



Raghul Sekar

# **REAL-TIME DATA PIPELINE: INTEGRATING AWS AND SNOWFLAKE**

# CONTENT

This guide provides hands-on experience with AWS tools and Snowflake.

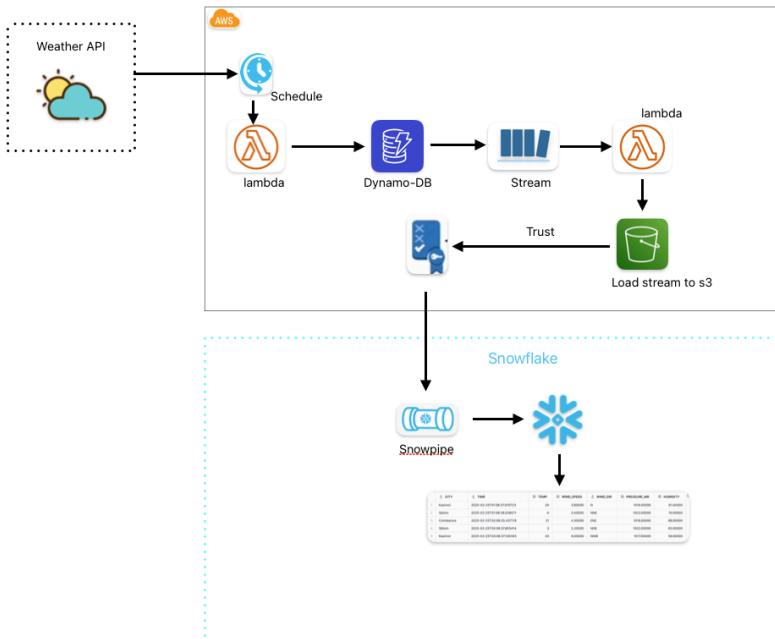
The project involves retrieving real-time weather data for various cities using an API key and storing it in DynamoDB every hour via EventBridge (CloudWatch Events). When a new record is added to DynamoDB, a DynamoDB stream triggers a Lambda function, which transfers the data to S3 in CSV format.

In Snowflake, a connection to S3 will be established using a newly created IAM role, with trust configured between Snowflake and AWS. A storage integration will be set up in Snowflake, allowing S3 to function as an external stage. Whenever new data is loaded into S3, Snowpipe—triggered by an event notification—will automatically ingest the new records into a Snowflake table.

# **SECTIONS**

1. Flow Diagram
2. Prerequisites
3. DynamoDB Configuration
4. Lambda Function 1: Get data into DynamoDB
5. S3 Storage Setup
6. Lambda Function 2: Data Transfer to S3
7. Snowflake Integration
8. Conclusion

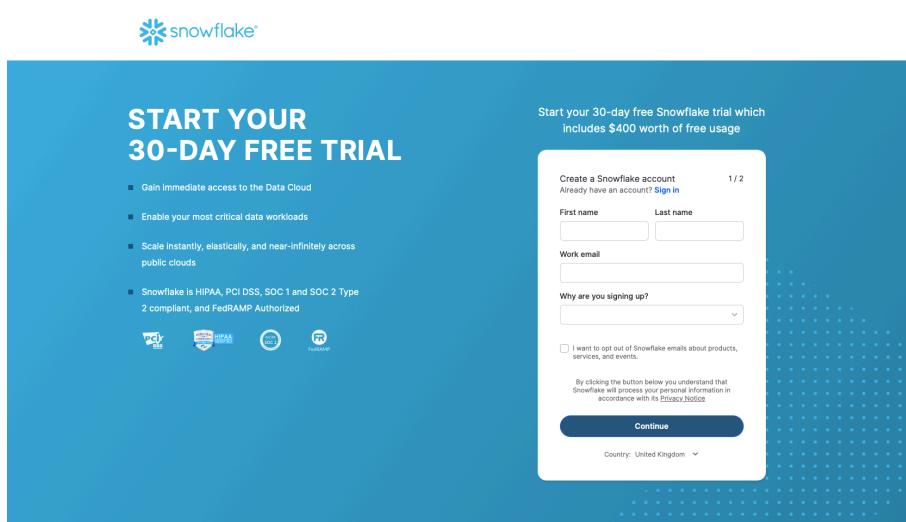
# 1. Flow Diagram



## 2. Prerequisites

Sign up for Snowflake, AWS, and WeatherAPI, and obtain the Weather API key, which will be used later in the project.

Get the scripts from my [Git repo](#)



A screenshot of the AWS CloudTrail sign-up page. The top navigation bar includes the AWS logo, search bar, and links for "Products", "Solutions", "Pricing", "Documentation", "Learn", "Partner Network", "AWS Marketplace", "Customer Enablement", "Events", "Explore More", "About AWS", "Contact Us", "Support", "English", "My Account", and "Sign In to the Console". The main content area has a yellow gradient background. It shows a message "One copy of management events delivered with the AWS Free Tier" and the title "AWS CloudTrail". Below it, the text "Track user activity and API usage on AWS and in hybrid and multicloud environments" is displayed. A "Complete sign up" button is at the bottom.

### Benefits

Aggregate and consolidate multisource events



Immutably store audit-worthy events



Derive insights and analyze unusual activity





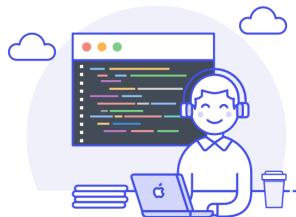
Features Pricing API Explorer Docs Weather Contact Signup Login

## JSON and XML Weather API and Geolocation Developer API

Designed for developers by developers, Weather API is the ultimate weather and geolocation API trusted by 600,000+ users worldwide. Integrate weather in .

[Sign Up](#)

[View Docs](#)



### Real Time, Forecasted, Future, Marine and Historical Weather

Free Weather Forecast in JSON and XML for commercial and non-commercial use



#### Real-time Weather API

Real-time or current JSON weather and XML weather API provides



#### Historical and Future Weather

14 day daily, hourly and 15 min interval weather forecast, future weather



#### Geolocation API

Air Quality Data, Time zone, Astronomy, IP lookup, Sports API, Currency API, Map API



#### Technology

Lightweight XML and JSON format with 200 ms average response time. Bulk request with

[Contact us](#)

Welcome Back

API key: 3de3217b36d9439b896175913252402 [Copy](#) LIVE TRIAL Ends on 10/Mar/2025

Pro Plus Plan

5,000,000 Calls per Month

0 Calls Made

10/Mar/25 Trial End Date

Note: If you are on a trial plan then after the trial plan ends your API key will be automatically moved to Free plan if you do not wish to upgrade to a paid plan.

