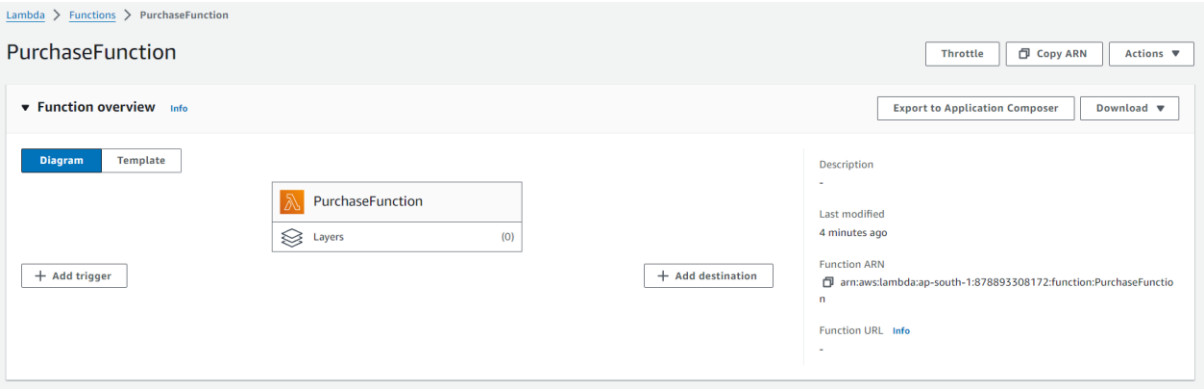
😊 Create Step Functions State Machine Using AWS Lambda

1. So, this lambda is well-suited for task states because lambda functions are serverless and easy to write. We can write code in the mission console or favourite editor a place, handle the details, or provide competing for our function and run it.
2. Now in your AWS Console go to Lambda and create two functions.
3. Your first function is **PurchaseFunction** you must choose Python 3.11 for this and just create your lambda function.

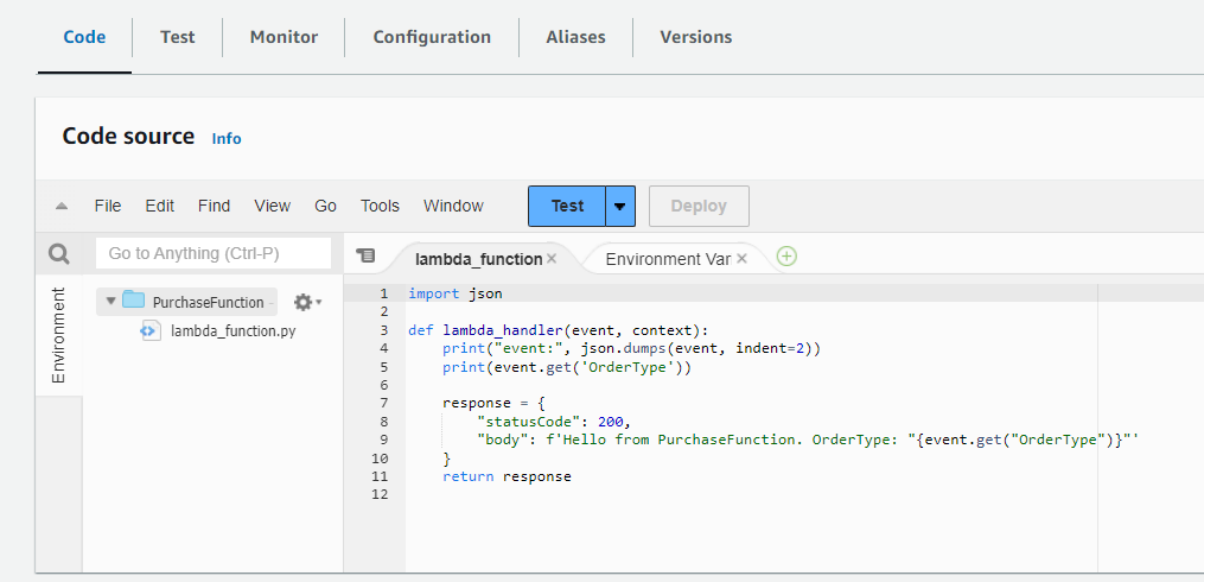


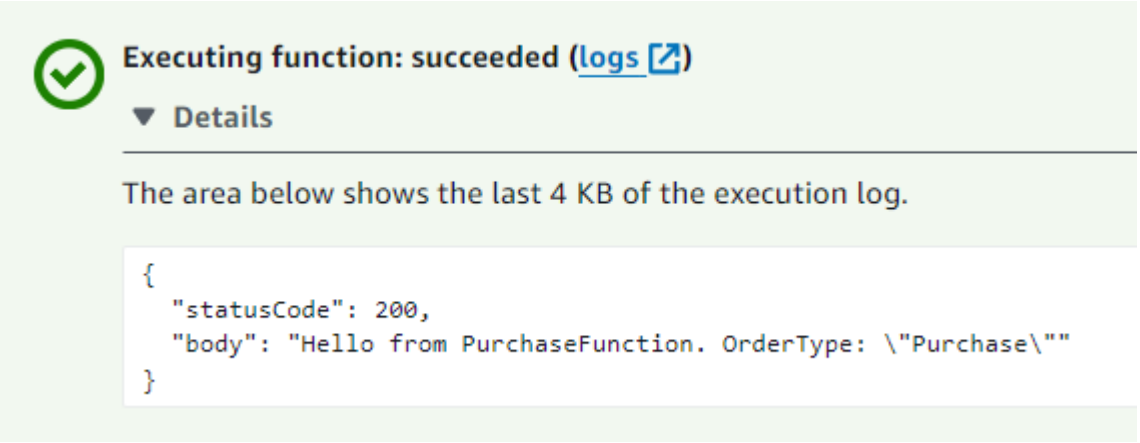
4. And you must paste the code given below.

```
import json

def lambda_handler(event, context):
    print("event:", json.dumps(event, indent=2))
    print(event.get('OrderType'))

    response = {
        "statusCode": 200,
        "body": f'Hello from PurchaseFunction. OrderType: "{event.get("OrderType")}"'
    }
    return response
```

