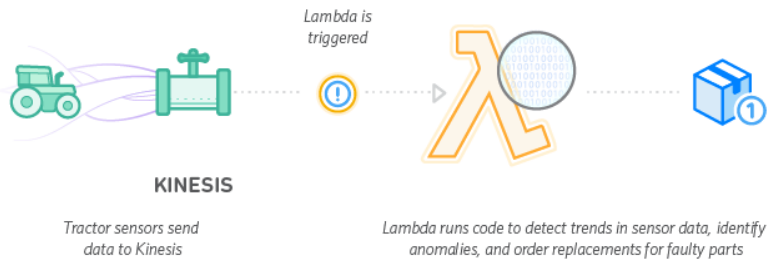


AWS Final Project

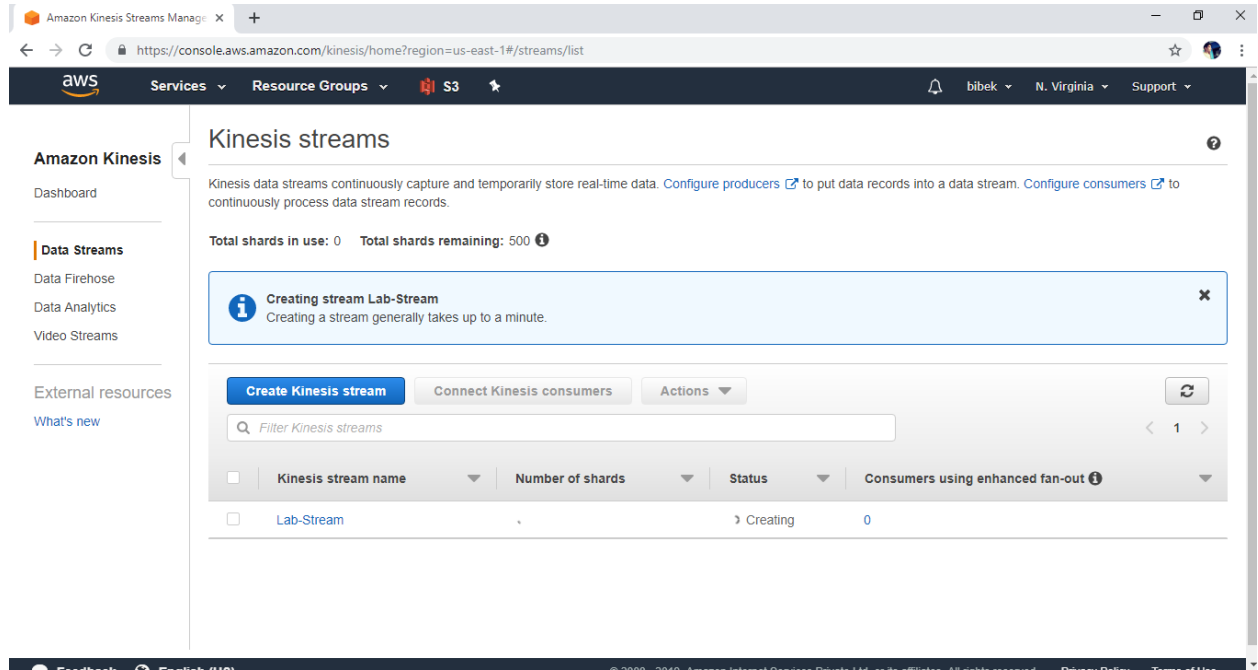
Serverless Architectures with Amazon DynamoDB and Amazon Kinesis Streams with AWS Lambda

Submitted by : Bibek Kumar Agrawal

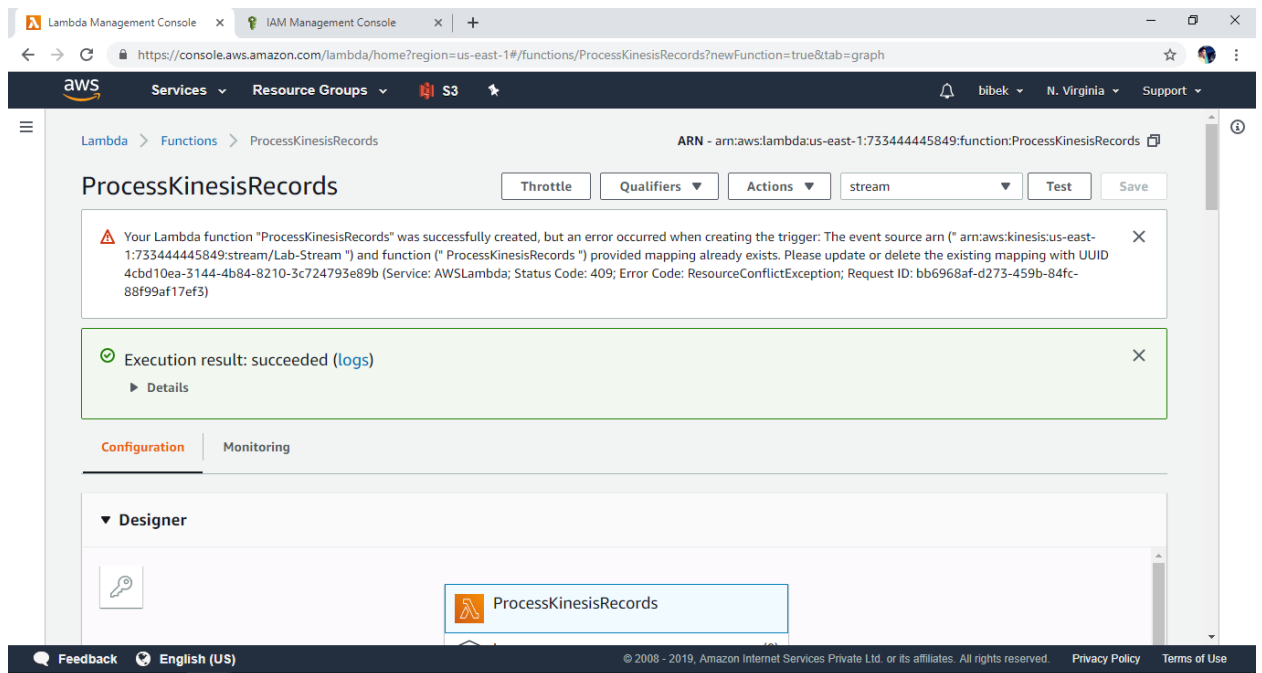
Example: Sensors in Tractor Detect Need for a Spare Part and Automatically Place Order



