Project 6: Calendly Booking Data Analytics

Overview

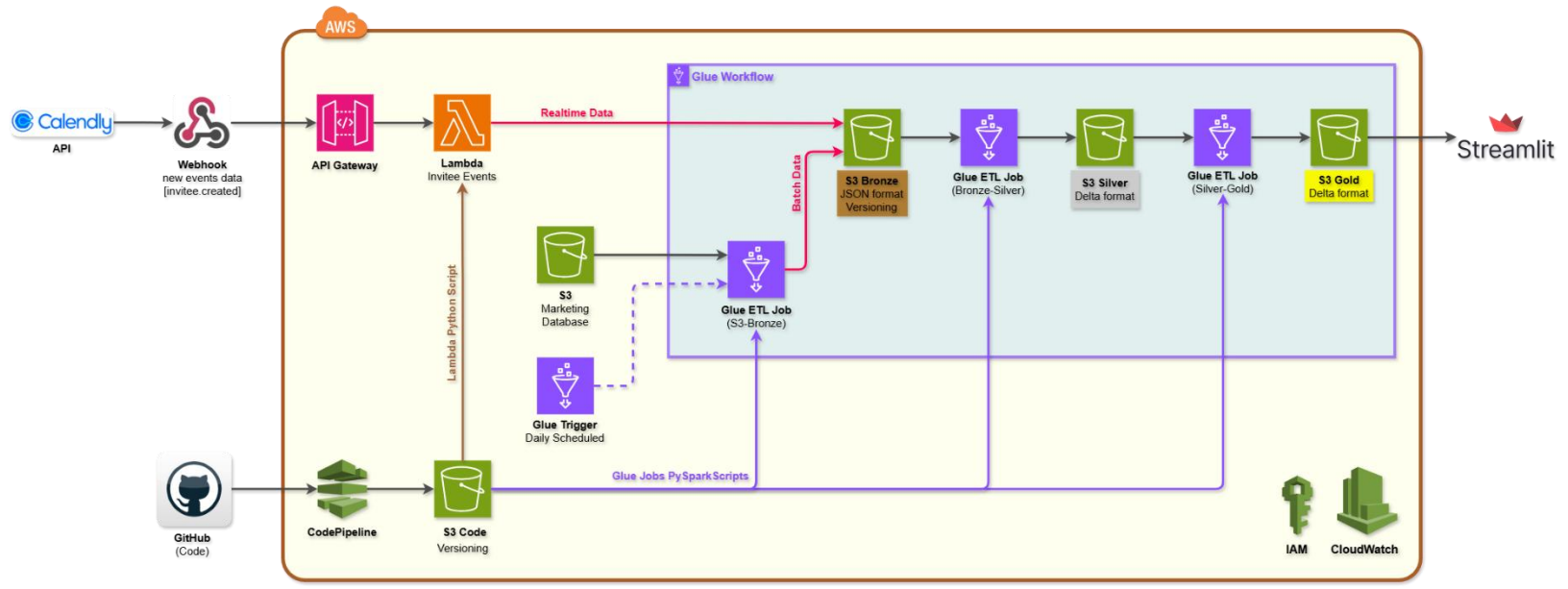
In this project, we work with JSON-formatted data collected from two primary sources: real-time booking (invitee) events from the Calendly platform via Webhooks, and batch marketing cost data retrieved from a publicly available database stored in the DEA S3 bucket. The goal is to integrate these datasets, generate meaningful business metrics, and provide insights that help optimize marketing spend and evaluate campaign conversion rates.

This project implements a hybrid data engineering pipeline that combines real-time, event-driven ingestion with batch data processing. Newly created booking events are captured through Calendly Webhooks, while daily marketing cost data is ingested on a scheduled basis. All data is processed using a fully serverless AWS architecture to ensure scalability, reliability, and cost efficiency.

