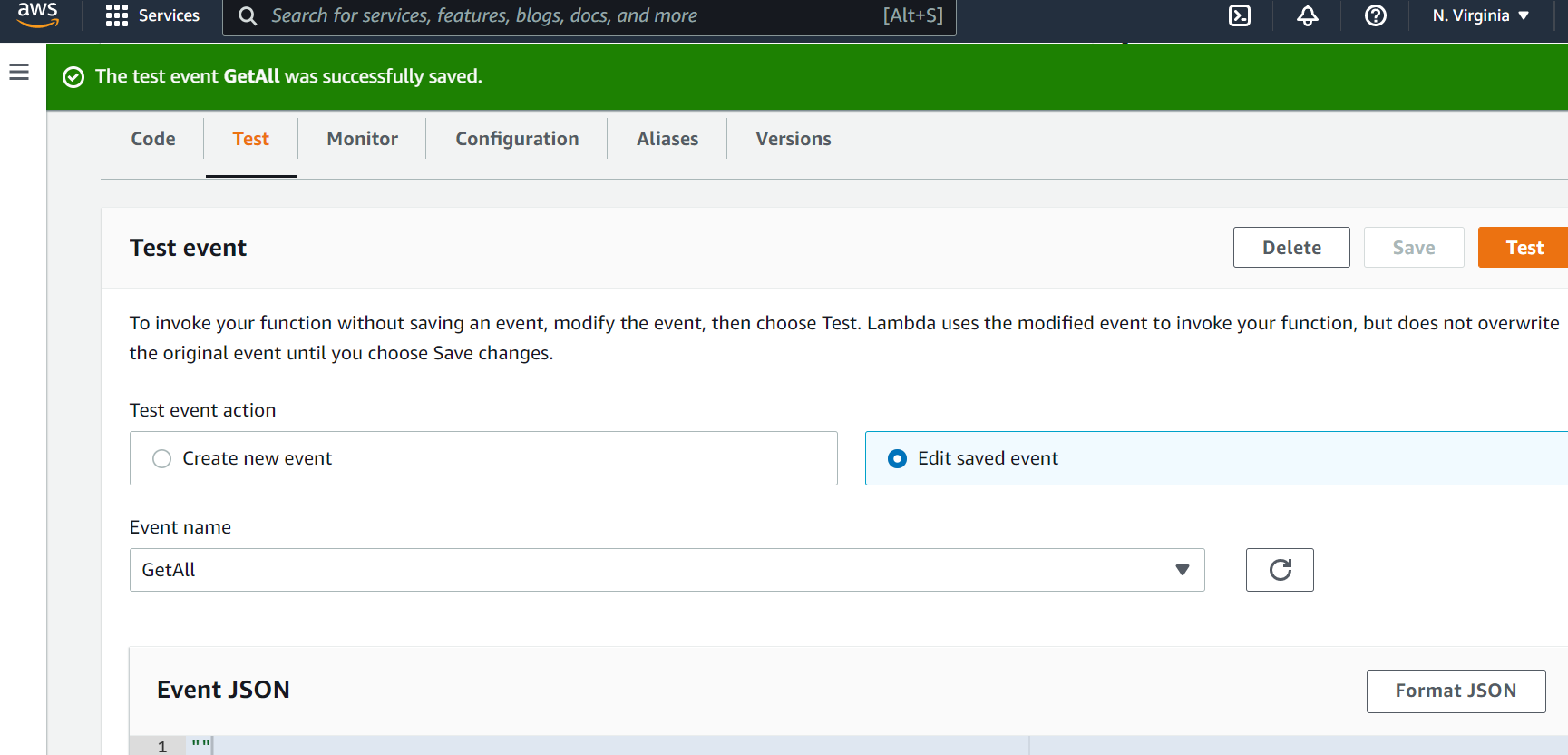
1. **getAllOrder** method, we are passing input as **“”** quotes.



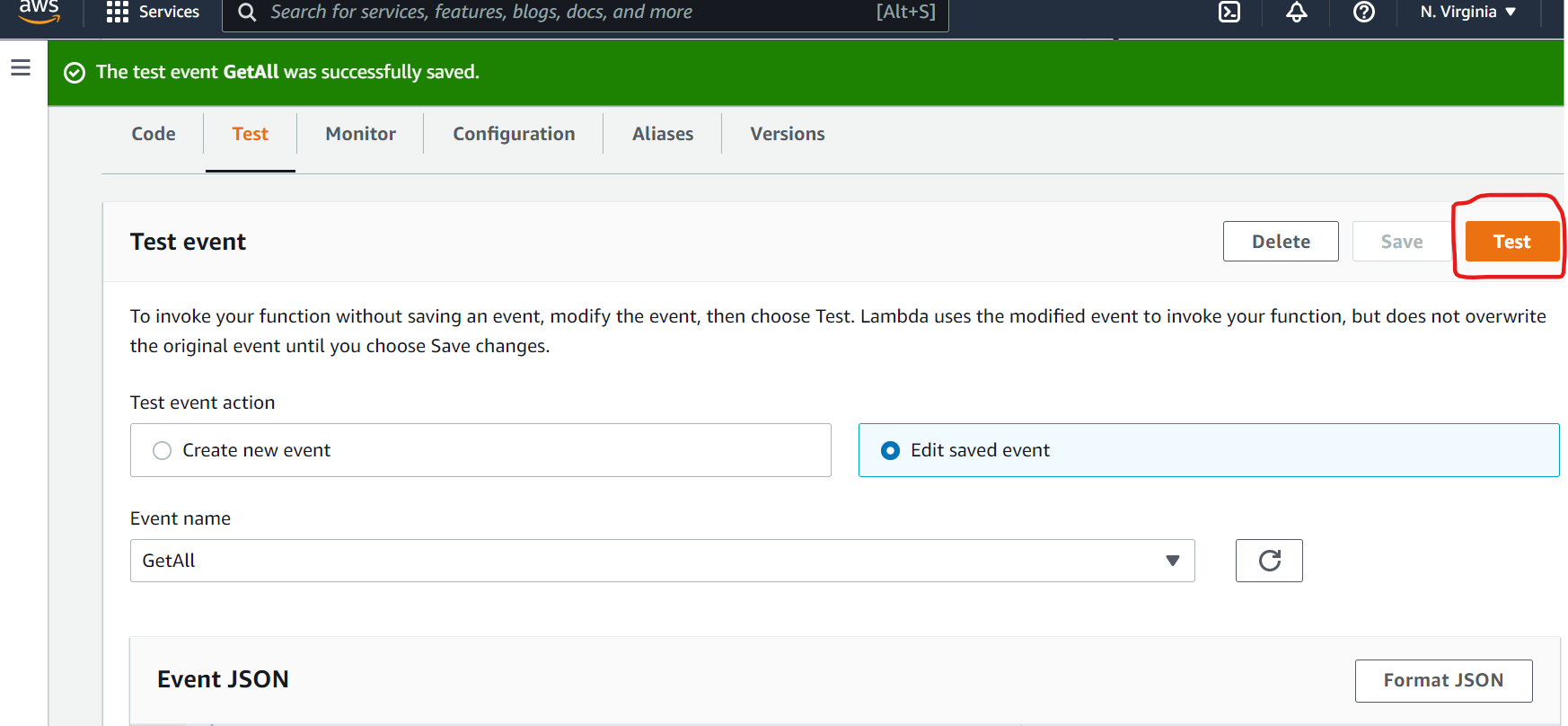
Scroll up and Click on **Save** Button.



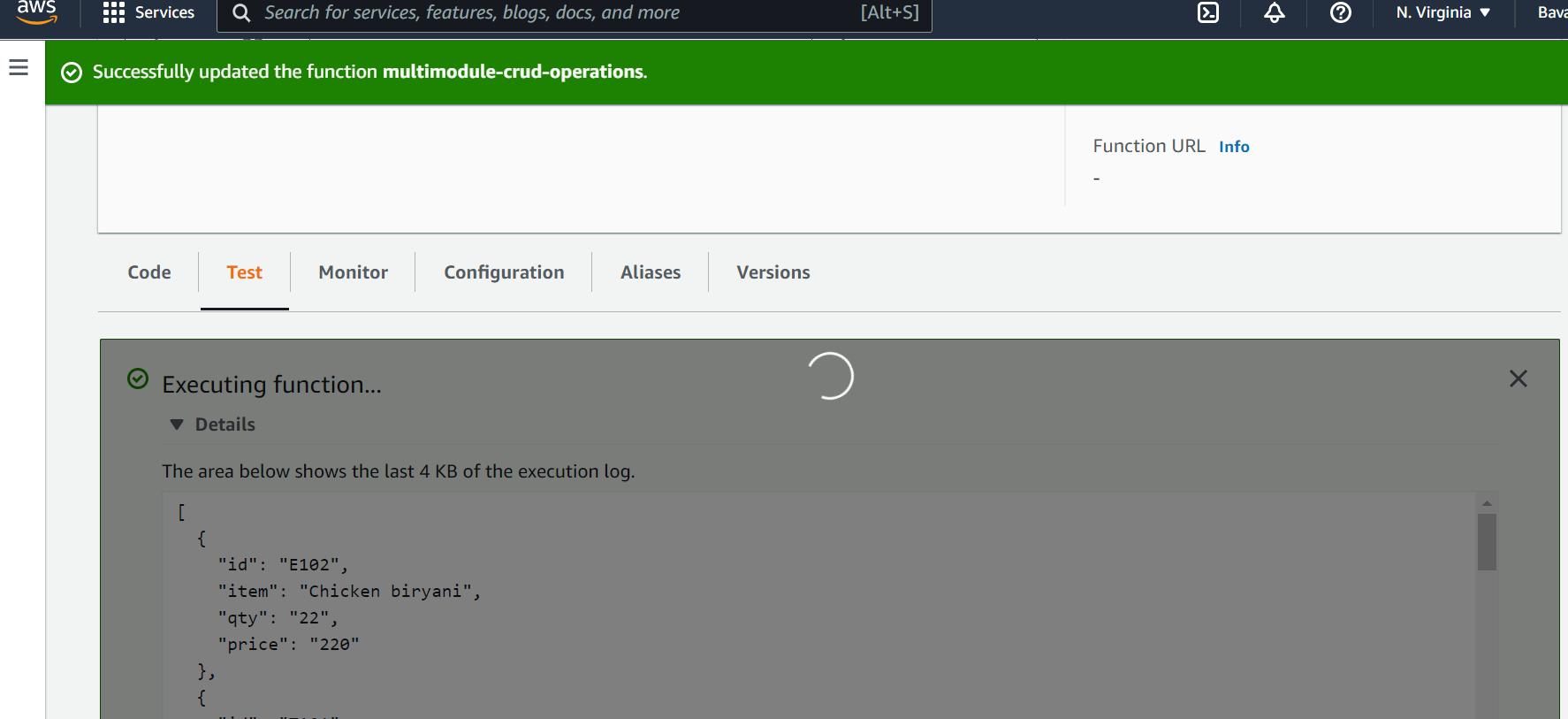
Then below page will appear.



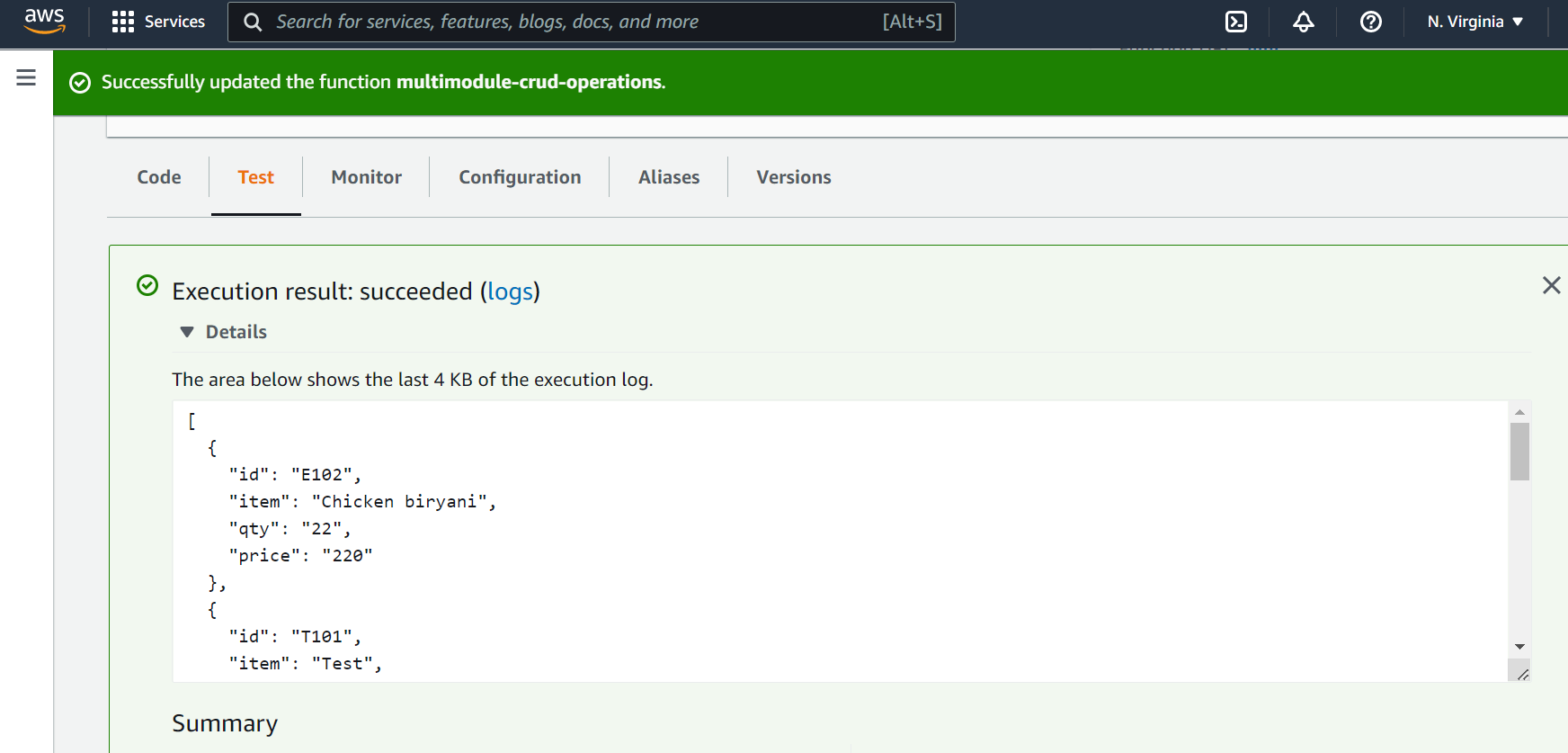
Click on **Test** Button to test spring cloud function :getAllOrder



Then this page will appear.



