

AWS Glue Studio ワークショップ

2021/03/29

シニアエバンジェリスト

亀田 治伸

1. 以下の URL へアクセスし CloudFormation を起動します。

短縮

<https://amzn.to/36n5q1F>

フル

<https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/review?stackName=Glue-Studio-Blog&templateURL=https://aws-bigdata-blog.s3.amazonaws.com/artifacts/gluestudio/cftemplate/CFgluestudio.yaml>

2. 以下にチェックを付けて、「スタックの作成」を押します。

The following resource(s) require capabilities: [AWS::IAM::Role]

このテンプレートには、Identity and Access Management (IAM) リソースが含まれています。これらのリソースを個別に作成し、それぞれに最小限必要な権限を与えるかどうか確認してください。さらに、カスタム名が付けられているか確認してください。カスタム名が、ご利用の AWS アカウント内で一意のものであることを確認してください。 [詳細はこちら](#)

☐ AWS CloudFormation によって IAM リソースがカスタム名で作成される場合があることを承認します。

キャンセル 変更セットの作成 **スタックの作成**

3. 構築作業中となりますのでしばらく待ちます。(たまに更新ボタンを押して画面をリフレッシュしてください)

Glue-Studio-Blog

削除 更新する スタックアクション ▼ スタックの作成 ▼

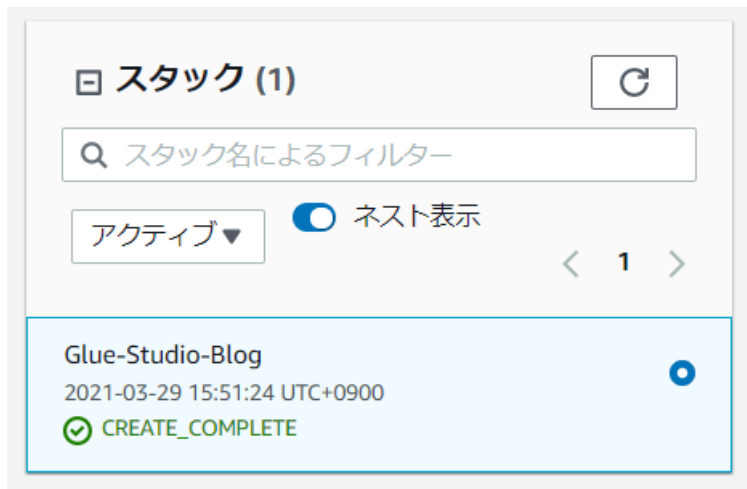
スタックの情報 イベント リソース 出力 パラメータ テンプレート 変更セット

イベント (1)

検索イベント

タイムスタンプ	論理 ID	ステータス	状況の理由
2021-03-29 15:51:24 UTC+0900	Glue-Studio-Blog	CREATE_IN_PROGRESS	User Initiated

4. 以下が表示されれば、作成完了です。



- リソースタブを見ると、Glue のデータベースが 1 つ、テーブルが 3 つ、S3 バケット、IAM ロールが作成されたことがわかります。

Glue-Studio-Blog

削除 更新する スタックアクション ▼ スタックの作成 ▼

スタックの情報 イベント **リソース** 出力 パラメータ テンプレート 変更セット

リソース (6)

Q リソースの検索

論理 ID ▲	物理 ID ▼	タイプ ▼	ステータス ▼	状況の理由 ▼	モジュール ▼
AWSGlueStudioParkingTicketCount	parking_tickets_count	AWS::Glue::Table	CREATE_COMPLETE	-	-
AWSGlueStudioRole	AWSGlueStudioRole	AWS::IAM::Role	CREATE_COMPLETE	-	-
AWSGlueStudioS3Bucket	glue-studio-blog-294963776963	AWS::S3::Bucket	CREATE_COMPLETE	-	-
AWSGlueStudioTableTickets	tickets	AWS::Glue::Table	CREATE_COMPLETE	-	-
AWSGlueStudioTableTrials	trials	AWS::Glue::Table	CREATE_COMPLETE	-	-
AWSGlueStudioTicketSYYZDB	yyz-tickets	AWS::Glue::Database	CREATE_COMPLETE	-	-

- AWS Glue のマネージメントコンソールにいきます。(管理者画面上の検索窓に glue と入力してください)
- 左のペインから「Glue Studio」を選んでください