5. Also, you can test the code using the below statement.

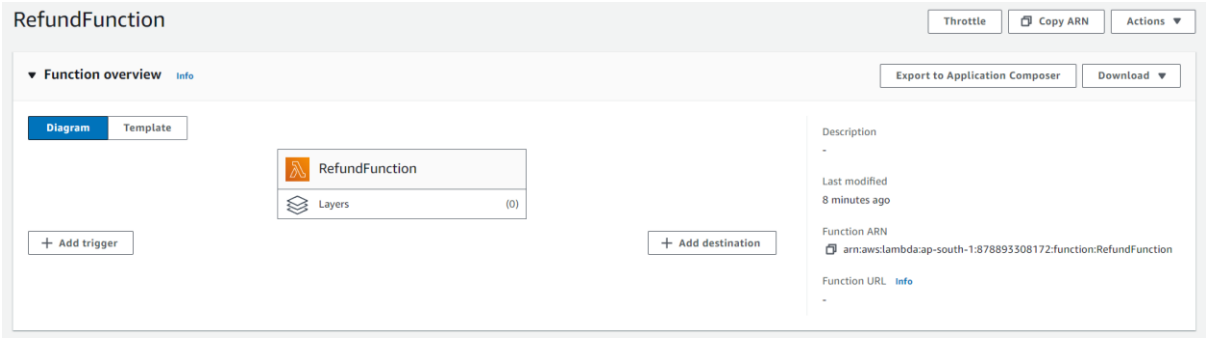
```
{
  "OrderType": "Purchase"
}
```

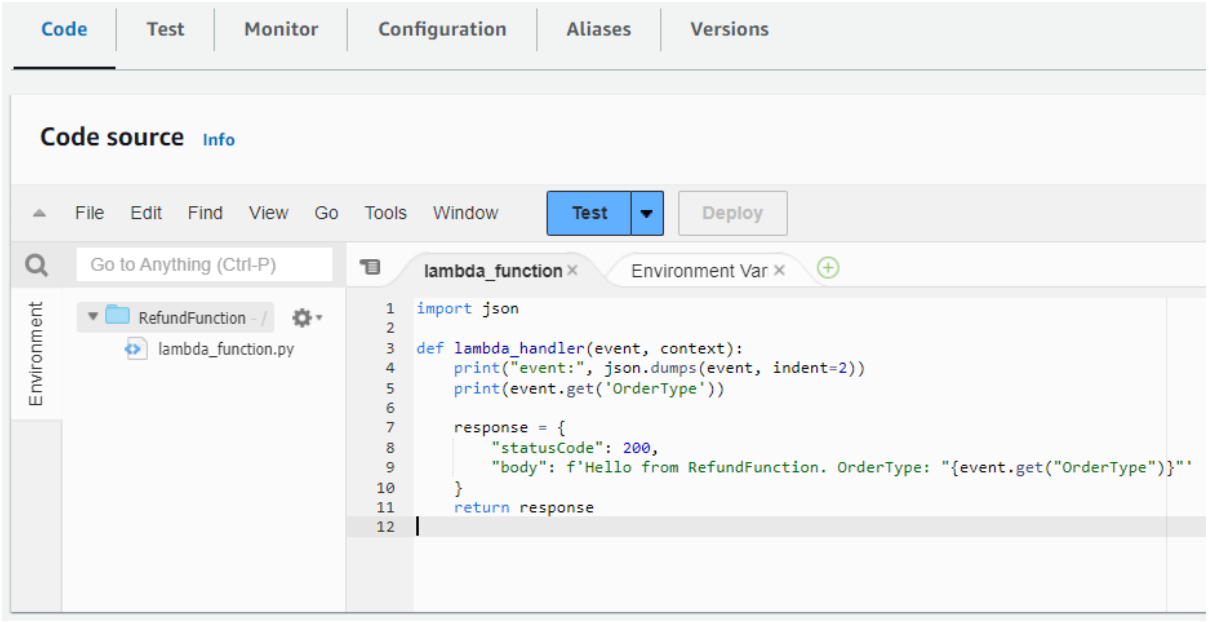




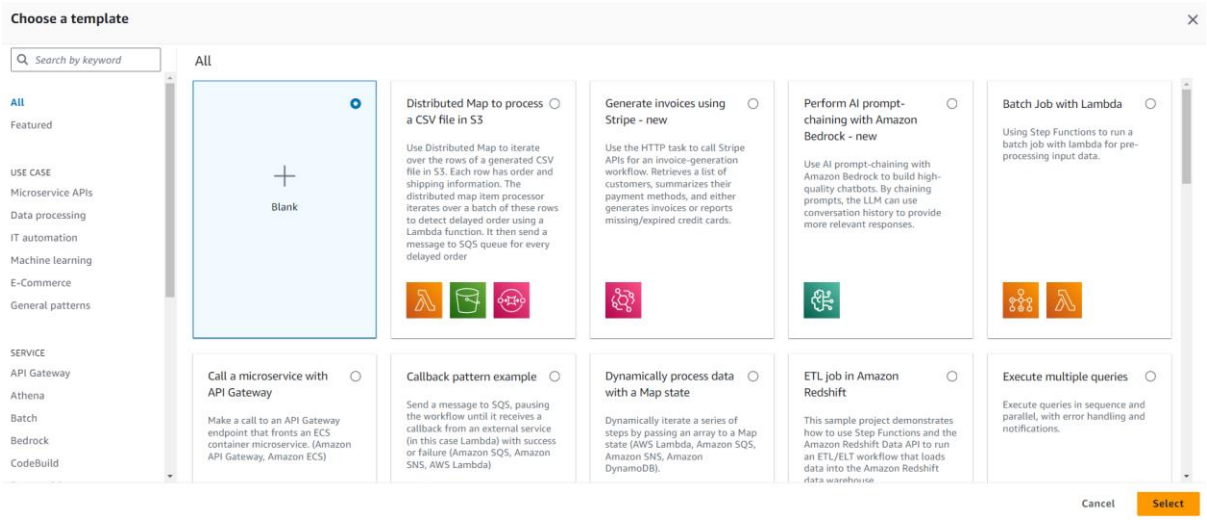
6. After that you must create another function **RefundFunction** using the same Python 3.11 as your runtime and just create your function.
7. Then you must add the code given below.

```
import json

def lambda_handler(event, context):
    print("event:", json.dumps(event, indent=2))
    print(event.get('OrderType'))

    response = {
        "statusCode": 200,
        "body": f'Hello from RefundFunction. OrderType: "{event.get("OrderType")}"'
    }
    return response
```