Output:



# getOrderById Method invocation in AWS LAMBDA

**In Root Directory ->go to below path**

**spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\src\main\resources\aws-credentials.properties**

**And uncomment getOrderById method definition like this**

**#****spring.cloud.****function.definition=getAllOrder**

**spring.cloud.function.definition=getOrderById**

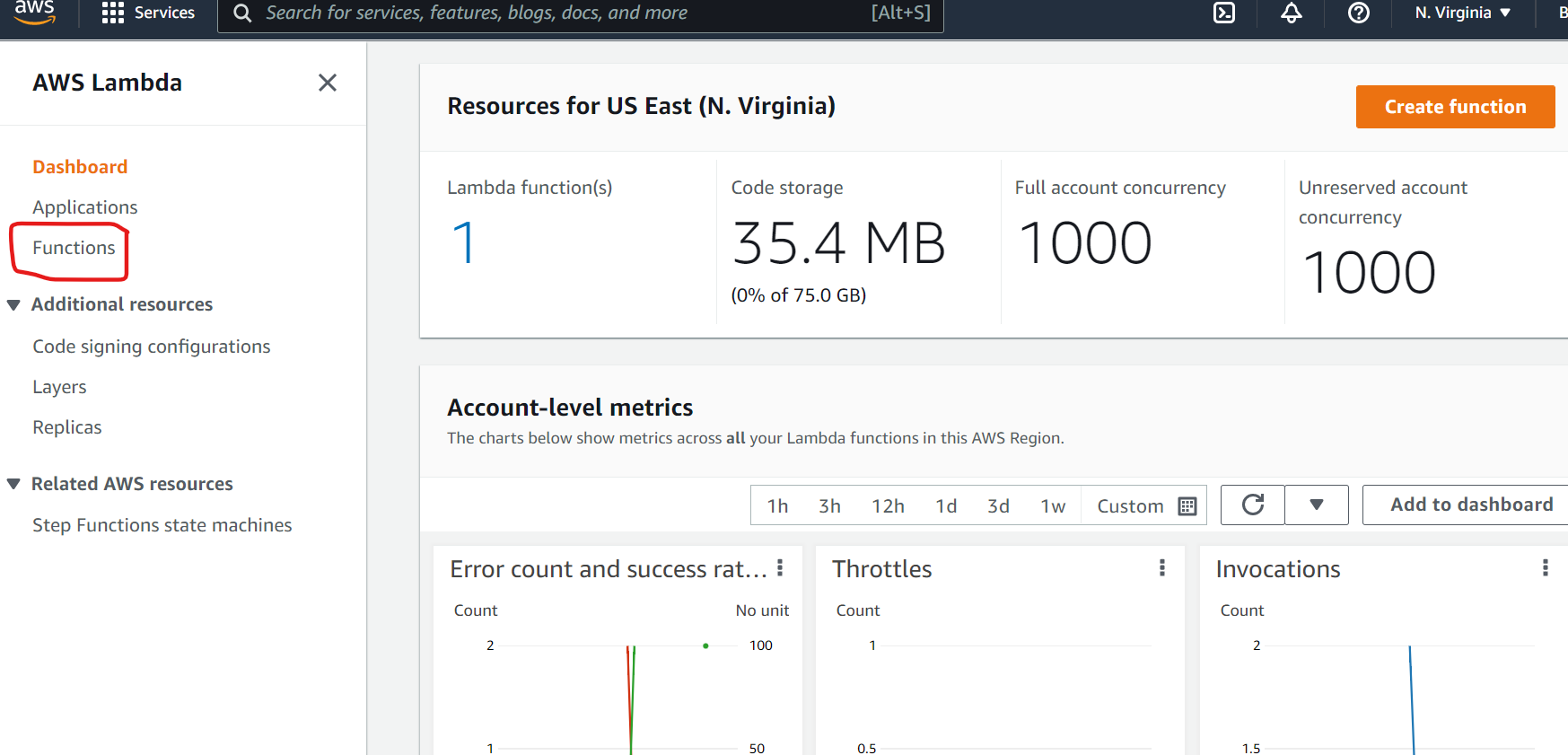
**#spring.cloud.function.definition=saveOrder**

**#spring.cloud.function.definition=deleteOrder**

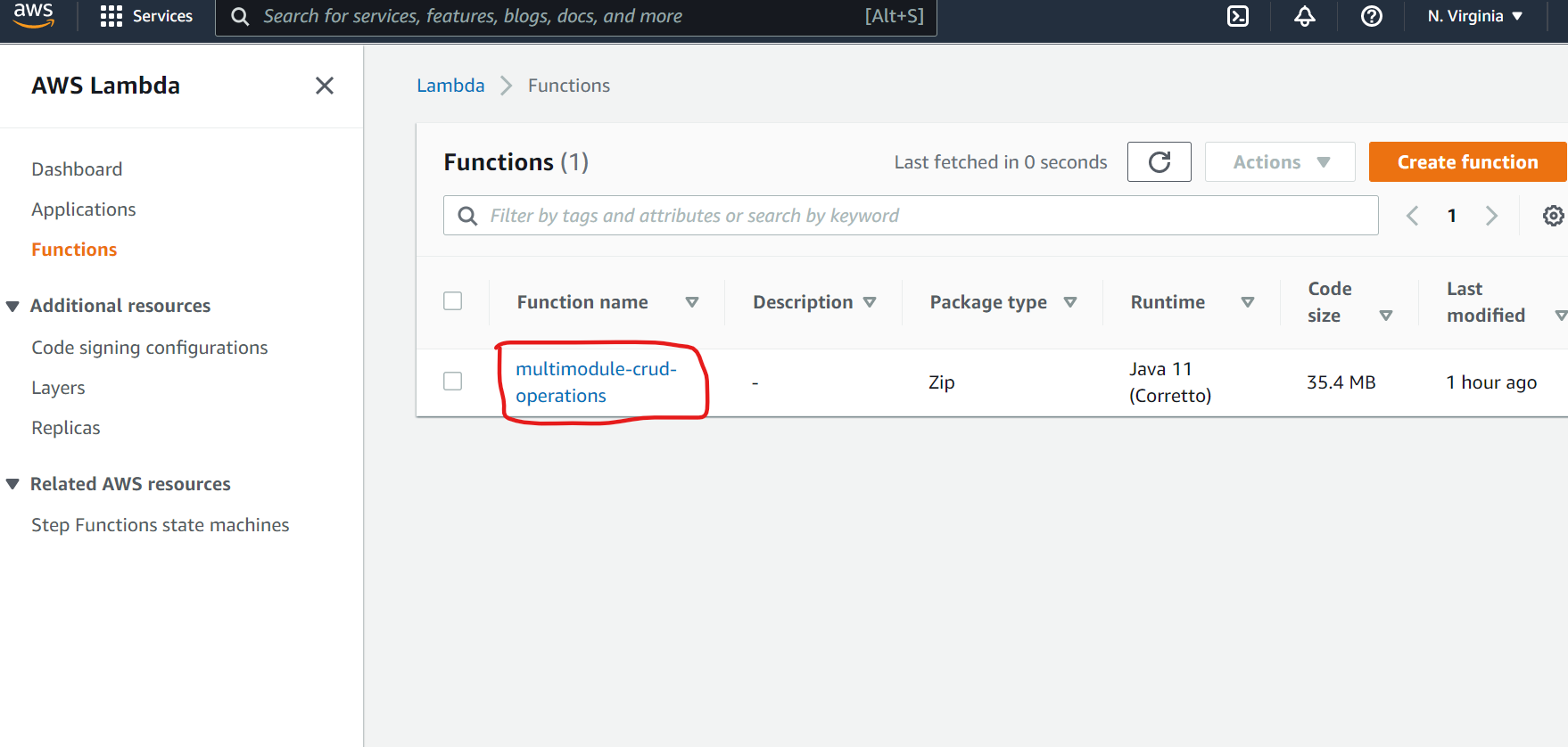
**Then use the commands below to clean and build spring boot application.**

* **./gradlew clean**
* **./gradlew build**

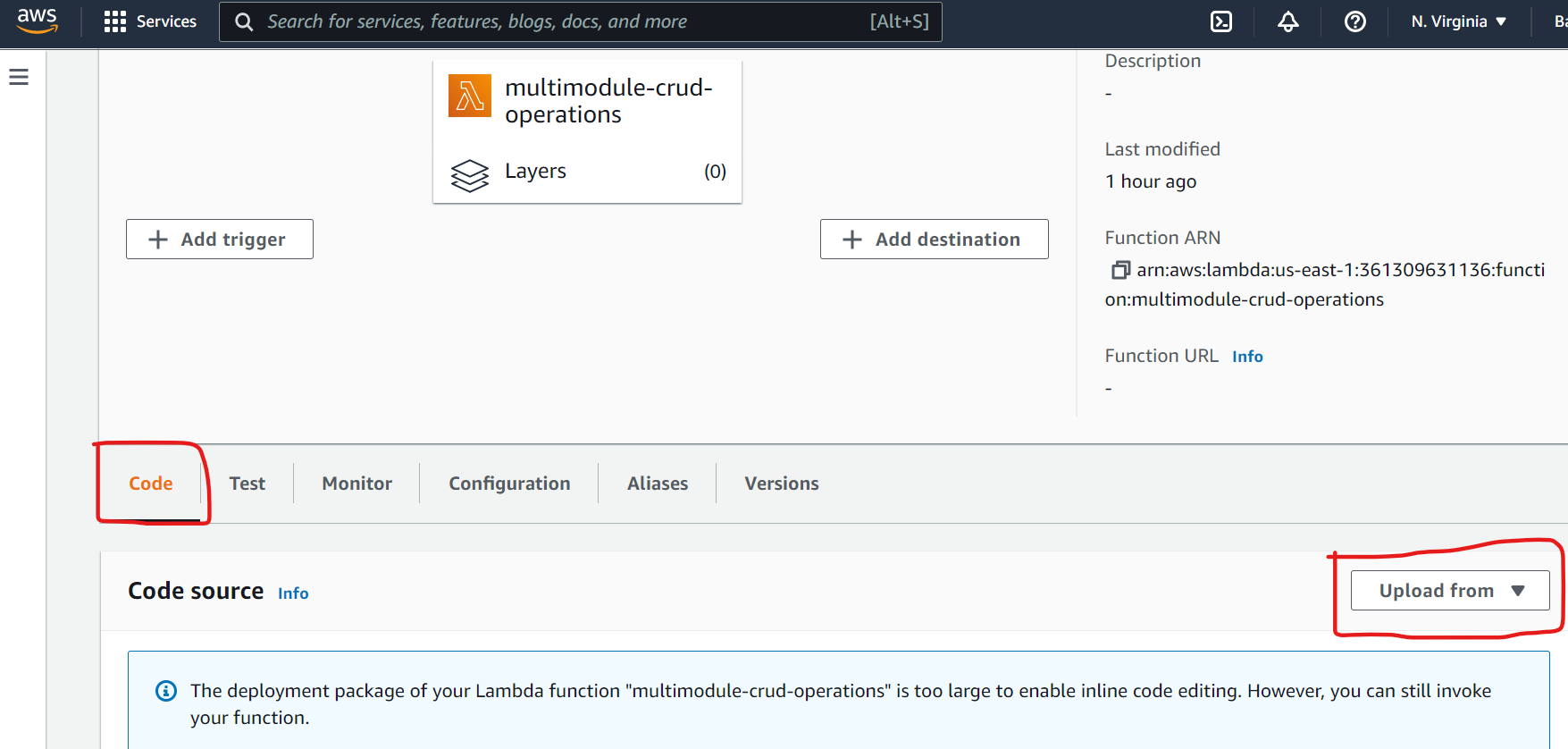
Go to Aws Services->Select Lambda->Click on **Functions**.