ETL

AWS Glue Studio

New

Blueprints

ワークフロー

ジョブ

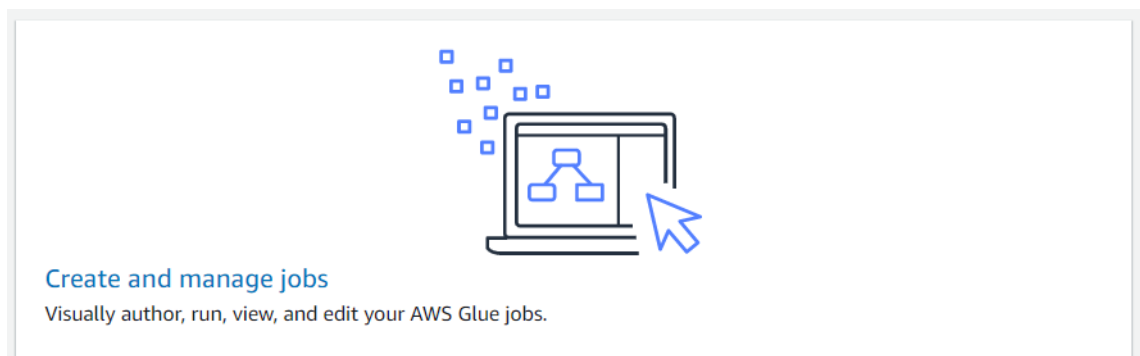
ML 変換

トリガー

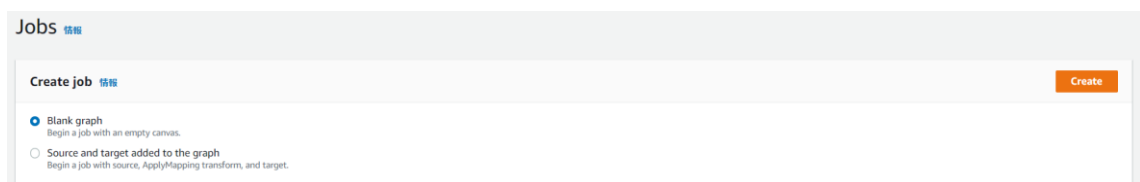
開発エンドポイント

ノートブック

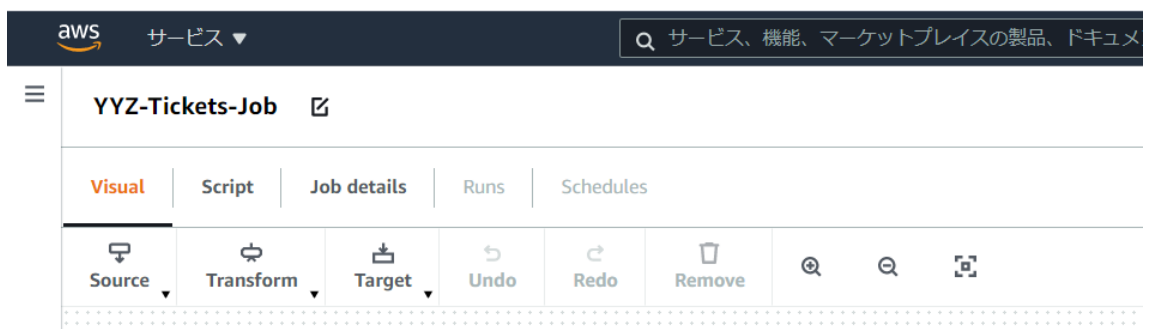
8. 「Create and Manage jobs」 を選びます



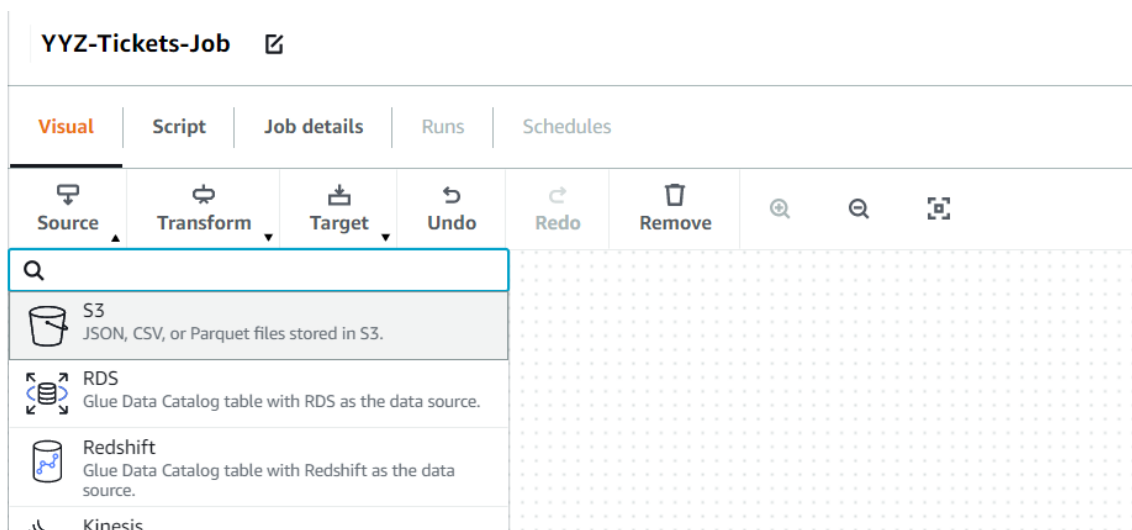
9. 「Blank Graph」 を選び、「Create」 をおします



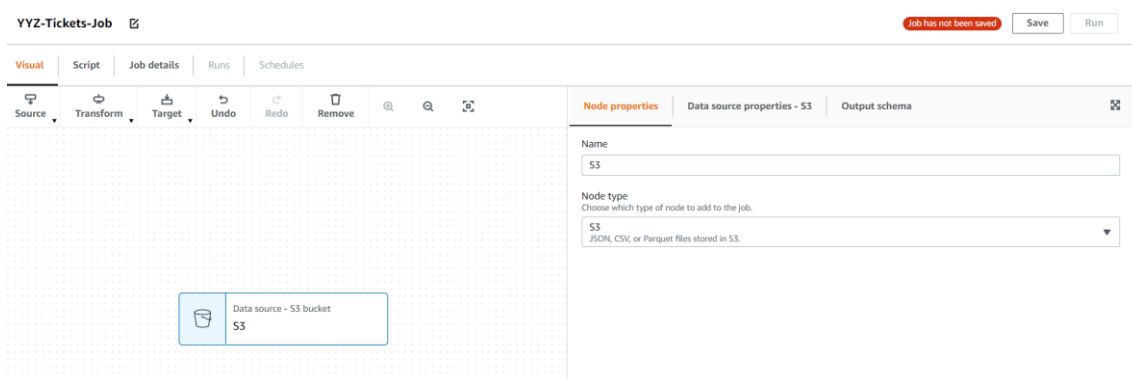
10. 左上 Job 名を「YYZ-Tickets-Job」に変更します。(名前は任意でなんでも OK です)



