The model I chose for the database is DynamoDB with provisioned capacity of 20 reads and 20 writes.

That means that under the current configuration, it is possible to perform 20 reads and 20 writes per second , which in our scenario would be suitable for about 1,200 users.

With provisioned capacity it is also possible to set an autoscaler to adjust the table’s capacity based on the specified utilization rate and ensure application performance in the case of the app usage expanding to hundreds and thousands of users.

In this case, I have one table that contains all the information. The table has 2 types of entities – users and groups. It is built in such a way that all queries depend only on the field ‘id’, and therefore don’t overload the database with complicated queries.

Given an infrequent access mode, the cost would be about 66$ a month. Frequent access would raise the cost to about 180$ upfront and additional 12$ monthly. This is assuming that the database total size is under 1GB.

The average item size in the table is 120B. hundreds of users would result in about 12,000B, which is less than a GB and would not increase the cost. In fact, we would only need to increase the database size to over a GB when we reach about 8 million users.

In terms of read and write actions, more actions would result at a higher cost. The numbers shown above correspond to 100 reads and 100 writes per second ~ 6,000 a minute, which is suitable for hundreds of users.

Thousands of users would cost 170,000$ upfront and 10,000$ monthly, and a million users would cost 165,000$ and 1,000$ monthly.