Back up and restore

1. DynamoDB -> table -> items tab
2. Cloud9 terminal:
    - aws dynamodb create-backup --table-name Music --backup-name MusicBackup

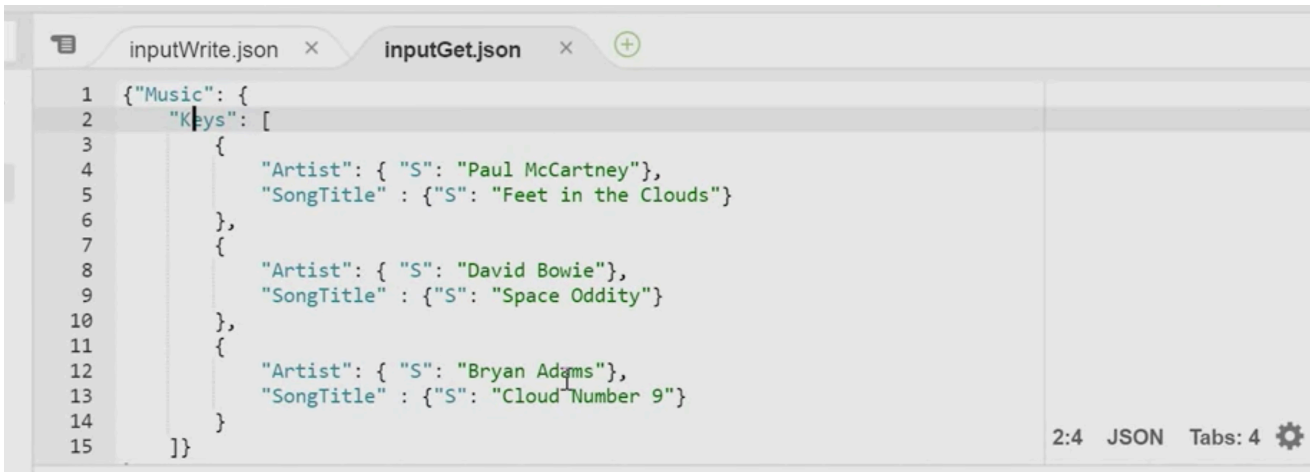
```
Result:

{
    "BackupDetails": {
        "BackupArn": "arn:aws:dynamodb:us-east-
1:766491063190:table/Music/backup/01594002161412-39a56be8",
        "BackupName": "MusicBackup",
        "BackupSizeBytes": 332,
        "BackupStatus": "CREATING",
        "BackupType": "USER",
        "BackupCreationDateTime": 1594002161.412
    }
}
```

    - aws dynamodb describe-backup --backup-arn arn:aws:dynamodb:us-east-1:766491063190:table/Music/backup/01594002161412-39a56be8
    - **restore the table:** aws dynamodb restore-table-from-backup --target-table-name MusicRestored --backup-arn arn:aws:dynamodb:us-east-1:766491063190:table/Music/backup/01594002161412-39a56be8
    - **see the resotred table:** aws dynamodb describe-table --table-name MusicRestored
    - **delete the table:** aws dynamodb delete-table --table-name dragons

DynamoDB API working with Datasets

1. **scan with a limit:** aws dynamodb scan --table-name Music --max-items 3
2. **batch write items:** dynamodb batch-write-item --request-items file://inputWrite.json

3. **batch get items:** aws dynamodb batch-get-item --request-items file://inputGet.json



Week 3 Notes and Resources

KEY TOPICS

Backup and Recovery, the SDK, and Monitoring and Performance

This week, we focused how to create and restore backups of your DynamoDB tables as well as dove a bit deeper in the DynamoDB SDK. We also looked into using tools such as Cloudwatch to monitor and optimize your DynamoDB performance.

Backup and Recovery

Backups

When you create an on-demand backup, a time marker of the request is cataloged. The backup is created asynchronously by applying all changes until the time of the request to the last full table snapshot. Backup requests are processed instantaneously and become available for restore within minutes.

