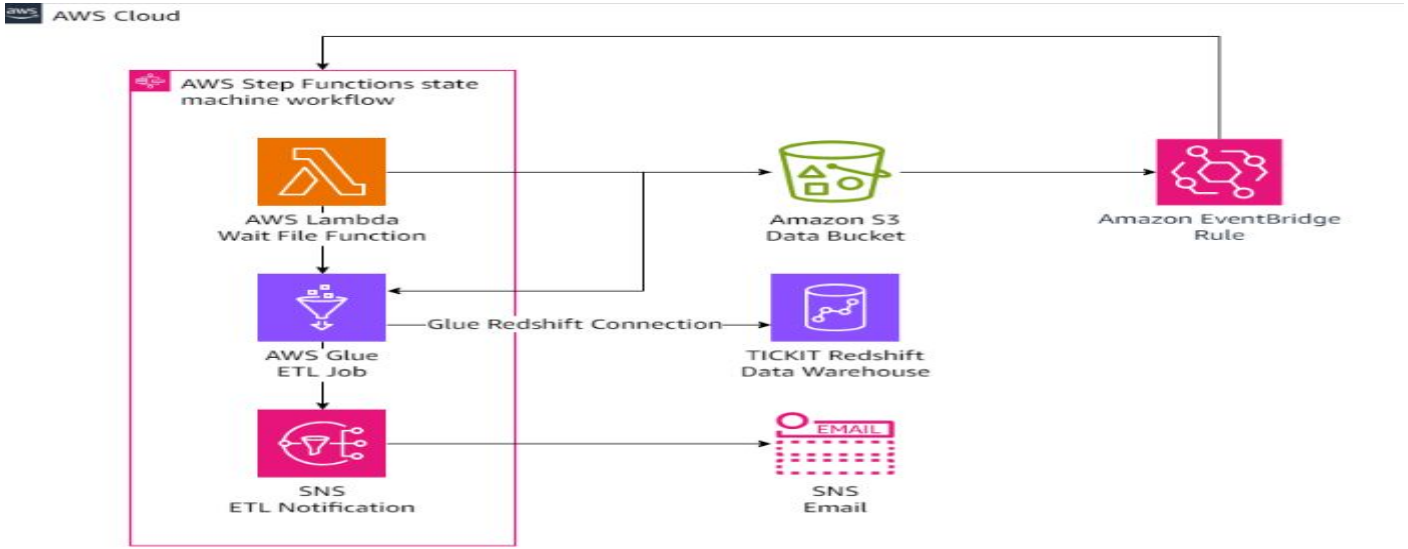
Create Glue ETL job, create an SNS topic to send a notification when the ETL job is complete, and create a Step Functions state machine workflow that combines these services together with Event Bridge to orchestrate the ETL workflow.

Objectives

By the end of this lab, you should be able to do the following:

- Create a Glue ETL job.
- Create an SNS topic.
- Create a new Step Functions state machine workflow using Workflow Studio.
- Configure an Amazon Event Bridge rule.