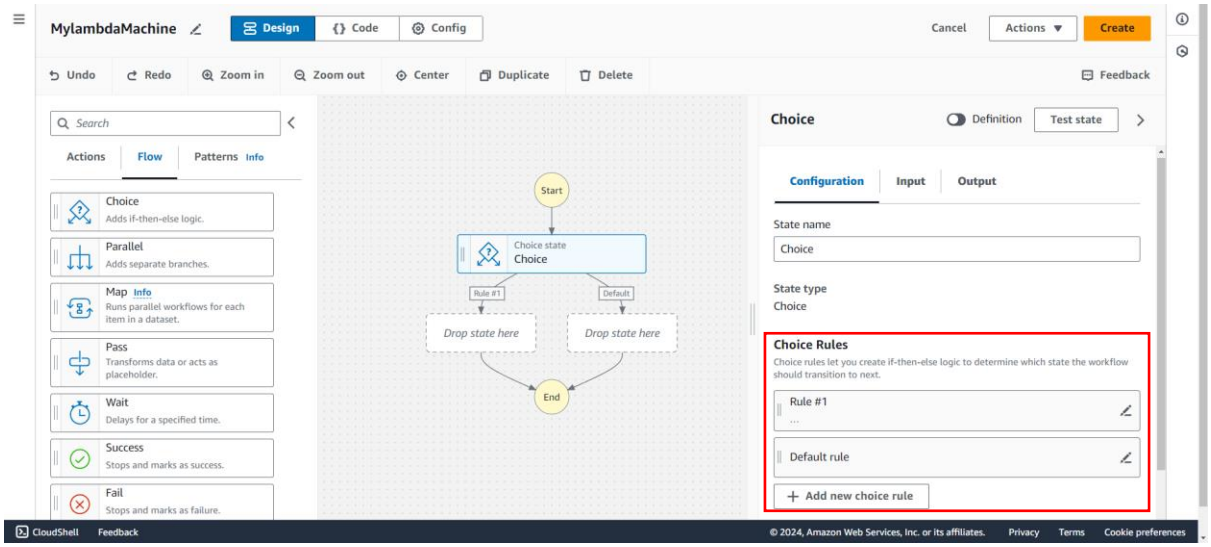




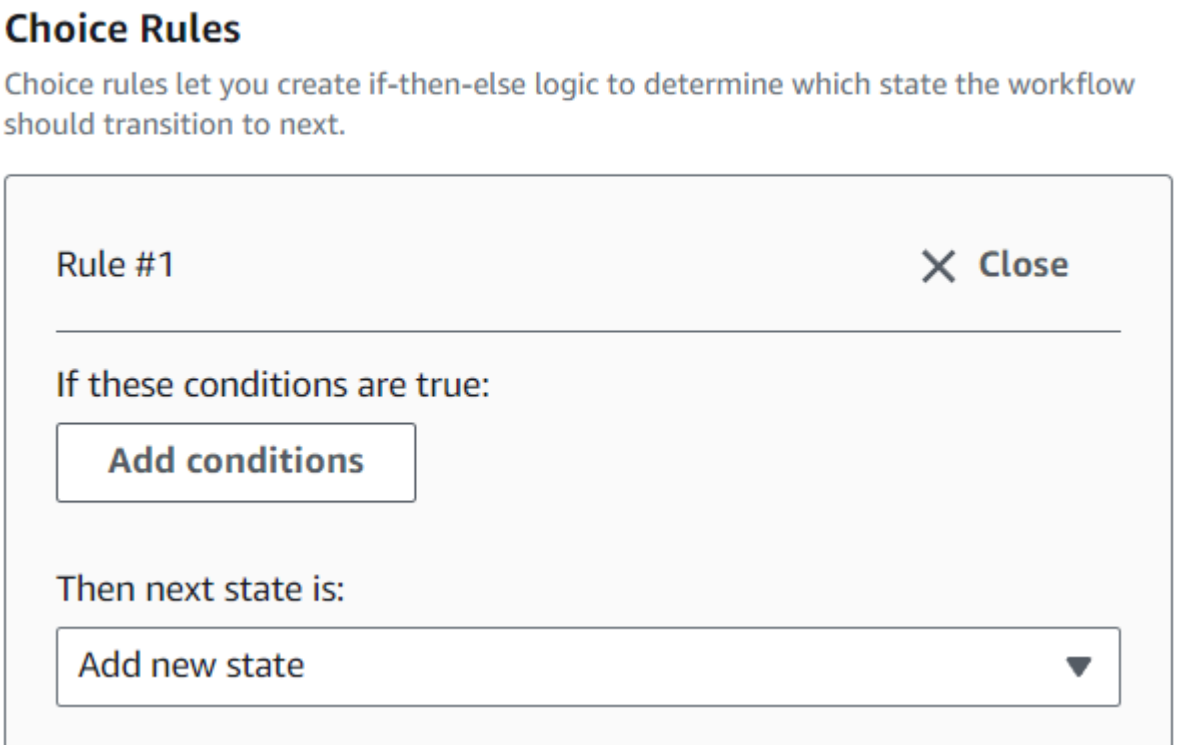
8. Now go to Step Function create a new machine choose the blank template to start with and click on Select.