Then below page will appear. Click on **multimodule-crud-operations**

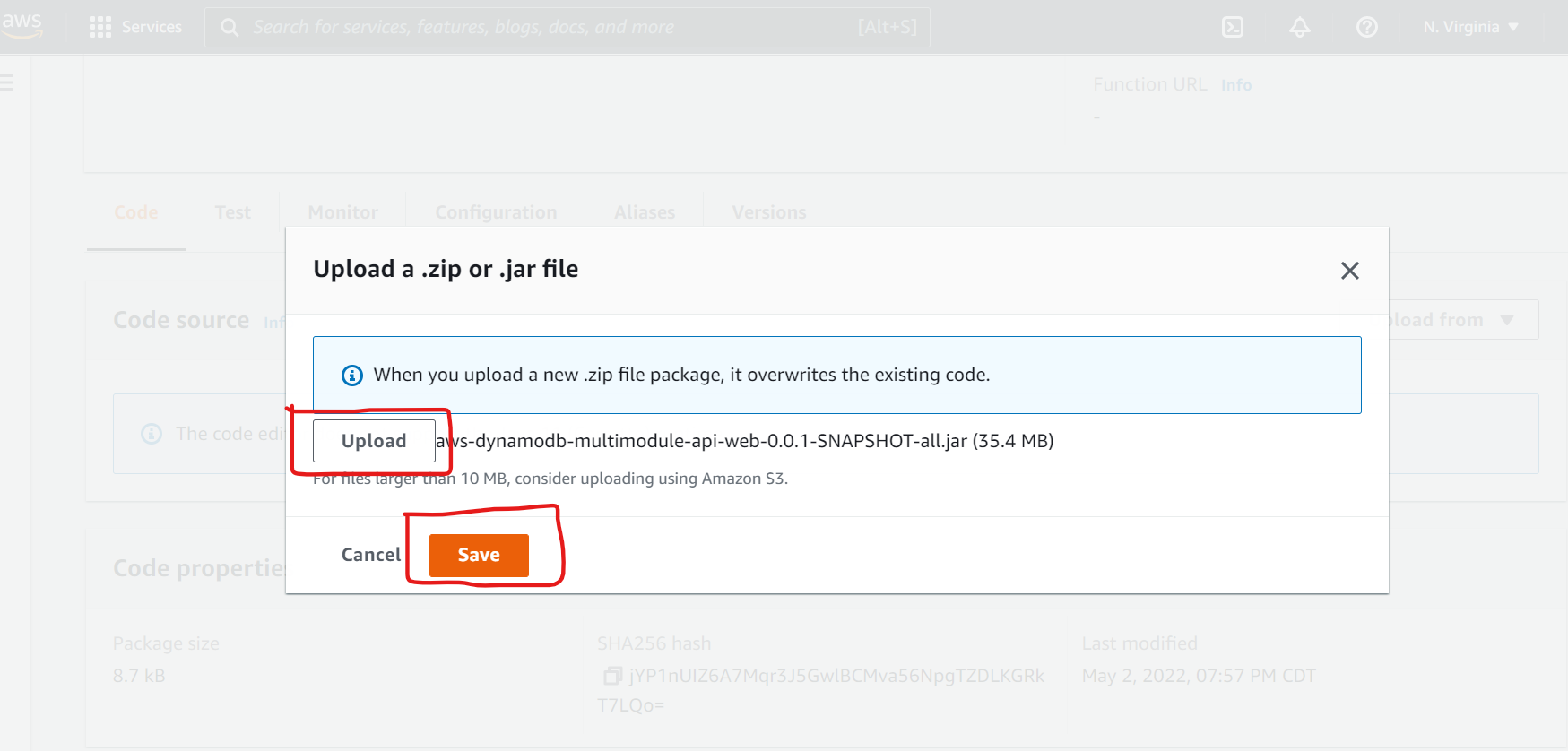


This below screen will appear. Then Click on **Code. Select upload from**

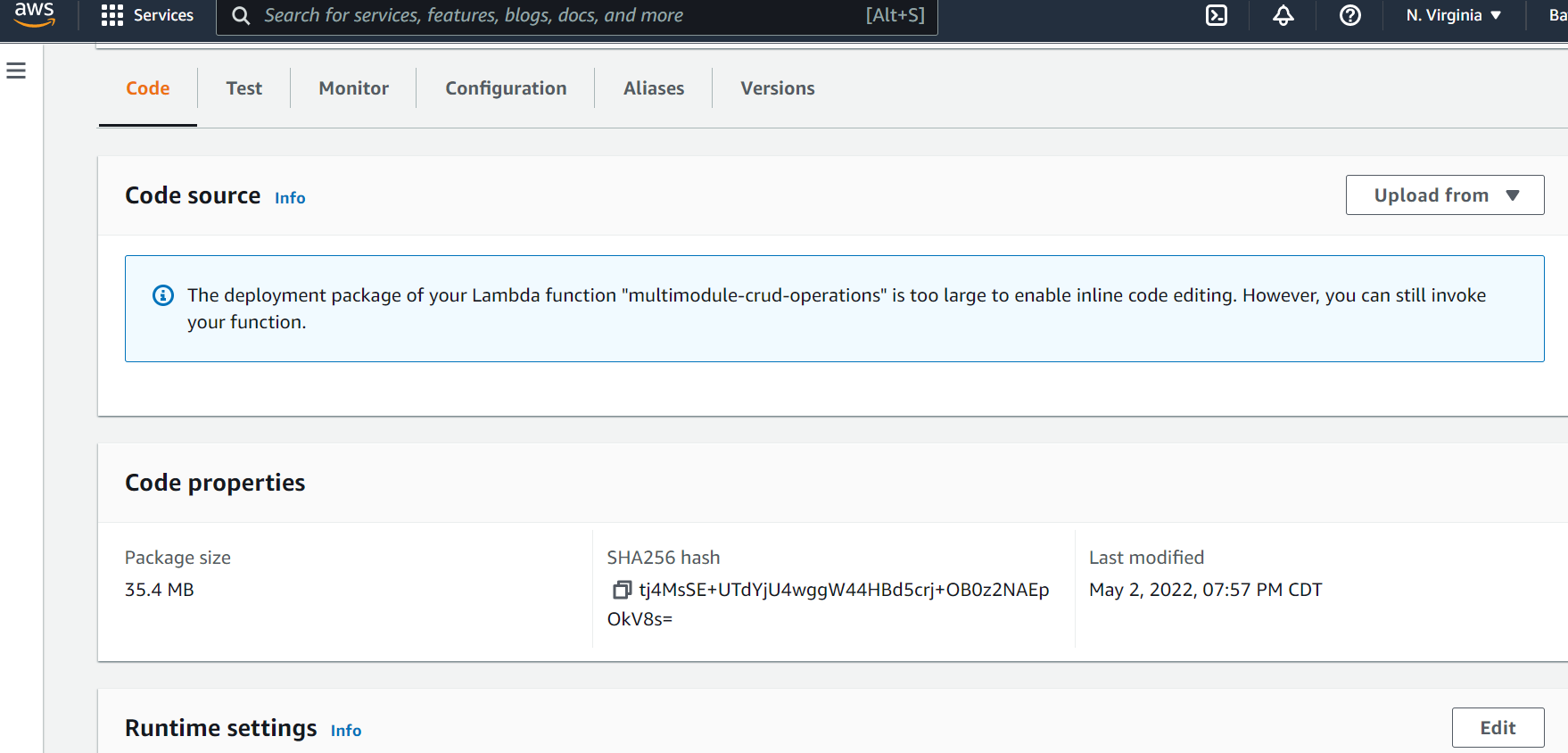


Then the screen below will appear. Click on **Upload from** Button and Select jar file from Your **Folder Path\spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\build\libs\aws-dynamodb-multimodule-api-web-0.0.1-SNAPSHOT-all.jar .**

Click on the **Save** Button.

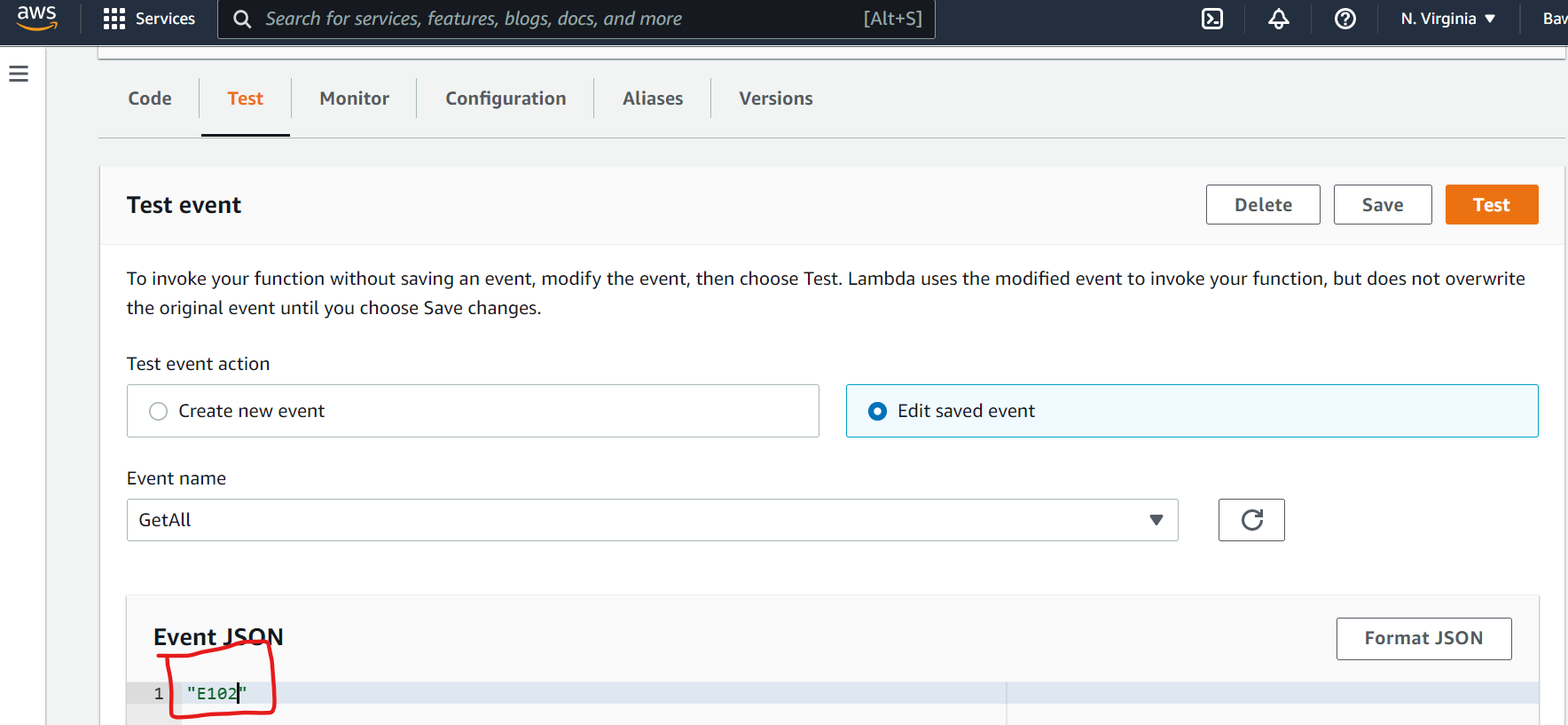


Then below screen will appear

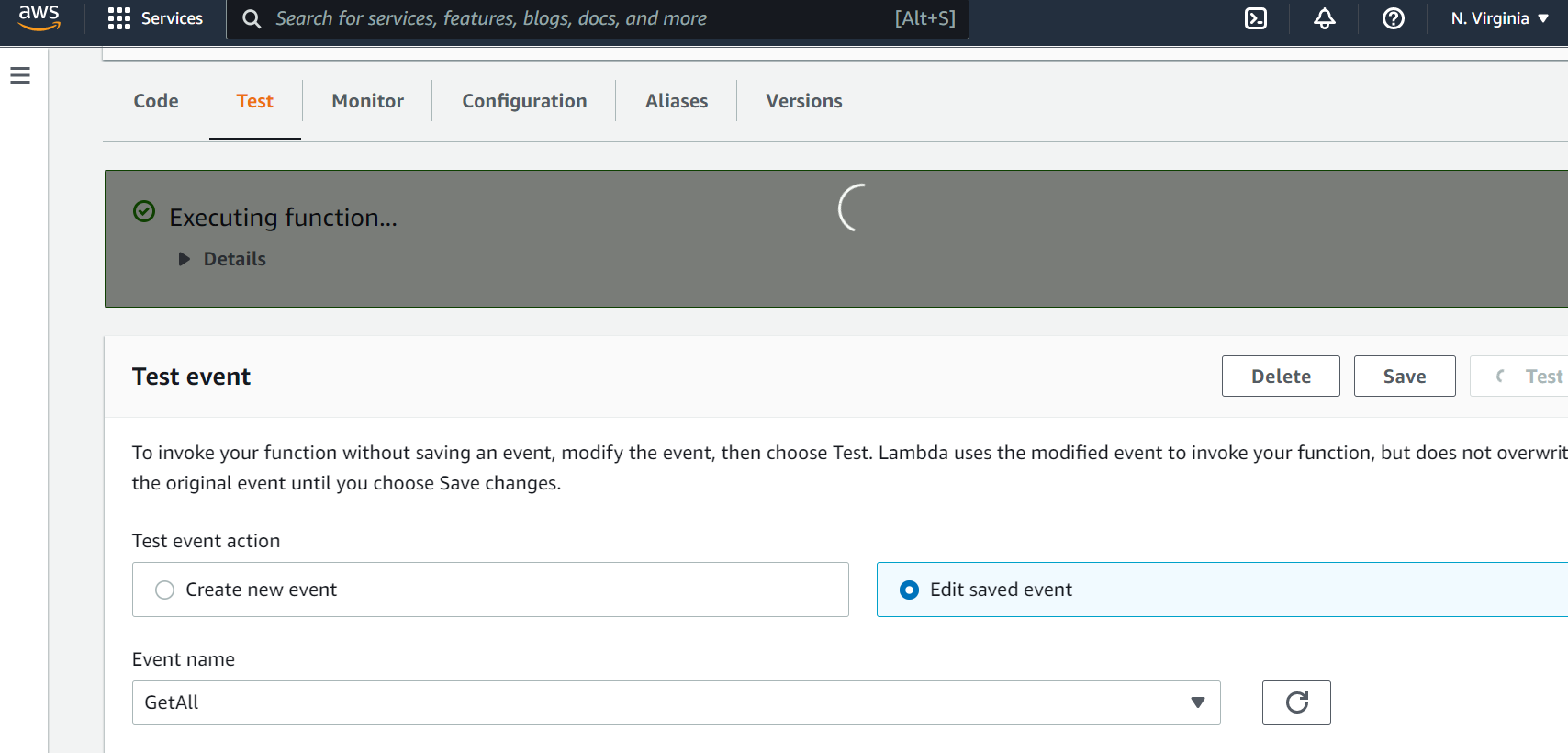


Then Click on **Test** tab.

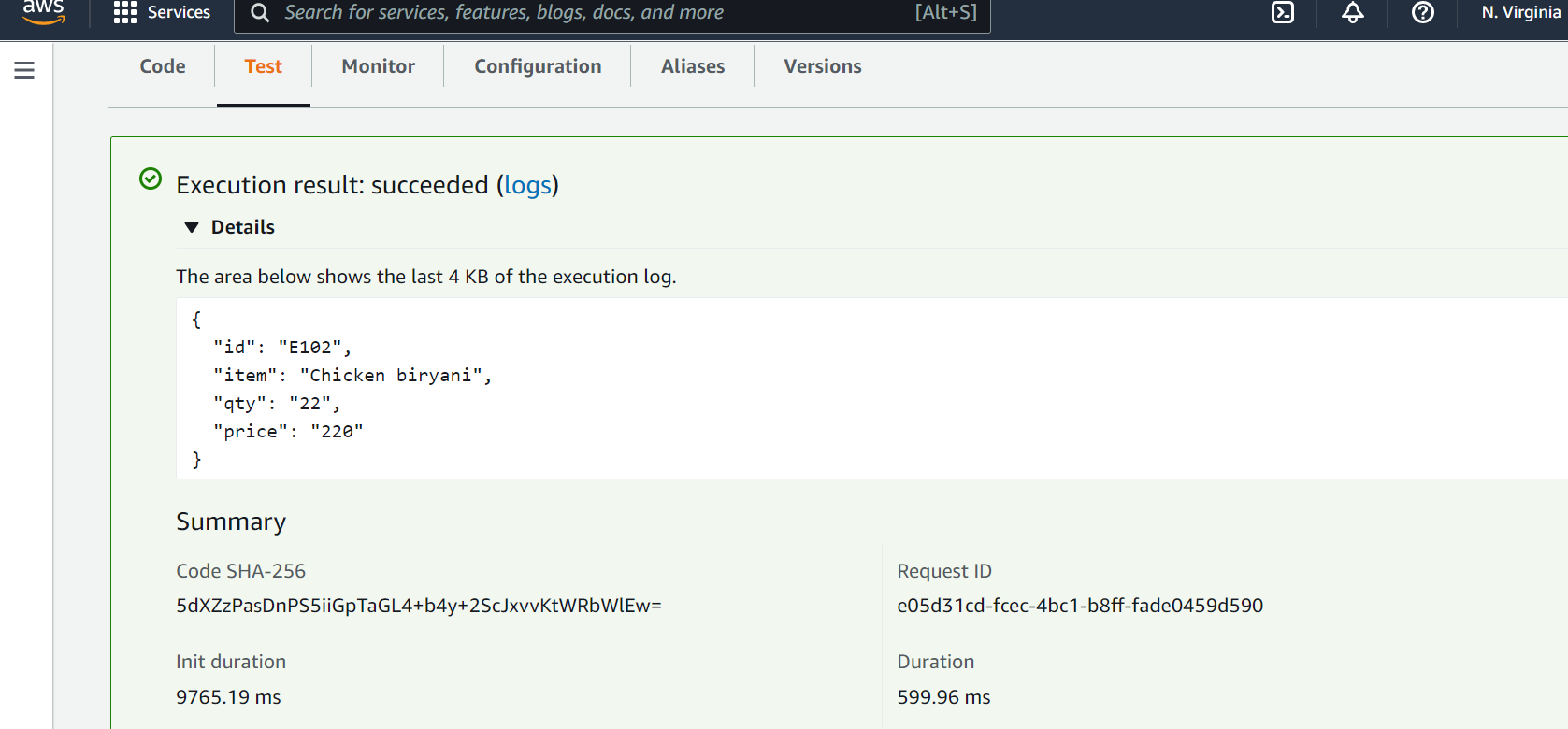
**Ex: getOrderById** method, we are passing input as **“E102”.**



Then Click on **Test** Button.