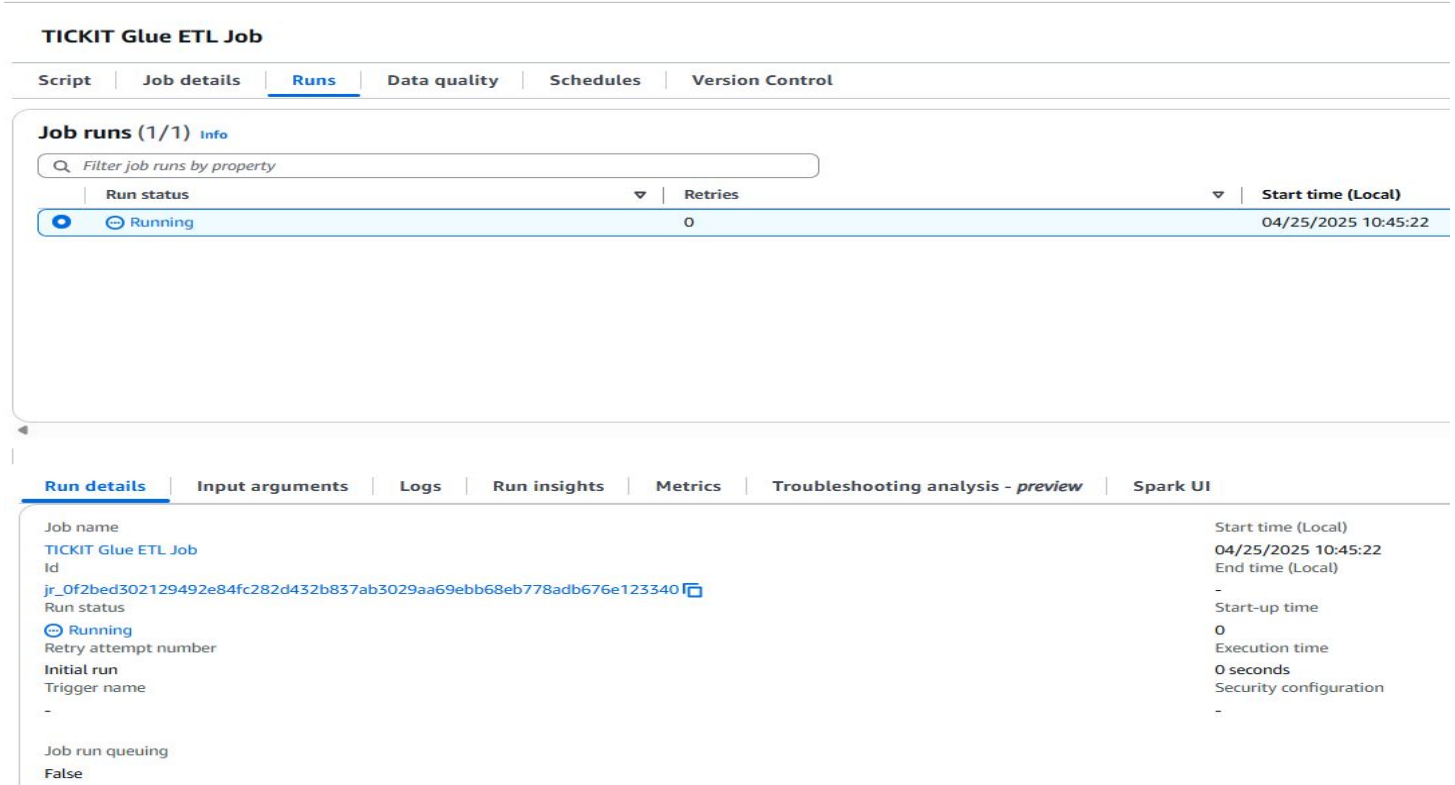


3. AWS Management Console, in the search bar, search for and choose `AWS Glue`.
4. In the left navigation pane, choose ETL jobs.
5. Created TICKIT Glue ETL Job.
   You can see the Glue job definition in the Script section. Each step is saved as a node. This job completes the following steps:
   - Loads data from four S3 buckets.
   - Applies data mapping to each file.
   - Joins all of the data together into one table.
   - Applies data mapping to the final table.
   - Uses a SQL query to aggregate columns for some of the important category names, the number of tickets sold, the average price paid per ticket, and the average commission per ticket.
   - Writes the data from Glue to a Redshift table called users_sales.

Test out the script by saving the Glue job and running it.

6. Choose **Run**.
7. Choose Runs tab.
   There is one running job listed. Once the job is complete a new table is created in your Redshift cluster called users_sales.

For the glue job script refer to **Gluejob.py** file



- At the top of the AWS Management Console, in the search bar, search for and choose `Amazon Redshift`.
- On the Serverless dashboard page, choose **Query data**.

- In the new browser tab, you might get the following caution in a red color banner: User information couldn't be retrieved. You must have an account to use Redshift Query Editor V2. Choose x to close the prompt.
- To create an account to use Redshift Query Editor V2, on the AWS KMS encryption page, leave all the values at the default settings and choose **Configure account**.

  The query editor now opens in a new browser tab.

- In the Redshift query editor v2 pane, choose the vertical ellipses (three dots) at Serverless:labworkgrp.
- Choose Create connection.
- On the Connect to labworkgroup window prompt, choose AWS Secrets Manager.
- Select Choose a secret drop-down menu and choose redshift!labnamespace-dbadmin.
- Choose **Create connection**.
- On the top right, choose the newly created tickitdb database instead of the *dev* database.
- To list the users, enter the following query:

```
select userid, city, state, likemusicals, likeopera, musicals, plays, opera, pop, tickets_sold,
average_price_paid_per_ticket, average_commission_per_ticket

from users_sales

order by tickets_sold desc

limit 10;
```

Task 2: Create an SNS topic

As part of the ETL orchestration strategy, AnyCompany Events wants to be notified when the ETL jobs run and if there are any errors.

