

CLOUD COMPUTING

Deploy Sales ETL as a Serverless Function

Goal:

Deploy the output of your ETL pipeline as a cloud-based analytics API.

Tasks:

Use the ETL output stored in AWS S3 or Azure Blob Storage.

Create an AWS Lambda or Azure Function that:

Reads aggregated sales data.

Runs KPI calculations (e.g., total revenue, top region).

Returns results as JSON via API Gateway / Azure HTTP Trigger.

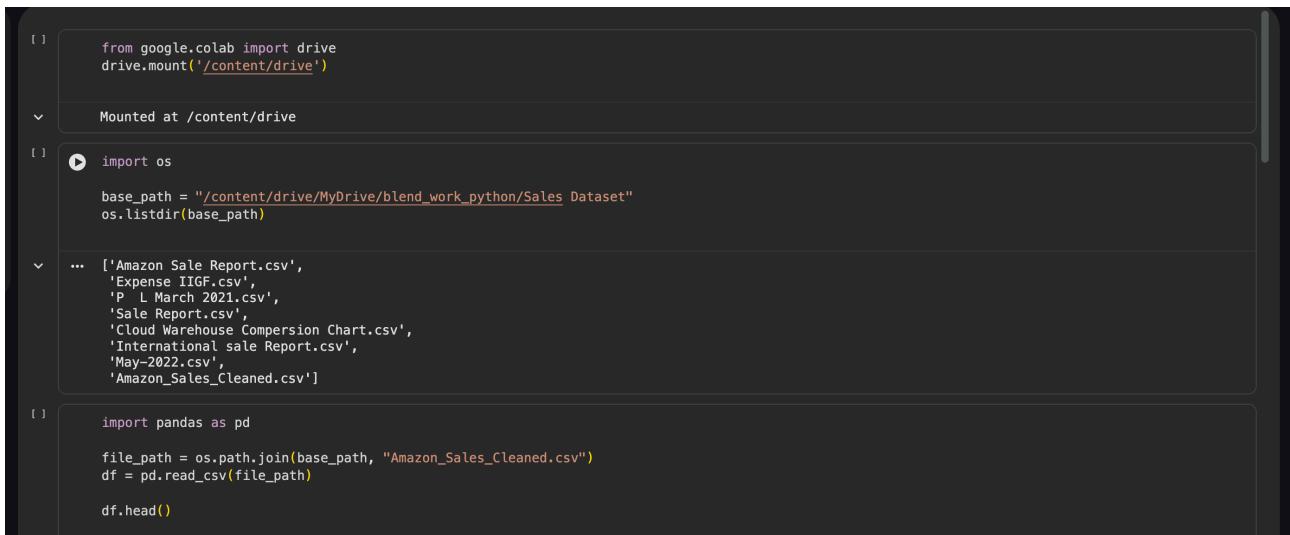
Configure a scheduled trigger (e.g., CloudWatch/EventBridge).

Log execution results to CloudWatch or Application Insights.

Tech: AWS Lambda / Azure Functions, Boto3 / Azure SDK

Deliverables: Deployed function link + code repo + screenshots (Lambda/Function console, API JSON response, logs)

Dataset: https://wgcp-my.sharepoint.com/:f/g/personal/ritish_jogi_blend360_com/EmxzwFjNkaxPuCw2mQ0abr0BGg6XzIPlj22VogFVtQniyg?e=agISyr



```
from google.colab import drive
drive.mount('/content/drive')

import os

base_path = "/content/drive/MyDrive/blend_work_python/Sales Dataset"
os.listdir(base_path)

['Amazon Sale Report.csv',
 'Expense II GF.csv',
 'P_L March 2021.csv',
 'Sale Report.csv',
 'Cloud Warehouse Comprision Chart.csv',
 'International sale Report.csv',
 'May-2022.csv',
 'Amazon_Sales_Cleaned.csv']

import pandas as pd

file_path = os.path.join(base_path, "Amazon_Sales_Cleaned.csv")
df = pd.read_csv(file_path)

df.head()
```

