



Amazon DynamoDB

Topics to be covered--Dynamodb

- 1) Paas Introduction
- 2) Dynamodb Introduction
- 3) Creating table
- 4) Creating Items
- 5) Backup
- 6) Integrate dynamodb with lambda and s3 bucket

Amazon DynamoDB

- ✓ **Amazon DynamoDB** is a fully managed NoSQL database service that allows to create database tables that can store and retrieve any amount of data.
- ✓ It automatically manages the data traffic of tables over multiple servers and maintains performance.
- ✓ It also relieves the customers from the burden of operating and scaling a distributed database. Hence, hardware provisioning, setup, configuration, replication, software patching, cluster scaling, etc. is managed by Amazon.

Benefits of Amazon DynamoDB

- ✓ **Managed service** – There is no need to hire experts to manage NoSQL installation. Developers need not worry about setting up, configuring a distributed database cluster, managing ongoing cluster operations, etc. It handles all the complexities of scaling, partitions and re-partitions data over more machine resources to meet I/O performance requirements.
- ✓ **Scalable** – DynamoDB will spread automatically with the amount of data stored as the table grows.
- ✓ **Fast** – Amazon DynamoDB provides high throughput at very low latency. **Durable and highly available** – Amazon DynamoDB replicates data over at least 3 different data centers' results. The system operates and serves data even under various failure conditions.
- ✓ **Flexible:** Amazon DynamoDB allows creation of dynamic tables, i.e. the table can have any number of attributes, including multi-valued attributes.
- ✓ **Cost-effective:** Payment is for what we use without any minimum charges. Its pricing structure is simple and easy to calculate.



Amazon RDS and Amazon DynamoDB


Factors	Relational (Amazon RDS)	NoSQL (Amazon DynamoDB)
Application Type	<ul style="list-style-type: none"> Existing database apps Business process-centric apps 	<ul style="list-style-type: none"> New web-scale applications Large number of small writes and reads
Application Characteristics	<ul style="list-style-type: none"> Relational data models, transactions Complex queries, joins, and updates 	<ul style="list-style-type: none"> Simple data models, transactions Range queries, simple updates
Scaling	Application or DBA-architected (clustering, partitions, sharding)	Seamless, on-demand scaling based on application requirements
QoS	<ul style="list-style-type: none"> Performance—depends on data model, indexing, query, and storage optimization Reliability and availability Durability 	<ul style="list-style-type: none"> Performance—Automatically optimized by the system Reliability and availability Durability

S3	DynamoDB
S3 is file storage.	DynamoDB is a Database.
It fits best for unstructured data.	Best for (semi) structured data
Uses flat organisation called buckets to store files and each bucket contains objects	Uses items and attributes for its tables. Each item contains different number of attributes
Size limit is 5TB	Size limit is 400 kb.
Unit price per GB usage	After throughput is specified, a unit price per GB is charged.

DynamoDB Lab

Services—Dynamodb—Create Table

 Services ▾ Resource Groups ▾ 

 bhimdeepakumar ▾ Mumbai ▾ Support ▾





Amazon DynamoDB


Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications.

[Create table](#)[Getting started guide](#)

DynamoDB Lab

Table Name –Student, Primary Key – Roll-no ----Create

 Services ▾ Resource Groups ▾ 

 bhimdeepakkumar ▾ Mumbai ▾ Support

Create DynamoDB table

Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name*

Student 

Primary key*

Partition key

Roll-no

String ▾ 

☐ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

 You do not have the required role to enable Auto Scaling by default. Please refer to [documentation](#).

+ Add tags

NEW!

DynamoDB Lab

Items – create items

The screenshot displays the AWS Management Console interface for a DynamoDB table named 'Student'. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a star icon. On the left sidebar, there are buttons for 'Create table' and 'Delete table', followed by a search bar 'Filter by table name' and a dropdown 'Choose a table ...'. Below this is a table listing available tables, with 'Student' selected. The main panel shows the 'Items' tab for the 'Student' table. It includes tabs for 'Overview', 'Items', 'Metrics', 'Alarms', 'Capacity', 'Indexes', and 'Global secondary indexes'. There are buttons for 'Create item' and 'Actions'. A search bar shows 'Scan: [Table] Student: Roll-no'. Below this is a filter section with a dropdown 'Scan', a text input '[Table] Student: Roll-no', and an 'Add filter' button. A 'Start search' button is also present. At the bottom, there is a table header with a checkbox, the attribute name 'Roll-no', and an information icon. A light blue box at the bottom right contains the text: 'An item consists of one or more attributes. Each attribute consists of a name required are those that make up the primary key. [More info](#)'.