In this task, you create an SNS topic that you later integrate into your ETL workflow using a Step Functions state machine.

19. At the top of the AWS Management Console, in the search bar, search for and choose `Simple Notification Service`.
20. For Topic name, enter `TICKIT-ETL-Notification`.
21. Choose **Next step**.
22. Choose **Create topic**.
    You have created a topic. Now, create a subscription to your topic using an email address you have access to.
23. Choose **Create subscription**.
24. For Protocol, select Email.
25. For Endpoint, enter an email address you have access to.
26. Choose **Create subscription**.
    You subscription has been created.
27. Navigate to your email, wait for an email from AWS Notifications (*no-reply@sns.amazonaws.com*), open the email, and choose Confirm subscription.
    You can use this SNS topic in your Step Functions state machine workflow to notify you of ETL successes and failures.

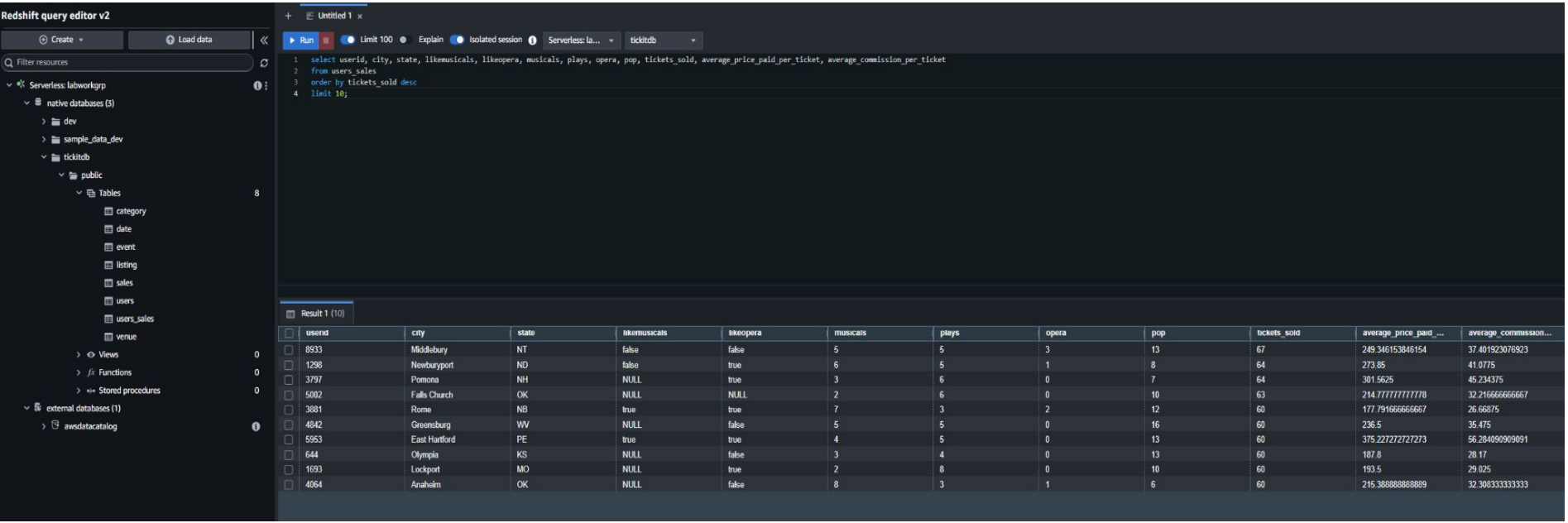
To list the users, enter the following query:

```
select userid, city, state, likemusicals, likeopera, musicals, plays, opera, pop, tickets_sold,
average_price_paid_per_ticket, average_commission_per_ticket

from users_sales

order by tickets_sold desc

  limit 10;
```



