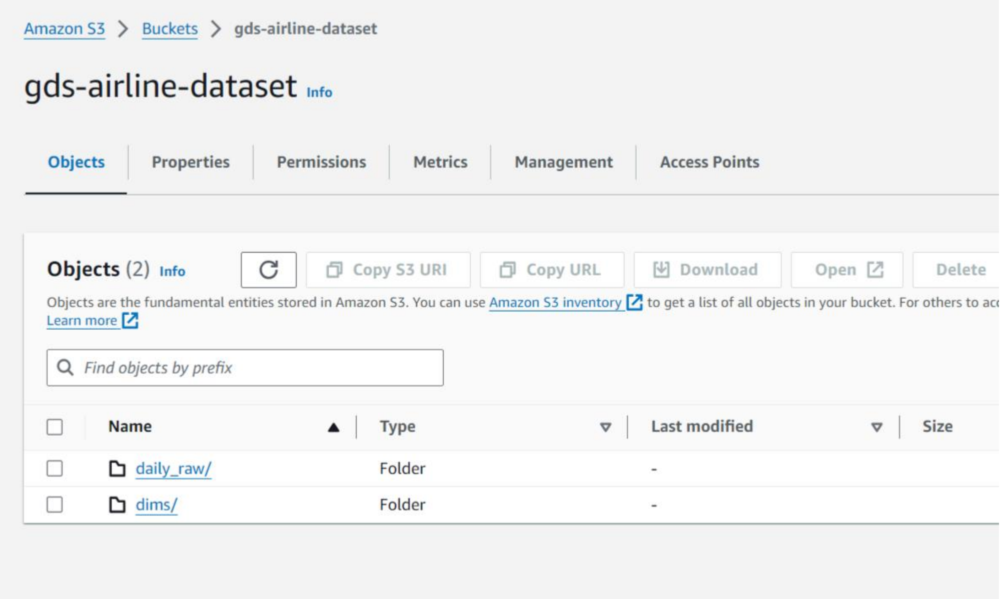
Airline Data Ingestion

1. this project was about daily incremental data load into redshift fact table(destination table).
2. created two separate folders in s3 bucket- one for daily raw where we have flights partitioned data and other for dimension table airport data.



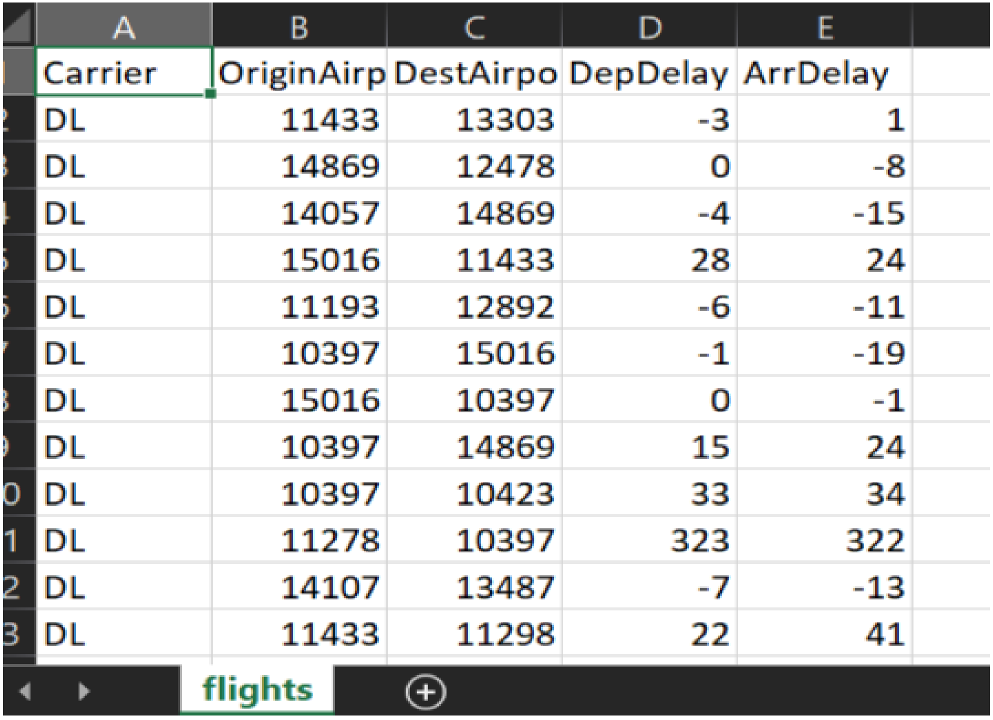
gds-airline-dataset

Objects (2) Info

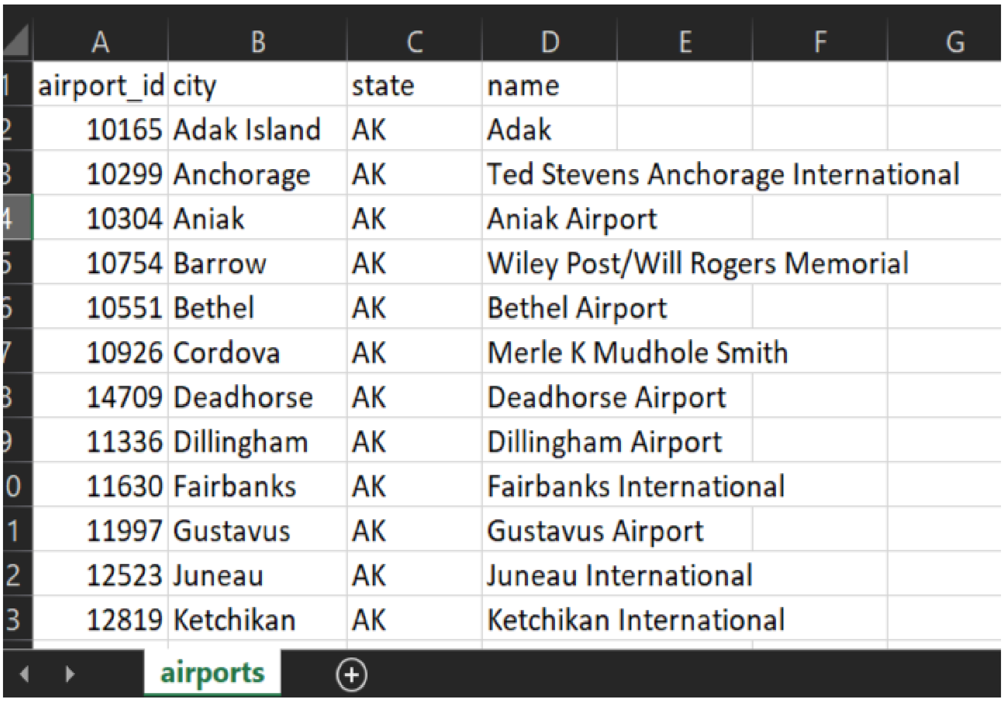
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to acc
Learn more

□	Name ▲	Type	Last modified	Size
□	daily_raw/	Folder	-	
□	dims/	Folder	-	

3. here flights data acting as fact table and airports data as dimension table. Fact table contains numerical data and foreign keys for referenced dimension tables. Dimension tables contain descriptive information. Like here its containing information for each airport.

