

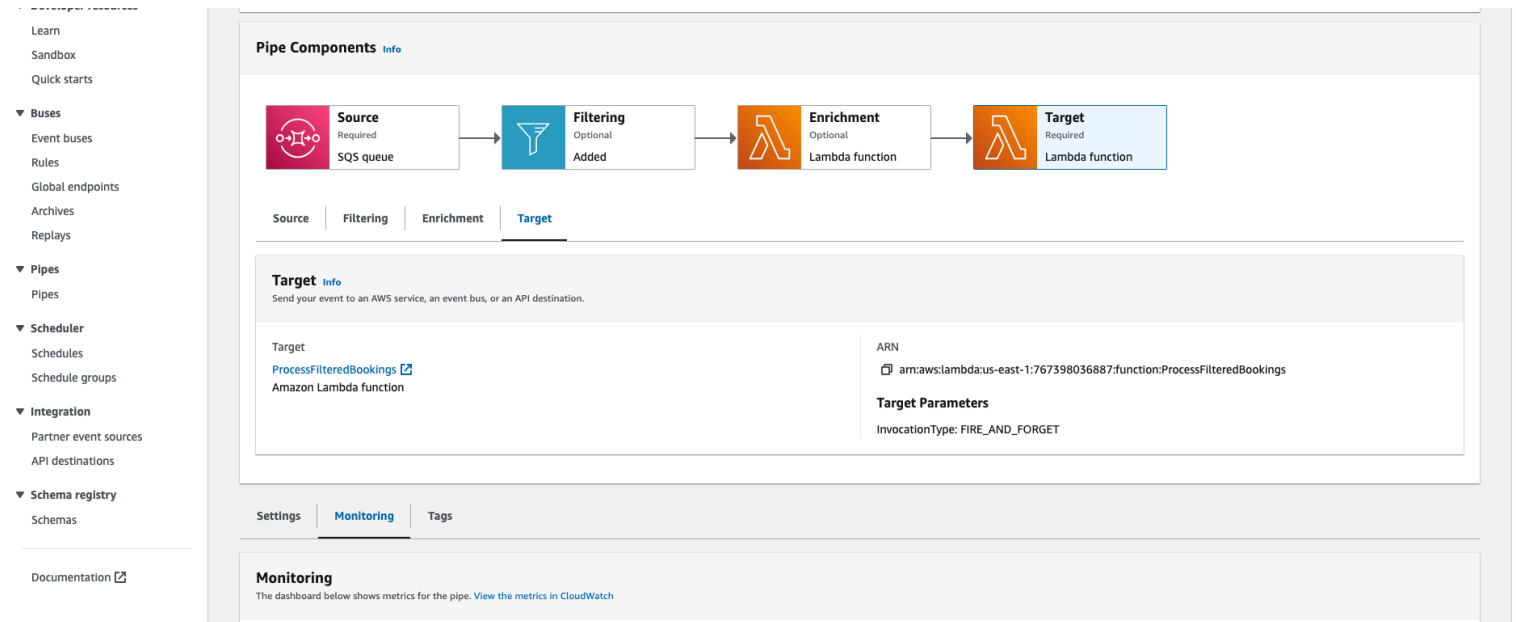
SQS Assignment:

1). Source code repository for lambda functions:

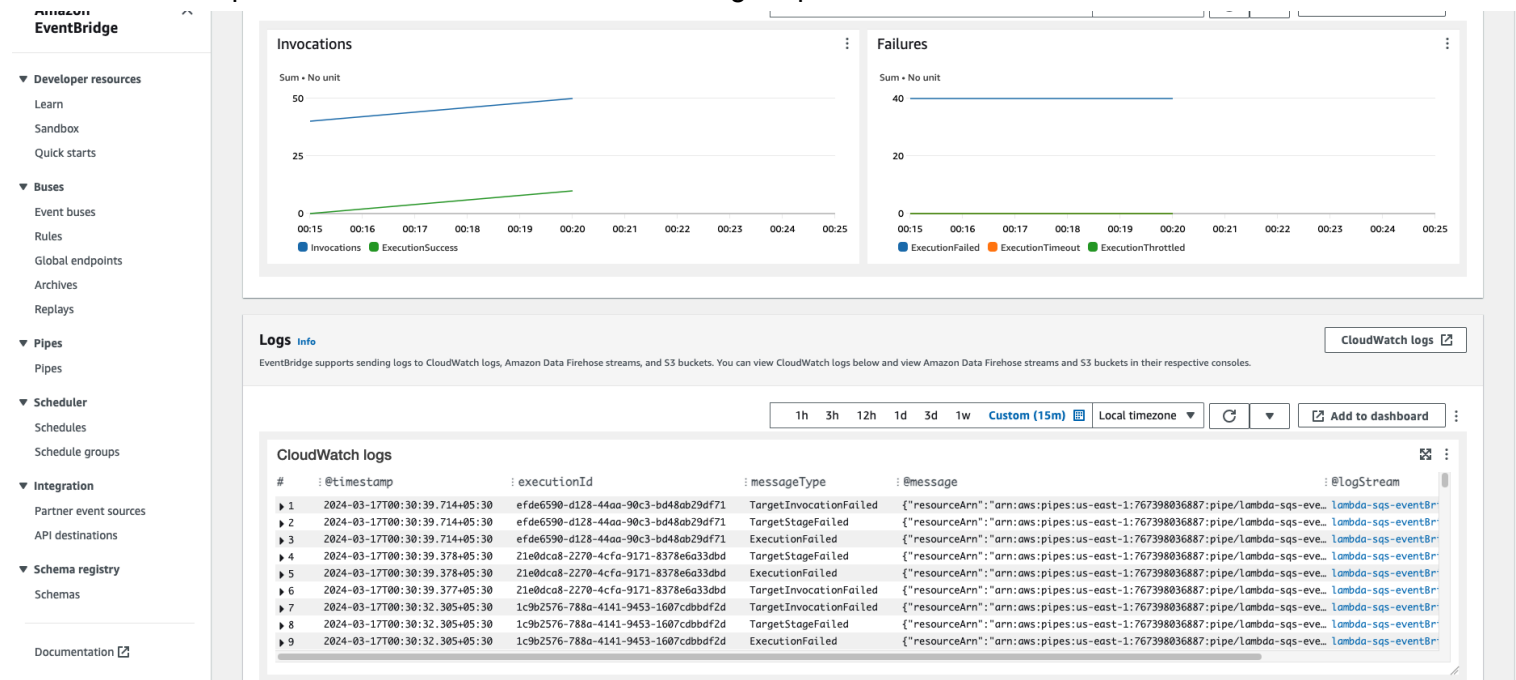
<https://github.com/desininja/AirBnb-Stream-Data-Ingestion>

2). Screenshot of logs demonstrating the successful execution of code:

EventBridge Pipe Flow:



Invocations from producer lambda to SQS to Event Bridge Pipe:



Producer data produced 10 messages:

✕
Successfully updated the function **ProduceAirbnbBookingData**.

Code
Test
Monitor
Configuration
Aliases
Versions

Code source Info

File

Edit

Find

View

Go

Tools

Window

Test

Deploy

Go to Anything (⌘ P)

producer_lambda

Environment Var

Execution result

Upload from

Status: Succeeded | Max memory used: 69 MB | Time: 564.04 ms

Environment

ProduceAirbnbBook

producer_lambda_func

Test Event Name

test1

Response

```
{
  "statusCode": 200,
  "body": "\\Booking details published to S3S1\\\""}
}
```