Output:



# saveOrder Method invocation in AWS LAMBDA

**In Root Directory ->go to below path**

**spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\src\main\resources\aws-credentials.properties**

**And uncomment saveOrder method definition like this**

**#spring.cloud.function.definition=getAllOrder**

**#spring.cloud.function.definition=getOrderById**

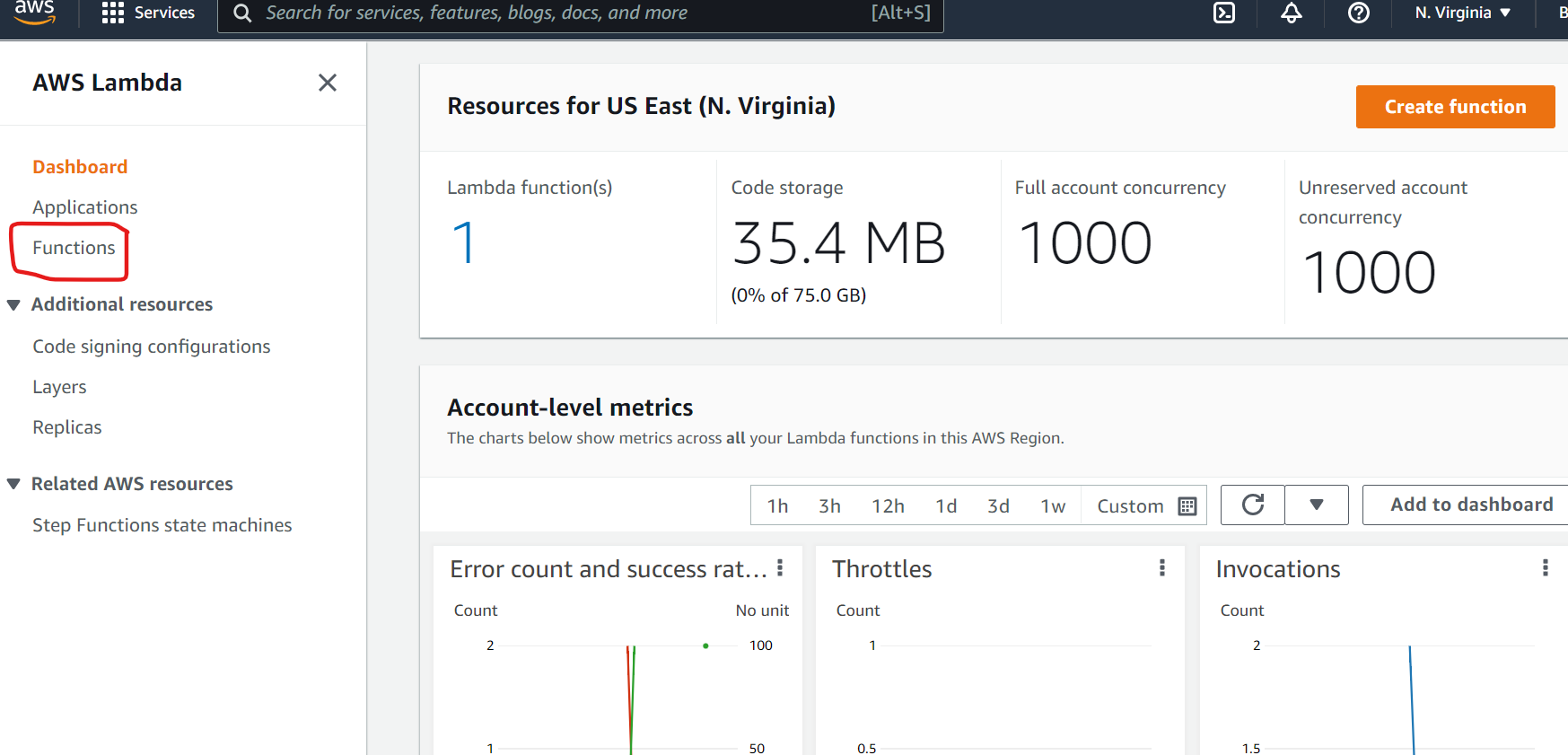
**spring.cloud.function.definition=saveOrder**

**#spring.cloud.function.definition=deleteOrder**

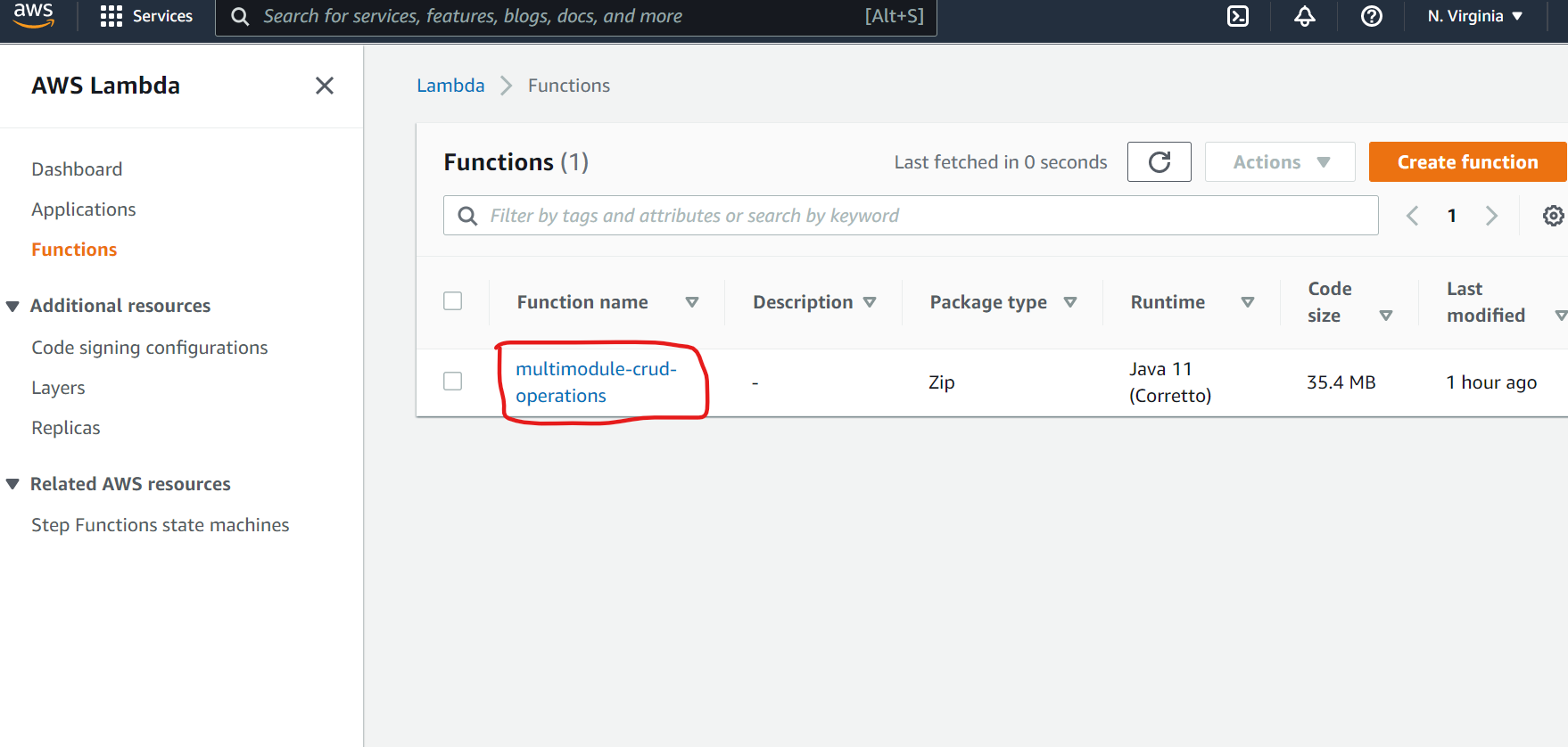
**Then use the commands below to clean and build spring boot application.**

* **./gradlew clean**
* **./gradlew build**

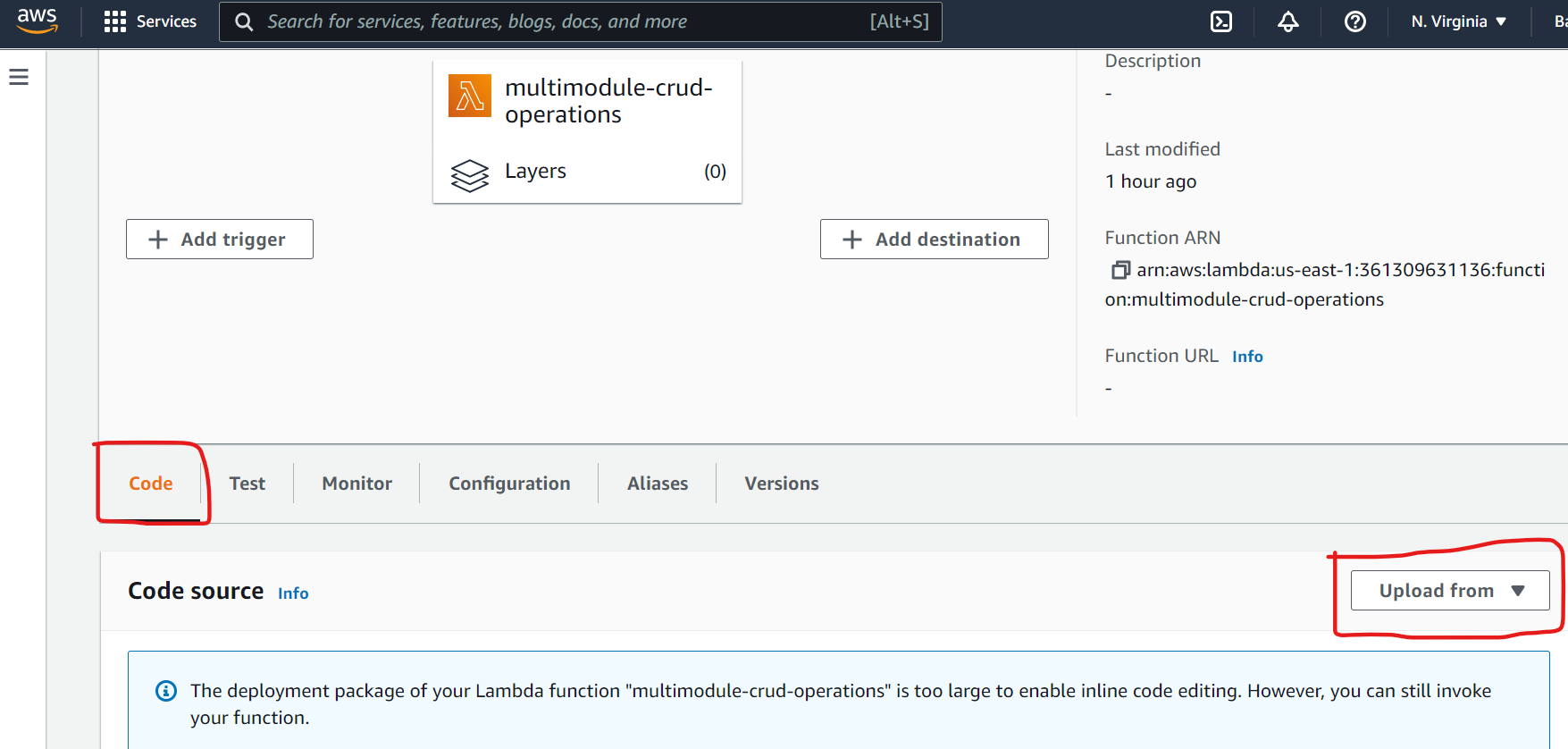
Go to Aws Services->Select Lambda->Click on **Functions**.