Real-time booking events are delivered in HTTP format from the Calendly Webhook to an AWS API Gateway endpoint, which triggers a Lambda function. These raw events are stored in the S3 Bronze bucket in their original JSON format. In parallel, an AWS Glue ETL (Spark) job is scheduled to ingest daily marketing cost data from the DEA public S3 bucket and store it in a dedicated folder within the S3 Bronze layer.

A Bronze–Silver–Gold data modeling approach is applied to ensure clean, structured, and analytics-ready datasets. Two additional Glue ETL jobs perform data transformation and aggregation, progressively refining the data and loading it into the Silver and Gold S3 buckets. The Gold layer contains curated datasets optimized for analytics and reporting.

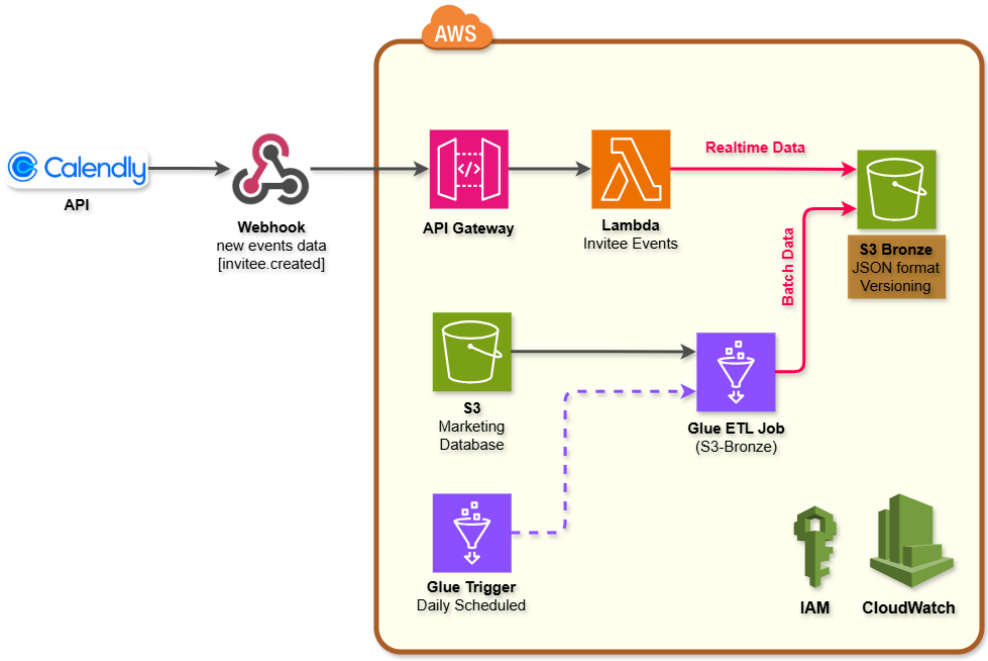
Finally, Streamlit is used to visualize key business KPIs and performance metrics through an interactive dashboard. AWS CodePipeline automates the deployment process by pulling Python and PySpark scripts from a GitHub repository and deploying them to AWS Lambda and Glue jobs. AWS Glue Triggers handle daily scheduling, while AWS Glue Workflows orchestrate the end-to-end pipeline, resulting in a scalable, modular, and production-ready data architecture aligned with modern data engineering best practices.



Step 1 – Data Ingestion

1-1- Objective

The objective of this step is to ingest raw invitee event files from Calenly API Webhook, in **realtime format**, and also daily marketing cost data from the DEA public S3 bucket, in the **batch format**, into an Amazon S3 bucket, named **project6-bronze-bucket**, in the JSON format. To comply with the project requirement that allows me to use Python, I will implement **AWS Lambda** for realtime data ingestion, via **AWS API Gateway**. Moreover, I used **AWS Glue ETL** job for batch data ingestion. This hybrid approach ensures that the ingestion process is both scalable and aligned with other AWS services in orchestration.