Function Logs

```
START RequestId: 6360a464-395b-4fde-be73-438a309879a2 Version: SLATEST
{"bookingId": "10b74905d-9e1f-4d2f-b00e-bf4dd87c2cd", "userId": "User86", "propertyId": "Property170", "location": "Washington, Vietnam", "startDate": "2025-03-15", "endDate": "2025-04-06", "bookingId": "907f6354-42bb-4822-9d95-f40c3b153167", "userId": "User84", "propertyId": "Property112", "location": "CaliforniaTexas, Uruguay", "startDate": "2024-11-09", "endDate": "2024-11-09", "bookingId": "8ff8fec2-66ec-4670-b9d9-a26b18d20c49", "userId": "User77", "propertyId": "Property290", "location": "Wisconsin, Luxembourg", "startDate": "2025-01-27", "endDate": "2025-02-08", "bookingId": "caf293cc-ebaa-4524-bfbd-084edae45c7c", "userId": "User58", "propertyId": "Property296", "location": "Virginia, United Arab Emirates", "startDate": "2024-10-07", "endDate": "2024-10-07", "bookingId": "cfa379a3-e990-475e-a63c-dd4dc6047c1a", "userId": "User75", "propertyId": "Property238", "location": "Utah, Zambia", "startDate": "2025-02-14", "endDate": "2025-03-06", "bookingId": "a3464001-981f-4d01-b537-ae790b1f337c", "userId": "User43", "propertyId": "Property97", "location": "Vermont, India", "startDate": "2024-12-13", "endDate": "2025-01-01", "bookingId": "51ac0d1e-0071-49e0-b5c8-4284d23c9a66", "userId": "User58", "propertyId": "Property131", "location": "CaliforniaTexas, Luxembourg", "startDate": "2024-11-06", "endDate": "2024-11-06", "bookingId": "4edf1607-3fbd-4d19-8cda-746643f868d2", "userId": "User4", "propertyId": "Property20", "location": "CaliforniaTexas, Luxembourg", "startDate": "2024-05-13", "endDate": "2024-05-13", "bookingId": "8c157829-a8eb-4712-9892-e77ff07bd9d3", "userId": "User93", "propertyId": "Property23", "location": "New Delhi, Madagascar", "startDate": "2024-06-25", "endDate": "2024-07-15", "bookingId": "58060a03-598f-4f73-b55d-74415b7ab124", "userId": "User67", "propertyId": "Property112", "location": "West Virginia, Zambia", "startDate": "2024-10-29", "endDate": "2024-11-23"}
END RequestId: 6360a464-395b-4fde-be73-438a309879a2
REPORT RequestId: 6360a464-395b-4fde-be73-438a309879a2 Duration: 564.04 ms Billed Duration: 565 ms Memory Size: 128 MB Max Memory Used: 69 MB Init Duration: 360.94 ms
```

Request ID

6360a464-395b-4fde-be73-438a309879a2

Enrichment lambda logs:

CloudWatch

Favorites and recents

Dashboards

Alarms

Logs

Log groups

Log Anomalies

Live Tail

Logs Insights

Metrics

X-Ray traces

Events

Application Signals

Network monitoring

Insights

Settings

Getting Started

What's new

CloudWatch > Log groups > /aws/lambda/Enrichment > 2024/03/16/[\$LATEST]8f19fec1576b4c7588365c4631061e99

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events

Clear 1m 30m 1h 12h Custom Local timezone Display

Timestamp

Message

There are older events to load. [Load more.](#)

2024-03-17T00:22:32.205+05:30

Added 0 index

Added 0 index

Copy

2024-03-17T00:22:32.205+05:30

{'bookingId': 'ee82bbf0-eb29-4325-98cf-5cc039733d19', 'userId': 'User18', 'propertyId': 'Property242', 'location': 'Vermont, Turkey', 'startDate': '2024-03-18', 'endDate': '2024-03-19', 'price': 99.87}

{'bookingId': 'ee82bbf0-eb29-4325-98cf-5cc039733d19', 'userId': 'User18', 'propertyId': 'Property242', 'location': 'Vermont, Turkey', 'startDate': '2024-03-18', 'endDate': '2024-03-19', 'price': 99.87}

Copy

2024-03-17T00:22:32.206+05:30

END RequestId: 88c76e27-bb86-4607-ab0c-2f721c95f920

END RequestId: 88c76e27-bb86-4607-ab0c-2f721c95f920

Copy

2024-03-17T00:22:32.206+05:30

REPORT RequestId: 88c76e27-bb86-4607-ab0c-2f721c95f920 Duration: 1.54 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 33 MB

REPORT RequestId: 88c76e27-bb86-4607-ab0c-2f721c95f920 Duration: 1.54 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 33 MB

Copy

2024-03-17T00:22:39.296+05:30

START RequestId: 8a0f5cdd-5e41-4f35-980f-bfe8cb43f354 Version: \$LATEST

START RequestId: 8a0f5cdd-5e41-4f35-980f-bfe8cb43f354 Version: \$LATEST

Copy

2024-03-17T00:22:39.297+05:30

Printing event as it is:

Printing event as it is:

Copy

2024-03-17T00:22:39.297+05:30

[{"bookingId": "15dc7700-6439-4007-a9f4-f0c8f7085c8d", "userId": "User49", "propertyId": "Property294", "location": "New Delhi, Malaysia", "startDate": "2024-05-27", "endDate": "2024-05-27", "price": 100.00}]

[{"bookingId": "15dc7700-6439-4007-a9f4-f0c8f7085c8d", "userId": "User49", "propertyId": "Property294", "location": "New Delhi, Malaysia", "startDate": "2024-05-27", "endDate": "2024-05-27", "price": 100.00}]