Get Started

- Learn how to form HTTP request to get weather from API Explorer or use our NEW Swagger Tool.
- Complete weather API documentation.
- Weather icons and weather lookup code list.
- Want to choose which weather field to return in the API response? Change it from API response fields.
- Looking to upgrade/downgrade your API plan? Visit our Upgrade/Downgrade plan section.

Regenerate API Key

Has your key been compromised? You can generate a new key.

Before you proceed?

- You will need to change your apps to use the new key.
- Your statistics will be reset.
- This action cannot be undone.

I agree to above and would like to regenerate api key.

[Contact us](#)

# 3. DynamoDB Configuration

In this session we will create a dynamodb table to store data.

The screenshot shows the AWS Services console. The search bar at the top contains the text "dynamodb". In the "Services" section, the "DynamoDB" service is highlighted with a blue border. Other services listed include Amazon DocumentDB and CloudFront. On the left sidebar, there's a "Recent" section with icons for Lambda, S3, IAM, and other services like RDS and Oracle. The main navigation bar at the top includes links for "United States (N. Virginia)" and the user "raghul\_sekar".

The screenshot shows the DynamoDB Dashboard. The left sidebar has sections for "Tables", "Exports", "Imports", "Integrations", "Reserved capacity", and "DAX". Under "DAX", there are sections for "Clusters", "Subnet groups", "Parameter groups", and "Events". The main dashboard has sections for "Favorite tables", "Alarms", and "DAX clusters". A red arrow points to the "Create resources" button in the "Create resources" section, which contains text about creating an Amazon DynamoDB table for fast and predictable database performance at any scale. It also mentions Amazon DynamoDB Accelerator (DAX) and provides a link to "Create DAX cluster". The right sidebar shows a "What's new" section with updates from February 2023, including support for auto-approval of quotas, warm throughput for tables, and support for FIPS 140-5 interface.

aws Search [Option+S]

☰ dynamoDB > Tables > Create table United States (N. Virginia) raphul\_sekar

### Create table

**Table details** Info  
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**  
This will be used to identify your table.  
**weather**  
Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.)

**Partition key**  
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.  
**city** String  
1 to 255 characters and case sensitive.

**Sort key - optional**  
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.  
**time** String  
1 to 255 characters and case sensitive.

**And Click Create**

### Table settings

**Default settings**

The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose **Customize settings**.

**Customize settings**

Use these advanced features to make DynamoDB work better for your needs.

### Default table settings

These are the default settings for your new table. You can change some of these settings after creating the table.

Setting	Value	Editable after creation
Table class	DynamoDB Standard	Yes
Capacity mode	On-demand	Yes
Maximum read capacity units	-	Yes

aws Search [Option+S]

☰ dynamoDB > Tables United States (N. Virginia) raphul\_sekar

**DynamoDB** Actions Delete Create table

**Tables (1)** Info

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Write capa
<b>weather</b>	<span style="color: green;">Active</span>	<b>city (\$)</b>	<b>time (\$)</b>	0	0	<span style="color: orange;">Off</span>	<span style="color: orange;">☆</span>	On-demand	On-demand

**New Table is created**

# 4. Lambda Function 1: Get data into DynamoDB

In this session, we will set up a Lambda function to load data from the Weather API into DynamoDB every hour.

The screenshot shows the AWS Lambda service search results. The search bar at the top contains the text "lambda". Below the search bar, there are two sections: "Services" and "Features". The "Services" section contains three items: "Lambda" (Run code without thinking about servers), "CodeBuild" (Build and Test Code), and "AWS Signer" (Ensuring trust and integrity of your code). The "Features" section has a "Show more" link. On the left side, there is a sidebar with a "project-v" icon, a "Objects" section containing a "Find" button, and a "Related services" section with links to Services, Features, Resources, Documentation, Knowledge articles, Marketplace, Blog posts, Events, and Tutorials. The top right corner shows the region "United States (N. Virginia)" and the user "raghul\_sekar".

The screenshot shows the AWS Lambda dashboard for the United States (N. Virginia) region. The top navigation bar includes a search bar, a "[Option+S]" key indicator, and account information. The main area displays "Resources for United States (N. Virginia)" with metrics: 2 Lambda function(s), 1.7 kB code storage (0% of 75 GB), Full account concurrency 10, and Unreserved account concurrency 10. A red arrow points to the "Create function" button. Below this, the "Top 10 functions" section shows three charts: "Errors", "Invocations", and "Concurrent Executions". The "Errors" chart shows 1 error at 15:00, 0.5 errors at 16:00, and 0 errors at 17:00. The "Invocations" chart shows 7 invocations at 15:00, 3.5 invocations at 16:00, and 0 invocations at 17:00. The "Concurrent Executions" chart shows 4 concurrent executions at 15:00, 2 concurrent executions at 16:00, and 0 concurrent executions at 17:00. The left sidebar includes sections for Lambda Dashboard, Additional resources (Code signing configurations, Event source mappings, Layers, Replicas), and Related AWS resources (Step Functions state machines).

**Create function** [Info](#)

Choose one of the following options to create your function.

Author from scratch Start with a simple Hello World example.

Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.

Container image Select a container image to deploy for your function.