1. Create an Amazon kinesis stream.



2. Create lambda function and test it.



3. Create tables in Dynamo DB.

The screenshot shows the AWS DynamoDB console interface. On the left, the navigation menu includes DynamoDB, Dashboard, Tables, Backups, Reserved capacity, and Preferences. The 'Tables' section is selected, showing a list of tables with columns for Name and Status. The table 'GameScoreRecords' is listed with a status of 'Active'. The main panel displays the 'GameScoreRecords' table overview, including tabs for Overview, Items, Metrics, Alarms, Capacity, Indexes, Global Tables, and More. The 'Overview' tab is active, showing 'Recent alerts' (No CloudWatch alarms have been triggered for this table.) and 'Stream details' (Stream enabled: Yes, View type: New image, Latest stream ARN: arn:aws:dynamodb:us-east-1:733444445849:table/GameScoreRecords/stream/2018-09T08:46:53.806). The 'Table details' section shows the Table name (GameScoreRecords), Primary partition key (RecordID (Number)), Primary sort key (-), Point-in-time recovery (DISABLED, Enable), and Execution type (DEFAULT, Manage Execution).

4. Create a lambda function (Author from scratch) and test it.

The screenshot shows the AWS Lambda console interface for the 'AggregateScoresByUser' function. The function is in the 'Author from scratch' state. The code editor displays the following JavaScript code:

```
16 // Set the appropriate parameters for UpdateItem
17 // Refer to the ADD operation in the UpdateItem API for UpdateExpression
18 // http://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_UpdateItem.html
19 // Adds the specified value to the item, if attribute does not exist, set the attribute
20
21 var updateItemParams = {
22   TableName: "GameScoresByUser",
23   Key: { Username: newItemImage.Username },
24   UpdateExpression: 'ADD Score :attrValue',
25   ExpressionAttributeValues: { ':attrValue': newItemImage.Score }
26 }
27
28 // Make a callback function to execute once UpdateItem request completes
29 // It may be helpful to refer to the updateItem method for the Javascript SDK
30 // http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/DynamoDB.html#updateItem-property
31 var updateItemCallback = function(err, data) {
32   if (err) {
33     // log errors
34     console.log(err, err.stack);
35   } else {
36     // check if all requests are finished, if so, end the function
37     inflightRequests--;
38     if (inflightRequests === 0) {
39       context.succeed("Successfully processed " + event.Records.length + " records.");
40     }
41   }
42 }
43
44 // Send UpdateItem request to DynamoDB
45 dynamodb.updateItem(updateItemParams, updateItemCallback);
46 // Increase count for number of requests in flight
```

5. Verify in dynamo DB.

The screenshot shows the AWS DynamoDB console interface. On the left, the navigation pane lists various services, with 'DynamoDB' selected. Under 'DynamoDB', 'Tables' is chosen, and 'GameScoresByUser' is selected from the list. The main panel displays the 'GameScoresByUser' table details. The 'Items' tab is active, showing a single item with the key 'Username' and value 'Jane Doe', and a value 'Score' of 200. The table's scan configuration is set to 'Scan' and '[Table] GameScoresByUser: Username'.

Username	Score
Jane Doe	200

6. Trigger the update by updating values in scores table and confirming the updates in user tables.

The screenshot shows the AWS DynamoDB console interface, similar to the previous one, but with two items in the 'GameScoresByUser' table. The 'Items' tab is active, showing two items: 'Jane Doe' with a score of 200 and 'Shreya' with a score of 80. The table's scan configuration is set to 'Scan' and '[Table] GameScoresByUser: Username'.

Username	Score
Jane Doe	200
Shreya	80