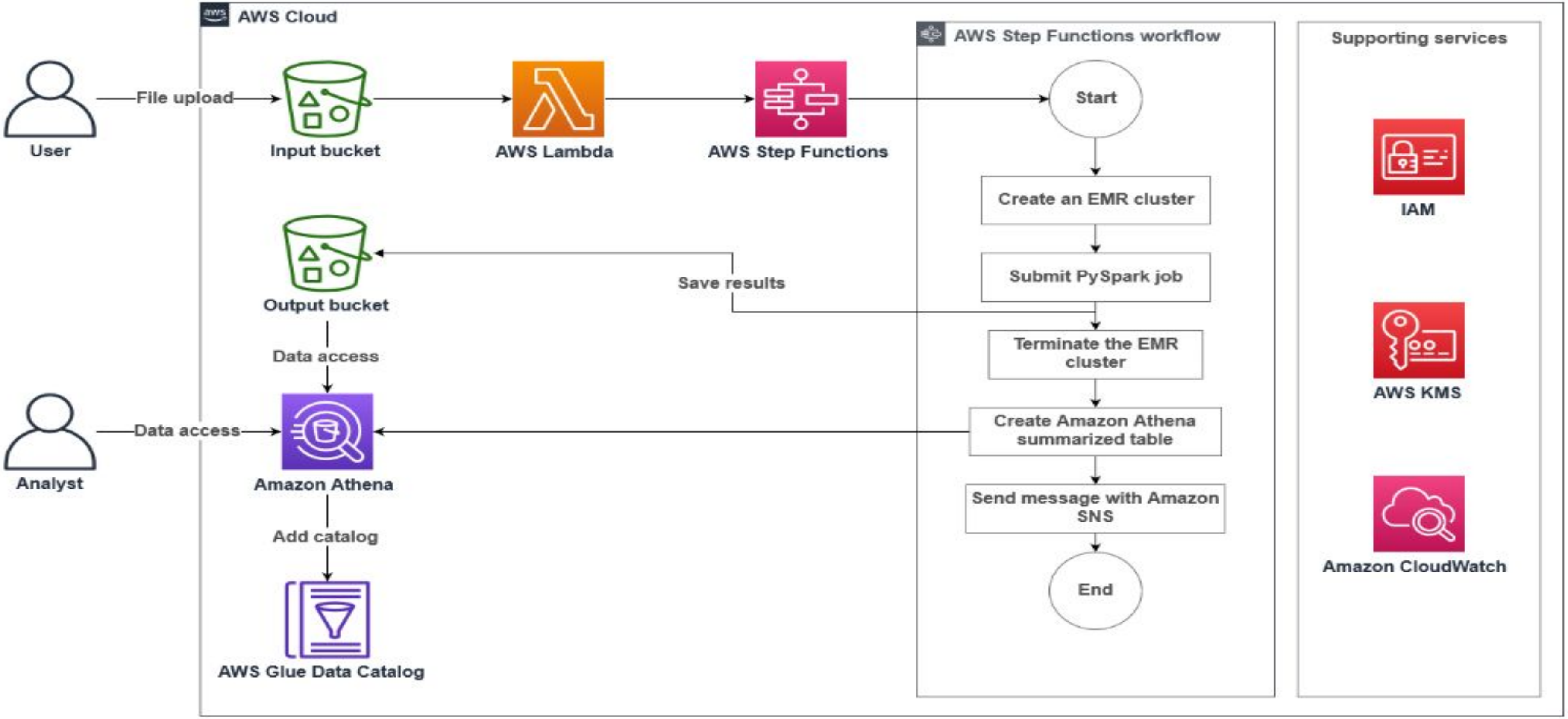
Run batch processing on 2020 stock data and append it to the 2019 stock data that has already been processed. You will use *Step Functions* to implement a solution that creates an *Amazon EMR* cluster and then terminates the cluster when it is no longer needed to conserve resources and reduce cost. You will use AWS Lambda, Amazon Simple Notification Service (Amazon SNS), and state machines to achieve your goal



Objectives

- Use S3 Event Notifications and AWS Lambda to automate the batch processing of data
- Use the *Step Functions* state machine language to:
    - Create an on-demand *Amazon EMR* cluster
    - Add an Apache Spark step job in *Amazon EMR* and create an *Amazon Athena* table to query the processed job
    - Add an *Amazon SNS* topic to send a notification
- Validate a Step Functions state machine run
- Review an *AWS Glue* table and validate the processed data using Athena.

