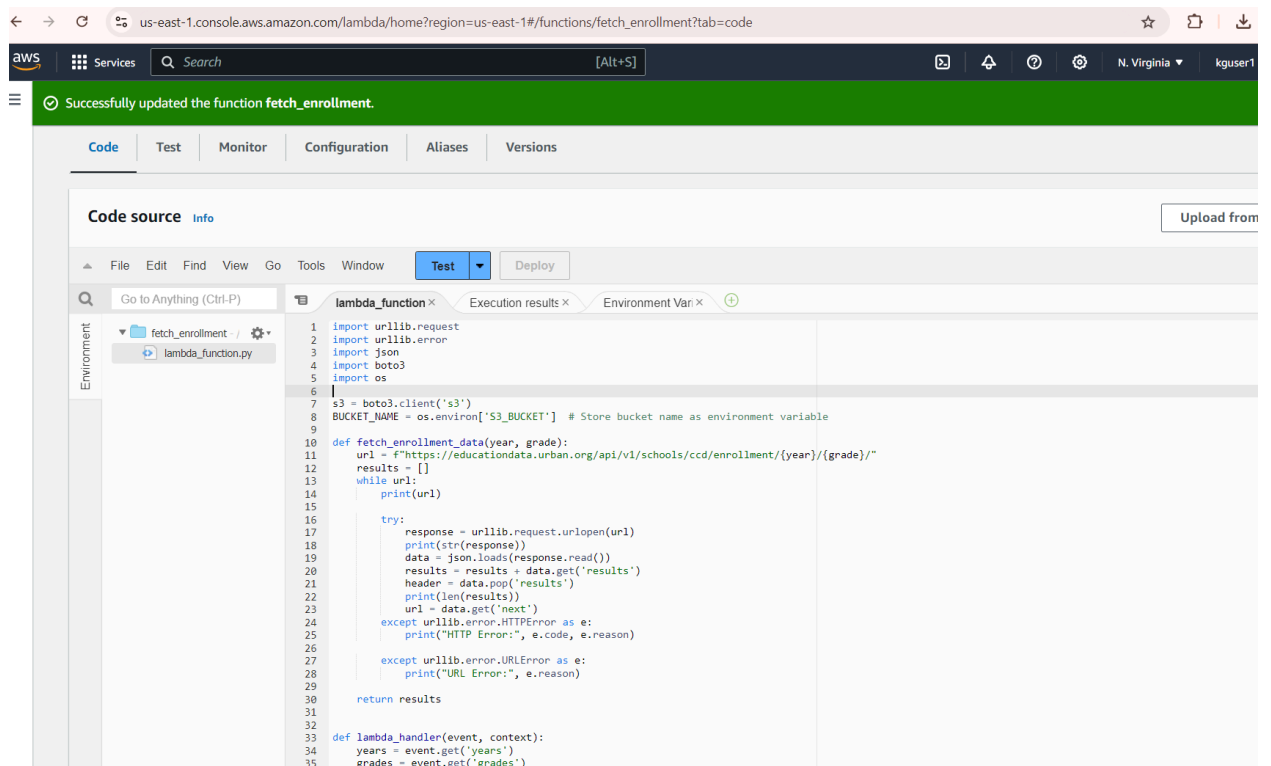


## Approach

### Fetch The Data

1. AWS Lambda suits this use case better as the source for the data is a REST endpoint.
2. This API doesn't need any credentials, if it needs any credentials AWS Secret Manager would be useful to store and retrieve the credentials.
3. There is no incremental parameter on the REST API endpoint like date or incremental id, so the incremental run would pull all the available data for the year and grade on a regular schedule. If there is an incremental way of pulling the data, we could use the db table like dynamo db to store the field value to be used for incremental data extract.