1-2- Landing Storage (Bronze Layer)

I used Amazon S3 service and create a bucket with name of **s3://project6-bronze-bucket/**. After that, I created a folder inside that bucket with the name of **events/** folder for ingestion of new raw invitee event files. This is a sample of **event** files:


```
{
  "created_at": "2025-07-09T06:04:34.000000Z",
  "created_by": "https://api.calendly.com/users/8ec9d4df-39c5-463a-8b40-bcb5d0a70edf",
  "event": "invitee.created",
  "payload": {
    "cancel_url": "https://calendly.com/cancellations/22a0f2d6-1bde-4fc1-95c1-d969df1da21d",
    "created_at": "2025-07-09T06:04:33.761871Z",
    "email": "chrisgarzon19@gmail.com",
    "event": "https://api.calendly.com/scheduled_events/1ac9e88e-eae3-4e4b-b979-d770cff02d72",
    "first_name": null,
    "invitee_scheduled_by": "https://api.calendly.com/users/8ec9d4df-39c5-463a-8b40-bcb5d0a70edf",
    "last_name": null,
    "name": "Ninad Magdum",
    "new_invitee": null,
    "no_show": null,
```

```
    "old_invitee": null,
    "payment": null,
    "questions_and_answers": [
      {
        "answer": "+91 91303 02575",
        "position": 0,
        "question": "What is your phone number?"
      }
    ],
    "reconfirmation": null,
    "reschedule_url": "https://calendly.com/reschedulings/22a0f2d6-1bde-4fc1-95c1-d969df1da21d",
    "rescheduled": false,
    "routing_form_submission": null,
    "scheduled_event": {
      "created_at": "2025-07-09T06:04:33.743275Z",
      "end_time": "2025-07-09T18:00:00.000000Z",
      "event_guests": [],
      "event_memberships": [
        {
          "user": "https://api.calendly.com/users/76e8f2b2-b38c-41c8-826f-4b61f2ba22ba",
          "user_email": "zan@dataengineeracademy.com",
          "user_name": "Zan Strmec"
        }
      ],
      "event_type": "https://api.calendly.com/event_types/d639ecd3-8718-4068-955a-436b10d72c78",
      "invitees_counter": {
        "total": 1,
        "active": 1,
        "limit": 1
      },
      "location": {
        "location": null,
        "type": "custom"
      },
      "meeting_notes_html": null,
      "meeting_notes_plain": null,
      "name": "Data Engineer Academy Info Session",
      "start_time": "2025-07-09T17:45:00.000000Z",
      "status": "active",
      "updated_at": "2025-07-09T06:04:33.743275Z",
      "uri": "https://api.calendly.com/scheduled_events/1ac9e88e-eae3-4e4b-b979-d770cff02d72"
    },
    "scheduling_method": null,
    "status": "active",
    "text_reminder_number": null,
    "timezone": "Asia/Calcutta",
    "tracking": {
      "utm_campaign": null,
      "utm_source": null,
      "utm_medium": null,
      "utm_content": null,
      "utm_term": null,
      "salesforce_uuid": null
    },
    "updated_at": "2025-07-09T06:04:33.761871Z",
    "uri": "https://api.calendly.com/scheduled_events/1ac9e88e-eae3-4e4b-b979-d770cff02d72/invitees/22a0f2d6-
1bde-4fc1-95c1-d969df1da21d"
  }
}
```

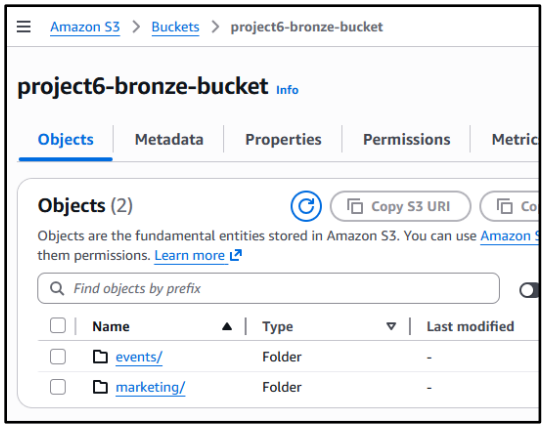

Then, I created another folder inside the bronze bucket with the name of **marketing/** folder for ingestion of daily marketing cost. This is a sample of **marketing** files:

https://dea-data-bucket.s3.us-east-1.amazonaws.com/calendly_spend_data/file_index.json

https://dea-data-bucket.s3.us-east-1.amazonaws.com/calendly_spend_data/spend_data_2026-01-13.json