Then below page will appear. Click on **multimodule-crud-operations**

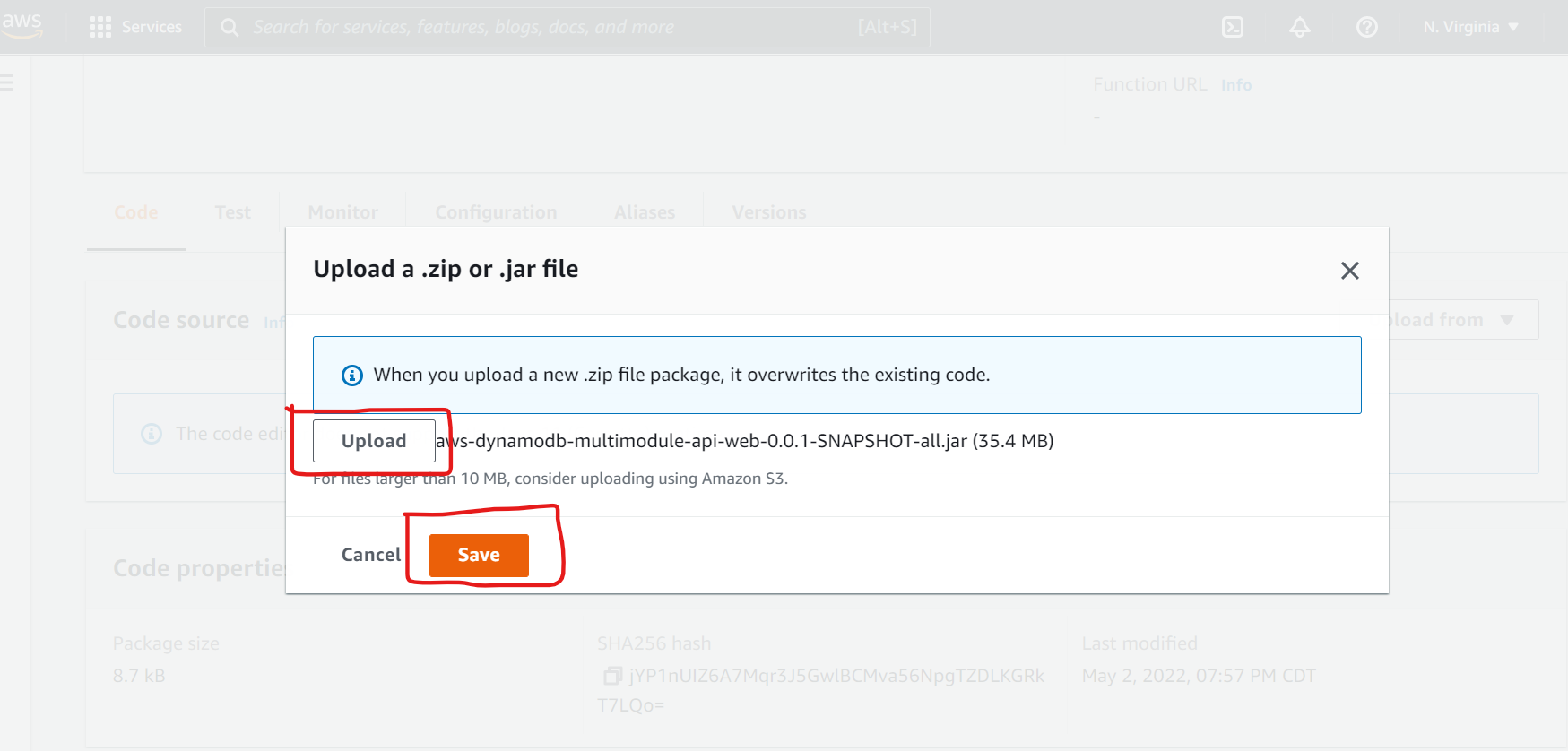


This below screen will appear. Then Click on **Code. Select upload from**

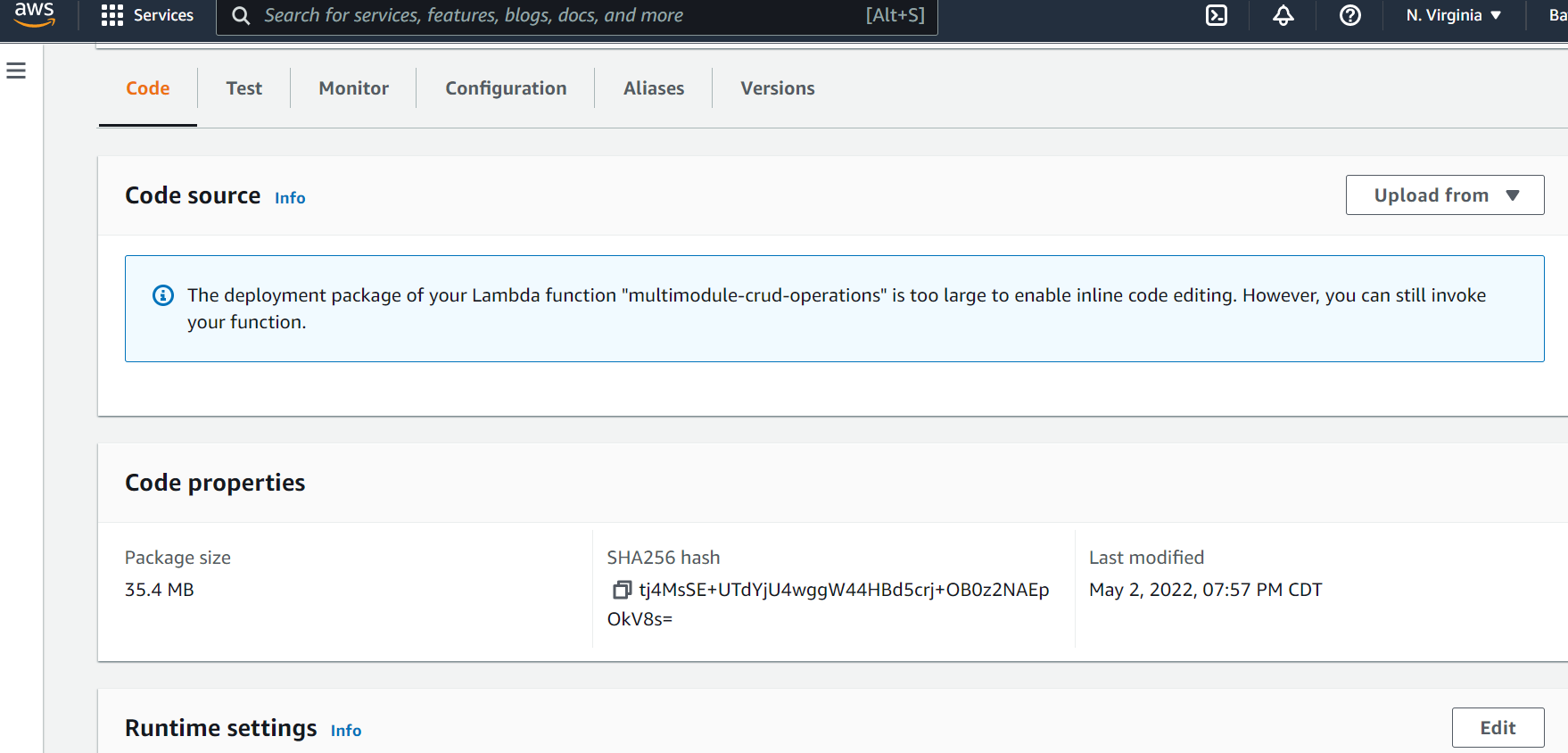


Then the screen below will appear. Click on **Upload from** Button and Select jar file from Your **Folder Path\spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\build\libs\aws-dynamodb-multimodule-api-web-0.0.1-SNAPSHOT-all.jar .**

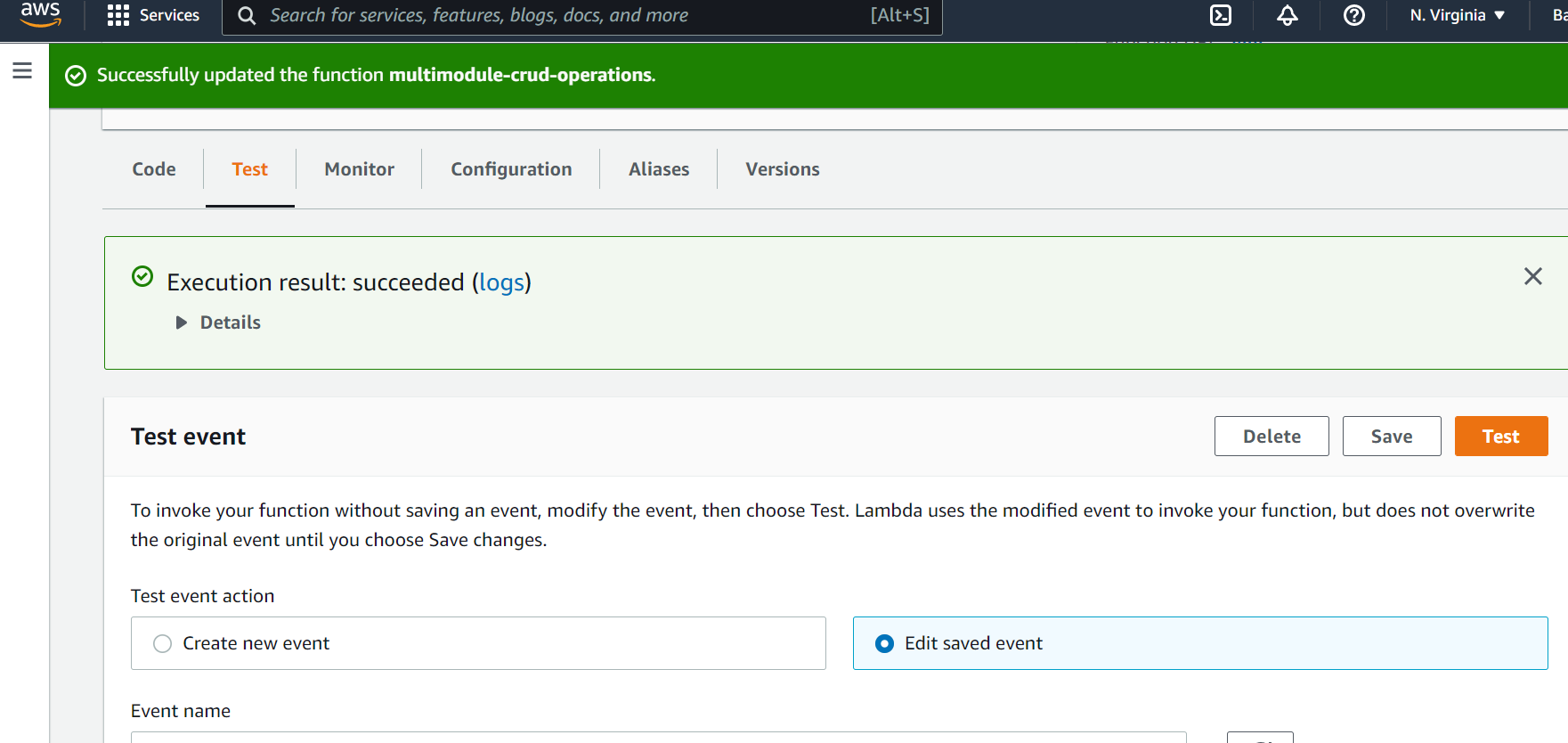
Click on the **Save** Button.



Then below screen will appear



Then Click on **Test** tab.



Scroll Down

**Ex: saveOrder** method, we are passing input as **Json format.**

**Ex:** {

"id": "T110",

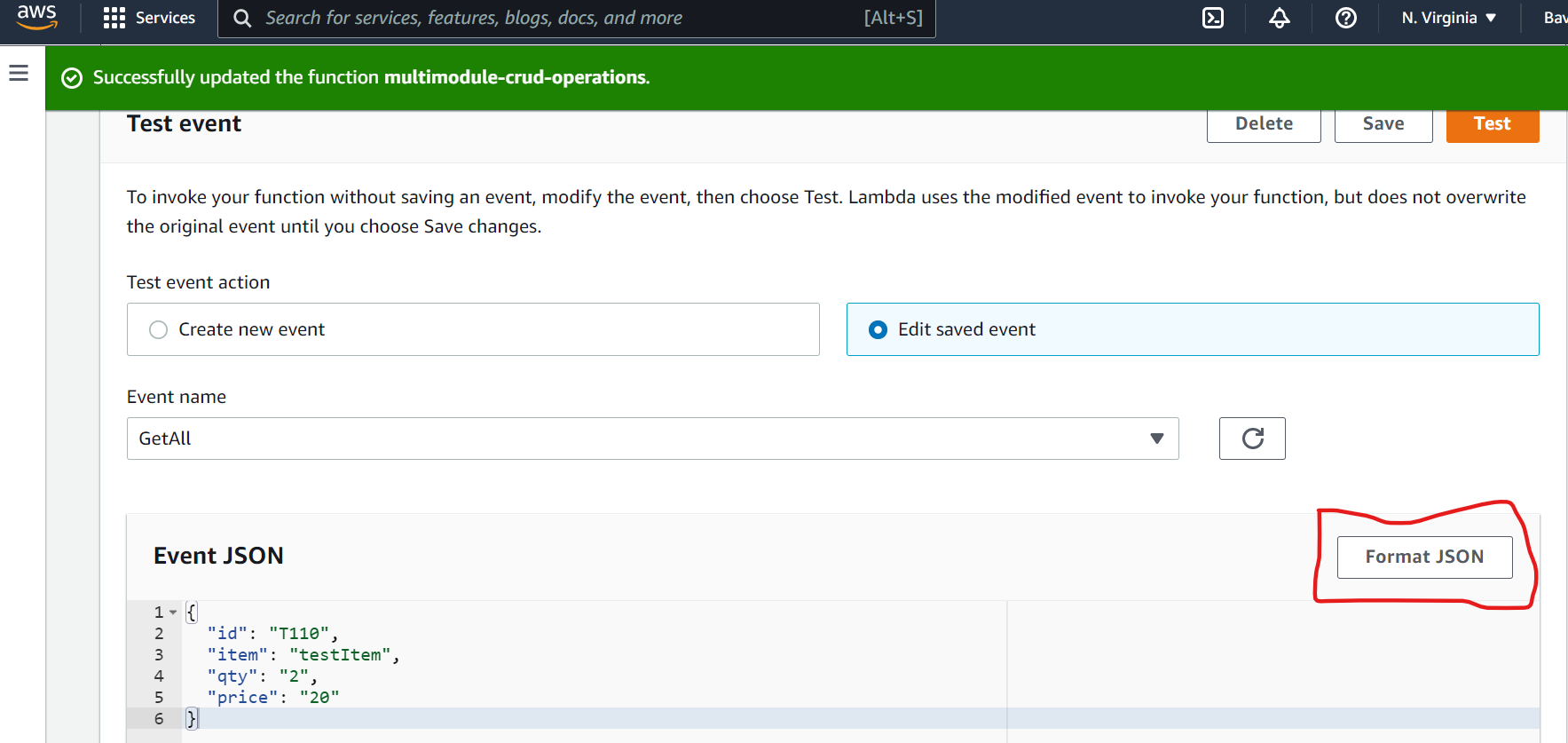
"item": "TestItem",

"qty": "1",

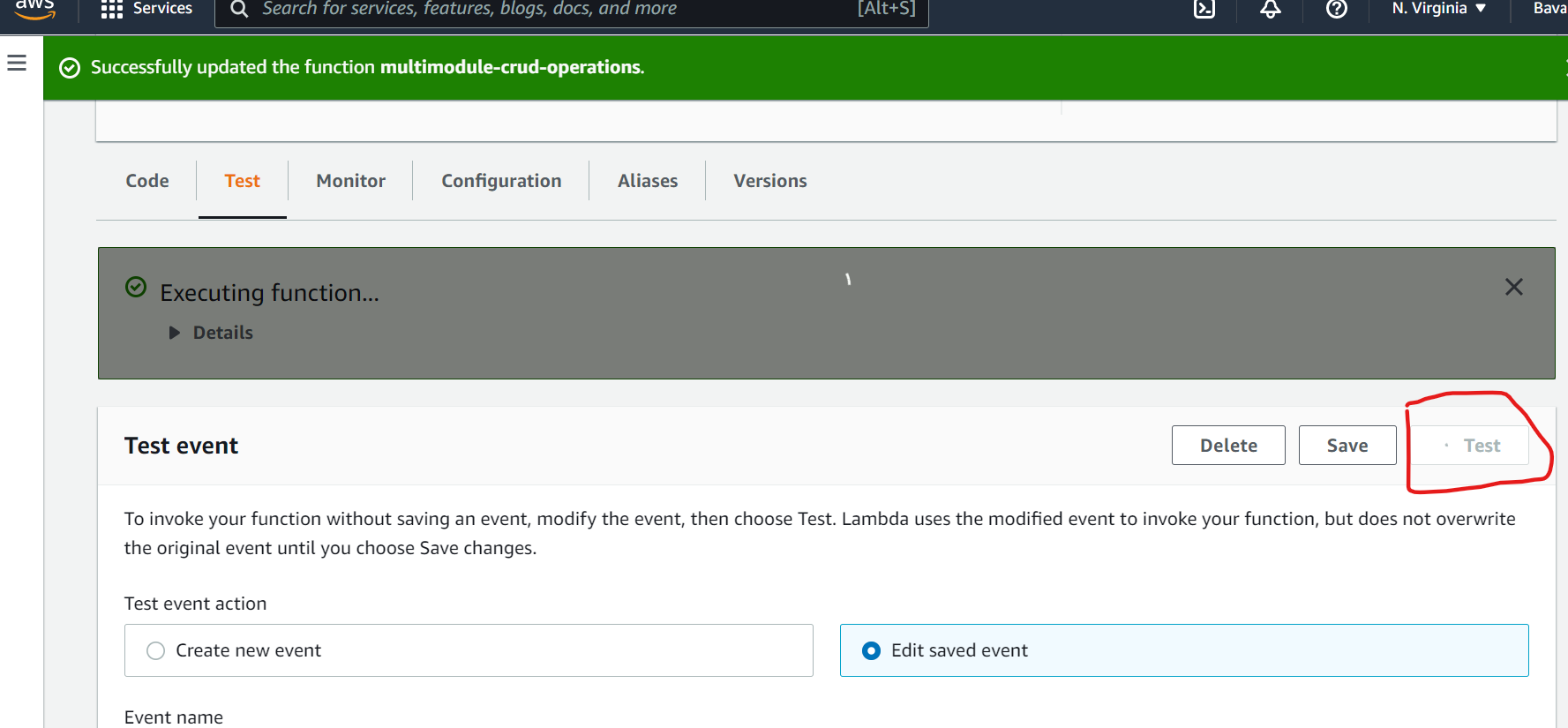
"price": "10.50"