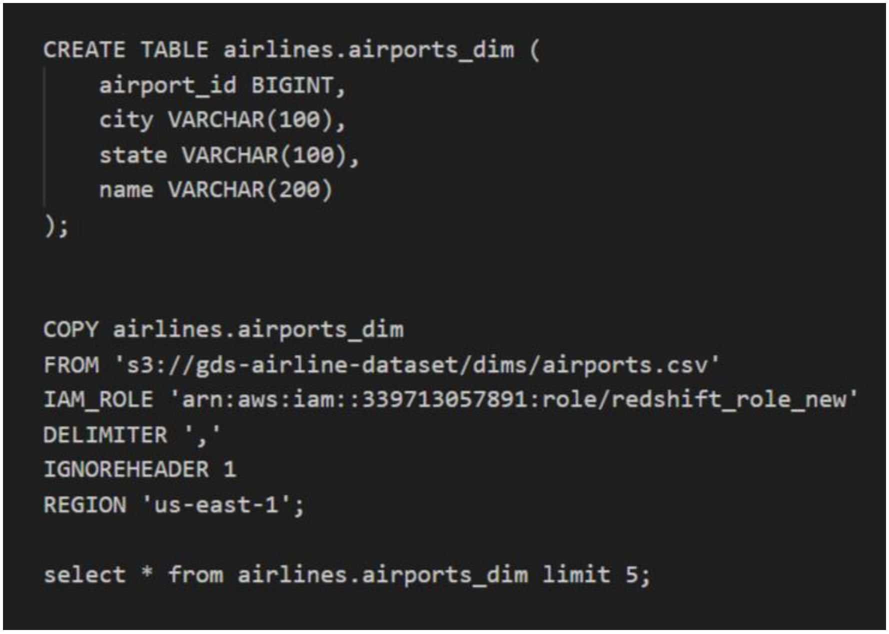

A
B
C
D
E
flights

Carrier	OriginAirp	DestAirpo	DepDelay	ArrDelay
DL	11433	13303	-3	1
DL	14869	12478	0	-8
DL	14057	14869	-4	-15
DL	15016	11433	28	24
DL	11193	12892	-6	-11
DL	10397	15016	-1	-19
DL	15016	10397	0	-1
DL	10397	14869	15	24
DL	10397	10423	33	34
DL	11278	10397	323	322
DL	14107	13487	-7	-13
DL	11433	11298	22	41

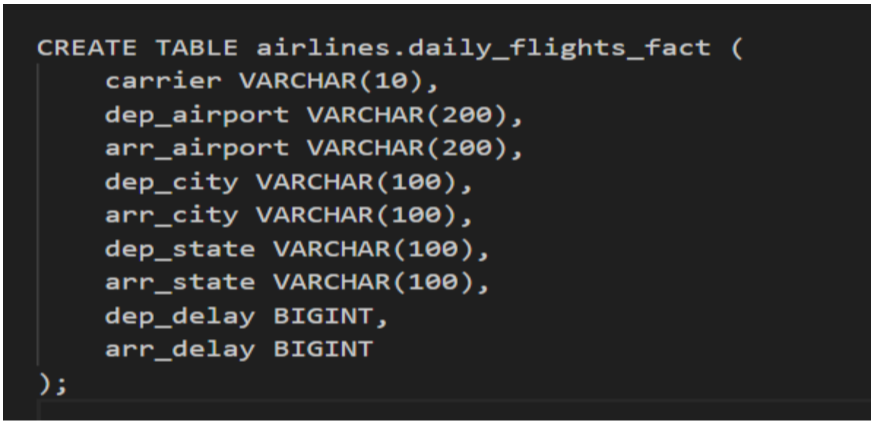


airport_id	city	state	name
10165	Adak Island	AK	Adak
10299	Anchorage	AK	Ted Stevens Anchorage International
10304	Aniak	AK	Aniak Airport
10754	Barrow	AK	Wiley Post/Will Rogers Memorial
10551	Bethel	AK	Bethel Airport
10926	Cordova	AK	Merle K Mudhole Smith
14709	Deadhorse	AK	Deadhorse Airport
11336	Dillingham	AK	Dillingham Airport
11630	Fairbanks	AK	Fairbanks International
11997	Gustavus	AK	Gustavus Airport
12523	Juneau	AK	Juneau International
12819	Ketchikan	AK	Ketchikan International

4. On the redshift warehouse, we created 2 tables. One is redshift dimension table 'airports_dim' where we load data from s3 bucket dimension data 'dims' folder. Other is redshift fact table 'daily_flights_fact' as destination table.

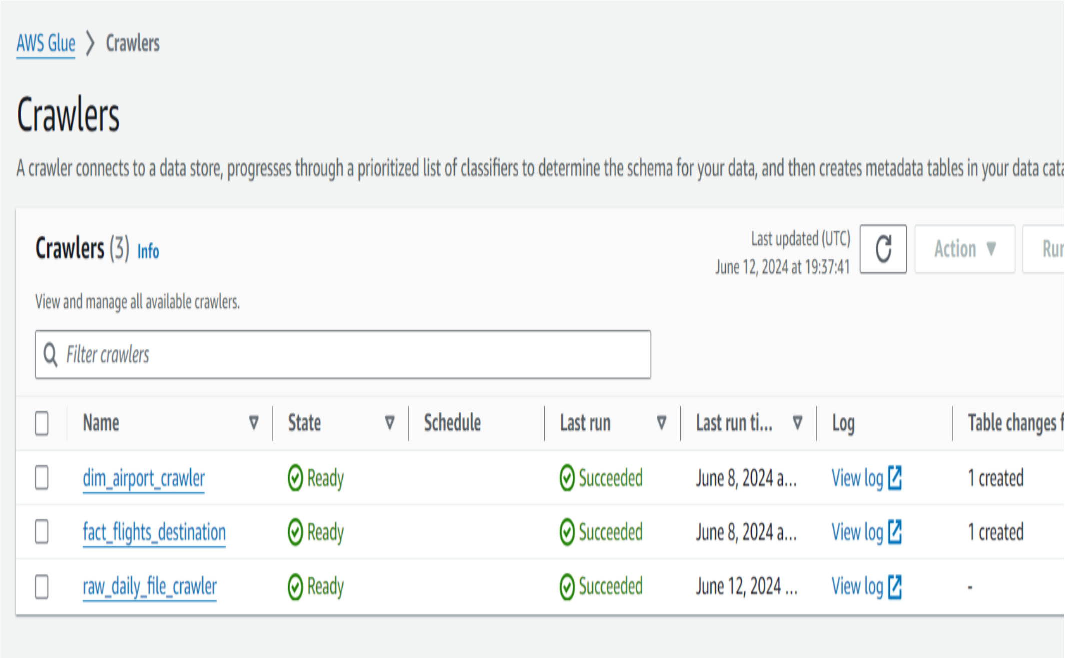


```
CREATE TABLE airlines.airports_dim (
    airport_id BIGINT,
    city VARCHAR(100),
    state VARCHAR(100),
    name VARCHAR(200)
);


COPY airlines.airports_dim
FROM 's3://gds-airline-dataset/dims/airports.csv'
IAM_ROLE 'arn:aws:iam::339713057891:role/redshift_role_new'
DELIMITER ','
IGNOREHEADER 1
REGION 'us-east-1';

select * from airlines.airports_dim limit 5;
```



```
CREATE TABLE airlines.daily_flights_fact (
    carrier VARCHAR(10),
    dep_airport VARCHAR(200),
    arr_airport VARCHAR(200),
    dep_city VARCHAR(100),
    arr_city VARCHAR(100),
    dep_state VARCHAR(100),
    arr_state VARCHAR(100),
    dep_delay BIGINT,
    arr_delay BIGINT
);
```