aws Services Resource Groups

Create table Delete table

Filter by table name X

Choose a table ... Actions

Name
Student

Student Close

Overview Items Metrics Alarms Capacity Indexes Global

Create item Actions

Scan: [Table] Student: Roll-no

Scan [Table] Student: Roll-no

+ Add filter

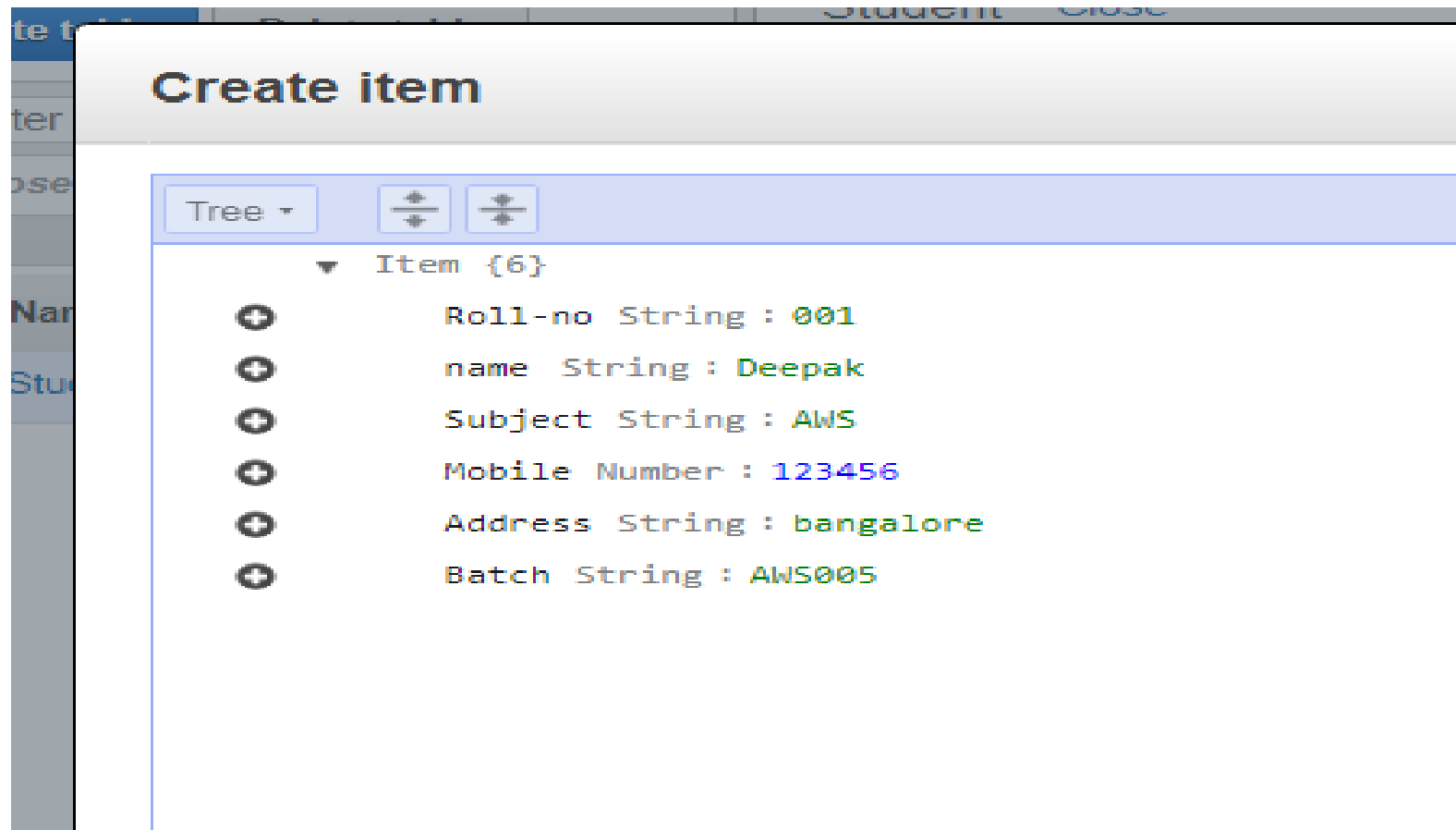
Start search

Roll-no

An item consists of one or more attributes. Each attribute consists of a name required are those that make up the primary key. [More info](#)