}

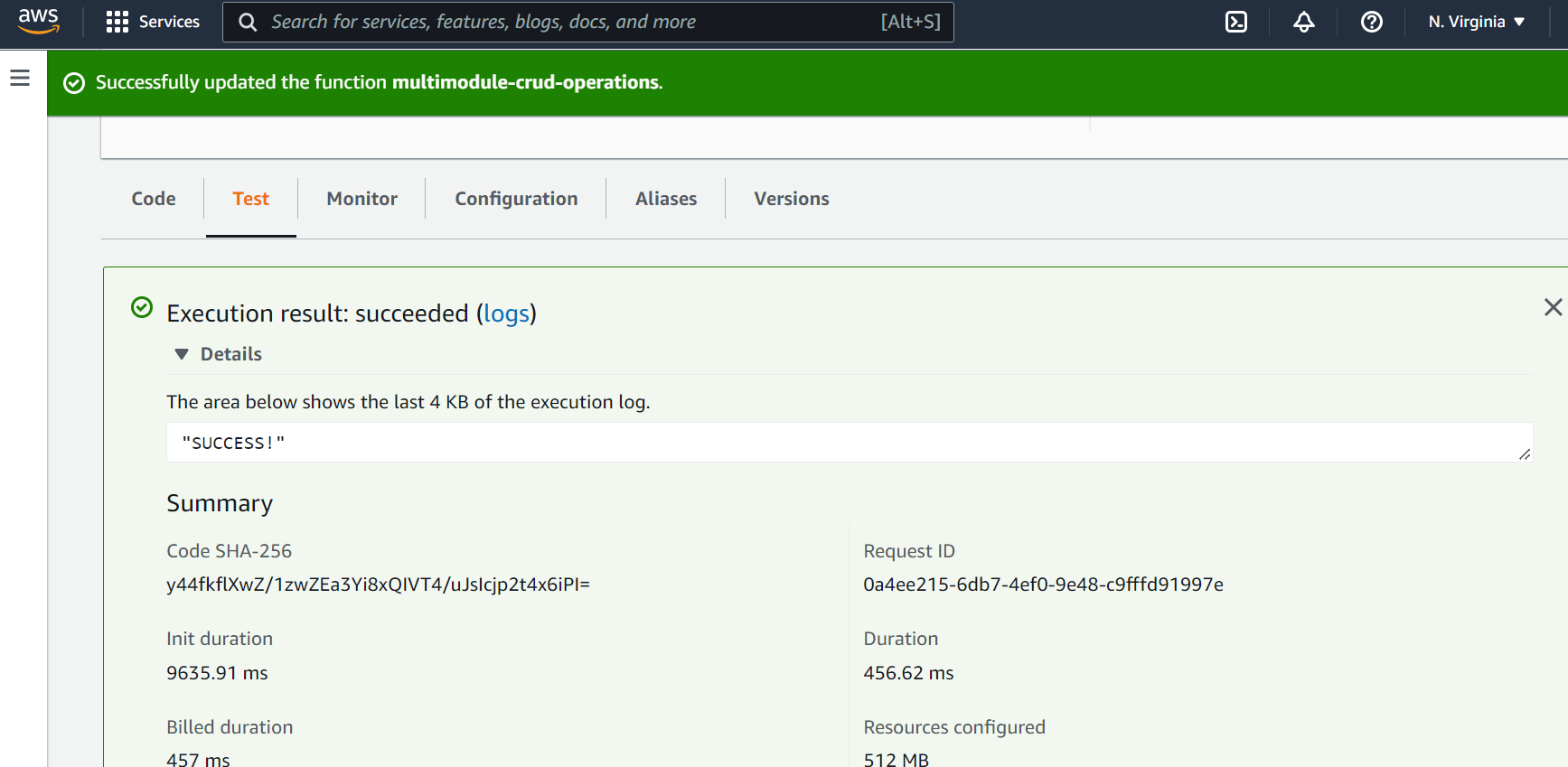
And click on **Format Json** Button.



Then Click on **Test** Button.

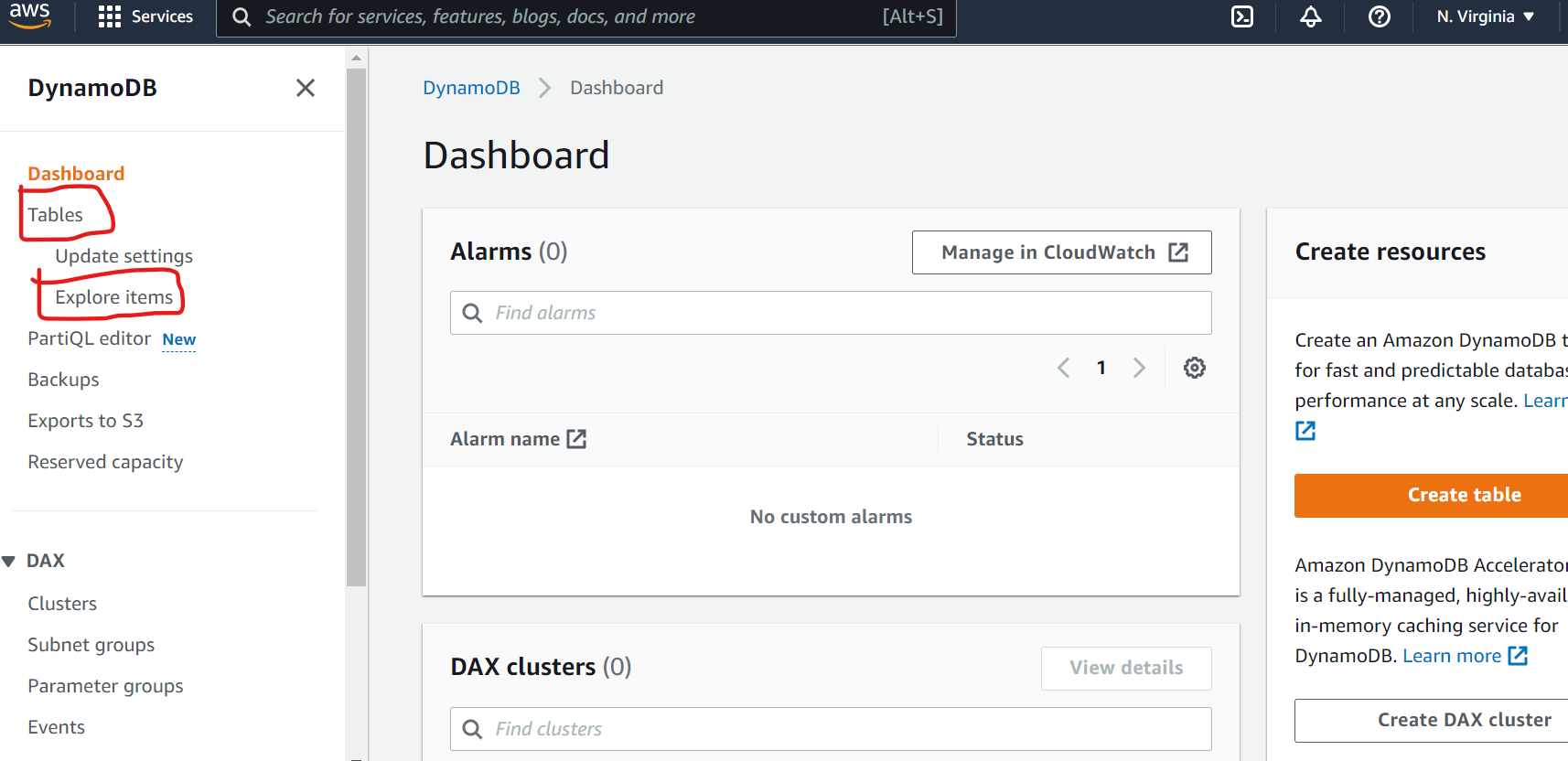


Output:

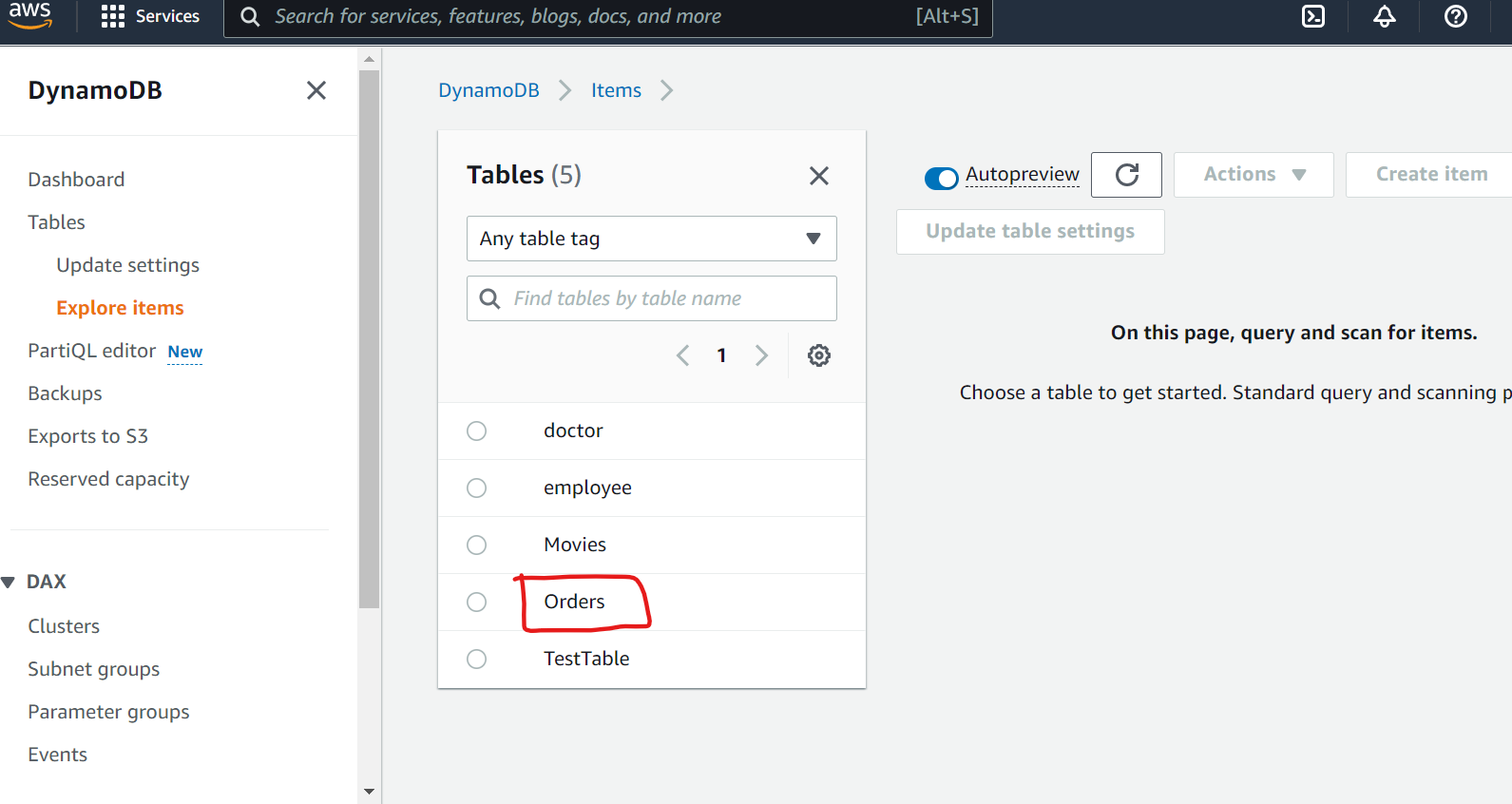


To check whether data is inserted successfully or not.