5. next we create crawlers over s3 daily raw flights data, redshift dimension table and redshift destination table which will be creating glue catalog metadata tables.



AWS Glue > Crawlers

Crawlers

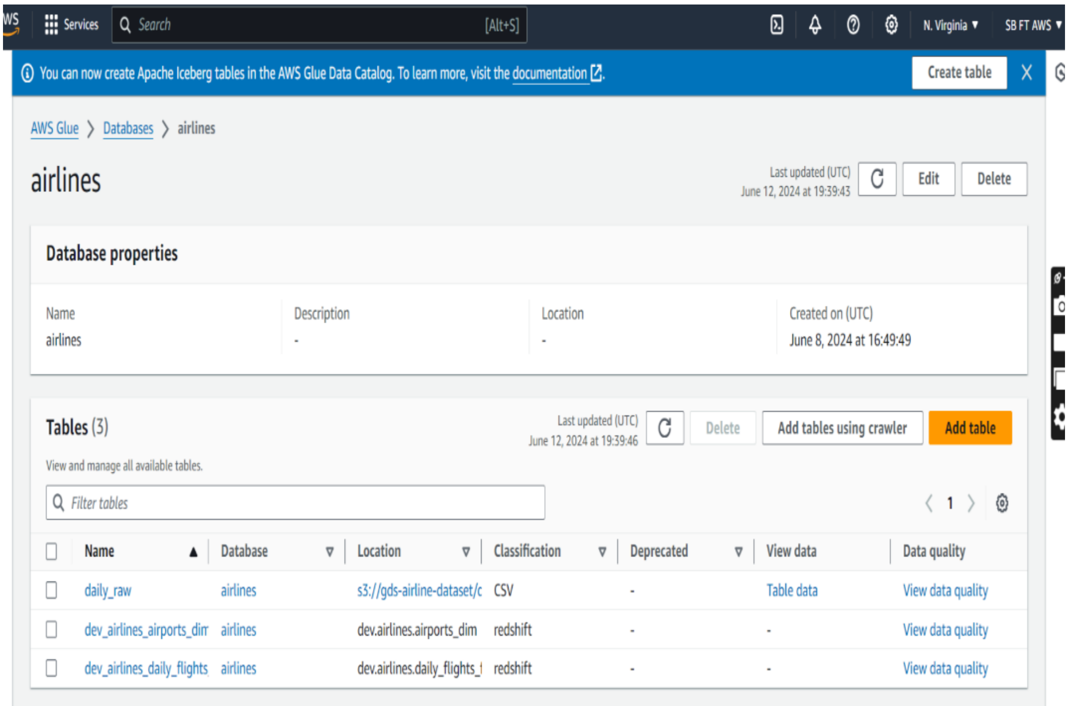
A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data cat

Crawlers (3) Info

Last updated (UTC)
June 12, 2024 at 19:37:41

View and manage all available crawlers.

	Name	State	Schedule	Last run	Last run ti...	Log	Table changes f
□	dim_airport_crawler	Ready		Succeeded	June 8, 2024 a...	View log	1 created
□	fact_flights_destination	Ready		Succeeded	June 8, 2024 a...	View log	1 created
□	raw_daily_file_crawler	Ready		Succeeded	June 12, 2024 ...	View log	-



You can now create Apache Iceberg tables in the AWS Glue Data Catalog. To learn more, visit the documentation.

AWS Glue > Databases > airlines

airlines

Last updated (UTC)
June 12, 2024 at 19:39:43

Database properties

Name	Description	Location	Created on (UTC)
airlines	-	-	June 8, 2024 at 16:49:49

Tables (3)

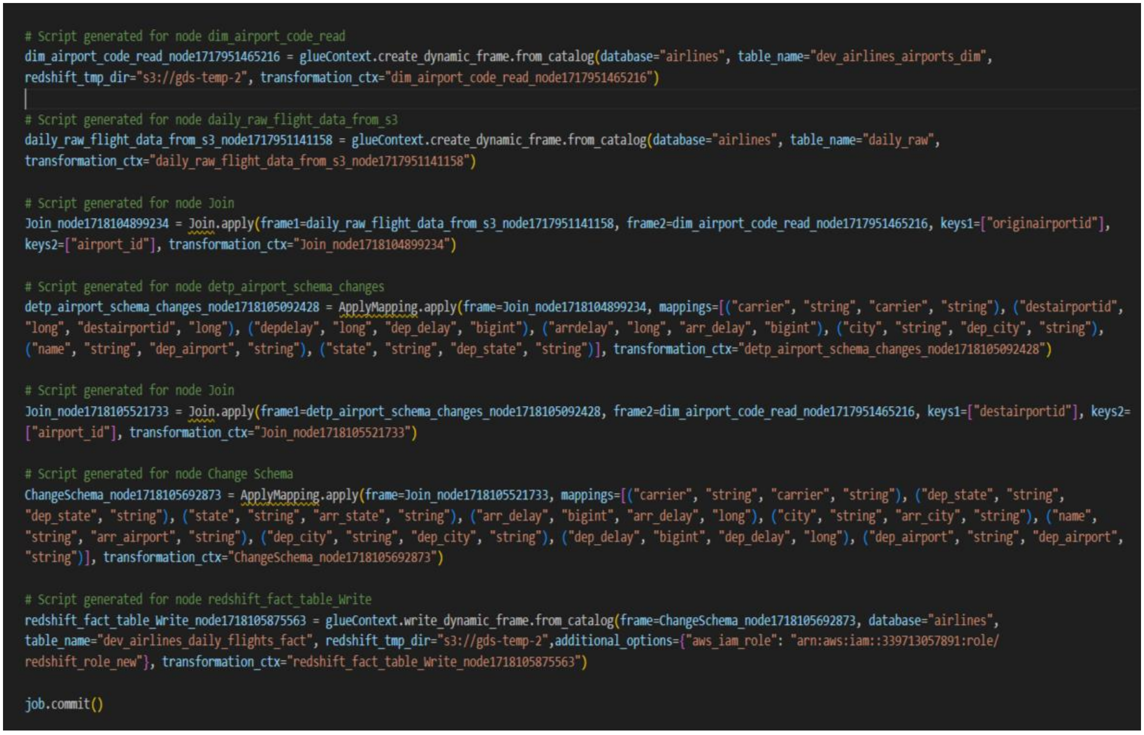
Last updated (UTC)
June 12, 2024 at 19:39:46

View and manage all available tables.