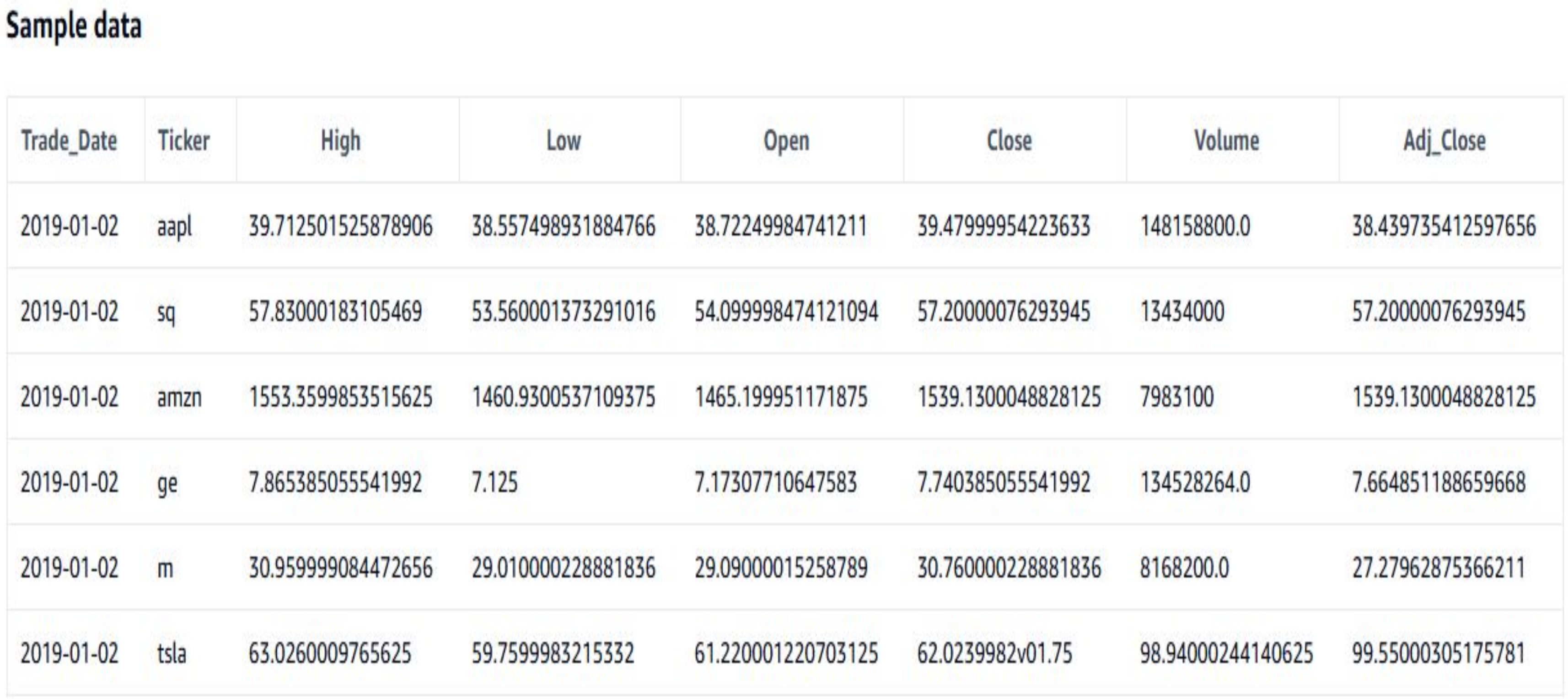
Review the contents in the Amazon Simple Storage Service (Amazon S3) bucket

3. At the top of the page, in the unified search bar, search for and choose
   `S3`
4. Select the bucket with databucket in its name.
5. The following folders were created for the task:

- data/ - store the input stock price file in CSV format.
- logs/ - store the *Amazon EMR* logs for troubleshooting any failures.
- output/ - store the Spark-processed data in Parquet format.
- results/ - store the *Amazon Athena* query results.
- scripts/ - store the PySpark script to process data.