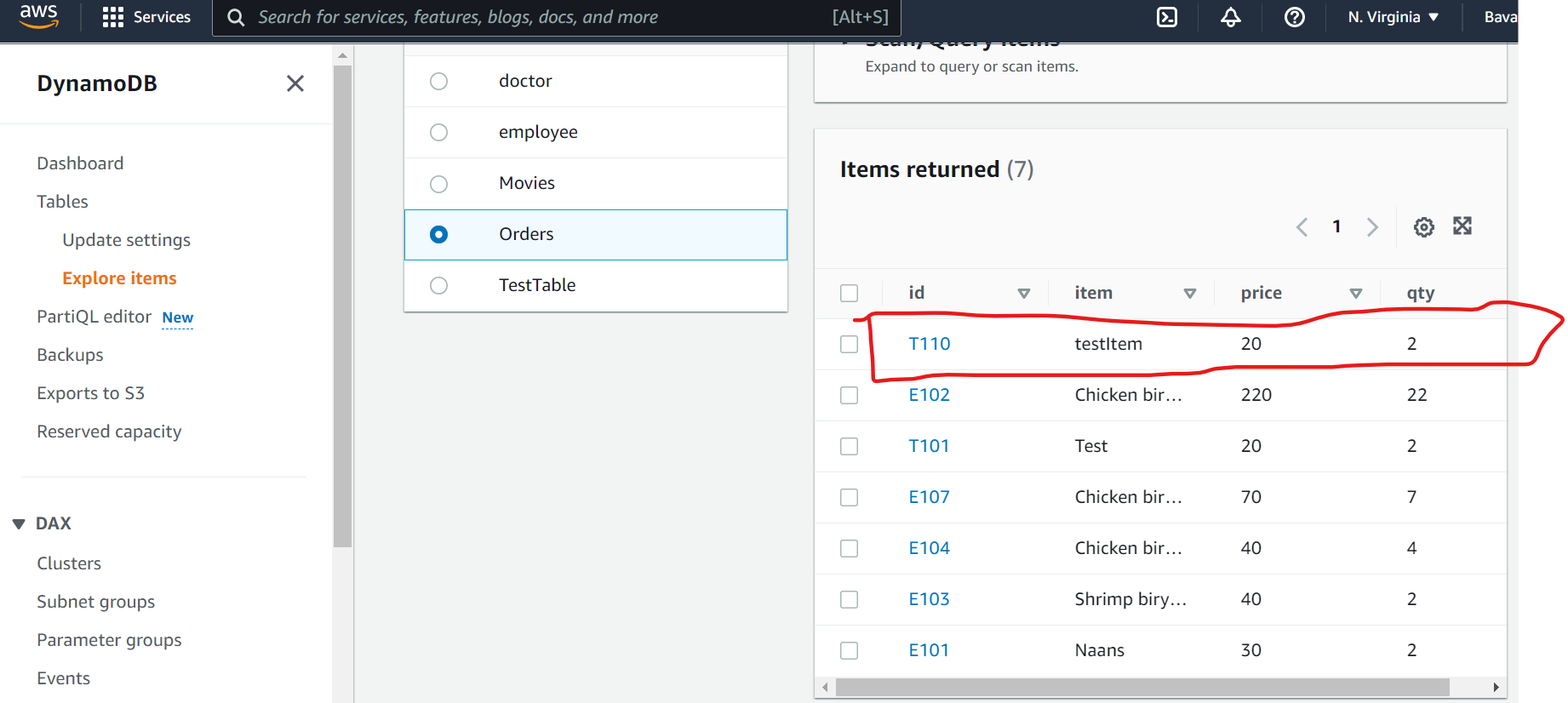
Go To Aws Services-->Dynamodb. Click on **Tables** and click on **Explore items**

.

Then select **Orders** Table



It displays Inserted data as shown below



For updating any record place json data in Event Json. Click on **Test** Button.

# deleteOrder Method invocation in AWS LAMBDA

**In Root Directory ->go to below path**

**spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\src\main\resources\aws-credentials.properties**

**And uncomment deleteOrder method definition like this**

**#spring.cloud.function.definition=getAllOrder**

**#spring.cloud.function.definition=getOrderById**

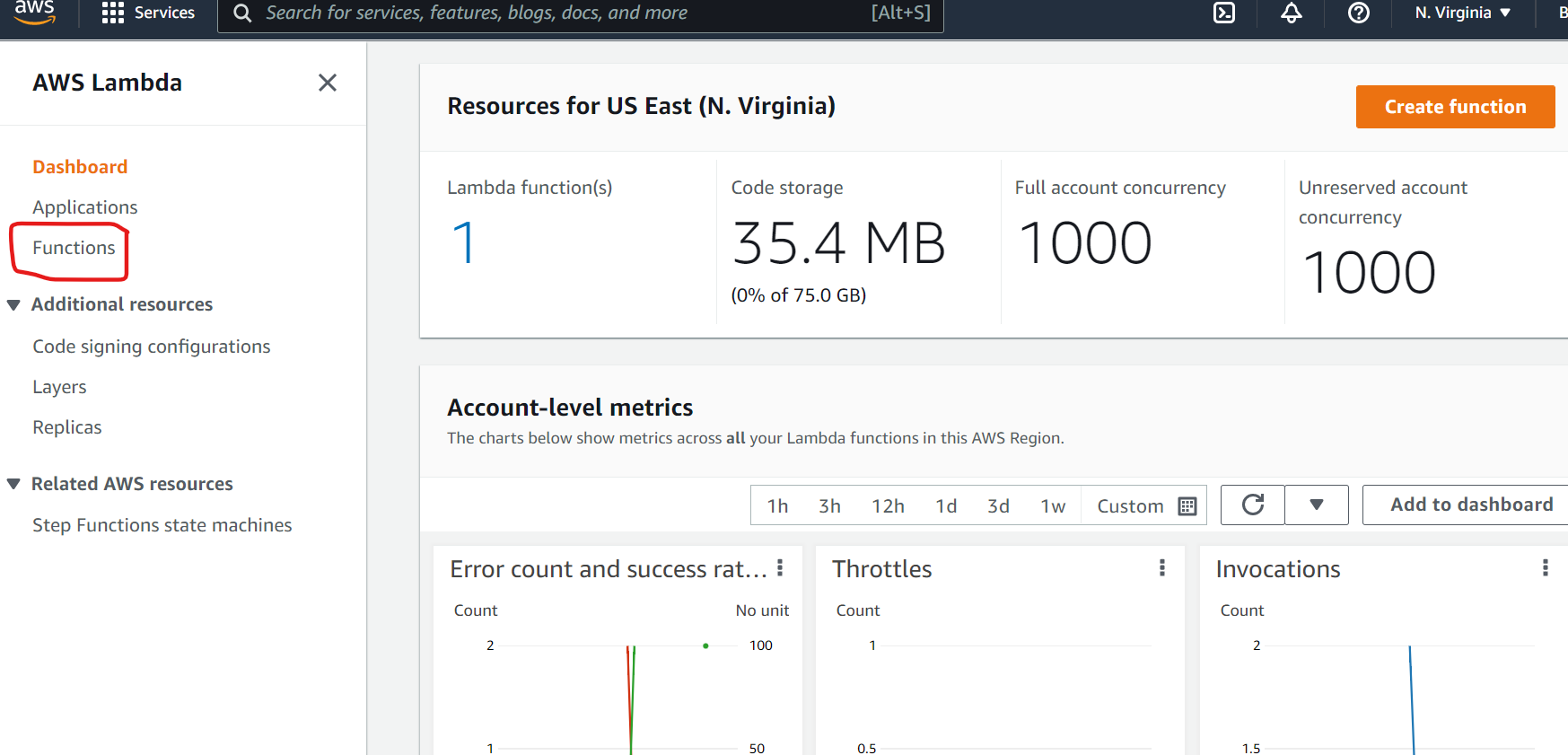
**# spring.cloud.function.definition=saveOrder**

**spring.cloud.function.definition=deleteOrder**

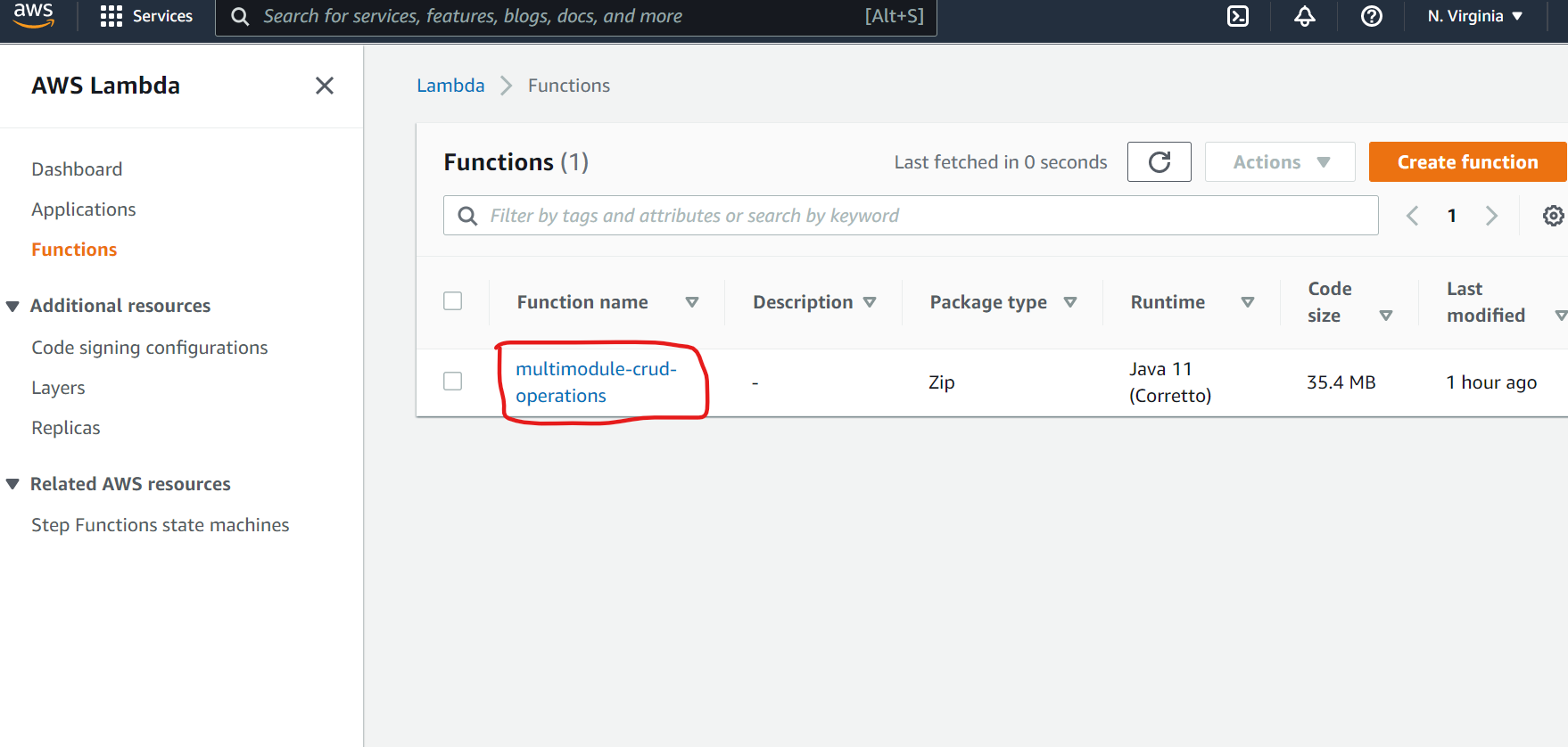
**Then use the commands below to clean and build spring boot application.**

* **./gradlew clean**
* **./gradlew build**

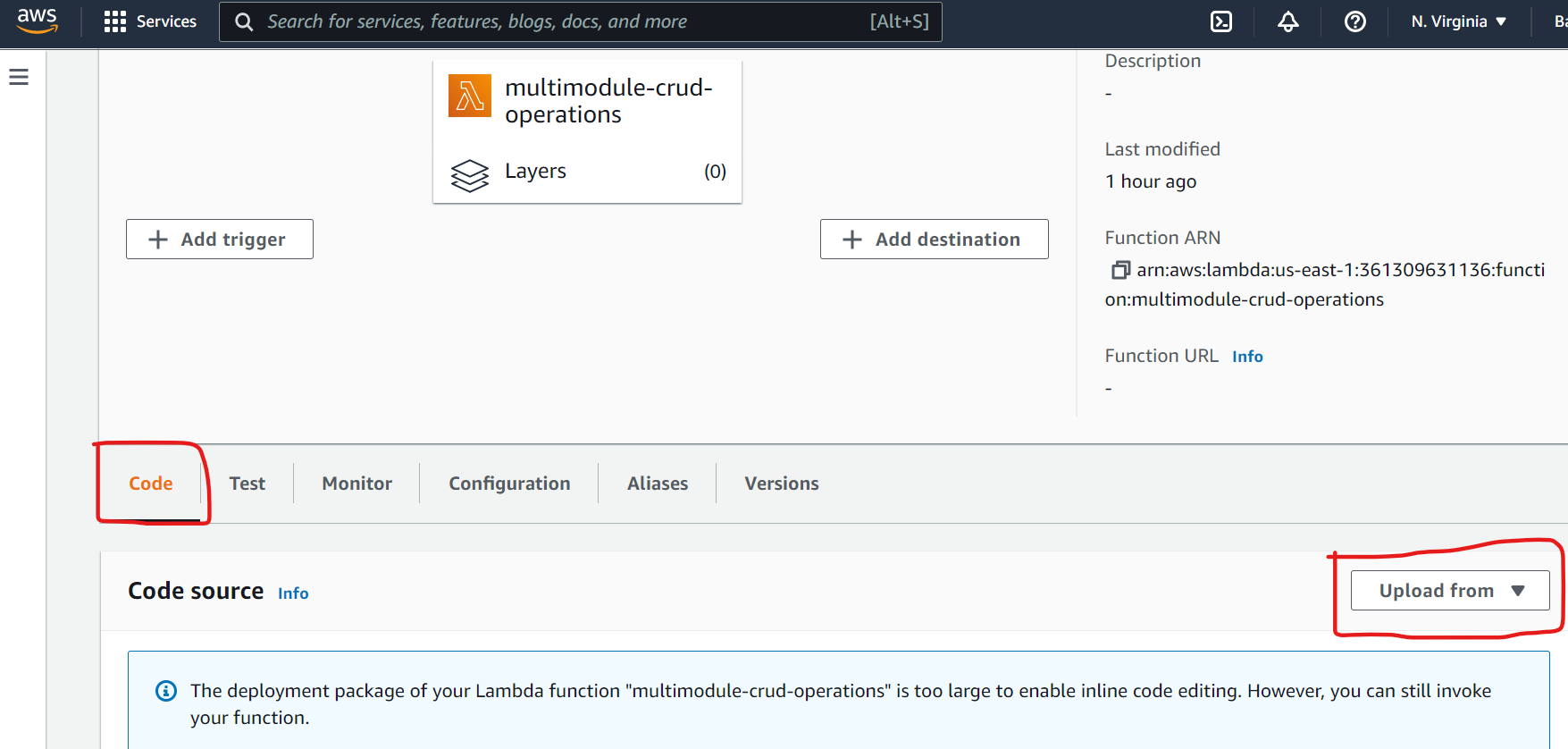
Go to Aws Services->Select Lambda->Click on **Functions**.



