

# BTS – MBDS

## Big Data Infrastructure

A3: AWS - DynamoDB

Ennio Maldonado  
03 March, 2021

# DynamoDB - Example Table

## Terraform script

```
main.tf
1 resource "aws_dynamodb_table" "basic-dynamodb-table" {
2   name           = "GameScores"
3   billing_mode    = "PROVISIONED"
4   read_capacity   = 20
5   write_capacity  = 20
6   hash_key        = "UserId"
7   range_key       = "GameTitle"
8
9   attribute {
10    name = "UserId"
11    type = "S"
12  }
13
14  attribute {
15    name = "GameTitle"
16    type = "S"
17  }
18
19  attribute {
20    name = "TopScore"
21    type = "N"
22  }
23
24  ttl {
25    attribute_name = "TimeToExist"
26    enabled        = false
27  }
28
29  global_secondary_index {
30    name           = "GameTitleIndex"
31    hash_key        = "GameTitle"
32    range_key       = "TopScore"
33    write_capacity  = 10
34    read_capacity   = 10
35    projection_type = "INCLUDE"
36    non_key_attributes = ["UserId"]
37  }
38
39  tags = {
40    Name           = "dynamodb-table-1"
41    Environment    = "production"
42  }
43 }
```

Terraform Apply -->

## Terraform Init

```
> terraform init
Initializing modules...
- disabled_dynamodb_table in ../..
- dynamodb_table in ../..

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/random v3.1.0...
- Installed hashicorp/random v3.1.0 (signed by HashiCorp)
- Installing hashicorp/aws v3.30.0...
- Installed hashicorp/aws v3.30.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
> terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_dynamodb_table.basic-dynamodb-table will be created
+ resource "aws_dynamodb_table" "basic-dynamodb-table" {
+   arn                = (known after apply)
+   billing_mode       = "PROVISIONED"
+   hash_key           = "UserId"
+   id                 = (known after apply)
+   name               = "GameScores"
+   range_key          = "GameTitle"
+   read_capacity      = 20
+   stream_arn         = (known after apply)
+   stream_label       = (known after apply)
+   stream_view_type   = (known after apply)
+   tags               = {
+     "Environment" = "production"
+     "Name"        = "dynamodb-table-1"
+   }
+   write_capacity     = 20
+
+   attribute {
+     name = "GameTitle"
+     type = "S"
+   }
+
+   attribute {
+     name = "TopScore"
+     type = "N"
+   }
+
+   attribute {
+     name = "UserId"
+     type = "S"
+   }
+
+   global_secondary_index {
+     hash_key        = "GameTitle"
+     name            = "GameTitleIndex"
+     non_key_attributes = [
+       "UserId",
+     ]
+     projection_type = "INCLUDE"
+     range_key       = "TopScore"
+     read_capacity   = 10
+     write_capacity  = 10
+   }
+
+   point_in_time_recovery {
+     enabled = (known after apply)
+   }
+
+   server_side_encryption {
+     enabled = (known after apply)
+     kms_key_arn = (known after apply)
+   }
+
+   ttl {
+     attribute_name = "TimeToExist"
+     enabled        = false
+   }
+ }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
```

# DynamoDB – Example Table

Terraform Output:

```
aws_dynamodb_table.basic-dynamodb-table: Creating...
aws_dynamodb_table.basic-dynamodb-table: Still creating... [10s elapsed]
aws_dynamodb_table.basic-dynamodb-table: Creation complete after 18s [id=GameScores]
```

Confirmed creation on DynamoDB dashboard

Filter by table name <input type="text"/> X Choose a table group <input type="text"/> Actions <input type="button" value="i"/>						
1 to 1 of 1 Tables << < > >>						
Name	Status	Partition key	Sort key	Indexes	Total read c	
<input type="radio"/> GameScores	Active	UserId (String)	GameTitle (String)	1	30	

