**AWS Turn-based game**

The game is running on a virtual environment on Cloud9 service. Players will have to post a json object contains their game ID, player ID and movement, which is changedHeap and changedHeapValue, to a game API to play the game. Each player will receive a message to their phone every time the other player has made their move. When the game is over, one player will receive the message to their phone informs that he/she has won the game, and the other will receive the message informs that he/she has lost the game. Firstly, the NIM game uses Cloud9 to create a virtual environment to store the game then it uses DynamoDB to create a database to store any game related information, such as game ID, users, heap, and the last player that made the move. The table has 7 column which is used to store gameId, heap1, heap2, heap3, which stores the movement of each player, lastMoveBy, which stores the last player that has made the move, and user1, user2 to store player’s username. Later, all of this data will be sent to each player’s phone to update the game’s current state. NIM also uses Simple Notification Service (SNS) to send messages to players to inform them of their opponent’s movement. There is a file called “sendMessage.js” that used to handle message sending to each player. The method “sendMessage()” is used to actually send the message in “message” parameter to the phone number given in “phoneNumber” parameter. Besides, NIM uses Amazon Cognito as an authentication method for create user, user login, get user information, and verify user. First the game creates a user pool to store users then create user pool client to call unauthorized method in user pool. The “login” method is used for authenticating if that user is stored in database or not, then the “verifyToken” is used for authorizing which action that user can do during the game. Finally, the game uses API Gateway to deploy the game’s API and Lambda Function, IAM to handle any request from the API to back-end database. This is a serverless application because Lambda Function and API Gateway are used to deploy the game and handle server-side works; for example, handle communication between API and back-end database or send notification to players, which mean I do not need to do anything related to server-side works.