		index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	ship-state	ship-postal-code	ship-country	promotion-ids	B2B	fu
0	0	405-5731545	8078784-5731545	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	SET389-KR-NP-S	SET389-KR-NP-S	Set	...	MAHARASHTRA	400081.0	IN	IN Core Free Shipping 2015/04/08 23-48-5-108		
1	1	171-1101146	9198151-1101146	2022-04-30	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	KARNATAKA	560085.0	IN	Amazon PLCC Free-Financing Universal Merchant ...		
2	2	404-7273146	0687676-7273146	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	MAHARASHTRA	410210.0	IN	IN Core Free Shipping 2015/04/08 23-48-5-108	True	
3	3	403-8133951	9615377-8133951	2022-04-30	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress	...	PUDUCHERRY	605008.0	IN	IN Core Free Shipping 2015/04/08 23-48-5-108		
4	4	407-7240320	1069790-7240320	2022-04-30	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL	Top	...	TAMIL NADU	600073.0	IN	IN Core Free Shipping 2015/04/08 23-48-5-108		

5 rows x 27 columns

ETL WORK AGGREGATIONS AND KPI

```
[ ] #etl work aggregation
#clean/transform
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df['Quarter'] = df['Date'].dt.to_period('Q')

[ ] #kpi aggregation
total_revenue = df['Revenue'].sum()

region_revenue = (
    df.groupby('ship-state')['Revenue']
    .sum()
    .reset_index()
)

top_region = region_revenue.sort_values(by='Revenue', ascending=False).iloc[0]

kpi_data = {
    "total_revenue": total_revenue,
    "top_region": top_region['ship-state'],
    "top_region_revenue": top_region['Revenue']
}

kpi_df = pd.DataFrame([kpi_data])
kpi_df
```

Saving ETL IN A CSV FILE

```
[ ] #saving ETL OP so that lambda can read now
output_path = "/content/drive/MyDrive/sales_kpi_summary.csv"
kpi_df.to_csv(output_path, index=False)

output_path

... '/content/drive/MyDrive/sales_kpi_summary.csv'
```

AMAZON AWS WORK

Console Home [Info](#)

Reset to default layout [+ Add widgets](#)

Recently visited [Info](#)

- [Lambda](#)
- [IAM](#)
- [S3](#)

[View all services](#)

Applications (0) [Info](#)

Create application

Region: Asia Pacific (Mumbai)

Select Region [ap-south-1 \(Current Region\)](#) [Find applications](#)

No applications
Get started by creating an application.

[Create application](#)

[Go to myApplications](#)

Welcome to AWS

Getting started with AWS [Info](#)

Learn the fundamentals and [Get valuable information](#)

AWS Health [Info](#)

Open issues **0** Past 7 days

Cost and usage [Info](#)

Caption

Uploaded the CSV FILE OF ETL into s3 bucket

Amazon S3 < sales-op-bhavya123

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

Objects (1)

[Actions](#) [Create folder](#) [Upload](#)

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"/>	sales_kpi_summary.csv	csv	January 5, 2026, 00:43:40 (UTC+05:30)	91.0 B	Standard

LAMBDA CODE

Lambda > Functions

Lambda

- Dashboard
- Applications
- Functions**

Additional resources

- Capacity providers [New](#)
- Code signing configurations
- Event source mappings
- Layers

Functions (1) Last fetched 5/1/2026, 7:57:31 AM

Function name	Description	Package type	Runtime	Type	Last modified
sales-function	-	Zip	Python 3.14	Standard	7 hours ago

Info

Learn cases i

Cre

In ti to:

CloudWatch Logs Live Tail

Filter

Select log groups

Select up to 5 log groups

Add filter pattern - optional

Learn syntax

Filter the results to include only log events that match the pattern

Filter log events

Filter patterns are case sensitive and should contain fewer than 1024 characters.