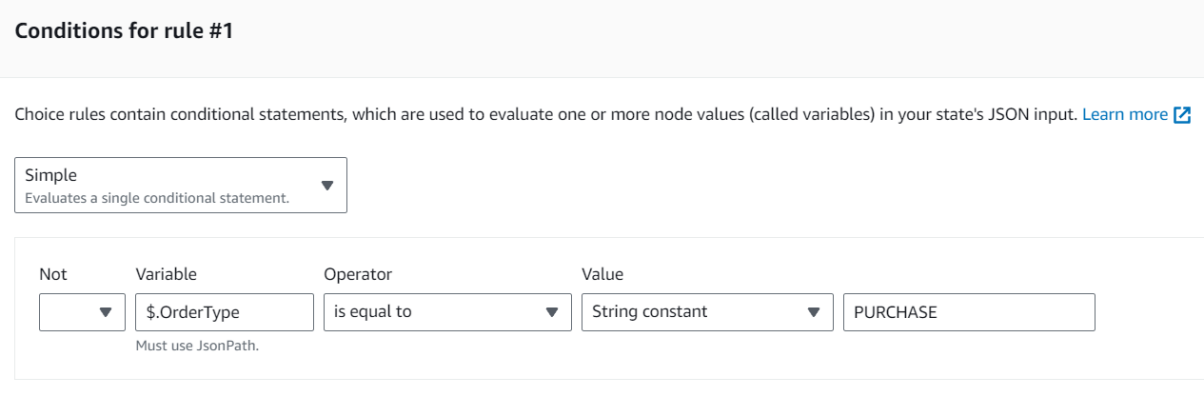


9. Now here from the flow choose choice step and drag it to middle and now you are going to create two rules. Ignore the default rule.

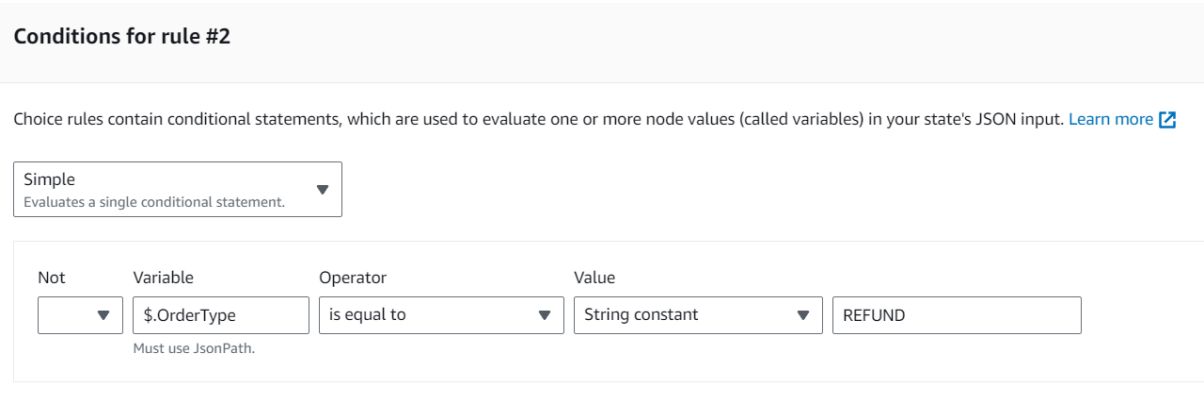


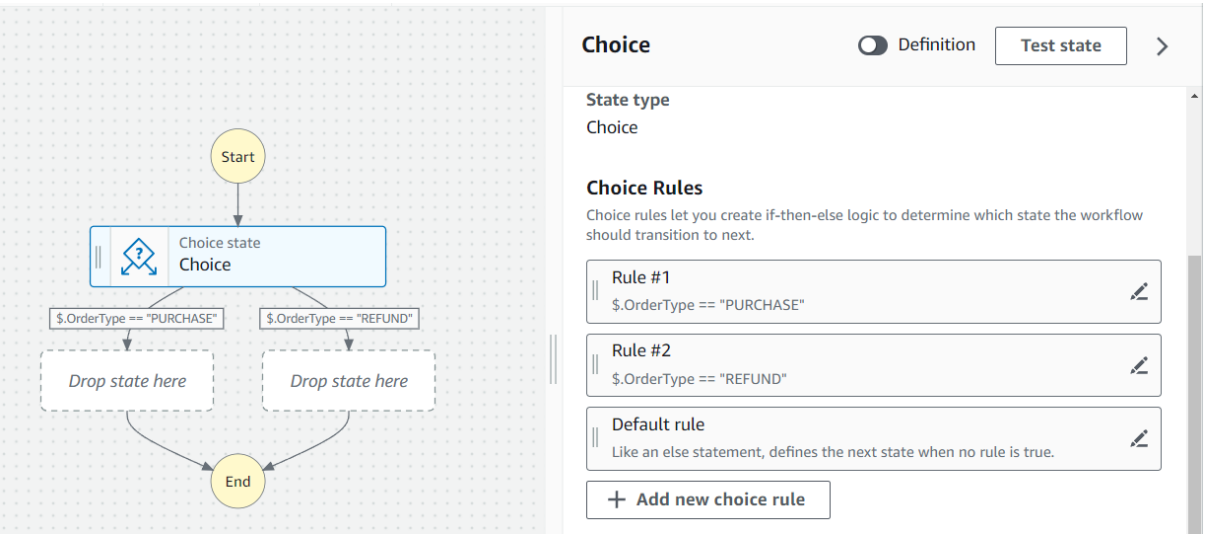
10. Choose rule 1 and click on add condition. Then you must choose the same conditions as you can see in the snapshot.