6. Select the name of the data/ folder.

A file called stock_prices_2019.csv is saved to the folder. This object lists stock prices for a variety of large tech companies (AAPL, SQ, AMZN, GE, M, TSLA, and MSFT) for the year 2019. Data columns you can find include Trade_Date, Ticker, High, Low, Open, Close, Volume, and Adj_Close.



Sample data

Trade_Date	Ticker	High	Low	Open	Close	Volume	Adj_Close
2019-01-02	aapl	39.712501525878906	38.557498931884766	38.72249984741211	39.47999954223633	148158800.0	38.43973541259765 6
2019-01-02	sq	57.83000183105469	53.560001373291016	54.099998474121094	57.20000076293945	13434000	57.20000076293945
2019-01-02	amzn	1553.3599853515625	1460.9300537109375	1465.199951171875	1539.1300048828125	7983100	1539.1300048828125
2019-01-02	ge	7.865385055541992	7.125	7.17307710647583	7.740385055541992	134528264.0	7.664851188659668
2019-01-02	m	30.959999084472656	29.010000228881836	29.09000015258789	30.760000228881836	8168200.0	27.27962875366211
2019-01-02	tsla	63.0260009765625	59.7599983215332	61.220001220703125	62.0239982v01.75	98.94000244140625	99.55000305175781

Configure S3 Event Notifications

In this task, you use the *Amazon S3* Event Notifications feature to run the *Lambda* function when a new file is uploaded to the *S3* folder. The *Lambda* function initiates the *Step Functions* run.

You can read more about S3 Event Notifications here.

15. At the top of the page, in the unified search bar, search for and choose `S3`
16. Select the bucket with databucket in its name.
17. Choose the Properties tab and scroll down to the Event notifications section.
18. Choose **Create event notification**
19. On the Create event notification page, configure:

- Event name: `file_upload`
- Prefix: `data/`
- Suffix: `.csv`
- Event types: Select Put.
- Destination: Select Lambda function.
- Choose Choose from your Lambda functions.
- Lambda function: Select the function with runFunction in its name.

20. Choose **Save changes**

Update and upload the PySpark script in the Amazon S3 dataBucket

Each night, your company receives the stock feed from various sources, but you want to identify high-volume trades so that you can serve a specific group of investors. This script identifies the trades with a volume greater than 100,000 shares. You need to add the script that completes this work to the scripts folder in your *S3* bucket so that the state machine can complete the stock data processing.

```
import sys

import time

from pyspark.sql import SparkSession

from pyspark.sql.functions import *

from pyspark.sql.types import *

bucket_name = "<dataBucket>"

spark =  SparkSession.builder.appName("stock-summary").getOrCreate()

stockDF =  spark.read.option("header",True).csv("s3://"+bucket_name+"/data/")

stockDF.registerTempTable("stock_data_view")

StockSummaryDF = spark.sql("SELECT `Trade_Date`, `Ticker`, `Close` FROM stock_data_view WHERE Volume > 100000
ORDER BY Close DESC")

StockSummaryDF.write.mode('overwrite').parquet("s3://"+bucket_name+"/output/")

   spark.stop()
```

- Replace *<dataBucket>* with the dataBucket value shown to the left of these instructions.
- Save the file as *script.py*.
- At the top of the page, in the unified search bar, search for and choose `S3`
- Select the bucket with databucket in its name.
- Select the scripts/ folder and then choose **Upload**
- On the Upload page, choose **Add files**
- Navigate to the folder in your local computer and select the script.py file you saved in a previous step.
- Choose **Upload**

Update AWS Lake Formation permissions

As part of the *Step Functions* state machine, you use *AWS Lake Formation* to create a database in your *AWS Glue* Data Catalog. You need to grant the *Step Functions* state machine permission to create a database.

- At the top of the page, in the unified search bar, search for and choose `AWS Lake Formation`
- If you are presented with a Welcome to Lake Formation pop-up window, configure below:
- Select Add myself
- Choose **Get Started**
- On the AWS Lake Formation page, under Administration in the left pane, choose Administrative roles and tasks.
- In the Database creators section, choose Grant and configure:
- From the IAM users and roles dropdown menu, select the IAM role with *-stepFunctionRole-* in its name.

You can also find the role by entering

`stepFunctionRole`in the search box.