Task 2: Create an SNS topic

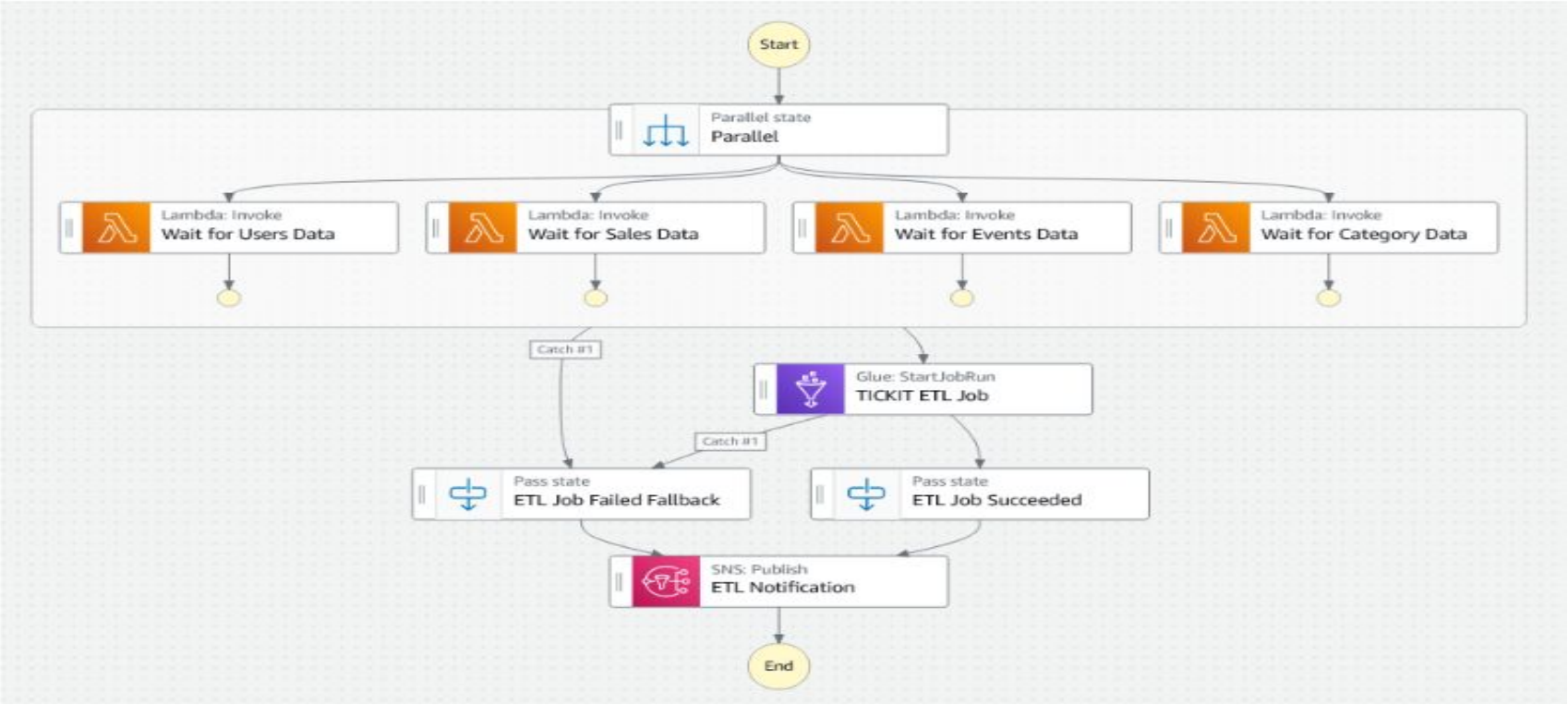
As part of the ETL orchestration strategy, AnyCompany Events wants to be notified when the ETL jobs run and if there are any errors.

In this task, you create an SNS topic that you later integrate into your ETL workflow using a Step Functions state machine.

19. At the top of the AWS Management Console, in the search bar, search for and choose `Simple Notification Service`.
20. For Topic name, enter `TICKIT-ETL-Notification`.
21. Choose **Next step**.
22. Choose **Create topic**.
    You have created a topic. Now, create a subscription to your topic using an email address you have access to.
23. Choose **Create subscription**.
24. For Protocol, select Email.
25. For Endpoint, enter an email address you have access to.
26. Choose **Create subscription**.
    You subscription has been created.
27. Navigate to your email, wait for an email from AWS Notifications (*no-reply@sns.amazonaws.com*), open the email, and choose Confirm subscription.
    You can use this SNS topic in your Step Functions state machine workflow to notify you of ETL successes and failures.