	Name	Database	Location	Classification	Deprecated	View data	Data quality
□	daily_raw	airlines	s3://gds-airline-dataset/c	CSV	-	Table data	View data quality
□	dev_airlines_airports_dim	airlines	dev.airlines.airports_dim	redshift	-	-	View data quality
□	dev_airlines_daily_flights	airlines	dev.airlines.daily_flights_t	redshift	-	-	View data quality

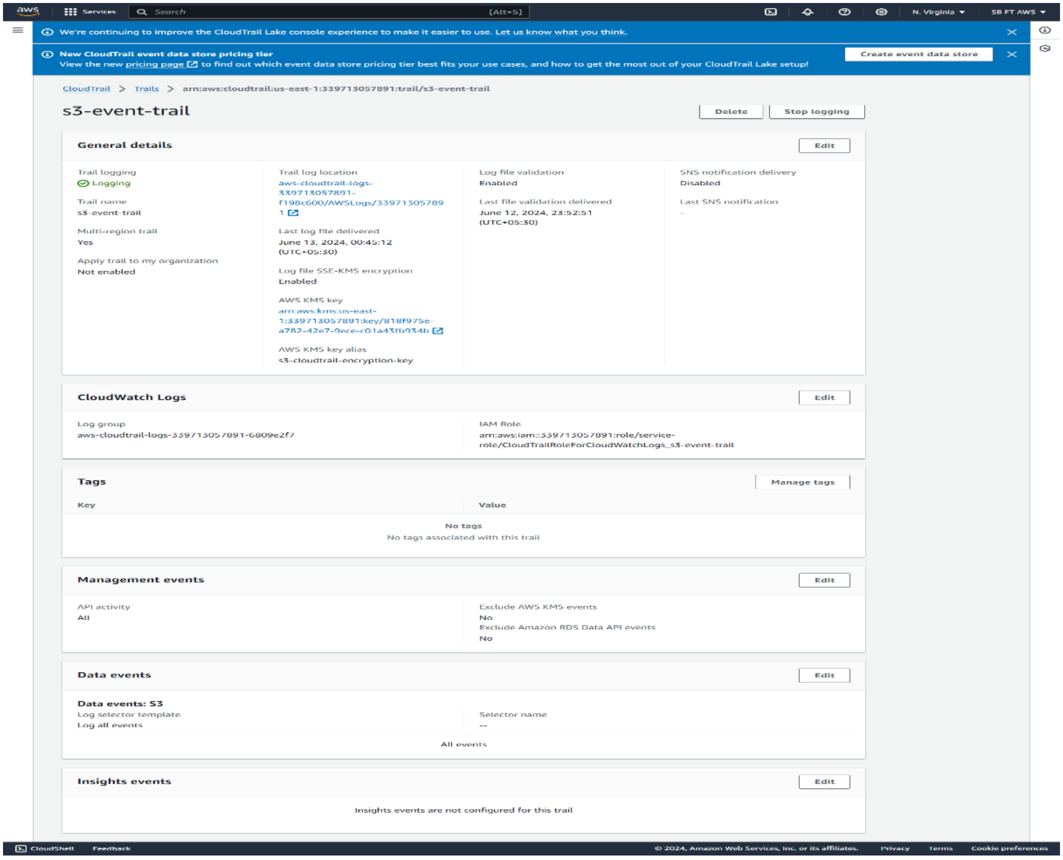
6. we used a visual ETL job 'airline_data_ingestion' where we start reading daily raw data received from glue catalog table 'daily_raw'. We also parallely reading glue catalog dimension table. In the next transformation we performed joining of these two tables and following this joining result we changed the schema same matching with destination redshift table. Once 'change schema' part is done, we wrote the final output to the redshift destination table

'dev_airlines_daily_flights'. We also kept 'Job Bookmark' enabled to receive only new or updated data.



```
# Script generated for node dim_airport_code_read
dim_airport_code_read_node1717951465216 = glueContext.create_dynamic_frame.from_catalog(database="airlines", table_name="dev_airlines_airports_dim",
redshift_tmp_dir="s3://gds-temp-2", transformation_ctx="dim_airport_code_read_node1717951465216")

# Script generated for node daily_raw_flight_data_from_s3
daily_raw_flight_data_from_s3_node1717951141158 = glueContext.create_dynamic_frame.from_catalog(database="airlines", table_name="daily_raw",
transformation_ctx="daily_raw_flight_data_from_s3_node1717951141158")

# Script generated for node Join
Join_node1718104899234 = Join.apply(frame1=daily_raw_flight_data_from_s3_node1717951141158, frame2=dim_airport_code_read_node1717951465216, keys1=["originairportid"],
keys2=["airport_id"], transformation_ctx="Join_node1718104899234")

# Script generated for node detp_airport_schema_changes
detp_airport_schema_changes_node1718105092428 = ApplyMapping.apply(frame=Join_node1718104899234, mappings=[("carrier", "string", "carrier", "string"), ("destairportid",
"long", "destairportid", "long"), ("depdelay", "long", "dep_delay", "bigint"), ("arrdelay", "long", "arr_delay", "bigint"), ("city", "string", "dep_city", "string"),
("name", "string", "dep_airport", "string"), ("state", "string", "dep_state", "string")], transformation_ctx="detp_airport_schema_changes_node1718105092428")

# Script generated for node Join
Join_node1718105521733 = Join.apply(frame1=detp_airport_schema_changes_node1718105092428, frame2=dim_airport_code_read_node1717951465216, keys1=["destairportid"], keys2=
["airport_id"], transformation_ctx="Join_node1718105521733")

# Script generated for node Change Schema
ChangeSchema_node1718105692873 = ApplyMapping.apply(frame=Join_node1718105521733, mappings=[("carrier", "string", "carrier", "string"), ("dep_state", "string",
"dep_state", "string"), ("state", "string", "arr_state", "string"), ("arr_delay", "bigint", "arr_delay", "long"), ("city", "string", "arr_city", "string"), ("name",
"string", "arr_airport", "string"), ("dep_city", "string", "dep_city", "string"), ("dep_delay", "bigint", "dep_delay", "long"), ("dep_airport", "string", "dep_airport",
"string")], transformation_ctx="ChangeSchema_node1718105692873")

# Script generated for node redshift_fact_table_Write
redshift_fact_table_Write_node1718105875563 = glueContext.write_dynamic_frame.from_catalog(frame=ChangeSchema_node1718105692873, database="airlines",
table_name="dev_airlines_daily_flights_fact", redshift_tmp_dir="s3://gds-temp-2",additional_options={"aws_iam_role": "arn:aws:iam::339713057891:role/
redshift_role_new"}, transformation_ctx="redshift_fact_table_Write_node1718105875563")

job.commit()
```

7. We configured CloudTrail data events to log S3 bucket API activity i.e. to get detailed records of actions taken by users, applications, or AWS services. Here S3 events getting passed to cloudtrail and we are receiving API call via Cloudtrail while setting up event bridge rule pattern.