The screenshot shows the AWS Lambda console interface for the function 'fetch\_enrollment'. The 'Code source' tab is selected, displaying the Python code for the lambda function. The code imports urllib, json, boto3, and os. It defines a function 'fetch\_enrollment\_data' that takes 'year' and 'grade' as parameters. Inside this function, it constructs a URL to the 'educationdata.urban.org' API, fetches the data using urllib, and returns the results. A 'lambda\_handler' function is also defined, which calls 'fetch\_enrollment\_data' with parameters from the event object.

```
1 import urllib.request
2 import urllib.error
3 import json
4 import boto3
5 import os
6
7 s3 = boto3.client('s3')
8 BUCKET_NAME = os.environ['S3_BUCKET'] # Store bucket name as environment variable
9
10 def fetch_enrollment_data(year, grade):
11     url = f"https://educationdata.urban.org/api/v1/schools/ccd/enrollment/{year}/{grade}/"
12     results = []
13     while url:
14         print(url)
15
16         try:
17             response = urllib.request.urlopen(url)
18             print(str(response))
19             data = json.loads(response.read())
20             results = results + data.get('results')
21             header = data.pop('results')
22             print(len(results))
23             url = data.get('next')
24         except urllib.error.HTTPError as e:
25             print("HTTP Error:", e.code, e.reason)
26         except urllib.error.URLError as e:
27             print("URL Error:", e.reason)
28
29     return results
30
31
32 def lambda_handler(event, context):
33     years = event.get('years')
34     grades = event.get('grades')
```

4. Scheduled the Lambda using AWS EventBridge scheduler to run every 12 hours. The scheduler passes the event with year and grade parameters to the Lambda function, this gives flexibility to expand the dataset with changing the code.
  - a. { "years": [2020, 2021], "grades": ["grade-pk"] }

us-east-1.console.aws.amazon.com/scheduler/home?region=us-east-1#schedules/default/ccd-fetch-enrollments-scheduler

Amazon EventBridge

cc-fetch-enrollments-scheduler

Enable Edit

### Schedule detail

Schedule name ccd-fetch-enrollments-scheduler	Status ⊖ Disabled	Schedule start time -	Flexible time window -
Description -	Schedule ARN arn:aws:scheduler:us-east-1:677276097216:schedule/default/cc-fetch-enrollments-scheduler	Schedule end time -	Created date Sep 19, 2024, 22:57:51 (UTC-04:00)
Schedule group name default	Action after completion NONE	Execution time zone America/New_York	Last modified date Sep 20, 2024, 16:14:03 (UTC-04:00)

Schedule Target Retry policy Dead-letter queue Encryption

### Schedule

Fixed rate [Info](#)

rate (12 hours)

5. AWS Glue is used to create the catalog table for the S3 data

us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data-catalog/tables/view/us\_education\_ccd\_enrollment\_data?database=ccd\_school&catalogId=67727...

AWS Glue

us\_education\_ccd\_enrollment\_data

Last updated (UTC) September 20, 2024 at 21:13:15 Version 3 (Current version)

Table overview Data quality - new

### Table details

Name us_education_ccd_enrollment_data	Classification JSON	Deprecated -
Database ccd_school	Location s3://us-education-ccd-enrollment-data/	Column statistics <a href="#">No statistics</a>
Description -	Connection -	
Last updated September 20, 2024 at 03:41:00		

► Advanced properties

6. AWS Athena is used to query the data

The screenshot displays the AWS Athena console interface. On the left, the 'Data' sidebar shows the 'Data source' as 'AwsDataCatalog' and the 'Database' as 'ccd\_school'. Under 'Tables and views', the table 'us\_education\_ccd\_enrollment\_data' is listed. The main area shows a SQL query titled 'top\_ten\_pk\_states' that ranks states by total enrollment for the year 2021. The query results are displayed in a table with columns: #, state, total\_enrollments, and rnk. The results show Texas as the top state with 245135 enrollments.

```
1 with enrollments_by_state as (  
2   select fips as state_id, sum(enrollment) as total_enrollments,  
3   rank() over (order by sum(enrollment) desc) as rnk  
4   from ccd_school.us_education_ccd_enrollment_data limit  
5   where year=2021  
6   group by fips  
7 )  
8 SELECT  
9   CASE state_id  
10    WHEN 1 THEN 'Alabama'  
11    WHEN 2 THEN 'Alaska'  
12    WHEN 3 THEN 'American Samoa'  
13    WHEN 4 THEN 'Arizona'  
14    WHEN 5 THEN 'Arkansas'  
15    WHEN 6 THEN 'California'
```

#	state	total_enrollments	rnk
1	Texas	245135	1

Possible Improvements that would need more time:

1. Cloud Formation templates could be used for creating infrastructure components
2. Could build CI/CD pipeline using tools like Github Actions or Jenkins to automate the build and deployment