DynamoDB Cross Region Replication using Kinesis.

Introduction

This document serves as a guide to setting up the Proof of Concept (POC) DynamoDB Cross Region Replication environment. It outlines the requirements and underlying services used within it's architecture. Sample code is added to showcase the abilities of the architecture but should not in any capacity be used within Production as is.

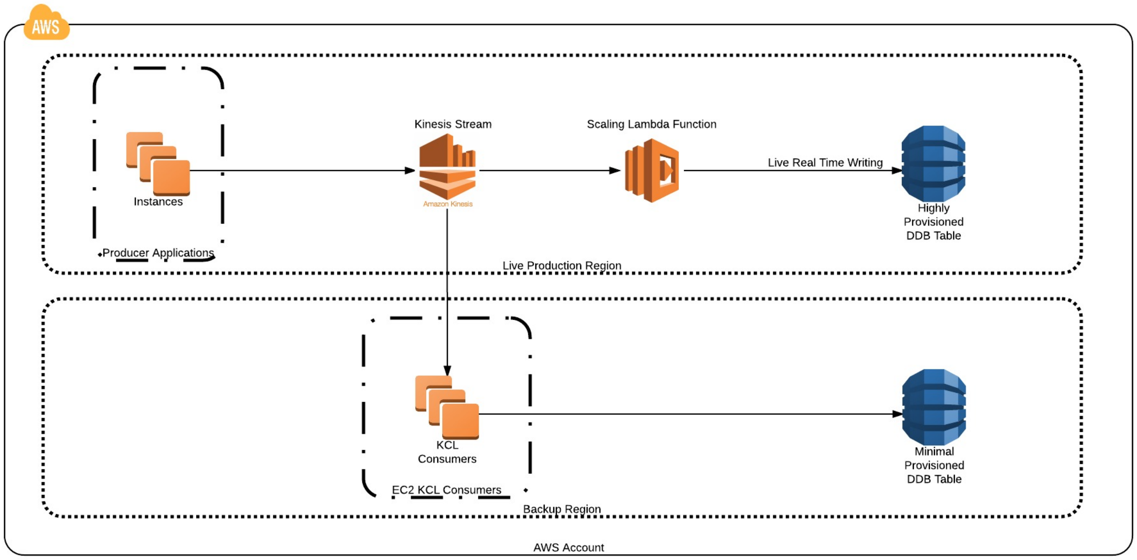
Outcome

This architecture will allow the user to write items to a production DynamoDB table provisioned for immediate access through high provisioning while lazily keeping track and duplicating the table in other environments and cross region.

Requirements

1. DynamoDB
2. Kinesis Stream
3. Kinesis KCL (Any Available Language)
4. Lambda NodeJS Function

Architecture



Producer Application

The producer application is the source of the data which will be submitted to the Kinesis stream. This essentially will lead to writing the data to DynamoDB and also function as the BinLog for the DynamoDB table that can be replayed.

Example Data:
{ Action: “PutItem”, Data: “This is my test data” }
{ Action: “BatchWrite”, Data: [{ item1: value },{ item2: value2 }] }

Lambda Function

The Lambda function is a singular function that can detect any changes within the Kinesis Stream. Data from the stream will be pushed to the Lambda function where the data can then be written to the DynamoDB Table. It is within this step where the data inside the Kinesis stream will be interpreted, transformed and written to DynamoDB.

To create and set up a Lambda Function follow the documentation found here:
http://docs.aws.amazon.com/lambda/latest/dg/walkthrough-kinesis-events-adminuser.html

Sample Code: https://github.com/mombergm/DynamoDB-Lazy-Replication/blob/master/Lambda-Consumer/avgKinesisConsumer.js

Lazy Consumer KCL application

By leveraging the Amazon Kinesis KCL harness, the information stored inside the Kinesis Stream can easily be accessed and replayed lazily at the performance the user requires. The KCL is a mirror of the Lambda Function which operates at a lower throughput, lazily consuming records over a period of 24 hours.

KCL: http://docs.aws.amazon.com/kinesis/latest/dev/developing-consumer-apps-with-kcl.html
Sample Code: https://github.com/mombergm/DynamoDB-Lazy-Replication/blob/master/Kinesis-KCL/samples/basic_sample/consumer/sample_kcl_app.js