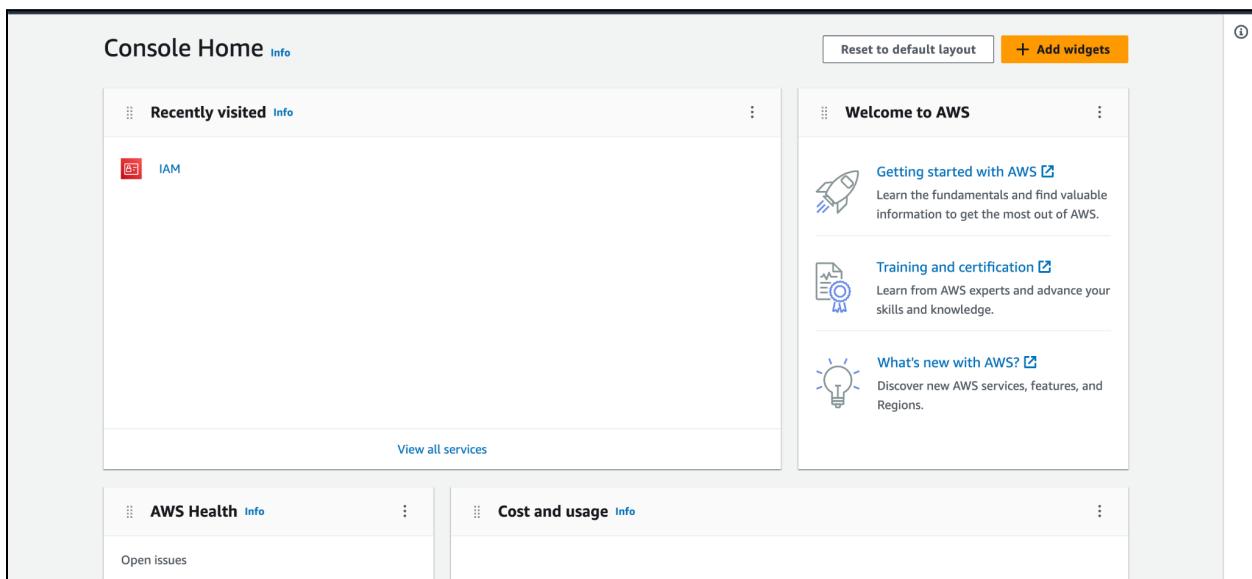


ETL AWS Project

Designed and deployed data pipelines to preprocess data, integrating data from various regions into data lakes, converting JSON and CSV data files into parquet formats and joining the data to make it available for the stakeholders (Data Analysts and Data Scientist) to be able to build reports on the cleaned data for business analysis.

Following are the steps:

1. Create IAM User



2. Configure CLI

- Downloaded AWS CLI tool.
- Open terminal check tool is installed by writing command:
aws on terminal
- Write command
aws configure
- Provide access key, secret access key, region and default output
- Check if it is configured or not by writing aws command. For eg:
aws s3 ls

3. Upload Data files in S3 storage

- Create Bucket

The screenshot shows the Amazon S3 console interface. On the left, there is a sidebar with various navigation options like Buckets, Storage Lens, and Feature spotlight. The main area displays a success message: "Successfully created bucket 'de-on-raw-us-east-1-dev'". Below this, there's an "Account snapshot" section and a table titled "Buckets (1) Info". The table lists one bucket:

Name	AWS Region	Access	Creation date
de-on-raw-us-east-1-dev	US East (N. Virginia) us-east-1	Bucket and objects not public	May 12, 2023, 23:29:05 (UTC-04:00)

- Uploading data into bucket

```
ETL_AWS % cd Data
Data % cd archive
archive % ls
Bvideos.csv      MX_category_id.json
N_category_id.json  MXVideos.csv
Nvideos.csv      RU_category_id.json
P_category_id.json  RUVideos.csv
Pvideos.csv      US_category_id.json
R_category_id.json  USVideos.csv
Rvideos.csv
archive % aws s3 cp . s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/ --recursive --exclude "*" --include "*.json"
o: 0
o: 0
#####
arwin.so    0x000000010e067663 s_print_stack_trace + 19
lib        0x00007ff818d29c1d _sigtramp + 29
          0x00007ff818934000 0x0 + 140703540920320
          0x0000600001342ec8 0x0 + 105553136463560
archive %
archive % aws s3 cp . s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/ --recursive --exclude "*" --include "*.json"

json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/IN_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/CA_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/GB_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/DE_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/JP_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/US_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/FR_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/KR_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/MX_category_id.json
json to s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/UU_category_id.json
archive %
```

Amazon S3

Amazon S3 > Buckets > de-on-raw-us-east1-dev > youtube/ > raw_statistics_reference_data/

raw_statistics_reference_data/

Objects | Properties

Objects (10)

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](#)

Actions | **Create folder**

Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	CA_category_id.json	json	May 16, 2023, 23:13:09 (UTC-04:00)	7.7 KB	Standard
<input type="checkbox"/>	DE_category_id.json	json	May 16, 2023, 23:13:09 (UTC-04:00)	7.7 KB	Standard
<input type="checkbox"/>	FR_category_id.json	json	May 16, 2023, 23:13:09 (UTC-04:00)	7.7 KB	Standard
<input type="checkbox"/>	GB_category_id.json	json	May 16, 2023, 23:13:09 (UTC-04:00)	8.0 KB	Standard
<input type="checkbox"/>	IN_category_id.json	json	May 16, 2023, 23:13:09 (UTC-04:00)	8.0 KB	Standard
<input type="checkbox"/>	JP_category_id.json	json	May 16, 2023, 23:13:09 (UTC-04:00)	8.0 KB	Standard