Point-in-Time Recovery for DynamoDB

You can enable point-in-time recovery as well as create on-demand backups for your Amazon DynamoDB tables. For more information regarding on-demand backups, see On-Demand Backup and Restore for DynamoDB

Point-in-time recovery helps protect your Amazon DynamoDB tables from accidental write or delete operations. With point in time recovery, you don't have to worry about creating, maintaining, or scheduling on-demand backups. For example, suppose that a test script writes accidentally to a production DynamoDB table. With point-in-time recovery, you can restore that table to any point in time during the last 35 days. DynamoDB maintains incremental backups of your table.

Amazon DynamoDB SDK

Every AWS SDK provides one or more programmatic interfaces for working with DynamoDB. These interfaces range from simple low-level DynamoDB wrappers to object-oriented persistence layers. The available interfaces vary depending on the AWS SDK and programming language that you use. See Getting Started with DynamoDB SDK for details and examples.

Monitoring and Performance

Understanding Read and Write Capacity

Amazon DynamoDB has two read/write capacity modes for processing reads and writes on your tables:

- On-demand
- Provisioned (default, free-tier eligible)

The read/write capacity mode controls how you are charged for read and write throughput and how you manage capacity. You can set the read/write capacity mode when creating a table or you can change it later.

On-Demand Mode

Amazon DynamoDB on-demand is a flexible billing option capable of serving thousands of requests per second without capacity planning. DynamoDB on-demand offers pay-per-request pricing for read and write requests so that you pay only for what you use.

When you choose on-demand mode, DynamoDB instantly accommodates your workloads as they ramp up or down to any previously reached traffic level. If a workload’s traffic level hits a new peak, DynamoDB adapts rapidly to accommodate the workload. Tables that use on-demand mode deliver the same single-digit millisecond latency, service-level agreement (SLA) commitment, and security that DynamoDB already offers. You can choose on-demand for both new and existing tables and you can continue using the existing DynamoDB APIs without changing code.

Provisioned Mode

If you choose provisioned mode, you specify the number of reads and writes per second that you require for your application. You can use auto scaling to adjust your table's provisioned capacity automatically in response to traffic changes. This helps you govern your DynamoDB use to stay at or below a defined request rate in order to obtain cost predictability.

Provisioned mode is a good option if any of the following are true:

- You have predictable application traffic.
- You run applications whose traffic is consistent or ramps gradually.
- You can forecast capacity requirements to control costs.

Adaptive Capacity

It's not always possible to distribute read and write activity evenly all the time. When data access is imbalanced, a "hot" partition can receive such a higher volume of read and write traffic compared to other partitions. In extreme cases, throttling can occur if a single partition receives more than 3,000 RCUs or 1,000 WCUs.

To better accommodate uneven access patterns, DynamoDB adaptive capacity enables your application to continue reading and writing to hot partitions without being throttled, provided that traffic does not exceed your table's total provisioned capacity or the partition maximum capacity. Adaptive capacity works by automatically and instantly increasing throughput capacity for partitions that receive more traffic.

Adaptive capacity is enabled automatically for every DynamoDB table, so you don't need to explicitly enable or disable it.

As of 23-May-2019, DynamoDB Adaptive Capacity is now instant .

Monitoring Tools

AWS provides tools that you can use to monitor DynamoDB. You can configure some of these tools to do the monitoring for you; some require manual intervention. We recommend that you automate monitoring tasks as much as possible.

AWS X-Ray is a service that collects data about requests that your application serves, and provides tools you can use to view, filter, and gain insights into that data to identify issues and opportunities for optimization. For any traced request to your application, you can see detailed information not only about the request and response, but also about calls that your application makes to downstream AWS resources, microservices, databases and HTTP web APIs.

WHAT YOU ACCOMPLISHED THIS WEEK

- You used the AWS SDK to continue to build out your application.
- You used Cloudwatch Metrics and Alarms as well as Cloudwatch Logs.
- You used AWS x-Ray to detect and dive into performance issues.