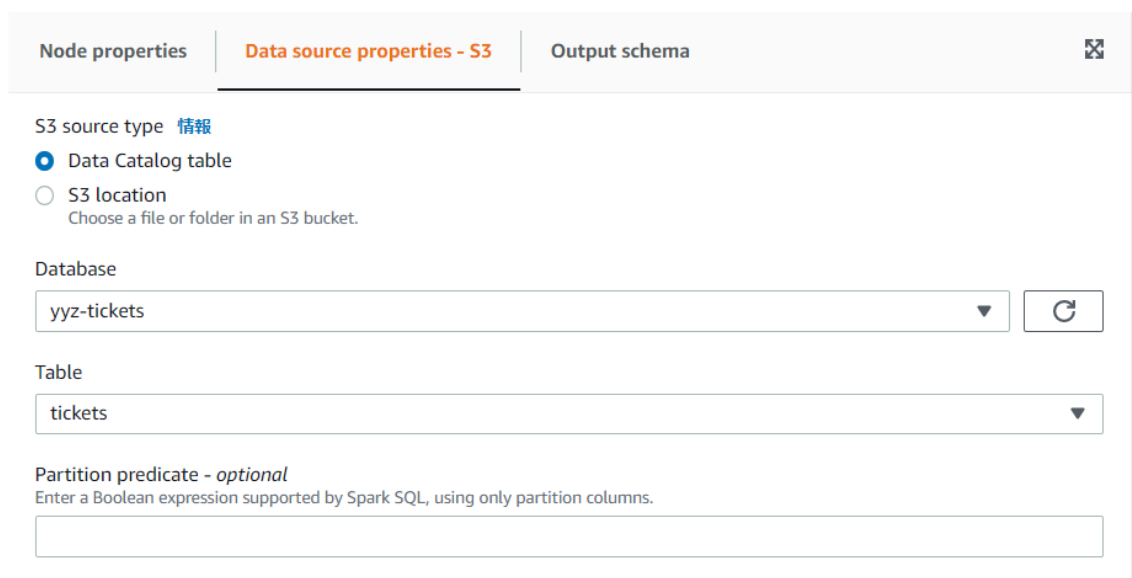
11. 「Visual」 → 「Source」 → 「S3」 を選びます



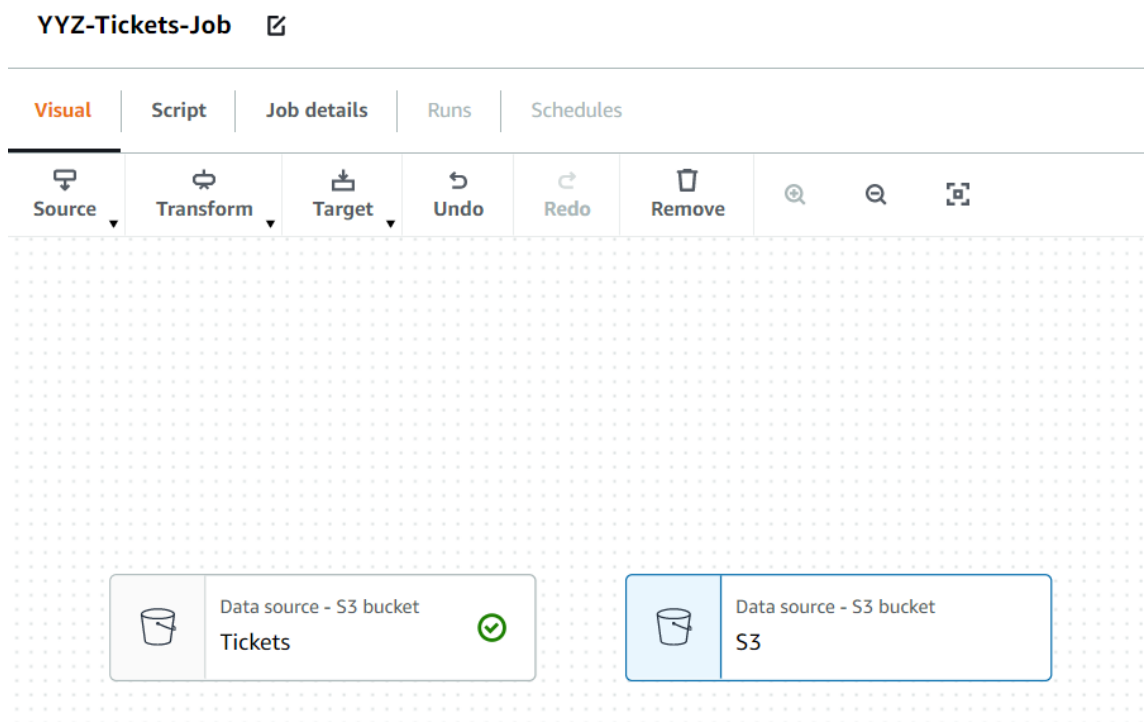
12. グラフ領域の S3 を選んで、画面右の「Note Properties」タブを選び、「Name」に Tickets と入力します



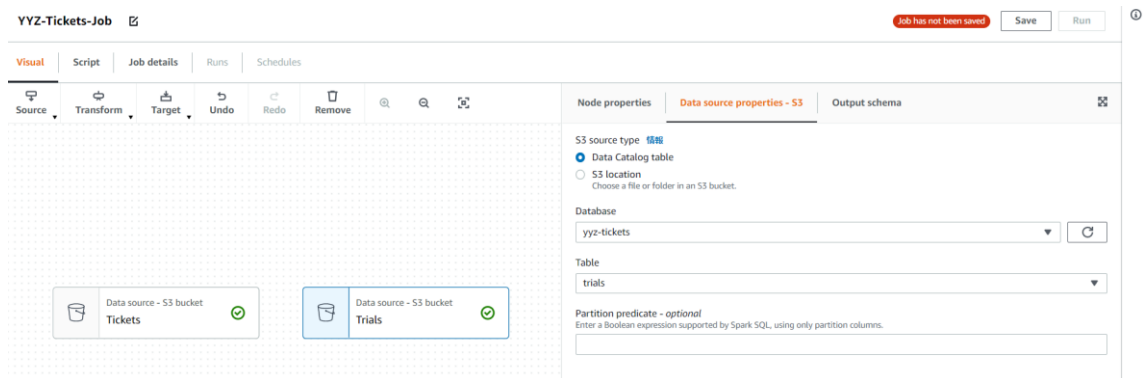
13. 「Data source properties S3」タブを開き、「Database」に yyz-tickets、「Table」に tickets を選びます。