- Under Catalog permissions, select Create database.
- Choose **Grant**

Subscribe to an Amazon SNS topic

- As part of this lab build, we have created an *Amazon SNS* topic for you. In this task, you subscribe to the topic and confirm the subscription. This will tell you when your batch processing job is complete.
- At the top of the page, in the unified search bar, search for and choose `Simple Notification Service`
- In the left pane, choose Topics.
- Select the topic with -TaskCompleteSNS- in its name.
- In the Subscriptions tab, choose **Create subscription** and configure:
- Protocol: Email
- Endpoint: An email address that can receive notifications from Amazon SNS
- Choose **Create subscription** to confirm.
- You should receive an email to confirm your subscription. Confirm the subscription by selecting the Confirm subscription link.
- In the left pane, choose Topics.
- Select the topic with TaskCompleteSNS in its name.
- In the Subscriptions section, you should see your subscription status as Confirmed.
- **Upload a file to the Amazon S3 bucket**
- In this task, you upload the 2020 stock data file to invoke the S3 event notification you created earlier, which runs a Lambda function to initiate a *Step Functions* step.
- To download the stock_prices_2020.csv file, choose the text link.
- At the top of the page, in the unified search bar, search for and choose `S3`
- Select the bucket with databucket in its name.
- Select data/ and choose **Upload**.
- On the Upload page, choose **Add files**.
- Navigate to the folder in your local computer and select the stock_price_2020.csv file you saved in a previous step.
- Choose **Upload**.
-

Validate Step Functions

- At the top of the page, in the unified search bar, search for and choose `Step Functions`
- If prompted to leave or stay on the page, choose Leave Page.
- In the left pane, choose State machines.
- On the State machines page, select the state machine with BatchProcessingStep in its name.
- In the Executions section, select the name of the running state machine.
- If you do not see a state machine running, you might have to re-upload the sample file in the S3 bucket.
- Choose the Details tab to view the state machine status. You should see the state machine in a Running status.

Wait until all the steps are green and the state machine is in a Succeeded status. It typically takes 10-15 minutes to complete the run.

After the task is complete, you will receive an email that confirms The Task is complete!

While you are waiting for the task to complete, learn more about the Amazon States Language here.

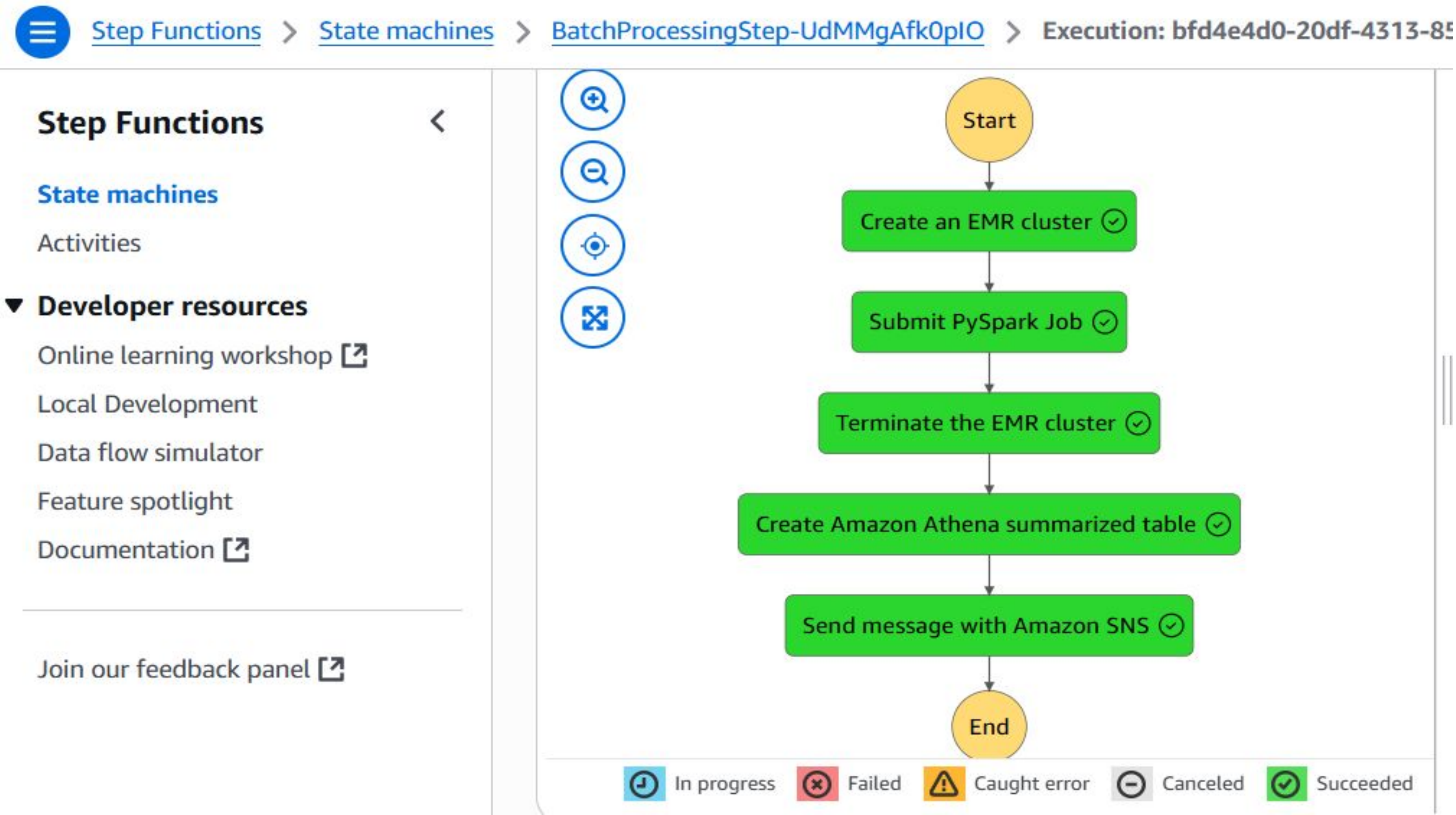
Update Lake Formation permissions to view the table