We're continuing to improve the CloudTrail Lake console experience to make it easier to use. Let us know what you think.

New CloudTrail event data store pricing tier
View the new pricing page to find out which event data store pricing tier best fits your use cases, and how to get the most out of your CloudTrail Lake setup!

CloudTrail > Trails > arn:aws:cloudtrail:us-east-1:339713057891:trail/s3-event-trail

s3-event-trail

General details

Trail logging
Logging

Trail name
s3-event-trail

Multi-region trail
Yes

Apply trail to my organization
Not enabled

Trail log location
aws-cloudtrail-logs-339713057891-f198c600/AWSLogs/339713057891

Last log file delivered
June 13, 2024, 00:45:12 (UTC+05:30)

Log file SSE-KMS encryption
Enabled

AWS KMS key
arn:aws:kms:us-east-1:339713057891:key/818f975e-a782-42e7-9ece-c01a43fb934b

AWS KMS key alias
s3-cloudtrail-encryption-key

Log file validation
Enabled

Last file validation delivered
June 12, 2024, 23:52:51 (UTC+05:30)

SNS notification delivery
Disabled

Last SNS notification
-

CloudWatch Logs

Log group
aws-cloudtrail-logs-339713057891-6809e2f7

IAM Role
arn:aws:iam::339713057891:role/service-role/CloudTrailRoleForCloudWatchLogs_s3-event-trail

Tags

Key	Value

No tags
No tags associated with this trail

Management events

API activity
All

Exclude AWS KMS events
No
Exclude Amazon RDS Data API events
No

Data events

Data events: S3
Log selector template
Log all events

Selector name
--

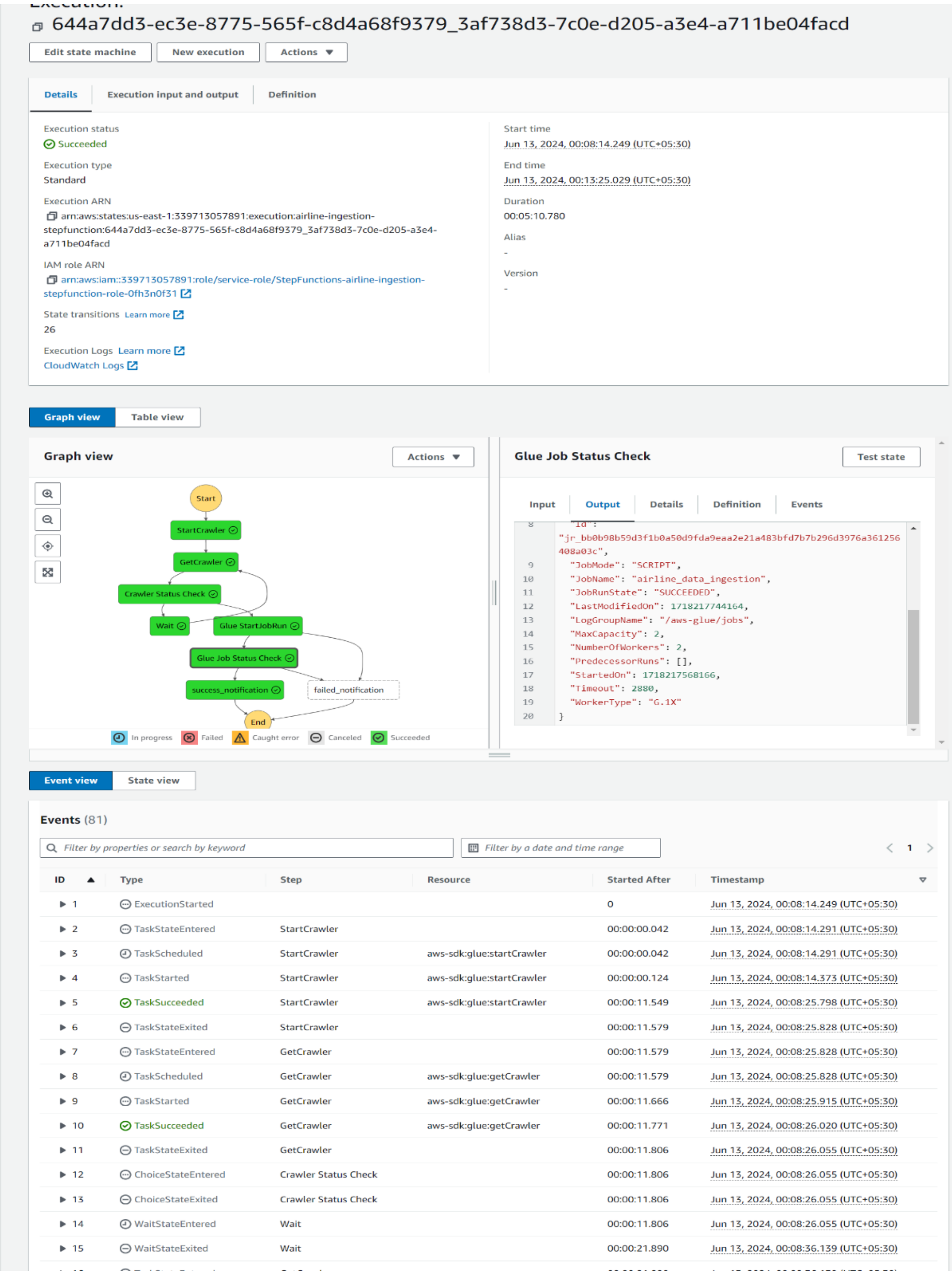
All events

Insights events

Insights events are not configured for this trail

© 2024, Amazon Web Services, Inc. or its affiliates.

8. Further we created 'airline-ingestion-stepfunction' step function to orchestrate multiple steps in your application workflows. As workflow executes, Step Functions tracks which step is being performed and which data is passed between steps. In case of network failure or any other we were able to check that at which point it failed.



