



---

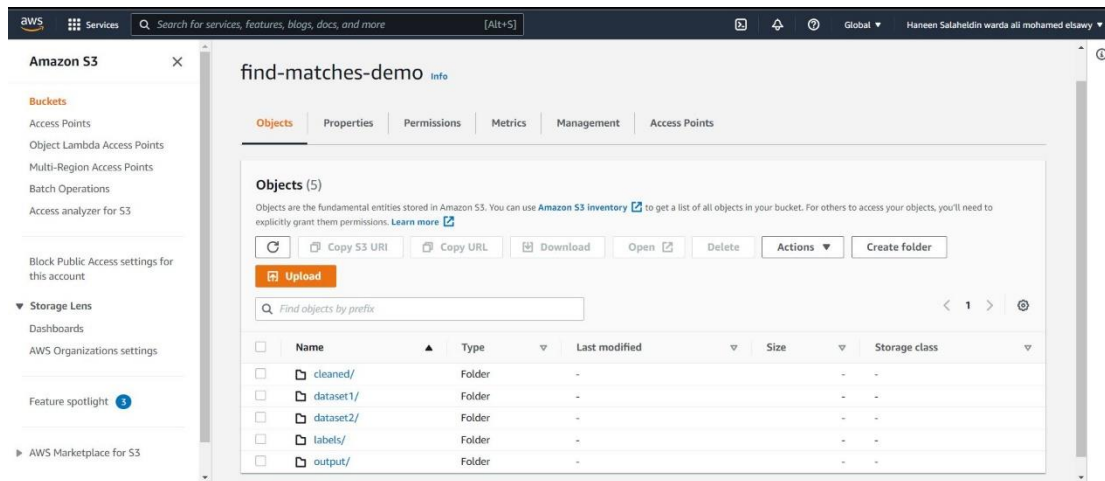
# HANEEN SALAH

---

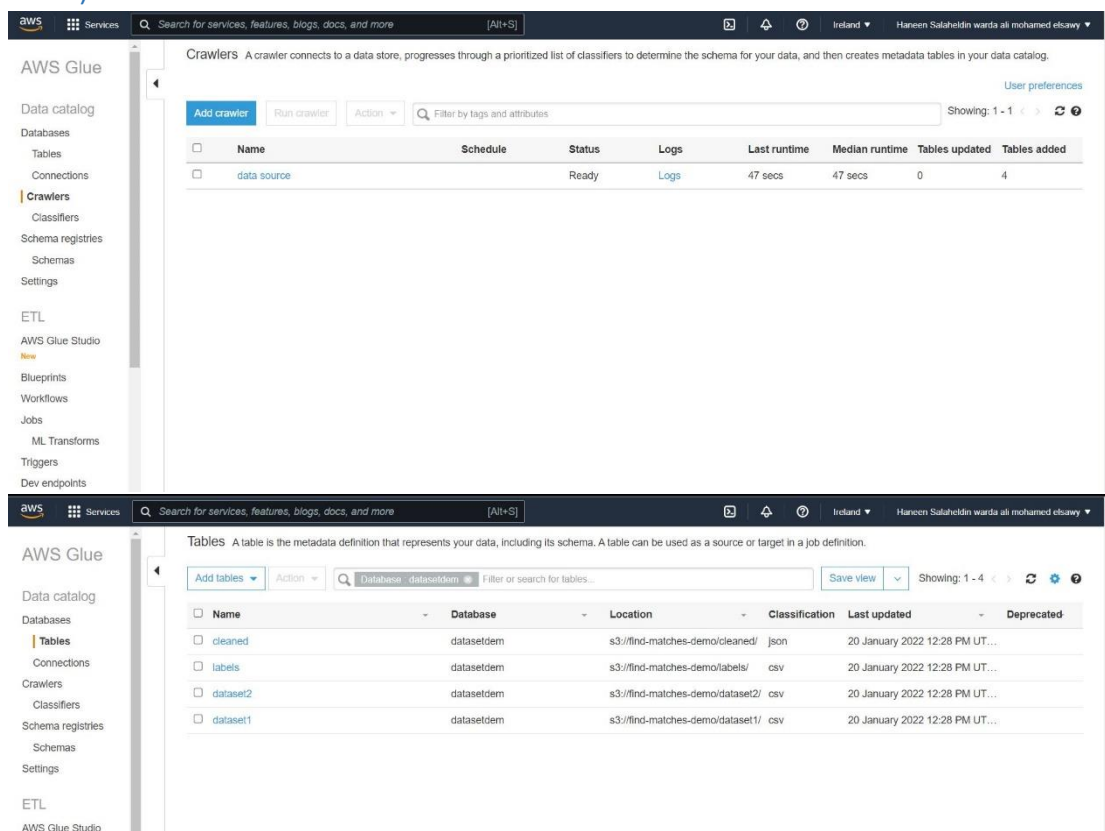
AWS-Find-matches



- 1) Create s3 bucket with folders as directories for dataset1, dataset2, output, labels, cleaned for the merged d1, d2 JSON file



- 2) Create crawler



### 3) Create ml-transform and upload labels

The screenshot shows the AWS Glue Machine learning transforms console. The left sidebar contains navigation options like Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL, AWS Glue Studio, Blueprints, Workflows, Jobs, ML Transforms (highlighted), Triggers, Dev endpoints, and Notebooks. The main panel displays the 'Machine learning transforms' section with a table of transforms. The 'find-match-transform' is selected, showing details in the 'Details' tab.