The first file is used to read the file names and use it as per availability. The second file is as a sample for today. This is a sample data for the second file:

```
[
  {
    "date": "2025-06-24",
    "channel": "facebook_paid_ads",
    "spend": 653.28
  },
  {
    "date": "2025-06-24",
    "channel": "youtube_paid_ads",
    "spend": 487.59
  },
  {
    "date": "2025-06-24",
    "channel": "tiktok_paid_ads",
    "spend": 345.12
  }
]
```

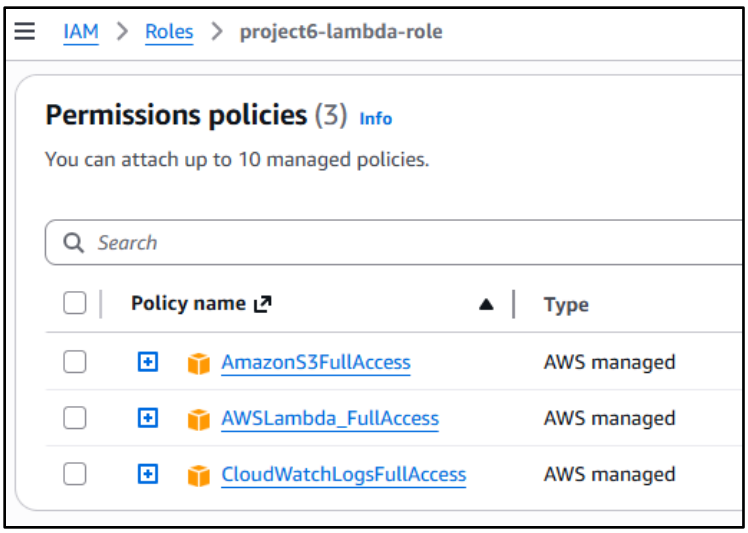



1-3- IAM Roles

Before starting data ingestion using Lambda, I configured an IAM role to enable Lambda to securely access the required AWS services during job execution. The IAM role was named **project6-lambda-role** and tagged with **project: 6**, and it was attached to the AWS Lambda functions to ensure secure and authorized access during execution.

The following permissions were configured:

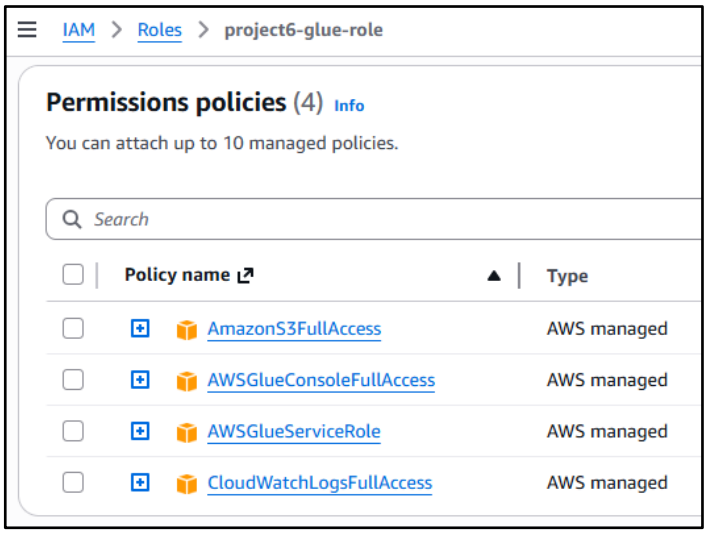
- **AmazonS3FullAccess:** Granted to allow Lambda to write to the Bronze S3 bucket.
- **AmazonLambda_FullAccess:** Granted to allow creation, execution, and management of Lambda functions involved in webhook ingestion and lead enrichment workflows.
- **CloudWatchLogsFullAccess:** Enabled logging and monitoring of Lambda function execution in Amazon CloudWatch.