GameScores Close

Overview Items Metrics Alarms Capacity Indexes Global Tables More

Manage DynamoDB stream

Table details

Table name	GameScores
Primary partition key	UserId (String)
Primary sort key	GameTitle (String)
Point-in-time recovery	DISABLED <a href="#">Enable</a>
Encryption Type	DEFAULT <a href="#">Manage Encryption</a>
KMS Master Key ARN	Not Applicable
Encryption Status	
CloudWatch Contributor Insights	DISABLED <a href="#">Manage Contributor Insights</a> <b>NEW</b>
Time to live attribute	DISABLED <a href="#">Manage TTL</a>
Table status	Active
Creation date	March 3, 2021 at 9:08:25 AM UTC+1
Read/write capacity mode	Provisioned
Last change to on-demand mode	-
Provisioned read capacity units	20 (Auto Scaling Error)
Provisioned write capacity units	20 (Auto Scaling Error)
Last decrease time	-
Last increase time	-
Storage size (in bytes)	0 bytes
Item count	0 <a href="#">Manage live count</a>
Region	US East (N. Virginia)
Amazon Resource Name (ARN)	arn:aws:dynamodb:us-east-1:662771546832:table/GameScores

Added a record to the table

```
> aws dynamodb put-item --table-name GameScores --item '{ "UserId": {"S": "BTS-007"}, "GameTitle": {"S": "Fortnite"}, "TopScore": {"N": "100"} }' --return-consumed-capacity TOTAL
```

# DynamoDB – Querying

```
> aws dynamodb scan --table-name GameScores
{
  "Items": [
    {
      "UserId": {
        "S": "BTS-007"
      },
      "GameTitle": {
        "S": "Fortnite"
      },
      "TopScore": {
        "N": "100"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 1,
  "ConsumedCapacity": null
}
```

# DynamoDB – Deleting resources

```
> terraform destroy
```

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

- destroy

Terraform will perform the following actions:

# **aws\_dynamodb\_table.basic-dynamodb-table** will be **destroyed**

```
- resource "aws_dynamodb_table" "basic-dynamodb-table" {  
  - arn          = "arn:aws:dynamodb:us-east-1:662771546832:table/GameScores" -> null  
  - billing_mode = "PROVISIONED" -> null  
  - hash_key    = "UserId" -> null  
  - id          = "GameScores" -> null  
  - name        = "GameScores" -> null  
  - range_key   = "GameTitle" -> null  
  - read_capacity = 20 -> null  
  - stream_enabled = false -> null  
  - tags         = {
```

# MongoDB on DataBricks

Connect MongoDB to DataBricks followed this guide:

<https://docs.databricks.com/data/data-sources/mongodb.html>

Quickstartv2

Edit

Clone

Restart

Terminate

Delete

Configuration

Notebooks (1)

Libraries

Event Log

Spark UI

Driver Logs

Metrics

Apps

Spark Cluster UI - Master ▼

Uninstall

Install New

<input type="checkbox"/>	Name	Type	Status	Source
<input type="checkbox"/>	org.mongodb.spark:mongo-spark-connector_2....	Maven	● Installed	

Quickstartv2

Edit

Clone

Restart

Terminate

Delete

Configuration

Notebooks (1)

Libraries

Event Log

Spark UI

Driver Logs

Metrics

Ap

Databricks Runtime Version

7.5 (includes Apache Spark 3.0.1, Scala 2.12)

New This Runtime version supports only Python 3.

Driver Type

Community Optimized 15.3 GB Memory, 2 Cores, 1 DBU

Instance

Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period. For more configuration options, please upgrade your Databricks subscription.

Instances

Spark

JDBC/ODBC

Permissions

Spark Config ⓘ

spark.mongodb.input.uri  
mongodb+srv://admin:admin@cluster0.u6kds.mongodb.net/sample\_airbnb?retryWrites=true&w=majority  
spark.mongodb.output.uri  
mongodb+srv://admin:admin@cluster0.u6kds.mongodb.net/sample\_airbnb?retryWrites=true&w=majority  
spark.databricks.delta.preview.enabled true

Environment Variables ⓘ

PYSPARK\_PYTHON=/databricks/python3/bin/python3