Amazon S3

Amazon S3 > Buckets > de-on-raw-us-east1-dev > youtube/ > raw_statistics_reference_data/

raw_statistics_reference_data/

Objects | Properties

Objects (10)

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](#)

Actions | **Create folder**

Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	region=ca/	Folder	-	-	-
<input type="checkbox"/>	region=de/	Folder	-	-	-
<input type="checkbox"/>	region=fr/	Folder	-	-	-
<input type="checkbox"/>	region=gb/	Folder	-	-	-
<input type="checkbox"/>	region=in/	Folder	-	-	-
<input type="checkbox"/>	region=jp/	Folder	-	-	-
<input type="checkbox"/>	region=kr/	Folder	-	-	-
<input type="checkbox"/>	region=mx/	Folder	-	-	-
<input type="checkbox"/>	region=ru/	Folder	-	-	-
<input type="checkbox"/>	region=us/	Folder	-	-	-

4. Create IAM roles for permissions

The screenshot shows the AWS IAM Roles page. A success message at the top states "Role de-glue-s3-role created." Below it, there's a search bar and a table listing three roles. The table has columns for Role name, Trusted entities, and Last activity. The roles listed are:

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
de-glue-s3-role	AWS Service: glue	-

Below the table, there's a section titled "Roles Anywhere" with two options: "Access AWS from your non AWS workloads" (X.509 Standard) and "Temporary credentials".

5. Creating AWS Glue Crawlers and Run Crawler Jobs

The screenshot shows the AWS Glue Crawler properties page for "de-on-raw-glue-catalog-1". A success message at the top says "One crawler successfully created" and "The following crawler is now created: 'de-on-raw-glue-catalog-1'".

The crawler properties are displayed in a table:

Name	IAM role	Database	State
de-on-raw-glue-catalog-1	de-glue-s3-role	de-raw	READY

Below the table, there are sections for Description, Security configuration, Lake Formation configuration, and Table prefix. There's also an "Advanced settings" button.

The "Crawler runs" tab is selected, showing 0 runs. It includes buttons for "Run crawler", "Stop run", "View CloudWatch logs", and "View run details". There are also "Filter data" and "Filter by a date and time range" input fields.

AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)

▼ Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings

► Data Integration and ETL
► Legacy pages

What's New

de-on-raw-glue-catalog-1

May 17, 2023 at 03:59:23 Run crawler Edit Delete

Crawler properties

Name de-on-raw-glue-catalog-1	IAM role de-glue-s3-role	Database de-raw	State READY
Description -	Security configuration -	Lake Formation configuration -	Table prefix -
Maximum table threshold -			

► Advanced settings

Crawler runs | Schedule | Data sources | Classifiers | Tags

Crawler runs (1)
The list of crawler runs for this crawler.

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
May 17, 2023 at 04:03:24	May 17, 2023 at 04:05:55	02 min 31 s	Completed	-	-

Stop run View CloudWatch logs View run details

AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring

Data Catalog tables
Data connections
Workflows (orchestration)

▼ Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings

► Data Integration and ETL
► Legacy pages

What's New

de-on-raw-glue-catalog-1

Crawler successfully starting
The following crawler is now starting: "de-on-raw-glue-catalog-1"

AWS Glue > Tables

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (1)
Last updated (UTC)
May 17, 2023 at 04:06:22

Name	Database	Location	Classification	Deprecated	View data
raw_statistics_reference_de-raw	s3://de-on-raw-us-east-1-	json	-	-	Table data

Add tables using crawler Add table

AWS Glue

Getting started
ETL jobs
 Visual ETL
 Notebooks
Job run monitoring

Data Catalog tables

Data connections
Workflows (orchestration)

▼ Data Catalog

Databases

 Tables

Stream schema registries
 Schemas

Connections
Crawlers
 Classifiers

Catalog settings

► Data Integration and ETL

► Legacy pages

What's New 

raw_statistics_reference_data

Name	Description	Database	Classification
raw_statistics_reference_data	-	de-raw	json

Location	Connection	Deprecated	Last updated
s3://de-on-raw-us-east1-dev/youtube/raw_statistics_reference_data/	-	-	May 17, 2023 at 04:05:55

Input format	Output format	Serde serialization lib
org.apache.hadoop.mapred.TextInputFormat	org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat	org.openx.data.jsonserde.JsonSerDe

Schema | Partitions | Indexes

Schema (3)

View and manage the table schema.