Also, I configured another IAM role to enable AWS Glue jobs to securely access the required AWS services during job execution. The IAM role was named **project6-glue-role** and tagged with **project: 6**, and it was attached to the AWS Glue ETL jobs to ensure secure and authorized access during execution.

The following permissions were configured:

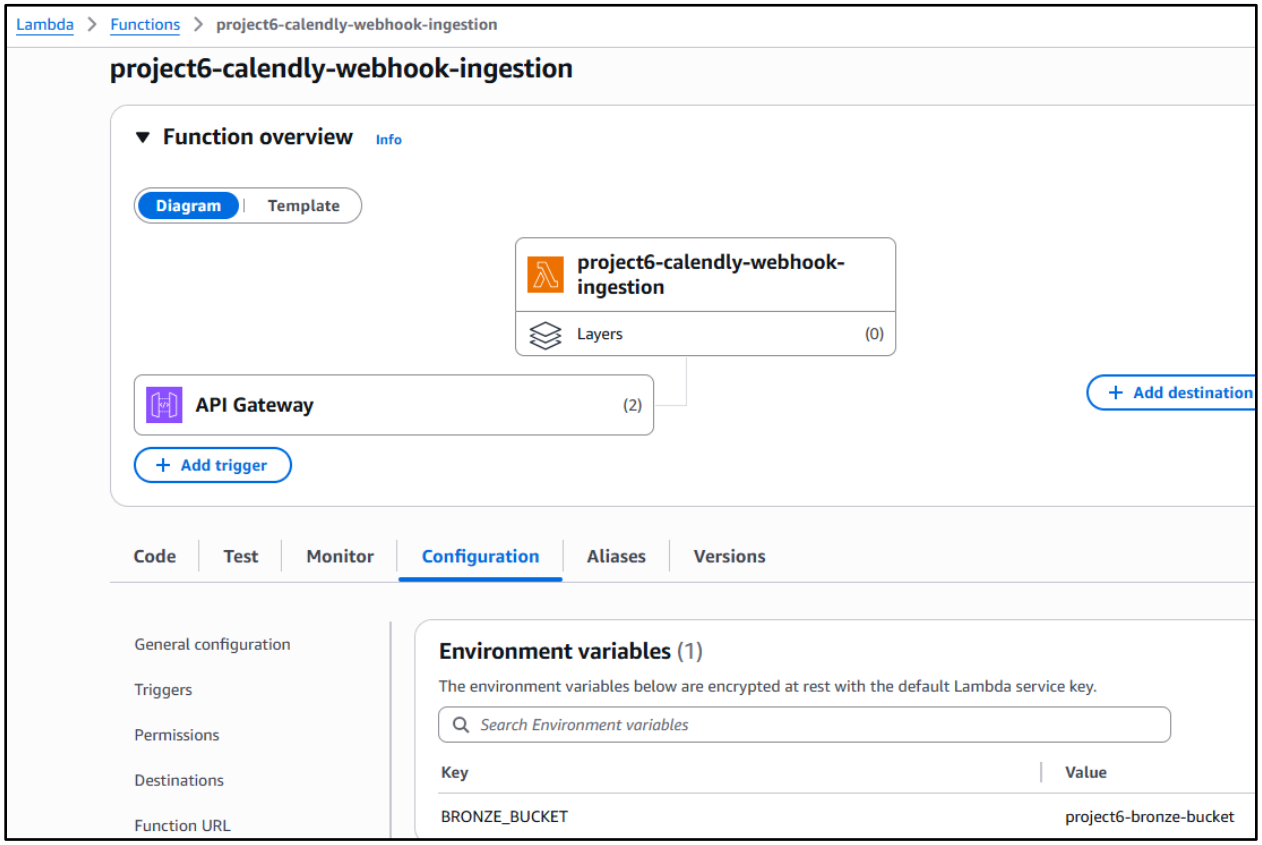
- **AmazonS3FullAccess:** Granted to allow AWS Glue to read from and write to the Bronze, Silver, and Gold S3 buckets.
- **CloudWatchLogsFullAccess:** Enabled logging and monitoring of Glue job execution in Amazon CloudWatch.
- **AWSGlueConsoleFullAccess:** Attached to my IAM user to allow me to create and manage AWS Glue resources, such as ETL jobs, and workflows, through the AWS Management Console.
- **AWSGlueServiceRole:** Attached to the Glue job runtime role, allowing AWS Glue to assume the role at execution time and access required services, including Amazon S3, AWS Secrets Manager, and Amazon CloudWatch Logs resources.