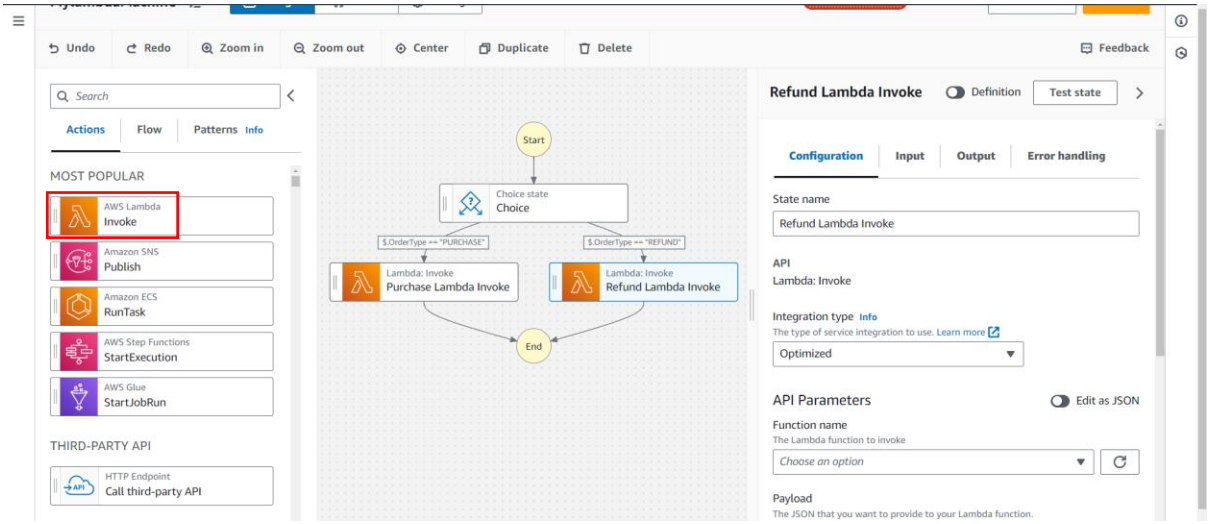


11. Then in a similar way you have to create rule 2 but for refund and save it. Also, you can see the diagram.

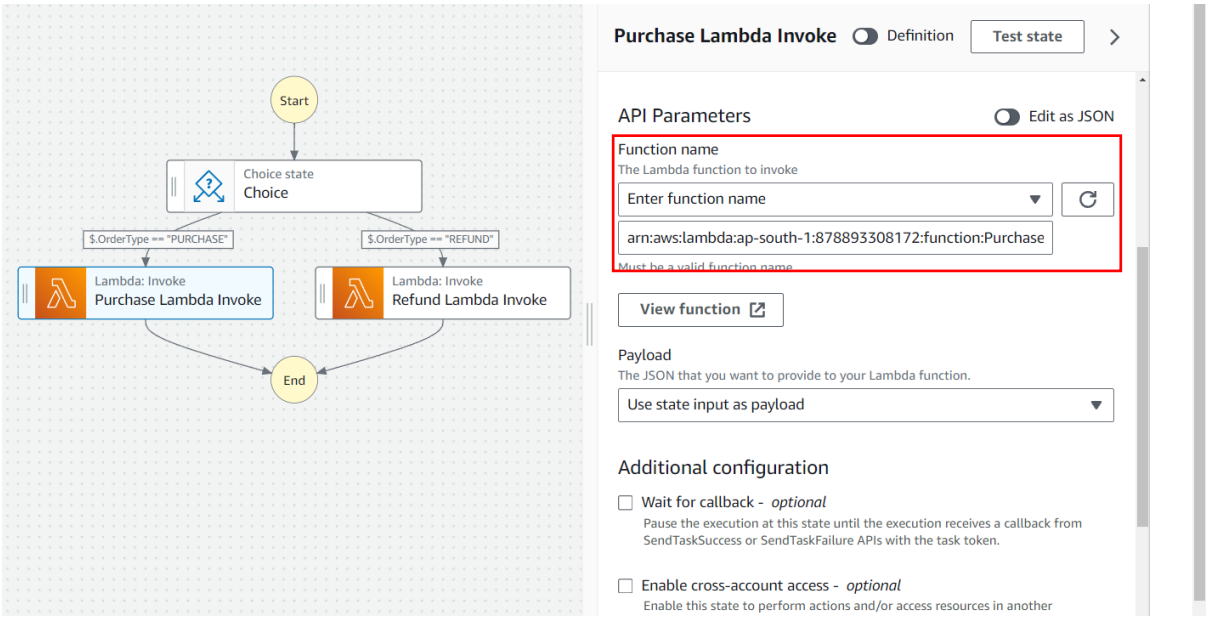


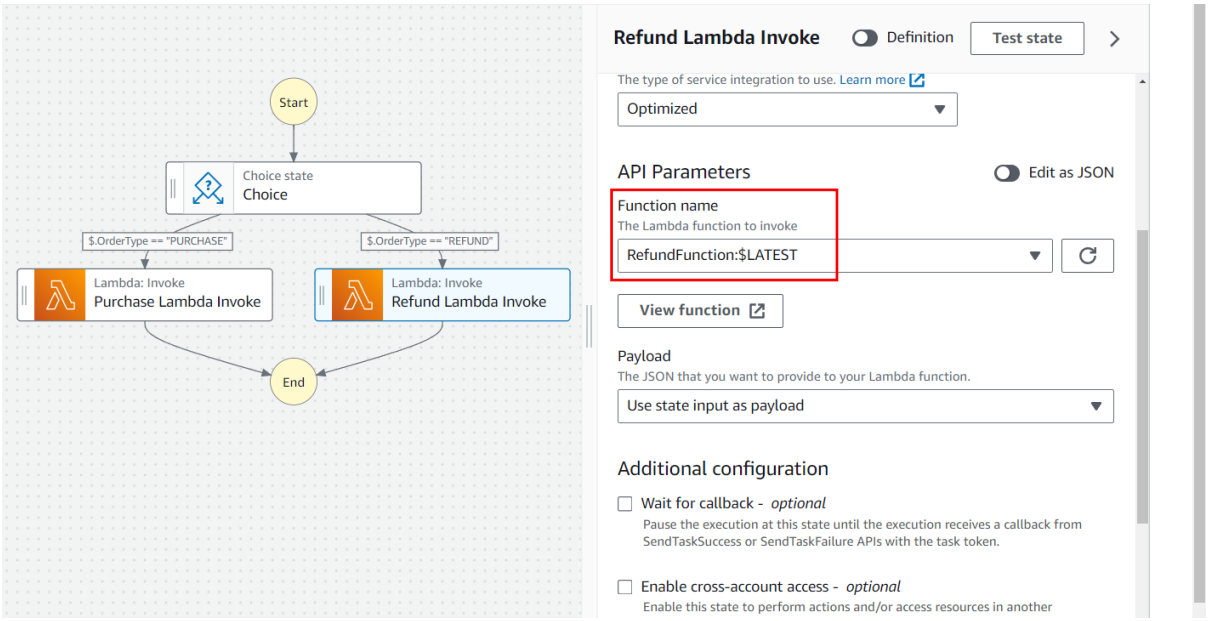


12. Now you need go to actions, choose AWS Lambda Invoke step, drag and drop it under choice, then just