Lab DynamoDB

Marino Oliveros Blanco NIU:1668563
Pere Mayol Carbonell NIU:1669503

Github Link: https://github.com/marinocom/Cloud-Lab-DM-GIA

Lab description:
Design of a cloud architecture to implement a service
We need a new weather simulation system that divides the world in 20 km long side squares. It should distribute the processing of the information of each grid square using the database system features of data partition.

We want to have a distributed database so that read and write operations could be located in a different partition depending on the location of the user that makes the request.

Session 2:
Design a new DynamoDB database architecture for a new application that will simulate the weather conditions of a list of distributed regions. You have to design a table, its attributes and a JSON file with some initial data points to make some queries during the next session. The design is open-ended, and the student must propose and justify their architecture.

Deliverable: Document explaining the design of the proposed database. The proposed architecture must be justified, detailing each database attribute, its type, and the reasoning behind its necessity. Additionally, an example of a JSON load file must be included.

Design of the Database.

Firstly, an idea of the database structure was though out to ensure it made sense, that the set partition and set key made sense, and also add the appropriate attributes ensuring the database is correctly built.

Primary key:

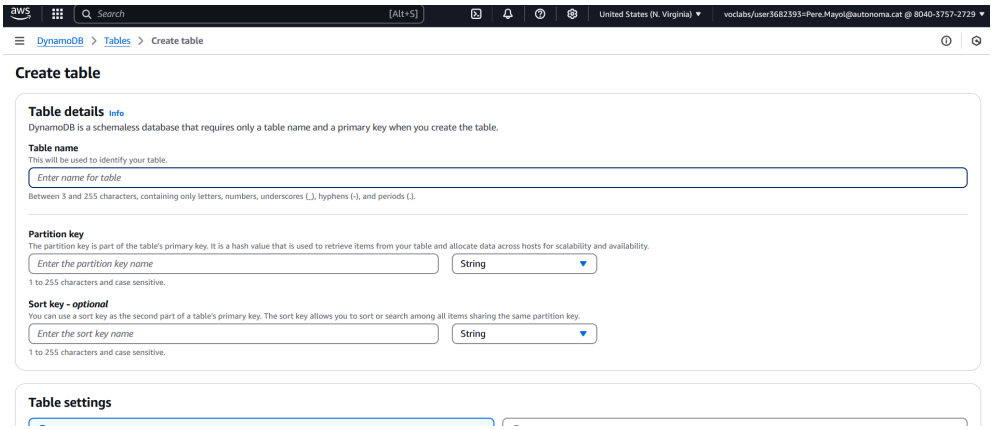
1. **Partition Key ⇒ RegionID [String]**, this is simply to ensure each region is identified using a descriptive string, to differentiate them easily. An example here could be “region_001”, depending on the number of regions or if more information is needed, the plasticity given by the data type could make us specify more.
2. **Sort Key ⇒ Timestamp [String]**, to capture at which time exactly all information is gathered, to make differentiations between same regions at different time steps. This would ensure the data for a certain region could be sorted a long time. An example here would be “01-Jan-2025 1:00 AM”.

Attributes

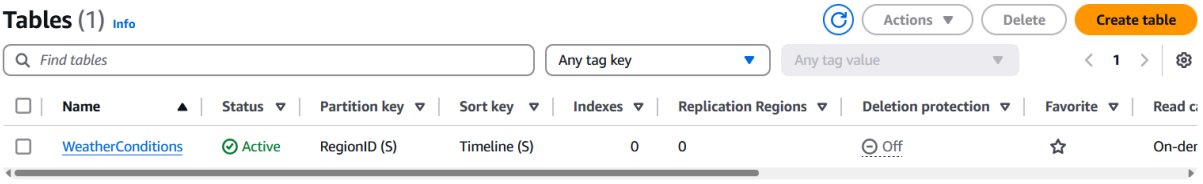
1. **Temperature [Number],** to capture temperature information for each region at each time step. We are using Celsius. Could be used to calculate important and significant insights on different areas.
2. **Humidity [Number],** percentage of humidity, important for patterns and understanding the weather completely.
3. **WindVelocity [Number],** integer to store the velocity of wind.
4. **Conditions [String],** string to capture the state of the weather in a general way, check for clouds or how the sky seems. Could also be used in a more descriptive way.
5. **Pressure [Number],** complement information we already have and have a more profound understanding of the general condition. We have used hectopascals as a unit.

Design of the Database in AWS Academy Leaner Lab.

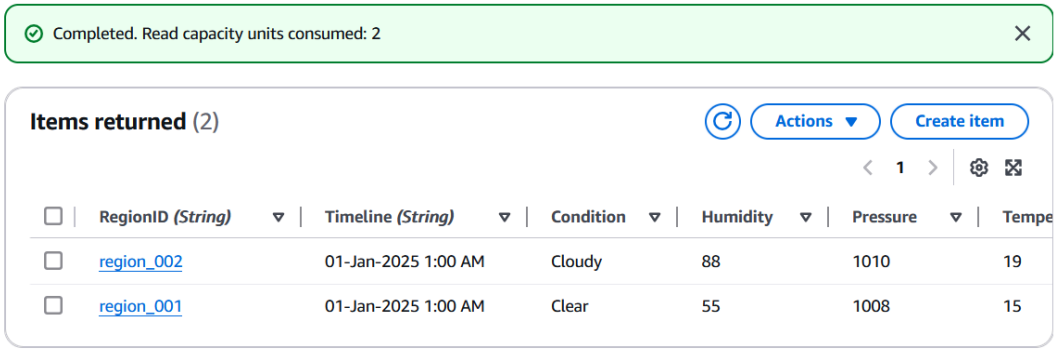
1. Enter the AWS Academy Leaner Lab and search for DynamoDB.
2. Create a new table and set the name as “WeatherConditions”, and its partition and sort keys as the ones mentioned before.



3. Wait until our newly created table is ready.



4. Now we can generate two regions and a random time step, so we can have some objects in the table to work with after.



They can be downloaded as a CSV:

	A	B	C	D	E	F	G
1	RegionID	Timeline	Conditi...	Humidity	Pressure	Temperat...	WindVelo...
2	region_0...	01-Jan-2025 1:00 A...	Cloudy	88	1010	19	35
3	region_0...	01-Jan-2025 1:00 A...	Clear	55	1008	15	10