644a7dd3-ec3e-8775-565f-c8d4a68f9379_3af738d3-7c0e-d205-a3e4-a711be04facd

Edit state machine | New execution | Actions ▼

Details | Execution input and output | Definition

Execution status
Succeeded

Execution type
Standard

Execution ARN
arn:aws:states:us-east-1:339713057891:execution:airline-ingestion-stepfunction:644a7dd3-ec3e-8775-565f-c8d4a68f9379_3af738d3-7c0e-d205-a3e4-a711be04facd

IAM role ARN
arn:aws:iam::339713057891:role/service-role/StepFunctions-airline-ingestion-stepfunction-role-0fh3n0f31

State transitions Learn more
26

Execution Logs Learn more
CloudWatch Logs

Start time
Jun 13, 2024, 00:08:14.249 (UTC+05:30)

End time
Jun 13, 2024, 00:13:25.029 (UTC+05:30)

Duration
00:05:10.780

Alias
-

Version
-

Graph view | Table view

Graph view

Glue Job Status Check

Test state

Input | Output | Details | Definition | Events

```
  "Id":
"jr_bb0b98b59d3f1b0a50d9fda9eaa2e21a483bfd7b7b296d3976a361256
408a03c",
  "JobMode": "SCRIPT",
  "JobName": "airline_data_ingestion",
  "JobRunState": "SUCCEEDED",
  "LastModifiedOn": 1718217744164,
  "LogGroupName": "/aws-glue/jobs",
  "MaxCapacity": 2,
  "NumberOfWorkers": 2,
  "PredecessorRuns": [],
  "StartedOn": 1718217568166,
  "Timeout": 2880,
  "WorkerType": "G.1X"
}
```

Event view | State view

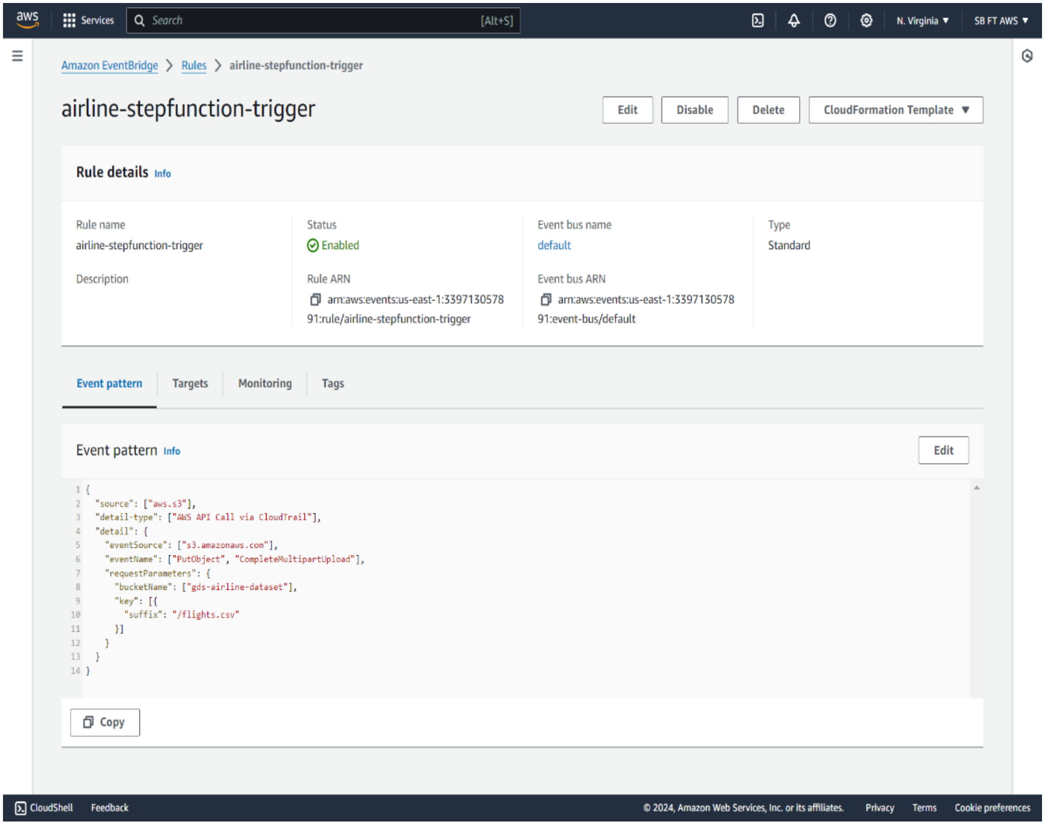
Events (81)

Filter by properties or search by keyword | Filter by a date and time range | 1

ID	Type	Step	Resource	Started After	Timestamp
▶ 1	ExecutionStarted			0	Jun 13, 2024, 00:08:14.249 (UTC+05:30)
▶ 2	TaskStateEntered	StartCrawler		00:00:00.042	Jun 13, 2024, 00:08:14.291 (UTC+05:30)
▶ 3	TaskScheduled	StartCrawler	aws-sdk:glue:startCrawler	00:00:00.042	Jun 13, 2024, 00:08:14.291 (UTC+05:30)
▶ 4	TaskStarted	StartCrawler	aws-sdk:glue:startCrawler	00:00:00.124	Jun 13, 2024, 00:08:14.373 (UTC+05:30)
▶ 5	TaskSucceeded	StartCrawler	aws-sdk:glue:startCrawler	00:00:11.549	Jun 13, 2024, 00:08:25.798 (UTC+05:30)
▶ 6	TaskStateExited	StartCrawler		00:00:11.579	Jun 13, 2024, 00:08:25.828 (UTC+05:30)
▶ 7	TaskStateEntered	GetCrawler		00:00:11.579	Jun 13, 2024, 00:08:25.828 (UTC+05:30)
▶ 8	TaskScheduled	GetCrawler	aws-sdk:glue:getCrawler	00:00:11.579	Jun 13, 2024, 00:08:25.828 (UTC+05:30)
▶ 9	TaskStarted	GetCrawler	aws-sdk:glue:getCrawler	00:00:11.666	Jun 13, 2024, 00:08:25.915 (UTC+05:30)
▶ 10	TaskSucceeded	GetCrawler	aws-sdk:glue:getCrawler	00:00:11.771	Jun 13, 2024, 00:08:26.020 (UTC+05:30)
▶ 11	TaskStateExited	GetCrawler		00:00:11.806	Jun 13, 2024, 00:08:26.055 (UTC+05:30)
▶ 12	ChoiceStateEntered	Crawler Status Check		00:00:11.806	Jun 13, 2024, 00:08:26.055 (UTC+05:30)
▶ 13	ChoiceStateExited	Crawler Status Check		00:00:11.806	Jun 13, 2024, 00:08:26.055 (UTC+05:30)
▶ 14	WaitStateEntered	Wait		00:00:11.806	Jun 13, 2024, 00:08:26.055 (UTC+05:30)
▶ 15	WaitStateExited	Wait		00:00:21.890	Jun 13, 2024, 00:08:36.139 (UTC+05:30)

9. So following that created an event bridge rule 'airline-stepfunction-trigger' with a custom event pattern to trigger our created step function. An event pattern is defined in json format where we are passing bucket name and file name as

suffix in the 'requestParameters'. Next we select step function to trigger as target with event bridge role access to step function.