Once you receive notification that the task is complete, you can grant yourself permissions to access the stock_summary table so you can view the processed data.

- At the top of the page, in the unified search bar, search for and choose `AWS Lake Formation`
- In the left navigation pane, choose Data catalog settings.
- Clear both checkboxes for Use only IAM access control….

Note: If you see the error “You don’t have permissions to access this resource”, ignore the error.

- Choose **Save**.
- In the left navigation pane, choose Data lake permissions from the Permissions section.
- Choose **Grant** and configure:
- Choose IAM users and roles.
- IAM users and roles: *Choose the Federated user you are logged in as (see instructions below for help finding this user)*.
- LF-Tags or catalog resources: *Named data catalog resources*
- Databases: *default*
- Tables - optional: *stock_summary*
- Table permissions: *Select*


Step Functions > State machines > BatchProcessingStep-UdMMgAfk0pIO > Execution: bfd4e4d0-20df-4313-85
Step Functions
State machines
Activities
▼ Developer resources
Online learning workshop
Local Development
Data flow simulator
Feature spotlight
Documentation
Join our feedback panel
Start
Create an EMR cluster ✓
Submit PySpark Job ✓
Terminate the EMR cluster ✓
Create Amazon Athena summarized table ✓
Send message with Amazon SNS ✓
End
In progress
Failed
Caught error
Canceled
Succeeded

Validate the AWS Glue table and run a query on Athena to validate the data

In this task, you review the stock_summary table in the AWS Glue Data Catalog and run a query in Amazon Athena to validate the data. In AWS Glue, a table is the metadata definition that represents your data, including its schema.

60. At the top of the page, in the unified search bar, search for and choose `AWS Glue`
61. In the left pane Data catalog section, under Databases, choose Tables.
62. On the Tables page, choose stock_summary.

You should see the schema and details of the table you created with Step Functions.

If no tables appear, delete anything that is in the search box.

63. At the top of the page, in the unified search bar, search for and choose `Athena`

Note: If you are brought to the *Amazon Athena* homepage, select **Launch query editor** to navigate to the *Query editor*.

Before querying your data, you must first configure it for use as an *Amazon Athena* database.

64. Configure the following:
- Data Source: AwsDataCatalog
- Database: default

If a Workgroup primary settings window opens, choose **Acknowledge**.

Under Tables, you can see the stock_summary table listed.