change their name. To change their name, you have to click on lambda invoke and from the configuration you can change the name.

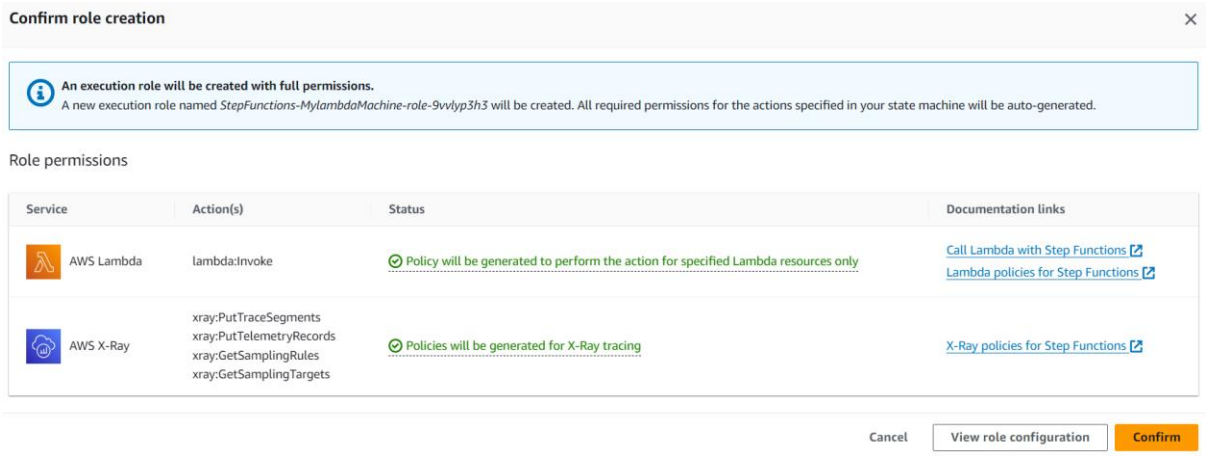


13. Now go to one of the lambda functions choose it and then in the function name choose Purchase lambda. Similarly, go to another function and choose the Refund function.