Transform name	Transform ID	Type	Label count	Status	Date created	Last modified	Description
find-match-transform	tfn-6e87c847a1856478529d5c7a7c2a861a4291	Find matching records	352	Ready for use	20 January 2022 12:2...	20 January 2022 12:3...	

The 'Details' tab for 'find-match-transform' shows the following configuration:

- Transform name:** find-match-transform
- Transform ID:** tfn-6e87c847a1856478529d5c7a7c2a861a4291
- Source database:** datasetdem
- Source data table:** cleaned
- Type:** Find matching records
- Spark Version:** 2.2
- IAM Role:** AWSGlueServiceRole-AWSGlueServiceRole-CrawlerTutorial1
- Status:** Ready for use
- Date created:** 20 January 2022 12:29 AM UTC+2
- Last modified:** 20 January 2022 12:30 AM UTC+2
- Description:** -
- Label count:** 352
- Precision-recall tradeoff:** 0.9
- Accuracy-cost tradeoff:** 1
- Force output to match labels:** false
- Worker type:** G.2X
- Number of workers:** 10
- Tags:** -
- Security configuration:** -
- Target encryption KMS key:** -

### 4) Create and run job

The screenshot shows the AWS Glue Jobs console. The left sidebar is the same as in the previous screenshot. The main panel displays the 'Jobs' section with a table of jobs. The 'find-match-job' is selected, showing details in the 'Details' tab.

Name	Type	ETL language	Script location	Last modified	Job bookmark
find-match-job	Spark	python	s3://aws-glue-s...	20 January 2022 12:39 A...	Disable

The 'Details' tab for 'find-match-job' shows the following configuration:

- Run ID:** jr\_ed84d766...
- Retry attempt status:** Succeeded
- Error:** -
- Output:** -
- Logs:** -
- Error logs:** -
- Glue version:** 2.0
- Maximu version capacity:** 10
- Triggered by:** -
- Start time:** 20 J...
- End time:** 20 J...
- Start-up time:** 7 secs
- Executi time:** 8 mins
- Timeou Delay:** 2880 mins
- Job run input:** s3://aws-glue-...

### 5) Output

The screenshot shows the Amazon S3 console. The left sidebar contains navigation options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3, Storage Lens, Dashboards, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main panel displays the 'Amazon S3' console for the 'find-matches-demo' bucket, showing the 'output/' folder. The 'Objects (80)' tab is selected, showing a list of objects.

Name	Type	Last modified	Size	Storage class
run-1642632528750-part-r-00000	-	January 20, 2022, 00:48:57 (UTC+02:00)	124.2 KB	Standard
run-1642632528750-part-r-00001	-	January 20, 2022, 00:48:57 (UTC+02:00)	127.7 KB	Standard
run-1642632528750-part-r-00002	-	January 20, 2022, 00:48:57 (UTC+02:00)	128.7 KB	Standard
run-1642632528750-part-r-00003	-	January 20, 2022, 00:48:57 (UTC+02:00)	118.4 KB	Standard
run-1642632528750-part-r-00004	-	January 20, 2022, 00:48:57 (UTC+02:00)	124.1 KB	Standard
run-1642632528750-part-r-00005	-	January 20, 2022, 00:48:57 (UTC+02:00)	116.6 KB	Standard
run-1642632528750-part-r-00006	-	January 20, 2022, 00:48:57 (UTC+02:00)	118.4 KB	Standard
run-1642632528750-part-r-00007	-	January 20, 2022, 00:48:57 (UTC+02:00)	124.4 KB	Standard

## 6) Merge multiple csv files into a single file

```
1  merge-mult-csv-out.py [3]
2  #!/usr/bin/env python
3  # coding: utf-8
4
5  # In[6]:
6
7  import os
8  import glob
9  import pandas as pd
10
11
12  # In[7]:
13
14
15  os.chdir("D:\AWS\out")
16
17
18  # In[8]:
19
20
21  extension = 'csv'
22  all_filenames = [i for i in glob.glob('*.{}'.format(extension))]
23
24
25  # In[9]:
26
27
28  #combine all files in the list
29  combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])
30  #export to csv
31  combined_csv.to_csv( "find-match.csv", index=False, encoding='utf-8-sig')
32
33
34  # In[ ]:
```

## 7) Csv final output sorted by match\_id

[illegible]