Copy

Final files in S3 location after Target lambda is processed:

airbnb-booking-records-target [info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

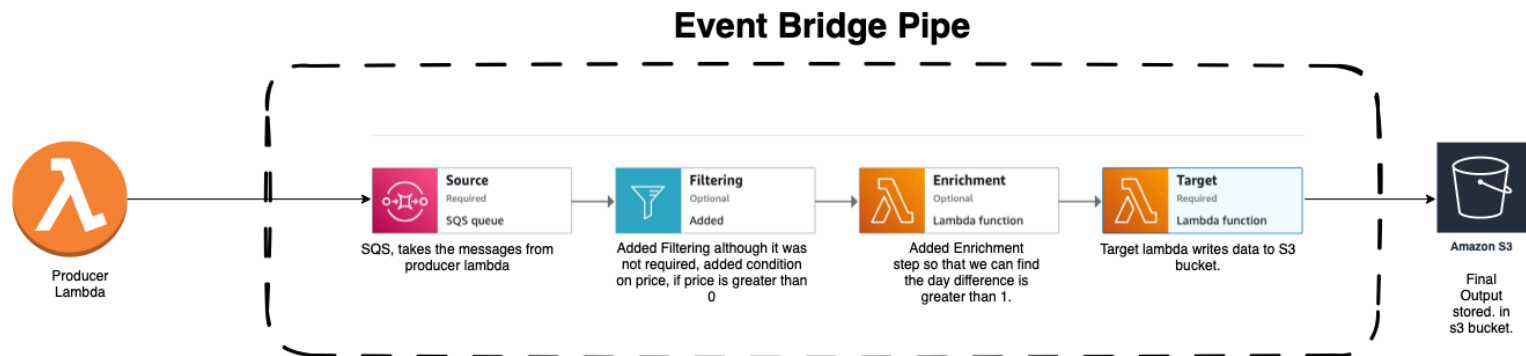
Objects (10) [info](#)

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 access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	51ac0d1e-0071-49ea-b5c8-4284d23c95a6.json	json	March 17, 2024, 00:22:03 (UTC+05:30)	214.0 B	Standard
<input type="checkbox"/>	cfa379a3-eb90-4756-a63c-dd4dc6047c1a.json	json	March 17, 2024, 00:22:03 (UTC+05:30)	199.0 B	Standard
<input type="checkbox"/>	0b74905d-9e1f-4d2f-b0be-1bfa8ed7c2cd.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	206.0 B	Standard
<input type="checkbox"/>	4edf1607-3fbd-4d19-8cda-746643f868d2.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	212.0 B	Standard
<input type="checkbox"/>	58060a03-598f-4f73-b55d-74415b7ab124.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	208.0 B	Standard
<input type="checkbox"/>	8c157829-a8eb-4712-9892-e77ff07bd9d3.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	206.0 B	Standard
<input type="checkbox"/>	8ff8fec2-66ec-4670-b09d-a26b18d20c49.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	208.0 B	Standard
<input type="checkbox"/>	90776354-42bb-4822-9a95-f40c5b1251d7.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	210.0 B	Standard
<input type="checkbox"/>	a3464801-981f-4db1-b537-ae7996b61f37c.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	200.0 B	Standard
<input type="checkbox"/>	caf293cc-ebaa-4524-bfbd-084edae45c7c.json	json	March 17, 2024, 00:22:02 (UTC+05:30)	216.0 B	Standard

3). Architecture Diagram:



As you can see in the above architecture, the producer lambda generates the mock data that is sent to SQS which act as a source for Event Bridge Pipe and passes that message to filtering stage(although this filtering stage was not required, but added it and filtered for price, if price is greater than 0. Then the Enrichment step where the difference between start time and end time is calculated and filtered if the difference is greater than 1. After that the filtered messages are passed to Target lambda which simply writes the received data to S3 bucket in json format.

4). Reflection on this assignment:

Initially I got confused on how to calculate the difference between starttime and endtime in the filter of EventBridge Pipe but when I revisited the session, Shashank clearly mentioned that for any transformation we can use an enrichment segment and apply a lambda for transformation so I applied it.

Later on I also faced a challenge with parsing event json for enrichment, I thought the event payload would be a json but it was a list and inside the list the json was present, and then for target the input was a string instead of dictionary type or json object. So I converted them into json/dictionary format at respective stages.

After doing all this, I achieved the objective and files got created in S3 bucket.

I still have some doubt regarding the invocation logs. I will ask them in the next session.