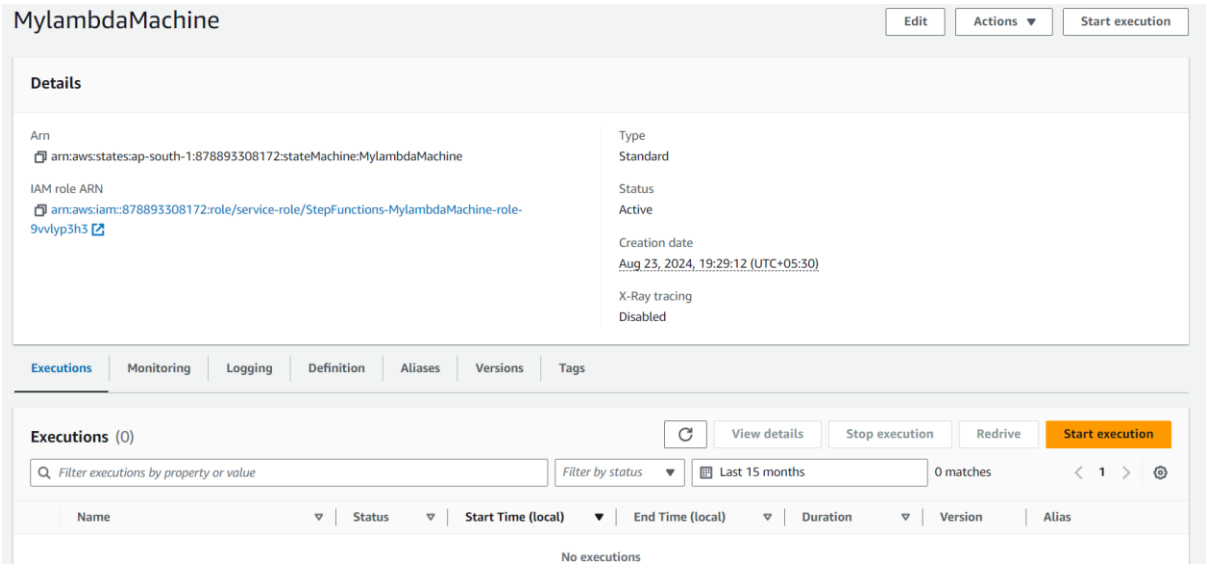




14. Now click on create and your step machine will be created. Just confirm the roles.

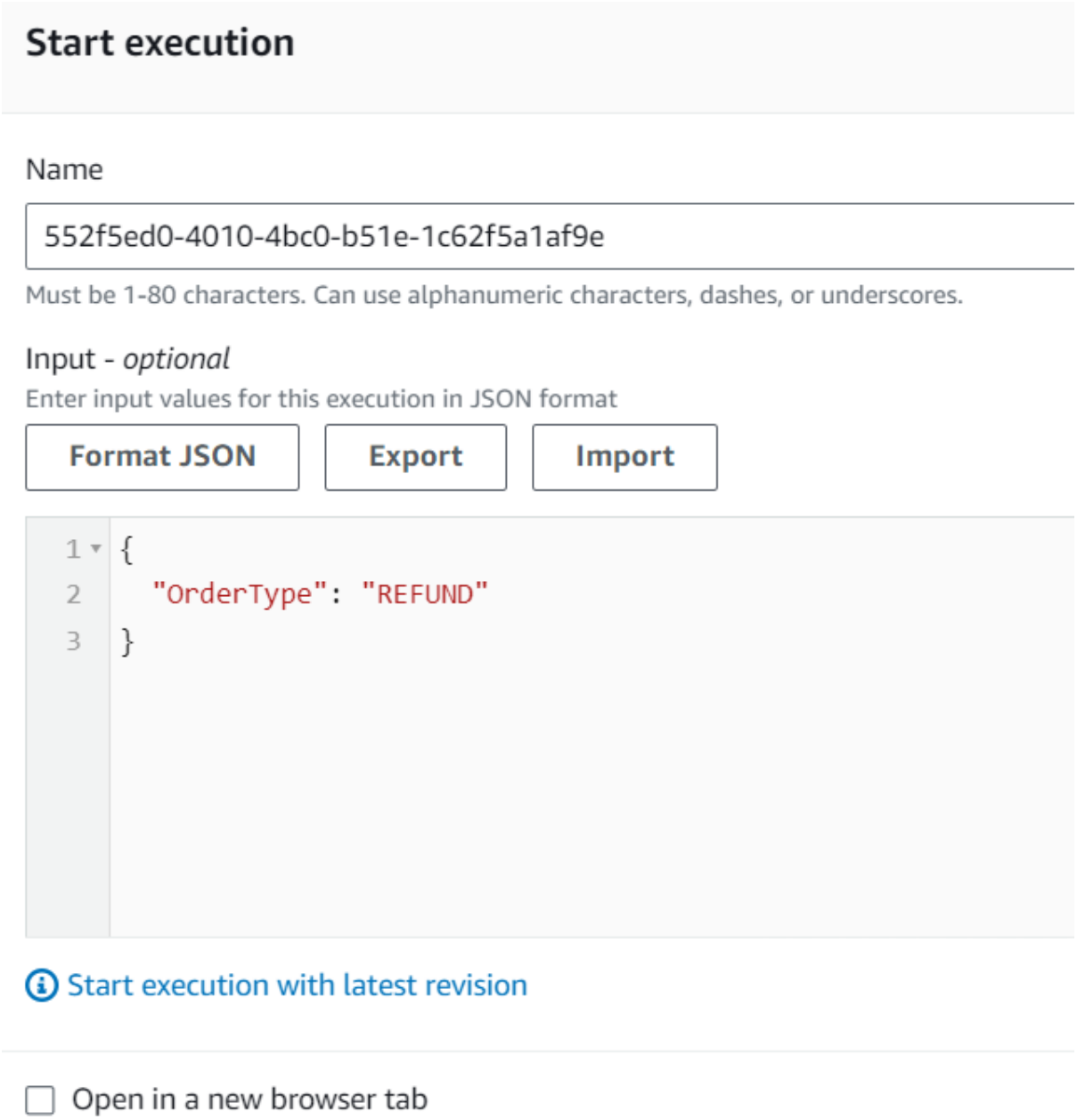


15. Then click to start your execution.

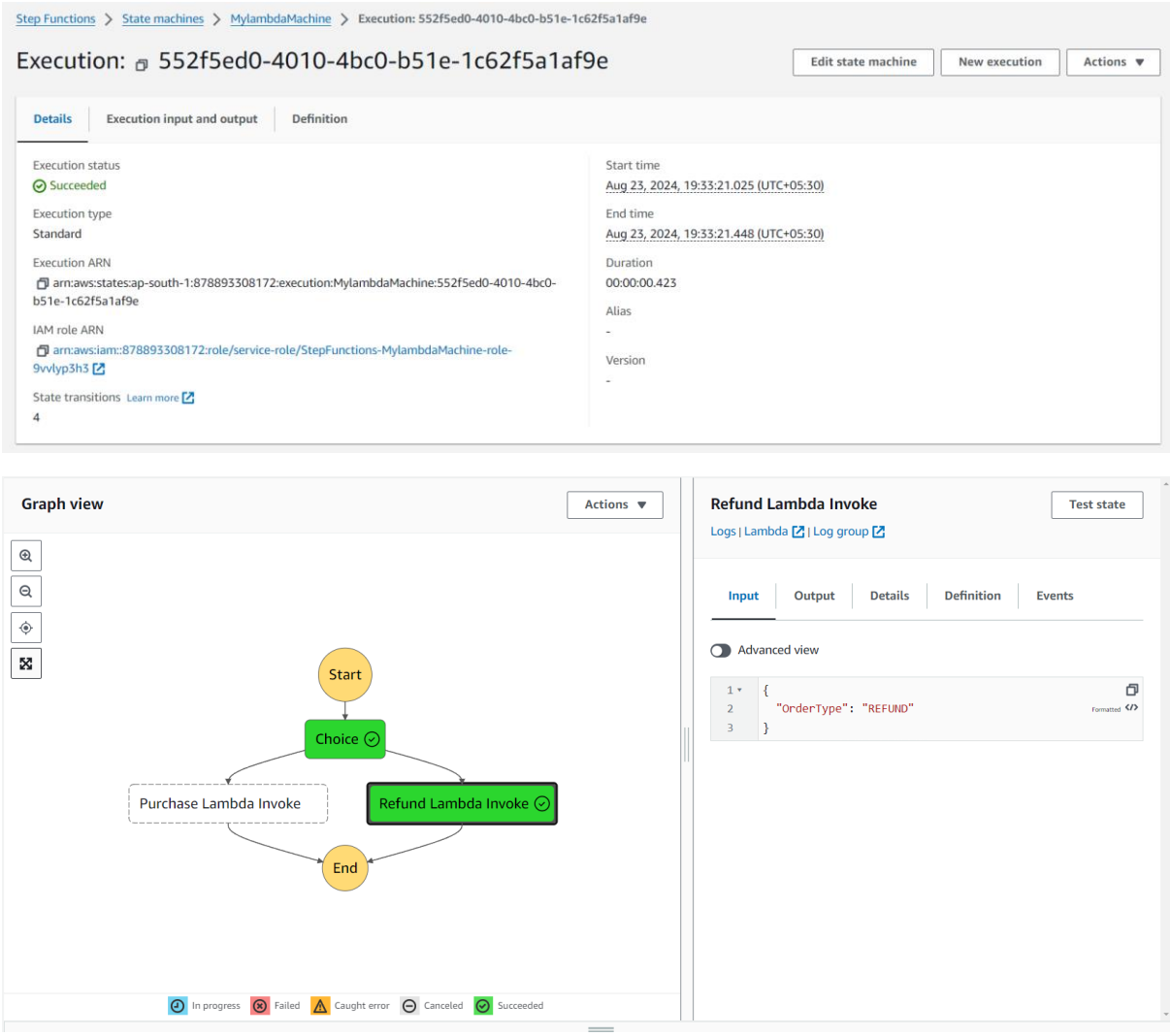


16. Now it depends on which function you want to execute, so you need to pass the statement for that. As you can see below is the Refund statement so the refund lambda function will be called.

```
{
  "OrderType": "Refund"
}
```



17. Below you can see that the execution succeeded.