**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.  
**fetch\_weather\_data\_2**

**Runtime** [Info](#)  
Choose the language to use to write your Lambda. Note that the console code editor supports only Node.js, Python, and Ruby.  
**Python 3.13**

**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  
**x86\_64**

**Permissions** [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

**Change default execution role**

**Additional Configurations**  
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

**Create function**

**fetch\_weather\_data\_2**

**Function overview** [Info](#)

**Description**  
-

**Last modified**  
22 seconds ago

**Function ARN**  
arn:aws:lambda:us-east-1:05182723074:function:fetch\_weather\_data\_2

**Function URL** [Info](#)

**Code** [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

**Code source** [Info](#)

**Upload from** [...](#)

**EXPLORER**

- FETCH\_WEATHER\_DATA\_2
  - lambda\_function.py

```
fetch_weather_data_2
lambda_function.py
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
```

fetch\_weather\_data\_2

**Throttle** **Copy ARN** **Actions ▾**

**Function overview** [Info](#)

**Diagram** **Template**

**Layers** (0)

**+ Add trigger** **+ Add destination**

**Description**

**Last modified** 52 seconds ago

**Function ARN** arn:aws:lambda:us-east-1:051826723074:function:fetch\_weather\_data\_2

**Function URL** [Info](#)

**Code** **Test** **Monitor** **Configuration** **Aliases** **Versions**

**General configuration** [Info](#)

**Triggers**

**Permissions**

**Destinations**

**Function URL**

**Environment variables**

**Tags**

**VPC**

General configuration	Memory	Ephemeral storage
Description	128 MB	512 MB
<b>Timeout</b>	0 min 3 sec	<a href="#">Edit</a>
<b>SnapStart</b>	None	

**Edit basic settings**

**Basic settings** [Info](#)

**Description - optional**

**Memory** [Info](#)  
Your function is allocated CPU proportional to the memory configured.  
 MB  
Set memory to between 128 MB and 10240 MB.

**Ephemeral storage** [Info](#)  
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)  
 MB  
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

**SnapStart** [Info](#)  
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#). For Python and .NET runtimes, [view pricing](#)

Supported runtimes: .NET 8 (C#/.NET/PowerShell), Java 11, Java 17, Java 21, Python 3.12, Python 3.13.

**Timeout**  
 min  sec

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#)  
 Use an existing role  
 Create a new role from AWS policy templates

**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
 [View the lambda-role-yb0eqTz role](#) on the IAM console.

**Cancel** **Save**

Screenshot of the AWS Lambda console showing the configuration for a function named "fetch\_weather\_data\_2".

The left sidebar shows various AWS services: General configuration, Triggers, Permissions (selected), Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, Monitoring and operations tools, Concurrency and recursion detection, Asynchronous invocation, Code signing, File systems, and State machines.

The main area displays the "Execution role" section. The role name is "fetch\_weather\_data\_2-role-ybiOeqt2". A red arrow points to this role name. Below it is the "Resource summary" section, which lists "Amazon CloudWatch Logs" with 3 actions and 2 resources. The "By resource" tab is selected. The resources listed are:

- arn:aws:logs:us-east-1:051826723074:log-group:/aws/lambda/fetch\_weather\_data\_2:\*
- arn:aws:logs:us-east-1:051826723074:log-group:/aws/lambda/fetch\_weather\_data\_2:\*

Actions associated with these resources include:

- Allow: logs:CreateLogGroup
- Allow: logs:CreateLogStream
- Allow: logs:PutLogEvents

A callout box highlights the policy statements obtained from the Lambda function:

- Managed policy AWSLambdaBasicExecutionRole-059f850f-1e3b-41b7-98ea-36df57344056, statement 0
- Managed policy AWSLambdaBasicExecutionRole-059f850f-1e3b-41b7-98ea-36df57344056, statement 1

Screenshot of the AWS IAM Roles page showing the details for the role "fetch\_weather\_data\_2-role-ybiOeqt2".

The left sidebar shows the IAM navigation menu: Identity and Access Management (IAM) (selected), Access management, and Access reports.

The main area shows the "Summary" of the role. It includes the creation date (February 25, 2025, 17:51 UTC), last activity (none), ARN (arn:aws:iam:051826723074:role/service-role/fetch\_weather\_data\_2-role-ybiOeqt2), and maximum session duration (1 hour). A red arrow points to the ARN.

The "Permissions" tab is selected. It shows the "Permissions policies" section, which allows attaching up to 10 managed policies. A red arrow points to the "Add permissions" button. The table below lists the attached policies:

Policy name	Type	Attached entities
AWSLambdaBasicExecutionRole-059f850f-1e3b-41...	Customer managed	1

The "Permissions boundary" section indicates "(not set)".

The "Generate policy based on CloudTrail events" section shows a note: "You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy." A red arrow points to the "Generate policy" button.

Screenshot of the AWS IAM Roles page showing the "fetch\_weather\_data\_2-role-ybi0eqt2" role. The "Add permissions" section is open, displaying a list of "Other permissions policies (1/1037)". A red arrow points to the first policy, "AmazonDynamoDBFullAccess", which is selected. Another red arrow points to the "Add permissions" button at the bottom right.

Attach policy to fetch\_weather\_data\_2-role-ybi0eqt2

▶ Current permissions policies (1)

Other permissions policies (1/1037)

Policy name	Type	Description
<input checked="" type="checkbox"/>  AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon DynamoDB. This policy is on a deprecation path. See the <a href="#">AWS Lambda migration guide</a> .
<input type="checkbox"/>  AmazonDynamoDBFullAccessWithDataPipeline	AWS managed	This policy is on a deprecation path. See the <a href="#">AWS Lambda migration guide</a> .
<input type="checkbox"/>  AmazonDynamoDBReadonlyAccess	AWS managed	Provides read only access to Amazon DynamoDB.
<input type="checkbox"/>  AWSLambdaDynamoDBExecutionRole	AWS managed	Provides list and read access to DynamoDB tables.
<input type="checkbox"/>  AWSLambdaDynamoDBStreamExecutionRole	AWS managed	Provides read access to DynamoDB Streams.

Cancel Add permissions

Screenshot of the AWS Lambda Functions page showing the "fetch\_weather\_data\_2" function. A red arrow points to the "Layers" section, which contains one layer named "Layers".

aws Search [Option+S]

☰ Lambda > Functions > fetch\_weather\_data\_2

## fetch\_weather\_data\_2

▼ Function overview [Info](#)

[Diagram](#) [Template](#)

+ Add trigger + Add destination

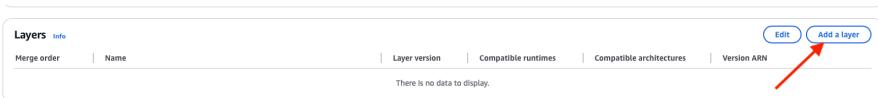
 **fetch\_weather\_data\_2**

 **Layers** (1)

Layers Info

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					

[Edit](#) [Add a layer](#)



aws [Option+S] Search United States (N. Virginia) raghu\_sekar

Lambda > Add layer

Add layer

Function runtime settings

Runtime Python 3.13

Architecture x86\_64

Choose a layer

Layer source Info

Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also create a new layer.