Task 3: Create a Step Functions state machine

create a Step Functions state machine that performs ETL jobs on four datasets uploaded to an S3 bucket. The ETL jobs and data joins are completed using AWS Glue and Lambda functions are used to wait for the initial file uploads. The state machine handles errors from the Glue jobs and returns a job status using an SNS notification.



Task 3.1: Create a state machine workflow with Workflow Studio

First, create a state machine workflow using Workflow Studio.

Workflow Studio for AWS Step Functions is a low-code visual workflow designer that lets you create serverless workflows by orchestrating AWS services. Using its drag-and-drop feature or the built-in code editor, you can create and edit workflows, control how input and output is filtered or transformed for each state, and configure error handling. As you drag and drop states to build your workflow, Workflow Studio validates your work and auto-generates code.

You can review the generated code or update the state machine definition within the code editor. When you are finished, you can save your workflow, run it, then examine the results in the Step Functions console. You can visually add and modify workflows to orchestrate the multiple services in your application.

28. At the top of the AWS Management Console, in the search bar, search for and choose `Step Functions`.
29. If the navigation menu is not displayed, choose located near the top left section of the page to display the service menu.
30. In the left navigation pane, choose State machines.
31. Choose **Create state machine**.
32. Choose **Select**.
    Workflow Studio opens with a Start and End node in the workspace.
33. Choose Config.
34. For State machine name, enter `TICKIT-ETL-State-Machine`.
35. For Execution role, choose the role with StepFunctionsRole in its name.
36. For Log level, choose ALL.
37. Choose Design.
    You are ready to start adding nodes to the state machine.
38. In the node pane, choose Flow.
39. Drag and drop Parallel between the Start and End nodes.

The Parallel state is going to be used for four Lambda functions. The four functions wait for the four data files to be present in the S3 data bucket. Once the files are present, the parallel state is completed and starts and AWS Glue job.

Create four Lambda nodes that wait for each of the four data files (users, sales, events, category).

40. In the node pane, choose Actions.
41. Drag and drop an AWS Lambda Invoke node into the Drop state here box.
42. For State name, enter `Wait for Users Data`.
43. For Function name, select WaitFileFunction:$LATEST.
    File contents: The *WaitFileFunction* contains the following code:

40. For Payload, select Enter payload.

In the text box, clear the JSON text and enter the following text:

```
{

  "S3_BUCKET": "DATA_BUCKET",

  "S3_KEY": "tickitdb/allusers_pipe.txt"
```

41. }
42. Replace the DATA_BUCKET placeholder value with the DataBucket value that is listed to the left of these instructions.
43. In the Wait for Users Data pane, choose Error handling.
44. For Retrier #1, choose the Edit Retrier #1 icon.
45. For Errors, enter `ValueError` and press Enter.
46. For Max attempts, enter `5`.
47. For Backoff rate, enter `1`.
    This error handling is configured to retry the function every minute five times. After five minutes of trying, the step fails if there is still a ValueError thrown.

40. Drag and drop an AWS Glue StartJobRun node between the Parallel node and the End node.
41. For State name, enter
`Run TICKIT ETL Job`.
42. To make sure the Step Function state machine waits until the Glue job is complete before moving to the next state, select Wait for task to complete.
43. In the Run TICKIT ETL Job pane, choose Arguments and Output.
- In the Arguments section, replace myJobName with `TICKIT Glue ETL Job`.Next, create a node that outputs a message that the ETL job succeeded.
57. In the node pane, choose Flow.
58. Drag and drop a Pass node between the TICKIT ETL Job node and the End node.
59. For State name, enter
`ETL Job Succeeded`.
60. Choose Output.For Result, enter the following text:
```
{  "Status": "The TICKIT ETL Job Completed Successfully"
}
```

Create a node to publish a message to your SNS topic.

62. In the node pane, choose Actions.
63. Drag and drop an Amazon SNS Publish node between the ETL Job Succeeded node and the End node.
64. For State name, enter
`ETL Notification`.
65. For Topic, choose the topic with TICKIT-ETL-Notification in its name.

Next, configure the Parallel state to catch errors and send an ETL failed notification.

66. Choose the Parallel state.
67. Choose Error handling.
68. Choose **Add new catcher**.
69. For Errors, choose States.TaskFailed.
70. For Fallback state, choose ETL Notification.

A Catch #1 line appears in the workflow between the Parallel state and the ETL Notification node.