18. Similarly, for Purchase. Click on new Execution and give the purchase statement.

Start execution

Name

ae9f07a3-650c-4012-a916-693a2ea645df

Must be 1-80 characters. Can use alphanumeric characters, dashes, or underscores.

Input - *optional*

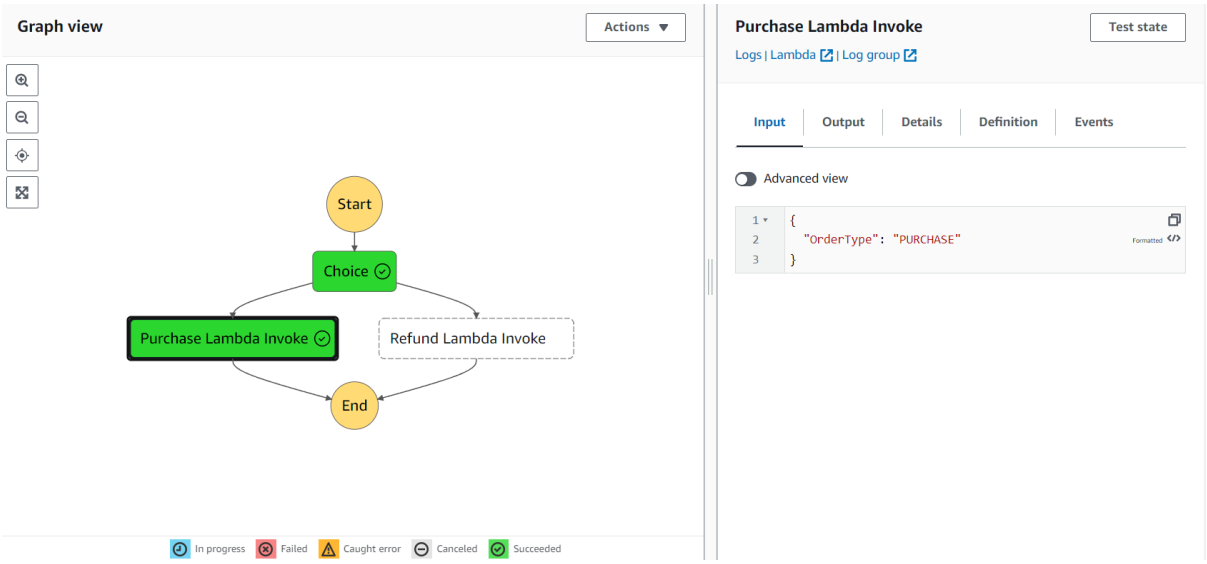
Enter input values for this execution in JSON format

Format JSON | Export | Import

```
1 {
2   "OrderType": "PURCHASE"
3 }
```

Start execution with latest revision

Open in a new browser tab



19. Once you are done then delete your lambda function and the step functions.