1-4- Lambda Function (Realtime Ingestion)

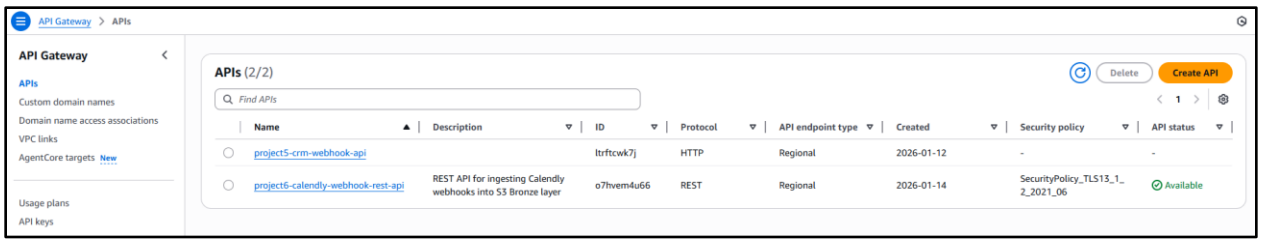
I created the Lambda function, named **project6-calendly-webhook-ingestion**, to perform raw data ingestion from Calendly Webhook API into S3 Bronze bucket. The script file name in the Python language is **project6-calendly-webhook-ingestion.py**. I will use this function later in CI/CD section. I added one environment variable to this function as Bronze bucket name.

```
Key: BRONZE_BUCKET
Value: project6-bronze-bucket
```



1-5- API Gateway

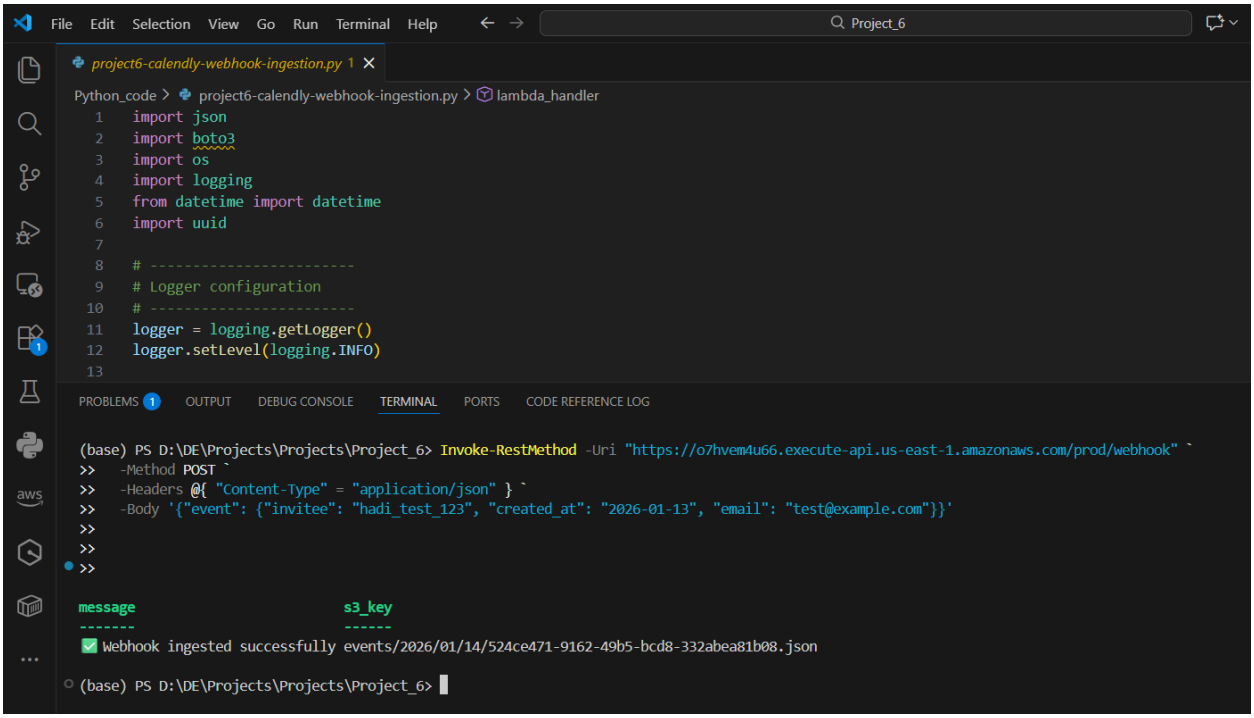
To get events from Calendly Webhook, in the realtime format, I created a REST API Gateway, named **project6-calendly-webhook-rest-api**, to fetch data from Calendly Webhook. Then, I created an URL stage in the AWS using API Gateway POST method. It will post events directly to my URL. This is my URL address: **https://o7hvem4u66.execute-api.us-east-1.amazonaws.com/prod/webhook**. I use this URL in Calendly Webhook configuration.