00:00:00 0 events/sec

View in colu

Select a log group to start your Live Tail session

Stop Start

LAMBDA CODE IN PYTHON3

```
import json
import boto3
import pandas as pd
from io import BytesIO

BUCKET_NAME = 'sales-op-bhavya123'
FILE_KEY = 'sales_kpi_summary.csv'

s3 = boto3.client('s3')

def lambda_handler(event, context):
    try:    response = s3.get_object(Bucket=BUCKET_NAME, Key=FILE_KEY)
        df = pd.read_csv(BytesIO(response['Body'].read()))

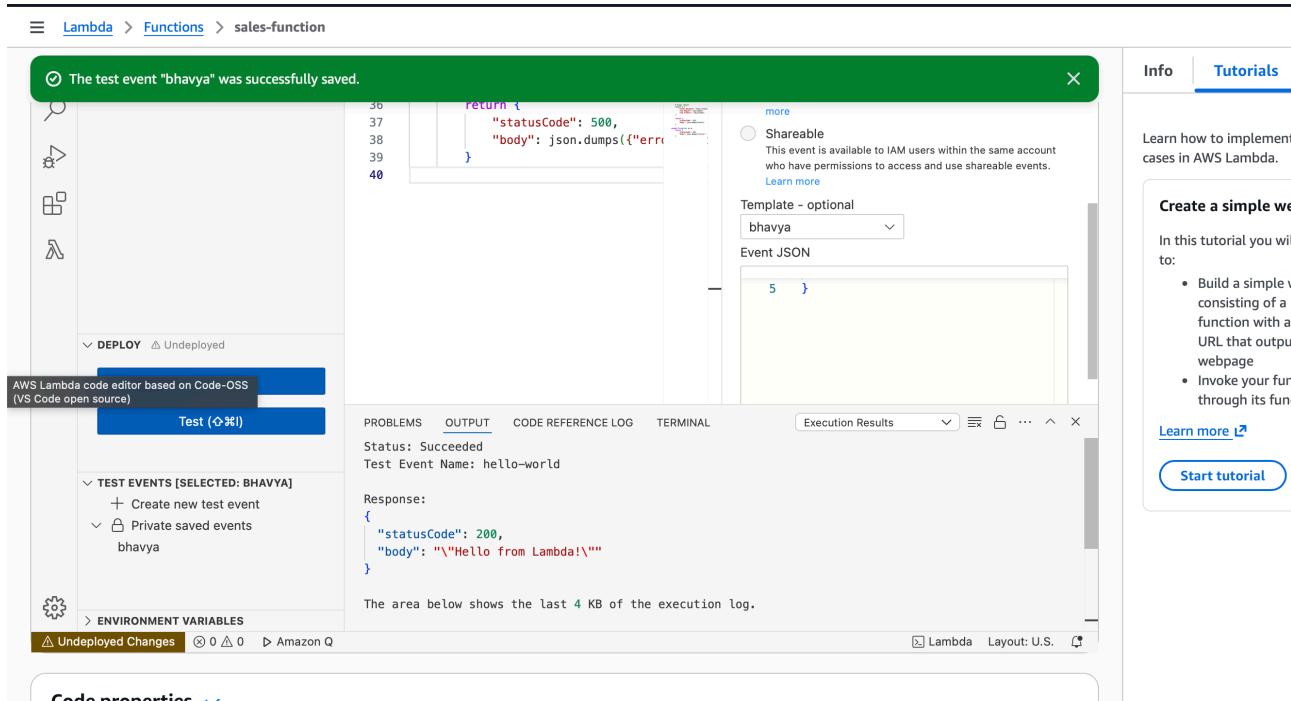
        total_revenue = float(df['Revenue'].sum())
        top_region = df.groupby('Region')['Revenue'].sum().idxmax()
        top_product = df.groupby('Product')['Revenue'].sum().idxmax()

        result = {
            "Total Revenue": total_revenue,
            "Top Region": top_region,
            "Top Product": top_product
        }

        return {
            "statusCode": 200,
            "body": json.dumps(result)
        }

    except Exception as e:
        return {
            "statusCode": 500,
            "body": json.dumps({"error": str(e)})
        }
```

Testing and getting correct response using AMAZON AWS



The screenshot shows the AWS Lambda Functions console. The top navigation bar includes 'Lambda', 'Functions', and 'sales-function'. A green notification bar at the top left says 'The test event "bhavya" was successfully saved.' The main area displays a code editor with Python code, showing a return statement that triggers a 500 error. To the right of the code editor is a 'Shareable' event configuration panel, which includes a 'Template - optional' dropdown set to 'bhavya' and an 'Event JSON' input field containing a simple JSON object. Below the code editor, the 'TEST EVENTS' section shows a selected event named 'bhavya'. The 'ENVIRONMENT VARIABLES' section is collapsed. At the bottom, there are tabs for 'PROBLEMS', 'OUTPUT', 'CODE REFERENCE LOG', and 'TERMINAL', with 'OUTPUT' being the active tab. The output shows a successful execution with status 'Succeeded' and a response body:

```
{"statusCode": 200, "body": "\"Hello from Lambda!\""}.
```

 The status bar at the bottom right indicates 'Lambda Layout: U.S.'