Amazon EventBridge > Rules > airline-stepfunction-trigger

airline-stepfunction-trigger

Edit
Disable
Delete
CloudFormation Template ▼

Rule details Info

Rule name	Status	Event bus name	Type
airline-stepfunction-trigger	Enabled	default	Standard

Description

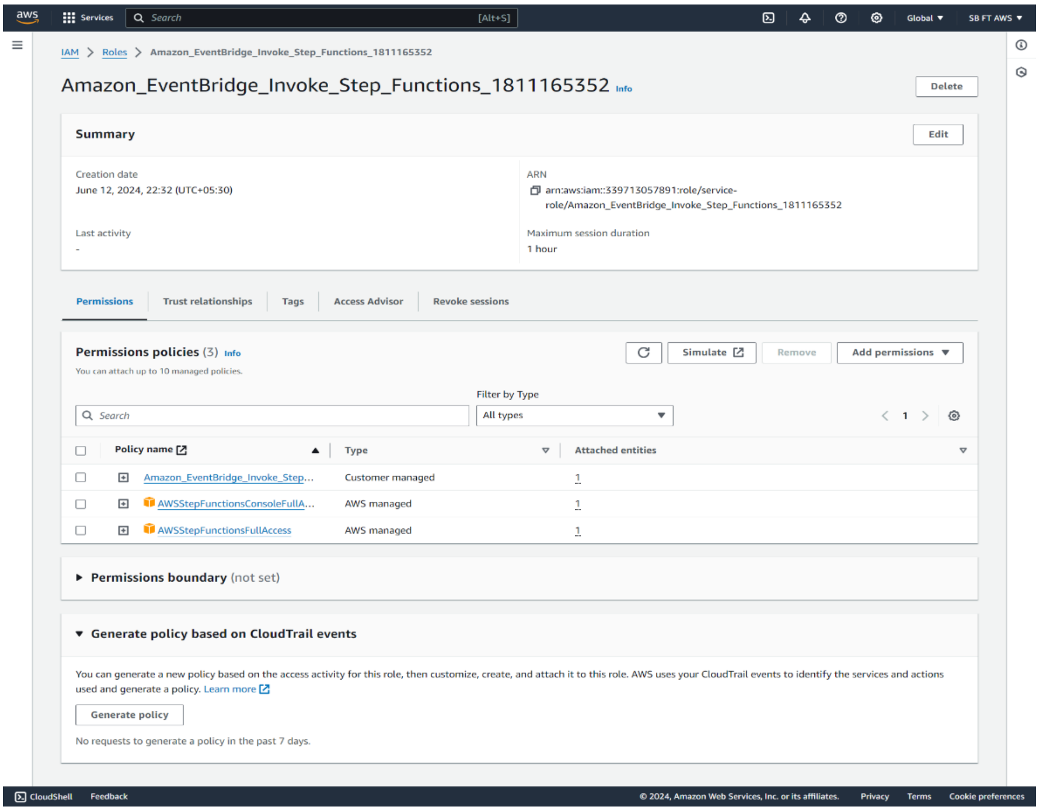
Rule ARN: arn:aws:events:us-east-1:339713057891:rule/airline-stepfunction-trigger

Event bus ARN: arn:aws:events:us-east-1:339713057891:event-bus/default

Event pattern
Targets
Monitoring
Tags

Event pattern Info

```
{
  "source": ["aws.s3"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": {
    "eventSource": ["s3.amazonaws.com"],
    "eventName": ["PutObject", "CompleteMultipartUpload"],
    "requestParameters": {
      "bucketName": ["gds-airline-dataset"],
      "key": [{
        "suffix": "/flights.csv"
      }]
    }
  }
}
```



IAM > Roles > Amazon_EventBridge_Invoke_Step_Functions_1811165352

Amazon_EventBridge_Invoke_Step_Functions_1811165352

Summary

Creation date: June 12, 2024, 22:32 (UTC+05:30)

ARN: arn:aws:iam::339713057891:role/service-role/Amazon_EventBridge_Invoke_Step_Functions_1811165352

Last activity: -

Maximum session duration: 1 hour

Permissions policies (3) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
Amazon_EventBridge_Invoke_Step...	Customer managed	1
AWSStepFunctionsConsoleFullA...	AWS managed	1
AWSStepFunctionsFullAccess	AWS managed	1

Permissions boundary (not set)

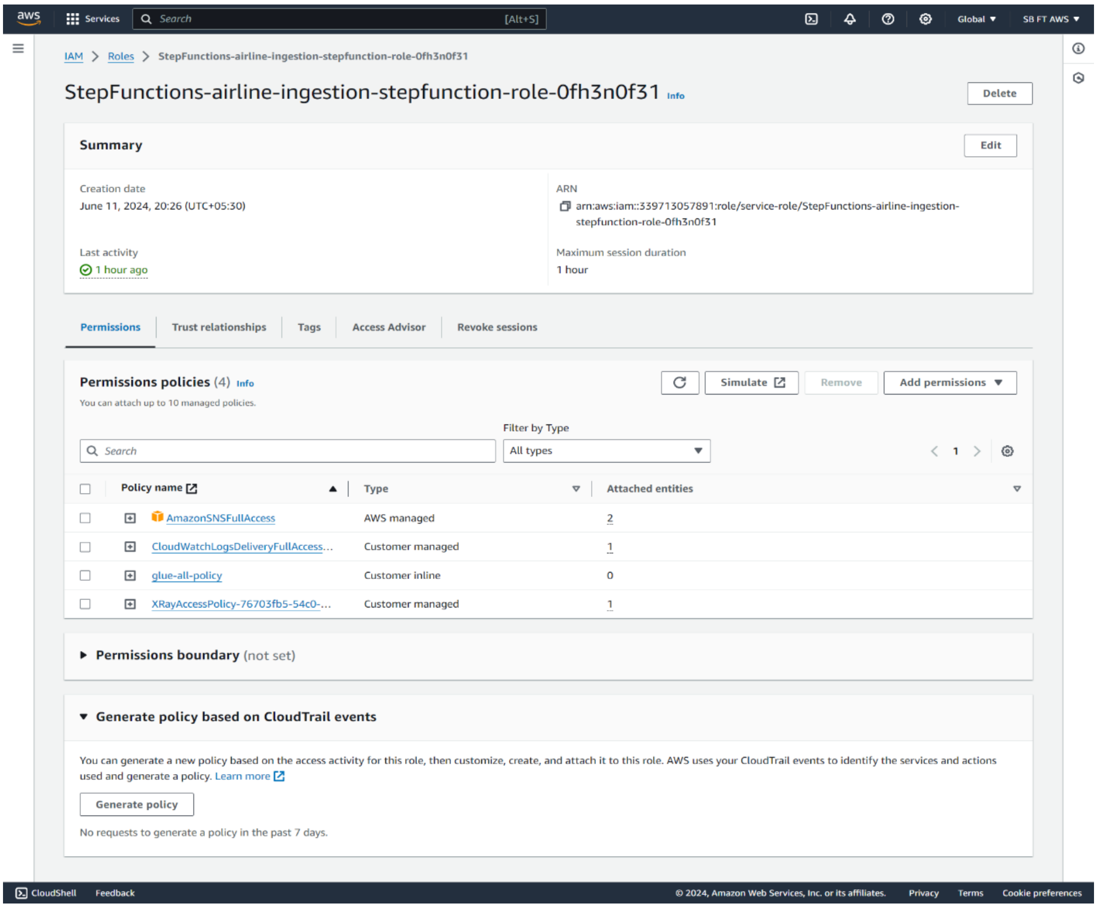
Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

No requests to generate a policy in the past 7 days.