DynamoDB Lab

Create items—Give multiple input ---- + Append –String ----Save
Create around 5 Rows like this



The screenshot shows the 'Create item' dialog box in the AWS DynamoDB console. The dialog has a title bar 'Create item' and a toolbar with a 'Tree' dropdown and two icons for adding and removing attributes. The main area displays a list of attributes for a new item, each preceded by a plus icon in a circle. The attributes are:

- Roll-no String : 001
- name String : Deepak
- Subject String : AWS
- Mobile Number : 123456
- Address String : bangalore
- Batch String : AWS005

DynamoDB Lab

We can filter and search the required data only

Create itemActions ▾

Scan: [Table] Student: Roll-no ^

Scan ▾

[Table] Student: Roll-no ▾

^

Filter

address

String ▾

= ▾

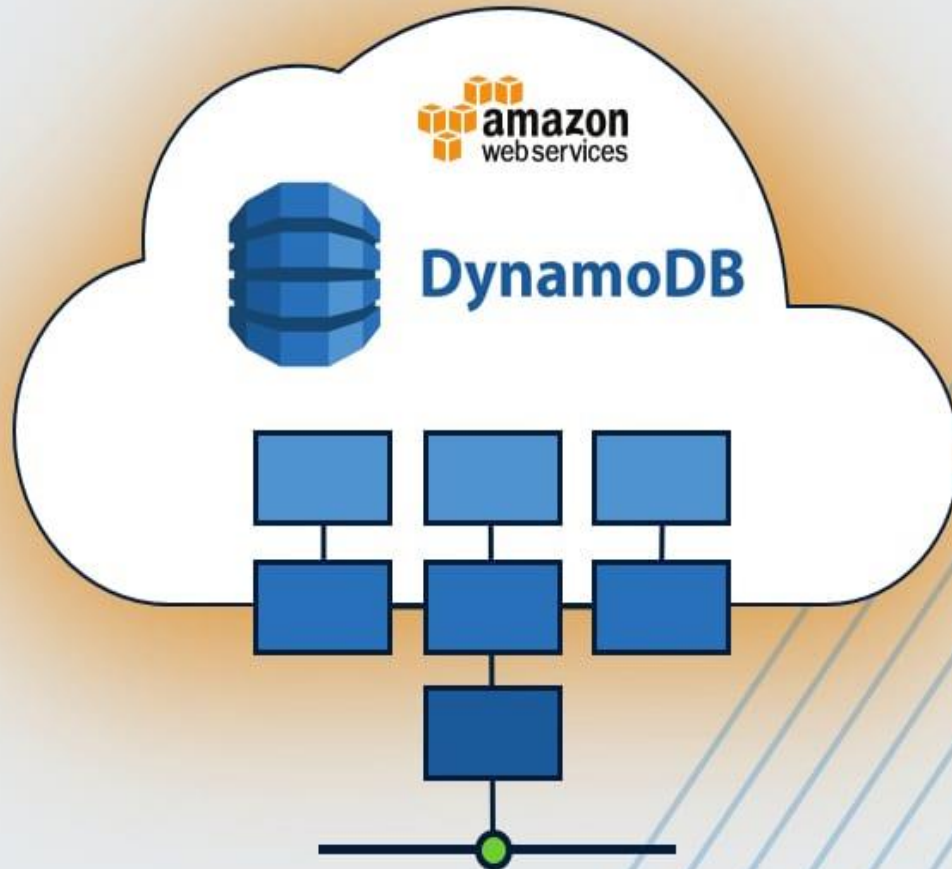
bangalore

✕

⊕ Add filter

Start search

Amazon DynamoDB Backup Service



Point-in-time Recovery

DynamoDB maintains continuous backups of your table for the last 35 days. [Learn more](#)