Edit schema as JSON Edit schema

#	Column name	Data type	Partition key	Comment
1	kind	string	-	-
2	etag	string	-	-
3	items	array	-	-

6. Create Lambda functions

- To upload JSON data in tabular format

The screenshot shows the AWS Lambda Code source interface. The top navigation bar includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (selected), 'Deploy', and 'Changes not deployed'. On the left, there's an 'Environment' sidebar with a dropdown for 'de-on-raw-us-east-1' and a file tree showing 'lambda_function.py'. The main area displays the following Python code:

```
1 import awswrangler as wr
2 import pandas as pd
3 import urllib.parse
4 import os
5
6 # Temporary hard-coded AWS Settings; i.e., to be set as OS variable in Lambda
7 os_input_s3_cleansed_layer = os.environ['s3_cleansed_layer']
8 os_input_glue_catalog_db_name = os.environ['glue_catalog_db_name']
9 os_input_glue_catalog_table_name = os.environ['glue_catalog_table_name']
10 os_input_write_data_operation = os.environ['write_data_operation']
11
12
13 def lambda_handler(event, context):
14     # Get the object from the event and show its content type
15     bucket = event['Records'][0]['s3']['bucket']['name']
16     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
17     try:
18
19         # Creating DF from content
20         df_raw = wr.s3.read_json('s3://{}{}'.format(bucket, key))
21
22         # Extract required columns:
23         df_step_1 = pd.json_normalize(df_raw['items'])
24
25         # Write to S3
26         wr_response = wr.s3.to_parquet(
27             df=df_step_1,
28             path=os_input_s3_cleansed_layer,
29             dataset=True,
30             database=os_input_glue_catalog_db_name,
31             table=os_input_glue_catalog_table_name,
32             mode=os_input_write_data_operation
33         )
34
35     except Exception as e:
36         print(e)
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1466
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1476
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1560
1561
1562
1563
1564
1565
1566
1566
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1576
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1666
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1676
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1760
1761
1762
1763
1764
1765
1766
1766
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1860
1861
1862
1863
1864
1865
1866
1866
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1876
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1960
1961
1962
1963
1964
1965
1966
1966
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1976
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2296
2297
2298
2298
2299
2299
2300
2301
2302
23
```

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (⌘ P)

Environment de-on-raw-us-east1 lambda_function Execution results

```
# Creating DF from content
df_raw = wr.s3.read_json('s3://{}{}'.format(bucket, key))

# Extract required columns:
df_step_1 = pd.json_normalize(df_raw['items'])

# Write to S3
wr_response = wr.s3.to_parquet(
    df=df_step_1,
    path=os_input_s3_cleansed_layer,
    dataset=True,
    database=os_input_glue_catalog_db_name,
    table=os_input_glue_catalog_table_name,
    mode=os_input_write_data_operation
)

return wr_response
except Exception as e:
    print(e)
    print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(bucket, key))
    raise e
```

1:1 Python Spaces: 4

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The test event s3_put was successfully saved.

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (⌘ P)

Environment de-on-raw-us-east1 lambda_function Execution result: X

Execution results Status: Succeeded Max memory used: 38 MB Time: 5.38 ms

Test Event Name s3.put

Response

```
{
    "statusCode": 200,
    "body": "\u261dHello from Lambda!\u261d"
}
```

Function Log

```
START RequestId: 6df0c30e-fa84-485c-a53c-5667081a3306 Version: $LATEST
END RequestId: 6df0c30e-fa84-485c-a53c-5667081a3306
REPORT RequestId: 6df0c30e-fa84-485c-a53c-5667081a3306 Duration: 5.38 ms Billed Duration: 6 ms Memory Size: 128 MB Max Memory Used: 38 MB
```

Request ID 6df0c30e-fa84-485c-a53c-5667081a3306

Successfully updated the function de-on-raw-us-east1-lambda-json-parquet.

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (% P) lambda_function Execution result:

Environment

Execution results

Test Event Name s3.put

Status: Succeeded Max memory used: 38 MB Time: 1.65 ms

Response

```
{ "statusCode": 200, "body": "\"Hello from Lambda!\""}  
Function Logs  
START RequestId: e42ce05c-6150-45e7-9143-b1e7db9d8d03 Version: $LATEST  
END RequestId: e42ce05c-6150-45e7-9143-b1e7db9d8d03  
REPORT RequestId: e42ce05c-6150-45e7-9143-b1e7db9d8d03 Duration: 1.65 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 38 MB  
Request ID e42ce05c-6150-45e7-9143-b1e7db9d8d03
```

Successfully updated the function de-on-raw-us-east1-lambda-json-parquet.

File Edit Find View Go Tools Window Test Deploy

Go to Anything (% P) lambda_function Execution result:

Environment

Execution results

Test Event Name s3.put

Status: Succeeded Max memory used: 128 MB Time: 8344.83 ms

Response

```
{ "paths": [ "s3://de-on-cleansed-us-east1-dev/youtube/62ff7993abff45aaa3902fa2496b34c0.snappy.parquet" ], "partitions_values": {} }  
Function Logs  
OpenBLAS WARNING - could not determine the L2 cache size on this system, assuming 256k  
START RequestId: afab67a0-d8cd-4d85-8a5c-019a21ec1e75 Version: $LATEST  
END RequestId: afab67a0-d8cd-4d85-8a5c-019a21ec1e75  
REPORT RequestId: afab67a0-d8cd-4d85-8a5c-019a21ec1e75 Duration: 8344.83 ms Billed Duration: 8345 ms Memory Size: 128 MB Max Memory Used:  
Request ID afab67a0-d8cd-4d85-8a5c-019a21ec1e75
```

- Successfully created table in Glue and dataset in tabular format

AWS Glue

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (2)

Last updated (UTC) May 25, 2023 at 20:35:33

Add table

Name	Database	Location	Classification	Deprecated	View data
cleaned_statistics_referer_de-raw	s3://de-on-cleansed-us-e	parquet	-	-	Table data
raw_statistics_reference_de-raw	s3://de-on-raw-us-east-1-	json	-	-	Table data

cleaned_statistics_reference_data

Location: s3://de-on-cleansed-us-east-1-dev/youtube

Connection: -

Deprecated: -

Last updated: May 25, 2023 at 20:35:01

Schema

View and manage the table schema.

#	Column name	Data type	Partition key	Comment
1	kind	string	-	-
2	etag	string	-	-
3	id	string	-	-
4	snippet_channel_id	string	-	-
5	snippet_title	string	-	-
6	snippet_assignable	boolean	-	-

- Loaded Data in S3 in parquet format

The screenshot shows the Amazon S3 console interface. On the left, there is a sidebar with various navigation options under 'Buckets'. The main area displays the contents of the 'de-on-cleaned-us-east1-dev' bucket. The 'Objects' tab is selected. A single object, 'youtube/', is listed in the table below. The object is a folder.