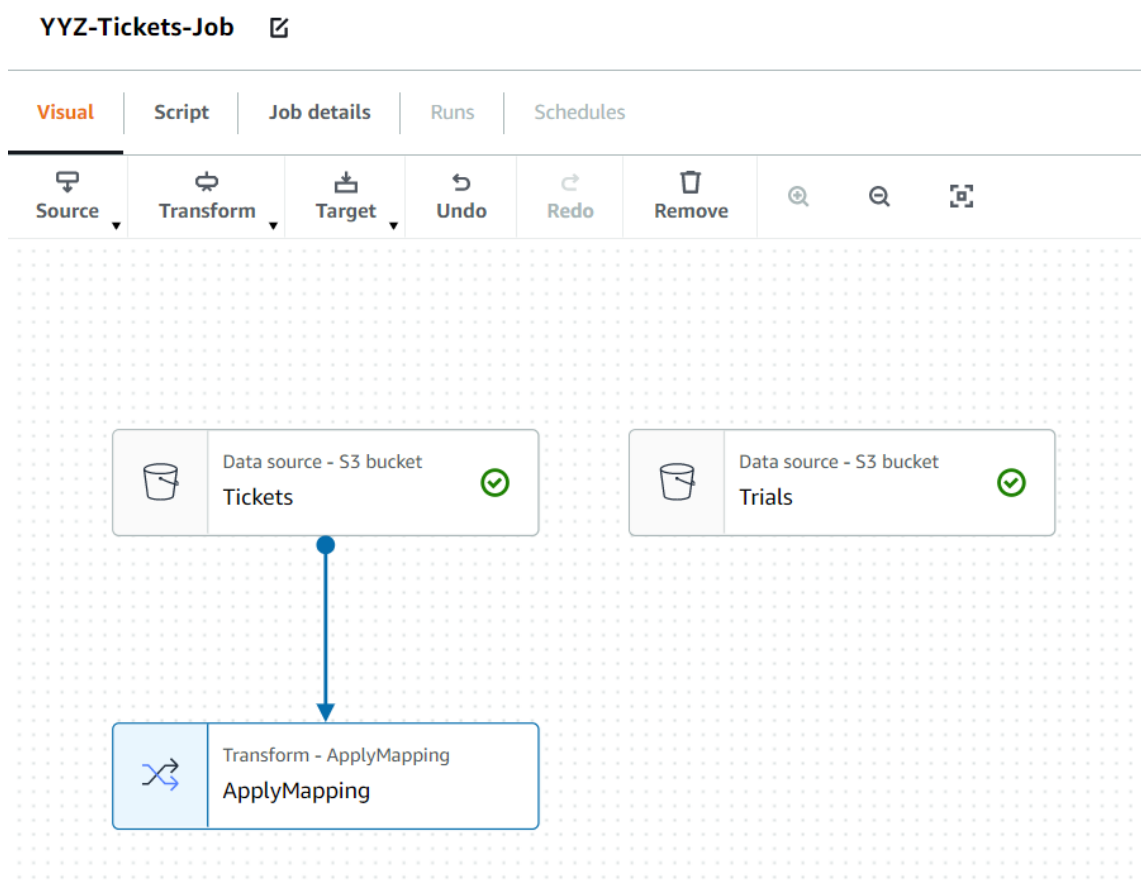
14. 2 個目の「Source」→「S3」を追加します。



- 「Name」に Trials と設定し、「Database」には先ほどと同じ yyz-tickets、「Table」に trials を選びます



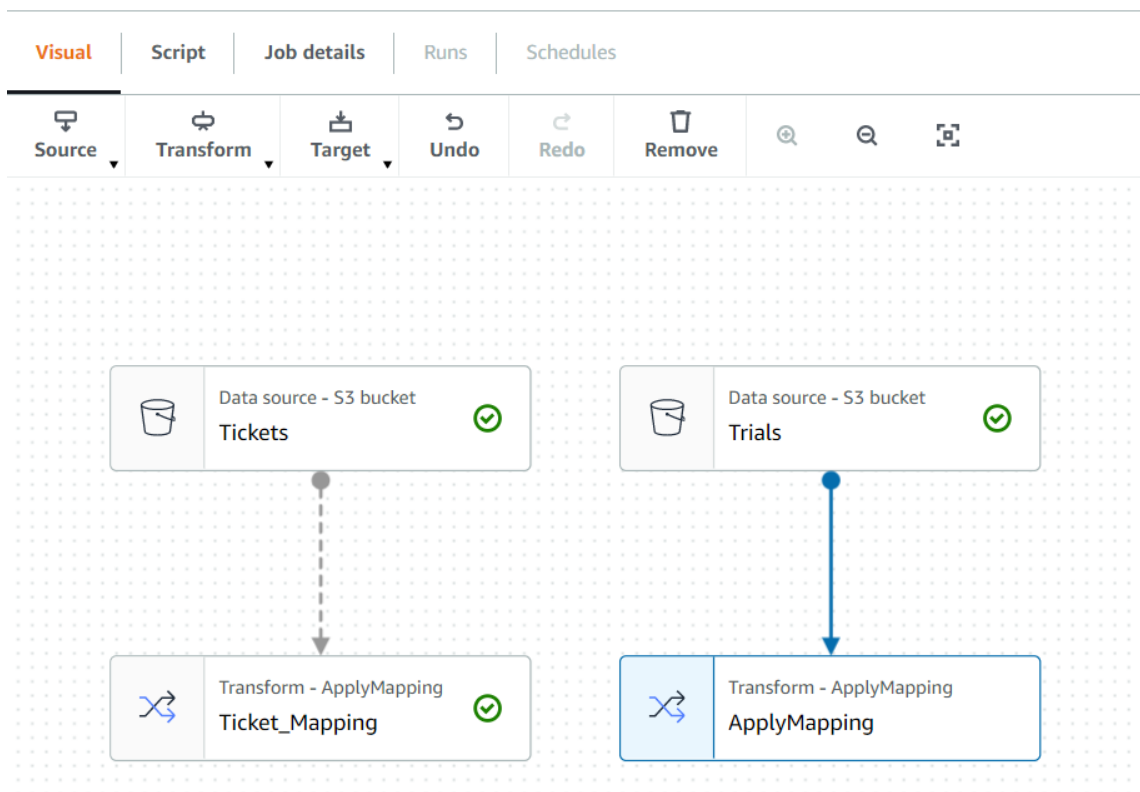
- グラフの Tickets を選び（薄い水色になります）、「Transform」から ApplyMapping を選びます。



17. 「Name」を Ticket_Mapping とし、「Transform」タブで ticket_number を decimal から int に変更します。さらに、Location1、Location2、Location3、Location4、Province のカラムを drop します

Apply mapping			
Source key	Target key	Data type	Drop
ticket_date	ticket_date	string ▼	<input type="checkbox"/>
ticket_number	ticket_number	int ▼	<input type="checkbox"/>
officer	officer	decimal ▼	<input type="checkbox"/>
infraction_code	infraction_code	decimal ▼	<input type="checkbox"/>
infraction_description	infraction_description	string ▼	<input type="checkbox"/>
set_fine_amount	set_fine_amount	decimal ▼	<input type="checkbox"/>
time_of_infraction	time_of_infraction	decimal ▼	<input type="checkbox"/>
location1			<input checked="" type="checkbox"/>
location2			<input checked="" type="checkbox"/>
location3			<input checked="" type="checkbox"/>
location4			<input checked="" type="checkbox"/>
province			<input checked="" type="checkbox"/>

18. 同様に、Trials の S3 の下にも、ApplyMapping をぶら下げます。

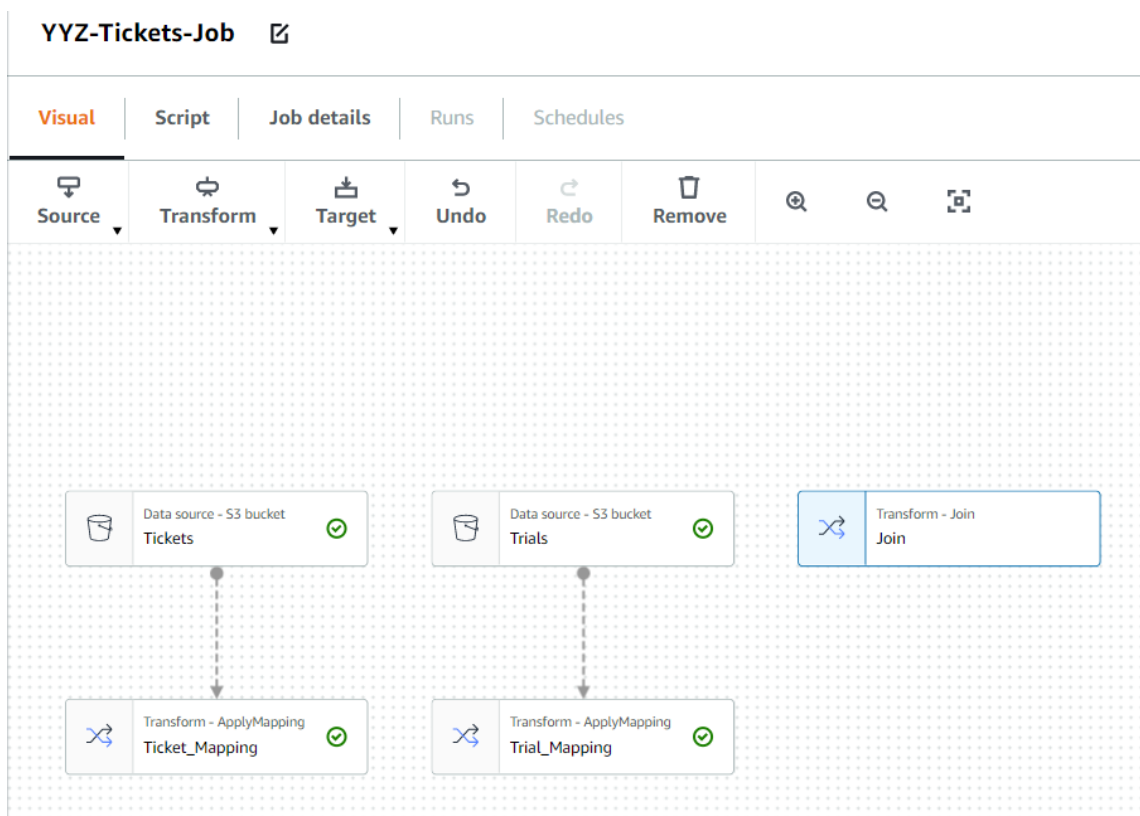


19. 「Name」を Trial_Mapping とし、parking_ticket_number の型を long から int に変更します

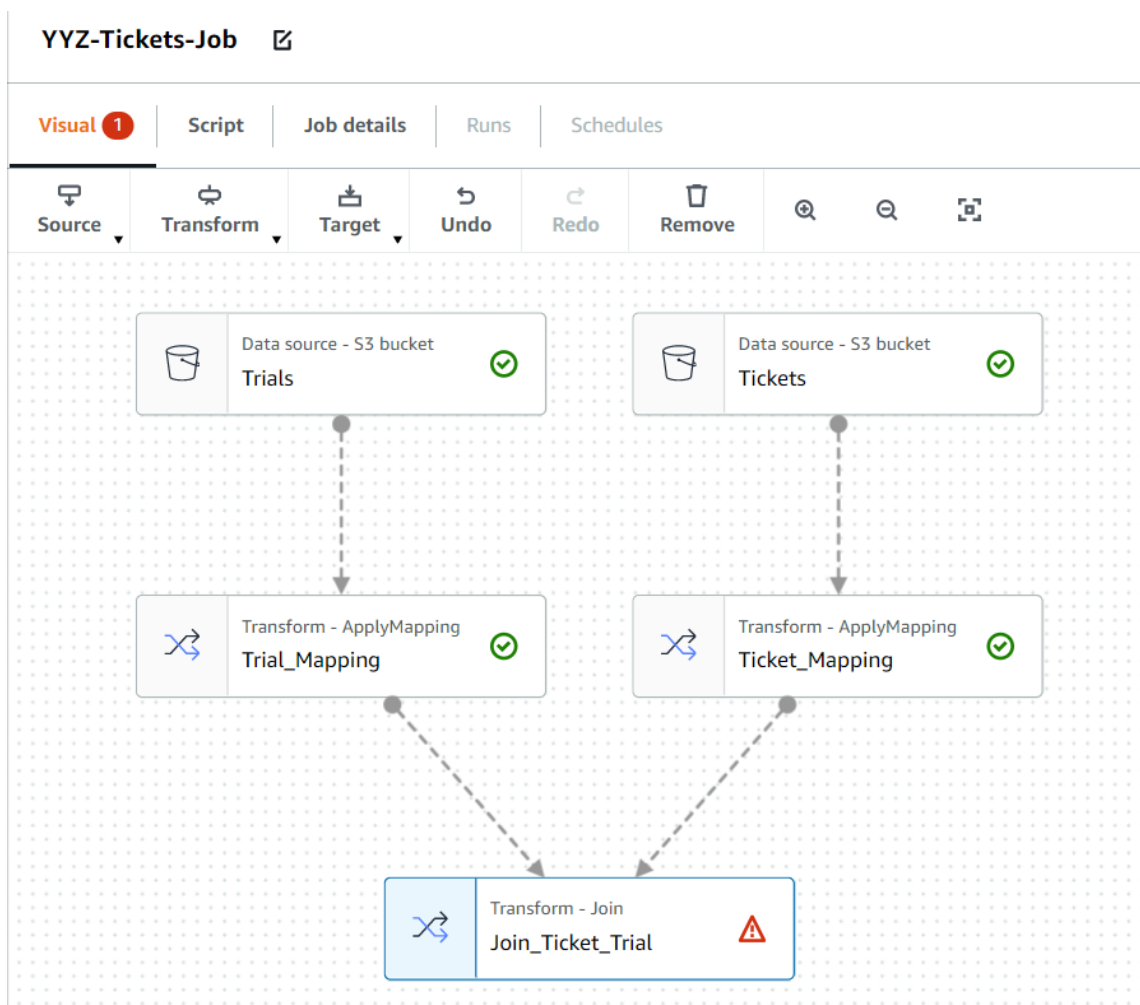
The screenshot shows the 'Transform' properties for the 'Trial_Mapping' node. The 'Apply mapping' table is displayed, showing the mapping of source keys to target keys and data types.

Source key	Target key	Data type	Drop
court_date	court_date	date	<input type="checkbox"/>
court_location	court_location	string	<input type="checkbox"/>
court_room	court_room	string	<input type="checkbox"/>
court_time	court_time	int	<input type="checkbox"/>
parking_ticket_number	parking_ticket_number	int	<input type="checkbox"/>
infraction_date	infraction_date	date	<input type="checkbox"/>
first_3_letters_name	first_3_letters_name	string	<input type="checkbox"/>
sentence	sentence	string	<input type="checkbox"/>


20. 以上でテーブル連結に必要な型の統一や、不要カラムの削除設定が完了しました。
「Transform」→「Join」を選択します




21. 「Name」を Join_Ticket_Trial と設定し、「Note parents」に先ほど作成した 2 つの Transform を選びます。



22. 「Transform」タブの「Join Conditions」で「Add condition」ボタンをおし、
parking_ticket_number と ticket_number を結びつけます


Node properties | **Transform** | Output schema 

Join type
Select what kind of join to perform.

 Inner join
Select all rows from both datasets that meet the join condition. ▼

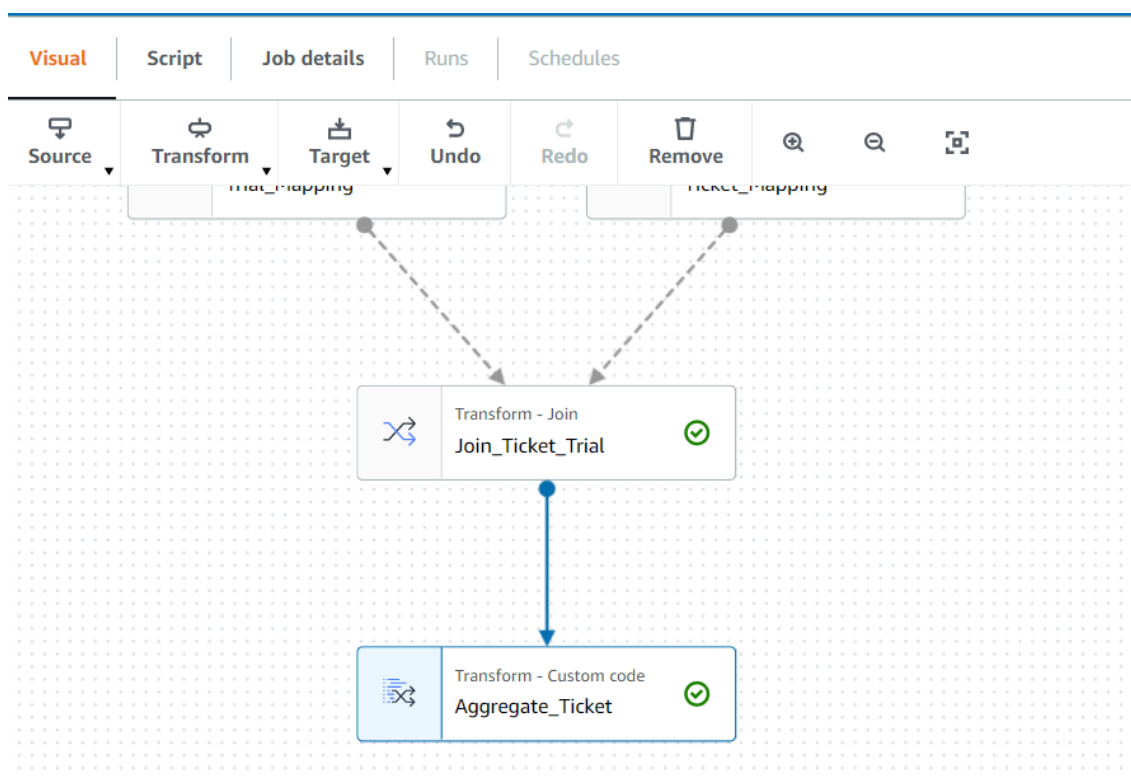
Join conditions
Select a key from each data input to set the condition of the join.

Trial_Mapping Ticket_Mapping

parking_ticket_number ▼ = ticket_number ▼ 

Add condition

23. Transform-Join を選択した状態で、「Transform」→「Custom Transform」を選び、「Name」を Aggregate_Ticket とします。



24. 「Transform」タブの「MyTransform」部分を Aggregate_Tickets に置換します。そして 2 行目以降に以下をペーストします。

```
selected = dfc.select(list(dfc.keys())[0]).toDF()
selected.createOrReplaceTempView("ticketcount")
totals = spark.sql("select court_location as location, infraction_description as
infraction, count(infraction_code) as total FROM ticketcount group by
infraction_description, infraction_code, court_location order by court_location
asc")
results = DynamicFrame.fromDF(totals, glueContext, "results")
return DynamicFrameCollection({"results": results}, glueContext)
```

Node properties

Transform

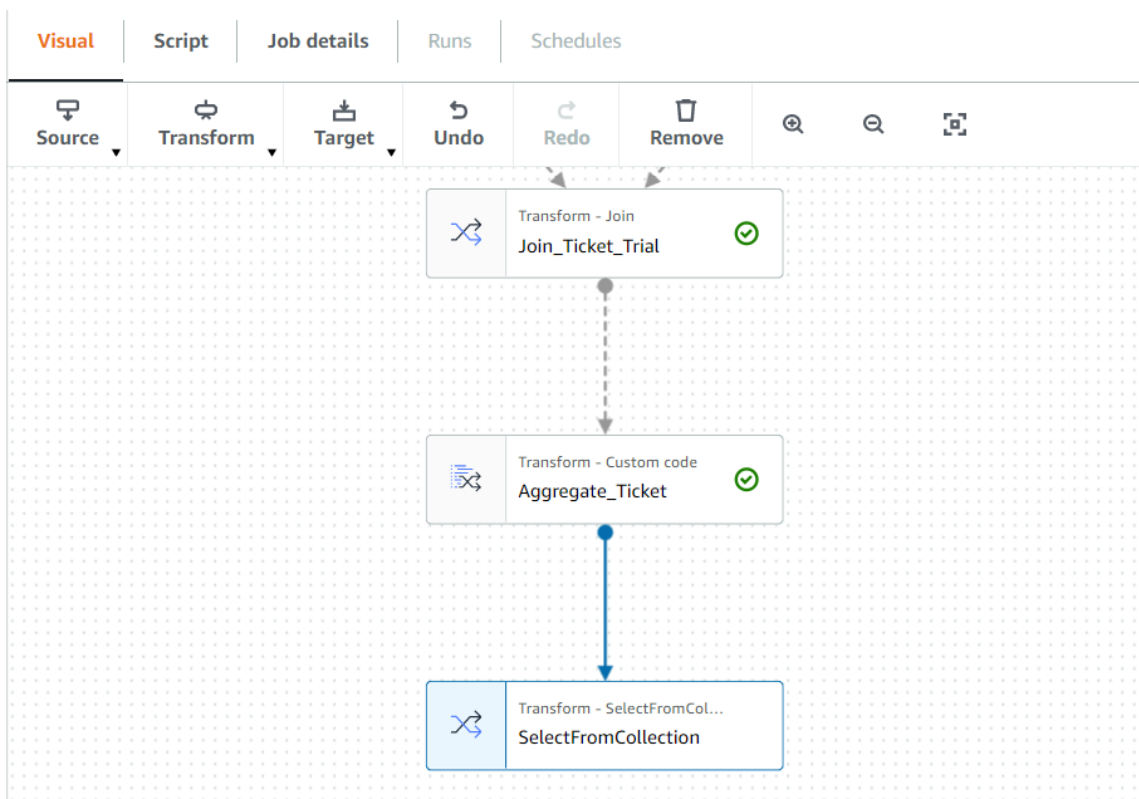
Output schema

Code block 情報

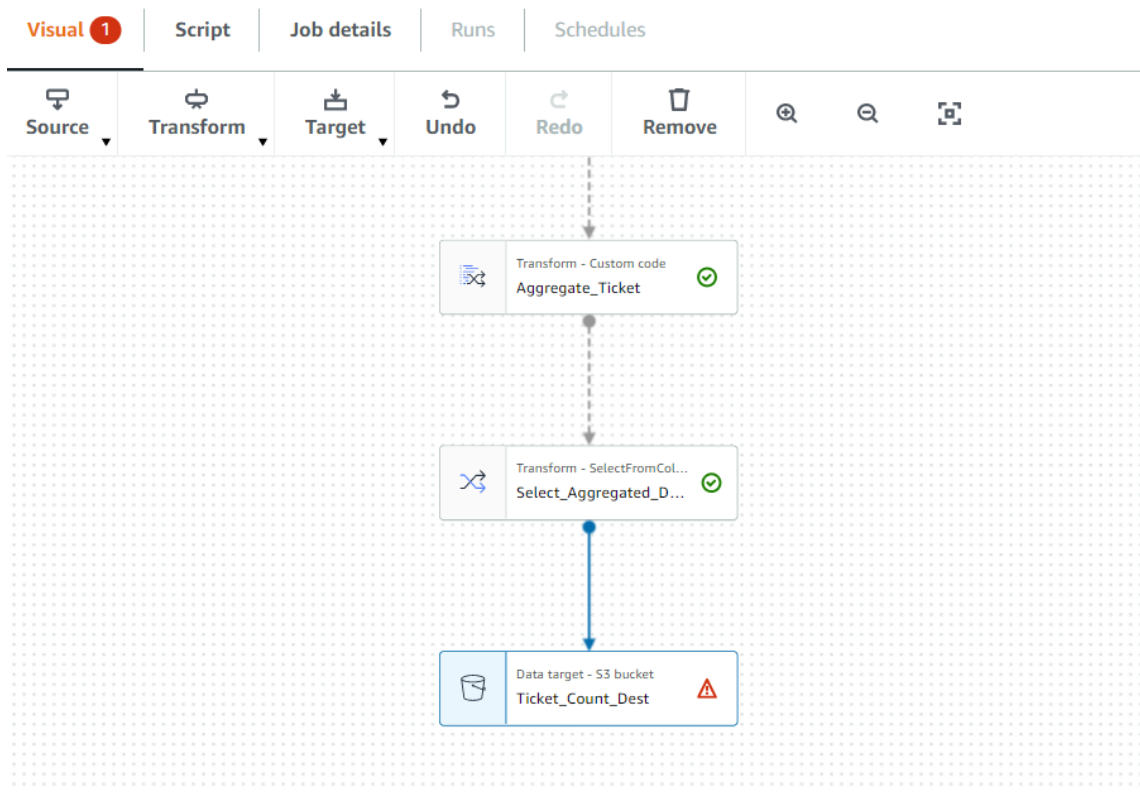
Enter a custom script to add to your job.

```
1 def Aggregate_Tickets (glueContext, dfc) -> DynamicFrameCollection:
2 selected = dfc.select(list(dfc.keys())[0]).toDF()
3 selected.createOrReplaceTempView("ticketcount")
4 totals = spark.sql("select count_location as location, infraction_description as infraction, count(i
5 results = DynamicFrame.fromDF(totals, glueContext, "results")
6 return DynamicFrameCollection({"results": results}, glueContext)
7
```

25. グラフ領域で「Aggregate_Ticket」を選んで「Transform」→「SelectFormCollection」を選びます。



26. 「Name」を Select_Aggregated_Data とします。
27. 「Target」→「S3」を選び、「Name」を Ticket_Count_Dest とします。

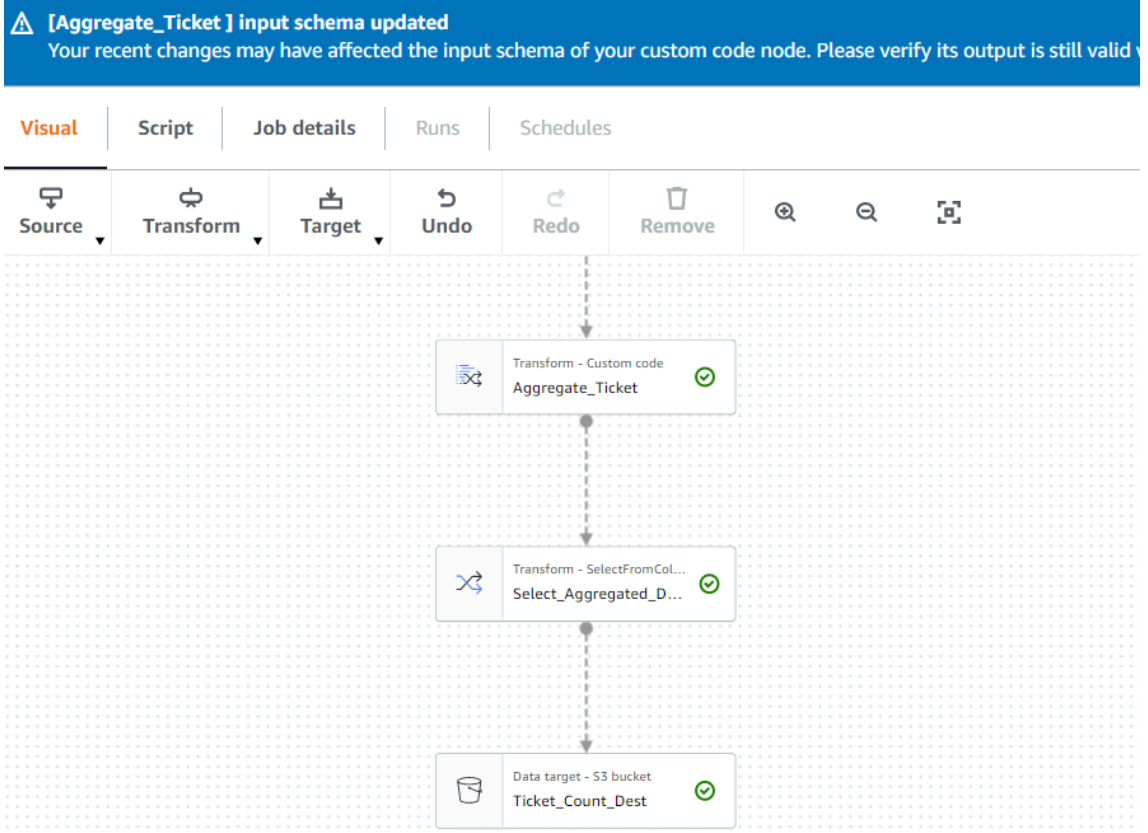


28. 「Data target properties -S3」タブで、「Format」を Parquet にし、Compression Type を GZIP にします。

Node properties	Data target properties - S3 1	Output schema
Format		
<div>Parquet ▼</div>		
Compression Type		
<div>GZIP ▼</div>		

29. 今日作成された S3 バケットを選び、後ろに「parking_tickets_count/」を付け加えます。(文字列は / で終わることに注意してください)
30. グラフ領域の上にあるタブで「Job details」を選びます。

YYZ-Tickets-Job



31. 今日作成された IAMRole を選びます


Basic properties 情報

Name

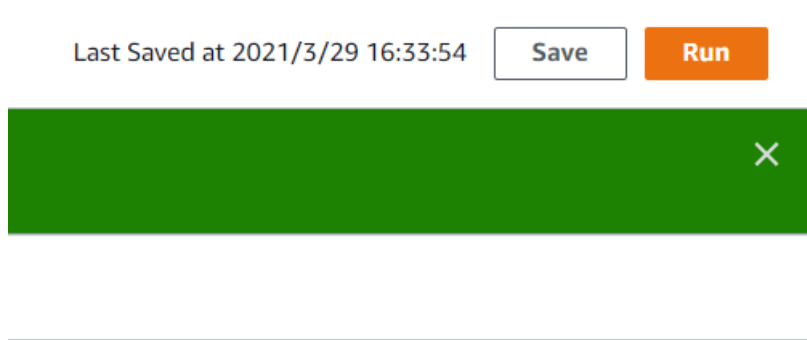
Description - optional

Descriptions can be up to 2048 characters long.

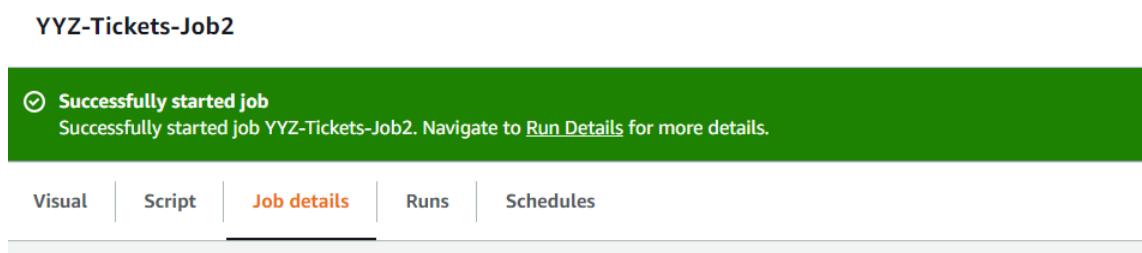
IAM Role
Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.