AWS layers Choose a layer from a list of layers provided by AWS.

Custom layers Choose a layer from a list of layers created by your AWS account.

Specify an ARN Specify a layer by providing the ARN.

AWS layers

Layers provided by AWS that are compatible with your function's runtime.

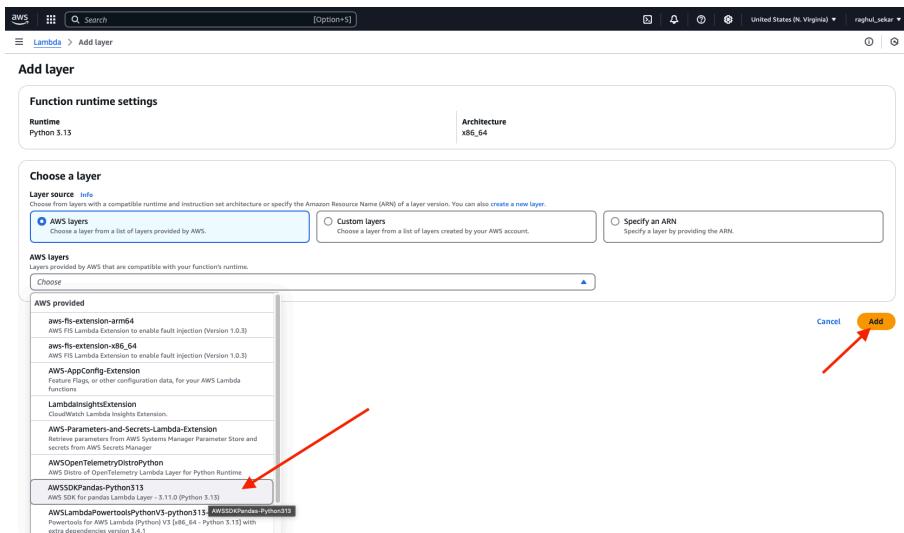
Choose

AWS provided

- aws-ls-extension-arm64 AWS FIS Lambda Extension to enable fault injection (Version 1.0.3)
- aws-ls-extension-x86\_64 AWS FIS Lambda Extension to enable fault injection (Version 1.0.3)
- AWS-AppConfig-Extension Feature Flags, or other configuration data, for your AWS Lambda function.
- LambdaInsightsExtension CloudWatch Lambda Insights Extension
- AWS-Parameters-and-Secrets-Lambda-Extension Retrieve parameters from AWS Systems Manager Parameter Store and secrets from AWS Secrets Manager.
- AWS-OpenTelemetry-Distro-Python AWS Stacks of OpenTelemetry Lambda Layer for Python Runtime
- AWS-SDK-Pandas-Python313** AWS SDK for pandas Lambda Layer - 3.11.0 (Python 3.13)
- AWS-Lambda-Powertools-PythonV3-python313 AWS-SDK-Pandas-Python313

PowerTools for AWS Lambda (Python) V3 x86\_64 - Python 3.13 with extra dependencies version 3.4.1

[Cancel](#) [Add](#)



The screenshot shows the AWS Lambda Code Source interface. On the left, the file tree displays 'EXPLORER' and 'DEPLOY' sections. Under 'EXPLORER', there is a folder 'FETCH\_WEATHER\_DATA\_2' containing a file 'lambda\_function.py'. Under 'DEPLOY', there are 'Deploy (OK!)' and 'Test (OK!)' buttons. The main area shows the code for 'lambda\_function.py':

```
1 # Script to get data from api and store it in dynamodb
2
3 import json
4
5 # http://api.weatherapi.com/v1/current.json?key=4e1d4da91no
6 from datetime import datetime
7 import requests
8 import boto3
9 from decimal import Decimal
10
11 dynamodb = boto3.resource('dynamodb')
12 table = dynamodb.Table('weather')
13
14 def get_weather_data(city):
15     url = "http://api.weatherapi.com/v1/current.json"
16     params = {
17         "q": city,
18         "key": "1de2121b36d433989617591325242"
19     }
20     response = requests.get(url, params=params)
21
22     return response.json()
23
24
25
26
27
28
29
30
31
32
33
34
35
36
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
305
306
307
307
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
15
```



aws [Option]+S]

[Lambda](#) > Add triggers

### Add trigger

**Trigger configuration** [Info](#)

Eventbridge (CloudWatch Events)

**Rule**  
Pick an existing rule or create a new one.  
 Create a new rule  
 Existing rule

**Rule name**  
Enter a name to uniquely identify your rule.  
**1hr\_trigger**

**Rule description**  
Provide an optional description for your rule.

**Rule type**  
Trigger your target based on an event pattern, or based on an automated schedule.  
 Event pattern  
 Schedule expression

**Schedule expression**  
Set trigger your target at a scheduled time using [Cron or rate expressions](#). Cron expressions are in UTC.  
**rate(1 hour)**

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

[Cancel](#) [Add](#)

aws [Option]+S]

[Lambda](#) > Functions > **fetch\_weather\_data\_2**

**fetch\_weather\_data\_2**

The trigger **1hr\_trigger** was successfully added to function **fetch\_weather\_data\_2**. The function is now receiving events from the trigger.

**Function overview** [Info](#)

[Diagram](#) [Template](#) [Export to Infrastructure Composer](#) [Download](#)

**fetch\_weather\_data\_2**

**Layers** (1)

**EventBridge (CloudWatch Events)**

[+ Add trigger](#)

**Description**

Last modified 4 minutes ago

**Function ARN** [am:aws:lambda:us-east-1:051826723074:function:fetch\\_weather\\_data\\_2](#)

**Function URL** [Info](#)

**Code** **Test** **Monitor** **Configuration** [Aliases](#) [Versions](#)

**General configuration**

**Execution role**

**Role name** [fetch\\_weather\\_data\\_2-role-yb10eq2](#)

**Resource summary**

To view the resources and actions that your function has permission to access, choose a service.

**AWS Application Auto Scaling** 7 actions, 1 resource

**By action** **By resource**

Once the function runs you can see data in DynamoDB table

The screenshot shows the AWS DynamoDB console with the 'weather' table selected. The left sidebar includes options like Dashboard, Tables, Explore Items, Partition editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. The main area shows the 'weather' table details, including a 'Scan or query items' section with a 'Scan' button, a 'Select attribute projection' dropdown set to 'All attributes', and a 'Filters' section. Below this is a green bar indicating 'Completed. Read capacity units consumed: 2'. The 'Items returned (20)' section displays 20 rows of weather data with columns: city (String), time (String), humidity, pressure\_mb, temp, wind\_dir, and wind\_spd. The data includes entries for Kochi, Norwich, Dehradun, and Kerela at various dates and times.

	city (String)	time (String)	humidity	pressure_mb	temp	wind_dir	wind_spd
1	Kochi	2025-02-25T18:31:51...	79	1014	26.3	NE	4.5
2	Kochi	2025-02-25T18:32:17...	79	1014	26.3	NE	4.5
3	Norwich	2025-02-25T18:31:50...	76	1014	7.2	VNW	5.8
4	Norwich	2025-02-25T18:32:16...	76	1014	7.2	VNW	5.8
5	Dehradun	2025-02-25T18:31:51...	33	1020	14	NE	6.7
6	Dehradun	2025-02-25T18:32:16...	33	1020	14	NE	6.7
7	Kerala	2025-02-25T18:31:51...	8	1011	32.3	ENE	4.5

# 5. S3 Storage Setup

In this session, we will set up a S3 bucket.

The screenshot shows the AWS S3 console homepage. The top navigation bar includes the AWS logo, a search bar with 's3', and account information for 'United States (N. Virginia)' and 'raghul\_sekar'. The main content area is titled 'Services' and features three cards: 'S3 Scalable Storage in the Cloud', 'S3 Glacier Archive Storage in the Cloud', and 'AWS Snow Family Large Scale Data Transport'. Below this is a section titled 'Features' with a 'Show more' link and a card for 'Imports from S3'. On the left sidebar, under 'Console', there are links for recent services like Lambda, S3, IAM, and CloudWatch. A vertical sidebar on the right contains 'Add widgets' and navigation controls.

The screenshot shows the Amazon S3 landing page. The top navigation bar includes the AWS logo, a search bar with '[Options+]', and account information for 'United States (N. Virginia)' and 'raghul\_sekar'. The main content area is titled 'Amazon S3' and features the sub-headline 'Store and retrieve any amount of data from anywhere'. It includes a brief description of S3 as an object storage service and a call-to-action button 'Create a bucket' with a red arrow pointing to it. To the left is a sidebar with 'Amazon S3' navigation, 'Storage' options, and links for 'Storage Lens', 'AWS Marketplace for S3', and 'Feature spotlight'. To the right are sections for 'Pricing' (with a note about no minimum fees), 'Resources' (including 'User guide', 'API reference', 'FAQs', 'Discussion forums', and 'S3 on the AWS news blog'), and 'Common tasks'.

**Create bucket** [Info](#)

Buckets are containers for data stored in S3.

**General configuration**

AWS Region  
US East (N. Virginia) us-east-1

**Bucket type** [Info](#)

General purpose Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

**Bucket name** [Info](#) 

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

**Copy settings from existing bucket - optional**  
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

**Object Ownership** [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Object Ownership**  
Bucket owner enforced

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or alt. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or its objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Account snapshot - updated every 24 hours** [All AWS Regions](#) [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. [Learn more](#)

**General purpose buckets** [Info](#) **Directory buckets**

**General purpose buckets (1)** [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.



Name	AWS Region	IAM Access Analyzer	Creation date
project-weather-raghul	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	February 24, 2025, 18:30:56 (UTC+00:00)

**project-weather-raghul** [Info](#)

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

**Objects (1)** [Create a subfolder called snowflake](#) [Download](#) [Open](#) [Delete](#) [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](#)

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

# 6. Lambda Function 2: Data Transfer to S3

In this session, we will set up a function to transfer data from DynamoDB to S3 using DynamoDB Streams.

To add the stream, click on trigger and follow the next step

Past my lambda\_function2 script here

```
lambda_function.py
...
# Put records from DynamoDB to S3
...
# S3 bucket and folder path
FOLDER_NAME = "snowflake"
# Lambda handler
def lambda_handler(event, context):
    records = []
    print(event)
    # Process each record from DynamoDB stream
    for record in event['Records']:
        dynamodb = boto3.resource('dynamodb')
        table = dynamodb.Table(record['dynamodb']['tableName'])
        response = table.get_item(
            Key=record['dynamodb']['key']
        )
        item = response['Item']
        records.append(item)
    s3 = boto3.client('s3')
    for record in records:
        file_name = record['id'] + '.json'
        file_content = json.dumps(record)
        s3.put_object(
            Bucket=FOLDER_NAME,
            Key=file_name,
            Body=file_content
        )
```

Trigger configuration

DynamoDB

event database event-source-mapping nosql polling

DynamoDB table

Event source mapping configuration

Activate trigger

Enable metrics

Batch size

Starting position

Batch window - optional

Additional settings

Cancel Save

aws Q Search [Option+S] Global raghu\_Lekar

IAM > Roles > db\_to\_s3-role-gclmbucc

**db\_to\_s3-role-gclmbucc** Info Delete Edit

**Identity and Access Management (IAM)**

Dashboard

**Access management**

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

**Access reports**

- Access Analyzer
- External access
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies [New](#)

IAM Identity Center [New](#)

AWS Organizations [New](#)

**Summary**

Creation date February 24, 2025, 18:34 (UTC)

Last activity 7 minutes ago

ARN arn:aws:iam:051826723074:role/service-role/db\_to\_s3-role-gclmbucc

Maximum session duration 1 hour

**Permissions** **Trust relationships** **Tags** **Last Accessed** **Revoke sessions**

Add these two permissions to the our function before testing it

**Permissions policies (3) info**

You can attach up to 10 managed policies.

Filter by Type Simulate Remove Add permissions

Policy name	Type	Attached entities
AmazonDynamoDBFullAccess	AWS managed	3
AmazonS3FullAccess	AWS managed	2
AWSLambdaBasicExecutionRole-6791f54f-60af-4c..	Customer managed	1

**Permissions boundary (not set)**

**Generate policy based on CloudTrail events**

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

[Generate policy](#)

No requests to generate a policy in the past 7 days.

lambda\_function.py

```
# put records from DynamoDB to S3
import json
import csv
import boto3
import os
import time
from io import StringIO
from datetime import datetime

s3 = boto3.client('s3')
BUCKET_NAME = "project-weather-raghul"
FOLDER_NAME = "snowflake/"

def lambda_handler(event, context):
    records = []
    print(event)

    # Process each record from DynamoDB stream
    for record in event['Records']:
        if record['eventName'] == 'INSERT':
            new_image = record['dynamodb']['NewImage']
            record_data = {
```

Change the bucket\_name to yours in my script

bucket\_name

aws Search [Option+5] United States (N. Virginia) raghu\_sekar

Amazon S3 > Buckets > project-weather-raghul > snowflake/

**snowflake/**

Once function is ran we can see csv in our s3 bucket

**Objects (50)**

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](#)

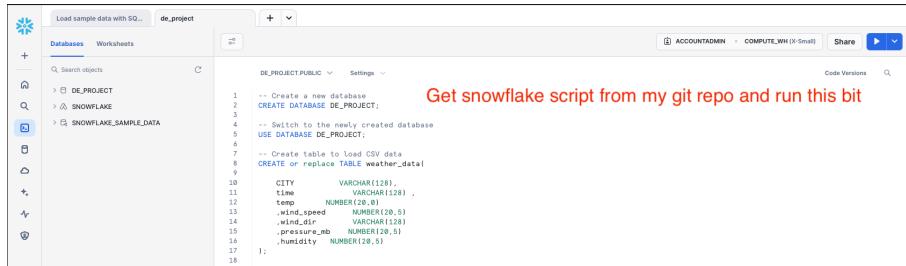
**Actions** Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

**Find objects by prefix**

Name	Type	Last modified	Size	Storage class
weather_data_2025-02-24_22-51-32.csv	csv	February 24, 2025, 22:51:34 (UTC+00:00)	111.0 B	Standard
weather_data_2025-02-24_22-51-33.csv	csv	February 24, 2025, 22:51:34 (UTC+00:00)	113.0 B	Standard
weather_data_2025-02-24_22-51-34.csv	csv	February 24, 2025, 22:51:35 (UTC+00:00)	116.0 B	Standard
weather_data_2025-02-24_23-08-37.csv	csv	February 24, 2025, 23:08:39 (UTC+00:00)	114.0 B	Standard
weather_data_2025-02-24_23-08-38.csv	csv	February 24, 2025, 23:08:39 (UTC+00:00)	113.0 B	Standard
weather_data_2025-02-25_00-08-38.csv	csv	February 25, 2025, 00:08:39 (UTC+00:00)	113.0 B	Standard
weather_data_2025-02-25_01-08-38.csv	csv	February 25, 2025, 01:08:40 (UTC+00:00)	112.0 B	Standard
weather_data_2025-02-25_01-08-39.csv	csv	February 25, 2025, 01:08:40 (UTC+00:00)	113.0 B	Standard
weather_data_2025-02-25_02-08-37.csv	csv	February 25, 2025, 02:08:39 (UTC+00:00)	117.0 B	Standard
weather_data_2025-02-25_02-08-38.csv	csv	February 25, 2025, 02:08:39 (UTC+00:00)	112.0 B	Standard
weather_data_2025-02-25_03-08-38.csv	csv	February 25, 2025, 03:08:40 (UTC+00:00)	112.0 B	Standard

# 7. Snowflake Integration

In this session, we will connect AWS and Snowflake and insert the data into a Snowflake table from S3.

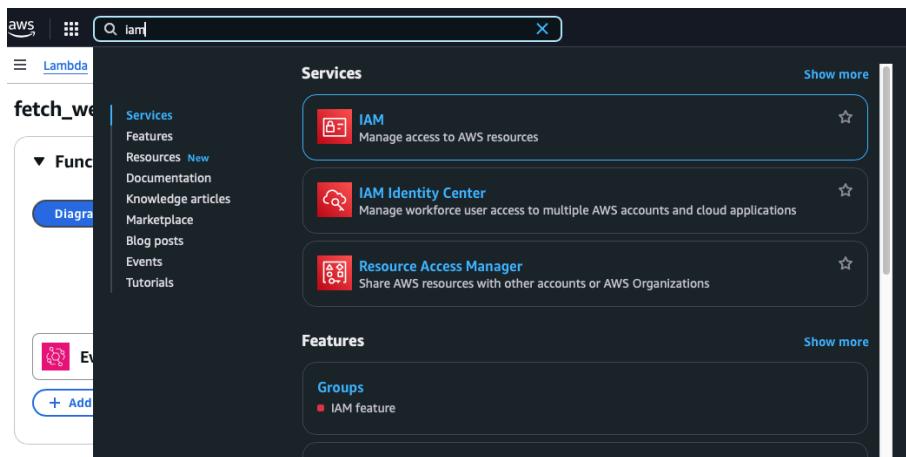


```
-- Create a new database
CREATE DATABASE DE_PROJECT;

-- Switch to the newly created database
USE DATABASE DE_PROJECT;

-- Create table to load CSV data
CREATE OR REPLACE TABLE weather_data(
    CITY      VARCHAR(128),
    LAT      NUMBER(10, 8),
    temp     NUMBER(10, 0),
    wind_spd  NUMBER(10, 5),
    wind_dir  VARCHAR(128),
    pressure_mb NUMBER(12, 5),
    humidity  NUMBER(10, 5)
);
```

Get snowflake script from my git repo and run this bit



aws | Search iam

Lambda

fetch\_weather

Function

Diagram

Event

+ Add

Services

- Features
- Resources New
- Documentation
- Knowledge articles
- Marketplace
- Blog posts
- Events
- Tutorials

Services

- IAM Manage access to AWS resources
- IAM Identity Center Manage workforce user access to multiple AWS accounts and cloud applications
- Resource Access Manager Share AWS resources with other accounts or AWS Organizations

Features

- Groups IAM feature

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected. The main area displays a table of roles with columns for 'Role name', 'Trusted entities', and 'Last activity'. A red arrow points to the 'Create role' button at the top right of the table.

Role name	Trusted entities	Last activity
AWS_TO_SNOWFLAKE	Account: 084828560849	19 hours ago
AWSServiceRoleForSupport	AWS Service: support(Service-Links)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor(Service)	-
db_to_c3-role-gcmibucc	AWS Service: lambda	13 minutes ago
fetch_weather_data_2-role-ybo0eqz2	AWS Service: lambda	16 minutes ago
fetch_weather_data-role-okq5crl41	AWS Service: lambda	17 minutes ago

This screenshot shows the 'Select trusted entity' step of the IAM role creation wizard. It includes a sidebar with steps 1 through 3. The main area has sections for 'Trusted entity type' (with 'AWS account' selected), 'An AWS account' (with 'This account (051826723074)' selected), and options for 'Require external ID' and 'Require MFA'. Red arrows point from the 'AWS account' selection in the first section to the account choice in the second, and from the 'Next' button at the bottom right to the 'Next' button in the bottom right corner of the main window.

Step 1 Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Add permissions [Info](#)

Permissions policies (1/1038) [Info](#)

Choose one or more policies to attach to your new role.

Filter by Type [All types](#) [15 matches](#)

Policy name	Type	Description
AmazonDMSReaderS3Backup	AWS managed	Provides access to manage S3 settings...
<b>AmazonS3FullAccess</b>	AWS managed	Provides full access to all buckets via t...
AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permis...
AmazonS3outpostsFullAccess	AWS managed	Provides full access to Amazon S3 on ...
AmazonS3outpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
AmazonS3ReadonlyAccess	AWS managed	Provides read only access to all bucket...
AmazonS3TableFullAccess	AWS managed	Provides full access to all S3 table bu...
AmazonS3TablesReadOnlyAccess	AWS managed	Provides read only access to all S3 tabl...
AWSBackupServiceRolePolicyForS3Backup	AWS managed	Policy containing permissions necessar...
AWSBackupServiceRolePolicyForS3Restore	AWS managed	Policy containing permissions necessar...
AWSQuickSetupSSMDeploymentS3BucketRolePolicy	AWS managed	This policy grants permissions for list...
QuickSightAccessForS3StorageManagementAnalyticsRe...	AWS managed	Policy used by QuickSight team to acc...
S3UnlockBucketPolicy	AWS managed	Provides access required to unlock a S...

▶ Set permissions boundary - *optional*

[Next](#) [Previous](#)

aws Search [Option+S] Global raghul\_jakar

IAM > Roles > Create role

Step 1  
Select trusted entity  
Step 2  
Add permissions  
Step 3  
Name, review, and create

### Name, review, and create

**Role details**

**Role name**  
Enter a meaningful name to identify the role  
**AWS\_TO\_SNOWFLAKE** 

Maximum 64 characters. Use alphanumeric and "+", "-", "\_" characters.

**Description**  
Add a short explanation for this role.  


Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: "+", "-", "/", "\", "%", "^", "!", "-"

### Step 1: Select trusted entities

**Trust policy** 

```
1 <[{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Action": "sts:AssumeRole", "Principal": {"AWS": "951826723074"}, "Condition": {}}], 2 ]>
```

### Step 2: Add permissions

**Permissions policy summary** 

Policy name	Type	Attached as
 <b>Attached as</b>		

AWS IAM > Roles > AWS\_TO\_SNOWFLAKE

**AWS\_TO\_SNOWFLAKE**

**Summary**

Creation date: February 24, 2025, 21:45 (UTC)

Last activity: 20 hours ago

ARN: arn:aws:iam::051826723074:role/AWS\_TO\_SNOWFLAKE

Maximum session duration: 1 hour

**Link to switch roles in console:** https://signin.aws.amazon.com/switchrole?roleName=AWS\_TO\_SNOWFLAKE&account=051826723074

**Permissions** | Trust relationships | Tags | Last Accessed | Revoke sessions

**Permissions policies (1)**

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonS3FullAccess	AWS managed	Z

**Permissions boundary (not set)**

**Generate policy based on CloudTrail events**

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

[Generate policy](#)

No requests to generate a policy in the past 7 days.

--Create integration object for external stage  
create or replace storage integration s3\_int  
type = external\_stage  
storage\_provider = s3  
enabled = true  
storage\_aws\_role\_arn = '[arn:aws:iam::051826723074:role/AWS\\_TO\\_SNOWFLAKE](#)'  
storage\_allowed\_locations = ('s3://project-weather-raghul/snowflake/');

**past it here and run this sql**

```

DE_PROJECT.PUBLIC  Settings ▾

27
28
29 --Describe integration object to fetch external_id and to be used in s3
30 DESC INTEGRATION s3_int; Red arrow pointing here
31
32 create or replace file format csv_format
33   type = csv
34   field_delimiter = ','
35   skip_header = 1
36   null_if = ('NULL', 'null')
37

```

↳ Results ▾ Chart

	property	property_type	property_value	property_default
1	ENABLED	Boolean	true	false
2	STORAGE_PROVIDER	String	S3	
3	STORAGE_ALLOWED_LOCATIONS	List	s3://project-weather-raghul/snowflake/	
4	STORAGE_BLOCKED_LOCATIONS	List		
5	STORAGE_AWS_JAM_USER_ARN	String	arn:aws:iam::084828560849:user/j46w0000-s	
6	STORAGE_AWS_ROLE_ARN	String	arn:aws:iam::051826723074:role/AWS_TO_SNOWFLAKE	
7	STORAGE_AWS_EXTERNAL_ID	String	UY39891_SFRole=2_e3geLr+X4KilxYwmCEOehfMosI=	
8	USE_PRIVATELINK_ENDPOINT	Boolean	false	false
9	COMMENT	String		

Copy both and  
use it in AWS to  
build trust

IAM > Roles > AWS\_TO\_SNOWFLAKE

**AWS\_TO\_SNOWFLAKE** [Info] Delete Edit

Identity and Access Management (IAM) Global raghul\_sekar

Search IAM

Summary

Creation date: February 24, 2025, 21:43 (UTC)

Last activity: 20 hours ago

ARN: arn:aws:iam::051826723074:role/AWS\_TO\_SNOWFLAKE

Maximum session duration: 1 hour

Link to switch roles in console: [https://signin.aws.amazon.com/switchrole?roleName=AWS\\_TO\\_SNOWFLAKE&account=051826723074](https://signin.aws.amazon.com/switchrole?roleName=AWS_TO_SNOWFLAKE&account=051826723074)

Permissions Trust relationships Tags Last Accessed Revoke sessions

Trusted entities

In role past it here

Entities that can assume this role under specified conditions.

```

1+ [
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "AWS": "arn:aws:iam::084828560849:user/j46w0000-s"
8       },
9       "Action": "sts:AssumeRole",
10      "Condition": {
11        "StringEquals": {
12          "sts:ExternalId": "UY39891_SFRole=2_e3geLr+X4KilxYwmCEOehfMosI="
13        }
14      }
15    }
16  ]
17 ];

```

Edit trust policy

IAM Identity Center AWS Organizations

```

39   create or replace stage ext_csv_stage
40     URL = 's3://project-weather-raghul/snowflake/'
41     STORAGE_INTEGRATION = s3_int
42     file_format = csv_format;
43
44   list @ext_csv_stage;

```

↳ Results ~ Chart We can see tables in stage using above sql line

	name	#	siz	md5	last_modified
1	s3://project-weather-raghul/snowflake/weather_data_2025-02-24_22-51-32.csv	111	9929dee525eb123f3489924492875273		Mon, 24 Feb 2025 22:51:3-
2	s3://project-weather-raghul/snowflake/weather_data_2025-02-24_22-51-33.csv	113	96f040350542d78c17d820137636378		Mon, 24 Feb 2025 22:51:3-
3	s3://project-weather-raghul/snowflake/weather_data_2025-02-24_22-51-34.csv	116	0c2da89ff9b06a262eb444ff582e38fb		Mon, 24 Feb 2025 22:51:3-
4	s3://project-weather-raghul/snowflake/weather_data_2025-02-24_23-08-37.csv	114	0d79aca5f0c6f9ff80e63ffffbc44d921		Mon, 24 Feb 2025 23:08:3
5	copy the pipe notification and link with s3	100	0	0	Mon, 24 Feb 2025 23:08:30

```
--create pipe to automate data ingestion from s3 to snowflake
create or replace pipe mypipe auto_ingest=true as
copy into weather_data
from @ext_csv_stage
on_error = CONTINUE;
```

```
show pipes;
```

52 | show pipes;

53

↳ Results ~ Chart copy the pipe notification and link with s3

	owner	notification_channel
1	ACCOUNTADMIN	arn:aws:sqs:us-east-1:084828560849:sf-snowpipe-AIDARHQBNDXISEUC35BGB-vF26:

AWS S3 Bucket: project-weather-raghul

Objects (1)

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

Actions: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload

AWS S3 Bucket: project-weather-raghul

Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) arn:aws:s3:::project-weather-raghul	Creation date February 24, 2025, 18:30:56 (UTC+00:00)
-----------------------------------------------	-------------------------------------------------------------------	----------------------------------------------------------

Event notifications (1)

Send a notification when specific events occur in your bucket. [Learn more](#)

Name	Event types	Filters	Destination type	Destination
sf_event_notification	All object create events	-	SQS queue	arn:aws:sqs:us-east-1:084828560849:sf-snowpipe-AIDAHQBNDX5EUCP\$IGB-vF26s70EWFASG6rM_bpeQ

Amazon EventBridge

For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. [Learn more](#) or see [EventBridge pricing](#)

Send notifications to Amazon EventBridge for all events in this bucket

Off

Edit

Create event notification

aws [ Option+S ] United States (N. Virginia) raghul\_sekar

Amazon S3 > Buckets > project-weather-raghul > Edit event notification

### Edit event notification Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

#### General configuration

**Event name**  
sf\_event\_notification

**Prefix - optional**  
Limit the notifications to objects with key starting with specified characters.  
`/images/`

**Suffix - optional**  
Limit the notifications to objects with key ending with specified characters.  
`.jpg`

#### Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

##### Object creation

All object create events `$s3:ObjectCreated:`

Put `$s3:ObjectCreated:Put`

Post `$s3:ObjectCreated:Post`

Copy `$s3:ObjectCreated:Copy`

Multipart upload completed `$s3:ObjectCreated:CompleteMultipartUpload`

#### Destination

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

**Destination**  
Choose a destination to publish the event. [Learn more](#)

Lambda function  
Run a Lambda function script based on S3 events.

SNS topic  
Fanout messages to systems for parallel processing or directly to people.

SQS queue  
Send notifications to an SQS queue to be read by a server.

**Specify SQS queue**

Choose from your queues

Enter SQS queue ARN

**SQS queue**  
`arn:aws:sqs:sus-east-1:084828560849:snowpipe-AIDARHQBNDXIEUC35BGB-fV26t70EWFASG6raM_bpQ`

**Save changes**

This will set the event notification to snowpipe when data is loaded in s3

54   | select \* from weather\_data;

55

Results Chart

Once notification triggered the snowpipe it starts loading the data into table

	CITY	TIME	TEMP	WIND_SPEED	WIND_DIR	PRESSURE_MB	HUMIDITY
1	Kashmir	2025-02-25T01:08:37.070723	20	5.80000	N	1016.00000	61.00000
2	Sikkim	2025-02-25T01:08:38.038571	0	5.40000	NNE	1023.00000	74.00000
3	Coimbatore	2025-02-25T02:08:35.437119	21	4.30000	ENE	1018.00000	88.00000
4	Sikkim	2025-02-25T02:08:37.855414	3	2.20000	NNE	1022.00000	65.00000
5	Kashmir	2025-02-25T03:08:37.126393	20	6.00000	NNW	1017.00000	58.00000

Query Details

Query duration 63ms

Rows 59

Query ID 01baa27e-0001-9608-0-[Show more](#)

## **8. Conclusion**

Once all the steps are completed, data will be loaded into DynamoDB every hour, transferred to S3, and then ingested into the Snowflake table.