10. On the success step function execution, we also set SNS mail notification in the workflow their to send success alert. Hence, to perform this we set up step

function role access to 'AmazonSNSFullAccess' alongwith some other service permission too such as 'CloudwatchDeliveryFullAccess', glue all policy.



IAM > Roles > StepFunctions-airline-ingestion-stepfunction-role-0fh3n0f31

StepFunctions-airline-ingestion-stepfunction-role-0fh3n0f31 Info

Summary

Creation date
June 11, 2024, 20:26 (UTC+05:30)

ARN
arn:aws:iam::339713057891:role/service-role/StepFunctions-airline-ingestion-stepfunction-role-0fh3n0f31

Last activity
1 hour ago

Maximum session duration
1 hour

Permissions policies (4) Info

You can attach up to 10 managed policies.

□	Policy name	Type	Attached entities
□	AmazonSNSFullAccess	AWS managed	2
□	CloudWatchLogsDeliveryFullAccess...	Customer managed	1
□	glue-all-policy	Customer inline	0
□	XRayAccessPolicy-76703fb5-54c0-...	Customer managed	1

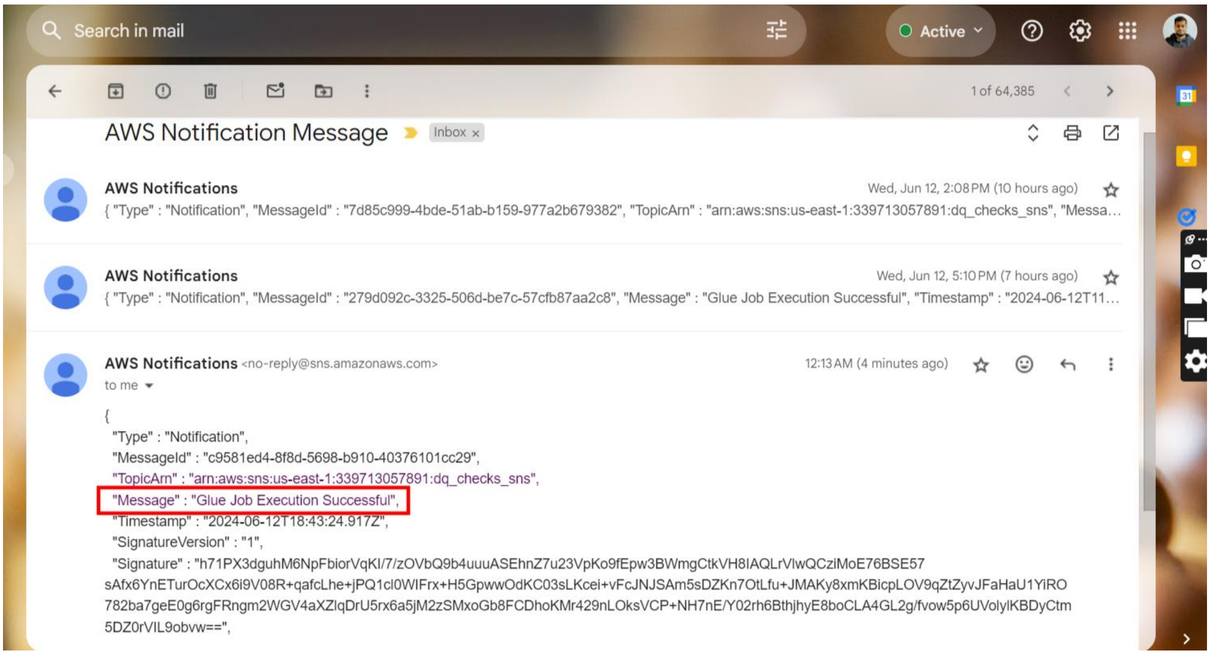
▸ **Permissions boundary** (not set)

▾ **Generate policy based on CloudTrail events**

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

No requests to generate a policy in the past 7 days.

11. That's how we will be getting success notification on success ETL job execution.



AWS Notification Message

AWS Notifications — Wed, Jun 12, 2:08 PM (10 hours ago)
{ "Type" : "Notification", "MessageId" : "7d85c999-4bde-51ab-b159-977a2b679382", "TopicArn" : "arn:aws:sns:us-east-1:339713057891:dq_checks_sns", "Messa…

AWS Notifications — Wed, Jun 12, 5:10 PM (7 hours ago)
{ "Type" : "Notification", "MessageId" : "279d092c-3325-506d-be7c-57cfb87aa2c8", "Message" : "Glue Job Execution Successful", "Timestamp" : "2024-06-12T11…

AWS Notifications <no-reply@sns.amazonaws.com> — 12:13 AM (4 minutes ago)
to me

```
{
  "Type" : "Notification",
  "MessageId" : "c9581ed4-8f8d-5698-b910-40376101cc29",
  "TopicArn" : "arn:aws:sns:us-east-1:339713057891:dq_checks_sns",
  "Message" : "Glue Job Execution Successful",
  "Timestamp" : "2024-06-12T18:43:24.917Z",
  "SignatureVersion" : "1",
  "Signature" : "h71PX3dguhM6NpFbiorVqKl/7/zOVbQ9b4uuuASEhnZ7u23VpKo9fEpw3BWmgCtkVH8IAQLrVlwQCziMoE76BSE57
sAfx6YnETurOcXCx6i9V08R+qafcLhe+jPQ1cl0WIFrx+H5GpwwOdKC03sLKcei+vFcJNJSAm5sDZKn7OtLfu+JMAKy8xmKBicpLOV9qZtZyvJFaHaU1YiRO
782ba7geE0g6rgFRngm2WGV4aXZlqDrU5rx6a5jM2zSMxoGb8FCDhoKMr429nLOksVCP+NH7nE/Y02rh6BthjhyE8boCLA4GL2g/fvow5p6UVolyIKBDyCtm
5DZ0rVIL9obvw==",
```