No description available.

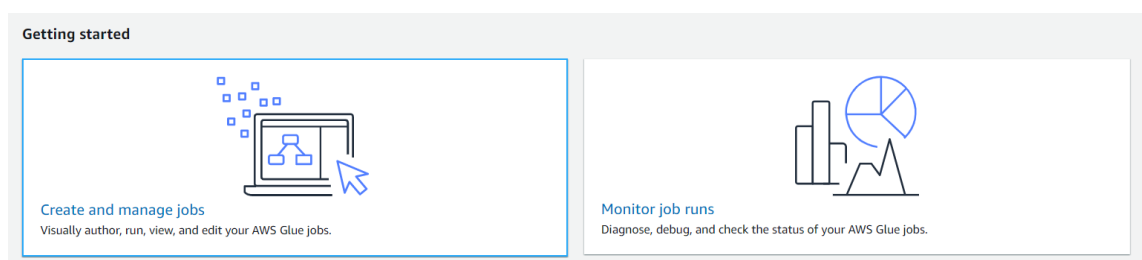
32. 「Job bookmark」を Disable に設定し、画面右上（ブラウザのスクロールバーで上に戻ります）で「Save」ボタンをおします。
33. 「Save」が終わったら「Run」ボタンをおします



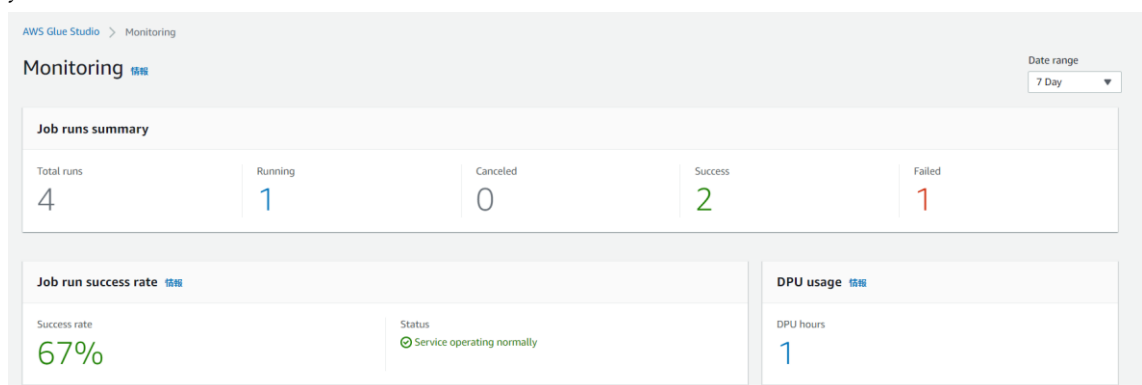
34. 以下の通り Job が開始されます。



35. ブラウザの別タブで再度 Glue Studio を開き、「Monitor job runs」を押して下さい



36. 以下のように Job の実行状況が可視化されます。何度かブラウザをリロードしながら job が完了するまで待ってください



37. Job が完了したら今日さ宇製された S3 バケットへ移動し、フォルダができていることを確認します。

glue-studio-blog-294963776963

オブジェクト | プロパティ | アクセス許可 | メトリクス | 管理 | アクセスポイント

オブジェクト (2)

オブジェクトは、Amazon S3 に保存された基本的なエンティティです。Amazon S3 インベントリを使用して、バケット内のすべてのオブジェクトのリストを取得できます。他のユーザーが自分のオブジェクトにアクセスできるためには、明示的にアクセス権限を付与する必要があります。詳細はこちら

リフレッシュ 削除 アクション ▼ フォルダの作成 アップロード

プレフィックスでオブジェクトを検索

<input type="checkbox"/>	名前 ▲	タイプ ▼	最終更新日時 ▼	サイズ ▼	ストレージクラス ▼
<input type="checkbox"/>	parking_tickets_count_\$folder\$	-	2021/03/29 04:35:53 PM JST	0 B	スタンダード
<input type="checkbox"/>	parking_tickets_count/	フォルダ	-	-	-

38. それでもいいのでフォルダの中に作成されている gzip 圧縮された parquet 形式のオブジェクトをクリックします

Amazon S3 > glue-studio-blog-294963776963 > parking_tickets_count/ > part-00002-99ea0435-c87c-4e7a-86c8-0a9873624174-c000.gz.parquet

part-00002-99ea0435-c87c-4e7a-86c8-0a9873624174-c000.gz.parquet

S3 URI をコピー オブジェクトアクション ▼

プロパティ | アクセス許可 | バージョン

オブジェクトの概要

所有者 hkameda	S3 URI s3://glue-studio-blog-294963776963/parking_tickets_count/part-00002-99ea0435-c87c-4e7a-86c8-0a9873624174-c000.gz.parquet
AWS リージョン 米国東部 (バージニア北部) us-east-1	Amazon リソースネーム (ARN) arn:aws:s3:::glue-studio-blog-294963776963/parking_tickets_count/part-00002-99ea0435-c87c-4e7a-86c8-0a9873624174-c000.gz.parquet
最終更新日時 2021/03/29 04:35:54 PM JST	エンティティタグ (Etag) 75e94ae3e30d7149229ff928f7fb6d51
サイズ 1.5 KB	オブジェクト URL
タイプ	

39. オブジェクトアクションで「S3 Select を使用したクエリ」を選んで実行してください。入力形式は Apache Parquet、出力形式は任意です。

入力設定

パス
s3://glue-studio-blog-294963776963/parking_tickets_count/part-00002-99ea0435-c87c-4e7a-86c8-0a9873624174-c000.gz.parquet

サイズ
1.5 KB (1581.0 B)

形式
☐ CSV
☐ JSON
☒ Apache Parquet

圧縮
Amazon S3 Select では、Apache Parquet オブジェクトのオブジェクト全体の圧縮はサポートされていません。

出力設定

形式
☒ CSV
☐ JSON

CSV 区切り記号
☒ カンマ
☐ タブ

40. 以下のようにデータ変換や Join が完了し正しくデータが生成されていることがわかります。

クエリ結果

[Close] を選択 するか移動すると、クエリ結果は使用できません。[Download results] を選択して、次のクエリ結果のコピーをダウンロードします。

[📄 結果のダウンロード](#)

ステータス

🟢 5 個のレコードを 268 ミリ秒で正常に返しました

返されたバイト数: 318 B

Raw

フォーマット済み

AP COMMERCIAL RESOLUTION N	STOP-(ON/OVER) (SIDEWK/FTP PATH)	1
AP COMMERCIAL RESOLUTION N	PARK MACHINE-REQD FEE NOT PAID	4
AP COMMERCIAL RESOLUTION N	STAND TAXI-SIGNED STD-NOT HIRE	1
AP COMMERCIAL RESOLUTION N	STOP-SIGNED HWY-PROHIBIT TM/DY	4
AP COMMERCIAL RESOLUTION N	STAND VEH.-PROHIBIT TIME/DAY	1

おつかれさまでした。以下を削除して下さい。

- Job（Glue Studio ではなく、Glue の画面からジョブを削除）
- CloudFormation スタック