Then below page will appear. Click on **multimodule-crud-operations**

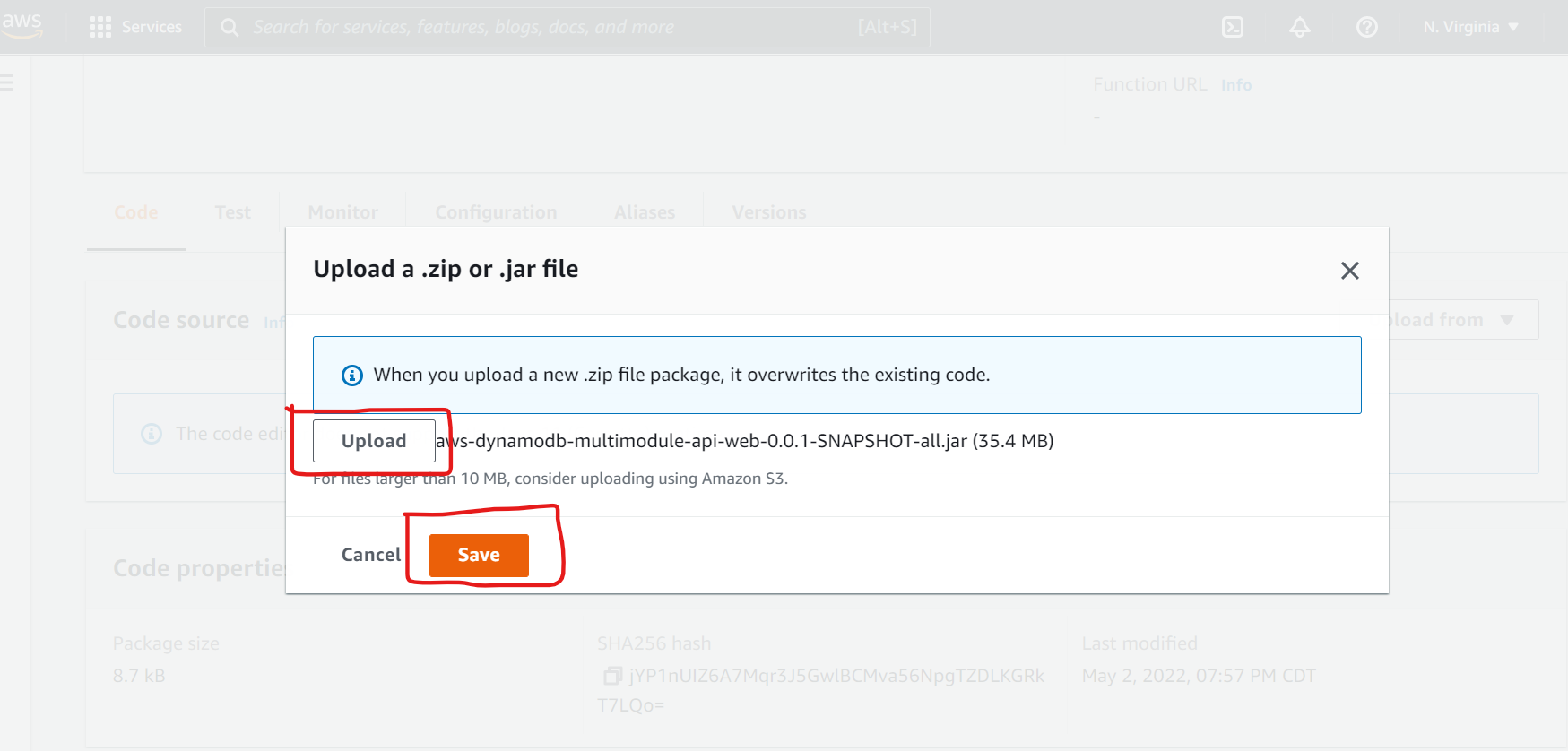


This below screen will appear. Then Click on **Code. Select upload from**

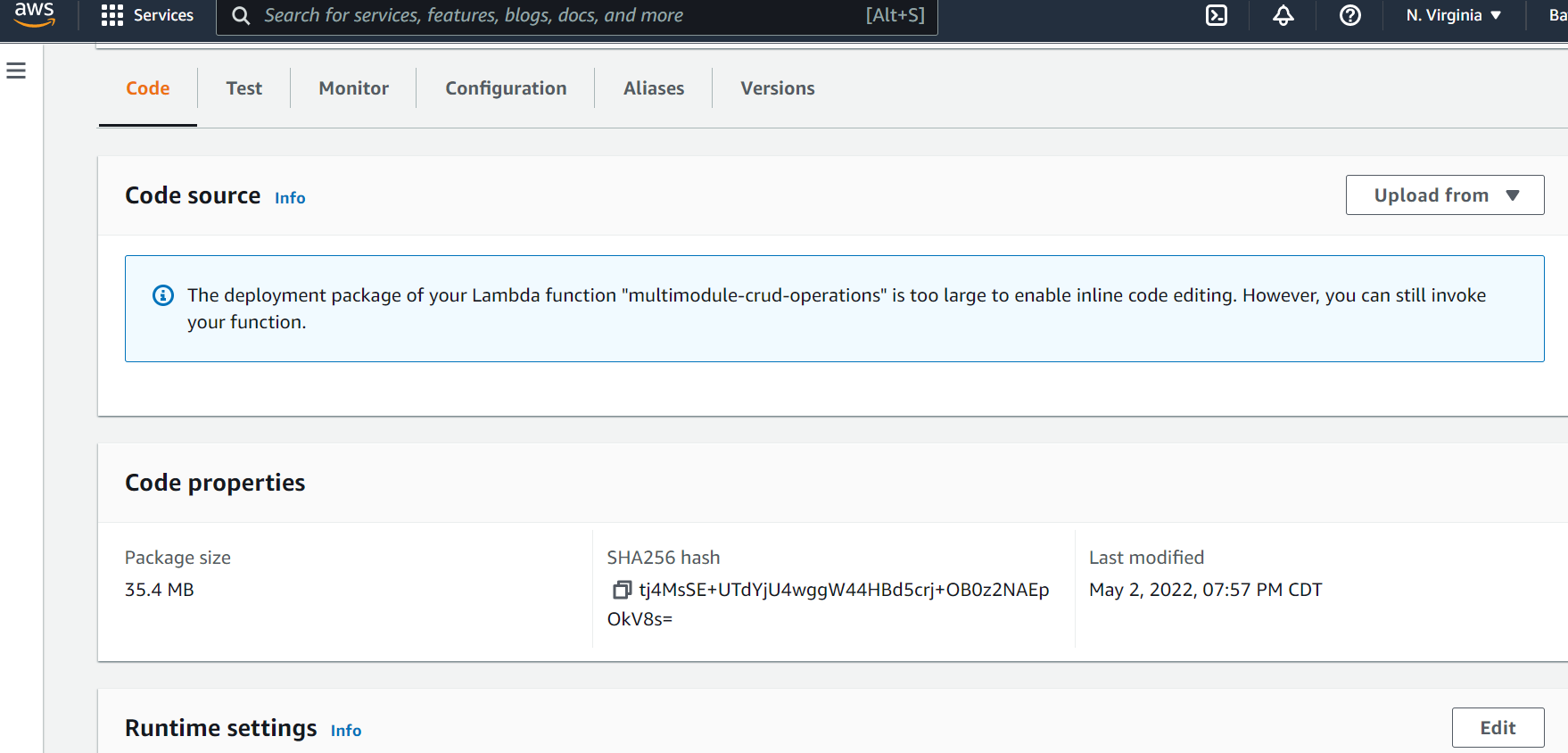


Then the screen below will appear. Click on **Upload from** Button and Select jar file from Your **Folder Path\spring-dynamodb-lambda-multimodule\aws-dynamodb-multimodule-api-web\build\libs\aws-dynamodb-multimodule-api-web-0.0.1-SNAPSHOT-all.jar .**

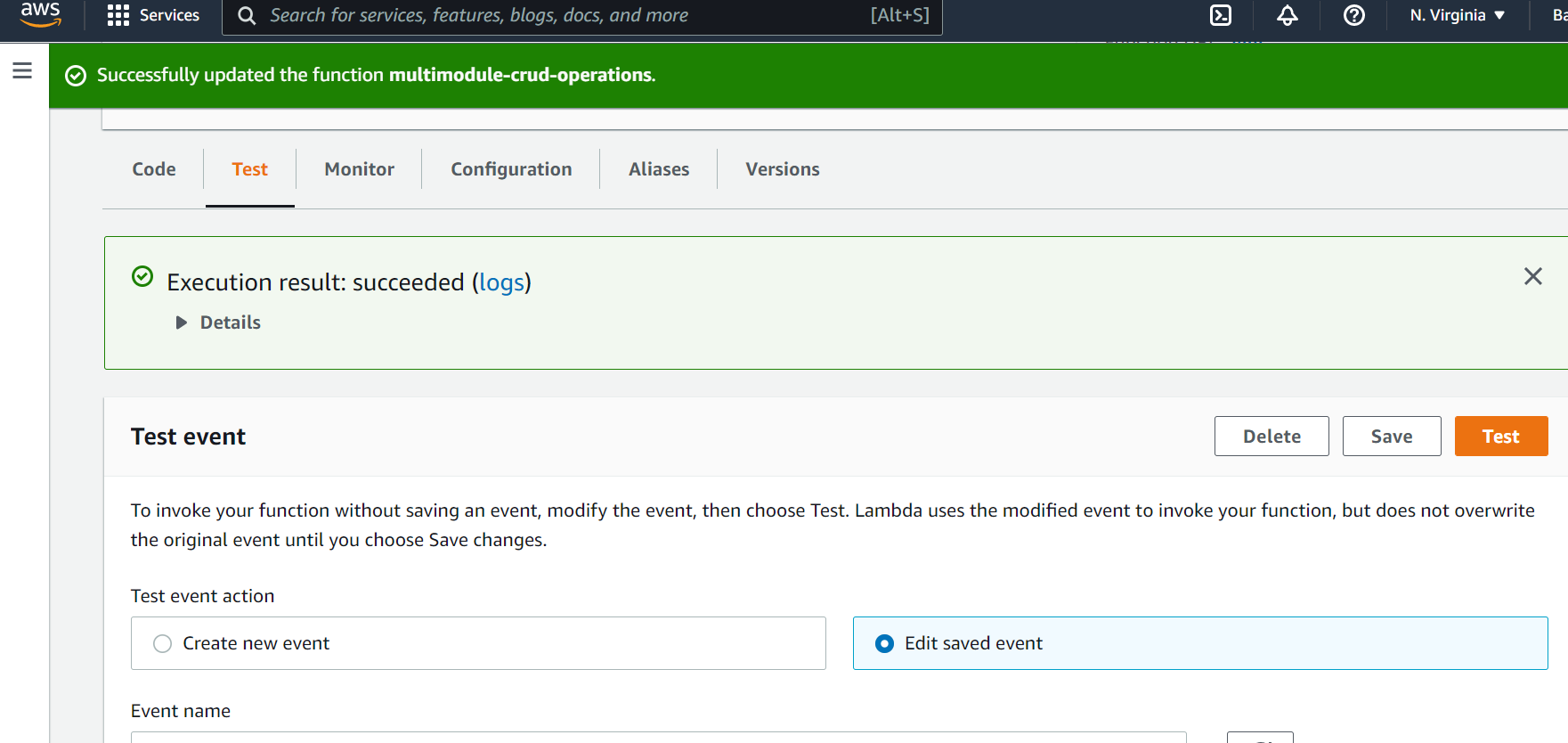
Click on the **Save** Button.



Then below screen will appear

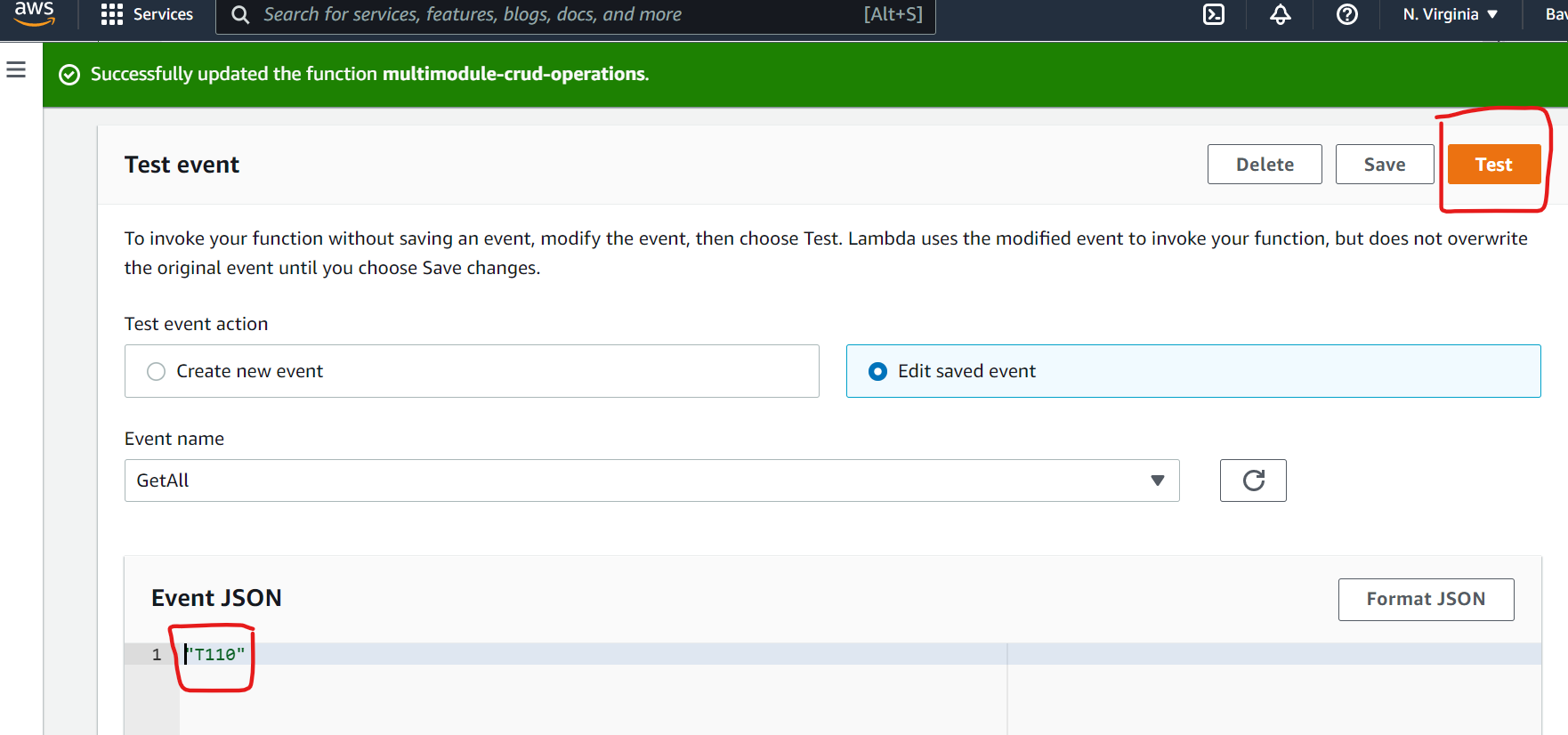


Then Click on **Test** tab.



Scroll Down

Ex: **deleteOrder** method, we are passing input as **“T110”** inside **Event Json** Section. Then, Click on **Test** Button.



**Output**:

