1) Assemble a small data file, that contains a list of JSON objects, **at least 20 to 30 objects**. Use data that comes from fakestore API or some other site.

2) Place the data file on an S3 bucket. You can use the same bucket you used for your web page assignment.

The file should be accessible through a URL. Give it appropriate ACL public permissions and enable CORS.

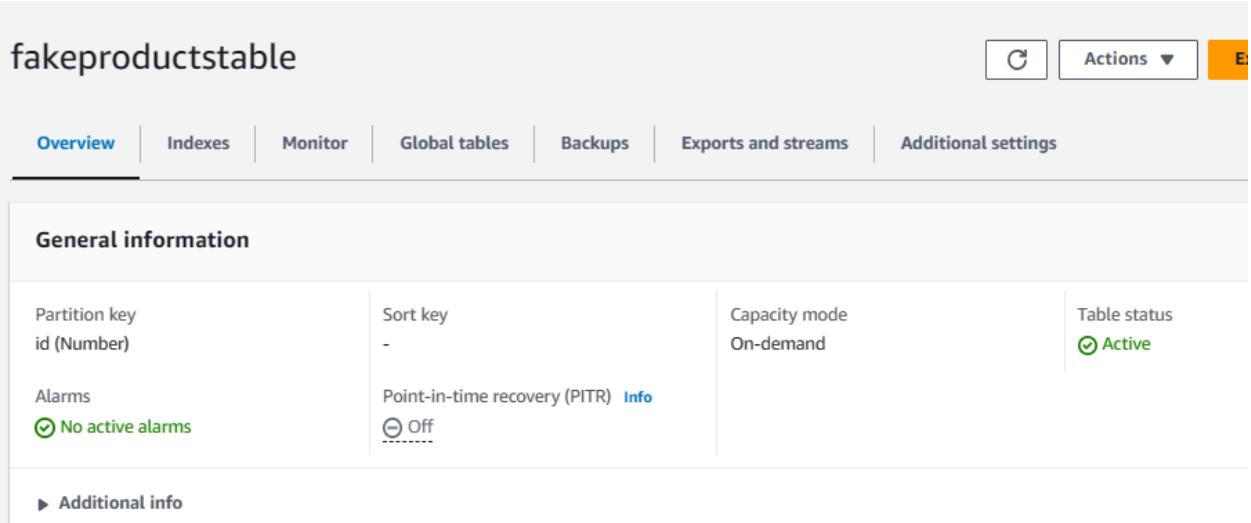
NOTES:

Please check Enable ACL (access control lists) when you create the bucket. That way you'd be able to make objects in the bucket public, by ACL.

Please make sure that the .json datafile that shows on the bucket link is well formed. That is, a list of json objects, each object is a complete object. If the data is not well formed, it will be rejected by DynamoDB.

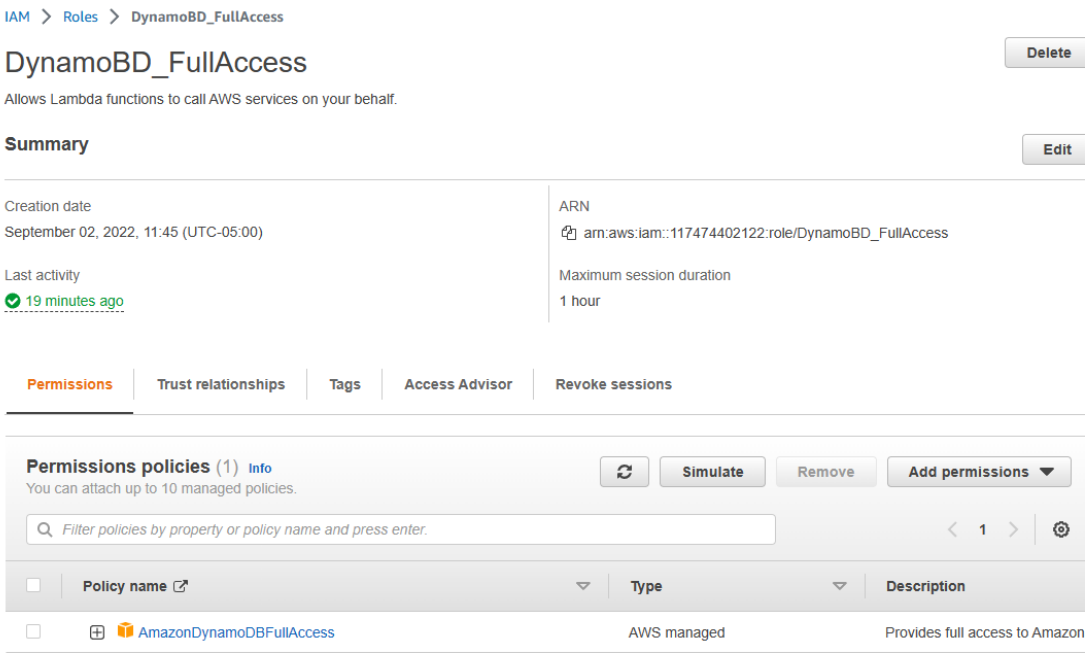
[{ }, { }, { }, { }, { }]

3) Create a new DynamoDB table, make sure the Partition key agrees with what you'd use as index in the data file. My sample data file uses id (Number)



Good practice: Actions, Edit Capacity, **on-demand** so that you won't get charged just for having the table.

4) Create an execution role for the Lambda function with a policy to access DynamoDB.



5) Write a Lambda function (**Runtime Python 3.13**) that reads the data from the URL of step 2) and batch writes it to the DynamoDB table. The function uses **Environment variables**.

// Code given in class

You may need to increase the **timeout** of the function.

6) If all OK, Deploy and Test the function to see that the function has added the Items to the table! It may take some time to get the table updated with the new Items!