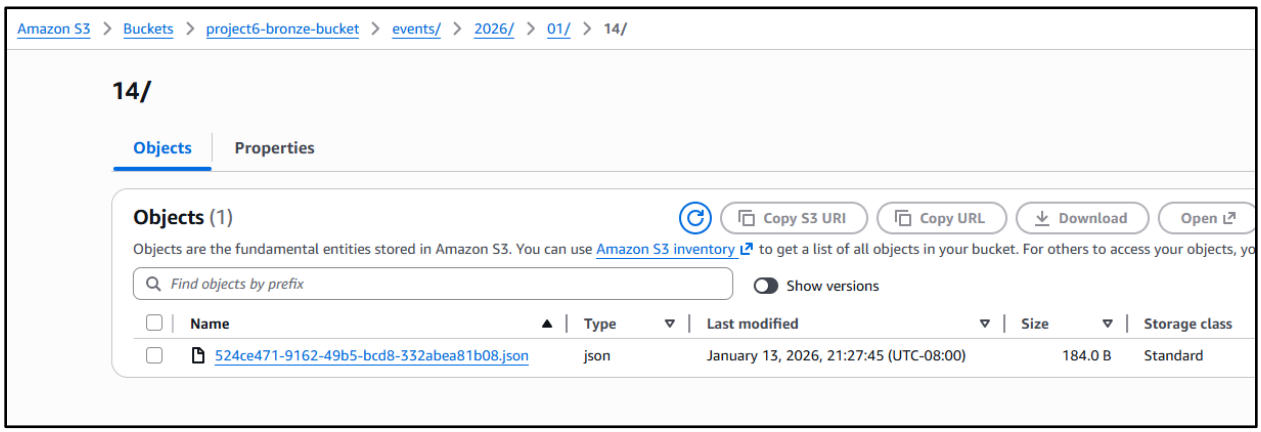


Now, to test my API Gateway, I am going to send a fake event manually to my URL in VS Code terminal:

```
Invoke-RestMethod -Uri "https://o7hvem4u66.execute-api.us-east-1.amazonaws.com/prod/webhook" `
  -Method POST `
  -Headers @{ "Content-Type" = "application/json" } `
  -Body '{"event": {"invitee": "hadi_test_123", "created_at": "2026-01-13", "email": "test@example.com"}}'
```

It works successfully!





1-6- Glue ETL Job (S3-Bronze)

Step 2 – Transformation