Status DISABLED **Enable**

Earliest restore date -

Latest restore date -

Restore to point-in-time

Create backup

Restore backup

Delete backup



Filter by backup name X

Time Range

Last 30 Days



Backup type

All backups



1 to 1 of 1 Backups



Backup name



Status



Creation time



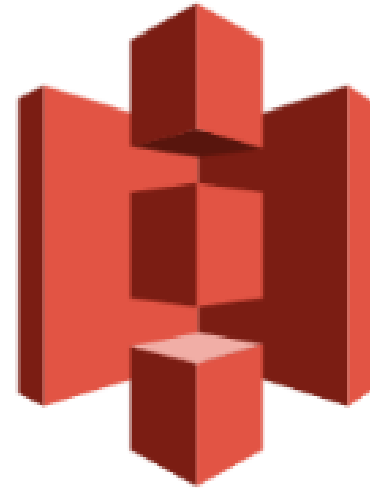
MusicBackup

Available

February 12, 2020 at 3:04:59 PM UTC-8



Amazon DynamoDB



Amazon S3

DynamoDB
test_results



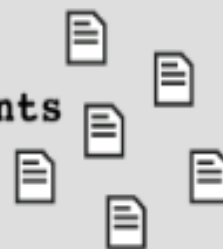
data

Backup Script

S3 Bucket



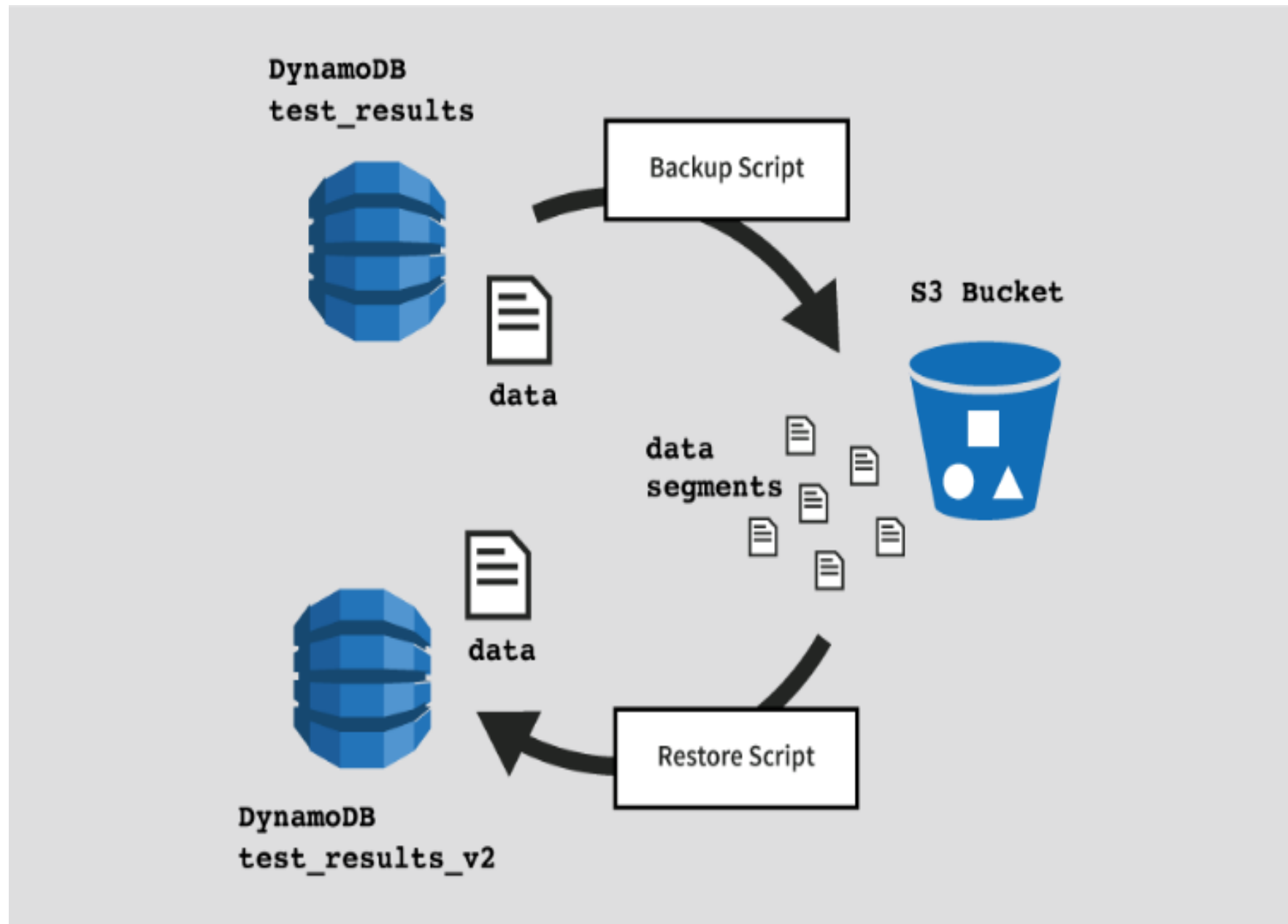
data
segments



data

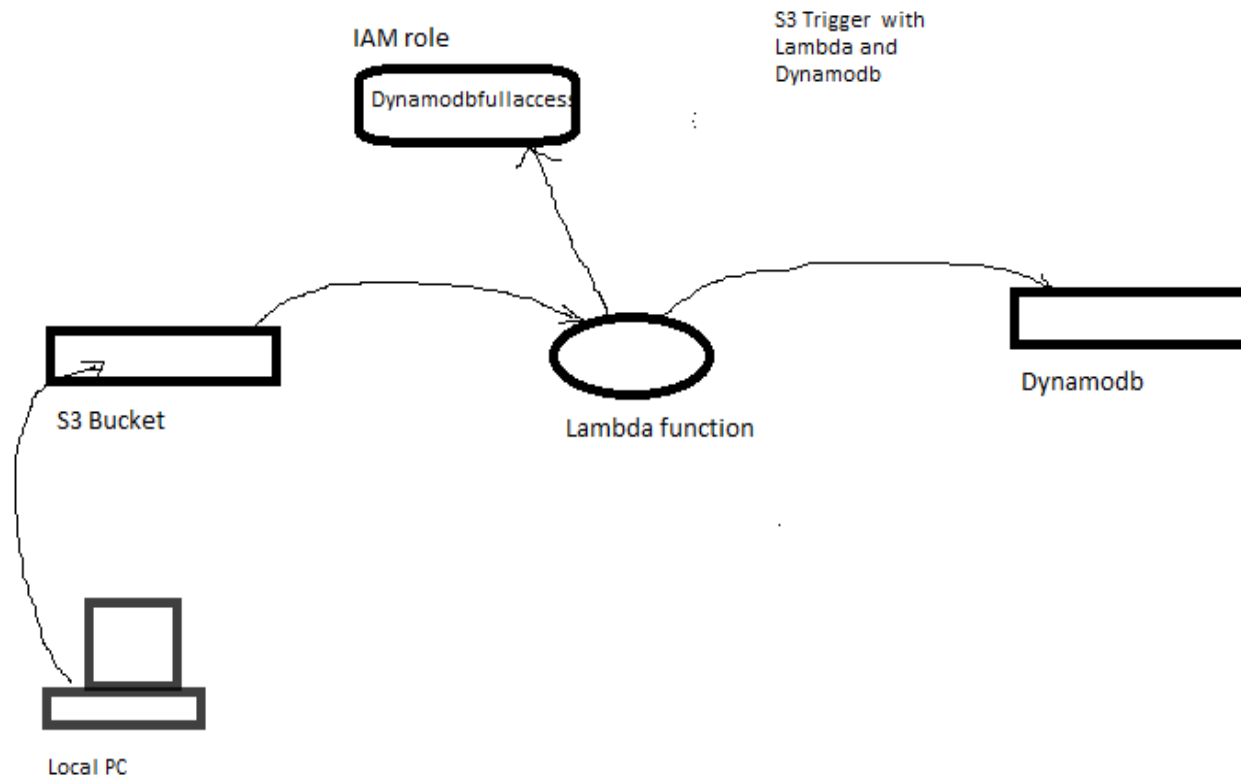
Restore Script

DynamoDB
test_results_v2



Lambda Lab --4

4) Create Lambda function to get trigger from s3 bucket and store the data in dynamodb.



Lambda Lab –4 --Steps

- 4) Create Lambda function to get trigger from s3 bucket and store the data in dynamodb.
- 1) Create one IAM role having Dynamodbfull access role.
- 2) Create one S3 bucket
- 3) Lambda-Create Function--After creating--Paste lambda code --Configuration--Add Trigger --S3 --fill detail --
- 4) Create Dynamodb table with name: newtable and Primary key-Unique --Go to items
- 5) Upload some files in S3 and check in Dynamodb

Lambda Lab –4--Code

```
import boto3
from uuid import uuid4
def lambda_handler(event, context):
    s3 = boto3.client("s3")
    dynamodb = boto3.resource('dynamodb')
    for record in event['Records']:
        bucket_name = record['s3']['bucket']['name']
        object_key = record['s3']['object']['key']
        size = record['s3']['object'].get('size', -1)
        event_name = record['eventName']
        event_time = record['eventTime']
        dynamoTable = dynamodb.Table('deepaktable')
        dynamoTable.put_item(
            Item={'deepakprimarykey': str(uuid4()), 'Bucket': bucket_name, 'Object':
object_key, 'Size': size, 'Event': event_name, 'EventTime': event_time})
```