Name	Type	Last modified	Size	Storage class
youtube/	Folder	-	-	-

This screenshot shows the contents of the 'youtube/' folder within the 'de-on-cleaned-us-east1-dev' bucket. The 'Properties' tab is selected. One file, '62ff7993abff45aaa3902fa2496b34c0.snappy.parquet', is listed in the table below. It is a parquet file from May 25, 2023.

Name	Type	Last modified	Size	Storage class
62ff7993abff45aaa3902fa2496b34c0.snappy.parquet	parquet	May 25, 2023, 16:35:01 (UTC-04:00)	6.1 KB	Standard

- Athena Query Output

The screenshot shows the AWS Athena Query Editor interface. At the top, there are tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. A dropdown for 'Workgroup' is set to 'primary'. On the left, a sidebar titled 'Data' shows 'Data source' as 'AwsDataCatalog' and 'Database' as 'de-raw'. Below this is a section for 'Tables and views' with a 'Create' button and a search bar. Under 'Tables', there are two entries: 'cleaned_statistics_reference_data' and 'raw_statistics_reference_data'. Under 'Views', there are zero entries. In the main area, a query named 'Query 1' is displayed with the SQL command:

```
1 SELECT * FROM "AwsDataCatalog"."de-raw"."cleaned_statistics_reference_data" limit 5;
```

Below the SQL editor are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. A note indicates 'Reuse query results *Athena engine version 3 only'. The status bar at the bottom shows 'Completed' with a green icon, and performance metrics: 'Time in queue: 150 ms', 'Run time: 547 ms', and 'Data scanned: 2.10 KB'.

This screenshot shows the results of the query from the previous screenshot. The interface is identical, but the results section is now populated. It shows a table titled 'Results (5)' with the following data:

#	kind	etag	id	snippet_channel
1	youtube#videoCategory	"ld9biNPKjAjgjV7EZ4EKeEGrhao/Xy1mb4_yLrHy_BmKmPBggty2mZQ"	1	UCBR8-60-B28
2	youtube#videoCategory	"ld9biNPKjAjgjV7EZ4EKeEGrhao/UZ1oLlz2dxlhO45ZTFR3a3NyTA"	2	UCBR8-60-B28
3	youtube#videoCategory	"ld9biNPKjAjgjV7EZ4EKeEGrhao/nqRlq97-xe5XRZTxbknKFVe5Lmg"	10	UCBR8-60-B28
4	youtube#videoCategory	"ld9biNPKjAjgjV7EZ4EKeEGrhao/HwXKamM1Q20q9BN-oBjavSGkfDI"	15	UCBR8-60-B28
5	youtube#videoCategory	"ld9biNPKjAjgjV7EZ4EKeEGrhao/9GQMSRjrZdHeb1OEM1XVQ9zbGec"	17	UCBR8-60-B28

7. Load CSV datasets using Glue Job

- Create Crawler

AWS Glue > Crawlers > de-on-raw-csv-glue-catalog-1

de-on-raw-csv-glue-catalog-1

Last updated (UTC) May 26, 2023 at 21:23:28 | Run crawler | Edit | Delete

Crawler properties

Name	IAM role	Database	State
de-on-raw-csv-glue-catalog-1	de-glue-s3-role	de-raw	READY
Description	Security configuration	Lake Formation configuration	Table prefix
-	-	-	-
Maximum table threshold			
-			
▶ Advanced settings			

Crawler runs | Schedule | Data sources | Classifiers | Tags

Crawler runs (0)

The list of crawler runs for this crawler.

Filter data | Filter by a date and time range | View run details | < 1 > | ⚙️

- Run Crawler

AWS Glue > Crawlers > de-on-raw-csv-glue-catalog-1

Last updated (UTC) May 26, 2023 at 21:23:28 | Run crawler | Edit | Delete

Crawler properties

Name	IAM role	Database	State
de-on-raw-csv-glue-catalog-1	de-glue-s3-role	de-raw	READY
Description	Security configuration	Lake Formation configuration	Table prefix
-	-	-	-
Maximum table threshold			
-			
▶ Advanced settings			

Crawler runs | Schedule | Data sources | Classifiers | Tags

Crawler runs (1)

The list of crawler runs for this crawler.

Filter data | Filter by a date and time range | View run details | < 1 > | ⚙️

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
May 26, 2023 at 21:25:09	May 26, 2023 at 21:26:17	01 min 07 s	Completed	-	1 table change, 10 partition changes

- Athena Output

The screenshot shows the Amazon Athena Query Results interface. At the top, there's a sidebar with 'Tables (3)' and 'Views (0)'. The main area displays a query result table with columns: #, video_id, trending_date, and title. The results are as follows:

#	video_id	trending_date	title
1	SbOwzAl9ZfQ	17.14.11	Capítulo 12 MasterChef 2017
2	kI0V6Xh-DnI	17.14.11	ALEXA EX-INTEGRANTE DEL GRUPO TIMBIRICHE RENUNCIA A "La Voz Mexico 7" TELEV
3	6L2ZF7Qzsbk	17.14.11	LOUIS CKAGÓ - EL PULSO DE LA REPÚBLICA
4	hcY52MFWMMDM	17.14.11	Sismo de 6.7 sacude Costa Rica 12 Noviembre 2017
5	_OXDcGPVAa4	17.14.11	DOG HACKS MUSAS LESSLIE LOS POLINESIOS

At the bottom, there are 'Copy' and 'Download results' buttons.

- Joining the tables and getting output

The screenshot shows the Amazon Athena Query Editor interface. It features three tabs for 'Query 1', 'Query 2', and 'Query 3'. The SQL code for 'Query 3' is displayed:

```

1 SELECT a.title,a.category_id, b.snippet_title FROM "AwsDataCatalog"."de-raw"."raw_statistics" a
2 INNER JOIN "AwsDataCatalog"."de-raw"."cleaned_statistics_reference_data" b
3 ON a.category_id=cast(b.id as int)
4 where a.region='ca'
5 limit 5;
    
```

The interface includes a sidebar for 'Data' (Data source: AwsDataCatalog, Database: de-raw), 'Tables and views' (Tables: cleaned_statistics_reference_data, raw_statistics, raw_statistics_reference_data), and a toolbar with 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create' buttons.

SQL Ln 5, Col 9

Run again Explain Cancel Clear Create

Reuse query results *Athena engine version 3 only

Completed Time in queue: 131 ms Run time: 2.09 sec Data scanned: 37.92 MB

Results (5)

Copy Download results

#	title	category_id	snippet
1	"Eminem - Walk On Water (Audio) ft. Beyoncé"	10	Music
2	"PLUSH - Bad Unboxing Fan Mail"	23	Come
3	"I Dare You: GOING BALDI?"	24	Enter
4	"Ed Sheeran - Perfect (Official Music Video)"	10	Music
5	"Jake Paul Says Alissa Violet CHEATED with LOGAN PAUL! #DramaAlert Team 10 vs Martinez Twins!"	25	News

8. Changing the data type of id from String to BigInt

- Edit datatype in glue table

AWS Glue

Getting started ETL jobs Visual ETL Notebooks Job run monitoring Data Catalog tables Data connections Workflows (orchestration)

Tables Stream schema registries Schemas Connections Crawlers Classifiers Catalog settings

Data Integration and ETL Legacy pages

What's New

Location s3://de-on-cleansed-us-east1-dev/youtube Connection - Deprecated - Last updated May 26, 2023 at 23:51:45

Input format org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat Output format org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat Serde serialization lib org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe

Schema (6) View and manage the table schema. Edit schema as JSON Edit schema

#	Column name	Data type	Partition key	Comment
1	kind	string	-	-
2	etag	string	-	-
3	id	bigint	-	-
4	snippet_channel_id	string	-	-
5	snippet_title	string	-	-
6	snippet_assignable	boolean	-	-

- Delete current parquet file in S3 Bucket

☰ **Successfully deleted objects**
View details below.

Delete objects: status

The information below will no longer be available after you navigate away from this page.

Summary

Source	Successfully deleted	Failed to delete
s3://de-on-cleansed-us-east1-dev/youtube/	1 object, 6.1 KB	0 objects

Failed to delete Configuration

Failed to delete (0)

Name	Folder	Type	Last modified	Size	Error
No objects failed to delete.					

Close

This screenshot shows the AWS S3 Delete Objects status page. At the top, a green banner indicates 'Successfully deleted objects' with a link to 'View details below'. Below the banner, the title 'Delete objects: status' is displayed. A message states that the information will be removed when navigating away. The 'Summary' section shows a single object was successfully deleted from the source 's3://de-on-cleansed-us-east1-dev/youtube/'. The 'Failed to delete' section shows zero objects. The 'Failed to delete' tab is selected, and its content area displays a table with no rows, indicating no failures.

- Run Lambda function again

☰ **Code** | Test | Monitor | Configuration | **Aliases** | Versions

Code source **Info**

File Edit Find View Go Tools Window **Test** Deploy

Upload from

Environment

Code source: **Info**

Execution results

Test Event Name: lambda_function

Status: **Succeeded** Max memory used: 128 MB Time: 7881.93 ms

Response:

```
{ "paths": [ "s3://de-on-cleansed-us-east1-dev/youtube/9655bd933741496a868e5890387ebbc8.snappy.parquet" ], "partitions_values": {} }
```

Function Logs

```
OpenBLAS WARNING - could not determine the L2 cache size on this system, assuming 256K
START RequestId: 72497904-4ec9-458a-97fd-cb30baf8321c Version: $LATEST
END RequestId: 72497904-4ec9-458a-97fd-cb30baf8321c
REPORT RequestId: 72497904-4ec9-458a-97fd-cb30baf8321c Duration: 7881.93 ms Billed Duration: 7882 ms Memory Size: 128 MB Max Memory Used: 128 MB
```

Request ID: 72497904-4ec9-458a-97fd-cb30baf8321c

This screenshot shows the AWS Lambda function test interface. The tabs at the top are 'Code', 'Test' (which is selected), 'Monitor', 'Configuration', 'Aliases', and 'Versions'. The 'Code source' tab is active, showing the file structure 'de-on-raw-us-east1' containing 'lambda_function.py'. The 'Test' tab shows the execution results for a test event named 'lambda_function'. The status is 'Succeeded' with a duration of 7881.93 ms and a maximum memory usage of 128 MB. The response body is a JSON object with 'paths' and 'partitions_values' fields. The 'Function Logs' section displays standard Lambda execution logs, including OpenBLAS warnings, request and end times, and a report summary.

- New parquet file created in S3 bucket

The screenshot shows the Amazon S3 console interface. On the left, there's a sidebar with various navigation options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. Below that are sections for Block Public Access settings, Storage Lens (Dashboards and AWS Organizations settings), Feature spotlight, and AWS Marketplace for S3. The main area shows a bucket named 'youtube/'. Under the 'Objects' tab, there is one object listed: '9655ba933741496a868e5890387ebc8.snappy.parquet'. This object is a parquet file, last modified on May 26, 2023, at 20:03:22 (UTC-04:00), with a size of 6.2 KB and a storage class of Standard. There are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, and Create folder.

- Successfully converted the data type

The screenshot shows the AWS Athena console. On the left, there are dropdown menus for Data source (set to AwsDataCatalog) and Database (set to de-raw). Below that is a 'Tables and views' section with a 'Create' button. The main area shows a query editor with the following SQL code:

```
1 SELECT a.title,a.category_id, b.snippet_title FROM "AwsDataCatalog"."de-raw"."raw_statistics" a
2 INNER JOIN "AwsDataCatalog"."de-raw"."cleaned_statistics_reference_data" b
3 ON a.category_id=b.id
4 where a.region='ca'
5 limit 5;
```

Below the query editor is a table titled 'Tables (3)' showing two tables: 'cleaned_statistics_reference_data' and 'raw_statistics'. The 'cleaned_statistics_reference_data' table has columns: kind (string), etag (string), id (string), snippet_channel_id (string), snippet_title (string), and snippetAssignable (boolean). The 'raw_statistics' table is Partitioned and has columns: video_id (string), trending_date (string), and title (string). To the right of the table list is a 'SQL' tab with 'Ln 3, Col 22' and a 'Run' button. Below the SQL tab is a 'Query results' section showing a green bar indicating 'Completed' with a timestamp of 'Time in queue: 131 ms' and 'Run time: 2.09 sec'. It also shows 'Data scanned: 37.92 MB'. At the bottom is a 'Results (5)' section with a table header showing columns: category_i and snipp.

```

1 SELECT a.title,a.category_id, b.snippet_title FROM "AwsDataCatalog"."de-raw"."raw_statistics" a
2 INNER JOIN "AwsDataCatalog"."de-raw"."cleaned_statistics_reference_data" b
3 ON a.category_id=b.id
4 where a.region='ca'
5 limit 5;

```

Tables (3)

Table	Type	Columns
cleaned_statistics_reference_data	Partitioned	kind, etag, id, snippet_channel_id, snippet_title, snippet_assignable
raw_statistics	Partitioned	category_id, title
raw_statistics_reference_data	Partitioned	category_id, snippet_title

Results (5)

#	title	category_id	snippet
1	"Eminem - Walk On Water (Audio) ft. Beyoncé"	10	Music
2	"PLUSH - Bad Unboxing Fan Mail"	23	Come
3	"I Dare You: GOING BALD?!"	24	Enter
4	"Ed Sheeran - Perfect (Official Music Video)"	10	Music
5	"Jake Paul Says Alissa Violet CHEATED with LOGAN PAUL! #DramaAlert Team 10 vs Martinez Twins!"	25	News

Tables (3)

Table	Type	Columns
cleaned_statistics_reference_data	Partitioned	kind, etag, id, snippet_channel_id, snippet_title, snippet_assignable
raw_statistics	Partitioned	category_id, title
raw_statistics_reference_data	Partitioned	category_id, snippet_title

Results (5)

#	title	category_id	snippet
1	"Eminem - Walk On Water (Audio) ft. Beyoncé"	10	Music
2	"PLUSH - Bad Unboxing Fan Mail"	23	Come
3	"I Dare You: GOING BALD?!"	24	Enter
4	"Ed Sheeran - Perfect (Official Music Video)"	10	Music
5	"Jake Paul Says Alissa Violet CHEATED with LOGAN PAUL! #DramaAlert Team 10 vs Martinez Twins!"	25	News

9. Convert all CSV data files into parquet format

- Create ETL glue job

de-on-youtube-cleansed-csv-to-parquet-etljob

Last modified on 5/27/2023, 4:30:28 PM [Try new UI](#) [Actions](#) [Save](#) [Run](#)

Successfully updated job
Successfully updated job de-on-youtube-cleansed-csv-to-parquet-etljob. To run the job choose the Run Job button.

[Script](#) [Job details](#) [Runs](#) [Data quality](#) [New](#) [Schedules](#) [Version Control](#)

Basic properties [Info](#)

Name
de-on-youtube-cleansed-csv-to-parquet-etljob

Description - optional

Descriptions can be up to 2048 characters long.

IAM Role
Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.
[de-glue-s3-role](#)

Type
The type of ETL job. This is set automatically based on the types of data sources you have selected.
[Python Shell](#)

Data Catalog

- Databases
- Tables
- Stream schema registries
- Schemas
- Connections
- Crawlers
- Classifiers
- Catalog settings

Data Integration and ETL

- ETL jobs**
- Visual ETL
- Notebooks
- Job run monitoring

Source [Amazon S3](#) **Target** [Amazon S3](#)

Your jobs (1) [Info](#)

Job name	Type	Last modified	AWS Glue version
de-on-youtube-cleansed-csv-to-parquet-etljob-01	Glue ETL	5/27/2023, 4:46:09 PM	2.0

● Run Glue Job

AWS Glue [X](#) **de-on-youtube-cleansed-csv-to-parquet-etljob-01** Last modified on 5/27/2023, 4:51:58 PM [Try new UI](#) [Actions](#) [Save](#) [Run](#)

[Getting started](#) [ETL jobs](#) [Visual ETL](#) [Notebooks](#) [Job run monitoring](#) [Data Catalog tables](#) [Data connections](#) [Workflows \(orchestration\)](#)

[Script](#) [Job details](#) [Runs](#) [Data quality](#) [New](#) [Schedules](#) [Version Control](#)

Job runs (1/2) [Info](#) Last updated (UTC) May 27, 2023 at 20:58:32 [C](#) [View details](#) [Stop job run](#) [Table View](#) [Card View](#)

Run status	Retry	Start time	End time	Duration	Capacity	Worker type
Succeeded	0	05/27/2023 16:52:01	05/27/2023 16:54:37	2 m 28 s	10 DPU	G.1X

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3 > Buckets > de-on-cleansed-us-east1-dev > youtube/

youtube/

Objects | Properties

Objects (3)

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	9655ba933741496a868e5890387ebbc8.snappy.parquet	parquet	May 26, 2023, 20:03:22 (UTC-04:00)	6.2 KB	Standard
<input type="checkbox"/>	raw_statistics/\$folder\$	-	May 27, 2023, 16:53:04 (UTC-04:00)	0 B	Standard
<input type="checkbox"/>	raw_statistics/	Folder	-	-	-

Copy S3 URI

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3 > Buckets > de-on-cleansed-us-east1-dev > youtube/ > raw_statistics/

raw_statistics/

Objects | Properties

Objects (6)

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	region=ca/\$folder\$	-	May 27, 2023, 16:54:21 (UTC-04:00)	0 B	Standard
<input type="checkbox"/>	region=ca/	Folder	-	-	-
<input type="checkbox"/>	region=gb/\$folder\$	-	May 27, 2023, 16:54:21 (UTC-04:00)	0 B	Standard
<input type="checkbox"/>	region=gb/	Folder	-	-	-
<input type="checkbox"/>	region=us/\$folder\$	-	May 27, 2023, 16:54:22 (UTC-04:00)	0 B	Standard
<input type="checkbox"/>	region=us/	Folder	-	-	-

Copy S3 URI

- Create crawlers for parquet files of regions data

AWS Glue

One crawler successfully created
The following crawler is now created: "de-on-cleaned-csv-glue-catalog-2"

AWS Glue > Crawlers > de-on-cleaned-csv-glue-catalog-2

Last updated (UTC) May 27, 2023 at 21:51:28 C Run crawler Edit Delete

Crawler properties

Name	IAM role	Database	State
de-on-cleaned-csv-glue-catalog-2	de-glue-s3-role	de-raw	READY
Description	Security configuration	Lake Formation configuration	Table prefix
-	-	-	-
Maximum table threshold			
-			
▶ Advanced settings			

Crawler runs Schedule Data sources Classifiers Tags

Crawler runs (0)
The list of crawler runs for this crawler.

Filter data Filter by a date and time range

AWS Glue

de-on-cleaned-csv-glue-catalog-2

Last updated (UTC) May 27, 2023 at 21:51:28 C Run crawler Edit Delete

Crawler properties

Name	IAM role	Database	State
de-on-cleaned-csv-glue-catalog-2	de-glue-s3-role	de-raw	READY
Description	Security configuration	Lake Formation configuration	Table prefix
-	-	-	-
Maximum table threshold			
-			
▶ Advanced settings			

Crawler runs Schedule Data sources Classifiers Tags

Crawler runs (1)
The list of crawler runs for this crawler.

Filter data Filter by a date and time range

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
May 27, 2023 at 21:52:34	May 27, 2023 at 21:53:23	49 s	Completed	0.063	1 table change, 3 partition changes

Tables and views Create

Tables (4)

- cleaned_statistics_reference_data** (Partitioned)
 - kind: string
 - etag: string
 - id: bigint
 - snippet_channel_id: string
 - snippet_title: string
 - snippet_assignable: boolean
- raw_statistics** (Partitioned)
 - raw_statistics_8ff6a1dcebe20d5ea2b9d1** (Partitioned)
 - video_id: string
 - 1ce4cc56e3** (Partitioned)
 - video_id: string

Views (0)

SQL Ln 1, Col 85

Run again Explain Cancel Clear Create

Reuse query results *Athena engine version 3 only

Query results **Query stats**

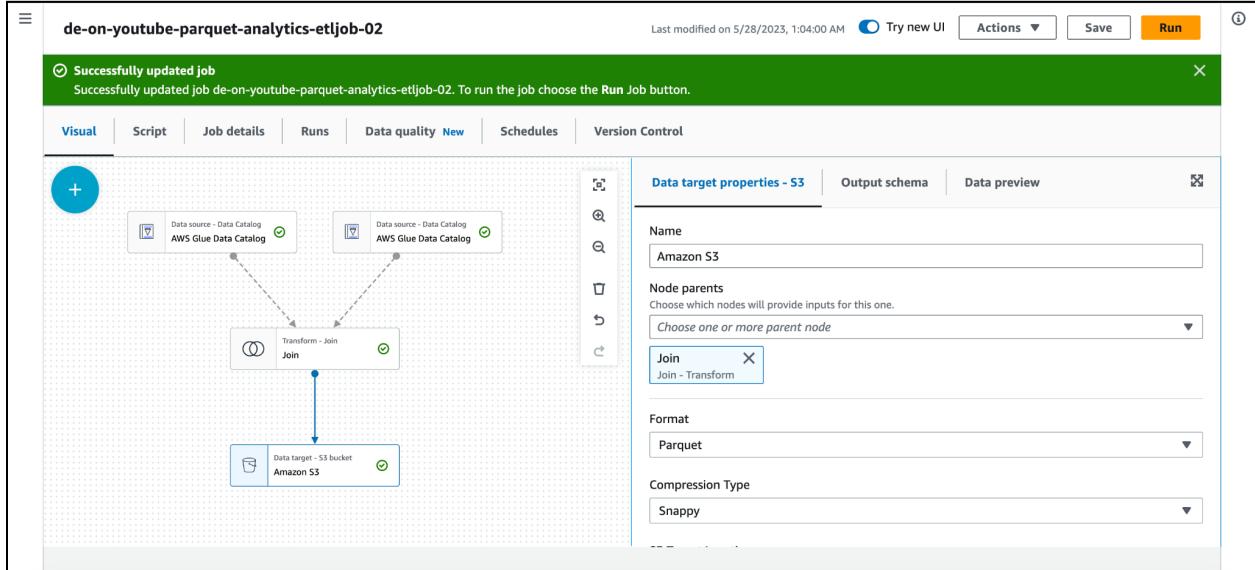
Completed Time in queue: 209 ms Run time: 471 ms Data scanned: 2.14 KB

Results (5)

#	kind	etag	id	snippet_chann
1	youtube#videoCategory	"ld9biNPkjAjqjV7EZ4EKeEGrhao/Xy1mB4_yLrHy_BmKmPBggty2mZQ"	1	UCBR8-60-B28
2	youtube#videoCategory	"ld9biNPkjAjqjV7EZ4EKeEGrhao/UZ1oLliz2dxlhO45ZTR3a3NyTA"	2	UCBR8-60-B28
3	youtube#videoCategory	"ld9biNPkjAjqjV7EZ4EKeEGrhao/nqRlq97-xe5XRZTxkknKFVe5Lmg"	10	UCBR8-60-B28
4	youtube#videoCategory	"ld9biNPkjAjqjV7EZ4EKeEGrhao/HwXKamM1Q20q9BN-oBJavSGkfDI"	15	UCBR8-60-B28

10. ETL pipeline for Reporting and Analytics

- Created ETL job for joining tables in AWS Glue



- Ran Job

AWS Glue

de-on-youtube-parquet-analytics-etljob-02

Last modified on 5/28/2023, 1:04:00 AM Try new UI Actions Save Run

Getting started ETL jobs Visual ETL Notebooks Job run monitoring Data Catalog tables Data connections Workflows (orchestration)

Data Catalog Databases Tables Stream schema registries Schemas Connections Crawlers Classifiers Catalog settings

Data Integration and ETL ETL jobs Visual ETL Notebooks Job run monitoring

Job runs (1/1) Info

Last updated (UTC) May 28, 2023 at 05:10:47

Run status Retry Start time End time Duration Capacity Worker type

Run status	Retry	Start time	End time	Duration	Capacity	Worker type
Succeeded	0	05/28/2023 01:06:02	05/28/2023 01:07:50	1 m 40 s	10 DPUs	G.1X

05/28/2023 01:06:02

Job name	Id	Run status	Glue version
de-on-youtube-parquet-analytics-etljob-02	jr_4d5622b1ff5131fc2deb8e9044c33f25c6e9e1298ce8afe5a0e68973d254298	Succeeded	3.0

Retry attempt number	Start time	End time	Start-up time
Initial run	May 28, 2023 1:06:02 AM	May 28, 2023 1:07:50 AM	7 seconds

Execution time	Last modified on	Trigger name	Security configuration
1 minute 40 seconds	May 28, 2023 1:07:50 AM	-	-

- Athena Output of One Final table

Amazon Athena > Query editor

Editor Recent queries Saved queries Settings Workgroup primary

Data

Data source: AwsDataCatalog Database: db_youtube_analytics

Tables and views: final_analytics

Query 1 : X | Query 2 : X | Query 3 : X | **Query 4 : X** | **Query 5 : X**

```
1 #create database db_youtube_analytics;
2
3
```

Data

Data source: AwsDataCatalog

Database: db_youtube_analytics

Tables and views: Create

Tables (1)

final_analytics	Partitioned
ratings_disabled	boolean
comments_disabled	boolean
snippet_title	string
trending_date	string
etag	string
video_id	string
thumbnail_link	string
snippet_assignable	boolean
kind	string

SQL: SELECT * FROM "AwsDataCatalog"."db_youtube_analytics"."final_analytics" limit 5;

Run again Explain Cancel Create Reuse query results *Athena engine version 3 only

Query results: Completed Time in queue: 179 ms Run time: 789 ms Data scanned: 193.17 KB

Results (5) Copy Download results Search rows

Filter tables and views

Tables (1)

final_analytics	Partitioned
ratings_disabled	boolean
comments_disabled	boolean
snippet_title	string
trending_date	string
etag	string
video_id	string
thumbnail_link	string
snippet_assignable	boolean
kind	string
comment_count	bigint

Views (0)

SQL: SELECT * FROM "AwsDataCatalog"."db_youtube_analytics"."final_analytics" limit 5;

Run again Explain Cancel Create Reuse query results *Athena engine version 3 only

Query results: Completed Time in queue: 179 ms Run time: 789 ms Data scanned: 193.17 KB

Results (5) Copy Download results Search rows

#	ratings_disabled	comments_disabled	snippet_title	trending_date	etag
1	false	false	Howto & Style	18.01.02	"ld9biNPkjAjgjV7EZ4EKeEGhrao
2	false	false	Howto & Style	18.02.06	"ld9biNPkjAjgjV7EZ4EKeEGhrao
3	false	false	Howto & Style	18.04.02	"ld9biNPkjAjgjV7EZ4EKeEGhrao
4	false	false	Howto & Style	18.29.05	"ld9biNPkjAjgjV7EZ4EKeEGhrao
5	false	false	Howto & Style	18.24.01	"ld9biNPkjAjgjV7EZ4EKeEGhrao