71. In the node pane, choose Flow.
72. Drag and drop a Pass node on the Catch #1 line.
73. For State name, enter
`ETL Job Failed Fallback`.
74. Choose Output.For Result, enter the following text:
```
{

  "Status": "The TICKIT ETL Job Failed"


}
```
76. Choose the Run TICKIT ETL Job node.
77. Choose Error handling.
78. Choose **Add new catcher**.
79. For Errors, choose States.TaskFailed.
80. For Fallback state, choose ETL Job Failed Fallback.
81. Choose **Create**.

Task 3.2: Test the state machine workflow

Now that the workflow has been created, test the state machine workflow with a manual run.

Choose **Execute**.

For Input, enter the following text:

```
{

  "Comment": "Start the ETL workflow"

}
```

Choose **Start execution**.

View the workflow as it runs. Once the success notification is sent, you can view the notification in your email.

Learn more: If you want to see how the *WaitFileFunction* Lambda function retries works, you can delete the sales_tab.txt file and rerun the Step Functions state machine. The *WaitFileFunction* retries every minute for five minutes until either the file is uploaded or it fails after the fifth attempt.

Task 4: Configure an S3 bucket with an Amazon Event Bridge rule

AnyCompany Events updates its sales, events, users, and categories data periodically. Whenever a new sales file is uploaded, they want the Step Functions state machine to automatically start. You can configure the data bucket with an Amazon Event Bridge rule that automatically starts the workflow when a specific file is uploaded to the bucket.

In this task, you create an Amazon Event Bridge rule and configure an S3 bucket to use the new rule.

85. At the top of the AWS Management Console, in the search bar, search for and choose `Amazon EventBridge`.
86. In the left navigation pane, under Buses, choose Rules.
87. Choose **Create rule**.
88. For Name, enter `TICKIT-Sales-File-Rule`.
89. Choose **Next**.
90. In the Event pattern section, for AWS service, choose Simple Storage Service (S3).
91. For Event type, choose Amazon S3 Event Notification.
92. For Event Type Specification 1, choose Specific event(s).
93. For Specific event(s), select Object Created.
94. For Event Type Specification 2, choose Specific bucket(s) by name.

For Specific bucket(s) by name, copy and paste the DataBucket value that is listed to the left of these instructions.
Take a moment to view the Event pattern that has been created for you. The final event pattern should look like the following event pattern:

```
{  "source": ["aws.s3"],

  "detail-type" : ["Object Created" ],

  "detail": {

    "bucket": {

      "name": ["labstack-0123456789-databucket-yw5h7yar7ril"  ]

    }  }}
```

95. Choose **Next**.
96. For Select a target, select Step Functions state machine.
97. For State machine, select TICKIT-ETL-State-Machine.
98. For Execution role, choose Use existing role.
99. For Role name, select EventBridgeRole.
100. Choose **Next**.
101. Choose **Next**.
102. Choose **Create rule**

104. At the top of the AWS Management Console, in the search bar, search for and choose `S3`.
105. Choose the bucket that has databucket in its name.
106. Choose Properties.
107. In the Amazon EventBridge section, choose **Edit**.
108. For Send notifications to Amazon EventBridge for all events in this bucket, choose On.
109. Choose **Save changes**.

Task Complete: You have successfully created an Amazon EventBridge rule and connected the rule to the Amazon S3 bucket and the Step Functions state machine workflow.

Your Glue ETL job is set up, your Step Functions state machine workflow waits for files in S3 and then starts the glue job, and an EventBridge rule starts the state machine workflow every time a new file is uploaded to the databucket. To test your ETL workflow, upload a new sales data file to the S3 bucket.

In this task, you upload data to S3 to start the state machine workflow. Once the workflow has been automatically started, you view the running workflow.

110. At the top of the AWS Management Console, in the search bar, search for and choose `S3`.
111. Choose the bucket with databucket in its name.
112. Choose the folder with tickitdb in its name.
113. Save the sales_tab.txt file to your local machine (You can usually right-click and choose “Save link as…” from your browser to do this).
114. Choose **Upload**.
115. Choose **Add files**.
116. Select the sales_tab.txt file you just downloaded and choose Open.
117. Choose **Upload**.

The file uploads to the S3 bucket. Once the upload finishes, the EventBridge rule starts the Step Functions state machine.

118. At the top of the AWS Management Console, in the search bar, search for and choose `Step Functions`.
119. Choose the TICKIT-ETL-State-Machine state machine.
120. Choose the most recent Execution with a status of Running.